SimCon: Simulated Context Sources for the Rapid Evaluation of Context-Aware Entities using Virtual Reality

First Author Name (Blank for Blind Review)
Affiliation (Blank for Blind Review)
Address (Blank for Blind Review)
e-mail address (Blank for Blind Review)

First Author Name (Blank for Blind Review)
Affiliation (Blank for Blind Review)
Address (Blank for Blind Review)
e-mail address (Blank for Blind Review)

First Author Name (Blank for Blind Review)
Affiliation (Blank for Blind Review)
Address (Blank for Blind Review)
e-mail address (Blank for Blind Review)

ABSTRACT

This paper describes the Platform For User Centric Design and Evaluation of Context-Aware Systems for rapid evaluation of context-aware entities using Virtual Reality. Specifically it details an experimental evaluation of the SimCon (Simulated Context) Tool Set for configuring and placing simulated context sources within a virtual environment. As a basis for this work, a high level model of the stages of context flow is described. Location tracking context sources are identified as the most appropriate for simulation within the virtual environment. Simulated context sources are modelled using the sensor modelling language (sensorML). We evaluate both the accuracy of the context simulations against their real world counterparts and the effort involved in placing and configuring simulated context sources. We show that for early rapid evaluation of context-aware entities which use location, there are savings in both time and effort using this approach that are of benefit to context-aware entity developers.

Author Keywords

Ubiquitous computing, context-aware, pervasive, evaluation, 3D virtual environment.

ACM Classification Keywords

Design, Reliability, Experimentation, Verification.

INTRODUCTION

Context-aware entities aim to do the right thing at the right time automatically for users [1]. To meet this requirement, a context-aware entity must match its own view of context with that of the users [2]. That is, if a user perceives a change to their situation (a situation is a higher abstraction of context [3]) differently than the context-aware entity perceives this change, (either due to delays in the process of context delivery to the application, sensor inaccuracy or incorrect inference) then the context aware entity may adapt its function at an inappropriate time or in an inappropriate way.

UbiComp 2009, Sep 30 – Oct 3, 2009, Orlando, Florida, USA.

Such evaluation is a non trivial matter [4]. Ubiquitous environments contain within them many interacting systems each with their own domain experts providing their own subjective views on “context”[5], from architects involved in the design of smart buildings, to the sensor experts, middleware experts, and finally to the designers and evaluators of entities which will adapt their function as a result of these sensor networks, in situ.

In addition, the building of ubiquitous environments involves the acquisition, installation and maintenance (cabling, power and configuration) of context sources (e.g. sensors) within those environments, presenting a considerable financial risk (changes to the underlying infrastructure of the building are often not possible) to both developers and evaluators if the desired level of context-awareness is not achieved for the user. Field-based evaluation itself may also require coordination of a number of participants moving around large environments, interacting with heterogeneous context sources and context-aware entities. These are both time consuming to organise and may be disruptive to the existing use of space [4].

Due to time and financial constraints in the development life cycle of context-aware systems, often the opportunity to conduct the kind of user-centred evaluations required to assess pervasive applications effectiveness in a wide range of situations is not possible [6]. A suite of tools, that meet the objectives of and are accessible to all interested parties, are required to conduct affordable early rapid prototyping of ubiquitous environments [7] [8],[9].

To cut costs and time to deployment, virtual reality simulation platforms have been identified for developing and evaluating such systems early on in the development life cycle [10], [11], [12], [13]. The Platform For User Centric Design and Evaluation of Context-Aware Systems (PUDECAS) [14] is distinguished from existing ubiquitous computing simulation approaches in the focus on providing a flexible, extensible and easy to configure platform for integrating Context Aware Entities seamlessly into the rapid prototyping process.

Here we present an evaluation of the SimCon (Simulated Context) Tool Set for the rapid evaluation of context-aware entities using virtual reality (VR). By hiding the underlying complexity of ubiquitous environments [19], the SimCon Tool enables evaluators to easily and rapidly configure context sources and place them within the PUDECAS VR Environment [14]. It also provides novel visualisation of location-based context within those environments to enhance the evaluation process of location based context-aware entities.

The platform does not set out to replace real world evaluations with real world context sources, but rather, through the use of VR, provide a means for rapid, repeatable evaluations over a range of situations and environments to reduce development costs and increase the predictability of deployed application behaviour. The design objectives of SimCon were to ensure the platform is:

i. Supportive of rapid prototyping: the process of deploying and configuring a context source within the virtual environment should be accessible and provide gains in time over the equivalent set up in a real world scenario.

ii. Effective: simulated context source must reflect real context sources to a level sufficient to meet experimental goals.

iii. Flexible: to handle the simulation of a range of context sources.

iv. Extensible: to deal with future growth of the platform.

The paper is structured as follows. To begin the simulated context model is described, first by introducing a refined definition of context appropriate to this paper, building on this to create a model of different stages of context processing, finally the modeling and simulation of context sources. Then the SimCon Tool Set is described (configuration and visualisation). Next a brief description of what is meant by a context aware entity is given followed by the types of sensors being simulated. Then the evaluation of SimCon: first a comparison of the simulated context sources against real world counterparts, and secondly, a usability evaluation of the tools for placing and configuring context sources. Lastly a section on related work is given before a conclusion and future work.

SIMCON CONTEXT MODELLING

Before we begin describing the process of simulating context within VR environments, an understanding of context is required. We begin by providing our own user-centric definition of context.

A User-Centric Context Definition

A precise definition of context within the scientific community is still an open discussion and a number of different viewpoints are common [5]. Within the Ubiquitous Domain an often cited definition is that of Dey and Abowd [15]:

"Context is any information that can be used to characterize the situation of an entity. An entity is a person, place, or object that is considered relevant to the interaction between a user and an application, including the user and applications themselves."

For the purpose of evaluating entities which make use of context, there must be some qualitative benefit for a user with regard to the function the entity provides them. For this reason an addition to Dey and Abowd's definition is introduced here:

Context is any information that can be used to characterize the situation of an entity to improve on its function with respect to a user's goals whilst minimizing the impact on its function with respect to other users' goals.

By creating a dependency between context and goals, this definition of what it means to be context-aware narrows the boundaries of the domain of discourse. When developing and evaluating a context-aware system the classification of an entity as context-aware is dependent on it achieving its aims with respect to the goals of the user as situated in a specific environment. Given this definition of context, the next section will specify a process model to capture the flow of simulated context.

Simulated Context Flow Model

A number of research papers have explored modelling context [16]. [17] and [18] both capture the concept of stages in context processing, upon which we base our simulated context model which captures the flow of context from a VR environment, up until the point of consumption by an entity. (Fig 1) shows this model, which divides context into four distinct categories: a Contum, Conditional, Combined and Consumed Context.

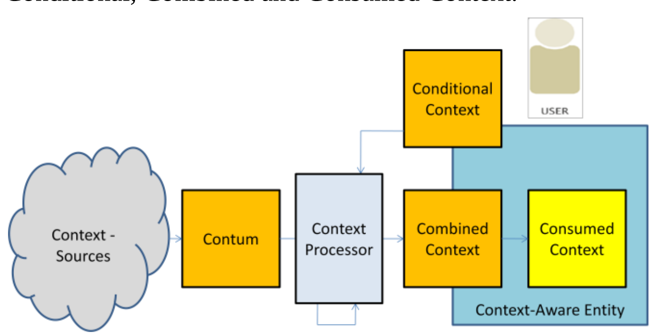

Figure 1 Context Flow Model

A Contum is the smallest indivisible value of context possible and is created by a low level context source. A contum (a contraction of context quantum) is a discrete unit of information with an associated level of uncertainty.

Uncertainty must be introduced into any concept of context from sensed data as all sensors introduce uncertainty through the process of sensing the physical environment.

Conditional Context is provided by a user explicitly, generally through an interface, thus making the user a potential context source. An example of a conditional context might be a calendar entry which specifies where you will be located at a particular time.

Combined Context is any combination of contums, conditional and/or additional combined contexts either fused or aggregated. Combined context might take two related sensor readings (contums) and fuse them into a more accurate location or aggregate temperature and location into a combined context. Alternatively, it may make use of historical context. This would be either handled as part of the context-aware entity itself, or by some context processor (e.g. ConStruct[20] which fuses low level context sources). A context processor potentially handles multiple context sources (sensor deployments providing contums, mobile interfaces providing conditional context) and provide access to this context for multiple context-aware entities.

A Consumed Context is the final stage of context and reflects a “situation” [3]. It is the highest abstraction inferred from lower level contexts and is directly used by an entity to adapt its function. This is where processing must take place to produce some outcome which relates to the user’s goals. Next we describe how we model low level context sources (sensors) for the purpose of simulating the appropriate contums for each context source.

Modelling Simulated Context Sources

Any entity which provides context to a context aware entity is a Context Source [6]. A simulated context source can only reflect reality as accurately as it is modeled. A trade off is required between realism, the effort to create the model and the processing requirements of complex simulations (as these may introduce delays in the real time interaction with the VR environment, human avatars and the context-aware entities under evaluation).

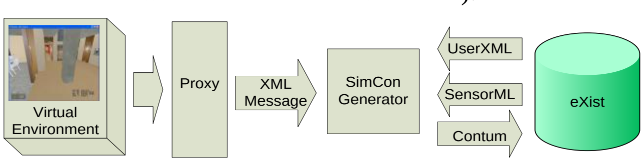

Figure 2 Data Flow

At minimum, the context-source simulation must be at a level of detail sufficient to support evaluation of a context-aware entity’s behavior.

The VR environment [14] maintains a global state of the world and generates XML encoded messages containing the precise location of the user’s avatar (Fig 2). As a result, we model low level context sources such as real-time location sensor systems, like Ubisense (see section on Location Systems), and simulate location-based contums.

We begin by capturing the most basic set of properties required to model these simulated context sources. The Sensor Modeling Language (sensorML) [21] provides a conceptual model for describing sensor systems. Based upon this model the following properties have been identified as essential to modeling simulated context sources based on sensors (table 1) for use within SimCon.

Sensor Property	Description
I.D.	Unique Identifier (Integer)
Type	Type of Context Source
Position	Origin (a Cartesian coordinate relative to an origin on a reference frame).
Envelope	The bounds of a sensor’s range.
Response Characteristics	A mapping of input (has a relation to location of context source) to expected output.
Output	A data value representation a measurement (coordinate, received signal strength etc.)
Accuracy	A measure of expected variation in observations/ measurements
Rate	The rate of rate at which a sensor produces observations/ measurements
Phenomena	The type of phenomena the sensor detects may influence placement.

Table 1. Sensor Properties for SimCon Sources

Optional properties include energy requirements (may influence response characteristics), connections (may influence placement) and costs, both capital and maintenance (may prohibit real time deployments).

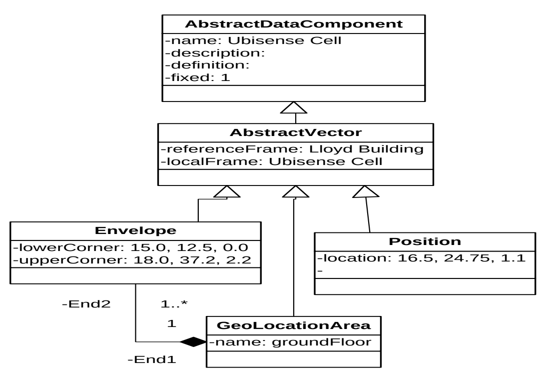

Figure 3 SensorML Description of Simulated Ubisense Cell Position

In addition to the conceptual model, sensorML defines an XML-based syntax for portable sensor description. To improve interoperability the SimCon Tool Set defines context sources using sensorML. Fig 3 illustrates how the area of an ubisense tag tracking sensor cell can be described in sensorML using UML notation for clarity. In practice these models are described in XML documents. Position describes the ubisense cell frame in terms of x, z and z coordinates. This in turn is referenced to the "ERI_FRAME" (which describes the geodetic position of the building).

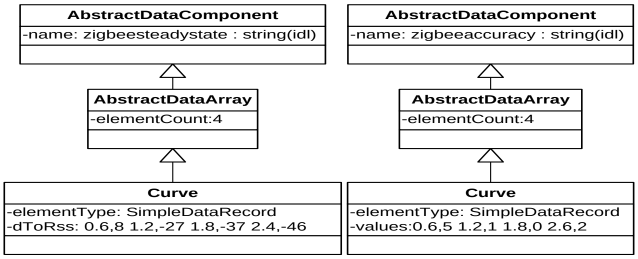

Figure 4: SensorML Description of the Steady State Response of a Zigbee Reciever and Accuracy model.

Fig 4 shows the "zigbeeTransceiverSteadyState" and "zigbeeAccuracy", together they give a steady state response curve which gives, for example, an expected received signal strength of 8 within 0.6 meters of the origin of the sensor with a standard deviation of a maximum of 5 (using a Gaussian distribution).

In the next section we will discuss the implementation of the Context Source Simulator, which uses the previous models to simulate Contums.

Context Source Simulation

The XML encoded data provided by the VR environment is accessed by SimCon Generator through a proxy (Fig 2). On startup, the SimCon Generator loads in all the sensorML sensor descriptions and userXML descriptions (see Section Configuration Tool) from an eXist database [22]. When a tagged avatars location falls within an appropriate bounded area, a Contum for that simulated context source is written back to the database (Fig 5).

We introduce noise using Gaussian noise [9]. This can be done to different levels of granularity around the origin of the simulated context source (Fig 5). For instance, a ubisense coordinate can be offset from its actual position, or a received signal strength can fall within a range around the expected steady state value. Ubisense simulation was modeled on specifications taken from the documentation provided (see section on Location Systems). ZigBee Transceiver mote simulation was modeled on calibration readings taken from within a lab in Blank. Readings were taken at distances of .6 meters from between transmitter and receiver up to 1.8 meter for each point. Once with the transmitter and receiver antennas aligned, once with the antenna at 90 degrees to each other and once with the antenna completely removed (as the antenna's orientation has an effect on the signal strength).

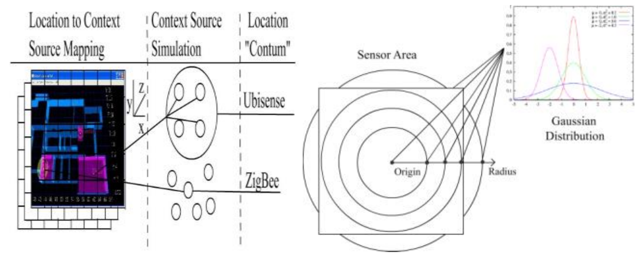

Figure 5 Location to Contum Mapping + Noise Model

We decided to simulate the motes with no antenna and with antenna (antenna orientation has considerable effect on signal strength, which is used for proximity detection), making modelling difficult (see Evaluation Section). With antenna, we took a mean of both the aligned and 90 degree readings, using the difference to simulate the movement of the antenna by distributing the outputs using a Gaussian function. If the location of the avatar does not change, no noise is introduced (it is assumed they are standing still).

SIMCON TOOL SET

The SimCon Tool Set allows an evaluator to place and configure simulated context sources and publish contums from those sources. An evaluator can then conduct rapid evaluations of context-aware entities when faced with varying degrees of context source fidelity. This is done by increasing or decreasing the accuracies and/or rates exhibited by a particular context source. The SimCon Tool Set consists of:

- Configuration tool: placement and configuration of simulated context sources.
- Visualisation tool: visualisation of sensor origin, bounded area, steady state/inaccuracy areas and real time visualisation of simulated context.

Configuration Tool

A UML Activity Diagram approach was found to be the most appropriate for capturing the first stage of context flow (see evaluation section). The user interface of the SimCon Configuration Tool is implemented using the Eclipse Graphical Modeling Framework (GMF). A user begins by creating a context simulator (Fig 6), which handles all context sources within a particular area (e.g. a building, or zone within a building). They then must specify an input of generated context, in this case the input is the Half Life 2 Simulator, and the type of generated context is location.

Next they must create one Simulated Context Source for each unique context source (type and id), each of which produces its own "Contum". A Simulated Context Source can either be configured directly through the properties section of the GMF tool or through a Java Standard Widget pop up by clicking twice on the Simulated Context Source. Through the widget the user can choose amongst a range of pre-configured Simulated Context Sources (see section

Location Systems) and either use the pre-set values, or customise these to meet their own requirements.

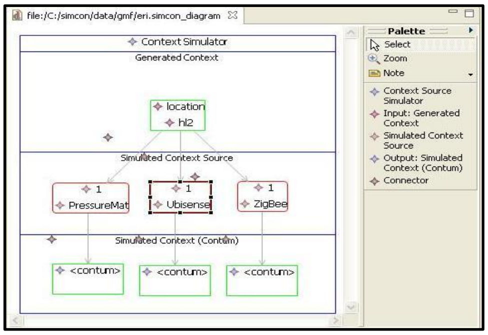

Figure 6: Simulated Context Source Configuration Tool

Each context source must be given an origin and a bounded area beyond which it does not function. In addition, areas of interference can be defined, which have an effect on a particular context source. For instance, by defining an area of noise within a Ubisense cell it is possible to increase the inaccuracies experienced in that area in the location context provided (by increasing the standard deviation of the inaccuracy model in those areas).For each avatar within the Virtual Environment a tag must also be specified for a context source (through the pop up widget). Users without such tags will not trigger events by that particular context source. This is stored in an XML file, currently called "userXML". The SimCon Configuration tool stores sensorML and userXML descriptions in eXist.

<userXML><name>Kris</name><contextSource><type> Ubisense</type><tagID>1</tagID></contextSource></use rXML>

Visualisation Tool

The SimCon Visualisation Tool (Fig 7) allows context source placement within the Virtual Environment by visualising the origin and boundaries of those context sources and providing a coordinate system. Secondly it provides real time feedback of users locations as they move around the environment, as well as novel visualisation of context (see evaluation section). (Fig 7) shows the three context sources, their bounded areas, origins and steady state response areas.

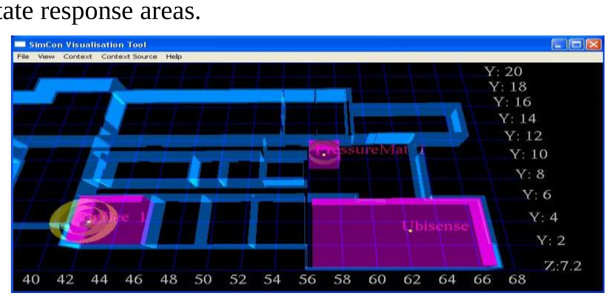

Figure 7: SimCon Visualisation Tool

The sensorML descriptions are accessed from eXist for additions or changes to these models in real time. In addition, changes in context are also displayed in real time (e.g. a tag position is represented as an avatar). Additional java code has been written to allow the movements of users within the environment (and where location is provided as a coordinate) to be replayed. This may then be displayed visually using the Visualisation tool for post experimental evaluations.

While the Virtual Reality environment is currently modeled in the Hammer editor [23] the process of building up the environment in the visualisation tool requires converting Computer Aided Design (CAD) data models into Visio files, which are then parsed for the relevant information (currently walls). Map files must also share a common coordinate system, or further calibration is required.

CONTEXT AWARE ENTITIES

The context provided by the SimCon Generator is utilised to evaluate context-aware entities. A context-aware entity is any entity which provides a function which may be improved using context, with relation to users' goals. A context aware entity can therefore be either an object (a door or application) or a space (a room).

An entity may either have its function improved by additional technology (actuators) and context, or the function of a software component of an existing application may be improved using context [6]. In real world environments a context-aware entity gathers information from context sources, either attached to the entity, or the entity has access to distributed context sources via a communications network.

As the context is currently saved in an eXist data base we have provided a proxy to this data base. A context-aware entity can connect to this proxy via TCP/IP, or connect directly to the eXist data base. Before we discuss the evaluation of SimCon, the next section will detail the sensor systems we deployed.

LOCATION SYSTEMS

A number of location systems exist providing location information in different forms [25]. We chose three types of location system which best suited the VR environment. These are the Ubisense Ultra Wide Band (UWB) Real Time Location System (RTLS), EM2420 ZigBee RF Transceiver Motes and x10 Pressure Mat (more detailed technical specifications relevant to this paper can be found here [33]).

Ubisense provides real time location information on user or equipment tags as they move within a cell in the form of cell number, x, y, z coordinate with reference to the sites origin, as well as a tag id and time stamp. The optimum accuracy of a sensed location is within 0.15 meters of its actual location; updates occur approximately every tenth of a second.

The ZigBee Transceiver Mote can be used to determine the proximity of a transmitter to a receiver using the received signal strength (RSS). The mote has been programmed to transmit an xml string containing this value every half a second. The RSS depends significantly on the orientation of the antenna. Objects near and around the transmitter and/or receiver also affect the RSS. In a static environment, output does not vary considerably.

To convert RSS to distance requires calibration of transceiver for the particular area it is in and programming the receiver with appropriate look up tables. Motes half a life span of roughly 27 hours and antennas are fragile. The X10 Pressure Mat detects when a circuit is open or closed due to weight being placed on the mat. This has been programmed to include the time of occurrence in the message update. An adult who places body weight on mat will activate the device.

SIMCON EVALUATION

As stated in the introduction we wished to determine how flexible the V.R. environment was when dealing with a number of simulated context sources. We chose 3 example context sources to meet objective (iii) and implemented them as simulated context sources within SimCon.

To meet objective (i) the usability of the tool set was evaluated, specifically the time taken to deploy and configure a simulated context source against the equivalent time to deploy a real world context source.

To meet objective (ii) the outputs gathered from simulated context sources were evaluated against the outputs of real world context sources in the same environment.

Before we detail these evaluations we will briefly introduce the three sensor systems which we evaluated as context.

SimCon Context Simulator Evaluation

We evaluated Pressure Mats, Ubisense and ZigBee transceivers data against the simulated context produced by the SimCon Context Simulator. These were conducted in a 12 x 6.5 meter room.

X10 Pressure Mats

A pressure mat was set up in a test room. A participant was required to walk over it, placing one foot on the mat using normal gait (distributing weight as they would when walking normally). This produced a context update which gave the mat id and time of occurrence. An area of floor in the Virtual Reality environment was then assigned as a pressure mat, when a user walks over this are an update is produced.

Ubisense

Simulated Ubisense context was evaluated against live sensor readings based upon a grid system within one particular cell. Each point upon the grid was placed 60 centimetres apart, with a total of 35 points. The grid was a rectangular shape (to meet the proportions of the room). First we examined readings taken from a tag placed at each point on the grid, and then a user held the tag over each point on the grid. The tag was updating (approximately every tenth of a second for 10 seconds).

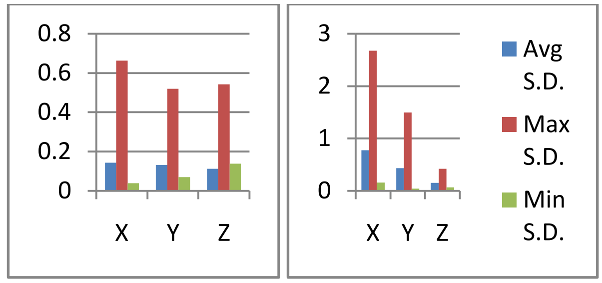

Chart 1. Live Ubisense Readings

Left of Chart 1 shows the mean standard deviation from the first set of 35 10 second readings, as well as a maximum deviation and minimum deviation from the mean. Right of Chart 1 shows the readings when a tag was held by a person. The Ubisense simulation was given a standard deviation of 0.15 meters (as given in Ubisense specification).

ZigBee Transceiver Motes

The ZigBee Transceiver mote was evaluated using the same grid structure as above. Twelve tables were placed together at the centre of the Lloyd common room. Each table is 120 x 60 centimetres. The tables were placed in two rows of six, giving a total width of 240 centimetres and total length of 360 centimetres. Readings were once again taken as per calibration for simulation.

The ZigBee receiver was placed at the exact centre point of these tables. Readings were then taken for each point, once with the transmitter and receiver antennas aligned, once with the antenna at 90 degrees to each other and once with the antenna completely removed (as the antenna's orientation has an effect on the signal strength).

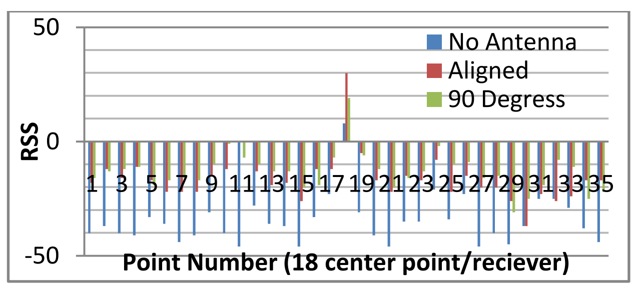

Chart 2. Live ZigBee Readings

Signal strength varied from 30 (antenna aligned, transmitter on receiver) to -46 (no antenna). The mean standard deviation was 0.3 for the set of readings with no antenna, 0.6 for those with antenna and 0.9 for those with antenna at 90 degrees.

Within the Virtual Reality simulator a simulated ZigBee receiver was placed in the same location (Fig 8). The receiver was given expected steady state outputs taken from earlier calibrations of ZigBee motes (see Context Source Simulator), first for no antenna, then for antenna with introduced inaccuracy to simulate antenna movement. The user's avatar was assigned a ZigBee transmitter and using the SimCon Visualisation Tool walked to corresponding points in the VR environment. For each point a reading was taken for the received signal strength between the transmitter and receiver. Table 2 shows the simulated readings taken for 4 points on 3 occasion against the real readings.

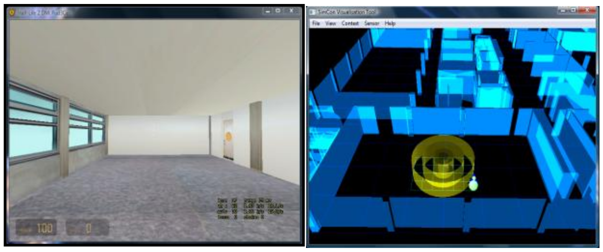

Figure 8 Virtual Reality + SimCon Visualisation

In Meters	Real	Sim1	Sim2	Sim3
Point 1: 0 - 0.6	27	26	17	31
Point 2: 0.6 – 1.2	-14	-6	-7	-7
Point 3: 1.2 – 1.8	-18	-15	-15	-15
Point 4: 1.8 – 2.4	-21	-18	-23	-19

Table 2: Simulated ZigBee v Real

Findings

The simulation of Ubisense fell very close to the specifications when the tag was left independent of a user. When a user was carrying the tag it was far less accurate. This in large part is due to the size of the room in which the ubisense cell was deployed and the use of only four sensors, meaning that the tag at times was visible to only two sensors. This caused considerable fluctuation of sensed location, often not updating location at all until the user moved to a new position. While it is possible to simulate greater inaccuracies by increasing the distribution, or defining areas of increased noise, the effects of body orientation and sensor coverage require more investigation.

The simulation of received signal strengths for ZigBee transceivers is challenging. To begin with, the environment and the placement of the receiver have an effect on the signal strength. Also, the signal strength of a transmitter when antennas are aligned and when antennas are not can vary due to the movement of users. While the removal of the antenna reduces the signal strength, packets were still being sent successfully, it allows for more accurate representations of steady state output as a user moves around (since orientation is no longer an issue), and hence improves the accuracy of the simulation.

We have also shown that with the addition of noise using Gaussian functions, it is possible to simulate antenna movement and create more realistic sensor outputs. While these will never fully reflect reality, they do improve the simulation process for the purpose of evaluating the impact of these types of issues. The x10 pressure mat is dependant on how it is deployed. Given various gaits and weights this introduces error. For the purpose of simulation we assumed a mat that is designed to cover a floor space that would minimise these errors (2 meters squared). As such, simulation was highly accurate.

SimCon Tool Set Evaluation

The SimCon Tool Set has been developed and evaluated in an incremental process. The purpose of the SimCon Tool set is to provide an accessible (for non domain experts), efficient method for evaluating context-aware applications. For this purpose the usability of the tool is of key importance.

The usability was evaluated under the following metrics:

- Qualitiative: efficiency, errors, satisfaction.
- Quantitative: error rate, success rate and time to complete a task.

For each evaluation participants accessed the software and Virtual Environment on a laptop. The evaluator sat and examined their behaviour, taking notes and providing assistance when asked, or when the participant made an error from which they could not recover.

The participants had an on-line questionnaire with pre and post question [23]. The questionnaire is based on guidelines as set down by Lumsden [23] and the layout as recommended by Tullis et al [23], which is modelled on the System Usability Scale (SUS). Also explained here were some of the basic concepts which were needed to conduct the experiment as well as details on the tasks. The questionnaires are accessible here [26].

Evaluations

The 1st evaluation evaluated two different approaches to the efficient modelling of the context delivery process (to create a Contum) [23]. It was found that the UML activity diagram approach provided a flexible and extensible method to achieve this.

The 2nd Evaluation looked at usability issues in the first stand alone GMF prototype of the SimCon Tool. Specifically measuring how rapidly they could configure two heterogeneous context sources for creating Contums. Six test subjects (PhD Students ranging from novice to expert knowledge of location based context sources) were asked to model a Ubisense sensor cell and ZigBee transceiver. The first task was to define a sensor cell area for Ubisense and to then adjust accuracy and time delays. The second task was to place a ZigBee receiver and define a steady state response (distance to signal strength).

The 3rd Evaluation built upon these findings. It set out to determine usability issues for the SimCon Modelling Tool with additional Visualisation Tool. Eleven test subjects (7 with little or no experience of location context sources, 3 with intermediate knowledge and 1 with expert knowledge) from Blank and Blank Computer Science department were asked to complete an exercise using these tools. A learning exercise was chosen as test subjects were forced to begin thinking about the processes involved as they were aware that they would be examined at the end on elements of the exercise. It also allowed us to do preliminary evaluations of PUDECAS as a learning aid.

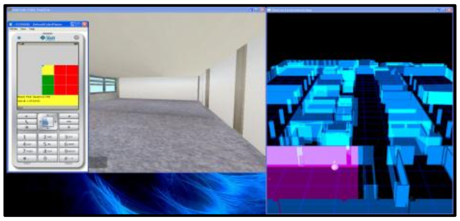

Figure 9 J2ME Maze Game

Test subjects were presented with an emulated J2ME Maze Game designed by four 4th year Information and Communications students studying in Blank. This was developed on PUDECAS and also evaluated using the physical Ubisense installation in the Blank building. The maze game itself is created and presented on the J2ME emulator 2D display (Fig 9), using the sensed location to present the user in the maze on the 2D display. They must then rely on this location to move around the maze by moving about in their sensed environment. If they step outside the given path they must start again. Not an easy task even for highly accurate locations, but given introduced inaccuracies by the sensing process, it becomes yet more difficult.

Participants in this experiment were to evaluate their application in order to find an appropriate envelope within which they could say the Maze Game would function satisfactorily. They were then required to set up and configure a Ubisense Cell within the Virtual Environment and to evaluate the J2ME Maze Game for differing levels of delay and accuracy of location updates. They also had access to the Visualisation Tool to see how changes in configuration of the sensors affected how the J2ME application "perceives" location.

The pre-questionnaire primed the test subjects by asking them about certain aspects of location-based sensor systems. An introduction explained some of the details of Ubisense and the J2ME Application. They then conducted a number of tasks and finally were asked post exercise questions, assessing if they had learnt anything about the Ubisense sensor system. They were then asked to assess the tool as an education aid.

Findings

The 1st evaluation is covered here [23]. The 2nd evaluation saw all participants completing the tasks, varying in times from 40 minutes to 1 hour 10 minutes. The majority of errors were due to features of the interface (position of buttons, tab features not being enabled). The terminology was at times confusing, for instance fidelity in reference to sensor accuracy. The introduction of steady state responses also caused some confusion for non-experts. Also, due to the extensive instructions the users did not engage with the evaluation as much as would have been hoped.

The newly implemented GMF interface made the process of setting up a sensor system an easy task for most users. While the questionnaire gave good indications about overall acceptance of the tool (participants found it to be over all a satisfying experience), the most valuable conclusions were drawn from observing them working and allowing them to discuss openly their experiences about the tool. The evaluation demonstrated that for a novice user it was possible to become familiar with the new tool interface and place and configure two different types of context source efficiently (with the use of instructions) within an environment.

The 3rd evaluation saw all participants complete the experiment ranging from 35 minutes to 55 minutes. The ability to load in preset values for the Ubisense system (e.g. the expected accuracy as taken from specifications) improved efficiency of task completion. Changes made in the terminology in both the interface and in the instructions, as well as changes in the layout of the interface, resulting in fewer errors than previous evaluations.

The use of the additional Visualisation Tool not only captured the participant's attention and improved their experience and satisfaction with the process; it also allowed them quickly evaluate how introduced delays and changes in accuracy of location had an affect on how this location was "viewed" by the context source in real time. This put the issue of jumpy inaccurate location and its effect on how the Maze Game performed quickly into perspective. There are still some issues with moving from tool to tool, in that the number of windows can become distracting to navigate. We are currently considering ways to improve this, possibly (as stated previously) by integrating the Visualisation Tool more into the GMF framework.

Another significant result was the retention of information about the Ubisense Sensor System. As the Usability Evaluation was framed as a learning exercise it allowed us to also assess this aspect of PUDECAS. It showed promising findings for novice and intermediate in discovering cursory facts about Ubisense (like costs, battery life of tags etc.) given in the introduction, although retention of these facts still requires evaluation.

Specifically, we wished to discover if the test subjects were better informed about how inaccuracy effects location based applications, and in the post questionnaires all participants showed either an improvement in their understanding of this concept, or reflected on the tools ability to elucidate such an understanding. Since we used the optimum readings given by Ubisense on their Sensory System there is some concern about skewing a user's perception of a Real World deployment of the Ubisense Sensor System. That is, while an expert user will be aware of the limitations of using Simulated Context Sources, a non expert may believe that this is exactly how a Simulated Context source will behave, which is not the case.

Further Results

In addition to these evaluations, PUDECAS was displayed in the HEA Transformations Exhibition in the Science Gallery in Blank as part of the NEMBES project [34]. A combination of live sensor data from the Environmental Research Institute and SimCon data was used to provide location context to an emulated mobile application as they interacted with a virtual representation of the same building.

RELATED WORK

Context Modelling

Various models have been proposed for representing context, among them logic based models, object oriented models and ontology based models [16]. Due to the properties of ontologies (knowledge sharing, extensibility) a number of research papers have looked at capturing context in ontology, employing a two tier approach [17], [27], having a core or upper ontology to define generic concepts (like space/location and person/user respectively) and a lower extensible ontology for adding to these generic concepts (like region/room).

Jang, et al [28] have developed a unified context 5W1H which loosens the coupling between services and context by making a distinction between types of context: preliminary, integrated, conditional and final. A similar approach is taken by Lee et al [17] and applies this to an owl context ontology. It does so by distinguishing between sensed context and combined context, as well as learned context and inferred context, thus capturing aspects of the flow of context.

Modelling Simulated Context Sources and Virtual Reality Test Beds

Bylund [10] distinguishes two types of context simulation tools: (i) a simulation suite that simulates context values and (ii) a semi-realistic simulation environment. Halteren [6] takes the former approach (i) and has looked at specifically simulating Quality of Context (QoC) issues. As context information represents real-world situations, QoC gives certain quality indicators, such as precision and decay [29], [30].

Taking the latter approach (ii), virtual reality simulations of pervasive computing environments have been used in a number of research efforts, specifically QuakeSim [10] and HPLab’s UbiWise[11] 3D Sim[12], UbiREAL[13] and PUDECAS[14]. These have demonstrated that 3D virtual reality computer game engines potentially provide a cost effective platform for simulating pervasive computing environments with sufficient realism to accurately test human interaction with pervasive computing software systems.

CONCLUSION

During the evaluation of the context sources output it was found that a perfect simulation of an environment would require modelling to a degree beyond which the return on investment makes it worth while, as every environment has its own unique signature on radio signals.

The strength of the simulation comes from its ability to do high level evaluations of context-aware entities which make use of heterogeneous sensing systems. It is possible to evaluate seemingly erratic behaviour by configuring context sources to distort outputs (using Gaussian noise), in order to determine an envelope in which a context-aware entity will behave correctly.

In conclusion, we have shown that the SimCon Tool Set provides an accessible, efficient, inexpensive (relatively) method for conducting rapid evaluations of context aware entities early in the development life cycle using a number of heterogeneous location based context sources.

FUTURE WORK

Since wireless sensors are highly dependant on their operating environment, our next goal is to incorporate more complex radio propagation models with SimCon to create more accurate simulated context sources. Gibney et Al [31] have a method for providing models of radio signal propagation within building spaces. It is also planned to enhance the visualisation aspects of SimCon, to enable the display of different types of context (e.g. temperature readings, air flow) to allow for real time visualisation of the impact of this changing context.

In addition, while SensorML has proved useful for modelling simulated sensors, both simulated and real world sensors, it is planned to incorporate SensorML concepts into the Industry Foundation Class [24] models typically used for architectural or civil engineering design of buildings or smart space. This should improve the over-all process of knowledge sharing within ubiquitous domain by using an open standard with which to describe ubiquitous environments. The ultimate goal is to create a suite of complimentary tools for all stages of the design process, from building plans, up to context aware entity evaluation within those spaces (automated of construction of the virtual environment and simulated context source models). Not only will such tools help with the design and evaluation, but will give different domain experts views into how their decisions impact on other areas of the process of designing and deploying smart spaces and their associated ubiquitous services.

ACKNOWLEDGMENTS

This work is supported by Blank under the Blank and by Blank under the Blank.

REFERENCES

[1] Scott C, Jennifer M. Prototypes in the wild: Lessons learned from evaluating three ubicomp systems. *IEEE Pervasive Computing* (2005).

[2] Schmidt, A. Ubiquitous Computing – Computing in Context. *A thesis on Context-Awareness, Context Aware Computing, and Ubiquitous Computing* (2002).

[3] Loke S. Representing and reasoning with situations for context-aware pervasive computing: a logic programming perspective, *The Knowledge Engineering Review* 2004, v.19 n.3, p.213-233.

[4] Scholtz, J. Consolvo, S. Toward a Framework for Evaluating Ubiquitous Computing Applications. *IEEE Pervasive Computing* 3, 2 (Apr. 2004), 82-88.

[5] Bazire, M. Brezillon, P. Understanding Context Before Using It. *Journal on Modeling and Using Context*, 2005, Page(s): 29 - 40.

[6] Ding, Broens, T. van Halteren, A. SimuContext: Simply Simulate Context. *Autonomic and Autonomous Systems*, 2006.

[7] Reilly, D. Dearman, D. Welsman-Dinelle, M. Inkpen, K. Evaluating early prototypes in context: trade-offs, challenges, and successes. *Pervasive Computing, IEEE* , vol.4, no.4, pp. 42-50, Oct.-Dec. 2005

[8] Weis, T. Knoll, M. Ulbrich, A. Muhl, G. and Brandle, A. Rapid Prototyping for Pervasive Applications. *IEEE Pervasive Computing* 6, 2 (Apr. 2007), 76-84.

[9] Li, Y. Hong, J. I. and Landay, A. Design Challenges and Principles for Wizard of Oz Testing of Location-Enhanced Applications. *IEEE Pervasive Computing* 6, 2 (Apr. 2007), 70-75.

[10] Bylund, M., Espinoza, F. Testing and Demonstrating Context-Aware Services with Quake III Arena. *Communications of the ACM*, (2002) Vol. 45, No 1, pp46-48.

[11] Barton, J. Vijayaraghavan. V. UbiWise, A Ubiquitous Wireless Infrastructure Simulation Environment. *HP technical report* (2002).

[12] Nazari, S Ali, A. Klar, F. 3DSim: Rapid Prototyping Ambient Intelligence. *SOc-EUSAI conference*. 2005.

[13] Nishikawa, H. Yamamoto, Ito, M. et Al.: UbiREAL: Realistic Smartspace Simulator for Systematic Testing. *Ubicomp 2006*: 459-476

[14] O'Neill, E. Lewis, D. McGlinn, K. Dobson, S. Rapid User-Centred Evaluation for Context-Aware Systems. *DSV-IS* 2006.

[15] Dey, A.K. Understanding and Using Context. *Personal Ubiquitous Computing*. 5, 1 (Jan. 2001), 4-7.

[16] Strang, T. Linnho-Popien, C. A Context Modelling Survey. *In: UbiComp 1st Int' Workshop on Advanced Context Modelling, Reasoning and Management* 2004.

[17] Lee, K. Kim, J. Lee, J. Lee, K. Implementation of Ontology Based Context-Awareness Framework for Ubiquitous Environment. *Multimedia and Ubiquitous Engineering*, 2007. MUE '07.

[18] Jang, S. Ko, E. Woo, W. Unified User-Centric Context: Who, Where, When, What, How and Why. *Personalized Context Modelling and Management for Ubicomp Applications* UbiPCMM 2005.

[19] Salber, D Dey, A.K Abowd, GD. The Context Toolkit:Aiding the development of context-enabled applications. In: *Proceedings of CHI'99*; 434–441

[20] Graeme, S. Lorcan, Paddy, N et Al. ConStruct-a Decentralised Context Infrastructure for Ubiquitous Computing Environments. *IT&T Conference*, 2005.

[21] Sensor Modelling Language (SensorML): http://vast.uah.edu/index.php

[22] eXist Open Source Native XML Database http://exist.sourceforge.net/

[23] McGlinn, K. O'Neill, E. Lewis. D. Modelling of Context and Context-Aware Services for Simulator Based Evaluation. Managing Ubiquitous Communications and Services (part of IM 2007)

[24] Industry Foundation Classes http://www.ifcwiki.org/index.php/Main_Page

[25] Hightower, J.; Borriello, G., Location systems for ubiquitous computing, *Computer* , vol.34, no.8, pp.57-66, Aug 2001

[26] Experimental Process. https://www.cs.tcd.ie/~mcglinnk/exp1,/exp2,/exp3.

[27] Wang, X. Dong, J. Chin, C. Hettiarachchi, S. Zhang, D. Semantic Space: An Infrastructure for Smart Spaces. *IEEE Pervasive Computing*, vol. 03, 2004.

[28] Yoosoo, O. Schmidt, A.; Woo, W. Designing, Developing, and Evaluating Context-Aware Systems. *Multimedia and Ubiquitous Engineering*, 2007.

[29] Wegdam, N. Sinderen, M. Trustworthiness and Quality of Context Information. *Young Computer Scientists*,. ICYCS 2008.

[30] Krause, M. Hochstatter, I. Challenges in Modelling and Using Quality of Context (QoC). *Mobility Aware Technologies and Applications*, 2005.

[31] Mc Gibney, A. Klepal, M. O'Donnell, J. Design of underlying network infrastructure of smart buildings. *Intelligent Environments, 2008 IET 4th International Conference on* , vol., no., pp.1-4, 21-22 July 2008

[32] Mara, P., McGlinn, K., O'Sullivan, D. et Al. Pervasive Knowledge-Based Networking for Maintenance Inspection in Smart Buildings. To Appear in MUCS 2009, Managing Ubiquitous Communications and Services.

[33] Sensor Types https://www.cs.tcd.ie/~mcglinnk/sensors/doc/sensor-types.doc