A Virtual Reality-based Context Simulator for Evaluating the Effect of Uncertainty on Smart Building Applications

A thesis submitted to the

University of Dublin, Trinity College

in fulfilment of the requirements for the degree of

Doctor of Philosophy

Kris McGlinn

Knowledge and Data Engineering Group,

School of Computer Science & Statistics,

Trinity College Dublin, Ireland,

Kris.McGlinn@scss.tcd.ie

Supervised by Dr. Dave Lewis

Co-supervised by Dr. Lucy Hederman

2012

Declaration

I declare that this thesis has not been submitted as an exercise for a degree at this or any other university and it is entirely my own work.

Signed: ____________________________

Kris McGlinn

Date

Permission to lend or copy

I agree to deposit this thesis in the University's open access institutional repository or allow the library to do so on my behalf, subject to Irish Copyright Legislation and Trinity College Library conditions of use and acknowledgement.

Signed: ___________________________

Kris McGlinn

Date

Acknowledgments

Abstract

Smart Building Applications (SBAs) adapt their behaviour in response to context information generated by sensors present in the smart building. SBA developers must however account for uncertainty in the accuracy of context information resulting from imperfection in the sensing technology and potentially complex interactions between the sensed phenomenon, the sensor and the physical environment. To address this challenge, Context Simulators are increasingly employed. These generate simulated context information based on an internal model of the building, it sensors and its occupants and their activities. Some simulators employ Virtual Reality (VR) environments, to more accurately capture the likely behaviour of users in a simulated smart building, and thereby model and simulate their interaction with sensors and, via the resulting context information, with SBAs. This type of evaluation can result in better designed SBAs by prototyping them in simulated conditions before being deployed into a real physical environment. A number of key challenges remain to be addressed however, before such a simulation-based SBA prototyping approach can become widely used. Firstly, any context simulator must be flexible in its approach to simulating context, so that it supports simulating different mixes of SBAs, context sources and smart buildings, using a selection of methods most suitable to the evaluation task at hand. Secondly, the context simulator must support simulating uncertainty in context. Lastly, the process of setting up a simulation should be efficient in use, satisfying to the developer and also, be effective in correctly evaluating the SBA behaviour.

This thesis presents the SimCon model which addresses these challenges by supporting simulation of context, in particular location context, with configurable levels of uncertainty for evaluating SBAs using visualisation. A comparative analysis of the state of the art is used to demonstrate the relative flexibility of the model. The SimCon model has been implemented in a simulation configuration and visualisation tool set which has undergone a number of evaluations with SBA developers to determine and improve its level of usability. Integration with existing building industry standards has

also been developed, to increase the leverage of such models and thereby the efficiency of the simulation configuration process.

Contents

Table of Tables

Table of Figures

Abbreviations

AI	Artificial Intelligence
AEC	Architecture, Engineering and Construction
API	Application Programming Interface
BIM	Building Information Model
BLC	Building Life Cycle
BLIS	Building Lifecycle Interoperable Software
BPMN	Business Process Modelling Notation
CAD	Computer Aided Design
CAVE	Cave Automatic Virtual Environment
CFD	Computational Fluid Dynamics
CIT	Cork Institute of Technology
CIF	Common Industry Format
CSUQ	Computer System Usability Questionnaire
CVE	Collaborative Virtual Environment
ECG	Electrocardiography
ERI	Environmental Research Institute
GLS	Generic Location Event Simulator
GMF	Graphical Modelling Framework
GML	Geographical Mark-up Language
GPS	Global Position Satellites
GSN	Global Sensor Network
GUID	Global Unique Identifier
GUI	Graphical User Interface
HL2	Half Life 2
HVAC	Heating Ventilation and Air Conditioning
ID	Identifier
IEEE	Institute of Electrical and Electronics Engineers
IDM	Information Delivery Manual

IFC	Industry Foundation Classes
ISO	International Organization for Standardization
ITS	Intelligent Transport System
JRE	Java Runtime Environment
KDEG	Knowledge and Data Engineering Group
LAN	Local Area Network
LCD	Liquid Crystal Display
MVD	Model View Definition
NASA	National Aeronautics and Space Administration
NIST	National Institute of Standards and Technology
OMG	Object Management Group
PAS	Publicly Available Specification
PDF	Probability Distribution Function
PIR	Passive Infrared Sensor
QUIS	Questionnaire for User Interaction Satisfaction
RCEU	Replaceable Code Emulation Unit
RF	Radio Frequency
RFID	Radio Frequency Identification
RSS	Received Signal Strength
SB	Smart Building
SBA	Smart Building Application
SENS	Sensor, Environment and Network Simulator
SensorML	Sensor Modelling Language
SDK	Software Development Kit
SL	Second Life
SQL	Structured Query Language p131
SWEET	Semantic Web for Earth and Environmental Terminology
SUMI	Software Usability Measurement Inventory
SUS	System Usability Scale
SWT	Standard Widget Toolkit

TCD	Trinity College Dublin
TEDS	Transducer Electronic Data Sheet
TCP/IP	Transmission Control Protocol/ Internet Protocol
UCD	University College Dublin
UCC-CE	University College Cork Civil Engineering
UML	Unified Modelling Notation
USE	Ease of Use Questionnaire
UWB	Ultra Wide Band
WLAN	Wireless Local Area Network
WoZ	Wizard of Oz
WSN	Wireless Sensor Network
VR	Virtual Reality
VRML	Virtual Reality Modelling Language
XML	Extensible Mark-up Language

Glossary

Accuracy: The closeness of agreement between a measured quantity value and a true quantity value

Avatar: A graphical representation of user in a simulated smart environment, which is controlled interactively by a human.

Bot: A simulated user of a smart environment.

Bot-driven context simulation: Context simulation which is affected by the behaviour of bots in the simulation environment.

Building Life Cycle (BLC): Defines the entire life of a building from inception and design, to construction, through operation and maintenance and on to eventual demolition/recycling (O'Sullivan, Keane et al. 2004).

Building Information Model (BIM): An integrated data model for storing all the information relevant to the BLC.

Context: Any information that can be used to characterise 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 (Addlesee, Jones et al. 1997).

Context Acquisition: A process of generating context information.

Context-Aware Application: Any application which makes use of context information.

Context Processing: A process of transforming low-level context into high-level context and situations.

Context Simulator: A piece of software which can generate simulated context.

Context Source: Any entity which generates context information relevant to a context-aware application (Broens and Halteren 2006).

Contum: A piece of discrete context information that is associated with a value of uncertainty.

High Level Context: Context information that is generated as a result of processing low level context. For example, fusing two low level contexts to form a new context or reasoning about a low level context with respect to some specific system state.

Interactive Context Simulation: A context simulator which supports user-driven context simulation i.e. simulated context which is affected by the interactions of a user with the simulation environment.

Low Level Context: Context generated by sensors.

Non-Interactive Context Simulation: A context simulator which does not support user-driven context simulation.

Precision: The closeness of agreement between indications or measured quantity values obtained by replicate measurements on the same or similar objects under specified conditions

Smart Building: A subset of smart environments.

Smart Building Application: A context-aware application used in an indoor setting and affected by low-level context gathering in that setting.

Smart Environment: Environments which are richly and invisibly interwoven with sensors, actuators, displays, and computational elements.

Situation: The highest level of abstraction of context and the level at which users define a particular context. An application acts upon a situation which is derived from either low or high level context.

Uncertainty (in Context): Uncertainty in context is defined by the precision of the context value and its timeliness (i.e. the variance in the latency between sensing and reporting that value).

Related Publications by the Author

AnzaldiI, G. Corchero, A. Gutu, D., Marinov, M. Dibley, M.J. McGlinn, K. "*Intelligent Building Energy Management through holistic knowledge based approach*" Accepted to the 15th International Conference of the Catalan Association of Artificial Intelligence CCIA 2012, pages 223 – 232, University of Alacant, October 24-26, 2012.

McGlinn, K., R. Brennan, O'Sullivan, D. Lewis, D. (2011). "*The SimCon Generator: An Interactive Context Simulator for Rapid Evaluation of Smart Building Applications using Virtual Reality*." 8th International Workshop on Managing Ubiquitous Communications and Services, pages 50-55, Seattle, USA, IEEE, 2011.

McGlinn, K., O'Neill, E., Gibney, A., O'Sullivan, D. Lewis, D.(2010). "*SimCon: A Tool to Support Rapid Evaluation of Smart Building Application Design using Context Simulation and Virtual Reality*." Journal of Universal Computer Science. V 16(15): pages 1992-2018, 2010.

McGlinn, K., E. Corry, O'Neill, E. Keane, M. Lewis, D. O'Sullivan D. (2010). "*Monitoring Smart Building Performance Using Simulation and Visualisation*". In Proceedings of Ubiquitous Computing for Sustainable Energy (UCSE2010), Ubicomp 2010 Workshop, Copenhagen, Sept 25th, ACM, 2010.

O'Neill, E., McGlinn, K., Lewis, D., Bailey, E., Dobson, S., and McCartney, K., *"Application Development using Modelling and Dynamical Systems Analysis*", In Proceedings of 1st International Workshop on Context-Aware Middleware and Services (CAMS '09), Dublin, Ireland, 16th June 2009, ACM, 2009.

Mara, P., McGlinn, K., Brennan, R., O'Sullivan, D. Keane, M., O'Donnell. 2009. "*Pervasive Knowledge-Based Networking for Maintenance Inspection in Smart Buildings"*. International Conference on Autonomic Computing Proceedings of the 6th international workshop on Managing ubiquitous communications and services, pages 59-66, Barcelona, Spain, ACM, New York, 2009.

McGlinn, K., E. O'Neill, Lewis, D. (2008). *"SimCon: A Tool for Modelling Context Sources for Rapid Evaluation of Pervasive Applications using Virtual Reality"*. Proceedings of the 5th Annual International Conference on Mobile and Ubiquitous Systems: Computing, Networking, and Services (Mobiquitous '08), pages 1-2, ICST, Trinity College Dublin, Ireland, July 21-25, 2008.

Kevin Feeney, D. L., Kris McGlinn, Declan O'Sullivan, Anne Holohan. "*Avoiding 'Big Brother' Anxiety with Progressive Self-Management of Ubiquitous Computing Services*", Proceedings of the 5th Annual International Conference on Mobile and Ubiquitous Systems: Computing, Networking, and Services (Mobiquitous '08), ACM, Trinity College Dublin, Ireland, July 21-25, 2008.

O'Neill, E., McGlinn, K., and Lewis, D., "*A Platform to Evaluate Usability in Adaptive Context-Aware Services through Adaptive Questioning*". In Adjunct Ubicomp '07 Proceedings, USE '07, 1st International Workshop on Ubiquitous Systems Evaluation, Innsbruck, Austria, 16th Sept 2007.

McGlinn, K., O'Neill, E., Lewis, D., "*Modelling of Context and Context-Aware Services for Simulator Based Evaluation*", In Proceedings of 4th International Workshop on Managing Ubiquitous Communications and Services (MUCS '07), Multicon Lecture Notes, Munich, Germany, May 25, 2007.

O'Neill, E., Lewis, D., McGlinn, K., Dobson, S., "*Rapid User-Centred Evaluation for Context-Aware Systems*", XIII International Workshop on Design, Specification and Verification of Interactive Systems (DSV-IS 2006), pages 220 - 233, Dublin, Ireland, July 2006, Springer LNCS, 2006.

1 Introduction

1.1 Motivation

The term “smart environment”, as defined by Cook and Das, denotes an environment which is “richly and invisibly interwoven with sensors, actuators, displays, and computational elements" (Cook and Das 2004). Smart environments are an enabler for context-aware applications, which are applications that can react to changing context. In Dey’s Ph.D. thesis “Towards a better understanding of context and context-awareness” he defined context as “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”(Dey 2000). When the smart environment in question is a building which enables context-aware applications, we call this a “smart building” (SB). Smart building applications (SBAs) are therefore a subset of context-aware applications and are applications which can react to changing context in buildings. SBAs can support building functions such as security (Song, Choi et al. 2008), energy management systems (Agarwal, Balaji et al. 2010), assisted living for elderly (Wood, Stankovic et al. 2008), and museum tour guides (Tsai, Chou et al. 2010).

Designing SBAs and then evaluating them to determine if they have met the needs of users is a non-trivial matter. As SBA design involves systems with many inter-connected components, each of which may have context-aware behaviour, the number of permutations of evaluation parameter can quickly grow and therefore become very complex to handle (Carter and Mankoff 2005). For example, the SBA may be required to function in more than one type of building, each of which will have its own unique architectural design and usage function (e.g. a hospital, or an office). Such buildings will require sensor deployments (e.g. wired or wireless sensor networks) to provide context information to the SBA and these may be costly to acquire, install and maintain (requiring expensive human resources and expertise). Establishing, configuring and maintaining a smart building for the purposes of conducting evaluations of an SBA therefore represents a significant cost (Reilly, Dearman et al. 2005).

Sensor deployments may also exhibit varying levels of uncertainty in the context information they generate (Bettini, Brdiczka et al. 2010). Uncertainty is a measure of lack of knowledge about a system. This can be the result of having incomplete information about the current state of an environment as a result of not having the right type or number of sensors to detect a phenomenon, or it can be the result of measurement uncertainty. Measurement uncertainty in sensors is the result of a number of factors. (Ye, McKeever et al. 2008) identified these as: the technical limitations of the sensor (for example, variation in sample rates, delays and the accuracy of the sensors); environmental noise (for example factors which affect the accuracy of the measurement, such as temperature, humidity or the properties of materials which may cause multi-path issues for wireless); and user behaviour (e.g. the user configuration of the sensors may affect accuracy or a user may neglect to carry a tag required for location detection).

For SBA developers it is important to consider the effects on their application of uncertainty in the sensing of context information, as uncertainty that is introduced at the sensor level will propagate through each subsequent level up until the point at which context is consumed by the application. (Schmidt 2002) has identified location as a major category of context. Uncertainty in location context can have a serious impact on the behaviour of an SBA. For example, if the location context representing the location of a person is not received by an SBA in a timely manner, the person's actual location may have changed before the SBA acts. Similarly, an SBA may act upon location data which places the person somewhere where they are not, due to inaccuracy in the sensing technology. As a result of SBA developers not sufficiently considering the effects of uncertainty in context the user may reject the application on the basis that it does not behave correctly (i.e. it acts upon contexts which are not matched to the context the user has experienced) (Damián-Reyes, Favela et al. 2011).

There are a number of methods for mitigating against uncertainty at the application level. These involve the use of different models to reason about uncertainty, for example, Bayesian networks (Gu, Pung et al. 2004; Truong, Lee et al. 2005; Ye, Coyle et al. 2007), hidden markov models (Liao, Patterson et al. 2007; Sanchez, Tentori et al. 2008), fuzzy logic (Cao, Xing et al. 2005; Cheung 2005; Delir Haghighi, Krishnaswamy et al. 2008), probabilistic logic (Ranganathan, Al-Muhtadi et al. 2004), and Dempster-Shafer theory (Wu, Siegel et al. 2002; Padovitz, Loke et al. 2005; Liao, Bi et al. 2011). These methods rely on programmed or learned knowledge about the behaviour of the sensors in order to assign values to support decision making. Whether the developer is using one of these techniques or developing their own specific heuristics for dealing with context uncertainty in their application, a clear understanding of the context uncertainty being dealt with and its impact on SBA development is required (Bettini, Brdiczka et al. 2010). Another method for mitigating against uncertainty in an application is to improve the intelligibility of the effects of context uncertainty for the users. This can be performed by visualising context uncertainty levels for users (Antifakos, Kern et al. 2005; Rukzio, Hamard et al. 2006; Dearman, Varshavsky et al. 2007; Lemelson, King et al. 2008; Lim and Dey 2011), so that they gain an improved understanding of why an application is behaving in a certain, perhaps unexpected, way, and thereby, it is theorized, trust in applications exhibiting imperfect context aware behaviour is improved (Lim and Dey 2011). Evaluations of this approach have seen various levels of success and some studies have found that while trust may be improved, giving the user additional information on uncertainty can have a negative impact on user performance (Rukzio, Hamard et al. 2006; Lim and Dey 2011).

This thesis aims to improve the SBA developer's understanding of the characteristics of context uncertainty and its impact on different SBAs in different settings, through visualisation in repeatable SBA evaluations. Ideally these evaluations should include users, as these result in the best designs (Carter and Mankoff 2005). Evaluation of early SBAs required field-based observations of prototypes with live sensor deployments, either in the target SB or in a specially constructed test laboratory

(Kidd, Orr et al. 1999; Consolvo, Arnstein et al. 2002; Yamazaki 2005). Such evaluations involved coordination of a number of users moving around the SB to interact with sensor and SBAs while performing a range of activities. This type of user-centric field-based evaluation is valuable in gathering direct user reactions to SBA behaviour over a very wide range of possible user, SB and SBA interaction scenarios. However, it is both time-consuming to organise and, when not conducted in dedicated test laboratories, disruptive to the buildings' on-going use. This makes the ideal of user centric rapid prototyping through iterative design and evaluation cycles expensive and problematic while the complexity of context-aware applications makes one short evaluation in a traditional waterfall design process highly risky (Reilly, Dearman et al. 2005). Ideally, SBA designers should be able to rapidly prototype their designs early and repeatedly evaluate them during their development (Davies, Landay et al. 2005; Weis, Knoll et al. 2007).

Numerous methods have been suggested to reduce the costs of conducting field-based evaluations, ranging from experience evaluation (Reilly, Dearman et al. 2005), paper prototyping (Carter and Mankoff 2005) and Wizard of Oz (WoZ) techniques (Reilly, Dearman et al. 2005), (Li, Hong et al. 2007). These require an investment of time from a third party, which must also have the necessary skills to conduct the test. Furthermore, for mobile applications, monitoring or following participants can be difficult to manage and can limit the realism (Reilly, Dearman et al. 2005).

To address these issues, simulation has been proposed as a method for evaluating smart environments to aid with the evaluation of context-aware applications and a number of context simulators currently exist for generating simulated context to evaluate both indoor and outdoor context-aware applications. To reflect the importance of user-driven design and prototype evaluation of SBA's, context simulators are classified here to distinguish those with no user interaction (Sanmugalingam and Coulouris 2002; Battestini, Flanagan et al. 2005; Broens and Halteren 2006; Nishikawa, Yamamoto et al. 2006; Jouve, Bruneau et al. 2009) and those which support the user's visualisation and interaction with the building, the latter of which in turn generates context information, such as user location (Barton and Vijayaraghavan 2002; Shirehjini and Klar 2005; Prendinger, Brandherm et al. 2009). Virtual Reality (VR) is often used to support the interactive, user-driven generation of context information. A popular candidate for supporting the interactive VR element, is the use of first-person view VR games engine like Quake (Bylund and Espinoza 2002) or Half Life 2 (O'Neill, Klepal et al. 2005), or third person 3D virtual worlds like Second Life (Prendinger, Brandherm et al. 2009). These are preferred over more immersive VR alternatives like CAVE (DeFanti, Acevedo et al. 2011), which may present users more realistic visual environments, but offer few advantages in terms of improved user interaction with sensors and the SBA while being an expensive resource typically beyond the reach of most SBA developers.

To enable SBA evaluation using an interactive context simulator a number of tasks must first be undertaken. These tasks range from identifying an appropriate set of test scenarios, placing and configuring of sensors, the integration of simulated context with the SBA prototype, the capture and use in SBA evaluation of user behaviour and the configuration, and visualisation and understanding of the impact of context uncertainty on the behaviour of an SBA design. The integration of these tasks and the ease with which they can be conducted and repeated by SBA evaluators is therefore key to the success of any context simulator. However, while usability has been identified as a key objective of context simulators to support developers in these complex tasks (Broens and Halteren 2006; Reynolds, Cahill et al. 2006), few focus on ease of use in their design and no existing context simulators have yet been evaluated to determine the usability of their models. A context simulator should therefore, using a standard definition of usability (ISO 1998), be evaluated to determine if it is effective, efficient and satisfying for SBA developers to use when conducting evaluation of SBA prototypes.

As the modelling of the VR representation of the smart building can present a considerable amount of effort (if the visual representation of the building must be modelled from scratch), some approaches have looked at methods for reducing the time to model the VR environment through re-use and sharing of visual models between application design and evaluation projects, facilitated by the use of standards like the Virtual Reality Modelling Language (VRML) (Nishikawa, Yamamoto et al. 2006). In the building industry Computer Aided Design (CAD) and simulation tools are common in evaluating architectural design, construction and energy and security management decisions (Papamichael, Chauvet et al. 1999). The need to share data across the Building Life Cycle (BLC) has led to the development of the Building Information Model (BIM) (Himanen 2003; O'Sullivan and Keane 2005; Eastman, Teicholz et al. 2011). The BLC is the process of conception, design, construction, operation, modernisation and finally recycling (or demolition) of a building (Himanen 2003). A BIM describes an integrated data model for storing all the information relevant to the building life cycle. The advantage of using BIM is that not only can the visual representation of the building be modelled (to support interaction), but also properties of the building can be modelled, which can then be used as parameters for context simulation, for example, to simulate context sensing uncertainty caused by building material.

This thesis therefore addresses the design, implementation and evaluation of a context simulation model that uses VR-based interactive context simulation, leverages existing building information models and support simulation and intelligibility of context uncertainty in the evaluation of SBA prototypes by SBA developers. The approach must assess the usability of tools that implement this model with the assumption that SBA developers will have to repeatedly evaluate different SBA prototypes with different user behaviours, in different buildings equipped with different sensor systems deployed in a wide variety of configurations.

The next section formally outlines the research question addressed by this thesis, the research objectives set to address this question and the approached taken to answering the question.

1.2 Research Hypothesis

This thesis sets out to evaluate the research question: "*What level of usability is achievable in a VR-based context simulator that supports creation and configuration of simulated context sources, their associated uncertainty, and which supports the evaluation of SBAs in a manner that is flexible, extensible, and leverages other models in the building life cycle*."

Usability here refers to how well SBA developers can effectively and efficiently evaluate aspects of a SBA's behaviour over a variety of context sources and associated uncertainty characteristics and their satisfaction with the experience. Flexibility refers to the context simulation model's capacity to model a range of heterogeneous contexts sources and uncertainty characteristics in the resulting simulated context. Extensibility refers to the context simulation model's capacity to be extended to handle new context types. The leveraging of other models in the building lifecycle is assessed by the level of interoperability achieved between the context simulation model and the building information model standards.

1.3 Research Objectives

To address the presented research question, the following objectives were defined:

1.3.1 Objective 1:

Conduct a state of the art review of current approaches to supporting SBA developers to evaluate SBAs through the use of simulation, in particular when faced with varying levels of uncertainty in context information. From this review, identify the key requirements for such a platform through analysis of the strengths and weaknesses of current approaches.

1.3.2 Objective 2:

Design a context simulation model which supports the creation and configuration of simulated context sources that generate user driven simulated context and which supports SBA developers evaluate SBA's when faced with varying levels of uncertainty in context. The model should be easily understood and manipulated for this task by SBA developers. The model must also be interoperable with existing standards to support integration with other tools involved in the building life cycle and thereby to leverage the models produced with these tools.

1.3.3 Objective 3:

Implement and evaluate the resulting model to determine if it has met the key requirement of being usable by SBA developers during configuration of user driven simulated context sources and during the evaluation of SBA's when faced with varying levels of uncertainty. The usability evaluation is further broken down into four task-based sub-objectives. These are that the context simulation model supports:

- SO1 Configuration of a variety of heterogeneous user driven simulated context sources.
- SO2 Evaluation of SBA using the simulation and visualisation of heterogeneous context sources.
- SO3 Configuration of uncertainty in user driven simulated context sources.
- SO4 Evaluation of SBA using simulation and visualisation of context sources with various uncertainty configurations.

1.4 Technical Approach

To meet objective 1 a background survey was conducted which examined aspects relevant to the research question. These included context and context modelling and issues related to uncertainty introduced into context generated by sensors. This type of uncertainty was then explored in greater detail. Different methods for addressing uncertainty were examined and methods which support SBA developers evaluating their applications early in development using simulation, so that they can develop applications which can handle varying levels of uncertainty, were highlighted for further exploration. A requirement was identified from this, that to successfully simulate smart buildings, modelling of both the *buildings* and *sensors* is required to a level sufficient for early rapid evaluation of the SBA using VR. A standard called the Industry Foundation Classes (IFC) and Sensor Modelling Language (SensorML (Botts 2002)) were identified as existing standards to address the modelling of the buildings and the sensors. Also identified was the need for any solution to be usable by SBA developers, and so some background of usability was examined. The background survey was followed by a state of the art review of smart building applications and the requirements of smart building application developers. From this a set of criteria were identified by which to analyse context simulators. This analysis identified key features, including strengths and weaknesses, of current context simulators. From this state of the art analysis the key requirements for the proposed context simulation model were identified.

To meet objective two and address the requirements identified in objective one, a context simulation model called the SimCon model was designed which supports the creation and configuration of uncertainty in simulated context sources, by SBA developers, and that generates simulated context from user interactions with a virtual reality building environment. The model also supports the evaluation of SBA's, by SBA developers, when faced with varying levels of uncertainty in context. The model shares concepts with the existing standard IFC, supporting the use of IFC building models were available. This model was then implemented to enable its evaluation. To determine the usability of the SimCon model when creating, placing and configuring simulated context sources with uncertainty, a tool called "SimConfig" has been developed.

To support the evaluation of SBAs when faced with varying levels of uncertainty in simulated context which is affected by the movement of human controlled avatars in the VR building; the "SimConGen" interactive context simulator has been implemented. The SimCon model also aims to support the analysis of simulated context to improve intelligibility of the effect of uncertainty, and for this purpose a visualisation tool called the "SimConViz" tool has been developed. Through the evaluation of each of these implementations, this thesis will show that the SimCon model enables placing and configuring simulated context sources within a VR building which generate context from user (and also bot) driven behaviour to support SBA developers evaluate an SBA when faced with varying levels of uncertainty in context.

1.5 Research Contributions

This section provides the major and minor contributions of this thesis.

The major contribution of this thesis is the SimCon model which addresses the complexity of developing and evaluating SBAs when faced with varying levels of uncertainty in context generated by sensors. The SimCon model addressed this complexity through the development of models which abstract from the underlying complexity of the causes of uncertainty, so that SBA developers may configure the properties of the simulated context sources and then evaluate their SBA in a manner which is satisfying and most importantly effective. Only InSitu (section 3.4.1.9) has been examined to determine its level of usability, but its focus is on specifying situations to determine unwanted behaviour and does not address the range of SBA developer tasks assembled here from the literature. Conducting usability evaluation of the subsets of tasks that the SimCon model addresses is an important contribution since improving the level of usability of context simulators, in particularly when dealing with practical problems around context uncertainty, is key to integrating SBA development as a mature part of the future smart building lifecycle. In particular, the SimCon model has contributed to this in two key respects.

The first is in the task of creating, placing and configuring simulated context source with varying levels of uncertainty. The causes of uncertainty in context-aware system behaviour are often difficult to ascertain, even for experts in sensing systems. For SBA developers who may not be experts in these systems, and who must develop SBAs which are robust in the face of varying levels of uncertainty, tool support for the configuration of uncertainty is a key requirement. The SimCon model allows the configuration of the uncertainty associated with a context source model in a way that enables the precision characteristic of that uncertainty to be replicated by the output of the simulated sensor during an interactive evaluation. It also supports configurable zones of differing imprecision for location context sources and a toolset for creating, configuring and placing of these zones, so that they can be related to the building geometry and the position of avatars when generating user driven location context. While a number of non-interactive context simulators have discussed modelling uncertainty, for example, the Generic Simulation Tool (section 3.4.1.6), and for configuring uncertainty, for example, SimuContext (section 3.4.1.4) neither of these have been evaluated in use by SBA developers to determine their level of usability. The only interactive context simulator to discuss uncertainty is the Simulation Framework in Second Life (section 3.4.2.6), but does not provide any models and only discusses the requirement for usable tools to support the SBA developer with configuring the simulator. It also presents no discussion of additional tools to support the evaluation process.

The second part of the contribution is the task of evaluating the SBA. Through the combination of the SimConGen simulated context generator, and the SimConViz context visualisation tool, the SimCon model has been shown to support SBA developers during the process of SBA evaluation by generating simulated context with varying, configured levels of precision and visualising the resulting uncertainty in context. A key finding of the research, through the summative evaluation, is that SBA developers ranging from novice to expert who were presented with the visualisation of varying levels of precision in location context, when placing simulated context sources to meet the requirements of an SBA under evaluation, exhibit a threefold improvement of placement over users who did not have this visualisation. This provides evidence on the importance of visual feedback in SBA development, whereas evaluation in providing such feedback during SBA usage indicates that in that role it is more of a distraction (Rukzio, Hamard et al. 2006; Lim and Dey 2011).

Finally, the SimCon model has demonstrated greater flexibility in the number of approaches it supports to simulating context, enabling SBA developers choose between the best suited context simulation approaches to meet the needs of their particular evaluation. To date, the most flexible simulator is DiaSim (section 3.4.1.8), which supports a number of the reviewed approaches to context simulation. It does not address interactive context simulation though and no existing interactive context simulator presents methods for simulating context other than location context. Also, only the Context Simulation Tool discusses at a high level the use of external models to support simulation, and SimCon is unique in its support for this as shown in its implementation.

The minor contribution of this thesis is the integration of the context simulation model into a standard model for describing buildings called Industry Foundation Classes. This is important because the user driven simulation of context in buildings is a complex task which can require visual representations of the building, properties of building elements and models of sensors which support context simulation. The main advantage of this is that the potential exists for importing existing building models directly into SimCon, reducing the time to develop interactive VR environments and improving consistency with other modelling activities in the BLC. IFC also supports modelling aspects of the building which may affect the value of simulated contexts. This type of parametric simulation can result in context simulation which is a closer match to how the physical system being simulated behaves in a particular environment. Therefore, integrating IFC into the context simulation approach has the potential of improving the realism of the simulations, i.e. how true the simulation is to how the physical systems behave.

1.6 Thesis Overview

Chapter 2 presents background literature relevant to this thesis. It begins by giving an introduction and definition to some of the key concepts, relating to context, context modelling and uncertainty in context. The current practices in building and sensor modelling are also presented. Finally, the topic of usability is discussed due to its relevance to the research question.

Chapter 3 presents a state of the art review of context simulators. It begins with a review of smart building applications and location sensing systems, identifying a typical set of activities related to developing SBAs which require location context are examined. Context simulation is suggested as a method to support the evaluation of SBAs. Next, a review of context simulators is given with respect to a set of identified criteria which must be met to support SBA developers. This review results in a table which captures the different features of the context simulators along with explanations of these features. Lastly, the chapter gives a summary of the strengths and weaknesses of existing context simulators, followed by a summary and conclusion to the chapter.

Chapter 4 presents the design of the context simulation model. It begins by identifying the requirements, resulting from the state of the art review. A high level view of where the context simulator fits into the process of SBA design and development is presented next, focussing on the specific tasks required for the developers of SBAs when evaluating their application when faced with varying levels of uncertainty in context using a context simulator. The result of this is an overview of the context simulation framework emphasising the role that the SimCon model plays in that framework. This leads to the presentation of the SimCon model and a detailed description of the different components that make up that model, as well as conceptual similarities with existing standards. Finally a conclusion and summary are given of the proposed design.

Chapter 5 presents the implementation of the SimCon model. It begins by introducing the different implemented components which make up the SimCon system beginning with a description of the prerequisite modelling which must be done to enable SimCon. Next the SimConfig tool for creating, placing and configuring SimCon sources in a simulated building is presented. This leads to the description of the SimConGen simulated context generator which includes an explanation of how the different context simulation approaches are implemented. Finally, the SimConViz tool is described, which enables visualisation of simulated context to further support evaluation of the SBA.

Chapter 6 presents the evaluation of SimCon to meet objective 3 of this thesis. The usability evaluation is broken into three parts, with the first and second having two and three formative sub evaluations respectively and the third concluding in one final summative evaluation of the SimCon model. The three evaluations end with a summary and a conclusion, culminating in a final summary and conclusion for all evaluations at the end of the chapter and a comparison of the SimCon model with existing context simulators.

Chapter 7 presents the conclusions of this thesis. It examines the extent to which this thesis has met the objectives set down in Chapter 1. It details the major and minor contributions of the thesis and concludes with an exploration of future work.

2 Background

2.1 Introduction

This chapter will describe the background to this thesis. It begins by giving an introduction and definition to some of the key concepts used in the thesis and in particular those related to context simulation. Next it explores the need for considering factors that vary the levels of uncertainty in context sensing when evaluating smart building applications (SBAs). The causes of uncertainty in context generated by context sources, like sensors, are then explored. From this the need for both building and sensor models to support simulation of context to evaluate SBAs will be argued. The current practice in building and sensor modelling are then presented.

Finally, the complexity of evaluating SBAs with varying levels of uncertainty in sensed context means the usability of the model is a key success factor. However, the analysis of the state of the art in the next chapter reveals that the assessment of the usability of tools in this area is poorly researched, so in advance of addressing such evaluation, the topic of usability is discussed.

2.2 Context, Context-Aware Computing and Context-Aware Applications

As this thesis sets out to address the simulation of context in buildings for the purpose of evaluating SBAs, this section will explore key concepts related to SBA development.

The term *context-aware computing* was first defined by Schilit and Theimer as ”the ability of a mobile user's applications to discover and react to changes in the environment they are situated in*”*. They also described *context* in the form of location, identities of persons and objects, and changes to those objects (Schilit and Theimer 1994). Since then numerous definitions have been given which apply to specific domains or stages in a specific domains. For example, Browne defined it as elements of the user’s environment which the computer knows about (Brown 1996) and in mobile computing Chen and Kotz defined context as “the set of environmental states and settings that either determines an application’s behaviour or in which an application event occurs and is interesting to the user" (Chen and Kotz 2000). Within the scientific community, in differing fields, the exact definition of context is still open for discussion due to its subjective nature (Bazire and Brézillon 2005). In 2000 Dey presented what is perhaps the most widely referenced definition of context, and the definition this thesis will use (Dey 2000):

"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."

Within the ubiquitous computing domain applications which use context are called "context-aware applications". Ubiquitous computing is a term which is widely attributed to Mark Weiser who described it as technologies that "weave themselves into the fabric of everyday life until they are indistinguishable from it" (Weiser 1999). Other terms often used interchangeably with ubiquitous computing are pervasive computing and ambient intelligence, and each share many similarities, although a study by Ronzani has found that, in the literature, pervasive computing tends to be used in relation to technologies deployed at home and for leisure (Ronzani 2009). Ronzani has found that ambient intelligence, on the other hand, is a more recent term and is more difficult to associate with any particular technology or domain. Cook and Das refer to environments which are "richly and invisibly interwoven with sensors, actuators, displays, and computational elements" as "smart environments" (Cook and Das 2004). This thesis shall use the term smart environment to refer to environments which enable context-aware applications, as defined by Cook and Das.

Smart buildings are defined as a subset of smart environments and the term smart home or smart building is therefore a term used to describe buildings which support context-aware applications (Fahy and Clarke 2004; Zhang, Gu et al. 2005; Lertlakkhanakul, Choi et al. 2008). Smart building applications (SBAs) are also then a subset of context-aware applications and in this thesis they are described in these terms. Although, when reviewing context-simulators, the term context-aware application may often be used interchangeably with SBA as many existing context simulators address both indoor and outdoor context-aware applications. Also, the approach taken in this thesis to context simulation, while applied to buildings, will be based on a flexible model which may be applied at a future date to evaluating outdoor context-aware applications. Before smart buildings and smart building modelling are discussed in more detail, some background in the area of context modelling and also issues of uncertainty will be discussed.

2.3 Context Modelling

At the sensor level, context is often referred to as "low level context" and is the result of the sensor measuring a phenomena in the environment (Bettini, Brdiczka et al. 2010). High level context and situations are seen as higher abstractions of low level context and which are the result of "context processing" (Damián-Reyes, Favela et al. 2011). A situation could be, for example, "in_meeting_now" (Loke 2006). Figure 2-1 gives and overview of these different stages of context and where they lie in relation to the SBA.

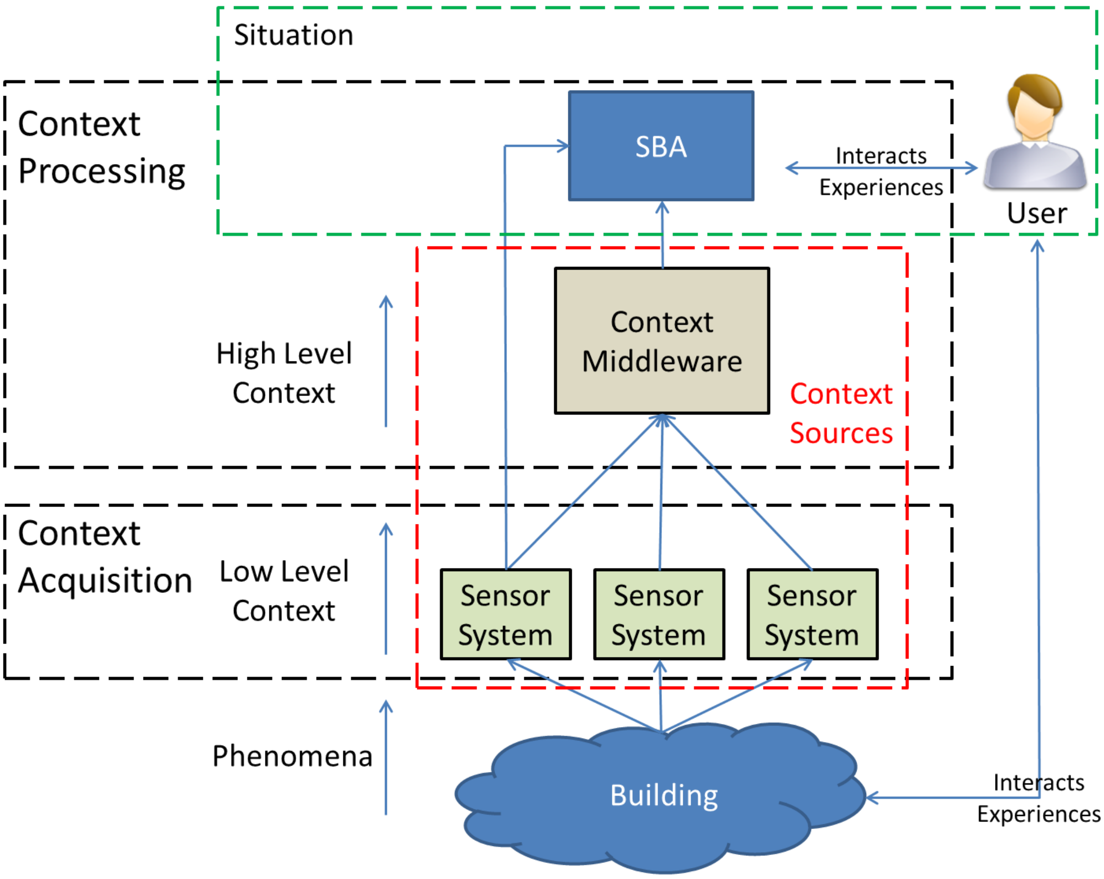


Figure 2-1 Context Acquisition and Processing, resulting in low and high level context and ending in a situation

The SBA must process low and/or high level contexts to determine if such a situation applies. The user experiences the building as they interact with it and other occupants and they may also be affected by the SBA in some way. For example, when an SBA determines a situation to have occurred and acts upon it, either through actuating a device in the building (for example, turning on the air conditioning in the meeting room) or it adjusts its own behaviour in some way (for example, a mobile phone turns itself to silent). The objective for the SBA developer is to ensure that the SBA's interpretation of the current situation matches with that of the users, and the SBA therefore behaves in an appropriate manner. This requires context acquisition and context processing to take place.

Context processing is done either by middleware or by the application and can result in the aggregation and/or filtering of low level context, and/or the creation of new more semantically rich high level context through context fusion (Wu, Siegel et al. 2002; Padovitz, Loke et al. 2005; Anagnostopoulos, Sekkas et al. 2007; Roy, Gu et al. 2010). Ultimately it is the application which acts upon context and all context processing relies on low level context which is gathered through a process called "context acquisition" (Damián-Reyes, Favela et al. 2011). Context acquisition is handled by context sources (Baldauf, Dustdar et al. 2007). This thesis makes use of Broens and Halterens definition of a context source as anything which generates context relevant to a context-aware application (Broens and Halteren 2006). A context source can therefore be a sensor which measures some phenomena (temperature, radio frequency) or it could also be a device that reports on the state of hardware (a device is on/off, a window is open/closed). Finally, it should be noted, that a context source may also include context processing which processes low level context to infer a higher level context. Context sources may therefore be an aspect of context processing.

For the purpose of simulation, a context source is anything which provides context to a system for evaluation. That system could be a standalone SBA, context processing middleware, or a combination of the two. Due to the prevalence of sensors for generating context in the literature

(Scholtz and Consolvo 2004), the focus of this thesis will be on simulating low level context sources which form part of context acquisition by measuring phenomena in the environment. This context will be used to evaluate SBAs which process low level context and use it to react to changing situations. The method employed to simulating low level context sources will aim to be flexible enough to model context sources which generate higher level context, though these will not be evaluated in this thesis.

Strang et al. have identified context models as a "key accessor to the context in any context-aware system"(Strang and Linnhoff-Popien 2004). For applications, context models are necessary in order to reason over changing situations and over the years a large number have been developed to address the needs of different application domains (Bettini, Brdiczka et al. 2010). Strang, Baldauf and Bettini (Strang and Linnhoff-Popien 2004; Baldauf, Dustdar et al. 2007; Bettini, Brdiczka et al. 2010) have identified the following types: ranging from: key-value models (Schilit, Adams et al. 1994), mark-up scheme models (Sheng and Benatallah 2005), graphical models (Henricksen, Indulska et al. 2003) (Sheng and Benatallah 2005), object oriented based models (Northover and Wilson 2004), ontology based models (Chen, Finin et al. 2005; Zhang, Gu et al. 2005), spatial context models (i.e. models of context which take space into consideration) (Schilit, Adams et al. 1994; Addlesee, Jones et al. 1997) and hybrid models (Henricksen, Livingstone et al. 2004).

Few context models have addressed the modelling of context sources, and even within the context simulation community, context source models tend to be developed in what seems to be an ad hoc manner with few examples provided. This issue is explored in greater detail in Chapter 3 where context simulators are reviewed. In the domain of context modelling, Korpipää and Mäntyjärvi have developed a model which focuses on abstracting raw sensor data into semantic context in order to develop mobile applications which are more usable (Korpipää and Mäntyjärvi 2003). They identify some key attributes of low level context sources which can inform the development of a model of context sources for the purposes of simulation. These are the definition of a "context" and "context type", "value", "confidence", "timestamp" and "source". *Context type* indicates a category of context, for example, location. *Context* is the symbolic value, for example a coordinate, and *value* indicates its numerical value. In their model, Korpipää and Mäntyjärvi model uncertainty from the application perspective, giving the context a *confidence* property which they describe as either a probability or fuzzy membership which can be reasoned over. As highlighted in the introduction chapter, uncertainty can have a significant impact on the behaviour of SBAs which must react to changing context. The next section will therefore explore uncertainty in context and different concepts, like confidence, in greater detail.

2.4 Uncertainty in Context

Uncertainty can be introduced into context both during the context acquisition and context processing stages identified in the previous section (Damián-Reyes, Favela et al. 2011). Processing uncertainty requires that context models capture aspects of uncertainty. In the context modelling community, concepts like up-to dateness (Buchholz, Küpper et al. 2003), frequency and timeliness (Lee, Ha et al. 2006), freshness (Lei, Sow et al. 2002; Ranganathan, Al-Muhtadi et al. 2004) and decay (Clear, Dobson et al. 2007), are all used in relation to the rate which context is generated and any additional delays. This information is important as the time between the actual event and the processing of the context will have a direct correlation to its accuracy. For example, a coordinate location of a mobile tag carried by a person moving at a meter per second and which takes a tenth of a second to be processed will be more accurate that a coordinate which takes a full second to process, as the person will have moved a greater distance in that time.

Concepts like accuracy (Lee, Ha et al. 2006), precision (Buchholz, Küpper et al. 2003; Clear, Dobson et al. 2007), and resolution (Ranganathan, Al-Muhtadi et al. 2004) are often related to accuracy, in that they attempt to measure the closeness of agreement between a context value and the true context value it is attempting to represent, and precision by quantifying the replicability of that closeness, although at times the terms accuracy and precision may be used interchangeably in the literature. In this thesis when discussing accuracy and precision, the definition given by the Joint Committee for Guides in Metrology in the international vocabulary of metrology is used, and they define *accuracy* as the closeness of agreement between a measured quantity value and a true quantity value, and *precision* as the closeness of agreement between indications or measured quantity values obtained by replicate measurements on the same or similar objects under specified conditions (Zlatanova and Verbree 2005). Other values related to uncertainty are confidence (Lei, Sow et al. 2002; Ranganathan, Al-Muhtadi et al. 2004; Clear, Dobson et al. 2007) and trustworthiness (Buchholz, Küpper et al. 2003). Both concepts attempt to assign a probability to the source of the context information that captures the likelihood that it is providing "correct" information, although once again, definitions may vary according to author and domain.

Numerous solutions have been suggested to deal with uncertainty at the processing stage, which involve modelling and reasoning about using low level context, for example, Bayesian networks (Gu, Pung et al. 2004; Truong, Lee et al. 2005; Ye, Coyle et al. 2007), hidden markov models (Liao, Patterson et al. 2007; Sanchez, Tentori et al. 2008), fuzzy logic (Cao, Xing et al. 2005; Cheung 2005; Delir Haghighi, Krishnaswamy et al. 2008), probabilistic logic (Ranganathan, Al-Muhtadi et al. 2004), and Dempster-Shafer theory (Wu, Siegel et al. 2002; Padovitz, Loke et al. 2005; Liao, Bi et al. 2011). These methods often rely on learned knowledge by the model developers about the behaviour of the sensors themselves in order to assign values to support decision making. They also rely on expert knowledge of the modelling process itself. This presents a considerable effort in terms of time and resources for the developers (Damián-Reyes, Favela et al. 2011).

Other methods include improving intelligibility for the users, by for example displaying uncertainty levels (Antifakos, Kern et al. 2005; Rukzio, Hamard et al. 2006; Dearman, Varshavsky et al. 2007; Lemelson, King et al. 2008; Lim and Dey 2011), so that they can understand why an application is behaving in a certain way, and as a result, it is theorized, trust is improved (Lim and Dey 2011). Dearman et al. found that visualisation of location uncertainty improved user performance with respect to a location based service (Dearman, Varshavsky et al. 2007). Conversely, Antifakos et al. found that displaying system confidence improved user trust in the application but also impacted negatively on their performance when uncertainty was high (Antifakos, Kern et al. 2005). Rukzio et al. also found that performance was affected negatively by displaying uncertainty (Rukzio, Hamard et al. 2006). Similarly to Ruzkio and Antifakos, Lim et al. found that displaying uncertainty when the application was behaving correctly resulted in a negative impression for the user, although, the user was more likely to forgive the application for behaving inappropriately when uncertainty was displayed. They suggest that by giving the user an active role, improving intelligibility may be useful when interactively debugging an application.

Therefore, this thesis does not focus on solutions which aim to develop models to reason about uncertainty or methods for improving intelligibility for end users, but rather on providing tools for SBA developers to test against uncertainty at the level of context acquisition, as it is uncertainty that is introduced at this level which propagates through each subsequent level up until the point at which context is consumed by the application (Ye, McKeever et al. 2008). By providing SBA developers a method to test and evaluate their applications early in development, they can develop more robust applications which can handle varying levels of uncertainty introduced during context acquisition. Chapter 3 will look at the use of simulation to meet this requirement. First, some background in uncertainty modelling is explored which can then be used to assess the approaches taken by existing context simulators and inform the approach taken in this thesis.

2.5 Modelling Uncertainty

Uncertainty in context sources like sensors is often described in terms of “error” (Truong, Lee et al. 2005). Sensor errors, such as noise from external sources, hardware noise, inaccuracies and imprecision, and various environment effects may seriously impact an application which is reliant on that measured data (Elnahrawy and Nath 2003). Webster divides sensor measurement errors into two types: systematic error (bias) and random error (noise) (Webster 1999). Systematic error can result from a range of factors from the imperfect calibration of measurement instruments, aging or damage to components, lack of resolution in sensors technology, changes in the environment which interfere with the measurement process or when the sensor influences the phenomena being observed, defined as invasiveness (for example using a large room temperature thermometer to measure a small cold object). The effect which error has on the output of a sensor can result in an offset to the measured value which is stable, or which is dependent on a changing variable in the system (like temperature, or battery life of the sensor). If the sensor is mobile, these types of error may be difficult to identify, but where the cause is known, the offset can be modelled as a single value, or a curve which maps the offset to some parameter (e.g. temperature). For the purpose of simulation, this approach can be applied to simulate many types of error which result in offsets which are predictable given sufficient understanding of the properties of the system. Chalmers gives a more complete list of these types of errors (Chapter 9) (Chalmers 2011).

One type of systematic error which requires a different approach to simulation is a fault. A fault which arises in the sensor may result in either an intermittent or temporary end of the output signal, or random behaviour (Chalmers 2011). The cause of such a fault can be due to the manufacturing or as a result of the environment and not related to the measured phenomena (e.g. impact, moisture, temperature or vibration). A fault may also be simply the result of loss of power. These same faults may also occur in the communication network. Saturation can also cause a fault when the sensor stimulus approaches or goes beyond the sensor design. The reliability (i.e. chance of a fault occurring which result in complete cessation of output) can be modelled as a Poisson distribution (Staroswiecki 2003).

Random error, on the other hand, is always present in a measurement and is caused by inherently unpredictable fluctuations in the readings of a sensor. A random error shows up as a different result for (supposedly) the same repeated measurement. The need to develop robust SBAs though, requires that they be evaluated to determine how they handle uncertainty (Padovitz, Loke et al.

2004). Therefore methods are needed for introducing pseudo-uncertainty in simulated context. While uncertainty is a measure of lack of knowledge about the system, there are known causes of uncertainty. To give an example, in real time location system like Ubisense (Steggles and Gschwind 2005) uncertainty can be the result of other electronic devices which may interfere with wireless signals, random hardware noise, interference from various building materials, temperature, humidity, background radiation, line of sight issues, its position relative to other objects and occupancy which collectively contribute to reflection, attenuation and multi-path issues (Zyren and Petrick 1998; Schmidt 2003). A method is needed therefore by which the effect of this uncertainty on the simulated context from a context source like Ubisense is modelled for simulation.

An uncertainty model for random error can be defined using a probability distribution function (PDF) which takes as its parameters knowledge about the simulated environment, i.e. given a certain environment a PDF is used to model the probability that a certain output will be generated at each measurement interval. This measurement interval could be, for example, that the rate of the sensor measurement is fifteen minutes. Where no a prior knowledge of the environment exists, the probability that a particular output is generated is assumed to follow a Gaussian (or Normal) distribution (Chalmers 2011), (Elnahrawy and Nath 2003). In the case of RSS simulation, radio wave propagation may follow a log-normal, Rayleigh, or a rice distribution (Barclay 2003). These are used where there are multiple independent random variables, to model the effects of multipath or scatter, or to model a steady, nonfading, component together with a random variable component with a Rayleigh distribution (respectively) (Barclay 2003). Sebastian et al. make the assertion that depending on the requirements, it may be safe to assume a Gaussian distribution for RSS simulation for RFID-based indoor location sensing algorithm (Sebastian 2005). Plagemann et al. have also examined the use of Gaussian distributions although they have not considered applying these models to simulation (Seco, Plagemann et al. 2010). Elnahrawy et al. calculate the variance of the

Gaussian distribution based on the specification of each sensors (precision) and on testing calibrated sensors under normal conditions (Elnahrawy and Nath 2003).

Calculating the variance of the Gaussian distribution for a particular sensor can therefore be performed by either the manufacturer or by the user after installation. Environmental factors can then be examined and used as parameters to derive an uncertainty model for specific contexts. It is this a priori knowledge which results in higher fidelity simulation when dealing with specific contexts. A trade-off is therefore required between the effort to create highly accurate simulations of uncertainty and the requirements for an SBA developer, who may wish to conduct rapid evaluation. Chapter 3 will show that currently, there is a lack of models to support simulation of context with uncertainty to support rapid prototyping. Another important aspect of uncertainty which has been discussed in this section is the need for models of both the context sources and the building to accurately model uncertainty, as the placement of the context source can result in different levels of uncertainty due to the properties of the surrounding environment. The next section shall look at some background on modelling for both buildings and also sensors (i.e. low level context sources).

2.6 Smart Building Modelling

A common approach since the 80's to reduce the costs of developing new designs of buildings and building elements has been the use of Computer Aided Design (CAD) tools, (Björk and Laakso 2010). These range from tools to conduct design analysis, to simulation models to capture building behaviour, to automatic interactive design (Eastman 1999). Traditional CAD models of buildings are generally composed of sets of lines, which can be interpreted as elements such as columns, doors and windows in 2D and more recently 3D. Unfortunately, many applications do not share the same models for representing a building, or do not share the same relationships among objects and rules of how objects are composed (Eastman, Teicholz et al. 2011). This results in aspects of the building having to be modelled multiple times by different domain experts using their respective tools and as a result, data can be lost or become unusable (due to lack of sufficient expertise) in later stages of the building life cycle (BLC) (Hassanien Serror, Inoue et al. 2008). The BLC defines the entire life of a building from inception and design, to construction, through operation and maintenance and on to eventual demolition/recycling (Clements-Croome 2004).

The concept of Building Information Modelling (BIM) has been developed as a direct response to these types of issues. A Building Information Model (also called BIM) describes an integrated data model for storing all the information relevant to the BLC. This can include a 3D model of an architectural design, electrical installations, fire protection, occupancy, energy consumption, costs, CO2 emissions, etc. A BIM provides more than just a consistent representation of objects; it also defines object parameters and relations to other objects. Collectively this data can be used for visualisation and simulation of the building throughout the BLC (Popov, Migilinskas et al. 2008). With the required amount of detail, the BIM can support simulation of the building before construction or implementation to evaluate new designs before changes become prohibitively expensive (Popov, Migilinskas et al. 2008). BIM therefore has the potential to be applied to the modelling of smart buildings for the purpose of evaluating SBAs, supporting not only the modelling of the visual representation of the building, but also data which can be used as parameters for simulation, like the material of walls.

Within the architecture, engineering and construction (AEC) community the Industry Foundation Class (IFC) is an open, freely available, non-proprietary data model which can be used to exchange and share BIM data between applications without the necessity to support numerous data models (ISO 2005). It supports CAD tools to automatically acquire building geometry and other building data required, which can be used for simulation (Bazjanac and Crawley 1997). IFC has seen major government client in Norway, US and Finland, as well as a growing commitment in China (Howard and Björk 2008). It is also the only BIM currently an ISO PAS standard. The International Organisation for Standardisation (ISO 2012) is the world's largest developers and publisher of international standards and IFC is therefore a primary candidate for BIM.

Unfortunately, the latest version of IFC (IFC2x4) only supports rudimentary descriptions of sensors, with properties like measured value, measurement area (for temperature sensors) (ISO 2005). They do not currently support the modelling of properties associated with uncertainty, like rate, precision or response curves. Response curves can be used to model relative measurement error versus the input value itself (Botts 2002). IFC therefore falls short of meeting the requirements for modelling context simulation with uncertainty. In the next section sensor models shall be explored in greater detail to determine if any of them currently support the modelling requirements for simulating context with uncertainty, with a view to integrating these concepts into IFC.

2.6.1 Sensor Modelling

The exact number of sensor types in existence is hard to determine. NASA's Semantic Web for Earth and Environmental Terminology (SWEET) list 415 types of sensor in their sensor ontology called "sweet" (NASA). The number of phenomena sensors can sense is also large and there is no common standard to the design and implementation of sensor systems. As a result the state of the art in sensor systems is of heterogeneous networks of different sensor types, ranging from weather stations to satellites (Botts 2002). This heterogeneity in sensor systems and also the heterogeneity in the communication interfaces has contributed to the challenge of creating interoperable sensors for diverse networks and applications (Hu, Robinson et al. 2007). These same issues must be addressed in SBA design where sensors are required to provide appropriate context. Basing context simulation on well-defined and widely accepted standards for sensor description and communication is a preferable approach for achieving interoperability.

In the early nineties the National Institute of Standards and Technology (NIST) set out to address the heterogeneity of transducers and communication interfaces with the IEEE 1451 standards family. IEEE 1451 is a collection of "smart" transducer interface standards that define a set of open, common and network independent communication interfaces for connecting transducers to microprocessors, instrumentation systems and control/field networks (Hu, Robinson et al. 2007). The transducer electronic data sheet (TEDS) can reside as an embedded memory within the transducer itself, or a virtual TEDS can exist as a data file accessible by the instrument or control system. This describes the transducer identification, calibration, correction data, measurement range, manufacturer related information, and so on. While TEDS describes basic sensor functionality (hardware, sensor calibration, sensing phenomenon, and quality of sensor readings) it is not able to describe higher level processing of sensor data (e.g., fusion, interpretation) which may be required when simulating higher level context information. Therefore it is not ideal for a context simulation model.

More recently, the Sensor Modelling Language has been developed (SensorML) (Botts 2002). This provides standard models and an Extensible Markup Language (XML) encoding for describing the process of measurement by sensors and instructions for deriving higher-level information (like context) from observations. SensorML can be used to create specific sensor profiles, which facilitate the processing, geo-location and integration of observed data from a myriad of sensors. SensorML also supports an extensible collection of properties for describing the sensor's response characteristics and quality of measurement. Response characteristics determine how a sensor will react to a phenomenon. These include specification for sensitivity (e.g., threshold, dynamic range, capacity, band width, etc.), accuracy and precision, and behaviour under certain environmental conditions (Botts 2002). SensorML can potentially be applied to a large number of sensor types whether it is measuring location, temperature, humidity or some other phenomena. SensorML therefore provides a suitable standard for describing the properties of context sources.

Now that the background to the modelling requirements relevant to this thesis has been discussed, the next section will explore briefly the concept of usability, as the following chapter will show that to support the development and evaluation of SBAs, SBA developers require usable tools for testing and evaluating them. Therefore, this chapter will end with a brief background into usability.

2.7 What is meant by usability?

For any newly developed object, be it a model, a tool, or anything a human interacts with, the usability of the object is an indicator of whether it has met its design requirements. The term usability can often be interchanged with usefulness (Landauer 1995). Useful features enable users to “do things” and usable features those that make the “doing” easy. Nielsen defined usability as a sub category of usefulness and has given what may be the best known measure of usability (Jokela, Iivari et al. 2003). These are: easy to learn, efficient to use, easy to remember, few errors and subjectively pleasing (Nielsen 1993).

In 1998 the International Organization for Standardization (ISO) released a multi-part standard for aspects of human computer interaction. In this they defined the usability of a product as “the extent to which the product can be used by specified users to achieve specified goals with effectiveness, efficiency, and satisfaction in a specified context of use” (ISO 1998). Usability studies, or testing, is therefore the evaluation of a product, or model, by testing it with real users (Nielsen 1993). The ISO definition has now becoming the de facto standard in usability studies and as such will be used in this thesis (Jokela, Iivari et al. 2003). Now that the term usability has been defined, a methodology for conducting usability testing is needed.

2.7.1 Usability Evaluation Structure

Building on the usability definition identified in ISO 9241-11, ISO also adopted the Common Industry Format (CIF) as a standard for usability reports (ISO 2006) with industry backing from companies such as Microsoft, Hewlett-Packard, Boeing, US West and Kodak (Scholtz, Wichansky et al. 2002). CIF does not describe how a usability test should be conducted, but it does make explicit the requirement that an evaluation, of the product/model, include measurements of its effectiveness and efficiency, as well as a measure of the user’s satisfaction.

CIF therefore provides a framework for conducting summative evaluations of usability. CIF makes a distinction between “summative” and “formative” evaluations. Formative evaluations are conducted during the development of a product; they are done to mould or improve the product and can be conducted without the need for a test administrator and participant to be co-present. Outputs of formative evaluations may include participant comments (attitude’s, sources of confusion, and reasons for actions) and other usability problems and suggested fixes determined through observation.

In contrast, summative evaluations are carried out at the end of the development stage. They set out to measure or validate the usability of the product. They look at comparing usable metrics and generating data to support claims about usability. Outputs of summative evaluations may include statistical measures of usability, for example success rate, average time to complete tasks, number of errors and/or number assists. It should be noted that summative and formative evaluation should not be seen as either opposing or even conflicting approaches, but rather complementary, being best used together to develop a model. This thesis shall therefore make use of both formative and summative evaluation to determine the usability of the implemented model.

2.8 Chapter Conclusions and Summary

This chapter set out to inform the reader regarding key concepts related to this thesis. These included definitions of terms like context, context source, context acquisition and processing, and context simulation. It identified the need for modelling uncertainty in simulated context as part of a method for simulating smart buildings to evaluate smart building applications (SBAs) as current methods for evaluating uncertainty in context-aware applications by improving intelligibility for users, have had mixed results.

To facilitate the simulation of uncertainty in context, the issue of uncertainty in low level context sources like sensors was explored. This identified the need for models of both the building environment and low level context sources (sensors) to support context simulation in buildings.

Current building modelling approaches using the Industry Foundation Classes were found to be lacking in the area of context source (sensor) modelling, which is required to support context simulation. The sensorML sensor modelling language does support modelling of context sources to a level sufficient to model context simulation with uncertainty. Therefore, it is suggested that a combination of IFC and sensorML can provide a basis to model context simulation in buildings. Finally, tools which support SBA developers evaluating their applications should be usable. The last section of the chapter examined what is meant by usability and identified a methodology for conducting usability evaluations of the proposed solution.

In summary, this chapter identified the key areas that this thesis will address. These are the issue of creating a usable model for evaluating SBAs when faced with uncertainty in context. This will be done through the development of a context simulation model which supports configuration of uncertainty in simulated context sources and support the evaluation of SBAs.

3 State of the Art

3.1 Introduction

This chapter begins with a state of the art review of smart building applications and location sensing systems. From this, a typical set of activities related to developing SBAs which require location context are examined. Next, some techniques to support SBA developers design and evaluate SBAs are explored and the weaknesses of these approaches are identified. Context simulation is suggested as a method which can address these weaknesses. To address the requirements of SBA developers a set of criteria for the context simulators are presented next. This will guide the reader through the following review of existing context simulators. This review will then inform the approach taken to context simulation in this thesis.

Next a more descriptive list of features of relevance to this thesis is given which results in a table for comparing the different simulators. This table is based upon a comparison framework developed by O'Neil, but is specifically focused on context simulation (O'Neill 2011). This framework will be described in greater detail after the review of the state of the art in context simulators. Finally the chapter gives a summary of the strengths and weaknesses of existing approaches and a summary of the chapter.

3.2 Smart Building Applications

A large number of indoor context-aware applications have been developed since Schilit first coined the term context-aware computing in 1994. Hong et al. provide a review of context-aware systems between the years 2000 and 2007. Among these systems, they list a number of applications and services in areas ranging from hospitals, to class rooms as well as tour guides (both indoor and outdoor) (Hong, Suh et al. 2009). Numerous other SBAs have been developed in the years since and these require different sensor technologies, ranging from security systems (Song, Choi et al. 2008) and energy management systems (Agarwal, Balaji et al. 2010) which both require passive infrared sensors, to assisted living for elderly (Wood, Stankovic et al. 2008) which require body sensors on a person's body, such as ECG, pulse oximeter and accelerometers as well as sensors placed in the environment, such as temperature, light and motion sensors, to museum tour guides which require wireless local area network (WLAN) for determining location (Tsai, Chou et al. 2010). Collectively these contribute towards a diverse set of SBA developers with various requirements depending on their particular field and the requirements of the SBA they are developing. What these developers have in common is that they tend to be persons with experience in developing software for communicating with and acting upon context generated by sensors.

Within Ubiquitous computing, location represents a major category of context (Schmidt 2002). As yet, SBA development does not benefit from a single dominant location technology such as the Global Position Satellites (GPS) (Kaplan and Hegarty 2006) offers for outdoor environments, due to the low accuracy of GPS indoors (Koyuncu and Yang 2010). SBA developers must therefore choose from a wide array of different sensor technologies for determining location with different costs, accuracy and usability characteristics, which are often sensitive to other building characteristics, e.g. its usage, its geometry and its material construction (Koyuncu and Yang 2010; Deak, Curran et al. 2012). The use of a redundant positioning architecture which integrate a number of different location systems can be used to improve accuracy (Pfeifer 2005), (Koyuncu and Yang 2010). The SBA developers must evaluate their SBA with each of these technologies, how they behave and how they interact, in the required building environments. Due to the prevalence of location context in the development of SBA's, this thesis will focus on evaluating the use of low level simulated context sources which generate location, although the model will aim to be flexible enough to support a wide range of different types of context source, should they be required by the SBA developer.

3.2.1 Indoor Location Systems and Context

Location context comes in different forms and Chalmers identifies some key issues arising around the classification of location context which are applied here to buildings (Chalmers 2011). These are: Whether the location is generated by the sensor or of another party (to be tracked or to identify co-location)? What are the terms of reference (grid system, place name, relative position from observer etc.) to be used? What is the scope (room, building, multiple buildings) that the location is required over? What are the limits of the sensors and supporting infrastructure being used (per building, multiple buildings)? And finally, are multiple sources of data available? These same considerations must be addressed by an SBA developer when considering what type of location context they require to enable the SBA.

How location systems sense location also varies greatly. Location sensing systems can be classified into two types, active (those that require tracked persons to participate actively) and passive (those that do not) (Deak, Curran et al. 2012). The vast majority of localisation systems require that the tracked person carries an electronic device, for example a transmitter, and are therefore active systems (Deak, Curran et al. 2012). Examples of these are Ultra Wide Band (UWB) (Steggles and Gschwind 2005; Ubisense 2012), Radio Frequency Identification (RFID) (Hightower, Want et al. 2000; Ni, Liu et al. 2004) and Wireless Local Area Network based systems (WLANs) (Bahl and Padmanabhan 2000; LaMarca, Chawathe et al. 2005; AeroScout 2012), Infrared (Want, Hopper et al. 1992; Harter and Hopper 1994), BlueTooth (Blip 2012) Ultrasonic localisation (Harter, Hopper et al. 2002), and use of tv signals (Zlatanova and Verbree 2005). The location of a tag (which can provide identification) can be a coordinate or a proximity (to a receiver) or alternatively be the name of a room or an area in the building.

Passive localisation systems on the other hand do not require the person to carry an electronic device. Examples of passive systems are passive infrared sensors (PIRs) (Lee, Ha et al. 2006), those that use radio frequency and ultrasound sensors (Priyantha, Miu et al. 2001), those which use action detectors, for example pressure mats (Orr and Abowd 2000) and contact switches on doors like magnetic reeds, and those which use cameras (Brumitt, Meyers et al. 2000; Lo´ pez de Ipin˜ a, Mendonça et al. 2002). Other methods to detect presence include measuring CO2 levels, humidity or temperature, light levels (e.g. blinds opening, lights or visual displays being turned on, shadows being cast) or changing sound levels (e.g. voices, music), or biometric scanners. Some of these approaches can provide identification (Orr and Abowd 2000) while others do not (Lee, Ha et al. 2006). The location of a sensed person is with respect to the area that the sensor is measuring, for example, a PIR sensor will only generate a value when the level of infrared light falling on its detector changes. These systems need not be used in isolation and combinations of systems can often be used to reduce the uncertainty in an SBA, for example, combining PIRs with contact switches (Agarwal, Balaji et al. 2010).

Collectively these location systems provide location context which either includes identity or does not, and provides location as a coordinate, proximity or an area/zone. SBA developers would like to use the minimum amount of sensors and sensor types at the lowest costs while still meeting the requirements of their SBA. Selecting and evaluating these represents part of the process of developing an SBA. The next section will look specifically at the tasks an SBA developer must undertake when developing and evaluating an SBA in more detail, which includes the selection and configuration of low level context sources like location sensors.

3.2.2 Smart Building Application Development

The process of SBA development typically begins with a concept for the SBA. This consists of asking the question, "what functionality is the SBA to support?" e.g. is it to provide security alerts, automate building systems like HVAC, or perhaps provide navigation to building occupants. Another question which must be answered is "what types of building is it expected to function in?" e.g. an office building, a hospital, or even a particular type of room. Next the SBA developer must consider what context is required to support the SBA. For a location-aware SBA, context is needed in the form of location, so the type of location context required must be determined, for example a presence event tied to a particular location, and this could be provided by a PIR. Or perhaps a coordinate with identification is required, and so a context source which provides this type of low level context is needed, for example, Ubisense. Further configuration of the context sources may be also be required, for instance, to adjust the rate at which context is generated by a context source or adjust the placement of sensors to best meet the SBA requirements.

How this data is to be communicated to the application needs to be considered also. Systems like Ubisense can communicate with an SBA through a local area network. For mobile SBAs, wireless communication is required. If the application is to move between buildings, this connection must be established and maintained. The application may also need to have a model of the physical space. For example, a coordinate location may need to be associated with a particular zone or areas in the building, so that when the location falls inside that zone, the application can act. Where the SBA must change the state of the building, communication must also be established with the building systems in question, for example HVAC systems or automated doors. Once the SBA requirements have been identified, which include the type of buildings(s), context(s), the context source(s), and the communication between the context acquisition infrastructure and the SBA have been identified, the next stage is implementing the design of the SBA so that it can both communicate with the context acquisition infrastructure, process the context and also communicate with any devices in the building.

Once these tasks have been achieved, the next step is to evaluate the SBA to test its functionality in a variety of typical situations with users exhibiting different behaviours. These evaluations are then used to refine the different functionalities of the systems which enable the SBA. This includes examination of sensor configuration, for example, where the sensors are exhibiting too much uncertainty to enable the proper function of the SBA, the sensors may need to be re-configured or repositioned. The sensors may also be replaced with sensors which provide different types of context, for example a coordinate rather than a proximity location, or replaced with sensors which provide the same context type but with higher levels of precision or at higher rates. Context processing may also be examined to determine if change to it will enable the proper functions of the SBA. The building itself may even require re-design to enable an SBA.

The costs of conducting evaluations in smart buildings may be beyond SBA developers who do not have access to them. For those that do, managing all these considerations still presents a considerable challenge, as the SBA developers may not be experts in the technologies that enable SBAs, for example sensors.

3.3 Evaluating Smart Building Applications

In the previous section the complexity of SBA design and evaluation was identified. Developing solutions to support context-aware application design in the area of smart environments, is in itself a major avenue of research (Tang, Yu et al. 2010). Due to the costs of conducting field-based evaluations, numerous methods have been suggested to support the evaluation, ranging from experience evaluation (Reilly, Dearman et al. 2005), paper prototyping (Carter and Mankoff 2005) and Wizard of Oz (WoZ) techniques (Reilly, Dearman et al. 2005), (Li, Hong et al. 2007). The major drawbacks of these techniques are that they require the assistance of a second person to either change the pages, or play the role of the wizard which requires an investment of time from the wizard, who must also have the necessary skills to conduct the test. For mobile applications, monitoring or following participants can be difficult to manage and also limit the realism of the evaluation (Reilly, Dearman et al. 2005). Simulation has been proposed as a method for evaluating smart environments to aid with the evaluation of context-aware applications, and a number of context simulators which take this approach have been developed in the past decade. Before these context simulators are reviewed, a set of criteria based on the requirements of SBA developers shall first be explored.

3.3.1 Criteria for Context Simulation

As highlighted in the previous sections, the requirements of the SBA can vary according to its particular domain. To begin, a context simulator may be required to support different scales of smart buildings, from homes with one or two floors, to large buildings with multiple floors, rooms, sensors and occupants all interacting with each other and the SBA. Within these environments, different scenarios should be testable which are typical to the SBAs requirements. For example, a security tracking system which tracks authorised and un-authorised occupants would examine different occupants walking in those areas which have sensor infrastructure in place to track them.

The type of context sources the SBA requires should also be supported. The underlying context simulation model should therefore be flexible enough to meet this requirement where necessary. This means that the context simulators must be able to support the simulation of context. Context simulation consists of simulating context sources which generate low level context generated by sensors. For example location, presence, temperature etc. and higher level context types generated during context processing, like the combination of two coordinate locations to infer that a person carrying the tag is “running” or “sitting”. The SBA can then process this further to determine the situation, for example that the person is “at a meeting”.

The value of the simulated context has a relationship with both the environment, the occupant of the building and the context source type. For example, a context source which generates a location based on the position of a tag carried by a user in the environment and properties of the context source, or a context source which generates a temperature value of 30C, depending on the room it is placed, the number of occupants in that room and the properties of the context source. These in turn will have a level of uncertainty associated with them, in the form of precision (as defined in Chapter 2) and rate, with any additional delay, which are themselves related to the context source type and its placement in the environment. The context simulator should therefore account for these factors and this thesis will evaluate the former location context source, but it will also set out to account for the latter in the design.

The context simulator must also support SBA developers modelling and configuring the properties of the simulator. The modelling of interactive environments is a complex process in itself. Methods which reduce the burden of modelling these environments are therefore preferable over those which do not. By integrating the context simulator with existing Building Information Modelling (BIM) standards, the Industry Foundation Classes (IFC), it should be possible to import existing building models. As these models tend to be developed by the architects themselves, they should also be accurate in their representation of the actual building. Where new models must be developed, these can also be exported, increasing the potential for models to be reused in other areas of the building life cycle and reducing the effort for other disciplines as part of a whole building design cycle. Where IFC does not meet the modelling requirements necessary, for example, of sensors to enable context simulation, the sensor modelling standard sensorML provides capabilities for modelling the properties necessary for context simulation. This has the added benefit of integrating the building model into the wider sensor modelling community and the potential for considering areas beyond smart buildings and into outdoor smart environments.

The process of creating, placing and configuring simulated context sources must also be considered. Tools which support SBA developers by reducing the complexity of creating and configuring simulated context sources and which also facilitate testing and evaluation of SBAs a controlled way are needed. These tools must also support configuration of uncertainty, so that SBA developers may evaluate their SBA when faced with varying levels of uncertainty in context. For SBA developers who are not experts in the underlying context acquisition systems, these tools should have a level of usability that supports them in this process. Tools which improve the evaluation process by providing analysis capabilities can also make the context simulator more effective, for example by improving understanding of uncertainty and its impact on the SBA.

The next section examines the state of the art of context simulators with specific attention to how each context simulator meets the above criteria. It will also identify other useful features which will aim to be supported as part of a flexible context simulation model. This will be followed by a more in depth discussion of these features and how they are applied, culminating in a comparison framework to highlight the variation in the context simulators and the approaches they take to context simulation.

3.4 Context Simulators

A number of research efforts have looked into developing simulation suites that simulate context within smart environments. These are classified here as those which support non-interactive context simulation and those which support interactive context simulation. Interactive context simulators are those which support user interaction with the environment through an interface, for example, using a mouse and keyboard to control an avatar in a virtual 3D environment. This avatar in turn generates context as it interacts with the environment, for example, using the location of the avatar to generate coordinate based location contexts. Non-interactive context simulators have no such user interaction. Non-interactive context simulators can therefore be run in real time or at different speeds. This can be a useful feature when analysing the causes of error. Interactive simulators can also support this type of functionality through replay of a simulation, but the interactive aspect is therefore lost and this is purely a feature of analysis.

To generate location based contexts, the simulator must also simulate user behaviour. This simulated user behaviour, for example the movement of a programmed "bot", is used to generate location context. This thesis therefore makes a distinction between bots and avatars, as avatars are controlled by humans in real time, allowing them to interact with the environment and bots behave according to a set of predefine rules and their reactions are therefore deterministic (Sanmugalingam and Coulouris 2002; Fahy and Clarke 2004; Nishikawa, Yamamoto et al. 2006; Reynolds, Cahill et al. 2006; Jouve, Bruneau et al. 2009; Consel 2011).

As the users interaction with the application in situ is an important aspect of context-aware application design (Reilly, Dearman et al. 2005), a number of interactive context-simulators have been developed which support users moving around and interacting with a virtual smart environment and generating user driven simulated context (Barton and Vijayaraghavan 2002; Bylund and Espinoza 2002; Shirehjini and Klar 2005; Lertlakkhanakul, Choi et al. 2008; Prendinger, Brandherm et al. 2009). The context simulation model in this thesis will be designed to be flexible in its support for both, while focussing on interactive simulation due to the importance of exploring real user interacting with applications as part of the evaluation process. The review begins with non-interactive context simulators.

3.4.1 Non-interactive Context Simulators

3.4.1.1 Generic Location Event Simulator

Non-interactive context simulators began appearing in the literature as far back as the early 2000's. One of the earliest to appear in the literature is the standalone Generic Location Event Simulator (GLS) (Sanmugalingam and Coulouris 2002). GLS is designed for the visualisation, scalability testing and evaluation of location-aware event driven middleware and applications. In the paper they define objects called "locatables" whose location can be sensed. They then developed a simulator that models the behaviour of locatables in a simple model of a physical space. Simulated location context is generated in a format matching available sensor systems, to support integration with sensor enabled applications which require data from specific sensor types.

Different models can be plugged in and out of the system as required. These include mechanics and querying models to help define the behaviour of locatables, sensor and environment models to simulate the unique and dynamically changing physics present in a room, and a world model to model, for example, airports, railway stations etc. Other data relevant to the use of locatables may also be modelled, like the position of doors or the geometry of rooms. Human behaviour may also be defined using complex composite behaviours and then assigned to locatables. The simulation is synced to a global clock which controls the frequency and ordering of events being generated. For large scale systems, this is important if simulation threads need to be distributed for scalability reasons.

In GLS the simulated outputs may also be linked with outputs generated from physical sensors in the real world, supporting "hybrid simulations". This type of simulation can be useful when one or more sensor/context events are not available to an existing sensor/context infrastructure which can then be provided by simulation. They have simulated a number of sensor types, those which measure *geometric position*, using ActiveBadge (Want, Hopper et al. 1992), *symbolic position*, using a room that contained ActiveBadge, *vision*, using TRiP (Lo´ pez de Ipin˜ a, Mendonça et al. 2002) and *proximity*, using login/keyboard monitors, etc. GLS also provides some graph based visualisation of the behaviour of locatables to support the analysis process.

The drawbacks of GLS are that it does not specifically look at how real users interact with the environment, instead relying on behavioural models. While the paper provides an evaluation of the performance of GSL, and demonstrates scalability is possible with a large number (up to 50) simple pluggable locatables, there is no mention of how an evaluator uses GLS and there is no evaluation of how easy it is to define or configure the models or whether the simulator met the requirements of actual users. Models are also not based on any existing standards. This puts the onus of modelling new environments and simulations entirely on the developer. Also, while an environmental error model based upon pluggable statistical distributions is mentioned as a design criterion, it isn't apparent in the presentation of the implementation or evaluation.

In conclusion, GLS provides a scalable framework for simulation of location as part of an evaluation process involving context middleware and applications. It also addressed the need for supporting the evaluation process with visualisation tools for analyses of the simulation, although no evaluation of this is given to determine its effectiveness. They also describe pluggable models to provide flexibility to the simulator. This includes the concept of an environment and error model, which are identified as a necessary part of the context simulation process to support evaluation of applications.

3.4.1.2 Cass

In 2004 Clarke et al developed a context-aware simulation system for smart homes called “Cass” (Fahy and Clarke 2004). Cass sets out to evaluate smart home applications by detecting rule conflicts in context information, and determining the best sensors and devices to meet the requirements of the smart home. They set out to simulate both primitive and complex context. Primitive context is equivalent to low level context and is information acquired by physical sensors, with data like time and location included. Complex context is high-level context information, which contains additional knowledge, like relationship to users or objects. In creating this information, reference data relating to the primitive context information must be accessed, for example, the description of a tagged object (user model or product information).

Cass provides a 2D GUI with a visual representation of a home floor plan, with names of rooms and appliances. The 2D building model is used purely for visualising the home. The tool supports configuring, deleting and adding simulated context sources as well as visualisation of sensor placement and outputs, for example, a fire alert triggered in a room. It also supports definition of rules and actions. Actions are described as a set of devices. Rules are triggered when certain conditions are met, for example, when a simulated temperature is below a certain value, turn on a simulated heater. The simulation is defined by a sensor model which models sensor type, value, position and relations. Once again, the sensor model is not based on any existing standards, so interoperability is not a consideration in Cass.

The configuration tool also supports defining movement of what it terms “avatars” but are in fact better described as programmable bots as no real time interaction is supported. Cass also provides an additional tool for streaming all context events to a display so that the evaluator can see an overview of them. A case study which examined rule conflicts for typical services in a simulated smart home is presented. These rules are a fire alarm, an air conditioning unit, a dehumidifier and an alarm. They defined some rules and used these to determine conflicts in those rules.

In conclusion, while Cass is useful for determining rule conflicts in simulated devices, it does not deal with the issue of user interaction with context sources. It also does not address the issue of uncertainty in contextual data at all, assuming that events triggered by the avatar are reliable. There is no discussion of using standard models, which means that new simulation environment must be developed from scratch each time they are required. Cass also does go some way to supporting the developer in configuring the simulation by developing a tool for placing and configuring the sensor model. As no evaluation of the simulator to determine its usability is undertaken, it is difficult to validate whether the approach could be used by SBA developers.

3.4.1.3 SENS

SENS is a sensor, environment and network simulator (Sundresh, Kim et al. 2004). It is not concerned with application evaluation, rather with evaluation of placement of sensor nodes and communication between them. We include it here though as its approach to simulation is beneficial to understanding issues related to context simulation. SENS features a modular architecture to permit simulation of a range of different WSN scenarios. In particular, components to support sensor nodes communicating via wireless broadcast in an environment represented by tiles which they use to modulate sound and radio propagation.

To enable simulation, they use values from real sensors to represent the behaviour of component implementations. Users can assemble application-specific environments with different signal or sound propagation characteristics for sensors using a grid of interchangeable tiles to model the environment. Different tiles can be modelled with different propagation characteristics and therefore SENS takes into consideration the relationship between the environment and the behaviour of the sensors. Sensors each have a virtual clock to control the ordering of sent messages. Drops and collisions can also be simulated between messages. Drops are based on an error probability. Collisions between packets can result in errors (by either dropping packets or introducing random bit errors in the appropriate part of the packet), the granularity of which can be selected to trade-off between performance and accuracy.

Both collisions and drops are affected by an environment component. Developers use this to vary the simulated environment much more simply than attempting to configure a real physical environment to establish the same variations. The environment model effects radio prorogation and thus the measured signal strength for sensor nodes. Tiles receive and propagate a wave based on the source location, the amount of energy contained in that part of the wave and the delay profile along the edge which the wave entered the tile.

Sundresh and Kim have looked at simulating both 2D and 3D propagation, but their evaluations only examined delays in experimental sound propagation against simulated delays. Here they note that discrepancies between the two are due to both “random error”, although they do not specify a cause, and by variations in temperature and humidity, which are two parameters not included in the SENS simulation model. The main part of the evaluation was to examine the performance of SENS.

A SENS API is mentioned, but there is no mention of how this works or whether it or the SENS platform would meet the requirements of application designers as identified at the beginning of this chapter. There is also no mention of using standard models to define sensors to support interoperability with other simulators.

In conclusion, SENS provides a detailed approach to modelling aspects of sensor simulation like propagation and communication. Its use of a modular, grid based simulation which incorporates the environment to simulate uncertainty based on probability distributions points to a methodology for introducing things like random error into simulated context.

3.4.1.4 SimuContext

SimuContext is a context simulator for providing a range of contexts to help with context-aware application evaluation (Broens and Halteren 2006). It looks at specifically simulating Quality of Context (QoC) issues. As context information represents real-world situations, QoC gives certain quality indicators and SimuContext uses the following QoC parameters: precision, correctness, trustworthiness, resolution and up-to-dateness. It aims to provide rapid repeatable testing of application behaviour using a range of realistically simulated context sources. Developers use a SimuContext interface to specify the behaviour of simulated context sources and use these as inputs into their context-aware application. This interface gives different tabs to configure different aspects of the context simulation process and text based feedback on generated context events.

Broens and Halteren give high level examples of the context model, as well as some description regarding variation of context and introducing random values and events to simulate uncertainty. No description of how these are actually modelled is given though other than that a simulated context source will produce some value which an application can either request or subscribe to. The latter can include conditional values to support context processing, like only notify if temperature is below x. Simulated context can also be generated by using log files of existing context events. They claim that the context model is realistic and so it should facilitate the replacement of simulated context sources with physical ones, although no evidence is given about how realistic they are.

In conclusion, SimuContext provides an interesting framework and does look at the requirements of application designers by providing a graphical user interface to support ease of configuration and testing. They also address the need to support configuration of aspects of context related to uncertainty, like QoC. No mention is given of any evaluation or whether the interface met the requirements of application developers. Also, there is no mention of using standard models to improve interoperability with other context simulators.

3.4.1.5 UbiREAL

UbiREAL is described as a "realistic smartspace simulator for systematic testing" (Nishikawa, Yamamoto et al. 2006). UbiREAL provides functions for deploying virtual devices; simulating communication between them and reproducing physical quantities, e.g. temperature. UbiREAL sets out to meet the following criteria, 1) Support to design a smart space, 2) Realistic Context Generation, 3) Graceful Visualisation, 4) Software Compatibility and 5) Systematic testing. A typical scenario for UbiREAL involves an application which controls several devices in a smart building depending on the goals of several occupants. These goals are met by adjusting the ambient temperature and controlling music and television. UbiReal therefore addresses issues of combining the different requirements for these devices for the different occupants who may be sharing the same space.

UbiREAL supports the evaluation of such a scenario by providing a GUI for modelling and visualising the virtual Smart space. The graphical user interface (GUI) allows users to place virtual devices in a 3D virtual space. The environment and appearance of such devices is handled using existing commercial 3D modelling software or can be substituted using VRML. The 3D space is constructed from building floor plans. Devices can be either static or movable. Moveable objects, e.g. an occupant, require routes to be specified. As device states change, these changes are made visible through the 3D visualizer or through a 2D overview.

UbiREAL also provides a network simulator. This supports simple simulation of wired and wireless communication. For example, in the case of wireless communication, the position of devices and moveable objects is taken into consideration, i.e. communication is allowed when the distance between devices fall under some set value. Error is also said to be introduced according to the condition of communication, but how this is introduced is not explained. Communication between virtual and real devices is handled using TCP/IP protocol.

For the purposes of context simulation, perhaps of most interest is the approach to simulation of what they refer to as physical quantities and which they relate to physical phenomena like temperature, humidity, electricity and radio as well as phenomena like acoustic volume and illumination. They employ physical quantity simulators, each based on an appropriate mathematical formula for simulating a physical quantity. Each simulator periodically calculates the latest value of the physical quantity and sends it to subscribers if it has changed. The simulator uses parameters, such as room size, occupants, devices states, time between calculations etc. to calculate the new value. The model is not based though on any existing standards and so interoperability with other simulators is not explored.

To support systematic testing, they provide a formal model for service specification of context-aware applications. The specification is given as set of rules and a set of propositions. The example they give specifies a rule for an air-conditioning system. If a room exists at a certain temperature as defined by a user, and the user is in the room, turn on the air-conditioning. Unlike Cass, they address conflicting rules during specification. The proposed test function targets rule-based applications as the specification is derived from the application scenario.

The paper provides an evaluation of the systems performance. They test 25 devices, each assigned with two rules. As the simulation is integrated into the VR simulator, they must ensure that it does not impact on the realism of the visualisation. At this scale there is only a small influence on frame rate of their simulator, thus real time simulation of context is supported. The paper gives no indication regarding whether it has been successful in evaluating an actual smart space or how easy developers were able to create new simulations. The placement of context sources is described, but little data is given on how a developer uses the interface to model them, other than that they can plot routes for bots before they run a simulation.

In conclusion, UbiREAL does provide an interesting approach to integrating a 3D visual environment into a context simulator. Also, its approach to parametric modelling of context simulation sources supports flexible simulation, as different aspect of the environment can be used as input for the simulation process. UbiREALs use of VRML offers the benefits of a standard model approach as existing VRML models can be imported into the simulator, thus reducing the burden on the developer to model from scratch. While VRML import and export is supported by many CAD building modelling tools building models (Autodesk 2012; Graphisoft 2012), it is not common to find VRML models of existing buildings, which tend to be in 2D. Also, VRML does not support taking inputs from the building model as parameters to context simulation as VRML is purely for visualisation. BIM on the other hand does support such parametric simulations using aspects of the building like wall properties (materials, thickness, etc.).

3.4.1.6 Generic Simulation Tool

Reynolds et al. have presented initial work on a generic simulation tool suitable for the many scenarios encompassed by ubiquitous computing, such as simulation of sensors, actuators, and the environment (Reynolds, Cahill et al. 2006). In addition, an emulation framework for middleware and software under development is provided which interfaces with the simulation tool. In the paper they identify two key goals of simulation. To begin, given the diversity of scenarios in ubiquitous environments, the simulation model must be "flexible and sufficiently general, yet extensible enough in its base model to support these". They warn though, that if the model is too high level, there is a risk that it will be unusable because they do not meet specific domain requirements. They also identify the need for usable tools which expose the interface in an intuitive manner; therefore usability is a second goal. Finally, they identify the need for scalable simulation to account for the potentially large scenarios involved in ubiquitous environment, which are both indoor and outdoor.

Reynolds et al. go on to identify a set of abstractions that are required by the wide range of scenarios envisaged in ubiquitous computing. These common components include sensors, actuators, applications utilising these sensors or contextual information derived from them, and the environment in which these components exist. Changes in the environment, user's position and activities, and user input typically drive sensors; applications react to this input and offer feedback into the environment or the user by way of actuation. Using location, they then develop a grid-based approach to modelling the environment. Sensors are modelled upon this grid-based layered approach. Initial phenomena can be converted using a "pipeline" to the required granularity and format for output. This pipeline could be used to introduce error into simulated sensor data for example, using a Gaussian distribution. By combining these features into a pipeline, they claim that a very accurate simulation can be achieved, in that they accurately simulate inaccuracies in low level context.

Using a layered approach to modelling the environment, they claim flexibility is achieved, as many layers can be juxtaposed and interdependencies between layers specified by simulator users. Each layer is parameterised so that it represents the scale of the environment (i.e. a floor, or a room) and the granularity of the grid within the layer. For example, a layer modelling environment temperature can be linked into a layer modelling sensed temperature. The architecture of the simulator also supports incorporation of other simulators where required, although the only example they give is a network simulator and make no reference to the use of simulators for simulating context values. Using this system they have designed a proof-of-concept simulation for an Intelligent Transport System (ITS), in which they have simulated sensors (GPS), actuators (traffic lights) and applications which control those lights. The paper discusses a Replaceable Code Emulation Unit (RCEU) to support multiple APIs for configuration, but little explanation of how this works or appears is given.

In conclusion, the generic simulation environment details many of the criteria which have already been identified earlier in this thesis. Firstly, they recognise the importance of a flexible and extensible model to meet the various requirements of SBA developers. And secondly, they recognise the importance of the usability of the system for SBA developers, although no evaluation of the system's usability is given and no support for the evaluation process is discussed. They describe a layered approach to simulation which can be applied to support simulation of different phenomena, like temperature. They also detail a process for introducing error into the simulated values from sensors using Gaussian distributions. Unfortunately, there is no clear indication of how any of the components are actually modelled or whether they are interoperable with existing standards and as the simulator use case addresses outdoors and traffic simulation, little consideration is given to issues around smart building modelling.

3.4.1.7 UbiBuilding

Schmidt et al. present an open architecture for life-cycle support of context-aware applications in context aware systems which can scale over buildings with multiple floors and include many sensors and actuators (Oh, Schmidt et al. 2007). They define the life cycle support for these types of context-aware system as simulation, installation, debugging, and maintenance, adding new entities, removing old entities, and upgrading system components. To explore issues related to these they implemented a component called "UbiBuilding Simulator" which they refer to as a "building simulation system" and which is implemented in Macromedia Flash. This they used to simulate a building which comprises 5 floors of 19 rooms with in total of 41 actuators (9 doorlocks, 19 lights, and 13 heaters) and represents this in 2D from a side on perspective. The simulated buildingincludes rooms and actuators (door locks, heaters, lights) but no information on how the building is modelled is given.

UbiBuilding provides a text editor which supports developers adding, deleting and modifying rules in plain text. Reconfiguration requires no changes to the source code. To make it feasible for developers to understand how and why a context-aware system is performing certain actions, a component to help with explanation and accountability was added. This displays things like the intention of simulated bots and their current activity as a point of view from an outsider. UbiBuilding simulates three different sensors: a wearable activity sensor which can detect a user's activity and posture, an environment based status sensor which detects when a user is logged onto a pc, and an identify sensor on a mobile device. The simulated sensors generate context by clicking on buttons on the interface. The formats of the context can be changed depending on the type of sensor being simulated, meeting application demands.

These context events are described using a context model based on 5W1H (Jang, Ko et al. 2005; Jang and Woo 2005) which represents user-centric contextual information in terms of Who, What, Where, When, How and Why, and in varying stages, from preliminary context, to integrated, to conditional and finally combined. The paper presents an evaluation of this model, but did not set out to evaluate the simulator itself. In conclusion UbiBuilding is a basic context simulator, as context events are generated by clicking buttons on the interface, the simulation is not very realistic. No description of modelling of the environment is given either. UbiBuilding is included here for completeness.

3.4.1.8 DiaSuite, DiaSim

DiaSuite is a suite of tools which aim to cover the development life cycle of pervasive applications (Cassou, Bruneau et al. 2010; Consel 2011). The suite incorporates a design language (diaSpec) and a simulation environment called DiaSim which is integrated with an open source context simulator called Siafu (Miquel 2012). DiaSpec allows an expert to define a taxonomy by declaring the required entities of a system. Their definition requires an area expert. Each class of entity is characterised in terms of the types of data that the entity gathers from the environment and the actions that the entity supports. The data gathered from the environment by the entities is refined by context components to match the application need. These are then passed to a controller which makes decision by triggering entity actions, which are also declared in the taxonomy. Attributes can also be specified for an entity (e.g. location, ownership). The programmer is then required to implement the entities and components. This is done using a Java generated programming framework generated by the DiaSpec compiler.

DiaSuite also provides a graphical editor and parameterized simulator called DiaSim (Jouve, Bruneau et al. 2009). That is, the simulation model is made up of several parameters each of which affect the behaviour of each simulated device. For example, a light device may produce luminosity which feeds back into a light sensor. For sensor simulation, DiaSim employs “stimulus producers” that are classified by the type of one or more sensors. Each stimulus has a set of rules defining how they behave in terms of space, time and a notion of intensity. These can be modelled by a mathematical function which describe periodic behaviours (e.g., cars going to workplaces each day) or discrete behaviour (e.g., people moving from one room to another). These functions are typically provided by experts in the sensor technology being modelled. Other types of stimulus can be simulated by replaying logs of measurements collected in an actual environment. Where measurement logs are not available, an approximation model can be defined which attempts to approximate the value of the sensor reading, for example, by converting a coordinate into a context event like “room x”. DiaSim also supports modelling of the environment (walls and areas), although whether these form part of the parametric simulation is not clear.

The DiaSim 2D GUI interfaces with Siafu to support placement of context sources. Siafu is a context simulator which supports 2D rendering of environments, time control functionalities and the definition of agent, world and context models. The agent model supports way point to way point movement and random movements. The world model defines areas which the agent can then move toward and away from. DiaSim interfaces with Siafu context model so that sensors can be dragged and dropped onto the Siafu map and can be configured to detect stimuli and generate context, for example context based on the movement of agents provided by siafu. The 2D render displays an imported 2D image of the building and displays agents and services on this. It also displays the state of the primitive services by displaying a bubble of text above services, for example, when sensors generate events.

DiaSim ran a number of case studies which explored the development of a newscast app which displays relevant information on an LCD display based on the nationality and other information about persons in proximity. A security app which alerts when intrusion or theft takes place was also evaluated. Another evaluation looked at a light manager which controls lights based on outside luminosity, school calendar and occupancy. They applied this to a simulation of a large building with 3 floors of lecture halls, labs and recreation rooms. DiaSim allowed the paper authors to coordinate logic at a large scale for the simulated building, combining 75 services, 48 stimulus producers, 200 people and 6 applications, although little detail is given on the type of service, stimulus produces, or people. They report that findings from these evaluations allowed them to make the newscast app less sensitive to people passing the LCD display, so that they were required to stand a while before their presence had an effect, and optimize air conditioning using occupancy and class schedules.

DiaSim examines a number of methods to support context simulation, like the use of approximation models, mathematical functions and playback to simulate context, as explained above. Unfortunately, no samples are given regarding how this is done. There is also no description about whether it supports modelling of uncertainty in context, stating that the onus of modelling context sources falls on the developer. This may not be a realistic requirement where SBA developers are not experts in all the systems of a smart building for example. Also, no mention is made of whether the models (stimulus produces, or environment model) have been made to be interoperable with existing standards, which would reduce the burden of the evaluator to model the entirety of the smart building if such models already exist, for example for the building. And while DiaSim is intended for both outdoor and indoor context-aware systems, there is not much description given regarding how they would actually model the indoor environments, again saying only that their approach can integrate expert knowledge. Finally, without any evaluation given of how real users found using the system, it is hard to determine how effective it is for SBA developers themselves.

In conclusion, DiaSuite is a very interesting contribution to the field of context simulation. It meets the goal of flexibility through its simulation approach, bringing together a number of different approaches to meet the varying requirements of context-aware application development.

3.4.1.9 InSitu

The InSitu Toolset is a tool for facilitating cost-effective, repeatable and easily configurable experiments to support thorough testing of context-aware systems, particularly those which support indoor location tracking and exhibit actuations in the deployment environment (O'Neill 2011; O'Neill, Conlan et al. 2011). The main goal of InSitu is to provide experiential feedback to a designer of SBAs with a focus on identifying unwanted behaviour in context-aware systems. Such unwanted behaviours can either be problematic for the user, for example a light turning off and leaving the person in the dark, or result in the absence of beneficial behaviour for the user, for example a light not coming on when you are walking in a corridor.

To achieve this goal, a framework has been developed to support simulating indoor environments and analysing the context generated from the interaction of users with that environment. The analysis is achieved through the combination of five models, a system behaviour model, a spatial relations model, an activity boundary model, an environment model and a context model. The system behaviour mode is based on Drools (Drools 2012). Drools is an integrated logic platform, at the core of which is a rule engine which can be used to infer conclusions about the state of the design space based on the system state and surrounding contextual space. A Drool vocabulary is used for modelling facts which can be reasoned about using the rule engine, for example, if a visitor is in an area which is only for authorised staff, then a security system may have failed. The spatial relations model models buildings as a hierarchy of contained spaces (rooms, atriums) both horizontal and vertical. Each containment space must also be linked by doors or windows. Areas (which are equatable with zones) can also be defined, which need not be enclosed but indicate areas of interest. The model is encoded using XML. The activity-boundary model described the activities performed by users of the building, the location of those activities and the paths they take between activities.

The environment model is provided by the Tatus platform (O'Neill, Klepal et al. 2005). Tatus provides an interactive 3D environment using the Half-Life 2 games engine (HL2 2012) which is adapted to work as an interactive context simulator (O'Neill, Klepal et al. 2005). Tatus is also used as the basis for the SimCon tool-set and is discussed in greater detail in Chapter 5. Using an editor called Hammer, provided with the HL2 SDK (HL2 2012), an interactive model of the building can be created. Sensors can also be created, for example to trigger an event if an area of floor is walked on, or to return a coordinate in a particular area. This data on the state of the VR environment can be accessed at run-time through a proxy. Using the framework a test designer can develop a building environment, populate it with location sensors, define rules about the environment and then run a test to generate alerts when unwanted behaviours occur.

InSitu has been evaluated as a case study which involved two types of unwanted behaviour of the automated lighting, i.e. lights not turning until a person was at the exit of a floor (meaning they are left in the dark and also resulting in energy wastage). InSitu was also evaluated as a user trial examining ten users (five teams of two). The user trial did not examine the design of the system (i.e. develop code or position sensors), as it is assumed that the system for testing has been implemented and is ready for testing. Each user was then required to write pseudocode to ensure situation specifications could be captured for each team. They then were required to convert the code to Drools and create an alert report which would be triggered if unwanted situations occurred. The participants then modelled the behaviour of the bots using an excel sheet to define activities and transfers this to Hammer. The final task was to review the alert report after the simulation was run. The simulation was left run for a 12 hour period before they could complete the last task. Metrics were gathered in the form of questionnaires and time to complete tasks. The trials found that users could achieve the tasks, although there were room for improvement with respect to efficiency, and possible integration of aspects of the tool, for example the excel spread sheet for capturing activities into Hammer.

In conclusion, the strength of InSitu is that it provides a configurable simulation for providing detailed feedback on unwanted behaviours through the use of a VR simulator, which has been evaluated to determine levels of usability. The VR environment can generate context based on the location of bots in the VR environment, demonstrating the generation of presence, proximity and coordinate location. It does not examine any other methods for generating context or different types of context and does not examine the issue of simulation of uncertainty and its effect on application behaviour as its aim is prototyping the SBA rather than supporting its redesign, for example, through repositioning of context sources. As such, there is little detail given regarding properties of the sensor model, other than that they can be placed in areas and either be event based or return coordinates at a rate no higher than the frame rate of HL2. InSitu also does not examine how real users interact with the system in real time, rather relying on bots to generate context to determine when unwanted behaviours occur. InSitu does provide an interesting model for organising space which has conceptual similarities to building information modelling, like the definitions of areas. No consideration is given to mapping these models to existing standards or looking at existing standards to model the sensors and their behaviour. All feedback regarding simulation is also done via textual reports, so no additional visualisation tools have been developed to further support the analysis of the SBA behaviour for developers.

In this section we have examined a number of existing context simulators. The next section will look at interactive context simulators, after which a full review of the existing context simulators and features considered important for context simulation.

3.4.2 Interactive Context Simulators

A number of research efforts have looked at supporting user interaction with smart environments using visualisation. This are listed and reviewed below.

3.4.2.1 QuakeSim

(Bylund and Espinoza 2002) introduced a tool called QuakeSim which makes use of the Quake III Arena to simulate a 3D environment. The Quake engine allows multiple participant avatars to connect and become actors within the environment. Each participant has their own interface which provides a first person view on the environment and the ability to interact with objects in the simulated environment. QuakeSim provides tools for building new environments, modelling avatars and objects. The Quake III engine was modified to extract context in the form of position and altitude to simulate different types of sensors. The simulator was then used to test the context toolkit (Salber, Dey et al. 1999) for gathering, aggregating, interpreting and publishing sensor and context information. QuakeSim was also used to evaluate the GeoNotes (Espinoza, Persson et al. 2001) application. GeoNotes is a location based system for delivering reminders in the form of electronic notes etc. based on a user's location.

QuakeSim was the first context-simulator which demonstrates the usefulness of using interactive VR to provide real time location to a live system (Geonotes). It did not set out to explore the issues of supporting SBA developers doing experimentation with a range of SBAs. Also, there is little detail provided on how context or sensors are modelled or simulated other than that they use the location provided by the simulator to generate location context in the form of a coordinate or a presence in a certain room (by correlating with a floor plan) and by then integrating this with the Context Toolkit (Salber, Dey et al. 1999). It therefore does not address uncertainty in context. Also, no indication is given either on how these tools function, their ease of use for the SBA developers or the effort involved in modelling new environments and sensors. No additional tool support is given either to aid the evaluation process for SBA developers either.

3.4.2.2 UbiWise

UbiWise (Barton and Vijayaraghavan 2002) builds on the work of Espinoza by also making use of the Quake III games engine in order to simulate 3D environments. Prototypes of new devices and protocols are simulated with a Java program. The 3D environment view is maintained by UbiSim. UbiSim is a modification of the Quake III Arena which includes location reporting code from the QuakeSim project (Software 2012). The GtkRadiant graphical tool (Duffy 2012) was used to create a simple room to test an application developed on Wise (a component of UbiWise). Wise is a 2D Java application which allows applications to be modelled by designers. A device is simulated by a 2D canvas which has areas (buttons) tied to specific Java methods. The overall effect of Wise is to use a personal computer to simulate interaction with another computer, e.g. a digital camera, with different controls and outputs. These two application combined give UbiWise.

Designers can use UbiWise to emulate an application which corresponds to objects in Quake III. These can then be tested inside the 3D simulator. Sensors, networks of sensors, and location sensing technologies can all be simulated and various aspects of the technologies can be studied one at a time, but no explicit explanation of how they model these is given or whether they deal with issues like uncertainty in sensor data. Also, by tying their models to the Quake III games engine, they do not address interoperability outside this project. Finally, no additional tool support is described which may aid the SBA developers either and no evaluation of the tools to determine whether they meet the requirements of SBA developers is given.

3.4.2.3 3DSim

(Shirehjini and Klar 2005) have developed 3DSim, which is a 3D-based rapid prototyping and simulation environment that allows development of Ambient Intelligence building blocks (e.g., situation recognition, goal-based interaction). 3DSim offers open and standardized interfaces allowing (a) to integrate new devices and sensor components as well as (b) to interact with those devices and sensors, for example, to invoke actions on available devices to gather raw sensor data or to query context data managed by the environment monitoring subsystem. Therefore, Ambient Intelligent (AmI) building blocks built up on 3DSim could be easily deployed to real physical environments which support the same standards and interfaces such as the Universal Plug and Play (UPnP) standard.

At the core of 3DSim is the Collaborative Virtual Environment (CVE) which manages the 3D environment, a graphics engine called RenderWare (Criterion 2012), and a 3DSimclient which provides a user interface that visualises and supports interaction with the 3D-environment. Context events are passed to the system with a TCP based event interface. During a simulation, sensor data is triggered by humans using GUI elements, for example, an avatar can point at devices. Support for context-visualisation provided by physical sensor systems is described which allows the accuracy of context-aware systems to be tested and allows smart environment designers the ability to monitor the “awareness” of the current set up. For example, by visualising a physical smart space and then visualising the sensed location in that smart space. By overlapping the physical location with its sensed location it supports visually determining the accuracy of the received location. This relies though on the existence of a physical lab to evaluate.

Currently only a single meeting room has been simulated. This features smart projectors and shutters, and includes avatars to represent the human element of the environment. No evaluation has been conducted, for example, whether it met the requirements of designers using the system. Little information is given on how any of the simulation is modelled, what sensors are actually simulated and whether the simulations deal with uncertainty in sensor data, although they do identify the usefulness of the visualisation of such information.

3.4.2.4 eHomeSimulator

The eHomeSimulator is a simulator for realising mobility scenarios where user personalised preferences can be applied to new buildings (Armac and Retkowitz 2007). The simulator has two main purposes. Firstly, testing and evaluation and secondly, supporting demonstration of tools and example eHome services. The eHome simulated environment consists of three layers. The first layer contain locations the environment is composed of (e.g. rooms) is modelled as arrays of tiles overlaid on a 2D background of a building. This 2D background can be created in Google SketchUp and imported. Walls, furniture etc. is therefore displayed visually, and has no impact on simulation. The second layer contains devices present in the environment. These are represented as graphics for device representation. These can change colour according to current state of device. The third layer contains persons that are currently present in the environment. These are represented by avatars, which can move freely in all accessible areas, although only one avatar may be controlled at a time through a 2D interface. As such, it provides very basic interaction with the environment.

Its context simulation is also quite basic and seems to be driven by button clicks. Context values are displayed in the configuration window. The simulator is used to evaluate services, for example a service which switches devices on and off as a user moves between rooms and buildings according to their own personal prefaces. They provide a case study to demonstrate its usefulness in this regard, as a cost effective method to evaluate services based on context simulation like location and temperature. The paper does not provide a context model though and uncertainty in context data is not discussed. No discussion of using an existing standard to improve interoperability of the environment model or sensor/context models is given and no evaluation of its usability is undertaken, once again, making it hard to determine whether it was usable by SBA developers.

3.4.2.5 VPlaceSims

Choi et al. have employed a simulation platform which includes a third person virtual reality component for evaluating the interactions of users with smart homes called V-Placesims (Lertlakkhanakul, Choi et al. 2008). It is an online virtual environment platform for design collaboration between a project architect and users. This way, architects can evaluate design issues of smart homes early in the building life cycle. Choi et al. have also developed a special context-aware place data model. Unlike traditional CAD building models, this model embodies geometry information as well as semantic information regarding the relationship between occupants, objects and spaces. It includes objects such as furniture, equipment and appliances and the model is capable of spatial context awareness. Thus objects contain their own functions and status to interact with users and other objects.

Choi et al. recognise the need for a context-aware building data model for simulating smart home functions and services which together with VR, can be used to evaluate the smart homes behaviour. By developing their own building modelling approach though, they do not fit into the larger area of building information modelling which, as addressed earlier in this thesis, is a well-developed field in building design. By not examining existing BIM models the issue of interoperability is not addressed, meaning a complete building model must be created from scratch for a simulation to take place. Context simulation is handled through a spatial context model, so that contexts are determined by the activities of users. As such the sources of context, sensors, and their modelling are not addressed meaning issues like uncertainty in context are not addressed.

V-Placesims provides a tool (PlaceMaker) for modelling the environment and another tool V-PlaceLab to insert smart objects. These objects are then linked to avatars activities so that they activate when the user does certain things, for example sit down or enter a room. V-PlaceSims is intended to support the collaborative nature of smart environment design and address the need for feedback from users to designers of these environments, in this case the architect. As of yet, Choi et al. do not offer any evaluation of whether it has met the requirements of architects or SBA developers.

In conclusion, the paper provides a detailed approach to context-aware building modelling. By bringing together building modelling with VR they provide insights into the usefulness of integrating a building model into the simulation framework to handle issues like SBA development and user centred evaluation. As such, they make a valuable contribution to the area of smart building design.

3.4.2.6 Simulation Framework in Second Life

Brandherm et al. introduce a simulation framework in second life (SL) with evaluation functionality for the rapid prototyping of sensor-based systems (Prendinger, Brandherm et al. 2009). Sensor-based systems can connect to SL using a mediator which connects a proxy in SL and to an interface in the sensor based system. Virtual sensors can be created in SL and so the sensor fidelity is dependent on how these simulators are modelled. Little detail is given though on how these are modelled, with some examples of using received signal strength and distance between transmitters and receivers, which can be held by avatars, to look at issues like estimating position of tags. They then visualise the estimated position of the simulated tag as spheres in the VR environment. They discuss the possibility of modelling aspects of the environment so that they influence sensor behaviour, but this has not been implemented. The programming interface is script based, using the Linden Scripting Language, which is part of second life. They provide an example of a script which defines a virtual object with properties like position and orientation, but again, with no consideration of using standard models to improve interoperability with other systems. Consequently the approach is tied to second life, which reduces the flexibility of the system.

In conclusion, the paper presents some interesting uses of VR and visualisation of sensor data for evaluation of a sensor system. But, by not providing much in description of the models and lacking evaluation of usability of the system by SBA developers, it is hard to validate their approach. Also, the simulation framework does not incorporate a lot of features described in other context simulators, like parametric modelling, uncertainty modelling, bot driven simulation and tools for supporting the evaluation process. They do claim the model is both flexible and extensible, so these features may be included in future implementations. They also mention that they plan to implement a user-friendly interface and toolbox to support developers of ubiquitous computing systems in the future.

That concludes the state of the art review of both interactive and non-interactive context simulators. The next section will identify key properties of the simulators to inform the development of the context simulation model. In order to categorise the different simulators, the comparison framework developed by O'Neill will be employed (O'Neill 2011).

3.5 Categorising Properties of Existing Simulation Approaches

Here the comparison framework developed by O'Neill is used as a basis to categorise and analyse the different properties of the existing context simulators. This culminates in Table 3-1 for non-interactive context simulators and a second Table 3-2 which is developed for interactive context simulators (below). Using both these tables' strengths and weaknesses in current approaches shall be identified. Each category in both tables is described under its own heading, the table then gives an indication whether this is or has been met by an existing context simulator. Where a paper describes an aspect of simulation, the benefit of the doubt is given and it is assumed to be implemented. It is then given an "i" to indicate this. Some properties in the reviewed papers are discussed in terms like "can do" or "could be supported". These are indicated by a "c" to represent it as a conceptual model. Where a property is not discussed it is left blank. Before the tables are presented, each feature will now be described in greater detail.

3.5.1 Simulating Context and Uncertainty

In the previous state of the art review of context simulators, the different methods to support simulation were discussed. In this section these will be explored in more detail, and an examination of the different models used to support simulation will be explored. Where standard models are used, these will be highlighted, as they support reuse across the building life cycle.

3.5.1.1 Context Source Modelling for Simulation

As mentioned previously, a context source is any entity which provides contextual information relevant to a context-aware application (Broens and Halteren 2006). Of the context simulators reviewed above, context sources can be modelled as sensors (Fahy and Clarke 2004) or as abstract context generators (Broens and Halteren 2006). Here we cover some of the key properties that have been identified as being key to providing context simulation. These properties build on the framework developed by O'Neill with additional ones identified which are relevant to context simulation and which are taken from context simulators described here. These are –

Type: All the above context simulators define a type of a context source. For example, a Context Source may be of type temperature, indicating it provides a temperature value. Type definitions can also be useful if pre-set values for context sources are to be used.

Value: With the exception of 3D Sim, all the above context simulators define the output value of the context source. This could be a temperature, a location, or something like an activity (e.g. a meeting).

Rate: The property of rate is used in non-interactive context simulators, but is not so common in interactive context simulators. This, it is assumed, is because the rate is tied to the particular component which is providing the interaction, and these tend to have a fixed rate of producing messages tied to the games engine. The exception is the simulation framework in second life, where they have implemented a rate to control the rate of context generation. As a context source generally will not provide information in real time a rate should be defined. This rate may be due to the measurement rate, or due to the rate at which it can access the communication medium.

Placement: Placement of sensors is also common, although the reasons are often different. For V-PlaceSims placement of sensors is tied to the activities of human controlled avatars. In simulators like SENS, placement is used to relate the environment to the simulation of communication between wireless sensor nodes, so that the effects of objects like walls can be evaluated. This relationship is best expressed as its location (Reynolds, Cahill et al. 2006). Thus, the placement of context sources is essential to knowing how the environment will impact on its measurements. So too, simulated context will have a relationship to the simulated environment in which it is placed.

Variation: Context is often dynamic, i.e. changing due to the environment. This change can be influenced by many things. For example they can be influenced by the time, by the movement of occupants in the building or due to changes in temperature. It should be possible to model these variations within the context simulation model to support their simulation.

3.5.1.2 Context Simulation Approaches

With respect to modelling the variation in the value of the context generated by the context source, no interactive simulator has looked at this, basing all context variation on the movement of the human controlled avatars. For non-interactive simulators, a number of approaches have been identified for simulating changing values of context to mimic how they behave in the real world. For each of the following approaches examples are given for particular types of simulated context source. These are representative of a large number of potential context sources. For example, a method similar to temperature simulation may be used to simulate humidity, by adjusting the algorithm and parameters. These approaches are explored in greater detail here.

3.5.1.2.1 Time series and Mathematical Formula

Time series are a method for modelling variation over time. This consists of defining a period of time and a value associated with that time. For example, for time (t) generate value a, for (t2) generate another value (SimuContext). These types of models can be used together with mathematical formulae to simulate context and this has been applied by UbiREAL (Nishikawa, Yamamoto et al. 2006). They give an example of a formula they use to simulate temperature, taken from the physical sciences:

Equation 1

$$\Delta \mathrm{T}(\mathrm{t}) = \frac{\mathrm{Q}(\mathrm{t})}{\mathrm{C}}$$

Here C denotes heat capacity of the target area/zone in which the sensor is to be placed, for example a room, and t, ΔT(t) and Q(t) denote the unit of time, temperature difference and obtained heat quantity from previous evaluation, respectively. For each simulated context source, the temperature at that source is calculated periodically. Input parameters include its previous value, characteristics of the environment including room size, openings to those spaces, wall and window materials, heat of adjacent spaces, lights and electrical equipment, device states, occupancy and human behaviour, time elapsed from previous calculation (etc.).

Another example of how a mathematical formula may be used is in simulating radio propagation of an RF transmitter like in SENS (Sundresh, Kim et al. 2004). For example, there are well established formulae to describe the propagation of radio waves in free space (Barclay 2003). Barclay describes that a transmitter with power p_t in free space which radiates isotropically (uniformly in all directions) is known to give a power flux density s at distance r of

Equation 2

$$s = \frac{p_t}{4\pi r^2}$$

Such an approach can be taken to model the value of an RF receiver. Inputs and parameters to consider when modelling for simulation of values like received signal strength would be the distance between the transmitter and receiver, transmission loss, characteristics of the environment including objects that block line of sight and their materials, other objects and their reflective properties, openings and other contributors to refraction, and finally occupancy, i.e. the position of persons.

To achieve this, the model should support parametric modelling. By supporting the modelling of properties of the environment as parameters to simulation, a wide range of behaviours can be modelled, greatly improving the flexibility of the simulation process, and this has been identified as a requirement by Reynolds et al (Reynolds, Cahill et al. 2006), UbiReal (Nishikawa, Yamamoto et al. 2006), DiaSim (Jouve, Bruneau et al. 2009) and at a conceptual level in 3D Sim (Shirehjini and Klar 2005).

3.5.1.2.2 Playbacks

Another method for producing variation in context is the use of playbacks of context from logs. While this cannot be strictly considered simulation, it is included here for completeness, as it can be a quick method for generating “simulated” context. This approach takes sensor data from a sensor for a Room R for time T at a measurement interval Mi and should only be used where there is an existing environment with similar characteristics to the simulation environment for context logging. This sensor data is stored and then replayed. Examples of inputs and parameters are start time, end time, whether data is to be looped, and the type of location so that the sensor outputs can be best correlated with the simulated context source, where the simulated context source is not the same sensor as the one which measured the data. Playback is supported by SimuContext (Broens and Halteren 2006), DiaSim (Jouve, Bruneau et al. 2009) and the second life simulator (Prendinger, Brandherm et al. 2009). This feature also supports examination of context generated purely by simulation and not by playback with real context, as simulated context can be examined with overlays of real context to determine correctness, with appropriate analysis tools.

3.5.1.2.3 Statistical Models Based on Physical Readings

Another approach which is not explored in any of the above examples, but is added here for completeness, is the possibility of using models based on data mining of context generated by physical sensors for simulating other values, like signal strength between an RF transmitter and receiver, readings can be taken for location L for Room R for time T at a measurement interval Mi, and this is an approach explored by R. Sebastian (Sebastian 2005). This is not included in the above review, as it is does not aim to evaluate applications which use context. We include it here though, for completeness. Using this approach, readings are taken for a transmitter placed covering a grid of points with spacing distance between each S. At each point in the grid, a simulated output is based on the mean reading for that grid point. Orientation can also be included where the transmitter is dipole. Where the orientation of the transmitter is unknown, for example, it is mobile and in the pocket of a mobile avatar, it may be possible to model the potential variation due to changing orientation as the mean of the measured values at different orientations and the overall variation as a PDF. While this approach is similar to using playbacks, it can result in more variation in the simulation output still with grounding in how context is generated in a physical environment.

3.5.1.2.4 External Simulation Models

Another method to drive simulation is the integration of outputs from other simulation tools. While not discussed explicitly in the previous context simulators for context simulation, other than with respect to network simulation, existing simulation tools can provide input for simulating context. For example, it should be possible to integrate a computational fluid dynamic model for a building to simulate temperature sensor readings, or radio wave propagation to simulate received signal strength (which will be explored in greater detail in Chapter 4 and Chapter 5).

3.5.1.2.5 Grid Based Simulation

Some discussion has already been given regarding the placement of context sources and the effect of the environment. Grid based simulation approaches provide a method for defining characteristics of the environment which effect the simulation. While a number of simulators take this grid based approach, SENS (Sundresh, Kim et al. 2004), UbiREAL (Nishikawa, Yamamoto et al. 2006), Reynolds et al. (Reynolds, Cahill et al. 2006) and eHomeSimulator (Armac and Retkowitz 2007), a zone based approach may be more suitable, as zones with their own geometry provide a more flexible approach to defining areas and layering simulation effects.

3.5.1.2.6 Hybrid Simulation

Another feature of context simulation which is supported by a number of context simulators is hybrid simulations. These are GLS, SimuContext, UbiBuilding, DiaSim, QuakeSim, UbiWise and 3DSim. Hybrid simulators can support fusion of real and simulated context, for example, simulated location with physical sensors to monitor an actual person's body, thus supporting the evaluation of applications with context sources which may not be available. Hybrid simulation also supports testing the fidelity of context simulation as comparisons can be made between a physical context source and simulated equivalent in real time. This is similar to the method used when playbacks are applied and where the appropriate analysis tools, like visualisation of context, are available.

3.5.1.2.7 Network Simulation

Another feature which could be supported is network simulation. Wireless network communication, often using novel, ad hoc protocols, is a frequent component of wireless sensor network research. Integrating the simulation of network operation and performance in such cases to the impact of the resulting context information on an application could offer a beneficial use case. Network simulators can range from simply providing communication when two devices are in proximity with the simulator (Nishikawa, Yamamoto et al. 2006), to integration of external network simulators

(Reynolds et al.) such ns-2 (Fall and Varadhan 2007) or the proposed ns-3 (Henderson, Lacage et al. 2008).

3.5.1.2.8 Uncertainty Simulation

Another property, which this thesis sets out to develop and evaluate, is the issue of uncertainty, which was addressed previously. This is a key requirement, because as stated previously, assuming highly accurate context will result in applications which behave in an unreliable manner in the face of uncertainty in context data. Therefore, the simulator at the least should provide some method for introducing uncertainty or error into both the simulated values and the simulated rate, which can have a relationship to the communication network. Both the Generic Location Simulator and Generic Simulation Tool have identified this requirement at a conceptual level (Sanmugalingam and Coulouris 2002; Reynolds, Cahill et al. 2006). Both recommend that the use of Gaussian probability distributions to be used to introduce error into simulated context. Uncertainty in context information can also be modelled using a parametric approach, so that error distributions can be also affected by the different parameters of the environment, like position of entities, and materials of the walls, to name but a few, and SENS has specifically modelled these types of effects with its grid based model for modelling error (Sundresh, Kim et al. 2004).

3.5.1.3 Interactive User Drive Context Simulation

As has been identified previously, interactivity is an important aspect when evaluating SBAs. User driven context takes as an input data the position of user controlled avatars in interactive simulation environments like QuakeSim and Second Life. This data can be used to simulate a number of different types of location based context. For example, real time coordinate-location, received signal strength of RF transmitters, and presence events, like those generated by passive infra-red sensors and pressure mats. In addition, this data can be used as an input parameter to simulations of ambient phenomena such as temperature, CO2 levels and humidity, as occupancy has an affect on these. V-PlaceSims also supports modelling of interactions between users and the physical environment. These capture different actions and the impact on user sensing associated with activities, such as moving around a building, moving at different speeds, opening doors, or sitting down. Due to the importance of understanding user interaction with smart environments and SBAs to achieve good designs that are acceptable to users, this thesis will focus on this type of user driven context simulation. Of all the aforementioned interactive approaches to context simulation, only one addresses uncertainty. The second life simulator looks at uncertainty caused during the process of using two transmitters to determine location. Here uncertainty exists in the form of potential positions that the receiver could occupy. They present this estimated position visually to improve understanding of this for developers.

3.5.1.4 Agent and Activity Models for Bot Driven Context Simulation

In non-interactive context simulation, the simulation of people moving around the environment which is tied to the generation of location tracking context, is performed using different forms of preprogramed agents referred to as bots. This offers the potential advantage of close control of simulated user behaviour and good repeatability of that behaviour. A number of the aforementioned approaches look at this type of simulation, GLS, Cass, UbiREAL, Generic Simulation Tool and DiaSim. These usually rely on modelling of agents or bots by setting waypoints (DiaSim, UbiReal). This can also be achieved using games engines (QuakeSim), as they generally support the creation of bots which can behave independently of human control. These are usually implemented to exhibit some autonomous behaviour, e.g. reacting to the position or behaviour of other characters and including some random behaviour, for example to provide challenging non-player characters opponents inin first person shooter games. Therefore, carefully designed way points must be used to override this unpredictable behaviour when configuring such games platforms for context simulation purposes (O'Neill 2011)

In the domain of building performance simulation, a number of researchers have looked at developing user behaviour models. These have ranged from activity models to predict lighting energy performance (Reinhart 2004), to user interactions with windows and its effect on thermal comfort and energy use (Rijal, Tuohy et al. 2007), to larger sets of interactions including, in addition to aforementioned windows and lights, activation of heaters and fans (Nicol 2001) and additional activities like going to lunch, getting a drink, boiling the kettle, having a smoke (Tabak and De Vries 2010). These studies have mainly focussed on office spaces. This is due to the imperative to improve operational, safety and energy efficiency in office buildings in use. Office use cases also offer similarity between the activities of one office user to another (e.g. working at desk, going to meetings, etc.) and the predictability of work times for the majority of office users (e.g. a nine to five work day). These factors may make the task of modelling activities more fruitful and tractable than modelling activities for buildings which are less numerous and also have less predictable use (e.g. a university campus) and it has been claimed that building predictable usage is a pre-requisite for accurate predictions in simulation (Degelman 1999).

While this is the case for skeleton activities, when modelling intermediate activities the type of building has less impact on the predictability of the activity (Tabak and De Vries 2010). Intermediate activities include actions like “going to the toilet” or “having a drink”, while skeleton activities are directly linked to the role of the person, like giving a lecture or cleaning the toilet. These types of activities can be modelled using either probabilistic or S-curve methods (Tabak and De Vries 2010). As the focus of this thesis is not user activity modelling for simulation, but rather user driven context simulation, these methods will not be explored here in any more detail and are included so that any developed solution will also be flexible enough to integrate this type of modelling if required.

3.5.1.5 World, Environment and Building Models

The majority of the context simulators reviewed here only consider the environment to visualise the environment or building. These range from simple 2D representations (Cass, ubiBuilding, eHomesimulator) to more complex 3D interactive models (UbiREAL, QuakeSim, 3DSim, Second Life

Simulator). These models have no effect on the context simulation process itself, although they can be useful for placing them in the environment (Cass, eHomeSimulator, UbiReal).

It has been discussed previously, that the placement of a context source can result in variations in the value generated. Therefore, in order to accurately model simulated context sources, it is necessary to be able to take data on the environment as parameters into the simulation. A number of the aforementioned approaches look at using models of the environment to do this, from modelling walls, and areas like concrete ground and grass (SENS). The Context Simulation Tool and DiaSim both set out to support additional modelling of objects in the environment which may affect simulated context values. Most models though are discussed at a high level with few examples given with the exception of V-PlaceSims who provide a detailed description of a context building model. This model does not address simulation of low level context and does not address uncertainty modelling for simulated context sources.

3.5.1.6 Standard Models

In the background chapter, the benefit of using standard models was discussed. Standard models are used for describing buildings and sensors in related fields in the building life cycle, meaning that once a 3D models of a building, for example to represent it visually, is modelled, it need not be modelled again by other disciplines involved in other areas of the building life cycle. The only paper to explore the use of a standard model for modelling a smart environment is UbiREAL which uses VRML. By taking this approach, the option to import an existing 3D model can be chosen where an appropriate model exists. The downside of VRML is that it only captures how to visually represent a 3D environment and does not contain any information regarding the properties of the entities they set out to visually represent, for instance, the material of a wall or the thickness of a window. BIM on the other hand supports this type of modelling and the IFC standard support modelling properties of building elements, like their width, height, weight etc. These models can therefore be used to support parametric simulations.

With respect to context source models, none of the above approaches use a standard model to describe context sources such as sensors. SensorML provides a model which can capture all of the features identified in the above context source models. IFC does not support such modelling. An approach which integrates sensorML concepts with IFC would therefore provide benefits as part of integrating any approach into the wider building life cycle.

3.5.1.7 Modelling and Configuration Tools

Ideally, an easy-to-use tool should aid SBA developers in the process of modelling and configuring the simulation environment. Cass, SimuContext, UbiREAL, the Generic Simulation Tool, ubiBuilding and DiaSim, all provide tools for creating context sources, some of which support their placement or measurement rate configuration. Cass and DiaSim support modelling of different types of context simulation such as the use of time series to generate context value. Only the more recent interactive context simulators support configuration of context sources (eHomeSimulator, V-PlaceSims and Simulation Framework in Second Life), all of which support placement and with V-PlaceSims and the Simulation Framework in Second Life both support the configuration of transmitters used in sensing. Only one of the context simulators (SimuContext) specifically looks at modelling aspects related to uncertainty by supporting configuration of quality of context.

While tools for modelling the building are useful and supported by all but one of the interactive simulators, and one of the non-interactive (Jouve, Bruneau et al. 2009), the focus here is on the context source modelling and so these tools are not explored in detail. The approach in this thesis is to rely on external tools for the modelling of the building itself so that existing models can be leveraged. Also, some context simulators support modelling of devices and services, but again the focus of this thesis is on context sources to evaluate devices and services which are being developed external to the simulator.

3.5.1.8 Testing

Building on from O'Neill's comparison framework, the testing process itself is examined here. Some key features of a testing environment are that it supports repeatable experimentation. Simulators allow certain aspects of the environment to be configured to produce the same results with each simulation. Non-deterministic elements can be introduced using stochastic modelling of aspects of context simulation. This should be done in a repeatable manner so that the range of variation of the stochastic process is known and configurable. Interactive simulators also provide non-deterministic simulation due to the behaviour of the individual controlling the avatar. This should also be supported so that user centric evaluation of applications can take place.

Scalability should also be a feature of SBA evaluation, allowing for buildings with multiple floors, rooms, sensors, devices and occupants. This thesis does not set out to evaluate scalability, but a number of the existing simulators have looked at performance and scalability issues, specifically GLS, SENS, UbiReal.

3.5.1.9 Visualisation and Analysis Tools

Visualisation of the environment so that the SBA evaluator can see different situations and their associated context dynamics arise in situ, is preferable to a black box scenario where the internal dynamics of the system are unknown by any other means than the outputs of the simulator. Support for SBA evaluators in examining and analysing the SBA's context driven behaviour and the characteristics of the context information that have resulted in this behaviour is therefore a key requirement. A major issue for context aware SBA developers is understanding the role of uncertainty in context information on the range of possible SBA behaviour. This is a complex problem and tools which can improve intelligibility of SBA and context processing behaviour for SBA developers should be a part of the context simulation model.

One existing tool which supports visualisation of sensor data is Cass. Cass provides simulated sensor data overlaid on the building representations. Cass also provides streamed events with a time line to visualise when a context event had occurred. 3DSim also discusses the use of visualisation of objects in the VR environment based on the location generated by physical sensors in the equivalent physical environment to evaluate the accuracy of the location. However, 3DSim does not examine the use of simulated context to evaluate issues like accuracy. SimuContext shows simple textual outputs of the sensor value to support analysis. GLS provides a tool for visualising the movement of bots as graphs. Brandherm et al. also give some visual cues in the form of spheres inserted into the VR environment which can be used to highlight the issue of triangulation when using received signal strength between transmitters. DiaSim and the eHomeSimulator also support visualisation of location on a 2D monitor, but none of these incorporate analysis of uncertainty.

Due to the amount of data the evaluation may generate in larger simulated building environments, it may be necessary to replay a simulation run so that it can be examined repeatedly and a number of simulators support this, specifically GLS, Cass, SimuContext, UbiREAL, DiaSim.

3.5.1.10 Deployment Environment

This category refers to the simulated deployment environment used in the design process and is covered to a large degree by the existing work by O'Neill, though some additional categories have been integrated. For example, heterogeneity of context sources now has four sub categories. These are, whether the context simulator supports heterogeneity in low level context such as sensor outputs or higher level context like activities and whether it supports heterogeneity in location (proximity, coordinate, presence events) and heterogeneity in other sources, such as temperature, humidity, etc. The non-interactive context simulators often discuss location context, but do not go into specifics about the types of location context they support simulating, given only one example if any. For example, the Generic Location Simulator appears to support coordinate locations of mobile entities ('locatables'), but while it discusses a sensor model it does not go into any details about how it might use those coordinates to generate different types of location context. SimuContext discusses location and speed, which are both related to location, but not much detail is given on how these are simulated. Only DiaSim discusses in any depth the different types of location context it can support simulating, giving examples of motion detection (which would indicate presence), coordinate location and also higher level location contexts, for example “room”. DiaSim also supports generating other types of low level and higher level context, like temperature and also contexts like “fire”. The Generic Simulation Tool supports simulating types of context in the form of temperature, light levels or noise. UbiBuilding discusses simulating sound, force, ambient lighting and higher level contexts like activities. UbiREAL supports a range of low level contexts like temperature, light levels and humidity, and so does Cass. SENS simulates radio and sound wave propagation, and so simulated received signal strengths and sound are generated, which are referred to here as low level contexts.

QuakeSim and UbiWise both support simulation of presence and coordinates, as well as higher level contexts like rooms. QuakSim also indicates the possibility of simulating thermal sensors, but does not go into any details. 3DSim supports simulation of location, but does not indicate what types. It discusses also that the avatars can perform activities, like sitting or writing, although the work does not explicitly specify how this might be used to generate context. It also supports light intensity and colour of light as well as device states. The eHomeSimulator supports temperature, humidity and illumination sensors and also sensors related to presence (person detectors). V-PlaceSims supports context in the form of locations and activities. Examples of these activities are “sitting down” and “on sofa”. The Simulation Framework in Second Life also supports location contexts in the form of coordinates and also proximity using RFID tags.

Also included here are whether there is heterogeneity in devices and bots (e.g. types of bots) and buildings (e.g. homes, offices). Scalability looks at whether the scale of buildings, context sources, devices and/or bots is addressed. This thesis does not set out to address these areas and they are included for completeness

3.5.1.11 Simulation Evaluation

This category simply indicates what aspects of existing simulators have been evaluated. As can be seen the majority are case studies, with performance evaluation a close second. SENS has conducted some evaluation of the fidelity of its simulation to the real world, i.e. how closely does the simulated context match context generated by an equivalent physical context source in an equivalent environment. Usability studies have not been undertaken.

3.5.1.12 Example Models Provided

This category is included to indicate where formal descriptions of models are provided and whether they are based on any existing standard.

3.5.2 Comparison Framework

Table 3-1 presents a review of non-interactive context simulators to support comparison of the features identified above and which are relevant to this thesis. Table 3-2 gives a similar table for interactive context simulators. Two additional fields have been added to Table 3-2 to cover the property of user driven context simulation, through interaction between users and the environment in real time and also whether the context simulation is dependent on the visualisation engine.

Table 3-1 Comparison Framework – Non-Interactive Simulators

Non-Interactive Context Simulators		2002	2004	2004	2006	2006	2006	2007	2011	2011
		GLS	Cass	SENS	SimuContex t	UbiREAL	Generic Simulation Tool	ubiBuilding	DiaSuite, DiaSim	InSitu
Context Simulation	Clocking Mechanism	i	i	i		i	i		i	i
	Extensible Modular/Layered Architecture	i	i	i	i	i	i		i	
	Grid Based			i		i	i			
	Play Backs				i				i	
	Statistical Model	i							i	
	Mathematical Formula	i		i	?				i	

	Parametric Simulation							i	i		i	
	Bot Driven Simulation			i	i			i	i		i	i
	Uncertainty Simulation	Communication		c		i	i	c			c	
		Value (Precision)		c		i	i		i		c	i
	Hybrid Simulation			i			i			i	i	
	Network Simulation					i		i	i			
	Integrate Models from Other Simulation Tools			c		?			i		c	
Simulated Context Sources Modelling	Type			i	i	i	i	i	i	i	i	i
	Value			i	i	i	i	i	i	i	i	i
	Rate			i	i	i	i	i	i		c	i
	Placement			i	i	i		i		i	i	i
	Variation	Time Series/Mathematical Formula					i	i			i	
		Probability Distribution		c		i	i		c			
		Uncertainty		c		i	i		c			
	Sensor Specific Formats			i		i		i		i		i
Building Model	Walls					i			i		i	i
	Zones/Areas					i			i		i	i
	Others								i		c	i
	Visual Model			i	i			i		i		i
Deployment Environment	Scalability	Building		e				i			i	i
		Context Sources		e		i	i	i	i	i	i	i
		Devices						i	i	i	i	i
		Bots		e				i	i		i	i
	Heterogeneity	Building		c							i	i
		Context Sources	Low Level	i	i	i	i	i	i	i	i	i
			High Level		i				i		i	
			Location				i				i	i
			Other	c	i	i		i	i	i	i	
		Devices			i			i	i	i	i	
		Bots			i							
Example Models	(Generated) Context Model					m				m		
	Context Source/Sensor Model				m		m					
	World/Building Geometry Model							m				
	Environment/Error Model											
	Agent/Bot Model											
	Device/Service Model							m				
Configuration Tools	Building	Walls									i	
		Zones/Areas									i	
		Others										
	Context Sources	Type			i		i	i		i	i	i
		Rate			i		i				i	i
		Placement			i			i		i	i	i

		Transmitters/Receivers		i						i	
		Time Series		i						i	
		Uncertainty				i					
	Devices			i			i				
	Bots			i			i				
Testing	Repeatability		i	i	i	i	i	i	i	i	
	Scalability		i		i	i	i	i	i	i	
	Environment Visualisation	2D	i	i	i		i	i	i	i	i
		3D					i				
	Replay Simulation		i	i		i	i			i	
Analysis Tools	Context Monitoring	Real Time Visualisation		i			i			i	
		Graph	i		i						
		Text		i		i					i
		Streamed Events		i					i	i	
	Logging							i		i	
Simulator Evaluation	Performance		e		e		e				e
	Usability										e
	Realism/Fidelity		c		p						
	Case Study			e				e	e	e	e
Legend											
Not provided/discussed											
Conceptual		c									
Modelled (based on standard model)		m(s)									
Implemented		i									
Evaluated (partially)		e(p)									

Table 3-2 Comparison Framework – Interactive Simulators

Interactive Context Simulators		2001	2002	2005	2007	2008	2009
		QuakeSim	UbiWise	3DSim	eHomeSimulator	V-PlaceSims,	Simulation Framework in Second Life
Context Simulation	Clocking Mechanism	i	i	i		i	i
	Extensible Modular/Layered Architecture	i	i	i	i	i	i
	Grid Based				i		
	Play Backs						i
	Statistical Model						

	Mathematical Formula								i
	Parametric Simulation					c			
	Bot driven simulation							i	
	*User Driven Simulation			i	i	i	i	i	i
	*Simulated within VR Engine			i	i	i		i	i
	Uncertainty Simulation	Communication							
		Value (Precision)							i
	Hybrid Simulation			i	i	i			
	Network Simulation								
	Integrate Models from Other Simulation Tools								
Simulated Context Sources Modelling	Type			i	i	i	i	i	i
	Value			i	i		i	i	i
	Rate								i
	Placement					i	i	i	i
	Variation	Time Series/Mathematical Formula							
		Probability Distribution							
		Uncertainty							
	Sensor Specific Formats								
Building Model	Walls			i	i	i		i	i
	Zones/Areas					i	i	i	
	Others			i	i	i		i	i
	Purely a Visual Model			i	i	i	i		i
Deployment Environment	Scalability	Building		i	i		i	i	i
		Context Sources		i	i		i	i	c
		Devices					i	i	
		Bots						i	
	Heterogeneity	Building		i	i		i		i
		Simulated Context Sources	Low Level	i	i		i		i
			High Level	i	i	i		i	
			Location	i	i	i	i	i	i
			Other			i	i		
		Devices			i	i	i	i	
		Bots							
Example Models Provided	(Generated) Context Model							m	
	Context Source/Sensor Model					s		m	
	World/Building Geometry Model							m	
	Environment/Error Model								
	Agent/Bot Model							m	
	Device/Service Model					s	i	m	
Configuration Tools	Building	Walls		i	i			i	i
		Zones/Areas					i	i	
		Other (windows, etc.)		i	i			i	i
	Context Sources	Type					i	i	i
		Rate							i
		Placement					i	i	i
		Transmitters/Receivers						i	i
		Time Series							

		Uncertainty						
	Devices			i		i	i	
	Bots		c	c			i	
Testing	Repeatability				i			
	Scalability		i	i				
	Interactive Environment	2D				i		
		3D	i	i	i		i	i
Analysis Tools	Context Monitoring	Real Time Visualisation			i			i
		Graph						
		Text				i	i	
		Streamed Events						
	Logging							
Simulator Evaluation	Performance							
	Usability							
	Realism/Fidelity							
	Case Study		e	e		e	e	e
Legend								
Not provided/discussed								
Conceptual		c						
Modelled (based on standard model)		m(s)						
Implemented		i						
Evaluated (partially)		e(p)						

3.6 Summary of Key Findings of Context Simulator

This section gives a summary of the key findings with respect to the context simulators reviewed and which are highlighted in both Table 3-1 and Table 3-2.

3.6.1 Strengths of Existing Context Simulators

Context simulators must meet the various requirements of context-aware application developers. A key requirement therefore is that they be flexible in their approach to simulating smart environments. *Flexibility* is demonstrated in a number of areas in the different context simulators. For example, DiaSim is flexible in the number of context simulation methods it supports, such as mathematical formula, playbacks, parametric simulation and bot driven simulation. The second life simulator also supports mathematical formula and playbacks, but does not address bot simulations and only supports interactive context generation (Table 3-2). DiaSim also demonstrates flexibility in the number of heterogeneous context sources which it can simulate, simulating location context and other contexts, like temperature (Table 3-1). Another area of importance is flexibility in the modelling of the environment, not only for visual representation but also as part of a parametric approach to simulating context. The Generic Simulation Tool, DiaSim and V-PlaceSims all demonstrate flexibility in their representation of the environment (Table 3-1 and Table 3-2).

Extensibility, which is related to flexibility, is a further key requirement for context simulators and the ability to extend existing models to account for new and novel technologies is another required feature of a context simulation model. Extensibility is harder to determine as it requires knowledge of capabilities to model unknowns. As a result, we do not make any claims about the extensibility of the existing approaches.

Tools which support the complex task of modelling and configuring the simulation environment are likewise of importance as SBA developers may need to evaluate their SBAs with context generated by systems in which they are not expert. These include tools for modelling simulated context sources to tools for modelling the environment to support both interactive testing and to support parametric simulations. Both Cass and DiaSim support configuring the largest range of properties of simulated context sources (Table 3-1). DiaSim also supports modelling aspect of the environment such as walls and areas. V-PlaceSims and Second Life simulator similarly provide tools to support some configuration of context sources as well as modelling of the environment (Table 3-2).

Analysis tools are an important part of making the evaluation understandable and effective. Several approaches provide feedback in different forms, either using real time text (Cass, SimuContext, eHomeSimulator, V-PlaceSims) or visualisation of context events (Cass, UbiREAL, DiaSim, 3DSim, Second Life Simulator), graph based analysis (GLS, SENS) or streamed events (Cass, UbiBuilding, DiaSim).

Scalability is also an important aspect of context simulation. SBA evaluation may require simulation of environments with many floors, rooms, devices, bots and/or user controlled avatars all interacting. With the exception of Cass, all the non-interactive context simulators set out to address this, although only three (GLS, SENS, UbiREAL) evaluate the performance. With respect to interactive context simulators, only QuakeSim and UbiWise aim to be scalable, and again, no evaluations are undertaken. The issue of scalability is not addressed in this thesis, although the proposed solution will set out to be scalable.

Interactivity is a key requirement when conducting user centred design of context-aware applications. This is the advantage of the interactive context simulators reviewed.

Interoperability is demonstrated by UbiREALs use of the VRML standard which supports the import of existing models. This can reduce the time to model interactive environments, which can require a considerable investment of time a single building with several floors and room, as it can take a modeller up to twenty working days (O'Neill, Lewis et al. 2006).

3.6.2 Weaknesses of Existing Context Simulators

As can been seen from both tables a key area which current context simulators are lacking is in the modelling of uncertainty. In the non-interactive context simulators it is only mentioned on a conceptual basis, with the exception of SimuContext and SENS, although no evaluation is done to determine whether it is effective during evaluation of context-aware applications (although this is not a requirement of SENS). Only the Second Life Simulator interactive context simulator discusses simulating uncertainty in context, but does not specify how this would be done rather relying on the SBA developer develop the simulation. Also of particular note is the absence of any significant evaluation of the level of usability of the context simulators for application developers, although the need for the context simulators to be usable has been raised either explicitly (Reynolds, Cahill et al. 2006) or through the indication that tools for this purpose are required (Prendinger, Brandherm et al. 2009). InSitu does provide a usability study, but as it addresses the definitions of rules for identifying unwanted behaviours. This is not directly related to the issues of creating, placing and configuring simulated context sources as part of SBA design and evaluation.

Analysis tools to support evaluation of the issue of uncertainty are another area where existing context simulators are lacking. A number of tools provide visualisation of context, for example, as the location of objects (Cass, 3DSim), as streamed events (Cass, UbiBuilding), or as textual outputs (SimuContext). None of these incorporate analysis of uncertainty or tools to improve intelligibility for developers to enable them deal more effectively with uncertainty. Visualisation tools to address this issue could provide a method for rapidly evaluating the effects of uncertainty on applications. These tools must also be evaluated to determine how effective they are when evaluating applications, and as yet, no context simulator has looked at this.

Another weakness identified in a number of approaches is that they tie the context simulation to particular VR simulators. By not basing their models on any existing standard, this makes migration of the simulation framework difficult. One solution to this is to decouple the simulation framework from the VR engine. Standard models for describing the building and sensors, sufficiently to enable context simulation, will also support importing existing BIM's into the simulation framework. This makes SBA development an integrated part of smart building development and has the potential to provide benefits by reducing modelling time and support the simulation and evaluation of smart building applications within the context of the whole building design through to operation.

A further weakness of the interactive simulators is the lack of additional context simulation support. Existing approaches look at simulating location context and do not set out how they would simulate other values. While this is valid and plays to the strengths of the VR engines, by not considering how additional context may be simulated, it reduces the potential of the simulator. One solution is to integrate properties of non-interactive context simulators with the context simulation model. The resulting context simulator is therefore more flexible. Together with the decoupled approach this means a more flexible context simulator can be developed which can use both bots which are independent of a VR environment and also user driven context simulation using an interactive VR environment.

3.7 Chapter Summary and Conclusion

This chapter commenced by examining the requirements of SBA developers and identifying the use of simulation as a method for supporting the evaluation of SBAs. Following from this, it presented a set of criteria to guide the analysis of existing context simulators through first examining non-interactive context simulators and next interactive context simulators including respectively, those which do and do not support user interaction with the building environment. These were then reviewed against the criteria established. From this review a list of features were identified and described which the different context simulators support. The result of this is presented in two tables which identify the different features and the simulators which support them. Finally, a summary was presented of key findings, identifying the strengths and also the limitations in the current state of the art which this thesis addresses, which are used to establish requirements for the advancement in the context simulation state of the art undertaken in this thesis

These limitations demonstrate that with respect to current context simulators, there is a lack of design and evaluation of the usability of these systems. This has been identified as a key requirement, such that the level of usability for SBA developers should be appropriate and suitable for achieving the task of evaluating the SBA when faced with varying levels of uncertainty in context. Usability must be considered during the process of modelling and configuring the simulation, and as such, methods which reduce the complexity of the task of creating, placing and configuring simulated context source with additional uncertainty, while still supporting the effective and efficient evaluation of the SBA are required. The different steps in the SBA evaluation process must also be addressed. The review identified a lack of tools to aid with the analysis of the uncertainty on the SBA. In chapter 2, the visualisation of uncertainty has been shown to improve user understanding of uncertainty for users at the cost of reducing their efficiency using the application. A similar approach to visualising uncertainty could therefore improve the understanding for SBA developers who themselves may not be experts in context acquisition.

The usability of the model must be considered whilst still supporting the evaluation of a potentially wide range of different types of SBA which includes range of scenarios, buildings and sensors. The strength of many of the context simulators is their flexibility and extensibility, and also that they have the potential to handle varying scales of simulated environments. Flexibility is important therefore in maintaining range of contexts that can be simulated, and as such the context simulators each address simulating different types of context. In addition, the methods by which context is simulated varies from context simulator to context simulator, with non-interactive simulators generally supporting a wider range of methods to generating simulated context. Consequently, the proposed approach should also be flexible in the methods it supports for simulating context.

Finally, as the modelling of the built environment is a complex process in itself, methods which support importing existing standard models for buildings can reduce the burden on the SBA developers where existing building models have been developed. By integrating the context simulation model into standard models for buildings, the potential to import existing models developed at earlier stages of the building life cycle can be supported. Building information models such as Industry Foundation Classes also support the modelling of aspects of the building which may affect the simulation of context and so may support automation of the process of introducing parameters into the context simulation process. These criteria are therefore used to guide requirements for a model that addresses the research objective in the next chapter.

4 Design

4.1 Introduction

This chapter addresses objective 2 of this thesis which is the design of a context simulation model to support the usable creation and configuration of simulated context sources that generate simulated context from user interactions and which support usable evaluation of SBA's when faced with varying levels of uncertainty in the context information used by the SBA. The model must also be also be interoperable with existing building information standards to support leverage, and integration with, other tools involved in the building life cycle and the models they produce. This objective is described in Chapter 1 section 1.3.2. The design of the context simulation model is motivated by the state of the art review of existing context simulators as documented in Chapter 3. This chapter therefore begins by exploring the influences of the state of the art on the design requirements of the context simulation model, namely that the model be flexible, usable and interoperable. These requirements motivate the need for integrating the context simulation model with building information modelling and as such a process for documenting the development of the model, called the Information Delivery Manual (IDM) and taken from the building information modelling (BIM) domain, is used, in conjunction with Unified Modelling Language (UML) techniques.

Next, a high level view of where the process of rapid SBA prototyping and evaluation fits into the building life cycle is presented and this narrows the scope of the applied approach, highlighting the specific tasks required for the developers of SBAs when evaluating their application when faced with varying levels of uncertainty in context, which they wish to address using a context simulator. The result of this process model is an overview a notional context simulation framework that satisfies these requirements and the role the SimCon model plays in that framework. The SimCon model defines the conceptual design that specifically addresses the requirements raised by the thesis question, such that its design and implementation in a toolset is therefore the subject of the evaluation this thesis. This chapter therefore presents the SimCon model and a detailed description of the different components that make up that model, as well as conceptual similarities with existing standards. Finally a conclusion and summary are given of the proposed design.

4.2 Influence from State of the Art

Chapter 3 summarised the key findings with respect to Context Simulators. It identified strengths and weaknesses of current context simulators. These form the basis of the following design requirements of the proposed context simulation model:

To enable SBA developers evaluate their SBA, a context simulation model must support simulation of a range of scenarios. This may require that the SBA be evaluated in different building types, each having different sensors, context middleware, and actuators. Therefore flexibility is a key requirement for the model, as identified by Reynolds et al. when highlighting the requirements for a ubiquitous computing simulation and emulation environment (Reynolds, Cahill et al. 2006). The context simulation model should therefore be able to handle a range of heterogeneous context source types and contexts. Flexibility should also be supported in the approach to context simulation, as the state of the art review has shown that a number of such approaches exist, ranging from simple playbacks to more complex parametric simulations, as well as bot driven context simulation and user driven context simulation. The model should therefore be flexible enough to handle these different types of context simulation. Associated with flexibility is extensibility: where flexibility is an expression of a model's capability to handle different ranges of existing types, for example context sources, extensibility is an expression of a model's capability to deal with new novel requirements, for example, a type of simulation not already supported by existing context simulators. Extensibility can be evaluated by examining the effort involved to incorporate new requirements into the existing model.

The state of the art review also identified the lack of techniques for modelling uncertainty in both non-interactive (bot driven) and interactive (user driven) context simulators which is required for conducting evaluation of SBAs when faced with context with varying levels of uncertainty.

Interactivity is also required for conducting user-centric designs of SBAs whose behaviour has a relationship to building users and the context they generate (e.g. location). This leads to the first key design requirements of the context simulation model:

Requirement R1: The context simulation model should be flexible so as to support different context simulation approaches to simulating heterogeneous context source types which generate heterogeneous types of context and extensible so as to support context simulation with additional uncertainty.

Secondly, the literature review identified a lack of evaluation of the usability of the existing context simulators. As the purpose of context simulators is to support SBA developers to develop and evaluate context-aware applications, usability has been identified as a key requirement, as highlighted in the state of the art, for example by Reynolds et al. (Reynolds, Cahill et al. 2006). In Chapter 2, usability was described as the efficiency, effectiveness and satisfaction of the proposed model.

Efficiency is determined by measuring the time it takes to complete tasks. The design should therefore aim to minimise the time taken to complete the task of creating, placing and configuring simulated context sources. Effectiveness is determined by measuring whether users can achieve the tasks. Designing SBAs which must be capable of dealing with varying levels of uncertainty is a difficult task, especially for SBA developers who are not experts with sensors. Therefore the context simulation model should not only support modelling and configuring uncertainty, it should also improve intelligibility of the effects of uncertainty on SBA behaviour so that the SBA developers are in a better position to design their application to deal with these effects. Chapter 3 identified some existing context simulators which provide tools to support analysis, but none of these have been evaluated from a user’s perspective. ISO-9241 standard defines satisfaction with the product as “comfort and relevance of application”. Satisfaction is therefore a measure of whether users found the model usable and whether they subjectively “enjoyed” the experience. The proposed design should set out to create a satisfying experience for users while also meeting the requirements of efficiency and effectiveness. This leads to the following second key requirement:

Requirement R2: The context simulation model should be usable by SBA developers to achieve the goal of evaluating SBAs which must deal with varying levels of uncertainty in context. This requires that the model be both usable during configuration and also during the evaluation process by providing tools to support SBA analysis.

A final weakness identified by the literature review was the lack of use of standard models in existing context simulators. Basing the context simulation model on standards provides several benefits. Firstly, it reduces the burden of creating new models where there are existing models, for example, in the case of an existing visual representation of a building. Where such a model can provide suitable visualisation to support interaction with the simulator and analysis of its outputs, this will considerably reduce the amount of time required to create a simulation of a building. The same is true for buildings which have models of the sensor systems. Where sufficiently modelled, these can be imported to support context simulation, without the need for additional mappings between concepts. Models developed by the SBA developer can also be integrated back into the building life cycle, reducing the burden of modelling in other related domains, for example, during operation. By integrating the building model with the context simulation model, the possibility for more complex simulations is also possible, for example, where parameters such as wall materials are used as inputs into the simulation of context which is affected by these properties. Finally, by using concepts which are shared within the larger building industry, dissemination of the proposed context model can be better supported, as each domain within the building life cycle uses a standard schema for shared concepts. This leads to the third key requirement:

Requirement R3: The context simulation model should be interoperable with existing standards in building and sensor modelling.

As interoperability with existing standards is a requirement of the design of the simulated context model for evaluating SBAs, the next section will examine a standardised approach to capturing information requirements used in the area of building information modelling, before the process of SBA evaluation issuing context simulation is presented.

4.3 The Information Delivery Manual

In Chapter 2 the Industry Foundation Classes was identified as a typical candidate for Building Information Modelling (BIM) as it is an ISO standard. In practice, IFC has yet to make a significant impact in the Architecture, Engineering and Construction (AEC) communities (Eastman, Jeong et al. 2010). This is in part due to the manner in which different vendors implement the IFC model to their specific requirements, which has often resulted in data exchanges between tools resulting in imprecise or lost data (Pazlar and Turk 2008). To remedy this situation, the National Building Information Model Standard Committee (States 2012) has developed the Information Delivery Manual (IDM). An IDM documents the processes involved to complete a specific use case. A use case defines tasks (called activities) and data exchanges required to complete the use case as well as where the use case fits in relation to the Building Life Cycle (BLC), for example, by modelling where the task of configuring simulated context fits into the use case of evaluating an SBA and where this use case fits into the BLC.

Use cases are modelled using the Business Process Modelling Language developed by the Object Management Group (OMG)(White 2004; Group 2009). The Business Process Modelling Notation (BPMN) is a standard for graphically representing business process modelling. BPMN aims to be understood by all business stakeholders and is widely used for modelling business processes for Web business-to-business implementation (White 2004). While BPMN shares some similarities with Unified Modelling Language (UML) Activity Diagrams, it is also more complex in the number of objects that can be modelled, like events, gateways and the types of activities themselves (Brown 1996). BPMN also supports association of activities with data objects, which is an important aspect when considering data exchange requirements to complete a use case. It should be noted though, that the complexity of both models is dependent on how they are used, and while BPMN has proven itself to be usable by non-technical users like business people, users with IT backgrounds have reported that BPMN alone does not always meet the specific requirements for software implementation projects, which are supported through UML (Ochoa, Aries et al. 2011). BPMN supports modelling use cases as part of the IDM and as such enables communication within the broader BIM community, who can review, re-use and recommend changes to improve the overall quality (Eastman, Jeong et al. 2010). Therefore, this thesis uses BPMN to model the processes involved in using context simulation to evaluate SBAs when faced with varying levels of uncertainty in context, as opposed to activity diagrams. Although, where BPMN does not satisfy the modelling requirements for the design of the simulated context model, UML diagrams are used.

4.4 Smart Building Application Life Cycle

Chapter 2 gave a high level view of where SBA evaluation fits into the building life cycle. Figure 4-1 gives an overview of this process using BPMN. As IDM promotes the re-use of use cases, this was created based upon an existing process map taken from the BuildingSmart repository for describing the life cycle of electrical equipment (Wix 2007). This repository stores all IDM developments and components carried out using confluence (a web-based wiki written in Java (Espedokken 2008)). The use of existing templates should further support dissemination of the developed use cases. As can be seen in Figure 4-1, the SBA life cycle has similarities to that of the building life cycle, i.e. design, construction (implementation), operation and disposal. The use case begins with the initial requirements gathering stage, if the requirements are found to be achievable. The next stage is the design of the SBA. At both stages, a review is conducted to determine if the SBA can be developed, and if it can, the developer then sets out to implement the design. It is at this stage that a context simulator is required to evaluate the implemented design as part of a rapid prototyping cycle. A typical use case here would be to test if the SBA functions with sensors that provide context with a certain level of precision, and in the next section, this process will be explained in greater detail.

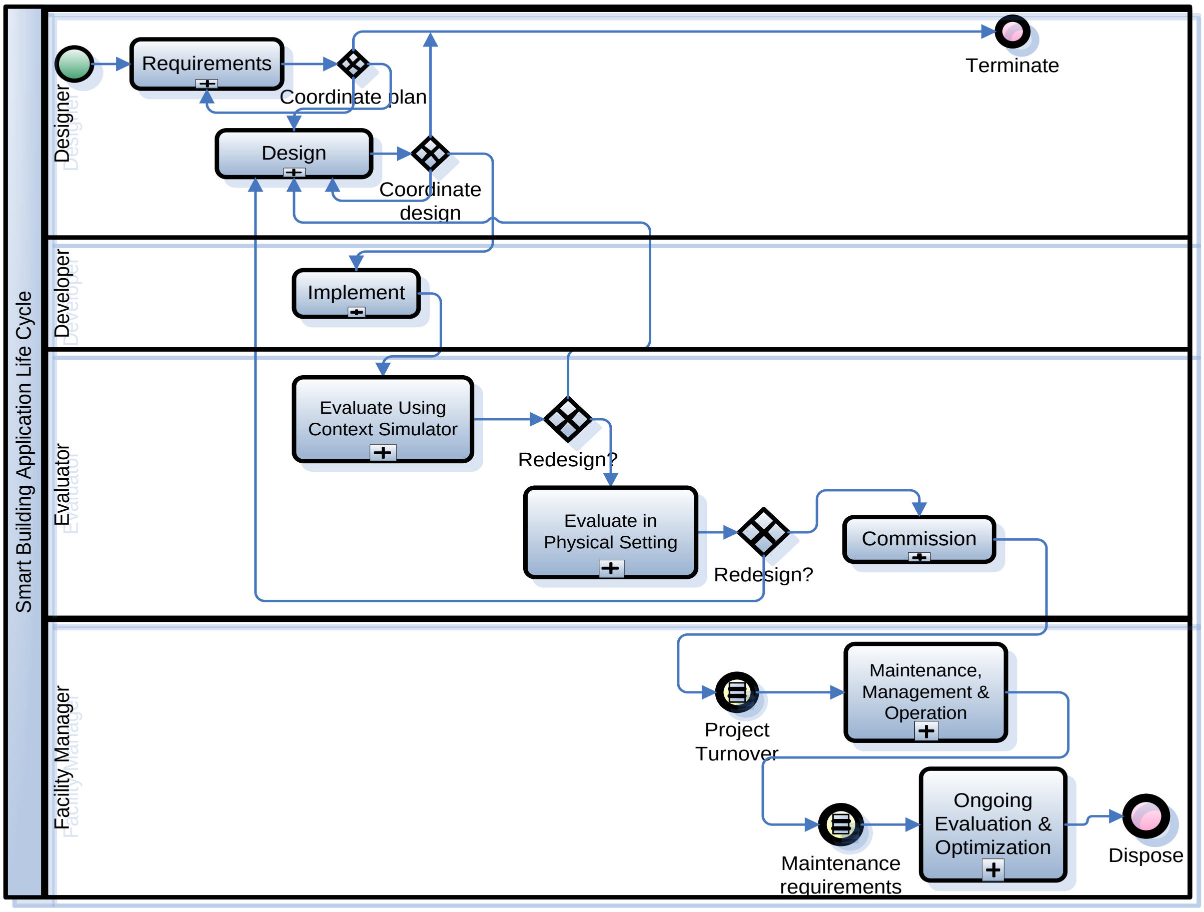


Figure 4-1 BPMN Overview SBA Life Cycle

If the SBA is found not to meet its requirements, either re-design is required or recommendations could be made for a re-design of the sensor systems for the SBA, for example, to provide higher levels of precision. If the SBA does meet its evaluation requirements, the next stage is to test it in a physical test bed. If it does not meet its evaluation requirements, it results in the same outcome as for the previous process. If it meets its evaluation requirements the SBA goes on to be commissioned and then during operation continuous testing can be conducted, and where possible, optimisation. As this thesis focuses on evaluation using context simulation, this stage will now be explored in greater detail.

4.4.1 Smart Building Application Evaluation Using Context Simulation

The process “Evaluate Using Context Simulation” identified in Figure 4-1 is shown in Figure 4-2. This process begins by determining the scenario for testing.

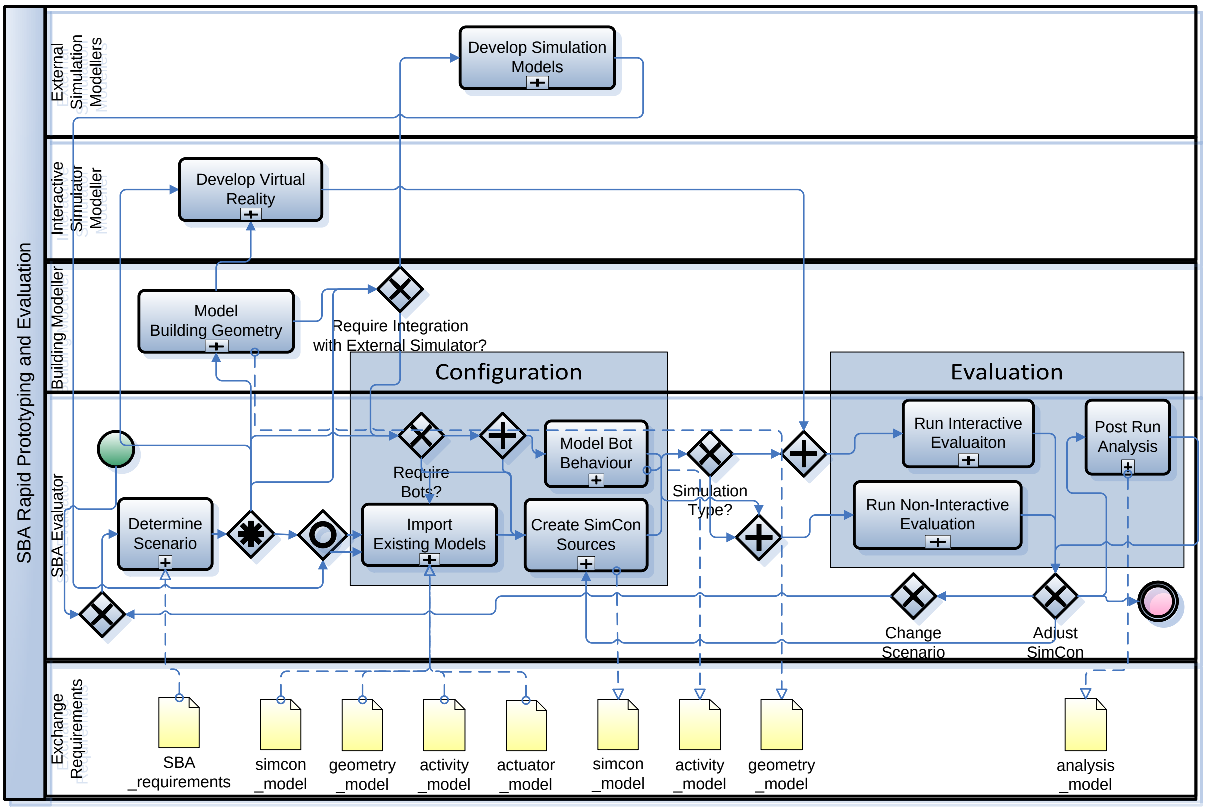


Figure 4-2 SBA Evaluation using Context Simulation

The scenario is determined by the requirements of the SBA. For example, the SBA may be intended for an existing building or it may be intended for a number of different building types. The SBA may be required to work with sensors that have already been installed in an existing building, or a range of sensor types found in different building types and with the existing building occupants or a range of building occupants for different building types. The SBA may also be required to work with actuators in the building, for example, to open or close doors. Finally, the scenario also determines whether the SBA is to be tested with real users interacting with it, the building and actuators. Once the scenario is defined, a number of questions must be answered. Is there an existing building model or models? Is there an existing Virtual Reality (VR) representation of the building for interactive testing? Are there existing models to support context simulation? Are there existing models of occupant activities for modelling bot behaviours? Are there models of the buildings actuators? And finally, is a simulation model from external simulator required, for example, to provide input for context simulation?

Where models do exist, support should be given to import them. Where they do not exist, they require modelling. A pre-requisite for the VR model is that some existing building model can be used as a blueprint. External simulation models also require that there be an existing building model, for example to model CFD (Gray and Salber 2001). Simulated context sources also require an existing building model to support their placement and the programming of paths for automatons (termed bots) representing stereotypical users moving around the simulated building. Once these have been defined the simulations are run either with real users interacting with the building using VR (represented in the simulation by their directly controlled avatars) or through the use of bots, or a combination of both. Finally, the simulation can be replayed for further post run analysis of the scenario. After the simulation run or post run analysis, the simulated context sources may be repositioned and/or reconfigured and the SBA re-evaluated. Alternatively, the entire scenario can be changed.

As the focus of this thesis is on user driven context simulation to support interactive evaluation of SBAs when faced with varying levels of uncertainty in context, the next sections will focus on the creation of a context simulation model which supports these features and which is usable by SBA developers. The model is designed to be flexible enough to handle all the context simulation approaches identified in Chapter 3 and also flexible enough to simulate a number of heterogeneous context types, fulfilling both **R1** and **R2**. Finally, it will be shown that the context simulation model shares concepts with both sensorML and IFC to meet R3. As the model has been developed through an iterative process of design and evaluation, where design decisions are the result of outcomes of evaluation these will be indicated and a reference to the evaluation in Chapter 6 which resulted in that design decision will be provided.

4.4.2 SimCon Context Simulator Overview

Figure 4-3 gives an overview of the SimCon architecture and where the SimCon context simulation models fits into the process described in the previous section.

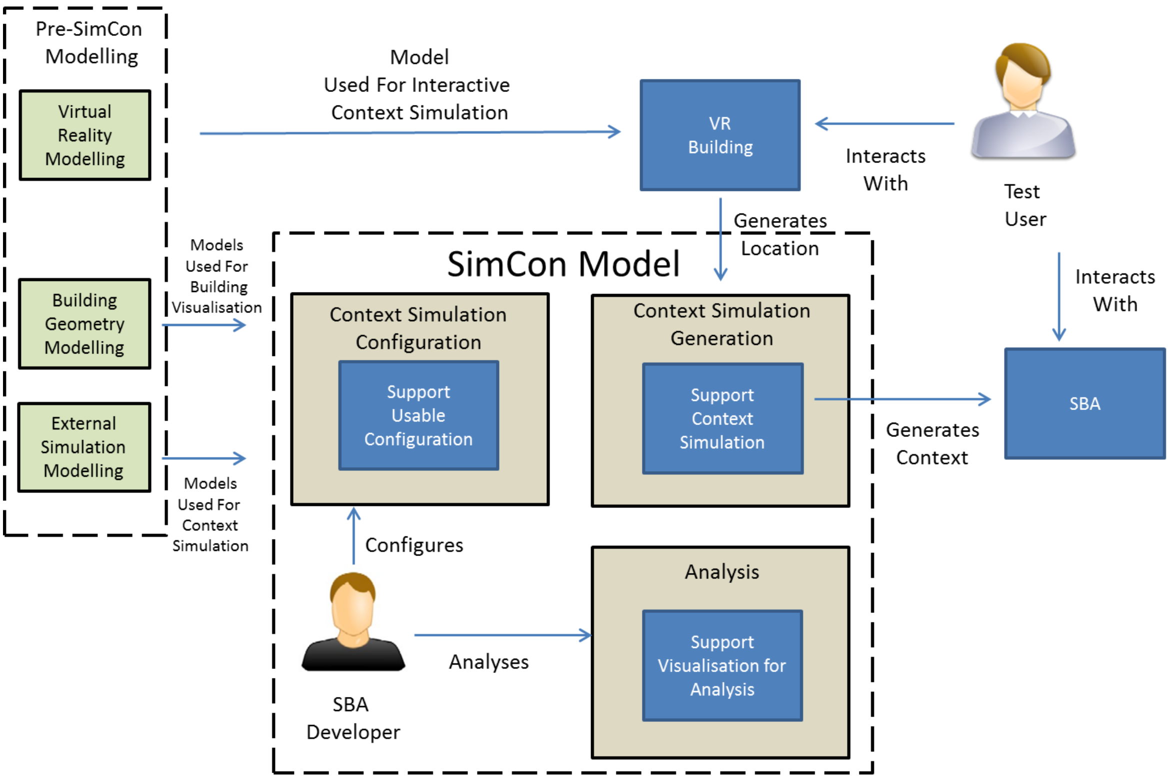


Figure 4-3 SimCon Architecture Overview

The left of the figure gives an overview of the pre-requisite modelling which occurs before the SimCon model can be used, and which is required to achieve certain types of simulation. While the overview does not explicitly capture the order in which modelling occurs, it is assumed that the building geometry model occurs first and the minimum requirements to enable use of the SimCon model and its implementation is a 2D model of the building geometry. All pre-SimCon modelling is handled by external tools and is not described here in any greater detail. The models will be referred to where they are relevant to context simulation. The SBA developer has two roles; the first is to configure instances of the SimCon model by creating, placing and configuring simulated context sources. The resulting model will drive the context simulation which is then fed into the SBA. The context simulator may also make use of data on the location of one or more user controlled avatars within a VR environment, supporting interactive context simulation. It should be noted that the SBA developer may also control an avatar if they wish to test the SBA themselves. The second SBA developer role is to use SimCon toolset implementation to analyse the behaviour of the SimCon sources to support the evaluation process. Tools for visualising context and uncertainty in context are employed at this stage to improve understanding for the SBA developer of the effects of uncertainty on the SBA behaviour. These roles cover both **R1** and **R2** as defined above. The next section will describe the SimCon model.

4.4.3 SimCon Context Simulation Model and Contum Definition

Figure 4-4 is a UML class diagram of the SimCon model. This model is derived from the requirements as set down at the beginning of this chapter and also through the state of the art review in Chapter 3. The model is designed to be flexible so that it supports simulation of different types of context, from low level to high level. It also is flexible in the number of simulation approaches it supports. The SimCon model begins with a model of the building. The building is identified using a global unique identifier. At a minimum a 2D geometric model of the building must exist to support SimCon, as a 2D representation is required when placing SimCon sources and also for visualising simulated context. A building is assigned a SimCon Generator which handles all the context simulation as well as any communication with the interactive VR environment, for example, via TCP/IP. It also maintains a global clock for all SimCon sources, so that it can control the generation of contums.

Contum[1] is a term introduced in (McGlinn, O'Neill et al. 2010) and explained in greater detail in this thesis. It is used to describe "a discrete piece of context which also has an associated level of uncertainty", where uncertainty is a measure of its precision and timeliness. Timeliness is related to the rate of context generation and any additional delays, e.g. due to the communication medium, and precision is the closeness of agreement between indications or measured quantity values obtained by replicate measurements on the same or similar objects under specified conditions. The level of uncertainty of a contum is an attempt to quantify the difference between the actual event and the measured event. In the case of location, this could be determined by measuring the position of an object according to an agreed coordinate system and then comparing this against the measured position of that object using a location technology, like Ubisense for example.

[1] While the term contum shares some similarities with the word quantum, the two are quite different, the only similarity being an understanding that there is an inherent level of uncertainty in all measurements due to the process of sensing phenomena.

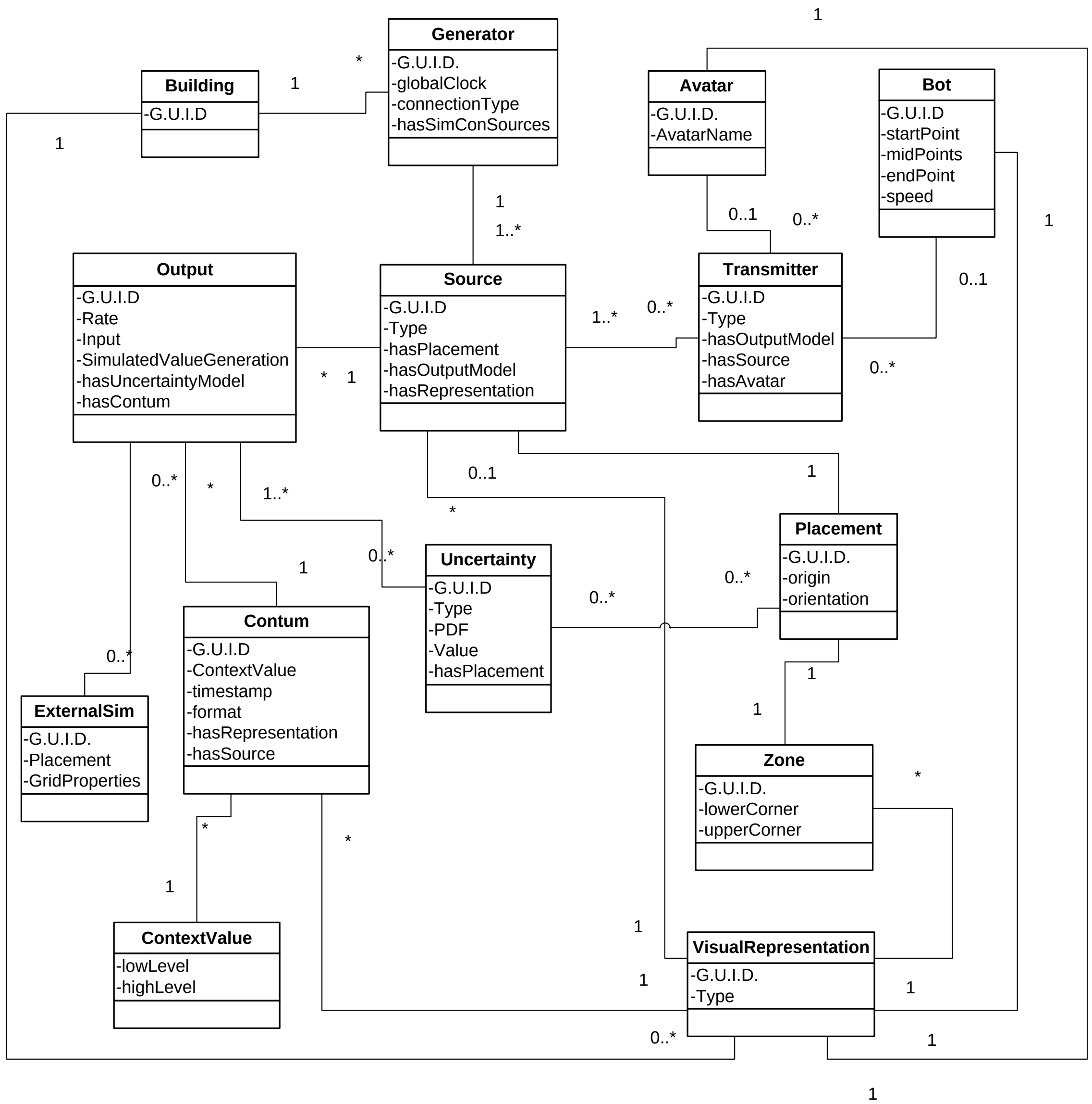


Figure 4-4 UML Class Diagram: SimCon Model

Contum generation can be synched with other SimCon sources, or each source can start at a pseudo random time. A SimCon Generator has one or more SimCon sources.

4.4.3.1 *SimCon Source Model*

The SimCon Source model gives all the fundamental properties of the SimCon source. Each SimCon source must have a globally unique identifier (GUID) and type defined. All SimCon sources which sense some phenomena must also have a placement defined, as the placement has a relationship to the values the SimCon source will generate. These values are generated by the Output model. Finally, a SimCon source has a representation for visually displaying it. Table 4-1 gives the properties of the SimCon Source model and a description of each property.

Table 4-1 SimCon Source Properties

Property	Description
G.U.I.D	Global Unique ID (A globally unique id).
Type	The type of SimCon source. This can be a generic type of context source, e.g. a passive infrared sensor or thermometer. Or a more specific type, e.g. a Ubisense cell.
hasPlacement	See Placement
hasOutputModel	See Output Model
hasRepresentation	See Representation

4.4.3.2 SimCon Placement Model

All SimCon sources which measure some phenomena require a placement. Table 4-2 gives the properties of the SimCon Placement model and a description of each property.

Table 4-2 SimCon Placement Model

Property	Description
G.U.I.D	Global Unique ID
Origin	The origin (a 3D point) of the SimCon source relative to the buildings coordinate system. For example, the position of a receiver or temperature sensor.
Orientation	The orientation (a 3D vector) may have an impact on wireless communication.
hasZone	The zone defines the boundaries of the simulation beyond which no parameters of anything associated with that zone are considered. For example a Ubisense cell has a boundary outside of which a tag does not generate a location value.

4.4.3.3 Output Model

All SimCon sources generate an output which is a contum. A contum is a discrete unit of context which has an associated level of uncertainty. Table 4-3 gives the properties of the SimCon Output model and a description of each property.

Table 4-3 SimCon Output Model

Property	Description
G.U.I.D	Global Unique ID
Rate	The rate at which context values are generated, for example every tenth of a second, or every fifteen minutes.
Input	Any inputs which can be used as parameters into the simulation process. For example, the location of a user controlled avatar.
SimulatedValueGeneration	How the simulated output values are generated, for example, bot or user driven values, playbacks of recorded values , statistical models based on recorded values, mathematical formulae to generate values, or values generated from external simulators.
hasUncertaintyModel	See Uncertainty Model
hasContum	See Contum

Three important concepts in the Output model are the input, output and SimulatedValueGeneration. These support the modelling of SimCon sources as a process as defined in sensorML, which takes one or more inputs, and based on parameters and methodologies, generate one or more outputs (Botts 2002). Examples of inputs in SimCon are the location of one or more avatars in a VR environment. SimCon models these processes using an activity diagram approach which is evaluated in Evaluation 1A (see Chapter 6 section 6.3.1) and found to be a more suitable method for capturing the process aspect of context simulation than the semantically equivalent sequence diagram. The SimulatedValueGeneration is handled using one of a number of different approaches identified from the state of the art in Chapter 3. Here, contums are generated on a configurable sample rate or in combination with additional inputs. For example, generating a presence event for a simulated passive infrared (PIR) context source can be triggered by an input indicating when an avatar walks into the PIR zone in the simulation. Modelling SimCon sources as processes becomes important when considering higher level simulated contexts, which may be generated from low level contexts configured as inputs, using well-understood concepts of process composition, where the output of one process is fed into the input of another. The different approaches for generating simulated context values and how they are modelled within SimCon are described in more detail next.

4.4.3.3.1 Playbacks and Statistical Models

One method for generating context is by replaying context values collected from physical context sources. SimCon models a playback as an array of values assigned a rate. The rate can be based on an actual time stamp provided by the context source, or a new rate may be configured by the SBA developer. The simulated context source can then be placed anywhere in the simulated environment. Careful consideration should be taken to place the simulated context source in an environment which is very similar to that of the physical context source and at the same time frame, for example, where the time of day will have a significant impact on the generated value.

Alternatively, the data can be used to derive statistical models of the context source. For example, the mean and standard deviation for a temperature sensor for a particular span of time can be determined. The mean is then used to simulate the generated value and the standard deviation to vary the value at each measurement interval. The SBA developer can then place and configure the SimCon source, for example with different rates or by using a time series. A time series is a multi-dimensional array which models a time period, a value and a variation for that value, for example, the 12 hours of a day and the different values and variation for each hour. This method can result a less predictable simulation that that of playbacks, while also remaining grounded in statistical models parameterised from actual sensor readings. The statistical approach also offers more flexibility in generating context in settings that do not match exactly the conditions under which playback data was initially captured. .

4.4.3.3.2 User Driven Simulation Model

As highlighted in Chapter 3, designs which include evaluations involving users in the design process through a series of iterations lead to the best designs (Carter and Mankoff 2005). For SBA evaluators, this requires evaluating the SBA usage with real users. VR supports this type of user-centred evaluation of SBAs without the need of a physical building. Sensors within a physical building may be influenced by the presence and activities of occupants. This thesis refers to simulation of context which is affected by occupants of the building as user driven context simulation. In order to simulate user driven context, data must be extracted from the VR environment on each avatar's location.

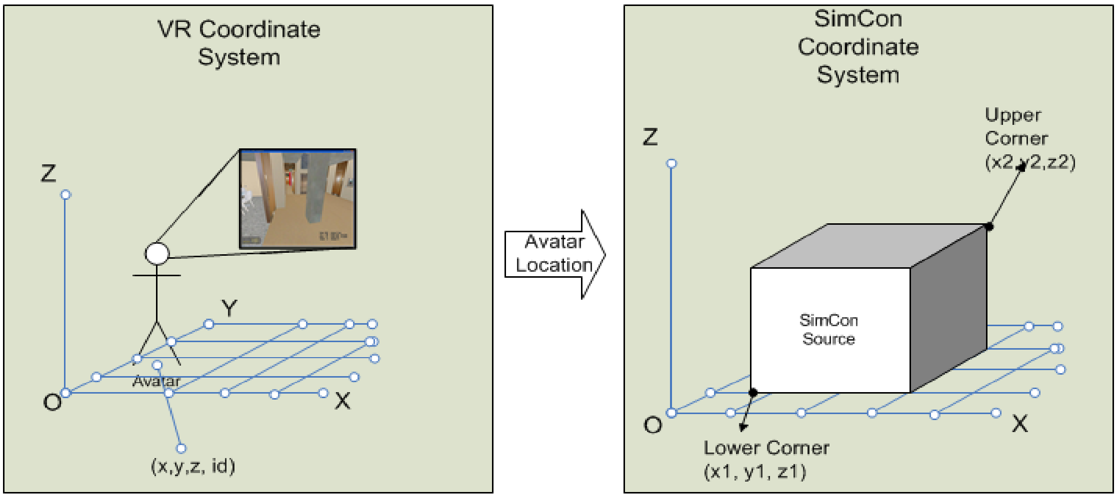


Figure 4-5 User Driven Context Simulation

A VR building maintains a global view on the position of each avatar in relation to the environment's origin, providing the user's avatar's x, y and z position at a specific time, tied to a Euclidean coordinate system (Figure 4-5). A SimCon source is defined with the same coordinate system as the VR environment. This way, the data can be used to simulate location context, for example in the case of tag based coordinate context generation, like Ubisense, or a presence event by a pressure mat or passive infrared sensor. It can also be used as parameters into other types of simulation, such as temperature, or CO2, where the number of occupants can affect the simulated value. The sensor zone defines the boundaries beyond which the avatar's location is no longer used as an input into the simulation. It should be noted that the VR engine places limits on what context can be realistically simulated for SBA evaluation. This will be discussed in more detail in the implementation Chapter 5.

4.4.3.3.3 Bot Driven Simulation Model

Bot driven context works on the same principle as user driven context, only now it is the position of the programmed bots which is used as an input into the context simulation. The process of modelling bots is outside the scope of this thesis and is examined in greater detail in the work of O'Neill (O'Neill, Lewis et al. 2009; O'Neill 2011). This thesis does include a rudimentary model for handling bot simulation, as it was found to be necessary from findings in Evaluation 2 Chapter 6. Table 4-4 gives the properties of this model. A bot is defined as a number of waypoints. It then moves in a straight line from one point to the next at a fixed speed. Waypoints should therefore not be set so that the bot moves through walls. The SimCon model also supports recording an avatars position as a user moves around the VR environment. In this way, bots can be "recorded" and played repeatedly. Replaying bots can allow an SBA developer explore a particular situation multiple times, as part of repeatable evaluations.

Table 4-4 SimCon Bot Model

Property	Description
G.U.I.D	Global Unique ID
StartPoint	The start point of a bots movement.
MidPoint[0…n]	A series of points which the bot moves from following a straight line
EndPoint	The point at which a bot stops.
Speed	The speed at which the bot moves, for example, 2.8 meters per second.

4.4.3.3.4 Simulation using Mathematical Formula

Chapter 3 discussed the use of a mathematical formula to simulate context. For example, to simulate the received signal strength between a transmitter and a receiver the formula given in Chapter 3 Equation 2 for a transmitter with power p_t radiating isotropically in free space (uniformly in all directions) is known to give a power flux density s at distance r of

$$s = \frac{p_t}{4\pi r^2}$$

The value p can be determined from the specifications of the transmitter. R is simply the distance between the transmitter and the receiver. Such a model will not give a very accurate representation of the actual RSS in a building environment, and so methods which set out to model the variance as Gaussian processes discussed in Chapter 2 (Seco, Plagemann et al. 2010) may result in more accurate simulations, although this method is not evaluated in this thesis and so cannot be validated. Other approaches can look at using different parameters as part of the simulation process and these are discussed in more detail in Chapter 2, but they are not addressed in this thesis.

4.4.3.3.5 External Simulators

External simulators can also provide values for context simulation. For example, existing simulation software such as ANSYS can generate thermal flows in buildings at discrete intervals of time (Gray and Salber 2001). Outputs are defined as three dimensional grids that may cover the entire volume of the building or an area of the building where granularity can be set so that each point in the grid represents a discrete value and a discrete volume of space. By associating a SimCon source with such a model, a value can then be generated representing the simulated value at that point, for example a temperature value. Once again, a time series can be defined so that the CFD model changes over time.

Radio Frequency (RF) propagation tools can also provide grid based outputs to drive simulation (McGlinn, O'Neill et al. 2010). Here the RF propagation model can be associated with a simulated transmitter. As the user moves his avatar and it changes orientation, the VR environment generates coordinate and orientation data. The grid based model can use these to update accordingly. Simulated receivers can then be used to generate received signal strength between the transmitter and the receivers, thus supporting interactivity at different levels of granularity, for example, for eight different orientations. Interactivity can also be achieved for CFD models, although this would require a large number of CFD outputs to account for all possible interactions of the user with the environment and the effects of that interaction on the thermal flows, and the extra processing required along with the additional time to develop the external simulation models are the main drawbacks of this method as part of a rapid prototyping cycle.

4.4.3.4 Transmitter Model

A SimCon source may be associated with a transmitter or tag. This can be either associated with an entity or a person (avatar). A transmitter model was developed to support multiple types of interactive heterogeneous context source simulations, as the context simulator was required to distinguish between avatars carrying transmitters and those which are not. Table 4-5 gives the properties of the SimCon Transmitter model and a description of each property.

Table 4-5 SimCon Transmitter Model

Property	Description
G.U.I.D	Global Unique ID
Type	Type of Transmitter, (e.g. a Ubisense tag or a ZigBee transmitter)
hasOutputModel	How the simulated output values are generated, for example, statistical models based on recorded values, mathematical formulae to generate values, or values generated from external simulators. See Output model.
hasSource	A transmitter must be associated with some type of receiver. See SimCon source model.
hasAvatar	A transmitter can be associated with an avatar.

hasPlacement	The position of the entity it is related to.

4.4.3.5 Uncertainty Model

Each SimCon source can have zero or more uncertainty models associated with it. The uncertainty model here sets out to support introducing uncertainty as part of a rapid prototyping cycle to lead to a better understanding for SBA developers regarding the impact of uncertainty on SBAs. This approach is evaluated in Chapter 6 from both the aspect of usable configuration and how effective the approach is for SBA developers evaluating their SBA when dealing with the effects of uncertainty in context. Table 4-6 gives the properties of the SimCon Uncertainty model and a description of each property.

Table 4-6 SimCon Uncertainty Model

Property	Description
G.U.I.D	Global Unique ID
Type	Defines what the uncertainty model is applied to, e.g. Output, Rate or Zone.
Distribution parameters	For example, a value defining the boundaries of a Gaussian distribution at one standard deviation.
Probability Distribution Function Type	The type of distribution, for example a Gaussian distribution, or a Poisson distribution.
hasPlacement	An uncertainty Zone has a placement, and this defines additional environmental interference for things like mobile transmitters.

An uncertainty model defines a variation in the output value of the SimCon source, for example to define its precision, or the rate of the SimCon source, for example to introduce a variable delay. The variation is defined using a probability distribution. It can be applied to any of the simulated context generated by any of the simulation approaches discussed in the Output model. For example, to simulate a Ubisense real time location system, the position generated by a VR environment can have a simulated transmitter associated with it. The x, y and z coordinate of this tag can be offset then by an amount given in the Ubisense specifications for its precision. This is given as 15cm 95% of the time (Steggles and Gschwind 2005). This value can then be used to model a Gaussian distribution which offsets the value on either side of one of the three axes (x, y or z). For example, modelling the variance as 15cms for one standard deviation means that at 2 standard deviations (95% of the time) the location will fall within 15cms of the actual location of that axis. Using this approach, if the simulated tag was now given a fixed location in the VR environment, SimCon would cause that location to vary by the amount specified in the Uncertainty model.

As highlighted in Chapter 2, the causes of uncertainty for location systems like Ubisense require knowledge of both the system and environmental effects. For example, the location returned by a Ubisense tag can vary from its actual position due to the environment. A method is needed to support easily and flexibly introducing uncertainty to reflect the effect introduced by the environment on the generated context. The use of uncertainty zones is therefore used to support the modelling of localised areas of assumed uncertainty producing effect impacting on SimCon sources. This gives more flexibility, as zones in buildings can be assigned uncertainty levels more appropriate to the geometry and materials of the building, for example where windows are likely to increase multipath issues for wireless transmitters. SimCon therefore supports modelling uncertainty zones to increase the amount of uncertainty related to the environment. These zones can overlap or be nested, allowing more complex patterns of differing precision to be assigned to an individual context source.

An uncertainty model can also be associated with other parameters where these are likely to increase or decrease the variability in the generated value due to uncertainty, like temperature, although this has not been evaluated further in this thesis.

4.4.3.6 Contum Model

A contum model defines a discrete unit of context which has an associated level of uncertainty that it devices from the associated source. Table 4-6 gives the properties of the SimCon Contum model and a description of each property.

Table 4-7 SimCon Contum Model

Property	Description
G.U.I.D	Global Unique ID
timeStamp	A timestamp generated by the SimCon Generator.
ContextValue	The value of the contum. Can be low or high level. For example, a presences event (on/off), a received signal strength, a coordinate, a temperature, an activity (running, walking), etc.
hasSource	The source of the contum
hasRepresentation	How the contum is visually represented, for visual debugging SBA when faced with effects like uncertainty.
hasFormat	The format of the contum, for example XML.

4.4.3.7 Visuals Representation Model

For visualisation purposes, aspects of the SimCon model are associated with visual representation models. The visual representations are the result of both the State of the Art review of existing context simulators, and findings from early evaluations of the SimCon model in Evaluation 1 Chapter 6. For example, a SimCon source has a position and zone. These must be represented visually for placement. Also required for placement is a visual representation of the building (for example walls and doors) and a coordinate system to support placement. Finally, for supporting intelligibility, avatar and bot position and contum visualisation (including simulated sensed avatar and bot location) are supported. A visual representation can be as simple as a geometric object, for example a cube, sphere, or cylinder, to more complex 3D objects with multiple vertices and faces. Context can thus be represented using combinations of shapes, for example, a concentric ring for proximity, and colours, for example, to represent different temperature values. The grid model for external simulation models can also be visualised providing feedback of the simulation visually as it occurs, by representing each point in the grid as a coloured sphere or square. Such visualisations can be turned on or off as needed during an experiment.

4.4.3.8 SimCon Model Summary

This section presented the design of the SimCon model to meet both **R1** and **R2** as defined at the beginning of this chapter. With respect to **R1,** the SimCon model has been designed to be flexible in its support for multiple context simulation approaches. These include playback and statistical models based on physical readings, the use of mathematical functions, and bot and user driven variation in location. It also supports generation of context values using external simulation models. Based upon these different simulation approaches a range of heterogeneous contexts can be simulated, including three types of location context (presence, proximity and coordinate) and also temperature. Also with respect to **R1,** the SimCon model addresses modelling uncertainty in low level context to support simulation, i.e. the context precision and timeliness properties. Precision is modelled through the use of probability density functions, timeliness through a combination of rate and also delays. The modelling of uncertainty zones supports flexibility in the approach to introducing varying levels of uncertainty in simulated context which has a relation to its position in the environment, for example wireless location tracking. With respect to **R2** SimCon has addressed usability both at the configuration stage and also at the evaluation stage through the modelling of a number of properties which can support SBA developers in the task of creating, placing and configuring simulated context sources with uncertainty. In particular, the uncertainty model through the use of configurable zones of varying precision abstracts from the complexity of modelling uncertainty in buildings. SimCon also models the visual representation of context, which supports the development of analysis tools for visualising the effects of uncertainty in context, which is again tied to the usability of SimCon.

The next section will examine **R3** and the interoperability of the SimCon model, by comparing it with existing sensor modelling approaches in both sensorML and IFC.

4.4.4 SimCon Model Shared Concepts

A requirement of the SimCon Model is that it is interoperable with existing standards. As a result the SimCon model concepts are grounded in sensorML (Botts 2002) and also the Industry Foundation Classes (ISO 2005). Table 4-8 shows a breakdown of the main SimCon concepts and equivalent concepts in sensorML and IFC. We only examine concepts which are relevant to the SimCon model, as these concepts are currently sufficient to support contum generation.

Table 4-8 SimCon Model Conceptual Mappings with two Existing Sensor Models

SimCon	SensorML	IFC 2x4
Source	Component	IfcSensor
Placement	Position	IfcPlacement
Zone	Envelope	IfcZone
Building	GeoLocationArea	IfcBuilding
Output	AbstractProcess	n/a
Representation	n/a	IfcProductRepresentation
Uncertainty	Response Curve	n/a
Transmitter	Component	n/a
Contum	n/a	n/a

As can be seen, SimCon, sensorML and IFC share concepts which are similar. For example, both sensorML and IFC have concepts related to uniquely identifying a sensor or context source. SensorML uses gml:id, defined in the Geographical Mark-up Language (GML) (Henricksen, Indulska et al. 2003) name space to uniquely identify all concepts in a sensorML model. IFC uses a unique global identifier IfcGloballyUniqueID*,* which is an IFC defined type that is a fixed length string value. The stored value is a compressed Globally Unique Identifier (GUID). The type of sensor can be defined using name in sensorML, and the IfcObject type in IFC. These concepts can be directly mapped to Source id and type.

The next concepts which share similarities in both sensorML and IFC models relate to the position or placement of the sensor. In sensorML, positioning is handled through the use of the Position concept. Positional data is used to specify the position of a local coordinate frame to an external reference coordinate frame. These are specified by the localFrame and referenceFrame attributes, respectively. SensorML also provides the means to define envelopes which specify a region and also a geolocation area which can contain multiple envelopes. Placement, Zone and Building in the SimCon Model can be mapped directly to these concepts.

The realization of an IFC object is called a "product". All products have a placement. Placement can either be absolute (relative to the world coordinate system), relative (relative to the object placement of another product), or constrained (relative to grid axes). SimCon Source placement can be mapped to IFC absolute placement. IfcSite and IfcBuilding are conceptually similar to Building, in that they contain information on the type and name of the building, and also additional data on its geodetic coordinates. IfcZone is a concept which can be mapped to SimCon Zone.

At this point the sensorML and IFC models begin to diverge in their capabilities. SensorML defines an abstract process using inputs, outputs and parameters, and this can be mapped to SimCon Output. IFC contains no capability for defining a process or arrays of values in a manner which would support all the approaches to context simulation. There is also little support in sensorML or IFC for modelling the output of a context source. A potential exists to use transducerML to define the outputs of context sources, although this is not explored in this thesis (Hofer, Schwinger et al. 2003). No such ability exists within IFC. Both sensorML and IFC do support the modelling of values, so the value of the contum can be represented. In sensorML this is modelled using “Output” and in IFC as “IfcPropertySingleValue”.

IFC also contains data on the visual representation of objects, which sensorML does not. SimCon currently only supports simple representations of the SimCon sources and contums, although IFC can be used to model all the necessary geometric data to render an object’s shape and also provides, through the use of extensions, the capability to contain additional data, for example textures, reflectivity of surfaces, etc. Finally, as IFC supports the modelling of the complete building, an integrated model based on IFC ultimately has the potential to provide all the data necessary for parametric simulations which make use of building elements and materials.

4.4.4.1 IFC Extensions to Support Context Simulation

This thesis has applied the IDM methodology to support both dissemination of the proposed context simulation model within the wider BIM community and also as a method for applying for standardisation of the developed model. Once the use cases have been defined in BPMN, IDM requires that the resulting data exchanges are modelled using Model View Definitions (MVDs) (Klyne, Reynolds et al. 2004). MVD is the standard methodology and format for documenting the software implementation requirements for standard Industry Foundation Classes (IFC) based data exchange. They support the mapping of generic concepts to particular instances of a model schema, for example, IFC. Appendix A gives an example of an MVD developed using software supplied by BLIS (See 2012). This contributes to the integration of the designed context simulation model with BIM and towards an integrated data model for evaluating SBAs.

4.5 Summary and Conclusion

Chapter 4 highlighted the requirements for the design of the SimCon model. It then presented the designed model using two standard approaches to capture the processes and the model. The first standard is the Information Delivery Manual which is used for dissemination of Building Information Models (BIMs) and also standardisation of Industry Foundation Classes (IFC) development. This supported the development of a use case for the approach. The second standard is Unified Modelling Language (UML) and class diagrams where used to represent the SimCon model. The SimCon model has several components. These include models for identifying the SimCon source and its type. Additional components support placement of the SimCon source in the building as well as defining zones which are used to determine the boundaries of simulation, i.e. beyond which point no building parameters affect the simulated value, and also for defining zones which introduce additional uncertainty in context simulation.

An Output model provides the capabilities to support a number of different approaches to simulating context. These are derived from the state of the art review in Chapter 3. Of most significance to this thesis is the approach taken to simulating user driven context and which is enabled by integration of SimCon with a Virtual Reality simulator. An Uncertainty model provides a method for introducing uncertainty into simulated values to generate "contums" and a Representation model has been developed to visualise the building and contums to aid with both the configuration of the simulation and also the evaluation process. The model has been designed to be usable by SBA developers, and this claim will be evaluated in Chapter 6. These combined features aim to meet **R1** and **R2** of this thesis. Finally, the SimCon model has been designed to share concepts with existing standards in both the sensor and building communities, thus improving interoperability beyond the scope of the thesis. The IDM process shall further enable the developed model to be integrated within the BIM community.

5 Implementation

5.1 Introduction

This chapter presents the implementation of the SimCon model presented in Chapter 4. It begins by introducing the different components which make up this implementation. These are called SimConfig, SimConGen and SimConViz and are arranged in the architecture introduced in section 5.2. They are implemented to validate the design concepts presented in the previous chapter and to provide a platform for evaluating the usability of the SimCon model, thereby addressing design requirements **R1** and **R2** which were defined in Chapter 4. The implementation section starts with a description of the pre-requisite modelling required to enable the use of the SimCon model in an implementation. As the thesis focuses on user driven context, this also required the creation of the interactive virtual building, for which the existing interactive context simulator, Tatus(O'Neill, Klepal et al. 2005), which had been used in previous context simulation model evaluation (see Chapter 3 section 3.4.1.9) was used.

This is followed by presenting the SimConfig tool for creating, placing and configuring SimCon sources in a simulated building is presented. This leads onto describing the SimConGen simulated context generator which includes an explanation of how the different context simulation approaches are implemented as well as how it connects to Half Life 2 to support user driven context simulation. To conclude, the SimConViz tool is described, which enables visualisation of simulated context to further support evaluation of the SBA.

5.2 Overview of the SimCon System

Figure 5-1 builds upon the overview diagram from the design detailed in Chapter 4. Here the different implemented components of the SimCon implementation are discussed together with their relation to the SimCon model. It begins with a discussion of the pre-SimCon modelling requirements.

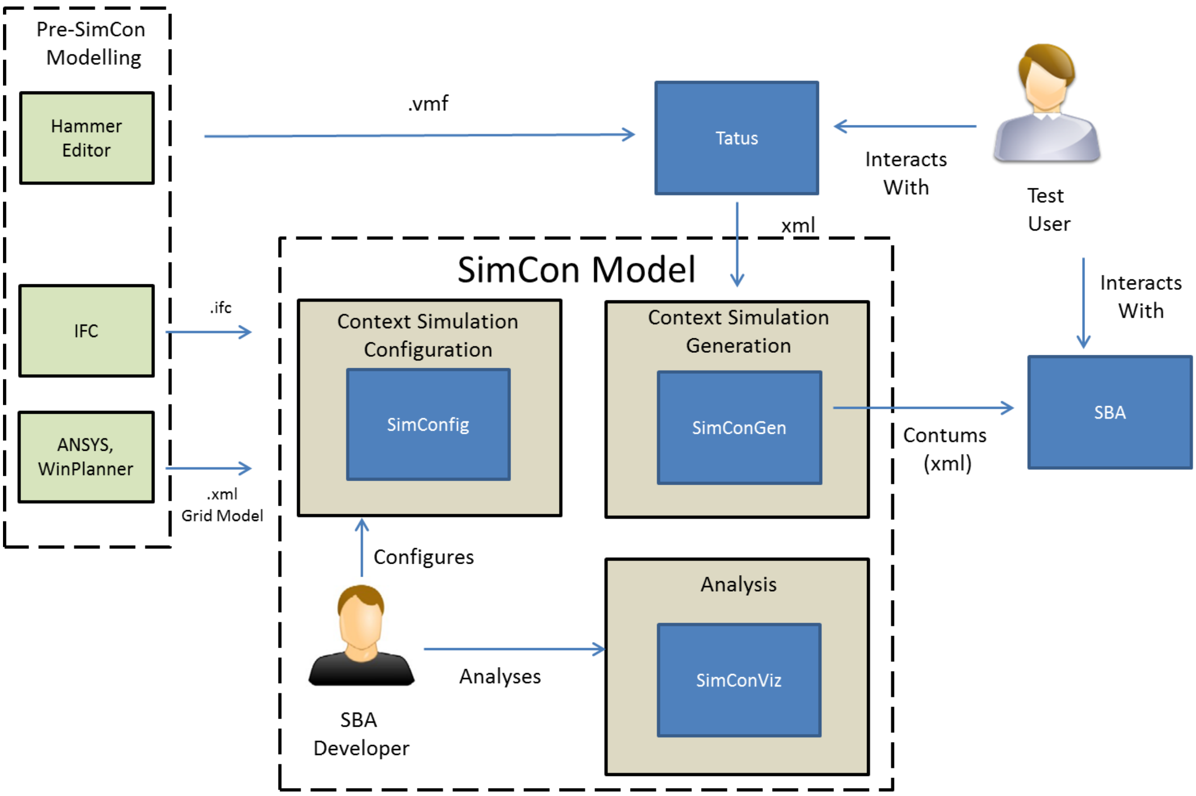


Figure 5-1 Overview SimCon Implementation: SimCon model components (Simconfig and SimConViz tools and SimConGen contum Generator), External Modelling requirements, Half Life 2 Games Engine and Smart Building Application.

5.3 Pre-SimCon Modelling

Before evaluation using the SimCon model can take place a 2D model of the building geometry must exist. The SimCon model will function without a 3D model as it supports both user driven (which requires a 3D model) and bot driven context simulation (which does not necessarily require a 3D model) to meet the requirements of flexibility in context simulation approaches. Also, while the ideal is that a 3D model of the building geometry has been created before evaluation using SimCon takes place, 2D models of building geometry are more common. The 2D geometry can be modelled using any of a number of commercial or free tools, for example Google SketchUp(Aberer, Hauswirth et al. 2006), Graphisoft ArchiCAD (Graphisoft 2012) or AutoDesk Revit Architecture (Autodesk 2012). The 2D geometric model is used to support the SimConfig and SimConViz tools and how it is integrated into these tools is discussed in section 5.3.2. First, the process of creating the 3D visual model of the building to support user driven context simulation is discussed. This component is currently provided by modelling tools used by the Tatus Platform which is presented next.

5.3.1 Pre-Modelling: The Tatus Platform

The interactive building component is provided by the Tatus platform which uses the Half Life 2 (HL2) games engine (HL2 2012) to exploit the 3D graphics engine and provides a realistic user experience with which to evaluate SBAs (O'Neill, Klepal et al. 2005). The Local Area Network (LAN) style implementation of the HL2 engine supports up to 32 users interacting in a single experiment over a network of PCs. A user interacts using the mouse and keyboard supporting movement of an avatar through the building in all directions on the horizontal plane and also by rotation of their field of view around the point which the avatar occupies in the simulator. The avatar can walk, run and jump and is subject to the physics of the games engine, for example gravity. Using the avatar the user can also open and close doors and pick up or drop items. The Software Development Kit (SDK), which comes with the games engine, provides tools for configuring the VR building. It also provides limited AI capabilities and scripted sequences to include non-player-characters (NPCs – equivalent to 'bots' as used in this thesis) and is part of parallel work conducted by O'Neill (O'Neill, Lewis et al. 2009; O'Neill 2011). Half Life 2 therefore provides a semi-realistic visual representation of a building which a user can move in and interact with for the purpose of testing and evaluating SBAs.

To create the virtual building for HL2 the associated Software Development Kit (SDK) provides a tool called Hammer to develop new "maps" or "levels". Each map is a combination of basic shapes and through the application of textures the map can be made to look like a semi-realistic building. There is no automated process available for importing an existing Building Information Model (BIM) like Industry Foundation Classes (IFC) into the Hammer editor, as it is a tool designed for computer games developers rather than for Building Life Cycle (BLC) users. A plugin does exist for importing 3D Google SketchUp models into HL2 which has the potential for improving the time for creating the VR models for HL2 although this is dependent on the complexity of the model. This process is examined in greater detail by O'Neill (O'Neill 2011).

Figure 5-2 Virtual Reality Buildings in Half Life 2 for supporting user driven context generation (Left: Environmental Research Institute Cork, Right: Lloyd Building TCD)

Alternatively, an existing BIM (for example IFC) can be converted into a 2D image and imported into the hammer editor to provide a 2D blue print for modelling in Hammer(O'Neill, Lewis et al. 2006). Each HL2 map files must have a coordinate system defined. At present, all maps are given an origin based at ground level and in the south east corner. Figure 5-2 shows some examples VR buildings (the Environmental Research Institute Cork (ERI 2011) and the Lloyd Building Trinity College Dublin (TCD 2012). To model the Lloyd building in Trinity College Dublin using the Hammer editor and a 2D blue print took an undergraduate intern, untrained in the Hammer map editor, 22 working days. The building features 104 rooms comprised of offices, computer labs and lecture rooms. In total these rooms are furnished with 520 desks, 352 chairs and 257 replica desktop computers. Once the building has been modelled some additional steps are required to enable SimCon. These are discussed in the next section.

5.3.2 Pre-Modelling: SimConfig and SimConViz

The SimConfig and SimConViz tool both require a visual representation of the building to support placement of SimCon sources and also visualisation of simulated context. To achieve this, all that is required is a 2D representation of the building walls. Currently there is no direct import of IFC into the SimCon model to populate building geometry. It does support the conversion of 2D geometric models into a format, i.e. x and y coordinates for wall segments, which SimCon can import through the use of Microsoft Visio. The Visio file is then parsed for the relevant information (currently walls) using a parser implemented in the Python programming language (Foundation 2012). It may then be necessary to adjust the 2D geometric model in a commercial CAD tool, for example ArchiCAD, to remove unwanted physical representations. These can include unwanted artefacts such as furniture or information relating to geometric properties of the building written directly into the 2D file, for example, “Fire Exit”, as these will be visually displayed otherwise. Also, where lines are repeated in the file (one on top of the other) it is advisable to flatten these to reduce the number of polygons which require being drawn and ArchiCAD supports this process. The entire exercise currently takes a person familiar with these tools between one or two hours for a building model of the size mentioned above, depending on the size and the number of unwanted artefacts. The resulting model can then be represented using both SimConfig and SimConViz.

The 2D and 2.5D VR Building Visualisation (Figure 5-3) are implemented in Java using the open graphical language openGL libraries version 1.2 (Shreiner 1999) which have been bound to the Standard Widget Toolkit (SWT) version 3.2 (Northover and Wilson 2004). OpenGL is a standard specification defining a cross-language, cross-platform Application Programming Interface (API) for writing applications that produce 2D and 3D computer graphics. OpenGL provides over 250 different function calls which can be used to draw complex three-dimensional scenes from simple primitives. These give a great degree of control over the underlying code and provide an extensible basis for visualising a wide range of objects from building elements, to graphical representation of SimCon sources and of contums.

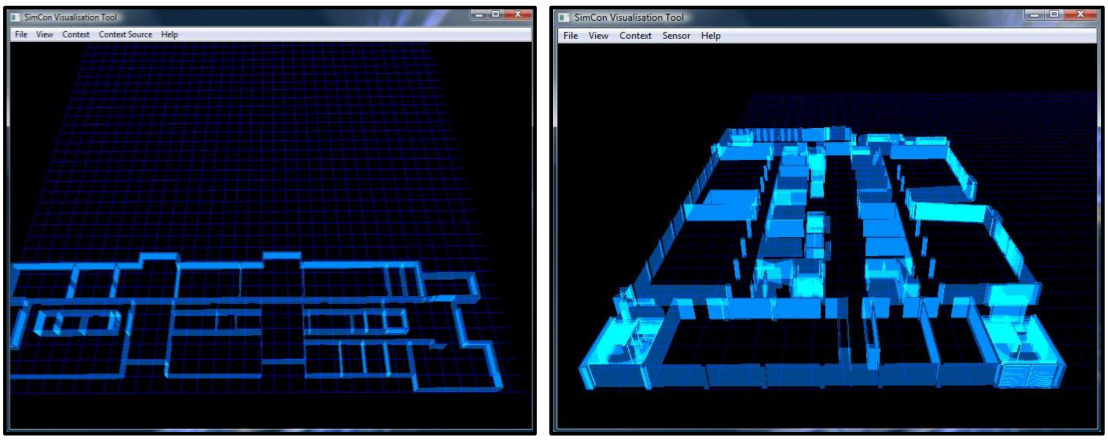


Figure 5-3 Visualisation of Building Model (2D and 2.5D): Supports placement of SimCon sources and also visualisation of contums

The frame rate of SimConViz is also 30 Hertz, i.e. the same as HL2. SWT is an open source widget toolkit for Java designed to provide efficient, portable access to the user-interface facilities of the operating systems on which it is implemented. While Swing provides many similar functions to SWT, the use of Java, the native look, feel, and performance of SWT and the use of SWT in the Eclipse platform (section 5.4.1) made it a suitable approach for supporting integration among SimCon component implementations, which are all developed in Eclipse and Java.

5.3.3 Pre-Modelling: Summary

This section detailed the implementation of the pre-requisite modelling required to enable SimCon for evaluating SBA behaviour. It detailed the modelling requirements for enabling user driven context simulation through the use of the Tatus platform which is an extension of the Half Life 2 games engine. It also detailed the modelling requirements for enabling visualisation of the building in SimConfig and SimConViz which are required for the placement of SimCon sources and visualisation of simulated context. This section does not address the issues related to the modelling requirements of external simulators as these were considered beyond the scope of this thesis. Nonetheless, the use of external models is included to demonstrate the flexibility of the context simulation approach. The next section will look at the core implementation of the SimCon model, beginning with the SimConfig tool.

5.4 The SimConfig Tool

The SimConfig tool is designed to meet both R1 and R2 by providing a usable interface to enable the creation, placement and configuration of heterogeneous user driven simulated context sources in a simulated building to support the evaluation of SBAs when faced with varying levels of uncertainty. The SimConfig tool was developed using an iterative process and underwent three versions (SimConfig V1, V2 and V3). Early implementations are discussed in more detail in the evaluation Chapter 6 (Evaluation 1 and 2). This section gives the final version SimConfig V3 but may reference aspects of those evaluations where relevant to the implementation.

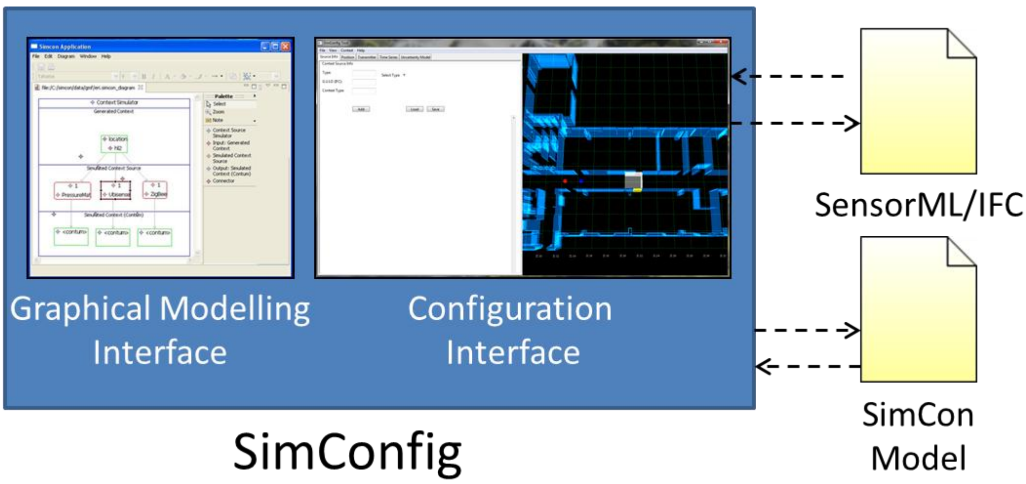


Figure 5-4 SimConfig Components: Graphical Modelling Interface and Configuration Interface

The SimConfig tool is comprised of two main components, a Graphical Modelling Interface, which supports creation of multiple SimCon sources for a building and associating SimCon sources with inputs and outputs, and Configuration Interface, which supports placement and configuration of SimCon sources (**Error! Reference source not found.**). This section begins by describing the implementation of the Graphical Modelling Interface and the rational for the choice of technologies.

5.4.1 SimConfig: Graphical Modelling Interface

To support the creation and configuration of SimCon sources, a graphical approach based on Unified Modelling Language (UML) activity diagrams proved themselves suitable to support SBA developers modelling SimCon sources as processes, each with an input and an output (see Chapter 6 Evaluation 1A). Both low and high level context sources can be modelled using this approach. This results in a flexible method for context simulation modelling. The implementation was done using Eclipse's Graphical Modelling Framework (GMF) (Eclipse 2010). This is a tool for developing graphical interfaces based on a particular domain model, for example, a Business Process Modelling Notation (BPMN) metamodel has been usedto generate a BPMN interface. Based on the SimCon model, a domain model was created which supported the modelling of SimCon sources using the activity diagram style approach. This process is easily repeated if the interface model requires extending, as new elements are introduced to the SimCon model (suiting therefore the iterative development approach taken).

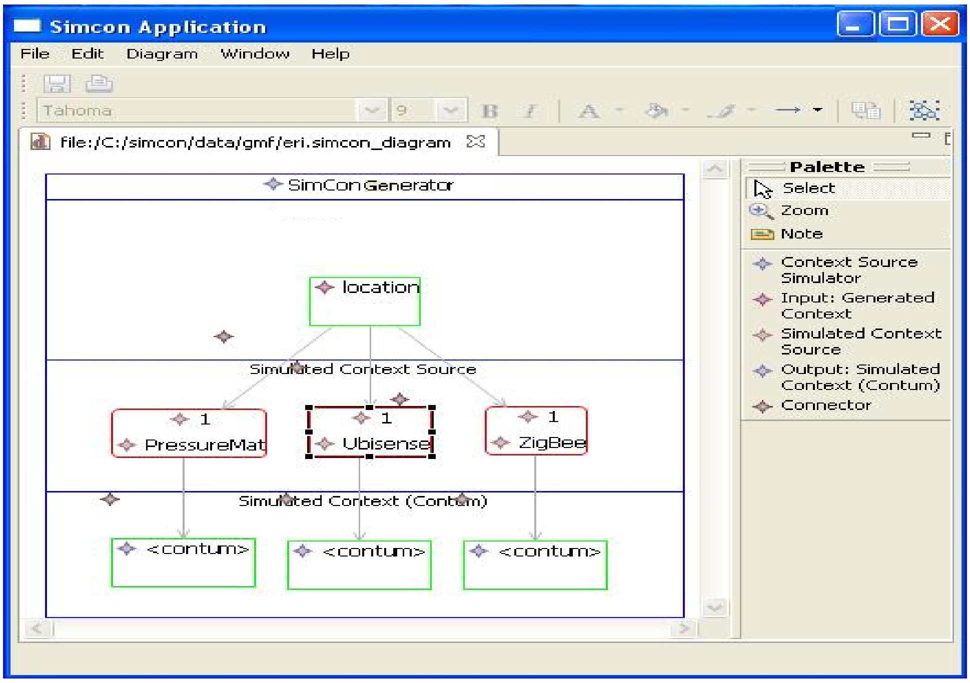


Figure 5-5 SimCon Configuration: The Graphical Modelling Tool for modelling of multiple SimCon sources as processes and which generate both low and high level context

Figure 5-5 shows the resulting SimConfig Graphical Modelling tool. The canvas on the left allows for the creation of any number of "SimCon Generators" depending on the requirements of the simulation scenario for the SBA. This contains all SimCon sources for that particular scenario which can be quickly dragged and dropped onto the canvas. Within each SimCon Generator a layered structure exists to separate inputs, SimCon sources, and outputs. As used for evaluation in the next chapter, the only input for SimCon sources has been the location data from the VR simulator, as this data is necessary to support user driven context simulation. SimConfig V3 therefore does not support manipulating SimCon sources as inputs, but this is supported in the Simcon model and may be easily included to support configuration of higher level contexts that are simulated based on low level simulation outputs. It also supports the modelling of external inputs from physical context sources using this approach, thus supporting hybrid context simulation, which is a feature found useful in number of existing context simulators (Chapter 3).

In the user interface, the top-most layer is where the input is defined, e.g. location from the VR simulator, thus corresponding to configuration of the simulated context acquisition process. The bottom layer indicates the output, e.g. the resulting contum. Additional information can be modelled with the contum “node” in the bottom layer for the purpose of testing, for example whether the data is to be written to a database (to support replays). The middle layer is where a SimCon source is created by clicking on a “SimCon Source” icon in the palette and dragging it into that layer. While the GMF interface provided a flexible and extensible graphical front end, the number of properties associated with the context simulation process proved too overcrowded for the GMF graphical layout. There was also no capability for defining pre-set values for context sources, which was identified as a method for improving the efficiency of the configuration process (Chapter 6 Evaluation 1). As a result a configuration interface (a widget) was designed and implemented based on the SimCon model. The configuration interface is accessible through a double click of the SimCon source element in the graphical interface, red bubbles in centre layer Figure 5-5, and will be described in the next section.

5.4.2 SimConfig: SimCon Source Configuration Interface

The configuration interface for SimCon sources has been developed alongside the iterative design of the SimCon model. Using the SWT and openGL interface, a range of types of SimCon sources can be placed and configured. Each of these generates a particular type of contum, e.g. presence, proximity, coordinate, temperature, and can be configured with pre-set values to have the properties of existing sensor systems as provided in their specifications or customised to meet specific requirements. The SWT interface also allowed for the creation of different tabs to configure each of the different aspects of the SimCon source, e.g. the Placement, the OutputModel, the Uncertainty Model, and also assign and configure simulated Transmitters with particular avatars in the VR engine, or with bots. Figure 5-6 and Figure 5-7 shows the configuration interface. Here the user has selected a Ubisense SimCon source. The source is automatically assigned a Global Unique Identifier (G.U.I.D.) and in this case a Ubisense SimCon source will also generate a contum of type “Coordinate” which is displayed here also.

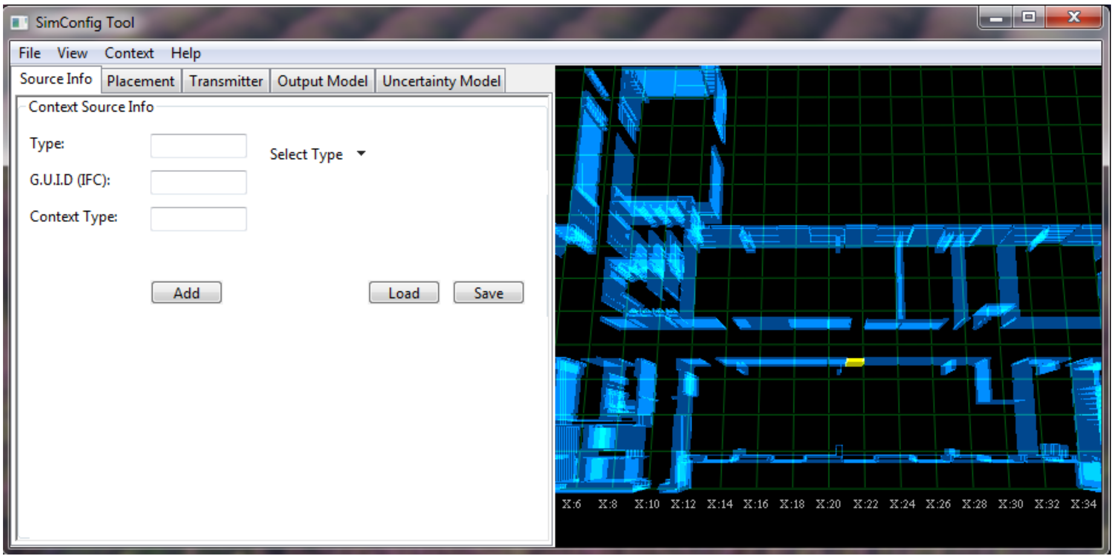


Figure 5-6 SimConfig: Configuring Source Info

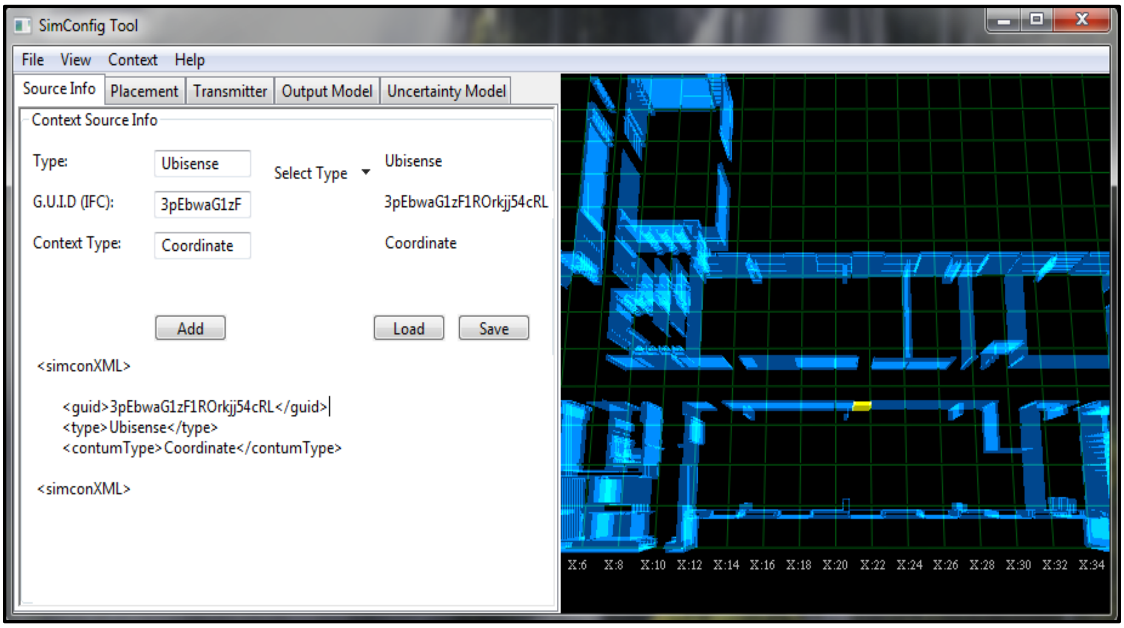


Figure 5-7 SimConfig: Configuring a Ubisense SimCon source

SimCon also provides an XML representation of the SimCon model to support editing of the values, also shown in Figure 5-7. Currently, this does not display sensorML or IFC, but SimCon V3 allows for sensorML and IFC descriptions to be imported into the SimConfig tool or exported. SensorML models were also used during a live demo (Mara, McGlinn et al. 2009) to define the properties of both real sensors and SimCon sources and will be discussed further in the section on SimConGen and Hybrid Simulation (section 5.5.7). The sensorML parser is written in Java using the JDom libraries (Hunter and Lear 2000). The IFC parser uses the OpenIFC libraries (Tulke, Tauscher et al. 2012). Currently, it only supports importing and exporting data on the sensor type, id and placement.

5.4.2.1 *SimConfig: SimCon Source Placement*

The visualisation component to support placement makes use of the 2.5D VR Building visualisation. The tool has the following features which can be accessed in a drop down menu at the top of the interface. Visualisation of:

- Building coordinate system (for SimCon source placement).
- SimCon Source type, id, bounded area and uncertainty zones.

It also provides functionality to rotate and move the building using the keyboard and mouse allowing users to zoom in and out to focus on areas of interest. Figure 5-8 shows the building at one angle and zoom, including the building coordinate system in meters. It also shows three context sources, their bounded areas and the origin of a ZigBee receiver at the centre of concentric rings in the left corner of the openGL visualisation. The concentric rings represent different values which will be generated by the simulated context source for received signal strength which will be returned by the SimCon source should a ZigBee transmitter be within one of these.

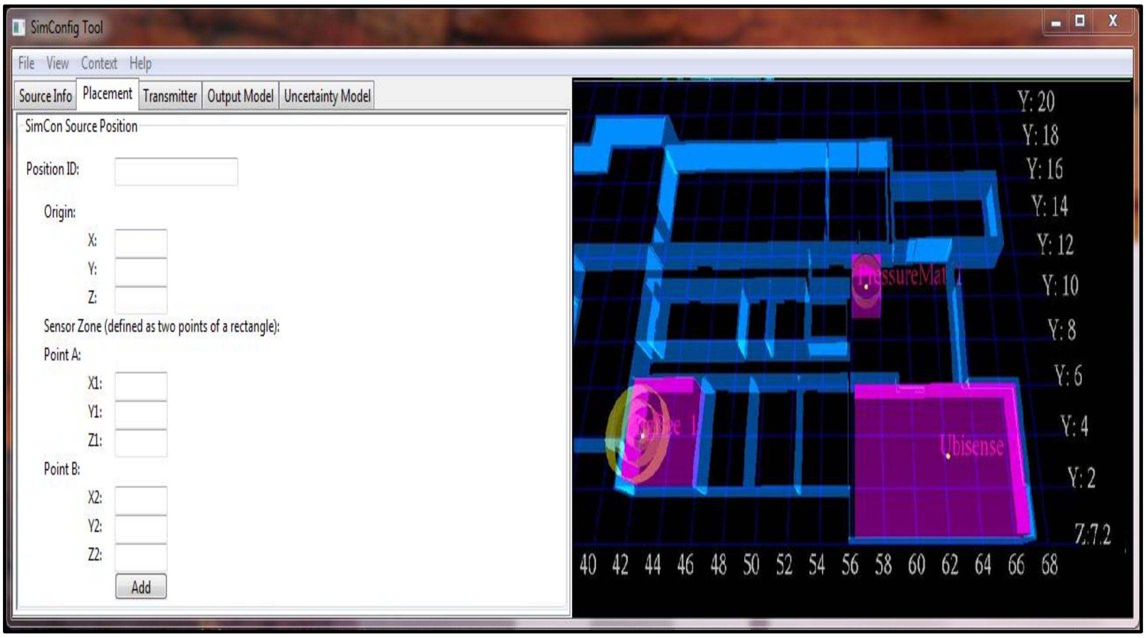


Figure 5-8: SimConfig Placement and Visualisation of SimCon sources (Type, Zone and Origin)

5.4.2.2 SimConfig: SimCon Source Output and Transmitter Model

The Output Model determines how the simulated context values are generated. SimCon currently supports a number of approaches to simulating context, which include the use of playbacks, statistical models, mathematical models and time series, user and bot driven context simulation and finally the use of external simulator models. How each of these methods is implemented is discussed in greater detail in the SimConGen section 5.5. Figure 5-9 shows a time series Output Model for temperature sensor. A time series consists of a number of values representing the time and the output vales. Here the time has been configured so that each value is generated every 15 minutes. The output value has been configured as sequentially holding 16, 17, 18, and 15 Celsius values, each for a one hour period. These values are based upon readings of actual sensors from the ERI building (ERI 2011) condensed into hourly periods in which the temperature was rising and then fell, providing a simple and easily reusable and reconfigurable approximation, that in this example was of sufficient granularity for the SBA under consideration.

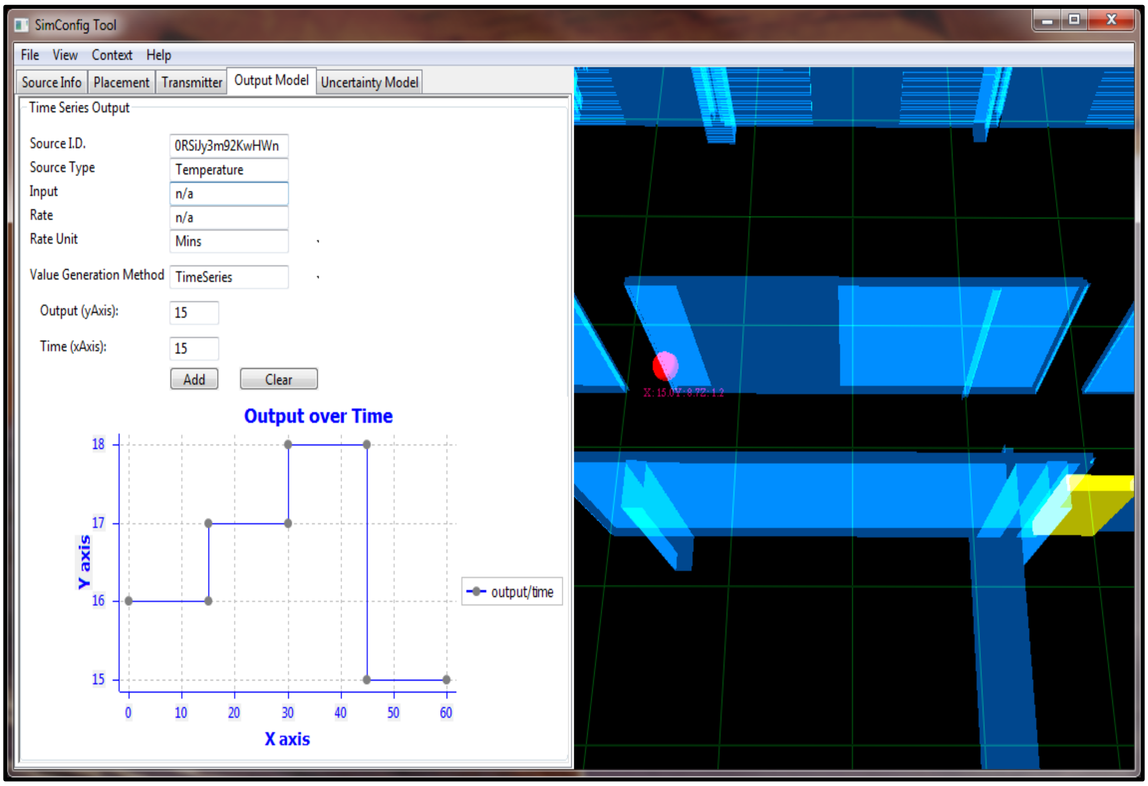


Figure 5-9 Output Model: Temperature SimCon Source Modelled as Time Series

Readings can also be based on the mean value of a period of time over days or months, for example, from Monday to Friday in an office environment. Such configurations, the granularity of the output model and approximation decisions are made at the discretion of the SBA developer. The unit of temperature is determined by the type of temperature sensor, in this case it is Celsius. A graphical display of the response values is given at the bottom of the configuration window. As there is no input indicated, this SimCon source is not user driven, i.e. the position of bots or avatars will not affect the temperature value. For SimCon sources like Ubisense, the input would be set to location, and this indicates that the Ubisense source requires location on the position of avatars in the VR environment to generate simulated coordinate values.

It should be noted that the rate for tags is defined in the transmitter model (Figure 5-10). The transmitter model also has the ability to model the output of the transmitter. For example, the signal strength can be modelled here using the formula for isotropic losses of free-space radiation addressed in the design. This type of model does not take into account propagation or transmission effects. Alternatively, the value can be set as a fixed received signal strength between two distances, for example, from 0 to 1 meters the received signal strength at the receiver will be value x. In Figure 5-10 the transmitter is a Ubisense tag and so it uses the location of the avatar in the VR building to generate the contum, and it does this every tenth of a second.

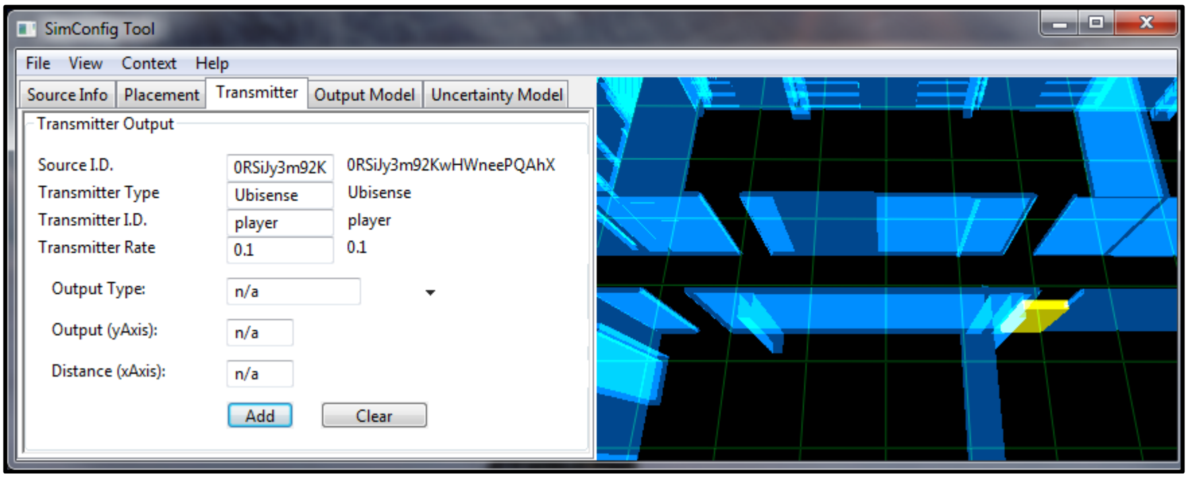


Figure 5-10 Assigning a Ubisense Transmitter to an Avatar named "player" with a transmission rate of 0.1 seconds.

The transmitter inherits the rate unit value, for example seconds or minutes, from the SimCon source itself. Now that the implemented configuration interface for the output model and transmitter has been presented, the next section will present how uncertainty is modelled for the generated value.

5.4.2.3 SimConfig: SimCon Source Uncertainty Model

Each SimCon source can have one or more uncertainty models associated with it. An uncertainty model can either be assigned to introduce a random delay in the rate, introduce variation in the generated value or be assigned to a zone for transmitters. Figure 5-11 shows an uncertainty model for a Ubisense SimCon source cell. Here the location value of the tag will be offset when in the uncertainty zone by up to one meter at one standard deviation. By applying the same model to the output of the Ubisense cell, all values within the cell will vary by according to that distribution, thereby supporting configuration. To have multiple areas providing varying uncertainty for a SimCon source multiple uncertainty zones must be defined. Each uncertainty zone model also has a unique ID which is used to associate it with a particular placement which is defined using the placement configuration screen.

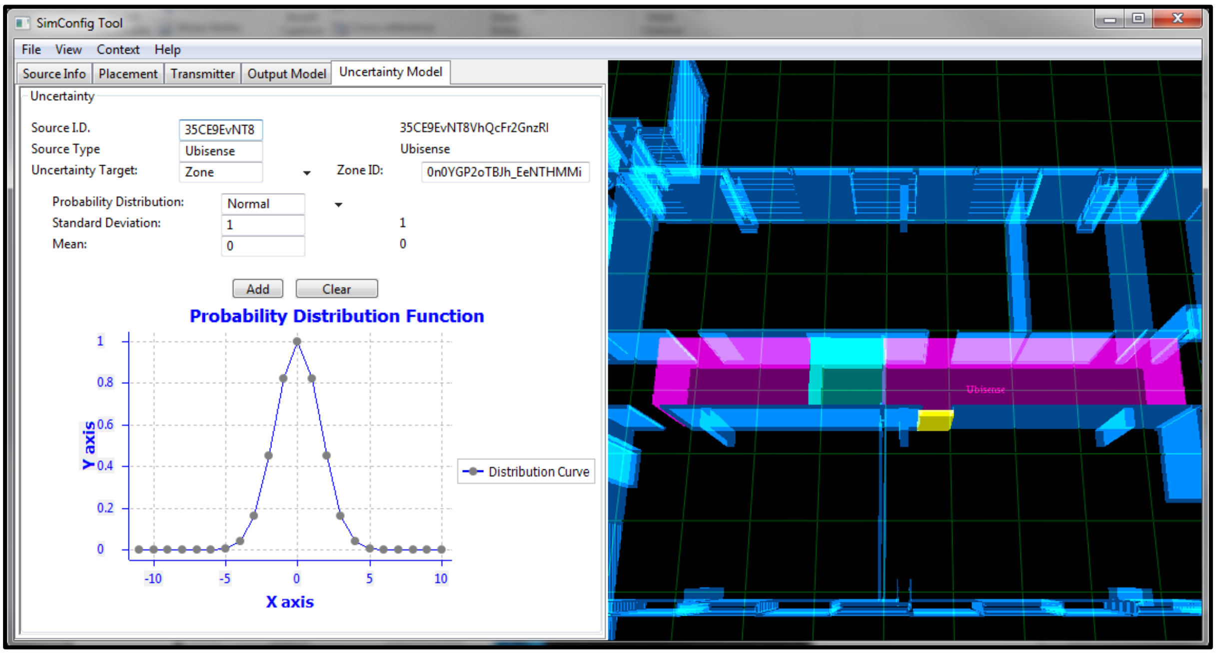


Figure 5-11 An Uncertainty Zone (green box) Model for a Ubisense cell

All uncertainty zone placements are stored in a single SimCon source model. Uncertainty zones in the implementation used for evaluation are not accumulative, i.e. overlapping zones do not result in composite uncertainty models, though the SimCon model supports this.

5.4.2.4 SimConfig: Importing and Exporting Models

The SimConfig tool widget also provides a means to import and export models into SimConfig and provide configuration through the GMF interface. The current implementation allows for sensorML descriptions to be imported into the SimConfig tool or exported, using the pop up widget. SensorML models were used during a live demo (Mara, McGlinn et al. 2009) to define the properties of both real sensors and SimCon Sources (section 5.5.7). The resulting sensorML descriptions are then saved into a database. Appendix B includes a SimConXML (Ubisense) and two sensorML (Ubisense and a Tyndall Mote) SimCon source descriptions. An export function has also been implemented which supports exporting sensor type, id and placement into an IFC file.

5.4.3 SimConfig: Summary

This section described the implementation of the SimConfig tool which consists of two components. A Graphical Modelling Interface and a Configuration Interface. The Graphical Modelling Interface supports an SBA developer to maintain a high level view of all context sources, their inputs and outputs. The Configuration Interface supports configuration and placement of SimCon sources. A description also of the importing and exporting feature of the SimCon model (SimConXML, SensorML and IFC) is also presented.

5.5 SimConGen: Simulated Context Generation

The SimConGen (simulated context generator) generates simulated contums. Of key importance to this thesis is the generation of user driven contums. This requires data provided by an external Platform named Tatus. In this section the implementation of the SimCon Generator which converts this data into user driven simulated context is described. Also described are a number of approaches which SimCon supports to simulating context, and while these are not evaluated in use, they demonstrate flexibility in the SimCon model. The section begins with showing how the SimCon Generator communicates with Tatus and collects data on each of the avatars location within the HL2 games engine.

5.5.1 Connection to Tatus

As a user moves their avatar through the VR building, xml messages are generated by the Half Life 2 games engine. Each of these records the avatars location relative to the building's coordinate system. A Java proxy had been previously implemented as part of Tatus which connects to the HL2 games server (O'Neill, Klepal et al. 2005). This is the means via which the SimCon implementation interfaces with the HL2 engine. The SimCon Generator is implemented in Java and connects to the proxy over a TCP/IP connection (Figure 5-12). The HL2 games engine updates the game state at the rendering engine frame rate which is 30 Hertz, i.e. 30 frames per second (fps). HL2 is currently configured to extract VR state data related to context simulation at a rate of approximately one per tenth of a second. Consequently no current user driven SimCon sources can have a rate which is higher than ten measurements per second. As none of the evaluations in this thesis which use user driven context simulation, which involved avatars moving at walking pace, required higher rates than this, that limit was not an issue. Migration to faster VR systems is possible, if required, as long as the VR system provides avatar location.

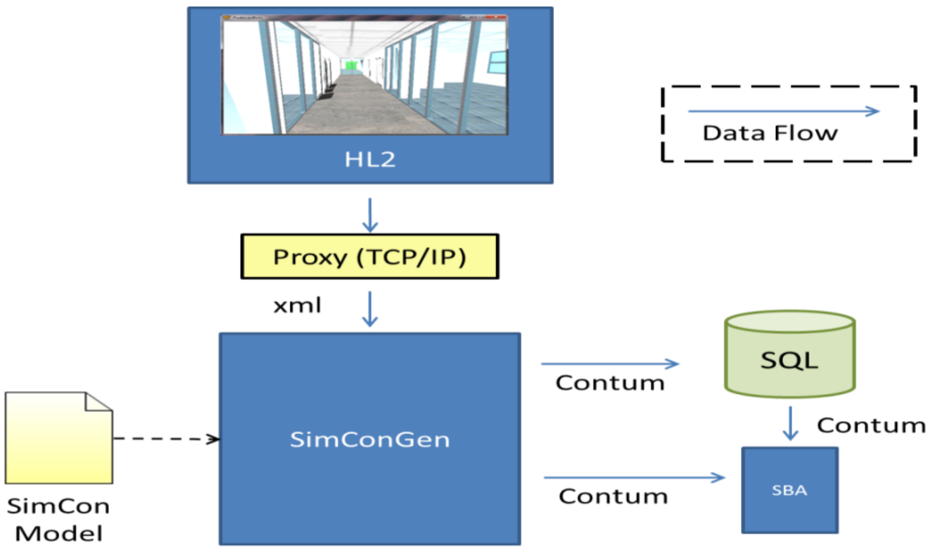


Figure 5-12 Data Communication between the VR Building, the SimCon Generator and the SBA.

The avatar has been configured to move at a maximum speed of 2.8 meters per second (walk). For the purpose of introducing uncertainty into a simulated coordinate of a transmitter; for example, a Ubisense tag, the uncertainty is applied to the transmitter as if it is static at and measurement time, i.e. it only effects the precision of the transmitter. The additional movement and the latency of the output will result in a further decrease of the accuracy of the coordinate. This "accumulated" uncertainty is not directly modelled, but results in a combination of the user movements, the precision, the measurement rate and any additional delays introduced to the rendering of the simulated context by the SimCon tools, i.e. 30 Hertz. When the SimCon sources have each been placed and configured, the SBA connects either directly to SimConGen via TCP/IP or connects to a data base which SimConGen is writing contums to, e.g. an SQL or eXist database. Table 5-1 gives an example of a Ubisense contum. It includes a tag id, a cell id, the type of contum, the type of source, the coordinate value and a timestamp.

Table 5-1: Contum (Ubisense)


```
<contum>
<type>coordinate</type>
<source>Ubisense</source>
<cellguid>2BqxkWxMfCovB_kH0rCosr</cellguid>
<guid>19$fPK0jz009xTHK7Hm7C2</guid>
<coordinate>
   <x>21.50316556043252</x>
   <y>6.84301621144843</y>
   <z>1.004022379100966</z>
</coordinate>
<timeStamp>1342024230206</timeStamp>
</contum>
```


5.5.2 SimConGen and the SimCon Model

Upon start-up the SimCon Generator creates a global clock using the Java Date object and reads in the SimCon model from where it is stored, e.g. a data base. The SimCon Generator must be restarted each time changes are made to the SimCon sources as the SimCon model is only read in then. The SimCon model is stored ether as sensorML or SimConXML and once the SimCon Generator loads it in it creates a Java array of SimCon sources. Each SimCon source has a placement, an output model and can have multiple transmitter models, each with their own output model, and multiple uncertainty models, all stored using Java arrays.

5.5.3 Non-User Driven Contum Generation

For non-user driven SimCon sources, i.e. those which the location of avatars has no effect, the rate of contum generation is fixed by the rate of the SimCon source. For example, a temperature SimCon course generates contums at a fixed rate where no uncertainty model has been created to introduce a delay to the rate. It begins generating contums either immediately upon start-up or else at a pseudo random amount of time no greater than its rate, in order to prevent all SimCon sources from having their contum generation synched with each other. When a contum is generated a time stamp is created using the global clock. The time stamp is then checked against the global clock each iteration of SimCon, e.g. every hundredth of a second, and if the current time subtracted from the global time is greater than the rate, a new contum is generated. Currently all evaluations using SimCon have been based upon user driven context simulation, and so, the issues for this type of simulation are explored in greater detail in the next section. Both temperature and humidity models have been implemented for generating non user driven context. These are based on approximately one hundred thousand sensor readings taken from the Environmental Research Institute in Cork (ERI 2011). It has been possible using these readings to create time series output models that account for changing temperature over a 24 hour period for specific days and months by using the mean values of the temperature and also to simulate the variation in these. These are once again subject to the weaknesses of this type of simulation identified in Chapter 3, i.e. that they are specific to the building in which they are taken.

5.5.4 User Driven Location Contum Generation

Once the SimCon model has been read into the SimCon Generator and a connection has been established to the proxy, the SimCon Generator listens for incoming messages from the VR building. When a message arrives in the SimCon Generator it is parsed and a Java object stores the avatarID and x, y and z coordinate. A vector of avatars is created to store how many avatars are currently within the virtual smart building. Location based contums are generated when the position of the avatar falls within the zone of a location based SimCon source. For example, a passive infrared (PIR) SimCon source will generate a presence event when an avatar walks within and through a PIR zone. The PIR SimCon source will keep triggering events as long as the avatar is moving within the zone to simulate the effect of their body disturbing the “normal” background radiation levels. If they stop moving or leave the zone, it will cease to trigger events.

For transmitter based location contums, like proximity and coordinate, a transmitter must be assigned to the avatar. SimCon maintains an array of avatars and an array of transmitters assigned to them. When the avatar location falls within a SimCon source zone in which the avatar has an appropriate transmitter assigned to them, for example a Ubisense zone and a Ubisense transmitter, the transmitter generates a Ubisense coordinate contum. This once again is assigned a timestamp, which is then checked against the current time and rate to determine if a contum for that transmitter is to be generated. As there is a tenth of a second limit imposed by the rate which the VR building generates location events all user driven location events with a rate of a tenth of a second, will synch with the tenth of a second arrival of these location events. As there are no evaluations in this thesis which look at more than one avatar carrying more than one transmitter at a time, this is not an issue. It may be an issue for scalability though where multiple users have location contums synched. Currently, no temperature SimCon sources have been implemented which use occupancy to affect the generated value but the same principle applies to this type of simulation and, once again, the state of the VR building has a limit of a tenth of a second. Appendix C section 10.3 gives examples of location based contums.

5.5.4.1 Bot Driven Location Contum Generation

Bots can be used in the same manner as avatars to generate location events. The benefit of bots is that there is no tenth of a second limit on the rate location events are generated as bots can be run at higher rates, for example, updating their position every hundredth of a second based on the Java system clock. Bots are used in Evaluation 3 to remove the non-deterministic element from the evaluation, as this was found to be an issue in Evaluation 2C. The disadvantage of this approach is that the spontaneity of real users is lost and their experience of the environment as they use an application. For determining the effectiveness of the SimCon approach to uncertainty modelling and visualisation, this non-deterministic aspect was not seen as crucial to the findings and it is assumed that the approach if effective for bots is also effective for user controlled avatars.

5.5.5 Uncertainty Generation

The SimCon Uncertainty model varies the simulated context value by a pseudo random amount using the nextGaussian() method from the Java (JRE version 6) Random object. For a value like a temperature, or received signal strength, this varies the value either positively or negatively according to the standard deviation. For a coordinate, the x, z and y have the random value applied to each axis. For output models which have multiple values, for example a time series, each value in that time series can have its own uncertainty value applied. In this case, the uncertainty model is stored as an array which corresponds with the array of values defined in the output model.

5.5.6 External Simulator Models

The current implementation of the SimCon Generator has integrated radio propagation models for simulating radio propagation. WinPlanner, which utilises IFC models as an input, is a wireless sensor design tool (Mc Gibney, Klepal et al. 2007; Antony Guinard 2009). WinPlanner uses IFC to capture the physical characteristics of the building where the network should be deployed including the number of walls, their position and material type. Based on these requirements, the design tool automatically optimises the number and the position of wireless devices to meet user defined application requirements. A key element to this design step is the propagation model (Figure 5-13).

```
<WinPlannerPropagation    floorplan="ERI_FIRST_FLOOR"    originx="0"
originy="0">
    <transmitter name="sensor_0" description="Temperature"
    x="55.51" y="10.57" z="0" frequency="2.4GHz" power="0">
        <prediction model="Motif Model"
        data="Signal Level" unit="dBmm">
            <gridpt x="17.4918" y="10.2249" z="0">-98.5791</gridpt>
```

Figure 5-13 Winplanner Radio Propagation Model

WinPlanner outputs have been integrated with SimCon to provide interactive simulation of contums which are influenced by the building structures. WinPlanner outputs the radio propagation over a grid of points within a floor of a building (Figure 5-13). Using a set of these, a SimCon Source has been implemented which detects the changing signal of a simulated dipole transmitter as the user interacts with the VR building. This has been implemented for both the Environmental Institute in Cork and also the NIMBUS building in Cork. For the NIMBUS building additional eight orientations were also modelled to facilitate changing direction, assuming it only rotates in one plane. Upon start-up SimCon loads in the radio propagation model for the building. Depending on the location of the transmitter and the orientation of the avatar, which is also extracted from HL2, the transmitter has a propagation model assigned to it and all SimCon source receivers return the grid value at their location for that transmitter at the rate of the receiver. Currently it is assumed that the transmitter rate is set to the same of that as the receiver.

A second model provided by ANSYS (Gray and Salber 2001) has been used to generate temperature values for sensors. Here a CFD model was developed for a room in the NIMBUS centre in Cork. The room was given an open door and a heat source (radiator) on the opposite corner. A series of 30 2D grids were then created as the temperature of the room changed over time due to the heat source and the open door. The temperature value at each point of the grid was then used to generate the simulated context value for a SimCon source placed in that same position.

The integration of WinPlanner and ANSYS models with SimCon demonstrates flexibility in the simulation approach. These types of models can then be used to provide values for simulated context. Uncertainty models again be applied to the SimCon sources which use these models as input, for example, to model uncertainty due to the manufacturing of a sensor.

5.5.7 Hybrid Simulation

Hybrid simulation requires integration of both simulated context from simulated context sources and context generated by physical sensor deployments as identified in the state of the art review in

Chapter 3. Hybrid simulation was implemented for SimCon during a live demonstration which took place in the Science Gallery in Trinity College Dublin (TCD). Live temperature and humidity sensor data was streamed from the ERI building in Cork via a Global Sensor Network (GSN) (Aberer, Hauswirth et al. 2006). GSN is a database software middleware which supports rapid and simple deployment of a wide range of wireless sensor network technologies. The software supports storing live sensor data in a MySQL database (Oracle 2012). It also provides a Java web interface. At this point in the implementation of SimCon no MySQL database support was implemented, and so the web based interface was used to query the last value returned by the temperature and humidity sensors. This data was then integrated with simulated coordinate location generated by SimConGen and the ERI VR building in the science gallery. This way, users could move around the VR building and using an emulated J2ME device, see in real time the temperature from sensors in different location around the ERI building. SensorML descriptions were used to describe both the SimCon Sources and physical sensor instalments (Mara, McGlinn et al. 2009). While no usability evaluation was conducted for this demo, it does demonstrate flexibility in the simulation approach.

5.5.8 Smart Building Application Integration

The SimCon Generator can be integrated directly into another Java application which can access the contums directly or, if required, write contums to a data base which is then queried by the application. The SimCon Generator has a Java package which supports contums to be written to both eXist (Meier 2012) and MySQL databases meaning that any application which can connect to the eXist or MySQL server can query the contum generation.

5.5.9 SimConGen Summary

This section detailed the implementation of the SimConGen simulated context generator. It explored the main features of SimConGen which are required to meet the objectives of this thesis. These are that it supports simulation of uncertainty in user driven context (location context) and simulation of heterogeneous types of context (coordinate, presence, proximity and temperature). It also demonstrated flexibility in the simulator by demonstrating its support for different simulation approaches, for example, by using location data from a VR environment, using inputs from physical sensors and hybrid simulation, which are all features identified in the state of the art review (Chapter 4). It also demonstrated extensibility, by incorporating external simulation models from external simulators into the context simulation approach, a feature which is currently not implemented in the state of the art with respect to context simulation. The next section will look at the SimConViz tool which is designed to support the evaluation process and meet objective R2 of this thesis.

5.6 SimConViz Tool

The SimConViz Tool provides visualisation of simulated contums. This proved itself to provide an intuitive means for SBA evaluators to quickly assess how contums "appear" to an application and also, in the case of location based contums, how uncertainty appears in location data to support evaluation of SBAs when faced with varying levels of uncertainty (Chapter 6 Evaluation 2A and Evaluation 3). Three location based contums have been implemented with visual representations. These are presence, proximity, and coordinate location. Also implemented are the visualisation of temperature, radio propagation and a CFD model of changing temperature in a room, although none of these methods have been evaluated regarding usability. The SimConViz tool has also been extended to provide visualisations for studying building performance (Keller, O'Donnell et al. 2008). In the following subsections descriptions are provided of each of these implementations. It should be noted that the SimConViz tool has also been used to display real sensor outputs as part of the Hybrid simulation discussed in section 5.5.7 . The SimConViz tool is built on top of the 2.5D VR Building visualisation. It requires access to both the SimCon model and incoming contums. This is necessary because contums do not contain any information on the sensors placement and so the SimCon model is required to associate the contum with its SimCon source, for example a presence event with a PIR, or proximity with a receiver.

5.6.1 Location Visualisation

In this section we describe the visualisation of three types of location context, coordinate, presence and proximity. Figure 5-14 shows a Proximity Transceiver. A blue avatar represents the avatar's position in the VR building within the yellow concentric ring. The yellow circles at the centre of the yellow concentric rings represent the origin of the SimCon Source receiver.

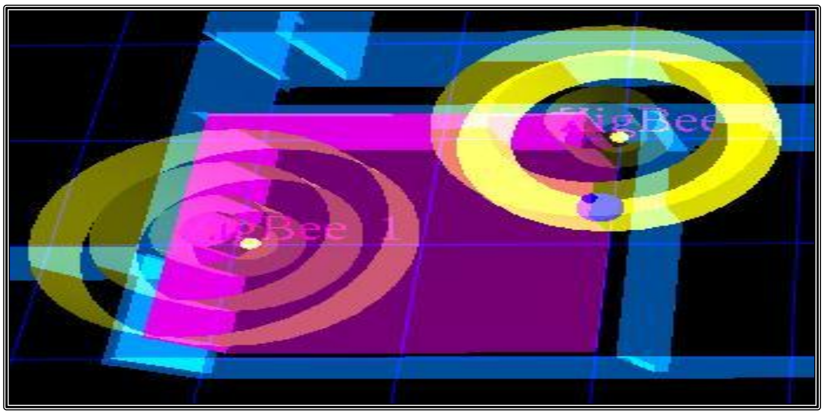


Figure 5-14 SimConViz using sensorML context source description to visualise a proximity contum

The concentric rings represent the proximity contum being generated so that, depending on the received signal strength between the transmitter and receiver, the proximity around the origin of the receiver is revealed. A similar method is used to represent on/off trigger events as a circle within the presence zone, so that when a presence event is triggered a circle appears in the context source zone until the trigger event times out, i.e. the point at which it no longer indicates presence. In this example we also show visualisation of the two context sources and their zones which are both placed in a room in the VR building of the NIMBUS Centre. This type of visualisation is useful for evaluators when they wish to quickly determine how the system views the building occupant's location. In the example in Figure 5-14 it becomes immediately apparent that in this case, proximity data may indicate the user is in either the hall, or one of the rooms. This visualisation is used in Evaluation 2B. In Figure 5-15 we also see the visualisation of a presence event, the shaded yellow sphere at the centre of the right box.

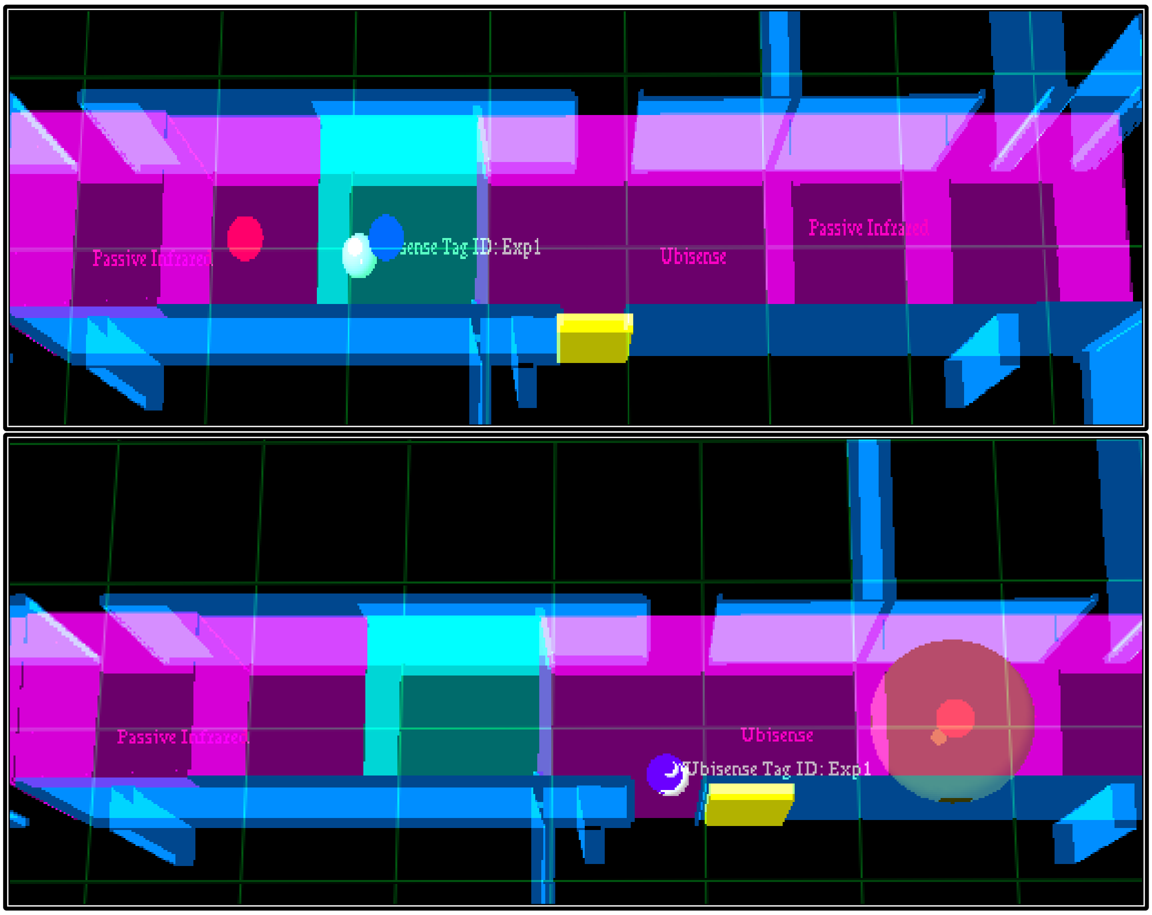


Figure 5-15 SimConViz: Visualisation of tagged and non-tagged avatars, and location and presence contums.

This is highlighted when either the blue (tagged) or red (non-tagged) avatar moves through a passive infrared zone. To represent coordinate location (i.e. the location of the tag), a green avatar is used (Figure 5-15). When there is an uncertainty model the position of the tag will be offset from the position of the user (green and blue representation), the visualisation tool provides visual feedback on the level of uncertainty by the amount the avatar moves around, i.e. the avatar is more "jumpy" as the level of imprecision is increased. The visualisation of coordinate location represented by an avatar has been shown to quickly provide evaluators with an intuitive representation of imprecision in location context by highlighting the effect of jumpy erratic contums on the systems view of the building in both Evaluations 2A and Evaluation 3. In Figure 5-15 there is also the visualisation of a door which is automated. Here the door opens when the tagged user is in front of it and neither one nor the other passive infrared SImCon source has been triggered by a non-tagged user. For more on the explanation of this SBA, see Evaluation 3 Chapter 6.

5.6.2 CFD and Temperature Context Visualisation

The ANSYS CFD model was used to visualise the changing temperature within a section of a room in the NIMBUS building in Cork. SimConViz updated the visualisation of the temperature at each time interval of the CFD time series being generated by SimConGen. As the heat source, Figure 5-16 centre left, gets hotter, heat flows to the open door and around the room. Reds and yellows are used to display higher temperatures, blues and greens lower temperatures.

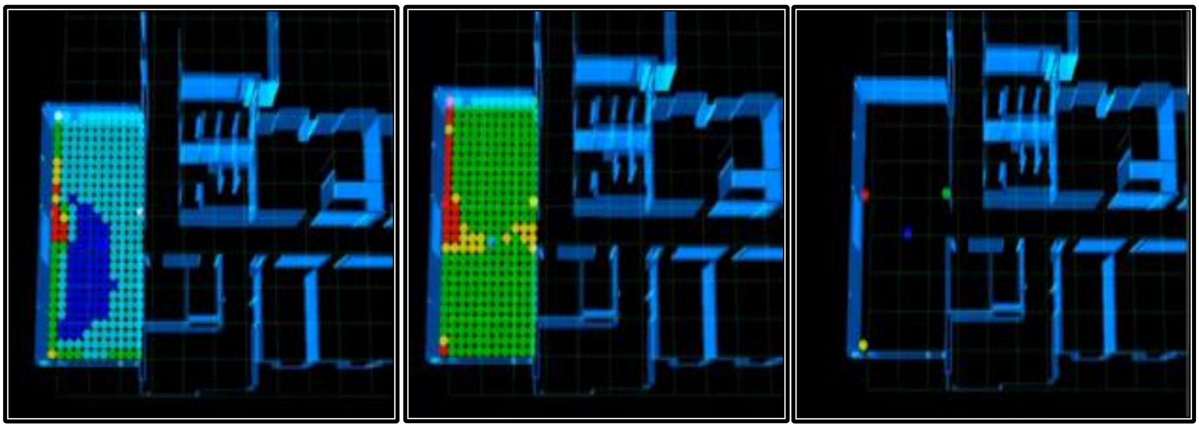


Figure 5-16 External Models for Temperature Simulation, and temperature at point for a room in the NIMBUS centre in Cork. Left centre a heat source has been defined, and an exit right centre an open door.

Temperature SimCon sources placed in any of these grid points will produce a temperature at the rate set by the SimCon source, so potentially, the visualisation of the temperature at the source may be different than the simulated temperature at that location because it only displays the last updated temperature. A visualisation of temperature was also used in the Science Gallery demo in Dublijn so that when a tagged avatar was in proximity to a simulated SimCon temperature source the temperature from the equivalent sensor in the physical ERI building was displayed on an emulated mobile application. Figure 5-17 shows the SimConViz tool displaying the temperatures. In this example, the higher temperatures were indicated by a bright orange and red. Lower temperatures were indicated by less intense orange/reds with decreased opacity. This use case was designed for a Smart Building manager to monitor the occupancy and temperature of the ERI building, but has not been evaluated in use. It is presented here to demonstrate the flexibility and extensibility of the SimCon model.

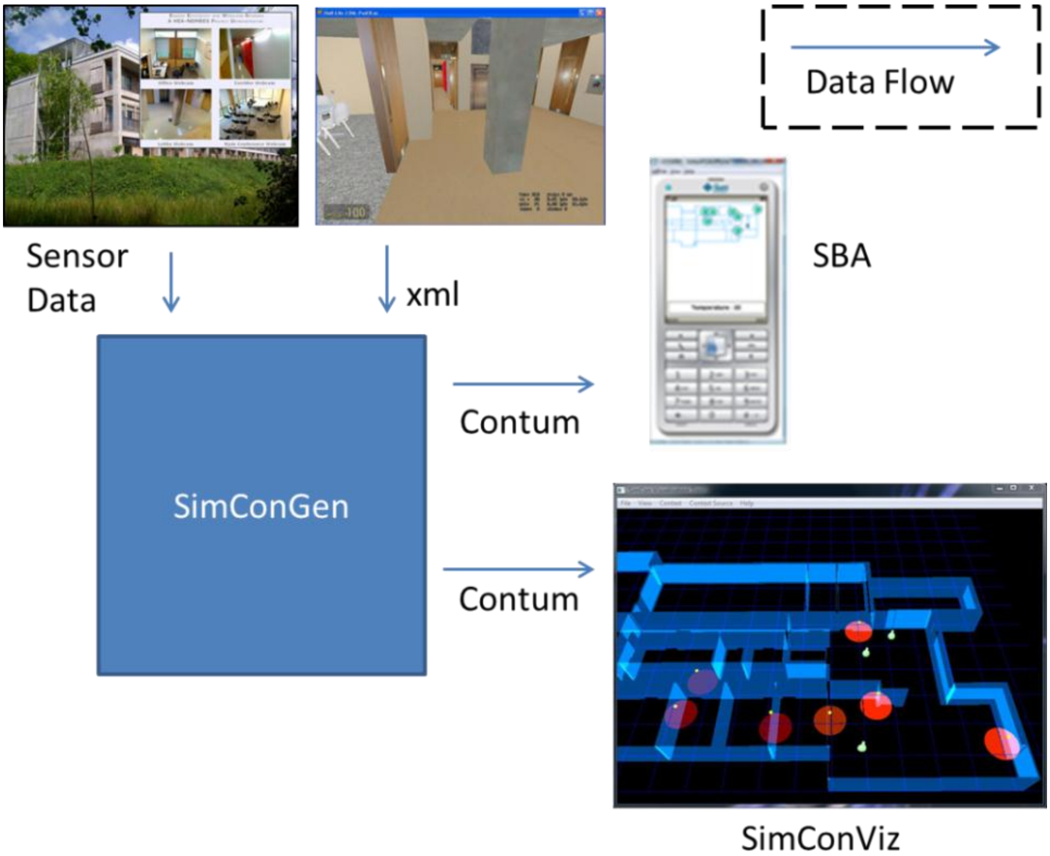


Figure 5-17 Displaying Hybrid simulation, simulated coordinate and real temperature using SimConGen and SimConViz.

5.6.3 Radio Propagation Visualisation

The integration of radio propagation models with the SimCon Generator also demonstrates flexibility in the simulation model. The SimConViz tool was extended to display the changing radio propagation of a transmitter as an avatar moved and changed orientation around a section of the NIMBUS building in Cork. This use case examined visualisation determining best placement of sensors when combining two types of simulated context generated by external simulation models. When a tagged user is in proximity to three SimCon sources, a heat source is actuated. The changing signal strength of the transmitter can be seen in Figure 5-16 as the user moves the avatar around the room. By integrating two types of simulation using SimCon, an SBA developer can quickly overlay the requirements of correct placement to determine when a user is in the room, and best placement for determining the mean temperature of the room.

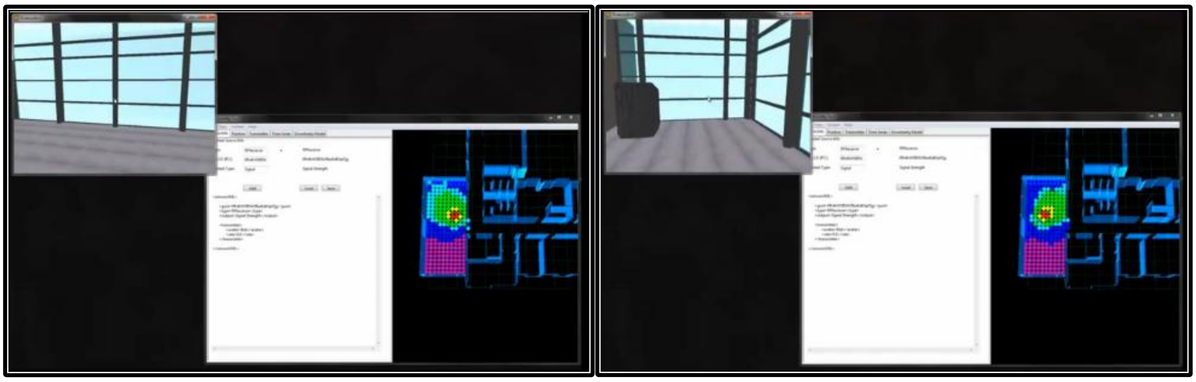


Figure 5-18 External Models for user driven RSS Simulation

The original implementation to support the grid based display of the external simulation models required the development some additional code. Once this code was implemented the ability to extend into more dimensions, or increase or decrease the granularity is a relatively trivial task for an openGL developer. However, performance is expected to become an issue when displaying larger numbers of grid points. As this does not relate directly to the main research question, this evaluation has not been addressed. This work is presented to demonstrate flexibility in the SimCon model.

5.6.4 Performance Metric Visualisation

The final use case presented here for SimConViz is for displaying performance metrics for zones (Keller, O'Donnell et al. 2008). A performance metric defines value or ranges of values for example, the energy consumption of a zone or building. In this example, a traffic light system was implemented to display when a zone was falling above or below acceptable temperature levels to maintain comfort. Replays of sensor data from the ERI building were used to generate the simulated context values for each SimCon source.

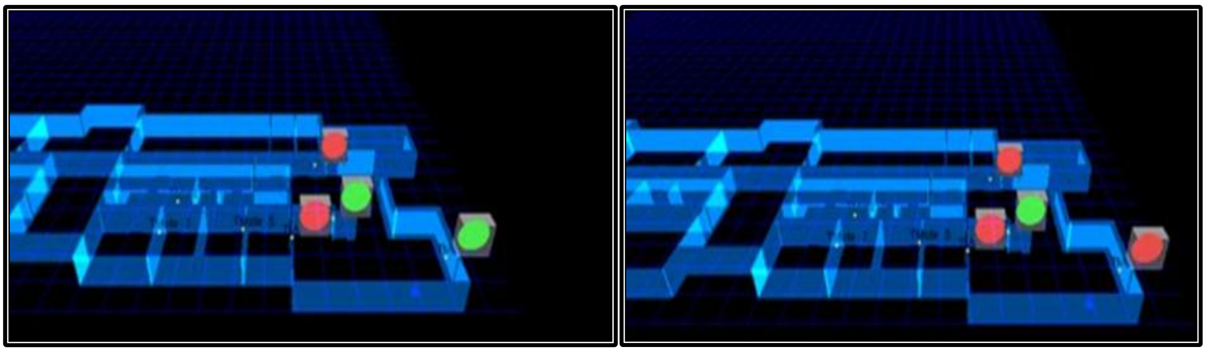


Figure 5-19 Visualising Performance Metrics for determining whether temperature is falling bove or below acceptable levels.

Again, this method has not been evaluated to determine whether it meets the requirements of SBA developers, but is presented here to demonstrate flexibility in the SimCon model. This type of implementation is relatively quick as the underlying code base already has the zones, source placement and other details required to associate the traffic lights. All that is required is for the geometric shapes to be added and changed depending on the values of the generate contums.

5.6.5 Summary SimConViz Tool

This section presented the implementation of SimConViz which is built upon openGL libraries. SimConViz has demonstrated itself to be a flexible and extensible method for supporting different types of visualisation of simulated context generated by SimConGen. A number of these approaches are presented here as use cases. The visualisation of uncertainty in coordinate based contums is evaluated in detail in Chapter 6 Evaluation 3, and to a lesser extent Evaluation 2A.

5.7 Chapter Summary and Conclusion

This chapter presented the implementation of the SimCon model. This consists of three components, the SimConfig tool for creating, placing and configuring simulated context sources, SimConGen for generating simulated context, and SimConViz for visualising context to support the evaluation of the SBA. The chapter began by presenting the pre-modelling requirements for enabling evaluation using SimCon. That is, modelling of the building and the Virtual Reality environment. It also briefly discussed some of the pre-modelling for external simulators, but as it was considered beyond the scope of the thesis question, is not explored in detail. Next the implementation of the SimConfig was presented. SimConfig consists of two interface, the Graphical Modelling Interface for maintain a high level view of all context sources, their inputs and outputs and the Configuration Interface, which supports configuration and placement of SimCon sources. A description also of the importing and exporting feature of the SimCon model (SimConXML, SensorML and IFC) was also presented.

The following section presented the implementation of SimConGen for generating simulated context with additional uncertainty. This included the implementation of the Tatus platform which provides SimConGen with data on avatar locations and how this ocation is used to generate user driven context simulation. It also described the implementation of uncertainty generation and how this is introduced into the process. Finally, it discussed the use of external simulators to drive context simulation and the use of hybrid simulations, which integrate real and simulated context.

The last section presented the SimConViz tool, which supports the visualisation of simulated (and real) context. A number of use cases were described to demonstrate flexibility and extensibility in the SimCon model. Reference was also made to aspects of the visualisation which have been evaluated in the next Chapter.

6 SimCon Evaluation

6.1 Introduction

This chapter describes the evaluation of the usability of the SimCon model. The usability evaluation is broken into three parts, with the first and second having two and three formative sub evaluations respectively and the third concluding in one final summative evaluation of the SimCon model. The three evaluations end with a summary and a conclusion, culminating in a final summary and conclusion for all evaluations at the end of the chapter. The chapter sets out to address the evaluation of the SimCon model as highlighted in research Objective 3 Chapter 1, the design and implementation of which have been presented in Chapter 4 and Chapter 5. These research objectives which were derived from the research question are reiterated here:

1. Conduct a state of the art review of current approaches to supporting smart building application developers evaluate smart building applications when faced with varying levels of uncertainty in context, through the use of simulation. Identify the key requirements of the proposed solution through analysis of the strengths and weaknesses of current approaches.

2. Design a context simulation model which supports the usable creation and configuration of simulated context sources that generate user driven simulated context and which support usable evaluation of SBA's when faced with varying levels of uncertainty in context, thus supporting the following key requirements which were identified in Chapter 4.

    - **Requirement R1**: The context simulation model should be flexible so as to support different context simulation approaches to simulating heterogeneous context source types which generate heterogeneous types of context and extensible so as to support context simulation with additional uncertainty.

    - **Requirement R2**: The context simulation model should be usable by SBA developers to achieve the goal of evaluating SBAs which must deal with varying levels of uncertainty in context. This requires that the model be both usable during configuration and also during the evaluation process by providing tools to support SBA analysis.

- **Requirement R3**: The context simulation model should be interoperable with existing standards in building and sensor modelling.

3. Implement and evaluate the resulting model to determine if it has met the key requirement of being usable by SBA developers during configuration of user driven simulated context sources and during the evaluation of SBA's when faced with varying levels of uncertainty. The usability evaluation is further broken down into four sub-objectives. These are that the context simulation model supports:

    - SO1: Usable configuration of uncertainty in user driven simulated context sources.
    - SO2: Evaluation of SBA through simulation and visualisation of uncertainty in context.
    - SO3: Usable configuration of user driven heterogeneous context sources.
    - SO4: Evaluation of SBA through simulation and visualisation of heterogeneous contexts.

The following sections begin with an explanation of the experimental methodology used to structure the usability evaluations in this chapter, including information on how the number of participants was chosen and the metrics recorded. This is followed by an overview of the evaluations before the evaluations themselves are presented.

6.2 Usability Evaluation: SimCon Model

6.2.1 Methodology and Metrics

Chapter 2 introduced the concept of formative and summative evaluations. Formative evaluations take place early in the development of a product and tend to rely on qualitative data, whereas a summative evaluation takes place towards the end of the development stage and looks at generating quantitative data to support claims about usability. The usability evaluations in this chapter shall take this approach by applying formative evaluations in the early and middle stages of development culminating in a final summative evaluation. The Common Industry Format was also introduced in Chapter 2 as a method for structuring usability evaluations. To comply with the CIF standard a usability report must include the following information (Bevan 1999):

- A description of the product/model.
- The goals of the test.
- Context of Use
    - The test participants and background.
    - The tasks the participants were asked to perform.
- The method or process by which the test was conducted.
- The experimental design of the test.
- The usability measures and data collected.
- Numerical results and analysis.

The usability evaluations in this chapter shall adhere to this structure.

6.2.2 Goals

For each experiment, whether formative or summative, the goals will be set out at the beginning. This includes the research question which will be addressed by the evaluation, as well a null and alternative hypothesis. These will guide the reader as to the objective of the experiment and whether the experiment has met that objective, i.e. to answer a particular research question. The formative evaluations may have a null and alternative hypothesis which is less specific than a summative evaluation, as they rely less on qualitative results.

6.2.3 Metrics

Formative and summative evaluations generally set out to measure different metrics. Sauro and Kindlund have created a quantitative model of usability based upon the ISO 9241 standard, which has resulted in four metrics. These are time to complete tasks, number of errors, whether a task is completed and the average satisfaction of users Figure 6-1 (Sauro and Kindlund 2005). These are similar to those defined by Nielsen (Nielsen 1993), which are: Success rate (whether users can complete a task), task time (time a task requires), error rate (number of errors per task), and user satisfaction. Measures of effectiveness have a strong correlation to the types of tasks. In the case of SimCon, these are related to both the ability of SBA developers to place and configure SimCon sources, as well how SimCon supports them during the task of evaluating an SBA when faced with varying levels of uncertainty. The evaluations of SimCon are designed so that the participants can complete the tasks with no help from an instructor. Errors are therefore classified as significant if the participant must ask for assistance from the instructor. Efficiency is measured by how long it takes to complete a task. This can be achieved through the observation of the participant. When they complete a task, they must move onto the next task. This time is recorded.

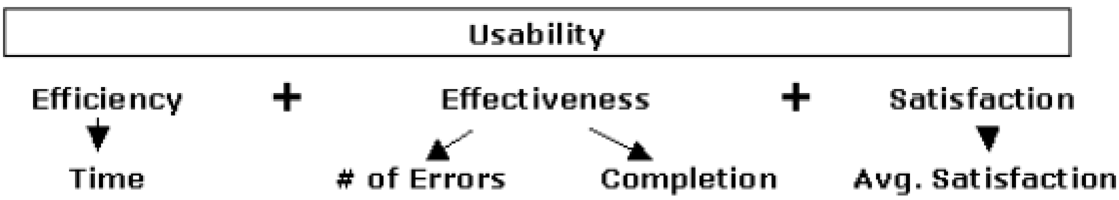


Figure 6-1. The ISO definition of Usability Converted into Quantifiable Metrics

User satisfaction is assessed through the use of questionnaires. A number of questionnaires are available to assess usability, for example the System Usability Scale (SUS) (Brooke 1996), the Ease of use (USE) questionnaire (Lund 2001), the Computer System Usability Questionnaire (CSUQ), the Questionnaire for User Interaction Satisfaction (QUIS) (Chin, Diehl et al. 1988) and the Software Usability Measurement Inventory (SUMI)(Kirakowski and Corbett 1993). Some of these, for example USE and CSUQ, suffer from bias in the form of positive phrasing, i.e. all the questions are positive, for example “It helps me be more effective” “It is easy to use”. Tullis and Stetson have evaluated SUS, CSUQ and QUIS and concluded that even though SUS is one of the more simple approaches, it resulted in the most reliable results across sample sizes (Tullis and Stetson 2004). They also note that when evaluating only one design, the most important information from the usability questionnaires is related to its diagnostic value with respect to improving the design. The SUS is used in this context for the formative evaluations.

SUS is a simple ten item scale giving a global view of subjective assessment of usability (Brooke 1996). It is a Likert scale in which the participant must choose between five or seven points ranging from agreement to disagreement on a particular statement. The statements in SUS are chosen to identify extreme expressions or attitudes and are listed in Appendix F 14.4. SUS also provides a point structure to assign to the answers of a particular test which rates overall satisfaction between zero and a hundred. Bangor and Kortum have provided a rating for these points (Figure 6-2) (Bangor, Kortum et al. 2009). They warn though that the rating of OK can be misleading and should not be considered satisfactory. They suggest that products with a score in the seventies should be deemed acceptable, and those below seventy still have usability issues which are cause for concern.

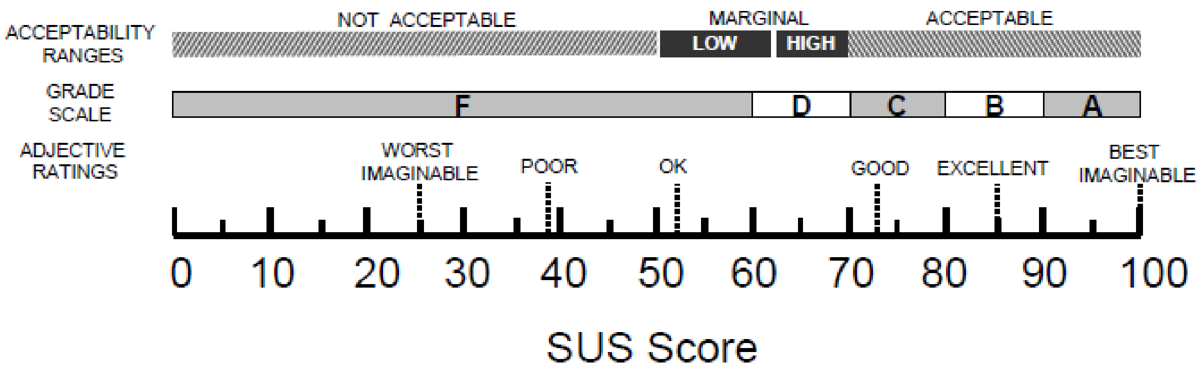


Figure 6-2 A comparison of the adjective ratings, acceptability scores, and school grading scales, in relation to the average SUS score (Bangor, Kortum et al. 2009)

Finally, while summative evaluations rely on quantitative metrics, these metrics are not exclusive to summative evaluations and may be incorporated into formative evaluation. For instance, task times can be a valuable diagnostic measure during formative-type tests. Now that the metrics to be taken have been explored, the next section will look at the selection of participants.

6.2.3.1 Number of Participants

Another question regarding usability is what number of participants is required for an effective evaluation of an implemented model. Nielsen and Landauer have created a mathematical model to determine the number of users needed for a usability evaluation (Nielsen and Landauer 1993). Their focus was on user testing and heuristic evaluation of interfaces. Heuristic evaluation requires experts in a field to sit and judge a model using established usability principles, for example those given by Nielsen (Nielsen 1994). Usability testing requires real users being observed in real time using the model. The mathematical model they use is a simple Poisson distribution. It is based on the assumption that the probability of any participant discovering a problem is independent of the outcome of a previous test. For user testing the discovery of a problem relies on, a: the participant experiencing the problem, and b: the test administrator realizing an issue has occurred. Unlike the participant, the test administrator's own understanding of the issues is increasing with each evaluation, i.e. they may begin to recognise a particular issue only after observing it multiple times. This potential issue can be ameliorated by carefully considering these possibilities when developing the experiment. Nonetheless, Nielsen and Landauer assert that it would seem likely that the experimenter's findings of issues can also be approximated by a Poisson distribution.

During evaluation failures which result in the complete cessation of the product will be quick to identify as the participant will no longer be able to interact with it. Other less significant problems will be harder to discover. They assert therefore that the number of usability problems found in a usability test with *i* participants are:

Equation 3

$$Found(i) = N(1-(1-L)^{n})$$

Based on this model, Nielsen and Landauer have argued that a user evaluation with only five users will be enough to identify between 50 and 85% of the problems (Nielsen and Landauer 1993). This is based on the assumption that the probability of discovering each problem L is 0.31, and they acknowledge that this is not always the case. Woolrych and Cockton have pointed out that to account for this, the value L should itself be based on a probability density function (PDF) that recognises variability in the probability of discovering a problem, as in situations where L is very low. The number of users will consequently need to be raised to discover the same number of errors as situations where L is a higher value. Woolrych and Cockton also point out that in heuristic evaluations experts differ in their ability to discover errors. In the worst case scenario where the experts selected for an experiment are all “bad” problem finders, the numbers of participant needed to discover errors consequentially increases.

The individual differences between non-expert users, their interaction with the tool under test, and the complexity of the tasks are also key variables which are neglected by Nielsen and Landaeur (Woolrych and Cockton 2001). Again, to account for these, they recommend that L should be replaced with a PDF parameterised by values that represent beliefs about the differences of the likely impact of the users, the task and the tool under test. Nielsen and Landaeur state that tests of five users should not be done in isolation but rather as part of an iterative evaluation cycle. Their reason for choosing five is related to cost/benefit issues. While the exact number of users to find all potential problems may vary according to the users, the tasks, and the system under test, they claim it is still likely a better strategy, when dealing with constrained time and resources, to evaluate initial evaluations of a design less thoroughly. This will be reflected in the number of users in earlier evaluations. This is the approach taken when conducting usability evaluations in the next sections.

In conclusion, this section identified a method for producing reports and metrics on the usability of a model which can be applied to the field of context simulator evaluation. It identified a set of metrics which can be used to evaluate the usability of the system quantitatively and these will be applied to the evaluation of the usability of the SimCon model. Finally, it identified the use of iterative design and evaluation cycles with small groups of users as an approach to discovering early errors in a model as part of an iterative design cycle, culminating in a larger summative evaluation.

6.2.4 Overview of Usability Evaluations

The SimCon model has undergone three stages of evaluation during its development. The SimConfig tool was developed to evaluate the usability of the SimCon model when placing and configuring SimCon sources. To evaluate the usability of the SimCon model when evaluating an SBA, SimConGen and SimConViz were developed, which provide simulated context and simulated context visualisation respectively. While each evaluation examines usability of the SimCon model in terms of effectiveness, efficiency and satisfaction, specific features related to the four sub objective outlined in objective 3 are presented in the overview in Figure 6-3. The first two evaluations, named Evaluation 1 and Evaluation 2, were formative evaluations. These took a structured approach based on the CIF and included quantitative metrics like time to complete tasks, significant errors and qualitative metrics which consisted of feedback during the evaluations and questions which measured their satisfaction with the approach. Evaluation 1 is divided into Evaluation 1A and 1B and both addressed part of the design requirement R2, to create a usable model for supporting the configuration of simulated context sources which includes the configuration of uncertainty.

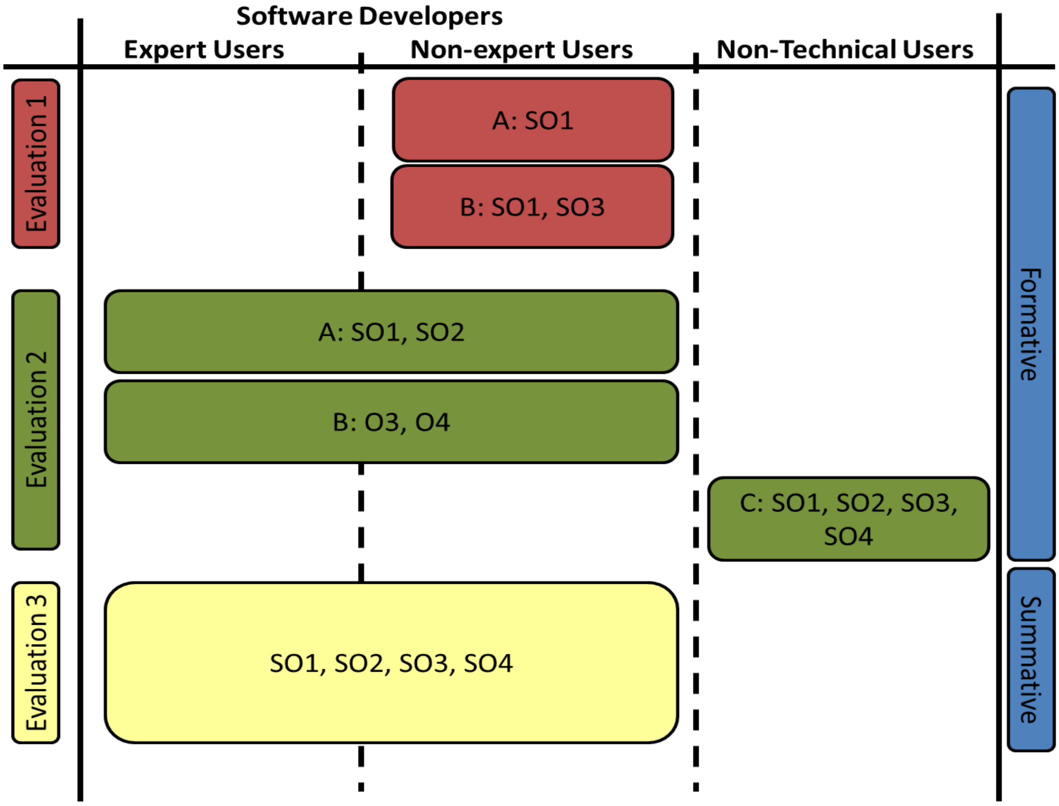


Figure 6-3 Overview Evaluations and Objectives

Evaluation 1A consisted of an initial prototype which explored two approaches for modelling context simulation and provided early qualitative feedback on the use of the SimCon model for this purpose. Evaluation 1B was an evaluation of version 1 of the model, and measured usability issues (time to complete tasks, significant errors, user satisfaction) for SBA developers when placing and configuring two different types of location SimCon sources. Due to a lack of existing studies in the area of context simulation, it is not possible to quantify a baseline time for setting up and configuring a context simulator to evaluate efficiency. Evaluation 1 therefore set out to establish a baseline by which additional evaluations can be compared, for example, by determining the time taken to configure a simulated context source.

Evaluation 1 was also crucial for determining the type of interface that would be required to test the SimCon model. No evaluation was conducted during Evaluation 1 to determine whether the SimCon model supported the evaluation of an actual SBA. It should be noted, that due to the small numbers of participants in Evaluation 1 (six and five) a four point scale was used in post questionnaires. This was done to avoid the central tendency bias which can lead to participants favouring neutral scores, as it was important to ascertain positive and negative attitudes in these evaluations. In later evaluations a more structured approach to ascertaining usability was taken based upon the System Usability Scale (SUS) described earlier and which allows users to give neutral answers. The SUS also removed, as described previously, any perceived bias that may have been evident in the earlier formative evaluation questionnaires. Findings from Evaluation 1 resulted in the development of version 2 of the SimCon model.

SimCon version 2 set out to improve upon SimCon version 1 by providing improved support for rapid configuration of simulated context sources as well as support for evaluating SBAs. Evaluation 2 addressed all of R2 and so examined usability issues of the SimConfig tool (time to complete tasks, significant errors and user satisfaction) for SBA developers and also whether the SimCon model supports them in evaluating an SBA. This was qualitatively evaluated through feedback from the users both as comments and through specific questions with respect to SimConGen and SimConViz. Evaluation 2 is divided into three evaluations, Evaluation 2A, 2B and 2C.

Evaluation 2A examined users of the SimCon model for its intended purpose, the evaluation of SBA design when dealing with varying levels of uncertainty in context. Evaluation 2A looked at a greater range of users than previous evaluations, ranging from novice to expert, and also set out to determine the effectiveness of the SimCon model to support the evaluation of an SBA when dealing with uncertainty in location context. Evaluation 2B was a smaller evaluation to gain feedback from a group of users (three) with backgrounds in smart building systems. The group comprised of two civil engineers with experience working on location aware systems and one expert in the area of simulation of indoor wireless sensor communication. They were tasked with placing and configuring three heterogeneous types of context source and using context visualisation to evaluate a hypothetical security application. Evaluation 2C looked at usability issues of the interface of SimCon for users with non-technical backgrounds when using SimCon to evaluate a hypothetical security system. Evaluation 2 was essential in identifying aspects of the SimCon model which could likely cause consistent issues with SBA developers and the feedback provided by the users, while often qualitative, was essential for the development of SimCon version 3.

Evaluation 3 evaluated R2 fully, namely the usability of the SimCon model when creating and configuring simulated context sources with additional uncertainty, and the use of simulation and visualisation of context to support the evaluation of an SBA. It examined typical SBA developers, ranging from users with no-experience in the area of SBA design and evaluation to expert users. Users were classified according to their skills and where two users had roughly equal skills, they were then randomly grouped and assigned to either the control or test group. The control and test group were then given a set of tasks to complete, with the test group having additional access to visualisation of the simulated context.

Table 6-1 Overview of Usability Evaluations

	Evaluation 1A	Evaluation 1B	Evaluation 2A	Evaluation 2B	Evaluation 2C	Evaluation 3
Type of Evaluation	Formative Comparative Usability Evaluation SimConfig Prototype	Formative Initial Usability Evaluation of SimConfig Version 1	Formative Usability Evaluation of SimCon Version 2	Formative Usability Evaluation of SimConfig Tool Usability	Formative Usability Evaluation of SimCon Version 2	Summative Comparative Usability Evaluation of SimCon Version 3
Goal	To investigate two graphical interfaces to support the modelling of contums in order to determine which provides greater efficiency and user satisfaction	To investigate usability of the SimConfig Tool when creating, placing and configuring two heterogeneous SimCon sources for creating user driven location-based contums	To investigate usability of the SimCon model when creating, placing and configuring a SimCon source to support participants evaluating the effect of varying accuracy in simulated location on a location-aware application.	To investigate usability of the SimConfig and SimConViz Tool when creating, placing and configuring three heterogeneous SimCon sources for creating location-based contums in order to evaluate a hypothetical security system.	To determine usability issues of SimCon and use of VR environment with non-technical users when evaluating a hypothetical SBA.	To investigate usability of SimCon for SBA developers who must evaluate the placement of two context sources to enable a smart building application which relies on two heterogeneous types of context, one of which is subject to varying levels of uncertainty
Evaluation Part-icipants	6 non-expert users	5 participants. 3 non-expert, 2 with background in pervasive computing. 1 with experience using sensor systems.	11 participants. 7 non-experts, 3 intermediate, 1 expert in location-aware systems.	3 participants, all with experience working on location aware systems, 2 with intermediate knowledge 1 with novice.	Part 1: 15 participants 11 users no knowledge of SBAs, 3 intermediate, 1 novice. Part 2: 4 participants	19 participants ranging from no-experience (6), novice (4), intermediate (6) and expert (3).
Tech-nologies Employed	UML Modelling Tool “Poseidon”	SimConfig - Graphical Modelling Framework (GMF)	SimConfig, SimConGen, SimConViz. J2ME App.	SimConfig, SimConGen, SimConViz.	SimConfig, SimConGen, SimConViz.	SimConfig, SimConGen, SimConViz.
Evaluation Method	Formative, structure based on Common Industry Format	Formative, structure based on Common Industry Format	Formative, structure based on Common Industry Format	Formative, structure based on Common Industry Format	Formative, structure based on Common Industry Format	Summative, structure based on Common Industry Format
Evaluation Metrics	Time to complete tasks, significant errors, how users rated tool in terms of usability.	Time to complete tasks, significant errors, how users rated tool in terms of usability.	Time to complete tasks, significant errors, how users rated tool in terms of usability. Questions regarding level of understanding with respect to the effect of uncertainty on the evaluated application.	Time to complete tasks, significant errors, how users rated tool in terms of usability. Questions regarding analysis of appropriate context types to meet requirements of the evaluated application.	Time to complete tasks, significant errors, how users rated tool in terms of usability.	Placement and understanding of reasoning for placement of SimCon sources to meet objectives of experiment. Time to complete tasks, significant errors, how users rated tool in terms of usability.

The outcomes of their placements, along with time to complete tasks, significant errors, satisfaction and question regarding level of understanding were then used to determine if the SimCon model does support SBA developers in the process of evaluating an SBA when faced with varying levels of uncertainty.

Table 6-1 Overview of Usability Evaluations gives a more descriptive overview of the usability evaluations to guide the reader. It gives the name of the experiment, the number of participants and salient aspects of their backgrounds and the different technologies employed the evaluation method and metrics.

6.3 Usability Evaluation 1

6.3.1 Evaluation 1A: Comparative Usability Evaluation SimConfig Prototype

6.3.1.1 Introduction

The initial SimConfig prototype set out to address requirements identified during the state of the art analysis of existing context simulators. The prototype thus looked at supporting the key feature of creation and configuration of SimCon sources in a usable manner to simulate user driven context. During the design, the need was identified for a method to capture the flow of context in order to support simulation of low and high levels contexts were identified. To support ease of modelling this flow, a graphical approach was designed for creating and configuring simulated context sources, from low level context up to higher level context was designed. At this stage of development, each SimCon source in this flow had the following properties associated with it: output, rate and uncertainty, including delays and accuracy; taken from the state of art examination of properties of context sources. It should be noted that in formative evaluations 1A and B, as well as 2A and 2B, the term accuracy was used to mean the variance in the offset of the true location from the measured location. In the later evaluations the term precision is used. This is because the term accuracy is often used interchangeably with precision as discussed in Chapter 2. The experimental instructions made clear what accuracy meant in the context of each experiment.

The prototype was decoupled from the Tatus game engine and the Hammer editor thus supporting flexibility with respect to the interactive visualisation component of the simulation framework requiring, only, that the type of data being generated by the VR engine was modelled. To generate user driven simulated context, the SimCon Generator must still be associated with an interactive VR environment to access data on its state, for example the coordinates of user controlled avatars. The experiment evaluated two approaches to visualising this simulation process. Both approaches were based on Unified Modelling Language (UML) techniques. UML includes a set of graphical notations for creating visual models of systems. The first approach was to use UML sequence diagrams. A sequence diagram captures the data being passed between entities in a system, in a sequential manner. The second approach was to use UML activity diagrams. Each activity is represented by a node, and each node can represent further subsets of activity diagrams. The direction of arrows connecting these nodes indicates the order in which activities occur.

6.3.1.2 Goal

The research question was: Which method for creating and configuring simulated context sources provides greater efficiency and user satisfaction? This led to the following hypothesis:

- **$H1_0$** *the null hypothesis implies no difference between times to complete tasks and user satisfaction for both methods*

- **$H1_a$** *the principal alternative hypothesis implies that there will be detectable difference in the efficiency and/or the satisfaction between the activity diagram approach and the sequence diagram approach*

Therefore the experiment goal was: “to investigate two graphical interfaces to support the modelling of contums in order to determine which provides greater efficiency and user satisfaction” and examined objectives SO1 as highlighted in Figure 6-3.

6.3.1.3 Participants and Background

As typical SBA developers were identified as those participants with backgrounds in software engineering, six participants with suitable backgrounds were chosen. Of the six participants, five were PhD students and one was a master student, four with backgrounds in computer science and all researching in computer science related topics in the Knowledge and Data Engineering Group (KDEG) in Trinity College Dublin (TCD). A set of pre-questionnaires, see below, set out to evaluate their background and experience. As the experiment was looking at using UML approaches to modelling context flow, the questionnaire set out to determine the participant's knowledge of UML and whether this had an impact on their experience of the interface. The questionnaire also set out to determine their experience with context and context-aware systems design and evaluation. One participant had no knowledge of UML diagrams, two were novice and three intermediate. Three participants rated their knowledge of context in relation to context aware systems as novice, and three as intermediate. Only one participants had frequently worked on designing a context-aware application, two infrequently, and three never. Finally, two participants had worked frequently on evaluating a context-aware application and three had never worked on evaluating one.

6.3.1.4 Experimental Description and Tasks Description

Before the experiment was run the participant was presented with a video of an Instant Messaging Application that used avatar location context from the VR building. This was done to highlight the purpose of the tasks, which were to model context using information provided by a VR environment to test a context-aware application. In addition, information was also provided to introduce the participant to new concepts.

6.3.1.4.1 Technologies and Tasks

Both approaches were based on Unified Modelling Language (UML) techniques. UML includes a set of graphical notations for creating visual models of systems. The first approach was to use UML sequence diagrams. A sequence diagram captures the data being passed between entities in a system, in a sequential manner. The second approach was to use UML activity diagrams. An activity diagram captures the flow of activities in a process. Each activity is represented by a node, and each node can represent further subsets of activity diagrams. The direction of arrows connecting these nodes indicates the order in which activities occur. Appendix C, section 11.1, Figure 11-1 shows an overview of the graphical layout of the SimConfig Tool which was implemented using Poseidon (GentleWare 2012). The first screen displays to the user the data being captured by the VR Building and the process by which this is converted to a contum. The prototype tool thus provides functionality to convert this xml message into a contum. Appendix C Figure 11-2 shows the Activity approach to modelling the Context transforms in closer detail.

Tasks were presented in a counter balanced order to prevent ordering effects, i.e. half of the users started using the activity approach and the other half using the sequence approach. The two tasks required the participant to use the SimConfig prototype to create a SimCon source to transform location information generated in the VR building into a contum using either a sequence diagram approach or an activity diagram approach. This involved creating an activity or a sequence node in UML and assigning values convert the incoming location data from Half-Life 2 into a contum. Thus, additional data had to be assigned in the form of an accuracy, delay and also the output format for the x, y and z position (Figure 11-2).

6.3.1.5 Findings

To answer the research question time to complete tasks and user satisfaction were recorded and are presented here. Significant errors are also presented as these are related to the overall usability of the approach.

6.3.1.5.1 Observation and Quantitative Metrics

All participants completed the tasks in times ranging from 24 minutes to 50 minutes. Table 6-2 shows the breakdown of these times. The tasks are named sequence and activity and are numbered depending on the order in which they were completed. The average time to complete both tasks was 17 minutes.

Table 6-2 1A Time to Complete Tasks (in minutes)

	Participant 1	Participant 2	Participant 3	Average (rounded up)	Standard Deviation (rounded up to nearest point)
Sequence 1	17	26	31	21	4.7
Activity 2	12	11	15	13	2.1
	Participant 4	Participant 5	Participant 6		
Sequence 2	19	10	9	13	5.6
Activity 1	31	14	17	21	9.1
				Total Average	Total Standard Deviation
Sequence				17	6.345602
Activity				17	7.339391

Significant errors are recorded in Appendix E section 13.1.1 and were addressed in subsequent evaluations of SimCon.

6.3.1.5.2 Post-Questionnaire Findings

The post questionnaire set out to discover usability issues of the taken approach, mainly, whether there was a significant difference between preferences for the activity and sequence approaches. Each participant was asked “Q3: Creating a new contum using the activity diagrams was?” “Q4: Creating a new contum using the sequence diagrams was?” the answers are in (Figure 14-1). The questions had a four point answering scheme, ranged from very difficult to very easy. All questions can be found in Appendix F section 14.1.

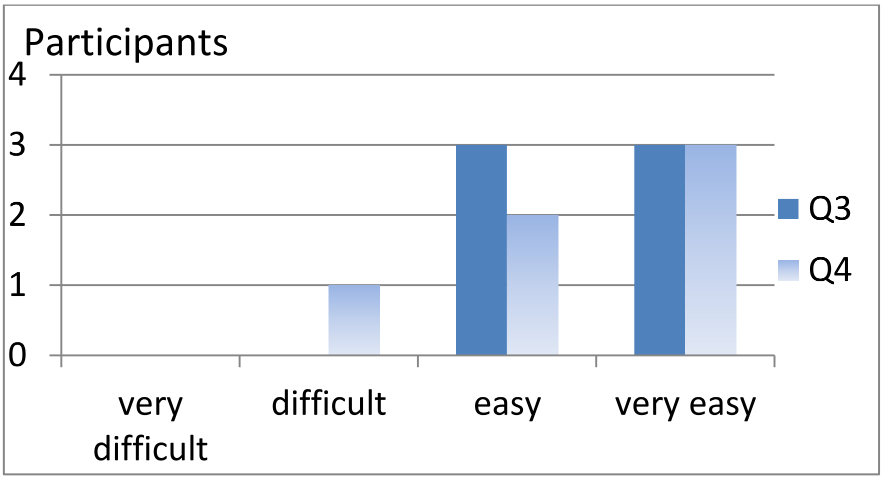


Figure 6-4 Post Questionnaire (Q3 and Q4)

6.3.1.6 *Interpretation of Findings*

Observation of participants proved itself to be highly informative for discovering errors in the interface. Observation showed subjects had trouble with interface elements, informing future changes. In discussion with the three experienced participants in the area of context-aware systems, they highlighted there was a difficulty making a cognitive link between the concepts of data being created by the VR building and the output in the form of a contum. SimCon therefore required some grounding with the VR building as participants found it difficult to understand the relationship between the configured SimCon sources, the VR environment and the position of avatars. It was concluded that new methods were required to improve SimCon source configuration and for grounding the process in a manner that makes the relationship between the simulated context source and the VR building more transparent to SBA developers.

The time to complete tasks was 17 minutes on average for both tasks, which suggest that neither approach was more efficient. It should be noted that participant 4, due to a misunderstanding of the purpose of the experiment took almost twice as long to achieve both tasks than the other two participants who did tasks in the same order. This is reflected in the larger standard deviation for completing task 1. This used the activity diagram and had an impact on the times to complete, which without participant 4 would have favoured the activity approach in terms of efficiency. Nonetheless, the questionnaires revealed that the activity diagram approach was found to be the preferred style by all participants, as all either found it easy or very easy to model using this method, whereas the sequence approach was found by one participant to be difficult. On this basis, as a preference was shown in reported usability, but with efficiency not detectably different, **H1$_a$** is considered true and therefore activity diagram representation for configuring context processing is selected over the sequence diagram representation.

6.3.2 Evaluation 1B: Formative Usability Evaluation of SimCon Version 1

The second evaluation looked at usability issues in the first standalone Graphical Modelling Framework (GMF) prototype of the SimConfig Tool through formative analysis. It set out to investigate whether the newly implemented interface for the SimCon model would show any significant deviation in errors and time to complete tasks than that of the previous evaluation. This involved users using the implemented tool and providing feedback regarding issues encountered, along with observation of them using the tool. As typical users of SimCon are identified to be persons with experience in the computer science domain, these were once again chosen to provide feedback on the approach. As efficiency forms part of the usability assessment of SimCon the time to complete tasks was also measured for comparison with previous and later implementations. The experiment also investigated the use of ground floor plans to improve the cognitive link between the virtual environment and the context simulation process, as identified an issue in the previous evaluation.

6.3.2.1 Goal

The research question was: does the SimConfig interface (SimCon Version 1 Appendix C Figure 11-3) provide a usable approach for typical SBA developers when creating and configuring two heterogeneous SimCon sources?

- **H1$_0$** *the null hypothesis implies that participants will be unable to complete the tasks.*

- **H1$_a$** *the principal alternative hypothesis implies that the approach taken will support technical users creating and configuring simulated context sources*

The experimental goal was: To investigate usability of the SimConfig Tool when creating, placing and configuring two heterogeneous SimCon sources for creating user driven location-based contums and examined objectives SO1 and SO3 (Figure 6-3).

6.3.2.2 Participants and Background

Five participants took part in this evaluation, each were PhD students in TCD. All had backgrounds in computer science. The pre-questionnaire set out to determine the background of the participants in more detail. Three of the participants had no experience with ubiquitous computing (Table 12-1). Two of the participants had experience with context-aware systems, one with “context aware and ubiquitous computing”, the other “pervasive computing, and middleware”. Only one had experience conducting research using sensor systems (Table 12-1).

6.3.2.3 Experimental Description and Tasks Description

The usability test began with an introduction which outlined the role of the participants. It explained the two types of location system on which the simulated context sources were based. These were Ubisense real time location and Tyndall ZigBee Proximity transceiver mote. Links were also provided with details on each of these systems. A download was provided of the prototype SimConfig tool was provided and also, due to findings in evaluation 1, a ground floor plan Figure 6-5. The ground floor plan created a cognitive link for users between the process of configuring the simulated context source and its relation to its position in the VR building.

Figure 6-5 also shows two Ubisense sensor cells. Each cell is bounded by a three dimensional box defined as a bottom and top corner. Each cell has a message rate and delay (representing additional time introduced as a result of the communication medium). It was explained that the message rate of 0.1 seconds indicates that a message is generated every tenth of a second. Each cell also has an uncertainty model defining the accuracy of the context source (an offset from the true location of the avatar).

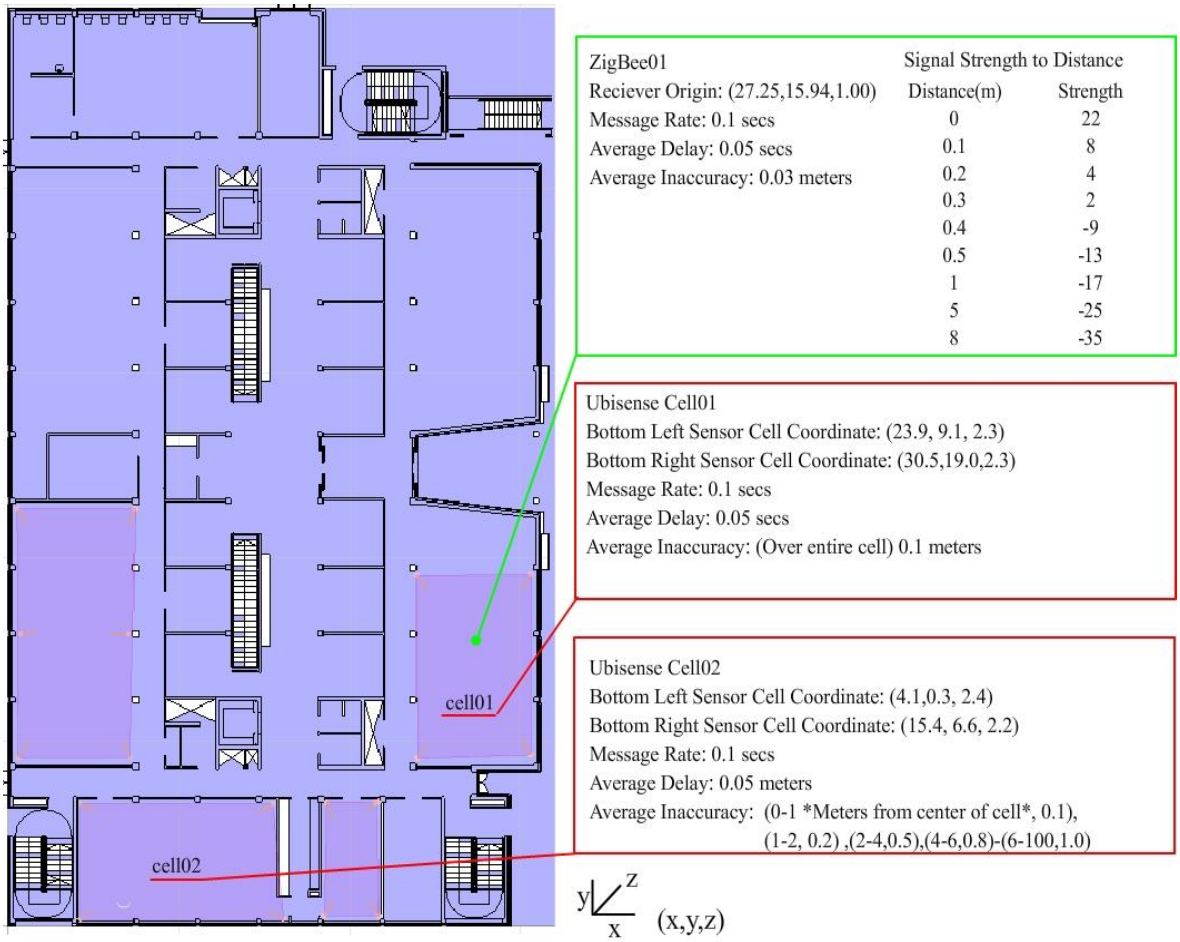


Figure 6-5 Virtual Reality Building Floor plan with Context Source Info.

Similarly a Tyndall ZigBee Proximity transceiver mote is also described. In addition to the other values, a Tyndall mote also has a list of values which relate the distance of the avatar from its origin to received signal strength (RSS). These values were taken from readings from analysis of an actual ZigBee deployment in the room represented by cell02, although for the purpose of this evaluation the actual values were not of importance, rather the ability to configure them. These specifications were supplied to each participant.

6.3.2.3.1 Technologies and Tasks

The SimConfig interface was developed in response to changes to the underlying SimCon model during its iterative design and development. This early interface is based on the activity diagram approach to capturing the flow of context as informed by evaluation 1A. Snapshots of this early interface are included in Appendix C Figure 11-3. Participants were given five tasks: Task 1 was to open, execute, and familiarise themselves with the SimConfig Tool. Task 2 had them create a SimCon Generator. Task 3 had them create and place a SimCon source, i.e a Ubisense Cell, and configure a blanket uncertainty model for the cell, an output rate, and introduce a random delay. Task 4 had them create a second SimCon source (i.e a Ubisense Cell) with a new uncertainty model. Task 5 had them configure a SimCon source (this time a Tyndall ZigBee Proximity Transceiver) and configure its steady state responses, delays and accuracy. The tasks themselves presented minimal instruction, but a link was provided to a more complete set of instructions.

6.3.2.4 Findings

This section presents findings of the experiment with respect to the time to complete tasks and user satisfaction. Significant errors are also presented as these are related to the overall usability of the approach.

6.3.2.4.1 Observation and Quantitative Metrics

Figure 6-6 shows the breakdown of the times to complete each task. All participants completed the tasks, varying in times from 27 minutes to 57 minutes. The average time to complete all tasks was 40 minutes. These figures can also be found in Appendix E section 13.2.1.

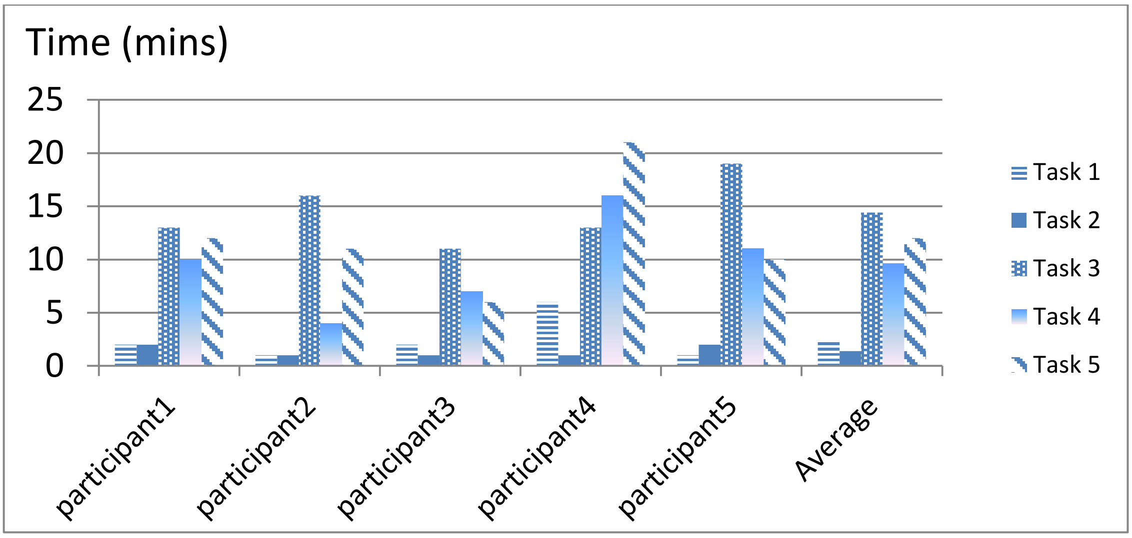


Figure 6-6 Time taken per task and average time per task.

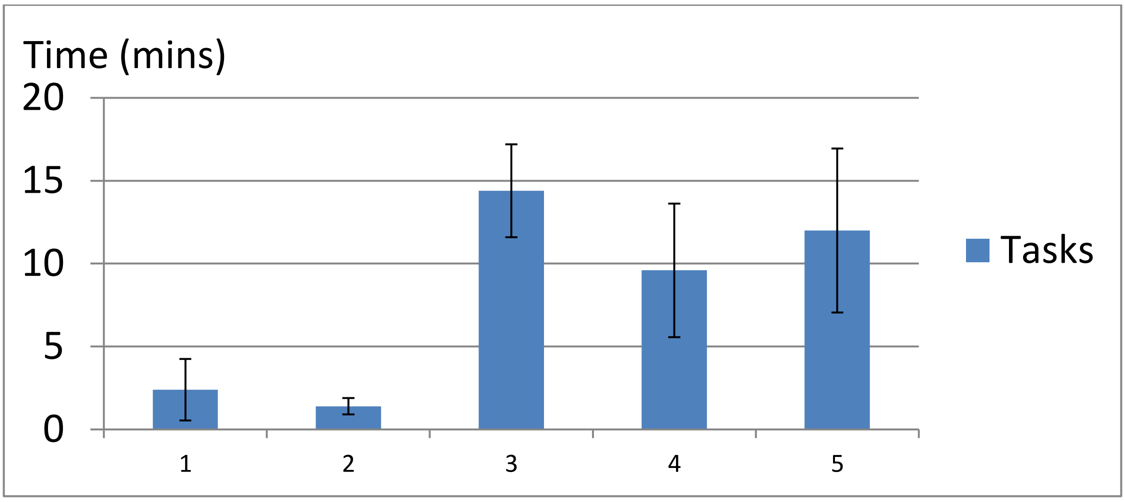


Figure 6-7 Average Time Per Task and Standard Deviation

Significant errors are recorded in Appendix E section 13.1.2.

6.3.2.4.2 Post-Questionnaire

The post questionnaire consisted of the following questions and results (Figure 6-8). The first five questions were related to tasks which were relevantly simply, like creating a SimCon generator and so these can be found at Appendix F section 14.3. The three most relevant to the research question are: “Q6: Setting the SimCon source area, delay and accuracy was?” “Q7: Configuring a SimCon source error distribution was?” and “Q8: Configuring a SimCon source response curve was?” The post questionnaires were, once again, on a four point scale ranging from very easy to very difficult.

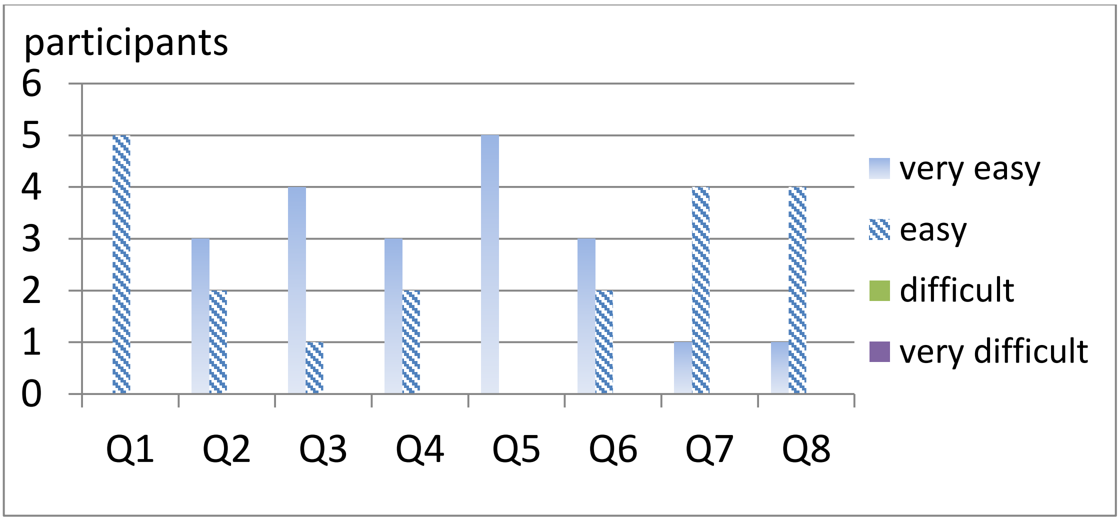


Figure 6-8 1B Post Question Answers (Appendix F section 14.3)

All the questions returned easy, or very easy.

6.3.2.5 Interpretation of Findings

The majority of errors were a result of bugs in the GMF prototype, which were later fixed. Errors related to terminology were fixed through better explanation or re-thinking of use. It was found that Task 1 and 2 were trivial for all users and thus required no more evaluation. Task 4 required the participant to repeat the process of task 3, adding a Ubisense SimCon source, with additional configuration of the uncertainty model. There was a marked improvement in times for the majority of participants to complete task 4 over task 3, which demonstrates practice effects when doing similar tasks within the context of this experiment. Task 5 required additional configuration of a large number of values and this resulted in a significant increase in the amount of time to complete the task. Comparatively, task 3 and 4 are similar to the tasks assigned in Evaluation 1A. The average time to complete these together with familiarisation with the tool was 29 minutes. The average time for Evaluation 1A to complete both tasks was 34 minutes. While the two evaluations are not exactly equitable, the small difference in time does indicate that SimCon Version 1 is at least similar in the efficiency for completing tasks related to configuring simulated context sources as the UML approach in Evaluation 1A.

Task 3, 4 and 5 required participants to read lists of numbers from the browser and then enter them into the SimConfig tool. This process was observed to be laborious and had an impact on the efficiency of completing tasks. In task 5 the average time to configure the Tyndall ZigBee Proximity SimCon source was an improvement over the configuration in task 4, as the configuration of error distributions and response curves were similar, and so repetition improved the efficiency (Figure 6-6). It should be noted that some terminology was confusing (the term “steady state” was unfamiliar to the users with no background in sensor or sensor related systems).

During observation, the absence of comments regarding the link between the context sources and the building suggest that the presentation of the 2D building blue print addressed this issue. It was also observed that this resulted in less confusion regarding the context of the experiment as a cognitive link could quickly be made between the context source creation and configuration and its relationship to an actual building. The post questionnaire gave good indications about the overall usability of the tool. The newly implemented GMF interface made the process of creating, placing and configuring a SimCon source an easy task for all participants regardless of their background. The questions did not however reveal enough detail about whether the users would chose to use this tool, although, the participant with experience in sensor deployments was enthusiastic about it in conversation and recommended that it could be improved through visual support for placement, perhaps through a graphical interface, once again reflecting similar context simulators reviewed in Chapter 3 (Norbisrath, Mosler et al. 2006; Jouve, Bruneau et al. 2009).

As all users were able to complete the tasks and answered either “easy” or “very easy” to questions regarding configuring the simulated context sources, this points to positive findings with respect to the principle alternative hypothesis for this formative evaluation **H1a** that the approach taken will support technical users creating and configuring simulated context sources. The number of participants (five) for this evaluation means that definitive claims about the level of usability of the tool cannot be made. It should be seen in the context of a formative evaluation as part of an iterative design, as described in the experimental methodology.

6.3.3 Summary and Conclusion Usability Test 1

SimCon Version 1 addressed the question “How usable is the SimCon model when creating, placing and configuring user driven simulated context sources to generate location based contums”. The evaluations did not look at how effective the SimCon model is when evaluating a smart building application, but did show that the GMF interface, based upon the activity diagram approach, proved itself to be a usable method to achieve this goal. The use of a visual representation of the building reduced the number of questions from participants in evaluation 1B compared to evaluation 1A, regarding what the link between the task and the goal of the SimCon model is, i.e. to create simulated context for evaluating SBAs. These formative evaluations therefore informed the next stage of development of SimCon Version 2.

6.4 Usability Evaluation 2

For SimCon Version 2, the link between the user, the VR environment and the simulated context is investigated in greater detail. As such, SimCon Version 2 required that user driven contum generation be implemented and methods for the evaluation of SBAs supported, as well as providing a usable tool for creating, placing and configuring simulated context sources. Here the three evaluations of SimCon version 2 are presented. Each evaluation presents the goals, description of experiment (including tasks and technologies) as well as findings and interpretation. A summary and conclusion of findings is given at the end of all three evaluations (Evaluation 2A, 2B and 2C).

6.4.1 Evaluation 2A: Formative Usability Evaluation of SimCon Version 2

The third evaluation 2A, set out to determine usability of SimCon when using the SimConfig tool to create and place a Ubisense SimCon source and configure it with different levels of uncertainty, and also the usability of SimConGen and SimConViz for completing the task of evaluating a location-aware application.

6.4.1.1 Goal

The research question was: does the SimConfig model provide a usable approach for typical SBA developers when creating and configuring a SimCon source to evaluate the effect of varying accuracy in simulated location on a location-aware application?

- **$H1_0$** *the null hypothesis implies that participants will be unable to complete the tasks.*
- **$H1_a$** *the principal alternative hypothesis implies that the approach taken will support technical users creating and configuring uncertainty in a simulated context source and support SBA evaluation by visualising simulated context.*

The experimental goal was: To investigate usability of the SimCon model when creating, placing and configuring a SimCon source to support participants evaluating the effect of varying delays and accuracy in simulated location on a location-aware application. In particular, the experiment set out to evaluate how effective the SimCon model was when used to evaluate an SBA with varying levels of delays and accuracy. It therefore examines objectives SO1 and SO2 (Figure 6-3).

6.4.1.2 Participants and Background

Eleven participants in all took part in this evaluation. This group included one Masters and five PhD students from TCD, three PhD students, and two post-doctoral graduates from University College Dublin (UCD) all with backgrounds in computer science. Seven of the participants had studied location sensor systems. Six had worked with location sensor systems. One postgraduate in UCD considered himself an expert in the field of location sensor systems, having helped to install and evaluate the Ubisense system in UCD. The three UCD students also had intermediate knowledge working with Ubisense (Appendix D section 12.3.1).

6.4.1.3 Experimental Description and Tasks Description

The introduction for this usability evaluation included a description of the role of the participant. It explained that they are taking the role of a mobile context-aware games developer and are currently designing a game that makes use of location context to manoeuvre around a virtual maze (by moving around a physical space and tracking that movement via sensors) and that they are concerned about how well the game will behave for varying levels of uncertainty in the real time sensing of the user's location. They were to make use of SimConfig to create, place and configure the simulated context sources and SimConGen to simulate these types of issues and evaluate their design. The usability test began with the evaluator showing the participant the maze game and how to work it. When the participant felt comfortable with the maze game they began the exercise.

6.4.1.3.1 Technologies and Tasks

Maze Game Application

The emulated J2ME Maze Game was designed by four final year undergraduate Information and Communications Technology students studying in TCD and was representative of the type of third party application that could be used during this evaluation. The maze is presented on a J2ME emulator 2D display (Figure 11-4). The game requires that the physical location of a player be sensed and provided as a coordinate. The application is calibrated by a player moving from one corner of a room to another and entering their location at each to set the size of the maze. Once this has been done, the game creates an internal model of a maze made of rows and columns of equally sized squares within this boundary. The player is then required to traverse a path of squares from a starting square at one side of the room to a finishing square at the other end of the room using just a representation of the maze and their location in it displayed on an emulated smart phone. If the player steps outside the boundary of a square they must go back to the start (Figure 11-4).

Initial development of the game took place on the Tatus platform. This meant that the location data being supplied by the Virtual Reality environment was delivered to the SBA without any model of uncertainty being imposed. With this high fidelity location, traversing the path was tricky. When the maze game was subsequently evaluated, using an actual physical Ubisense installation, it was found that inaccuracies in the sensed location made it extremely difficult to traverse the path without a sensed location falling outside the boundary of the path in the maze due to the variation in the sensed position provided by Ubisense from its actual position. Even when standing still, erratic inaccurate location readings made it appear to the game that the player had stepped outside the square they were in thus resetting the game so the player must start again. The exercise for the participants of this experiment was to explore how different levels of uncertainty affect the application and to determine a level of uncertainty in location beyond which they could say the application would not work.

SimConfig, SimConGen and SimConViz Tool

The SimConfig tool had the following features added according to findings in evaluation 2. An option was added to the pop up widget which supports loading pre-set values for sensor deployments. The layout was also adjusted to address issues which arose in experiment 2 (see Figure 11-5 bottom). A visualisation component was implemented which allows a user to visualise in 3rd person their location within the virtual environment, as a blue avatar, and also to visualise simulated context sources within the virtual environment, a green avatar. Figure 11-4 (Left) shows the implementation giving a 3rd person view of the VR building and the accurate location as given by the games engine used in this experiment. Figure 11-4 (right) shows the context source boundary deployed “in” the VR building.

There were four tasks. Task 1 asked participants to download the material, and then to familiarise themselves with SimConfig and SimConViz. To do this, they were required to move their avatars around the Virtual Environment (in the first person perspective) and observe their changing location in the visualisation tool (in a third person perspective). Task 2 asked them to create, place and configure a SimCon source and apply pre-set Ubisense properties, taken from the Ubisense product specifications, to provide a contum which reflects a Ubisense output. Task 3 had them view the SimCon source in the SimConfig tool, and also view the contum, visually represented by a green avatar, in real time as it was being produced by the SimCon source and test the operation of the virtual maze application. Task 4 had them reconfigure the accuracy and delay values on the SimCon source and test the application once again. Task 4 had them reconfigure the accuracy and delay values on the SimCon source and test the application once again. More extensive step by step instructions were only supplied in this evaluation for one task, task three, in which the user must place and configure a simulated context source. This could be accessed by a hyper link. Once again, a floor plan was supplied to give the users coordinates needed to position the simulated context sources.

6.4.1.4 Findings

To answer the research question, the findings of the experiment with respect to the time to complete tasks and user satisfaction are presented here. Significant errors are also presented as these are related to the overall usability of the approach.

6.4.1.4.1 Observation and Quantitative Metrics

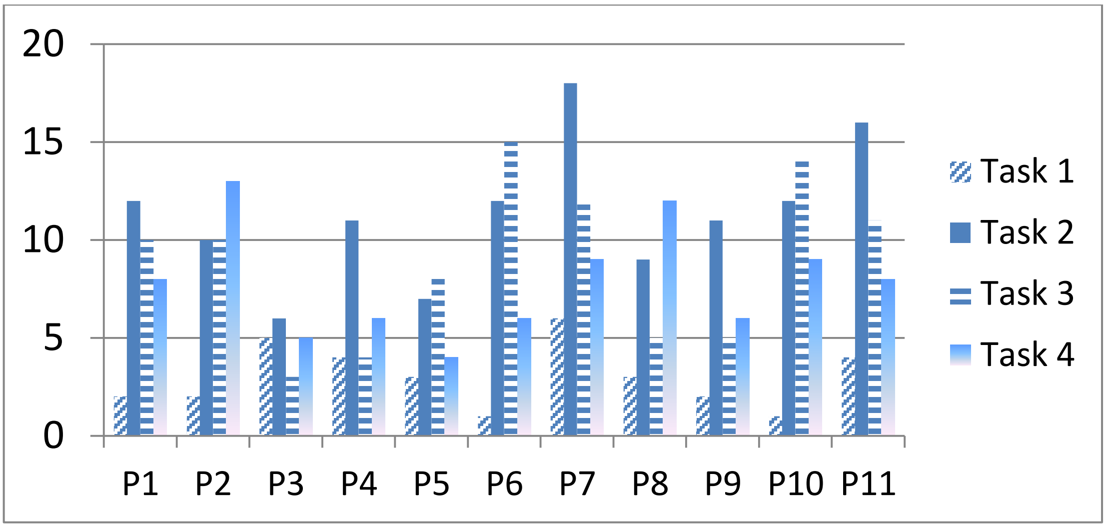


Figure 6-9 Time taken per task (minutes)

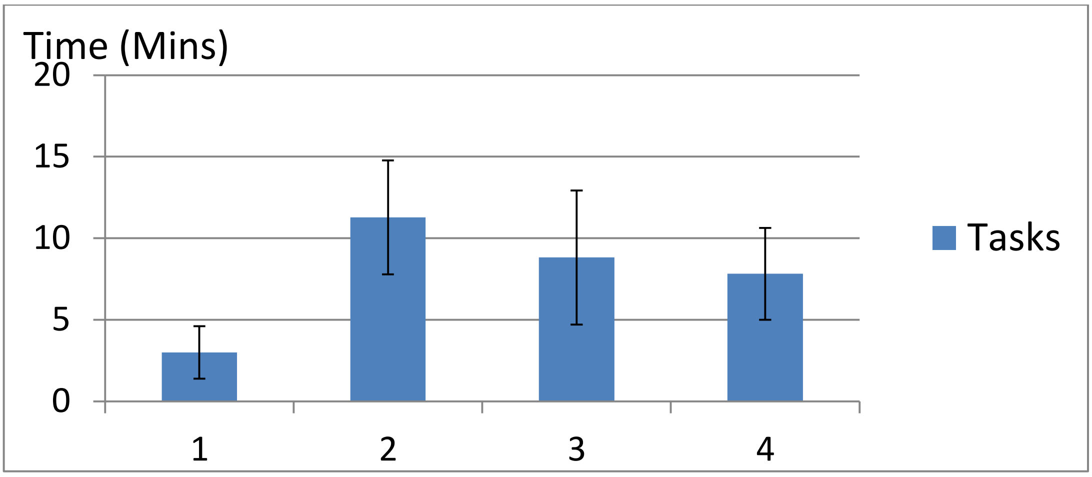


Figure 6-10 Average Time Per Task and Standard Deviation (minutes)

Significant errors are recorded in Appendix section 13.1.3.

All participants completed the tasks ranging in times from 19 to 45 minutes with an average 31 minutes to complete the tasks. Figure 6-9 shows how much time was spent on each task. Figure 6-10 gives an average time and standard deviation.

6.4.1.4.2 Post-Questionnaire

A short post questionnaire asked some questions regarding usability on a four point scale. The most relevant being “As a tool for highlighting how uncertainty effects an application would you say it was:” with available responses “very effective, effective, somewhat effective, not effective”.

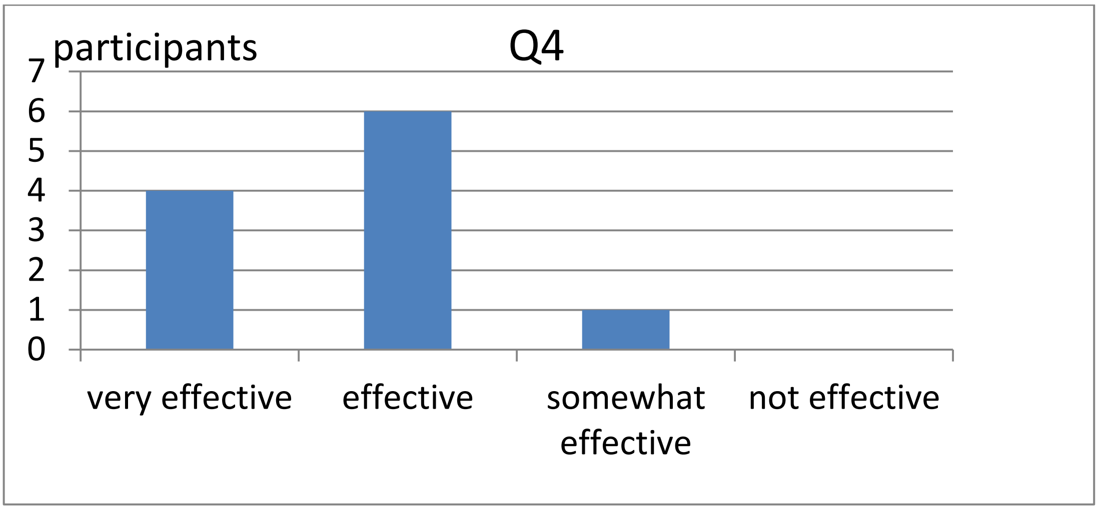


Figure 6-11 Q4 "As a tool for highlighting how uncertainty effects an application would you say it was:"

6.4.1.5 Interpretation

This evaluation again examined usability of the SimConfig tool when placing and configuring SimCon sources and in particular, how users found configuring additional uncertainty. Task 2 took on average 11 minutes (Figure 6-10). This task is similar to tasks 3 and 4 of Evaluation 1B which took 14 and 10 minutes respectively (Figure 6-6), showing no great deviation in times to complete between evaluation 2A and 1B for similar tasks. The time to complete task 4, which had the users configure additional uncertainty took on average 8 minutes, with a standard deviation of 2.4 minutes (Figure 6-13).

The further question this evaluation set out to answer was whether the visualisation of simulated context with varying levels of uncertainty would support the evaluation of the SBA. In task 4 the use of the additional SimConViz tool allowed participants to evaluate how introduced delays and changes in accuracy of location had an effect on how this location is "perceived" by the SBA in real time. Through feedback from the participants, this was seen as an effective strategy to quickly assess the impact of erratic location context and relate this to the application. The questionnaire reinforced this by revealing that four users found it very effective, six effective and one somewhat effective.

The evaluation also relied on observational data and feedback through open comments in the post questionnaire. Positive comments included: "Gives [the tool] a feel as to how modifying parameters effect the application very quickly" and "overall parts/application were intuitive -> I could see how they all worked together". The former comment was made by the expert user. Negative comments included: "Found it a little difficult moving between multiple application (visualisation tool, gmf tool, phone, instructions)". These findings indicate that for this formative evaluation, with respect to H1a, that the approach does support technical users creating and configuring uncertainty in a simulated context source and also the use of visualisation of uncertainty in simulated context was seen as an effective method to support the evaluation of SBA. However, this indication requires further quantitative metrics before any strong claims with this respect can be made.

6.4.2 Evaluation 2B: Formative Evaluation of SimConfig Tool Usability

Evaluation 2B set out to examine how usable the SimConfig tool is when creating, placing and configuring three simulated context sources providing heterogeneous types of location-based contums which were then required to evaluate a hypothetical security system in a building. Some minor additional changes were made to the SimConfig tool, since evaluation 2A. This included a coordinate system for placing SimCon sources and additional visualisation of SimCon source properties, the position, type and id (Figure 11-6). This experiment likewise set out to explore usage by participants with backgrounds related to SBA development to determine SimCon's suitability. Based on previous formative assessments, features were added to the SimCon model, including the ability to select pre-set values of a SimCon source by selecting a type of sensor (e.g. a Ubisense cell), with the aim of speeding up the process of simulated context source configuration. The ability to view the SimCon source in the SimConfig tool provided a means for the participant to understand the boundary and area of effect of the SimCon source, and see in real time its placement in the building.

6.4.2.1 Goal

The research question was: does the SimConfig interface provide a usable approach for SBA developers with backgrounds in smart building design when creating and configuring three SimCon sources to evaluating a hypothetical SBA?

- **$H1_0$** *the null hypothesis implies that participants will be unable to complete the tasks.*

- **$H1_a$** *the principal alternative hypothesis implies that the approach taken will support technical users creating and configuring three simulated context sources and will be able to evaluate the correct combination of contexts to meet the requirements of the SBA*

The experimental goal was: To investigate usability of the SimConfig and SimConViz Tool when creating, placing and configuring three heterogeneous SimCon sources for creating location-based contums in order to evaluate a hypothetical security system. In particular, the experiment set out to evaluate the use of three heterogeneous types of SimCon source and the visualisation of three types of simulated context. It therefore examines objectives SO3 and SO4 (Figure 6-3).

6.4.2.2 Participants and Background

Three participants in all took part in this evaluation. The group comprised of two PhD students with backgrounds in civil engineering and location aware systems from University College Cork (UCC-CE) and one post-doctoral graduate with a background in sensor simulation for the design of wireless sensor networks from Cork Institute of Technology. The evaluation was conducted in the Environmental Research Institute (ERI) in Cork. Each participant also had experience working with location tracking systems (2 with intermediate knowledge and 1 novice) and also working with applications which use indoor location tracking sensor systems (2 with intermediate knowledge and 1 novice) (Appendix D 12.3.1).

6.4.2.3 Experiment Description and Tasks Description

Each participant was given a description of the requirements for a hypothetical security application. This security system has two functions:

1. Alert security personnel when an unauthorised user has entered the building.

2. Alert security personnel when a tagged item has been moved by an unauthorised user.

To achieve function 2, either coordinate (e.g. Ubisense) or proximity (e.g. Tyndall ZigBee Proximity motes) or a combination of both may be used to (a) track authorised users and (b) track items. When an item is moved by a non-authorised user, an alert is sent to security. All authorised users must carry tags at all times. Security alerts can be provided to security through a mobile application, which either gives symbolic (room, area) or coordinate (x,y,z) location, depending on the location tracking system.

To achieve function 1, a combination of presence event context SimCon sources based on pressure mats, and real time location coordinates SimCon sources based on Ubisense, were used. Ubisense tracked the authorised (tagged) user so that when the authorised user steps on a pressure mat an alert is not sent to security. When a non-authorised user stepped on a pressure mat, an alert is sent giving the location of the pressure mat.

6.4.2.3.1 Technologies and Tasks

There were six tasks. For Task 1 the participant was to familiarise themselves with the SimConfig Tool. Task 2 the participant was to familiarise themselves with the SimConViz Tool. To do this, they were required to move around the VR building and observe the representation of avatars changing location in the SimConViz tool. Task 3 had the participant create, place and configure a simulated context source which provided location coordinate context and apply pre-set Ubisense properties. Task 4 had the participant deploy and configure a simulated context source which provided proximity context and apply pre-set Tyndall mote transceiver properties. Task 5 had the participant create, place and configure a simulated context source which provided presence context. They also had to apply pre-set pressure mat properties, configured as simple on/off alert. Task 6 had the participant view the new SimCon sources in the SimConfig tool. They could also use the SimConViz tool to visualise the contum in real time as it was being produced by SimConGen in order to come to an understanding of the systems view of the smart building. They were then required to analyse the context with respect to the security application they were developing to determine which types of contum best suited their requirements. Additional instructions were supplied for each task. More extensive step by step instructions were also supplied for the first four tasks. By task 5 it was assumed the participant would be able to place and configure a context source without the aid of instructions.

6.4.2.4 Findings

To answer the research question, findings of the experiment with respect to the time to complete tasks and user satisfaction are presented here. Significant errors are also presented as these are related to the overall usability of the approach.

6.4.2.4.1 Observation and Quantitative Metrics

All the participants completed all the tasks in times varying from 37 to 39 minutes. Figure 6-12 and Figure 6-13 shows a breakdown of time to complete each task, average time and standard deviation.

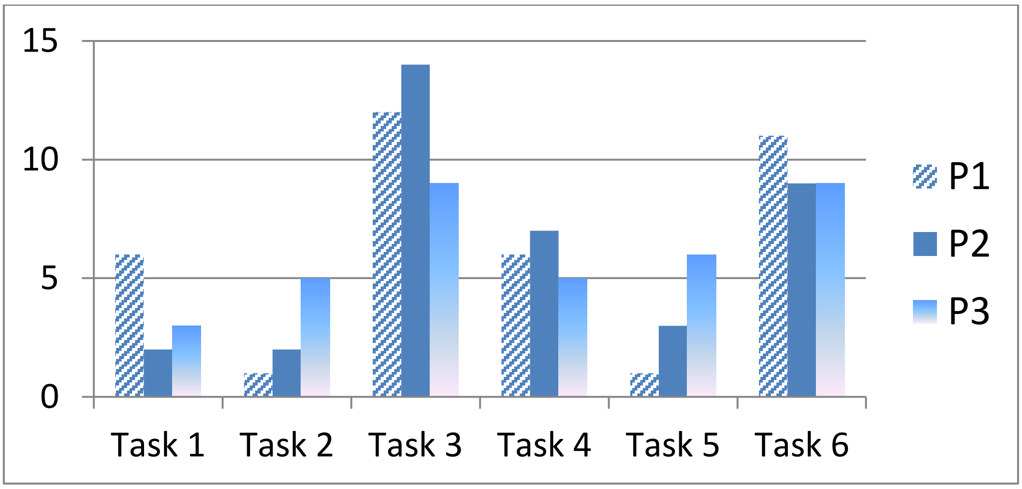


Figure 6-12 Time taken per task (mins)

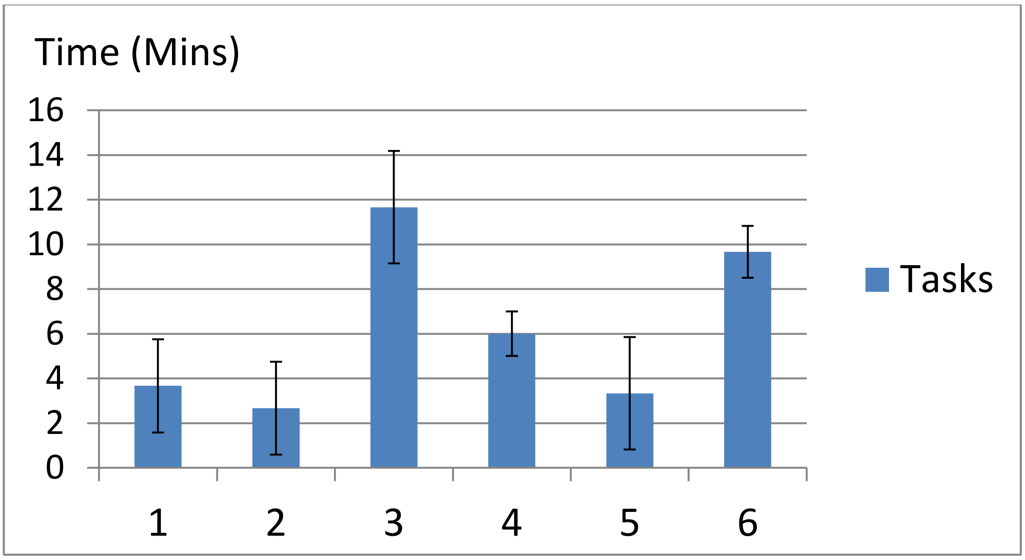


Figure 6-13 Average Time and Standard Deviation (mins)

Significant errors are recorded in Appendix E section 13.1.4.

6.4.2.4.2 Analysing the Security Application

Each user was required to analyse the contums to determine which SimCon source best met the requirements for the security application. This task was straight forward as the instructions gave details on the types of contums the application required and the types of SimCon sources which supply those contums. All users agreed that in order to determine whether non authorised users were in the building presence contums were required at strategic locations, i.e. doors, hallways.

6.4.2.4.3 Post-Questionnaire

The first questions were open questions. These were: “Did you feel these context sources met your requirements for the security application?” “Which context source or combination of context sources best met the security applications requirements and why?” “Do you think this tool would be useful in evaluating context-aware applications and why?” All participants answered yes for question 1. Question 2 was answered mainly by the way of discussion, and all participants agreed that the use of presence and coordinate locations would suffice for the security application. Question 3 had the following answers: The most experienced participant, the post doctorate with experience in sensor simulation, answered “It’s good to be able to visualise coverage regions” but “auto selecting bounded regions [for a SimCon source]” would be useful. He also would like to see the visualisation tool integrated with the configuration tool (the current implementation the SimConFig tool and SimConViz tool are separate applications). Another participant answered “Yes, stable, easy to learn and use”.

This questionnaire also evaluated usability using the System Usability Scale (SUS) (Appendix F section 14.4) as it was judged that the toolset had reached a sufficient level of maturity that overall usability, rather than just task-specific assessment, would be informative to assessing the improvement in usability. The questionnaire included the 10 standard SUS question plus an additional tool specific question; question (S11 – “I think this system would meet my specific requirements”). Each question had a range of answers from strongly agree to strongly disagree. The overall score was 69.16 for the SUS, which places it marginally below 70 on the SUS rating given by Bangor and Kortum, which is a cause for concern (Bangor, Kortum et al. 2009).

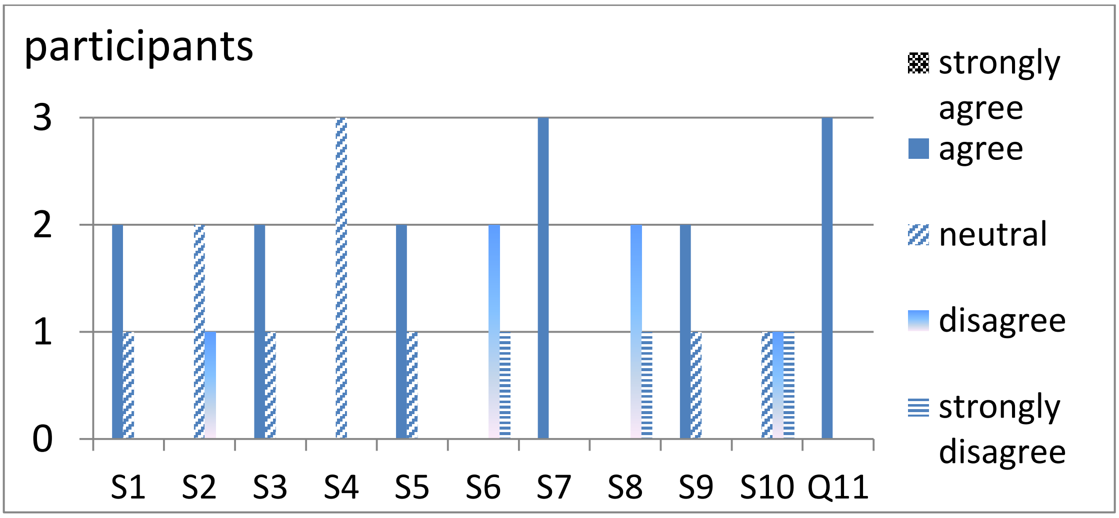


Figure 6-14 Post- Questionnaire Answers

6.4.2.5 *Interpretation of Findings*

The findings for placing and configuring the context sources are promising. On average, the time taken to place the second SimCon source was half that of the first. This tends to back up the claim in evaluation 1B that users become familiar with the different features of the tool quickly, although at this small level of participants, this claim requires more evaluation. Also, participants generally only consulted the instructions for the first and second deployment, after which they felt comfortable deploying a SimCon source independently of instruction. The analysis of the security application focused the participant's attention to a specific goal. As the task was straightforward, the participants had little difficulty completing it. There was also no indication that the combination of visualisations of different types of location context had a negative on their ability to achieve the tasks of evaluating the hypothetical SBA.

Once again, the numbers of participants in this experiment mean that definitive claims about usability cannot be made. Nonetheless the SUS did give good feedback on how each participant they found using the tool. Most notably, the majority agreed that it was easy to use and that they felt confident using it. The findings show that for this formative evaluation, with respect to H1a, that technical users were able to achieve the tasks and their responses to the questionnaires indicate a level of satisfaction with the approach to simulating and visualising context although the below 70 score indicates that additional attention on the usability of the tool was required.

6.4.3 Evaluation 2C: Formative Evaluation of SimCon Usability with Non-technical Users

In the previous evaluations all the participants had technical backgrounds in computer science and related fields. This evaluation set out to evaluate how users, without technical skills and who had little or no background knowledge of sensors and sensor enabled applications, but who are in the area of building design, i.e. architectural students, would find using the SimCon model, with particular attention on the use of SimConfig to create, place and configure a SimCon source. It also set out to identify aspects of the current implementation of the model which were causing consistent task errors using a larger set of participants, fifteen.

6.4.3.1 Goal

Hypothesis: The SimConfig interface will provide a usable approach for non-technical SBA developers when creating and configuring a SimCon source to evaluating the effect of varying accuracy in simulated location on a location-aware application."

- **$H1_0$** *the null hypothesis implies that participants will be unable to complete the tasks.*

- **$H1_a$** *the principal alternative hypothesis implies that SimConfig provides a usable approach to support non-technical users placing and configuring a SimCon source*

The experimental goal was: To determine usability issues of SimCon and use of VR environment with non-technical users when evaluating a hypothetical SBA. In particular, the experiment set out to evaluate non-technical users and identify any existing issues with the interface when using the range of its capabilities. It therefore examined objectives SO1, SO2, SO3 and SO4 (Figure 6-3).

6.4.3.2 Participants

Fifteen participants took part in all. All were third year students in BSc (Honours) Degree in Architecture, which is jointly offered by the Cork Institute of Technology (CIT) and University College Cork (UCC) as part of the Cork Centre for Architectural Education. These participants therefore represented users who may be required to design smart buildings, but who may not necessarily have strong technical backgrounds with respect to software engineering, a skill necessary to develop SBAs. The evaluation was conducted in an office in the Cork Centre for Architectural Education. Due to constraints on the number of PC's and the availability of students, evaluations were performed in groups, with a maximum of five students at any one time, with the mean number being three.

As it was known they did not have computer science backgrounds, the questionnaire was adapted to determine if they had any experience working with existing architectural CAD tools and manipulation of 3D building models. This would give insight into their competency with computers in the context of their discipline and whether this would have a noticeable impact on their use of the interface. The questionnaires revealed that 9 of the 15 had used CAD tools previously, 6 using Google Sketchup, 1 using AutoCad and 2 using both. The questionnaire also set out to determine their competency with respect to SBAs and sensor systems. 12 of the participants rated their knowledge of sensor systems as "No Knowledge" and 2 as "novice". 11 participants had no knowledge of sensor systems for detecting location, 3 considered themselves novices and 1 had intermediate knowledge.

6.4.3.3 Experiment Description and Tasks Description

In the experiment the SimCon model was assessed to evaluate if a set of non-technical users (architectural students) could create a simulated context source and examine the output with respect to the design of a security application for tracking authorised and non-authorised building occupants. A second evaluation then examined a sub set of these students creating a second simulated context source for tracking authorised users together allowing the security application determine if a user is authorised or not. Half of this sub set was required to configure the uncertainty exhibited by the application around a particular zone of the room and determine the impact this uncertainty had on the application.

The first part (Part 1) looked at a set of fifteen students. As the participants were working on separate computers, the times were only recorded at the beginning and end of the set of tasks for these participants. Due to access constraints (students had limited time) only four of these students were available to conduct the second part (Part 2) of the experiment, and they arrived in sets of two according to their availability. During part 2, the time to complete each task was taken. For both part 1 and 2 the number of significant errors, i.e. those which required intervention from the instructor were recorded. Part 2 also had the students fill out a usability questionnaire and some additional questions about issues related to uncertainty.

6.4.3.3.1 Technology and Tasks

The SimConfig, SimConViz and SimConGen components of SimCon were used. Additional implementations made to SimCon compared to the version used for evaluation 2B were the ability to assign and configure simulated tags/transmitters to avatars. Part 1 had the participants create a pressure mat SimCon source to detect when a person entered or left the building, for the purpose of alerting a security application that an unauthorised user had entered the building. The participants were then required to walk their avatar in and out of the virtual reality building and examine the output of the pressure mat (an xml contum message). They also had the SimConViz tool available to help in this respect by visualising the location event.

The following specific tasks were undertaken. Task 1 had them familiarise themselves with the VR building and SimConViz tools. Task 2 had them familiarise themselves with the SimConfig tool for creating, placing and configuring simulated context sources in a VR building. Task 3 had them create a pressure mat at the door of the NIMBUS building (NIMBUS 2012). The instructor then started up SimConGen so that the participant could then analyse the simulated context as it was generated, in order to evaluate the impact on the application. Part 2 examined four participants in two; groupA and groupB. GroupA were given the task to create a second SimCon source (a Ubisense real time location cell) and assign a tag to their avatar, so that they could track authorised users. The Ubisense outputs, in conjunction with the pressure mat outputs, were to then be used to alert the security application only when non-tagged users entered the building. GroupA then were to evaluate how uncertainty impacts on the application using the pre-set value of 0.15 cm at 1 standard deviation using a Gaussian distribution, offsetting the value by 0.15cms 95 percent of the time, which are the values given by the Ubisense specifications as discussed in Chapter 4 section 4.4.3.5. To do this they once again moved their avatar in and out of the building. They could examine both the xml data outputs or make use of the SimConViz tool to visualise the outputs. GroupB were then given a similar task with the addition of configuring extra uncertainty in an area around the entrance to the building (to simulate the effect of the glass on Ubisense precision). They configured the uncertainty zone with a 0.3 meters value at 1 standard deviation using a Gaussian distribution.

These values have been taken based upon findings through study of Ubisense conducted as part of this research where the standard deviation of the variance in the location for Ubisense was found to be between 0 and 0.5 meters at certain locations, attributable to the position of the sensors, the position of the tag, the presence of a human body, and the presence of large double pain glass windows (McGlinn, Brennan et al. 2011). It should be noted that these findings while used to inform the value of the uncertainty zone, are for the purpose of supporting the SBA developer increase uncertainty and not an attempt to create a high fidelity simulation of the environment. GroupB were then to evaluate how uncertainty impacts on the application in a similar manner as GroupA. The additional uncertainty would make the probability of the security system triggering an alert for a non-authorised user even when an authorised user was responsible for triggering it, as the location of the tag could potentially be detected in a different position than the pressure mat.

6.4.3.4 Findings

To answer the research question findings are presented here with respect to the time to complete tasks and user satisfaction. Significant errors are also presented as these are related to the overall usability of the approach.

6.4.3.4.1 Time to Complete Tasks

In part one the time to complete tasks varied from 14 to 45 minutes with an average time of 23 minutes.

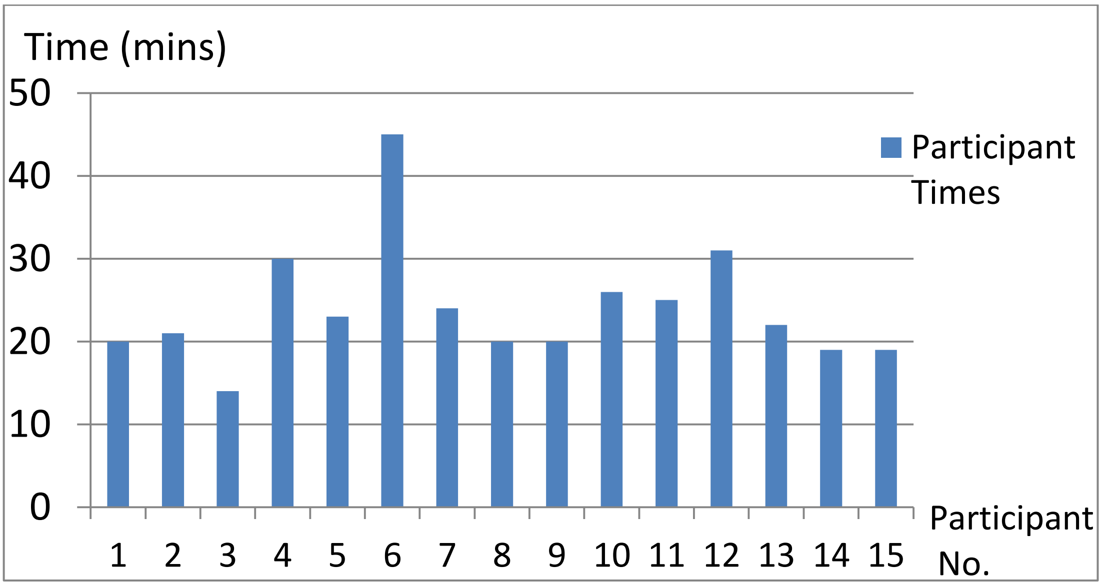


Figure 6-15 Times Part 1

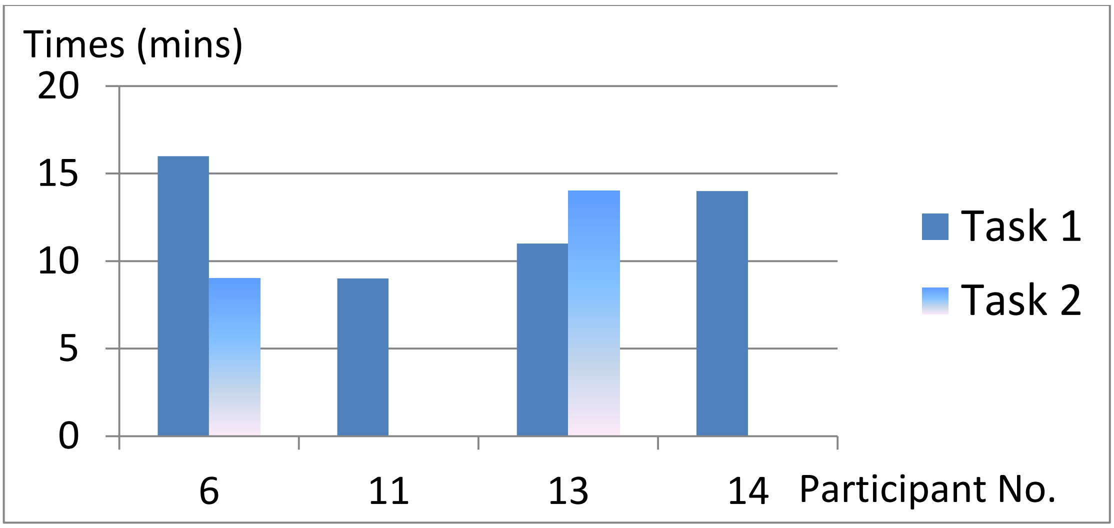


Figure 6-16 Times Part 2

Significant errors are recorded and can be found in Appendix section 13.1.5.

6.4.3.5 Part 2 Post-Questionnaire

The Standard Usability Scale indicates an overall usability of 51.25. Figure 6-18 and Figure 6-19 list post questionnaires. An additional question was also asked Q6: "What effect do you think uncertainty in user location has on the behaviour of the security application?" which only two participants answered, with "May not know when to act accordingly" and "Difficulty distinguishing friend or foe from entering the building". A table of these responses can be found in Appendix F section Table 14-1 Specific Usability Questions for Part 2 Participants

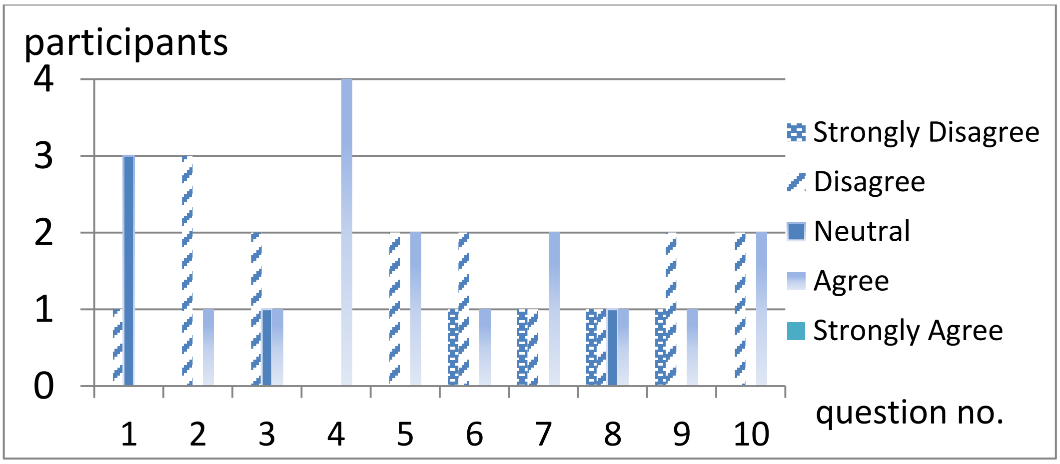


Figure 6-17 Standard Usability Scale

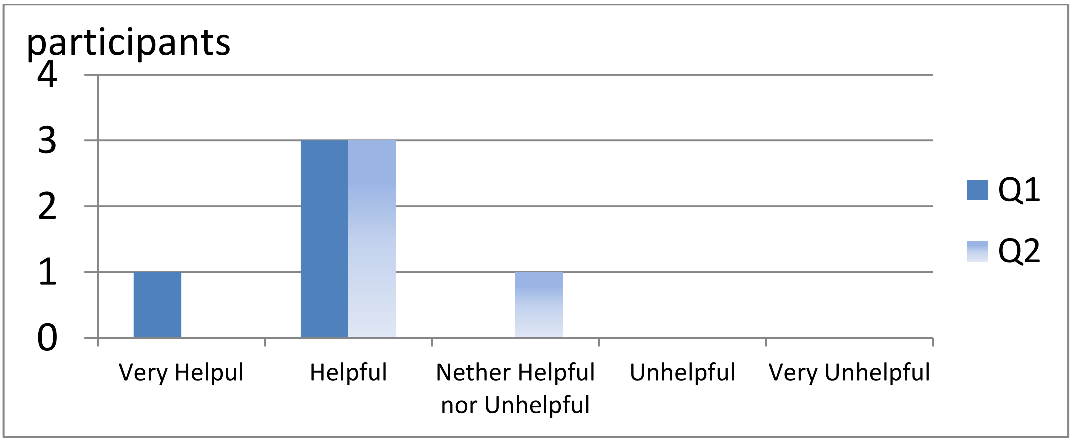


Figure 6-18 Q1: How helpful was sensor data (context) visualisation for understanding how the security application views user location? Q2: How helpful was user location visualisation for understanding how uncertainty impacts on the application for detecting intruders?

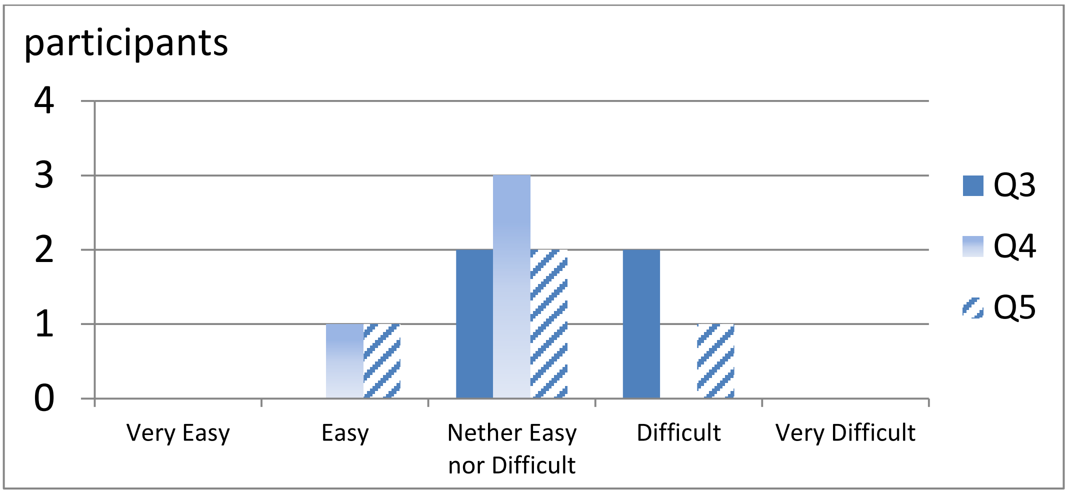


Figure 6-19 Q3: How easy was it to create a SimCon source type? Q4: How easy did you find configuring SimCon uncertainty? Q5: How easy did you find visualising sensor data (context)?

6.4.3.6 Interpretation of Findings

The times to complete tasks include the time to familiarize themselves with moving around the environment and the SimCon tools. A new issue of note arose during Evaluation 2C with respect to the use of the VR environment. Some participants found controlling the avatar difficult. This made it awkward for them to evaluate the application as they were required to move the avatar to generate user driven context. This issue related to the lack of experience of some users with first person VR and their mouse based control to move the avatar. This issue is discussed in the conclusion to the Evaluation 2 section and is addressed further in the final summative evaluation. The average time to complete the tasks for the 9 users who had experience using CAD tools and the six without was 21 and 28 minutes respectively, which would indicate that familiarity with the use of CAD tools had a positive effect on time to complete tasks. Comparing the time to complete tasks in Part 2, we see that the creation and placement of the SimCon source in task one takes an average of 12 minutes. This is considerably less time to complete a very similar task in Part 1 (23 minutes) (i.e. create, place and configure a SimCon source and analyse its output with respect to an application). This indicates that, as previous experiments have shown, familiarity with the interface improves efficiency.

The post questionnaire, although only conducted on four participants, still yielded a number of interesting findings (Figure 6-19 and Figure 6-20). Two of the participants found the creation of SimCon sources neither difficult nor easy, and two found it difficult. Three also found the process of visualising the position and boundaries of the SimCon source difficult, which requires more investigation as previous experiments did not indicate issues with this aspect of the tool interface. Visualising context had an equal split between finding this process easy and difficult, with two participants again finding it neither easy nor difficult. The negative responses may be indicative of a less computer literate test group. Of particular interest in this experiment was the process of configuring uncertainty. None of the four participants who took the Part 2 indicated any difficulty here, while one found the process easy. Promisingly, the visualisation of sensor data was seen as helpful by three participants and very helpful by one for understanding the application’s view of context. Three also found this helpful for understanding how uncertainty impacts on the application, with one finding it neither helpful nor unhelpful. This reinforces earlier findings which indicate that this type of visualisation of context data is useful for users when attempting to understand how applications must deal with location context and the effect of uncertainty.

The users were also asked what effect they thought uncertainty had on the application. Only two answered “May not know when to act accordingly” and “Difficulty distinguishing friend or foe from entering the building”. It is difficult to ascertain from these responses though their level of understanding.

6.4.3.6.1 Part 2 Usability Questionnaire

The Standard Usability Scale indicates an overall usability of 51.25 which is well below the 70 indicated by Bangor and Kortum needed to be satisfied about the usability of the system. Comparing the SUS score against a previous evaluation with civil engineers which gave a score of 69.16, we see a significant drop in user satisfaction. This may be as a result of the user’s background, as previous evaluations examined participants of post grad or doctorate level who had experience with computer systems.

With respect to H1a that SimConfig provides a usable approach to support non-technical users placing and configuring a SimCon source, the SUS scores would indicate that this is not the case for the four users who went on to do extra configuration. The 15 users nonetheless were able to achieve the tasks in less than half an hour. This which would suggest that SimConfig does support non-technical users placing and configuring SimCon sources. A summary of these findings and previous findings and their interpretation will now be provided in the next section.

6.4.4 Summary of Findings of Usability Evaluation 2 A, B and C and Implications for Final Summative Evaluation

These experiments consisted of formative evaluations of usability of the SimCon model as part of the iterative design and development of the SimCon model. Evaluation 2A set out to investigate usability in the SimConfig and SimConViz tool when creating, placing and configuring a SimCon source in order to evaluate the effect of varying accuracy in simulated location context to assess a a location-aware application. All participants completed the tasks, and the majority of participants seemed satisfied with the approach and reflected favourably on the use of the SimConViz tool for highlighting the effects of varying accuracy on the system perception of the state of the VR building.

Evaluation 2B set out to examine the use of a multiple SimCon sources and visualisation of generated contums. Although the numbers were small, the feedback from these participants, who each have experience working in the area of smart building design, provide valuable insights into the SimCon model. The need for more integration of the tools becoming apparent at this point as issues were repeatedly occurring with respect to the number of screens the users had to move between to complete the tasks. Also, while participants were able to identify the correct types of contum to meet the requirements of the SBA, it is still unclear exactly what benefit the SimCon model is providing in this respect. A comparative evaluation is therefore required which specifically set out to evaluate the effect of using SimConViz on the evaluation of an SBA. Most significantly though, the use of multiple types of context visualisation did not appear to introduce any greater cognitive burden on the participant and were shown to provide insights for the evaluators into the use of multiple sources of context.

Evaluation 2C set out to investigate usability issues for non-technical users when evaluating an SBA using the SimCon model. Due to the low numbers (four) who took part in the second part of the evaluation it is not possible to make definitive conclusions regarding the usability from the questionnaires alone. Nonetheless, observing this number of non-technical users gave a number of insights into the SimCon model which were not observed in previous evaluations. For example, the experience of the users with CAD tools had an influence on the time to complete tasks. Further evaluation will not look at non-technical users, and the potential for SimCon to be applied to non-technical users remains an open question that is not investigated further in this thesis. The issue related to the control of the avatar also became apparent in this experiment, where users with little experience with first person shooters had difficulty. While this was not reported in 2A or 2B, it was noted that this type human control over avatar movements could influence the repeatability of the experiments, as the freedom for the user to move unconstrained has an influence on the tasks at hand.

With respect to the user interface, for all evaluations (2A, B and C) the issue of clicking on the appropriate bubble to get access to the configuration tool from the GMF interface occurred and is a feature of GMF. This particular error is not repeated once discovered, but nonetheless created difficulty for the user, which it is assumed, had an overall negative influence on the user's experience of the tool and thus how they rate their satisfaction. The activity diagram approach taken with the GMF interface has proven itself to be a flexible method to add and remove context sources, not causing any difficulty for users. As the evaluations have concentrated on low level context sources and the use of one VR environment, the evaluation of the GMF interface is seen here as complete. As such, future evaluations will no longer explore the GMF interface in order to have greater time to explore other aspects of the SimCon model, like the configuration of uncertainty in context sources. While the usability questionnaires reflected an overall non negative response with respect to the integration of the different tools, further integration is required to further reduce errors resulting from moving between screens.

Also, the number of screens is a serious issue for evaluators and so impeded on the ability to evaluate the aspects of the SimCon model which are intended to demonstrate that it improves intelligibility, i.e. by allowing visual analysis of simulated context in real time as users interact with the building. Version 3 of SimCon will require integration of the visualisation and configuration into one interface so as not to distract from this goal. The removal of the GMF tool will also reduce the number of screens the participants are required to navigate and also the number of errors, some of which are attributable to it.

6.4.5 Usability Evaluation 2 Conclusion and Implications for Future Evaluations

The major contributions of these experiments were the evaluation of the level of usability possible in the implemented SimCon model, which supports the creation, placement and configuration of SimCon sources with varying levels of uncertainty to support SBA evaluation. Both Evaluation 2A and 2C indicated positive feedback with respect to the use of visualisation of location context to improve intelligibility of uncertainty in location context. Also, by reducing the complexity of the process of modelling uncertainty, the SimCon approach has proven itself to support all participants to complete the tasks of creating and placing SimCon sources in a simulated building and also configuring uncertainty. More research will need to be done to determine if this approach has in fact improved intelligibility though, and the next evaluation shall set out to evaluate this by comparing two groups of users, one with access to a textual output of context and the second with access to an additional visual display when evaluating the impact of uncertainty.

6.5 Usability Evaluation 3:

The formative evaluations presented previously addressed requirement **R2** defined in Chapter 4 by evaluating the level of usability of the SimConfig tool for creating, placing and configuring simulate context sources which generate context with varying levels of uncertainty, and also the level of usability of the SimConViz tool, which visualises the simulated context generated by SimConGen to support the analysis of varying levels of precision in simulated location context. A number of key findings led to changes in the design of the SimCon model and the implemented tool-set. SimConViz and SimConfig were integrated into one tool, as the issue for SBA developers moving between screens was identified. To further address the usability of the configuration process, SimConfig had multiple tabs created to divide the different aspects of the SimCon source model, for example placement and defining an uncertainty model.

It was also found that the use of user controlled avatars introduced an additional level of non-determinism, over that introduced by the uncertainty simulation, into experiments. This makes repeatability difficult. SimConGen was therefore extended to support bot driven simulation of context. Bots can be replayed multiple times and produce the same output, and so, bots are used in Evaluation 3. Bots have the additional benefit that they can be left interact with the simulator so that the evaluator is not required to interact with an additional interface to control the avatar, thus removing another screen which they are required to interact with. Findings with respect to the visualisation of uncertainty in context will still hold true whether it is bot driven or user driven context simulation.

6.5.1 Usability Evaluation 3: Final Summative Evaluation

The final summative evaluation looked at the evaluation of the level of usability of the combined features designed and implemented to meet the key objectives of the research question of this thesis. It therefore examines the configuration, simulation and visualisation of two heterogeneous types of location context to support the process of evaluating an SBA. From this evaluation, metrics are gathered on the overall usability of the whole system, which includes SUS scores and examination of its level of efficiency.

Secondly, it does a specific comparison of the effect of visualizing the uncertainty on the SBA developer against not having it in place. This addresses the effectiveness of using SimCon, as the process of supporting effective evaluation by being able to visualize the outcome of the uncertainty is key to supporting SBA developers face with accounting for context uncertainty in their designs.

6.5.2 Goal

The research question was: does the SimCon model provide a usable approach for typical SBA developers to evaluate the effect of varying precision in simulated location on a smart building application, by supporting creating and placing SimCon sources and also re-configuring SimCon sources with added uncertainty.

- **$H1_0$** *the null hypothesis implies that there will be no significant difference in the usability of the SimCon between users who have access to visualisation of simulated context and those that don't, in particular the placement of simulated context sources to address uncertainty in location context.*

- **$H1_a$** *the principal alternative hypothesis implies that the approach taken will support technical users creating and configuring simulated context sources and support the users correctly placing context sources to meet the requirements of the SBA when faced with varying levels of uncertainty in context.*

The research goal was: "to investigate usability of SimCon for SBA developers who must evaluate the placement of two context sources to enable a smart building application which relies on two heterogeneous types of context, one of which is subject to varying levels of uncertainty."

In particular, the experiment set out to evaluate how the visualisation of uncertainty could improve the effectiveness of the SBA evaluation process. The experiment involves two groups of typical SBA developers, a test and control group. The test group were given additional functionality over the control group, so that they could have a visual representation of the simulated context. The comparison of the two groups will therefore provide evidence on the relative importance of visualisation of uncertainty provided in concerts with evaluation. It therefore examines sub-objectives SO2, and SO4 (Figure 6-3). However, this experiment also continues the efficiency and satisfaction assessment across both groups using task timing and SUS evaluations. Therefore this evaluation provides usability assessment in support of all four sub-objectives, including the configuration of context sources and, to a lesser degree, dealing with heterogeneous context sources. Before this evaluation is described in greater detail, the next section will detail the experiment participants and their backgrounds.

6.5.2.1 *Participants and Backgrounds*

The experiment users classed playing the role of SBA developers. These were identified in Chapter 3 to be users with sufficient skills to be capable of developing an SBA, and so they required software development skills as well as some level of experience with context related systems. As such, users with background in computer science were once again targeted. This evaluation also wanted to identify particular skills within the test group which may have an impact on their ability to complete the tasks assigned them. Therefore a number of questions to gather information on their experience in certain areas were asked pre-experiment. As there were to be a test and control group, these questionnaires also made it possible to align the two groups so that they have equivalent experience in areas of importance to completing the tasks. When two participants were matched, they were randomly assigned to either the control or test group.

The questionnaires fell under five distinct categories: Experience developing and evaluating location-aware applications, experience developing and evaluating location systems, experience with sensor and context simulation, experience with uncertainty in context and context uncertainty modelling and finally, experience with probability and stochastic modelling. These were chosen due to their relevance for the tasks the participants were to be assigned. Each category had questions which could be rated “no experience”, “novice”, “intermediate” and “expert”. For each category then assigned scores (0-3). Respondents with the same scores totalled for each category were paired for allocation to between the control and test groups, but with category score equality prioritised by the order of categories given above. For a complete list of the questions and the answers see Appendix D section 12.5.2.

The test and control group were selected from participants assigned to one of four experience levels matching the four categories. This consisted of six users in the no experience category, four in the novice, six in the intermediate and three experts. Only three experts could be found for the experiment as finding additional experts in the field, who had not already used the tool, proved difficult in the population available. Two of these were assigned to the test group as their feedback was deemed important with respect to the features of the tool being evaluated. To conserve space, the no experience and novice users were grouped into a single group (here called non-expert for brevity) and the intermediate and expert were grouped into another (here called expert for brevity). For a breakdown of the four experience levels and assigning of participants, see Appendix D section 12.5.2.1.

Table 6-3 Control and Test Group along with Experience Levels

	Control	Test
No-Experience	3	3
Novice	2	2
Intermediate	3	3
Expert	1	2

6.5.2.2 Experimental Description and Tasks Description

The experiment set out to evaluate two groups of users with equivalent skill sets. Both groups took on the role of a smart building application developer who wished to install a security system to monitor a secure door to another building area. Both groups had identical tasks and instructions and both groups could visualise the bots as they moved through the environment. The exception was that the test group could visualise the simulated context generated by the simulated context sources.

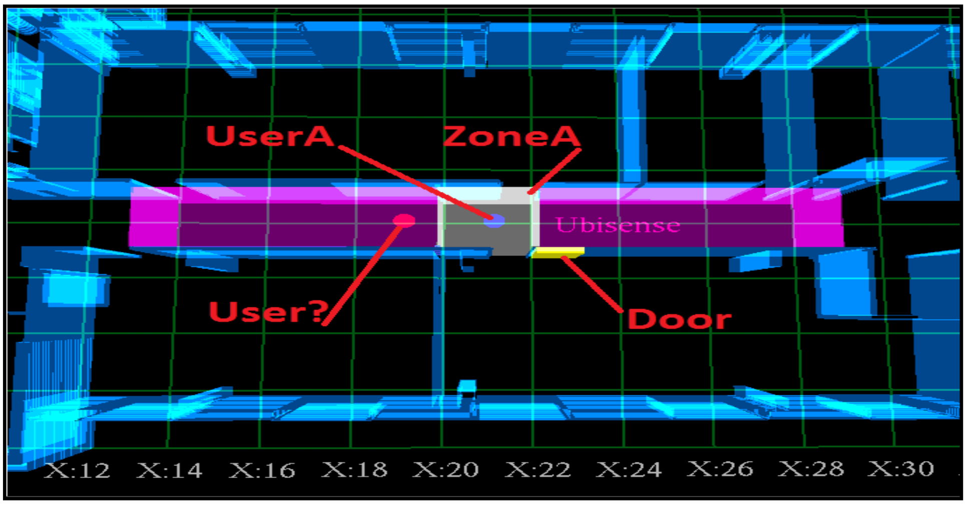


Figure 6-20 The corridor with a Ubisense cell (purple), zoneA (white) in front of the open door (yellow) and user? (red) and userA (blue).

Both groups were informed that in this test scenario a Ubisense system had already been installed in a corridor to track users carrying Ubisense tags, who were therefore assumed to be authorised personnel users (Figure 6-20). Their task was to design the placement of two Passive Infrared (PIR) motion sensors (on either side of an automatic door) to determine if a non-tagged user is tailgating when a tagged user opens the door. They were then required to test their approach using the SimCon toolset. The involvement of one tagged, user '**userA'** and one non-tagged user '**user?'** were to be considered in the scenario. Figure 6-22 gives an overview of the four situations which the placement of the PIR sensors was required to account for:

1) **userA** walks through **door**.

2) **user?** walks past **door**.

3) **userA** enters **door** and **user?** does not attempt to enter **door**.

4) **userA** attempts to enter **door** and **user?** also attempts to enter **door**.

For consistency between evaluation participants, and to ensure repeatablily between different SBA evaluation runs by the participants, the user behaviour was captured and replayed in the simulator as the movement of pre-programmed bots.

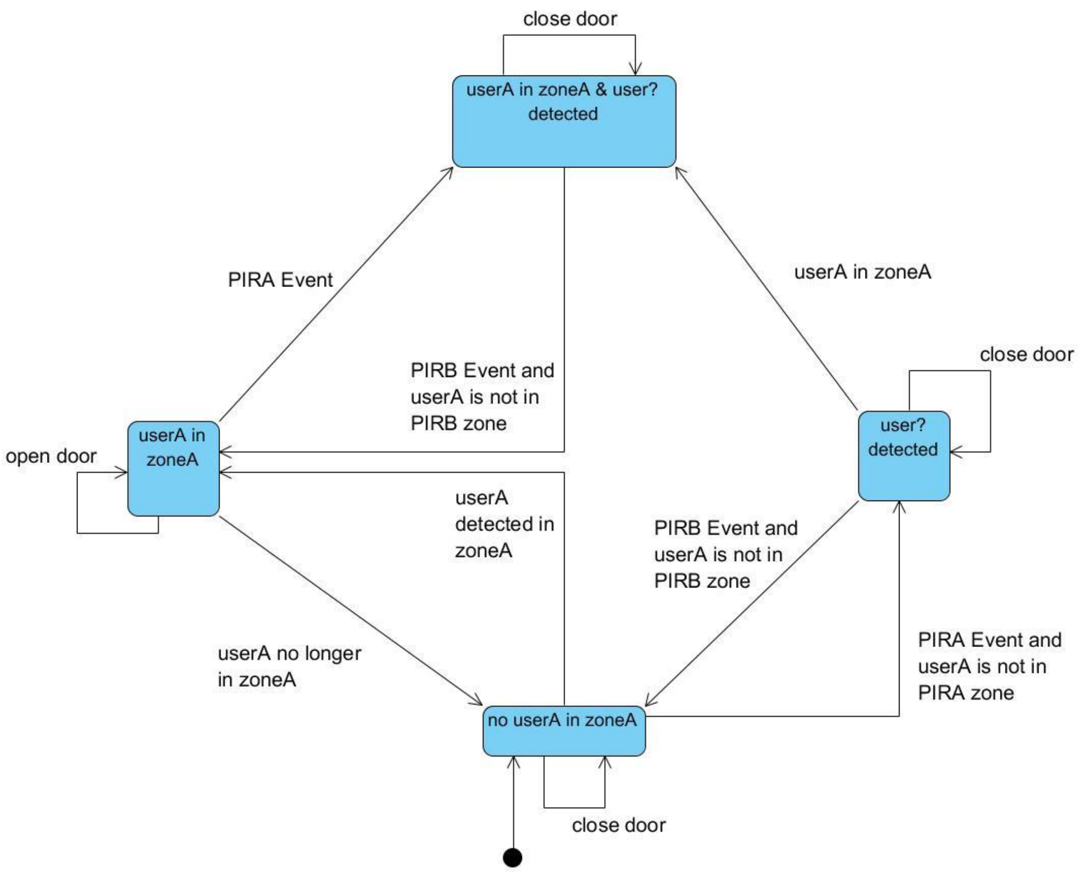


Figure 6-21 Automatic Door Finite State Machine

Each participant was shown how to play the four scenarios (by double clicking on a batch file named situation1 – situation4). They were also shown the security application prototype which provided simple textual feedback about its state, indicating one of following states: door opening, door closing, PIRA triggered door will no longer open, PIRB triggered door can now open. Figure 6-21 shows a finite state diagram describing this behaviour and Figure 6-22 shows an overview of these four situations. For a complete description of the experiment see Appendix H section 16.2.

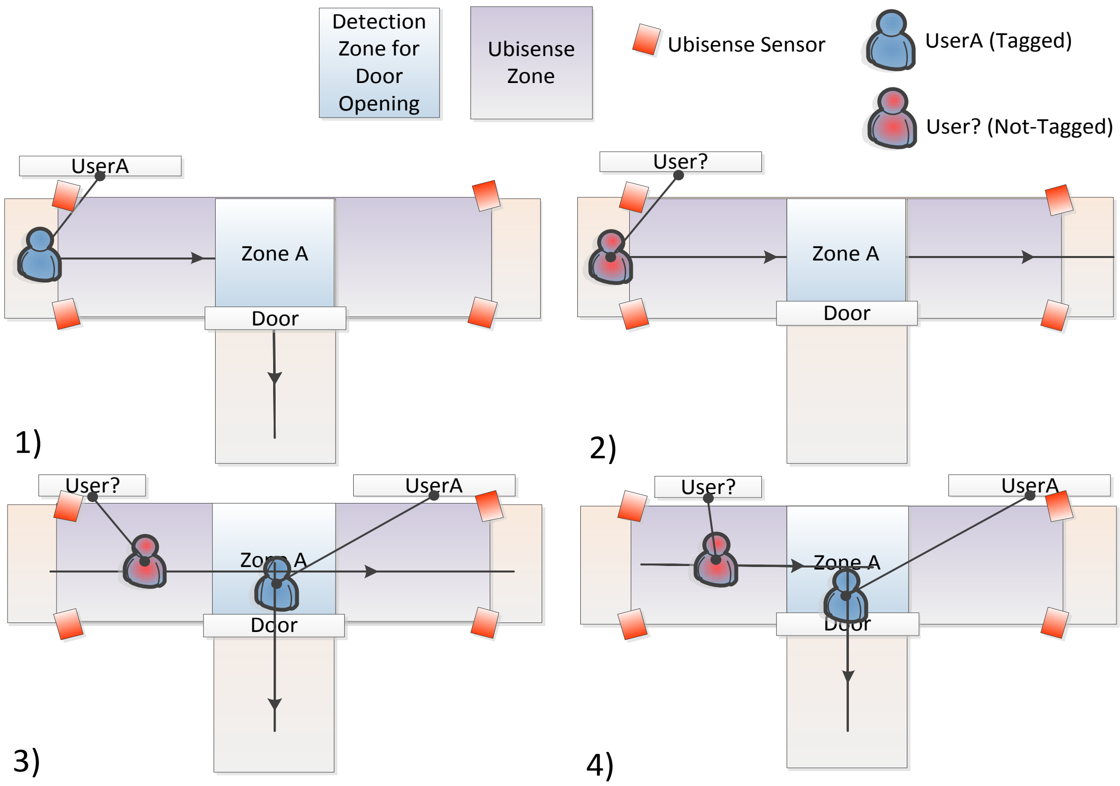


Figure 6-22 Experiment Situations

6.5.2.2.1 Technologies and Tasks

There were five tasks that participants were asked to complete. The first task required the participants to familiarise themselves with the SimCon tool, consisting of SimConfig and SimConViz. The second task was to use the SimConfig tool to place two PIR sensors. The first PIR (PIRA) could only be placed in three positions to the left of the ZoneA. The y position was fixed at the centre point of the corridor, and so they could only choose between three x position: x:15, x:17 and x:19. The second (PIRB) in three positions to the right (x:23, x:25 and x:27). These were to be placed to achieve two main objectives. The first and most important objective was:

- **To avoid incorrect identification of an authorised user** (resulting in the door not opening for the authorised user when there is no risk of a non-authorised user gaining access)

The second and of less priority:

- **To minimise the time taken for the door to be unable to open due to a PIR being triggered** (and thus avoid the inconvenience for tagged users having to wait for non-tagged users to trigger the second PIR once they have passed by the door and therefore toallow the door to open)

Task three had them replay the four situations four times each. This was to ensure that all users would have experienced the different situations at least four times to see the effects of their placements on the different situations, without uncertainty in context playing a role. At this point they could opt to change the placement of the sensors and run the situation a further four times. They could repeat this as many times as possible until they were satisfied the placement met the constraints on SBA behaviour they had been asked to achieve. The control group could examine the movement of the bot controlled avatar using SimConViz. They also could examine the state output from the security application (a command line feed) and the numerical output output of the sensors (also given through sequential a command line feed). The test group had access to the same features, but in addition they could view the sensor outputs visually using SimConViz, i.e. graphical indication of when a PIR was triggered and the simulated sensed position of the Ubisense tag overlaid onto the space layout (see Chapter 5 Figure 5-15).

As there was no context uncertainty configured in task three, the correct placement of the PIRs was the position closest to the door (x:19 and x:23). This is because there was no risk of incorrect identification via Ubisense and so these placements would result in the door being locked for the least amount of time when triggered by a non-authorised user.

Task four required the participants to introduce uncertainty by configuring precision for the Ubisense cell so that the tag would be offset by 0.05 meters at one standard deviation. This would create the possibility that the authorised users tag would be detected outside the PIR zone when the actual location of the user was in the PIR zone, resulting in the security system being falsely triggered and the authorised user in situation one being locked out of the room. It should be noted that this value is lower than that given in the Ubisense specifications. A lower value was chosen to make the likelihood of false identification by the security system more difficult to spot for this task. As the experiment evaluates varying the uncertainty, this would not impact the results. As this uncertainty was for the entire zone, it made no difference to the correct placement of the PIRs which remained at x:19 and x:23. After configuring the uncertainty, they once again had to evaluate each situation four times and then choose a new placement, repeating until they were satisfied that the constraint criteria for the SBA had been met.

Task five required them to configure precision for the Ubisense cell in a 2x2 meter zone on the left side of the door so that the tag would be offset by 0.3 meters at one standard deviation when in this area. This zone would exactly cover the PIRA zone if it was placed in position x:19. This uncertainty zone, it was described, was due to additional uncertainty introduced by RF reflective surfaces in that part of the corridor. This would increase the probability that the authorised user would trigger the PIRA, if it was left in the same area as the uncertainty zone. The optimum solution for the placement of PIRA in this task was position x:15, as there was an increased risk that placement in both position x:17 and x:19 would result in either a false identification and also the potential that a non-authorised user would be ignored. As the risk of an issue in position x:17 is a subtle one, position x:17 is also considered a correct placement for task five. After configuring the uncertainty, they once again had to evaluate each situation four times and then choose a new placement. For a complete description of the experiment and tasks see Appendix H section 16.2.

6.5.3 Findings

This section looks at the users collectively, and then as two groups, non-expert and expert. The findings have additional sections not present in previous evaluations. This includes a section on the positions selected for the PIRs, as these are used to determine whether the participant's successfully achieved their goals. The tables presented here only specify whether the participant correctly placed PIRA and PIRB (see Appendix G section 15 for the correct placements of PIRA or PIRB). Also, the findings for all participants do not include errors or comments made during the evaluation, as these are presented in the sub category groups. The post questionnaire also set out to assess participants' understanding of the effect of uncertainty.

6.5.4 Findings All Participants

6.5.4.1 Observation and Quantitative Metrics

6.5.4.1.1 Correct Positioning of PIRs

Table 6-4 Number of Participants who correctly placed the PIR Sensors: All participants

	Control (No. of participants out of 9)	Test (No. of participants out of 10)
Task 2	1	3
Task 3	8	8
Task 4	5	7
Task 5	3	9

6.5.4.1.2 Times to Complete Tasks

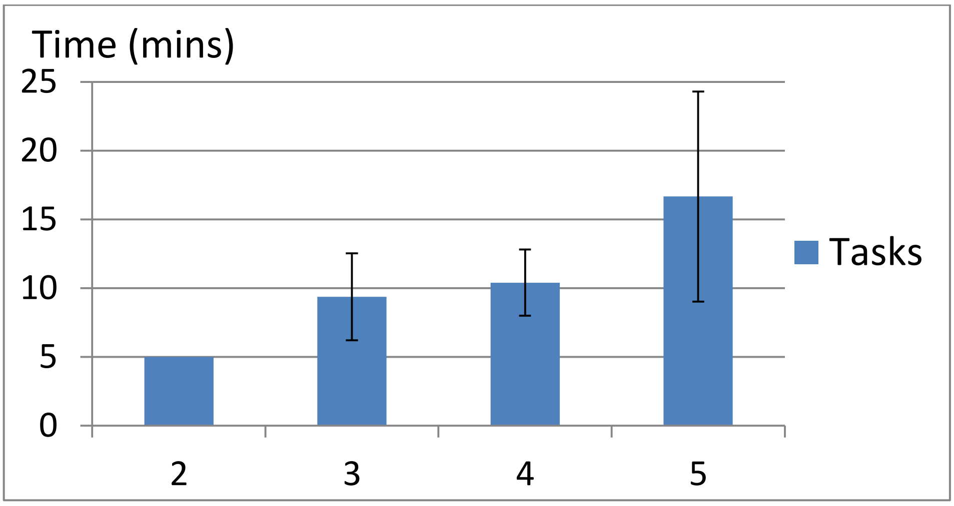


Figure 6-23 Average Time (and Standard Deviation) to Successfully Place PIR and Complete Task: All Participants (Control Group)

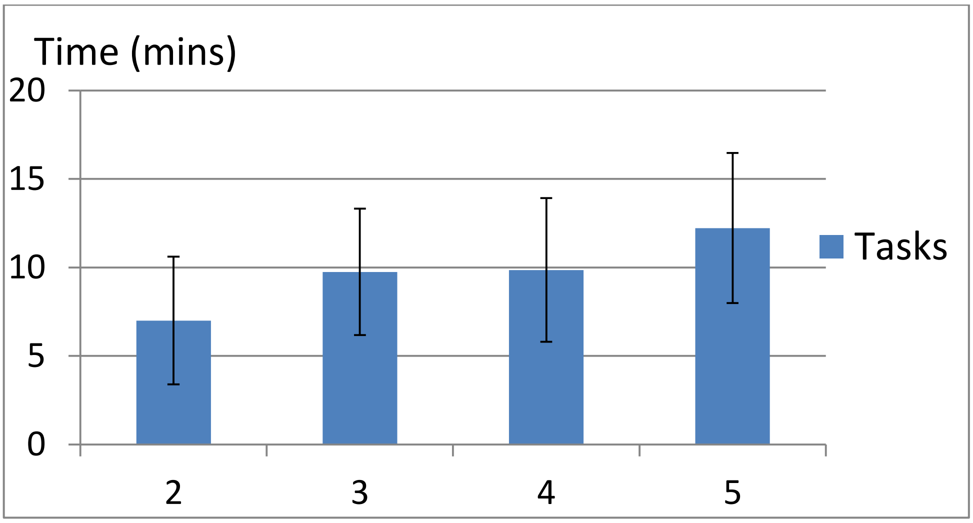


Figure 6-24 Average Time (and Standard Deviation) to Successfully Place PIR and Complete Task: All Participants (Test Group)

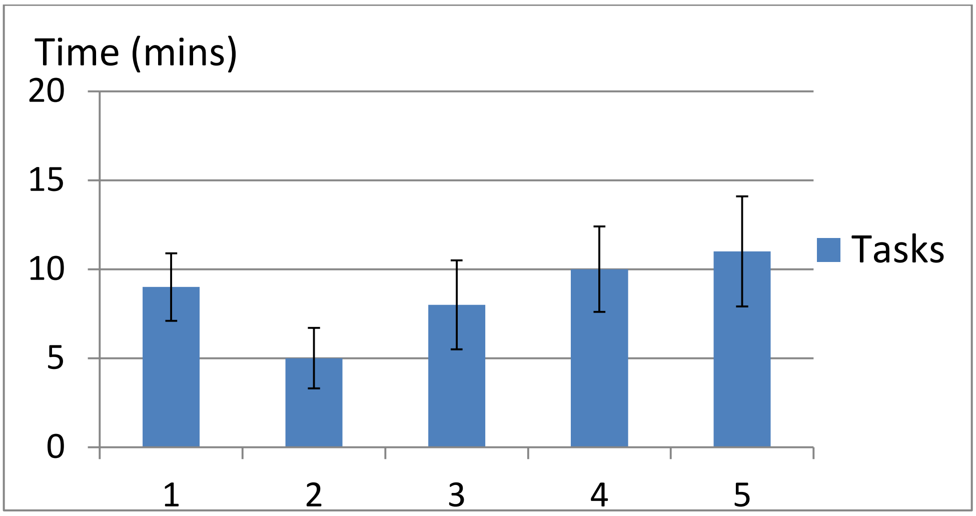


Figure 6-25 Average Time (and Standard Deviation) to Complete Tasks: All Participants (Control Group)

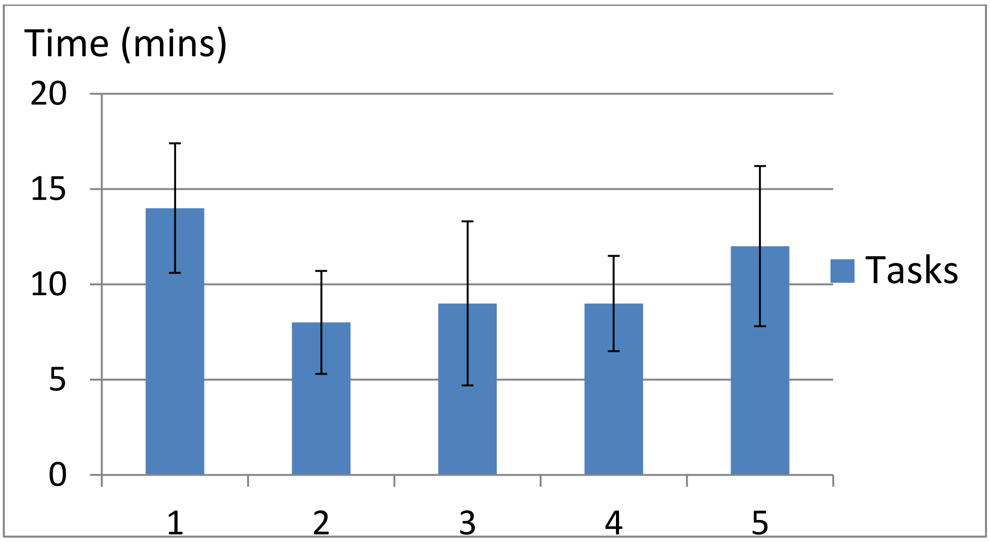


Figure 6-26 Average Time (and Standard Deviation) to Complete Tasks: All Participants (Test Group)

Total average time for control was 43 minutes with a standard deviation of 5.5. Total time for the test group was 52 minutes, with a standard deviation of 6.5.

6.5.4.2 Post-Questionnaire

6.5.4.2.1 Specific Questions about SimCon Tool

Specific questions set out to ascertain how the participants found using specific functions of the SimConfig interface, for example, question 3 was a statement "Creating a SimCon source was challenging " and question 6 was a statement "Configuring an uncertainty area was a challenge" which each participant had to answer on a five point likert scale. For a complete list of the questions and responses see Appendix F section 14.5.1.

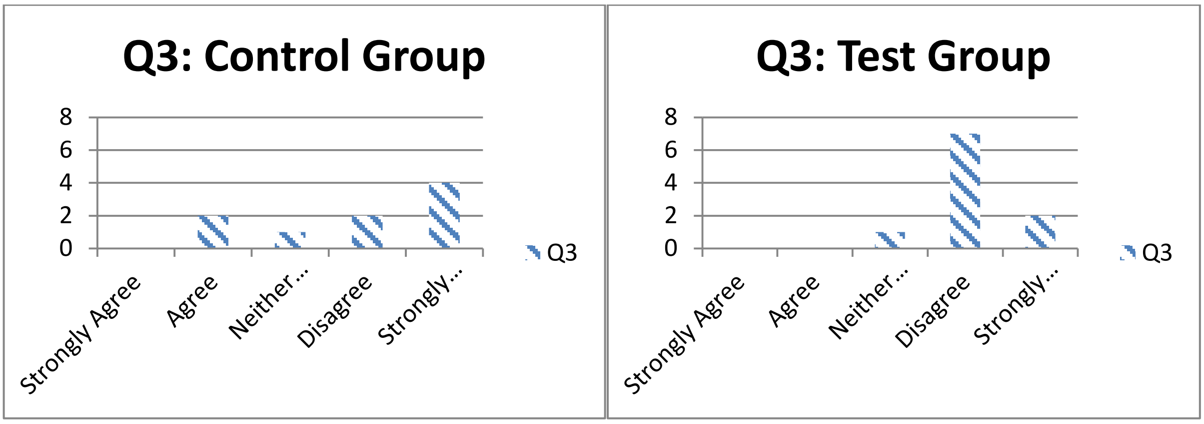


Figure 6-27 Q3 “Creating a SimCon source was challenging” All participants (control and test)

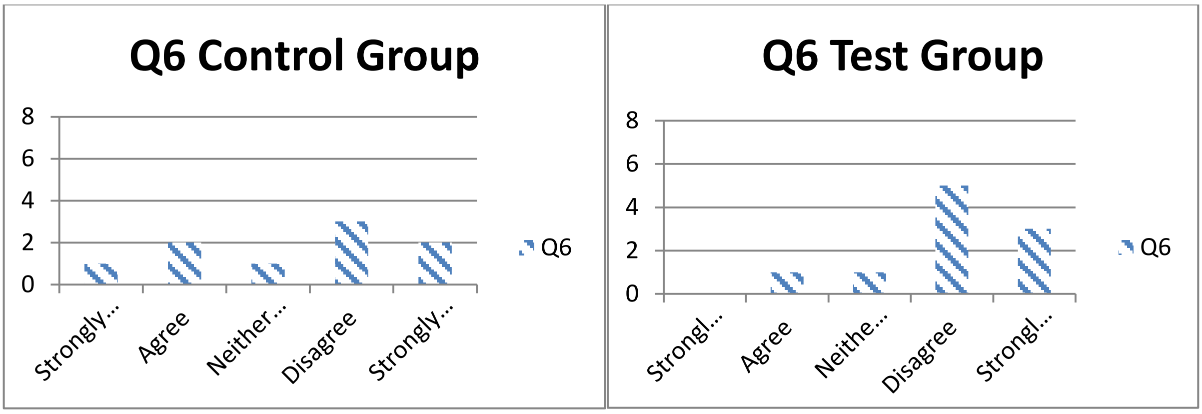


Figure 6-28 Q6 “Configuring an uncertainty area was a challenge” All participants (control and test)

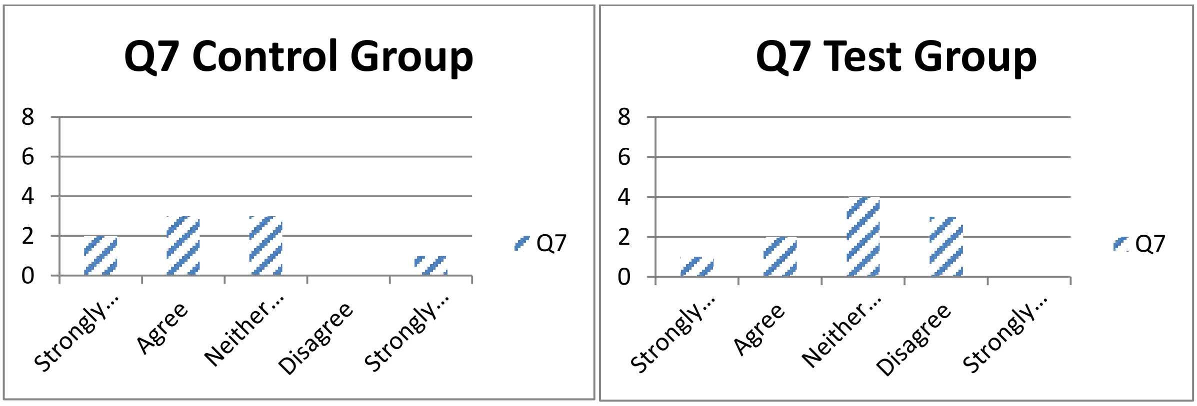


Figure 6-29 Q7 “I found the printed context output helped my analysis of the SBA” All participants (control and test)

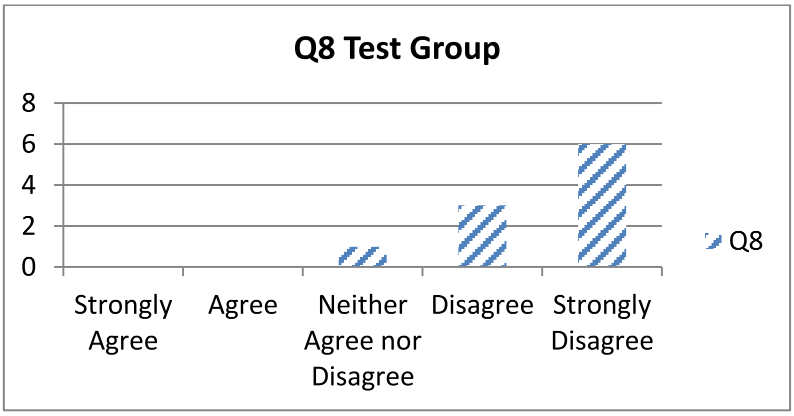


Figure 6-30 Q8 “I thought the visualisation of context did not add much to my analysis of the SBA” All participants

Of particular interest was how the participants found the use of the visual representation of context. For this, the control group was presented with the statement Q7 “I found the printed context output helped my analysis of the SBA” and the test was given this an additional statement Q8 “I thought the visualisation of context did not add much to my analysis of the SBA“.

6.5.4.2.2 Standard Usability Scale

The SUS returned the following scores: For the test group a score of 75 was recorded with a standard deviation of 8.6. For the control group the SUS score was 72 with a standard deviation of 7.6. For a complete breakdown of the scores see Appendix F section 14.5.3.

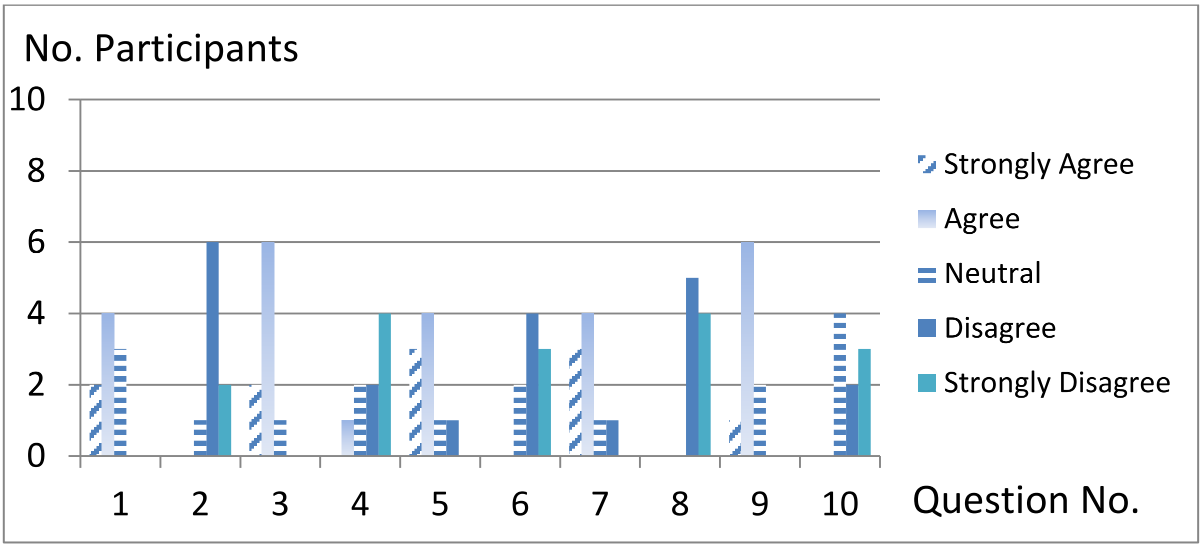


Figure 6-31 Control Group SUS Results

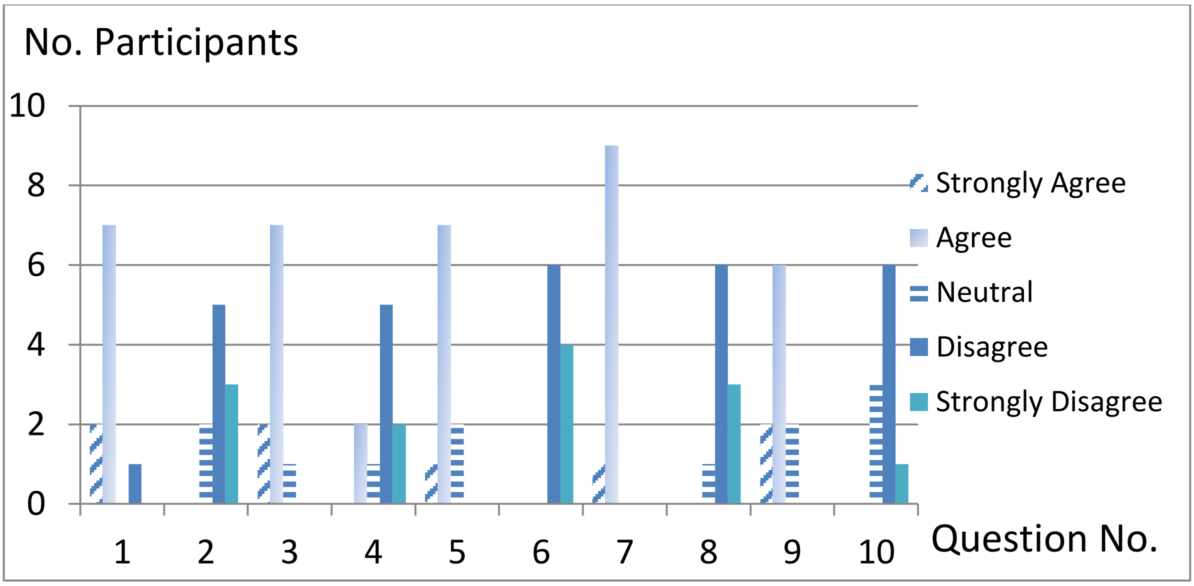


Figure 6-32 Test Group SUS Results

6.5.5 Findings for Non-Expert Participants

6.5.5.1 Observation and Quantitative Metrics

6.5.5.1.1 Correct Positioning of PIRs

Table 6-5 Number of Participants who correctly placed the PIR Sensors: Non-Expert Participants

	Control (No. of participants out of 9)	Test (No. of participants out of 10)
Task 2	1	0
Task 3	5	4
Task 4	3	4
Task 5	1	4

6.5.5.1.2 Average Times to Complete Tasks

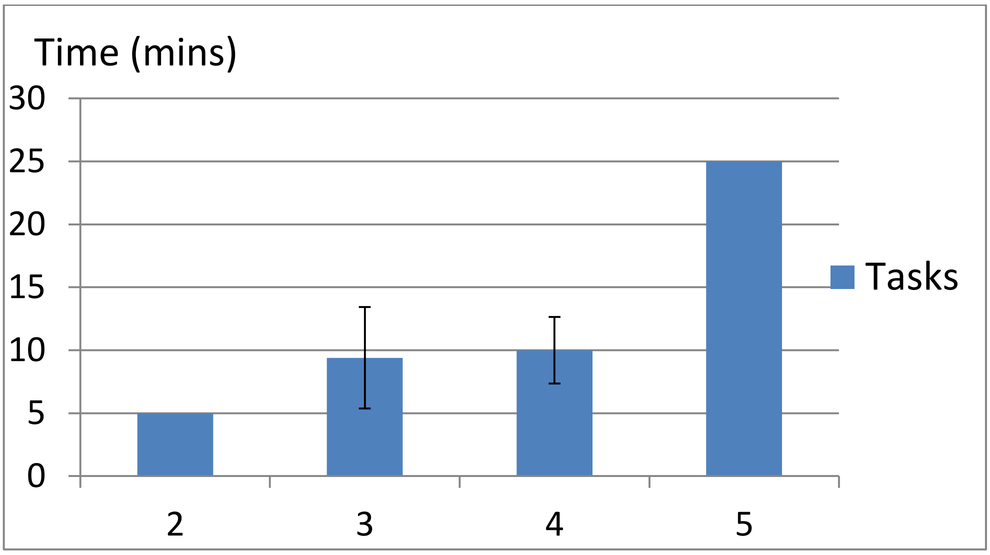


Figure 6-33 Average Time (and Standard Deviation) to Successfully Place PIR and Complete Task: Non-Expert (Control Group)

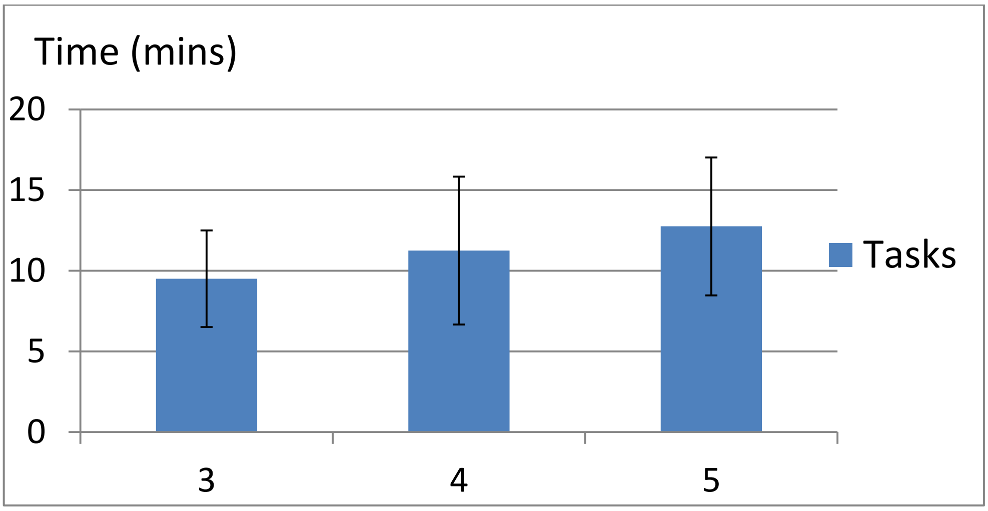


Figure 6-34 Average Time (and Standard Deviation) to Successfully Place PIR and Complete Task: Non-Expert (Test Group)

Total average time for control was 51 minutes with a standard deviation of 6.4. Total time for the test group was 54 minutes, with a standard deviation of 3.3.

6.5.5.1.3 Significant Errors and Comments

There were no significant errors to report; one user caused the interface to crash by not entering in any values for the z position for the uncertainty zone. The time to restart was deducted from the task and a check was put in place to prevent it re-occurring.

Comments made during the experiment are recorded here and are associated with the particular participants number, which can be cross referenced with findings (Appendix F section 12). "A" indicates test group and "B" indicates control. Some noteworthy comments include: (16A) – ""If I bring them together it will be less time" and "Not sure about the uncertainty and what affect it is having"(14A) – "Why is the authorised user moving around so much?"

6.5.5.2 Post-Questionnaire

6.5.5.2.1 Specific Questions about SimCon Tool

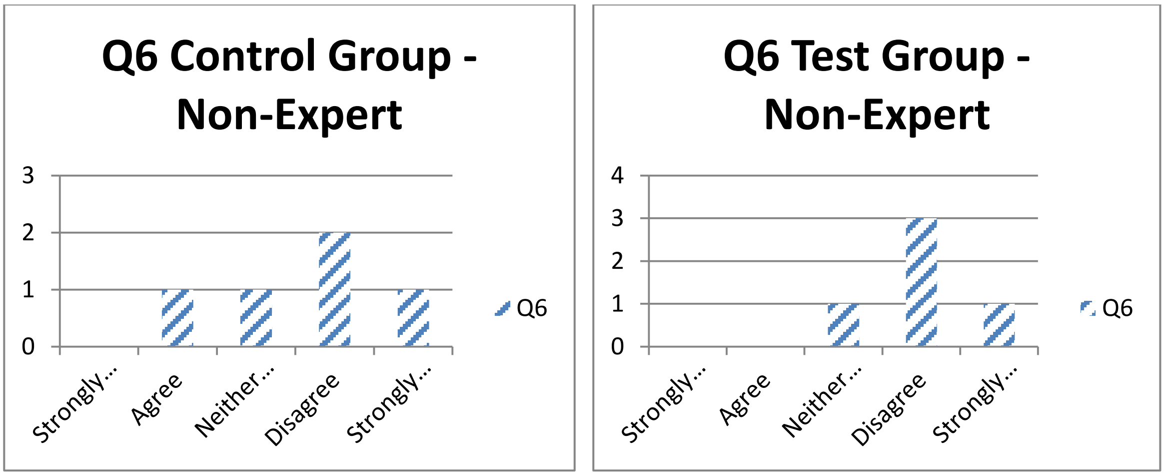
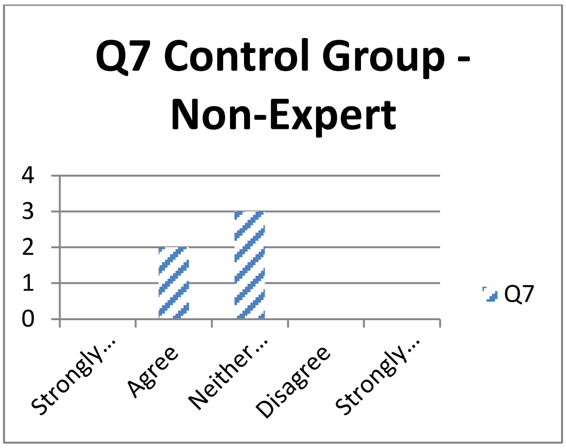


Figure 6-35 Q6 "Configuring an uncertainty area was a challenge" Non-Expert (control and test)



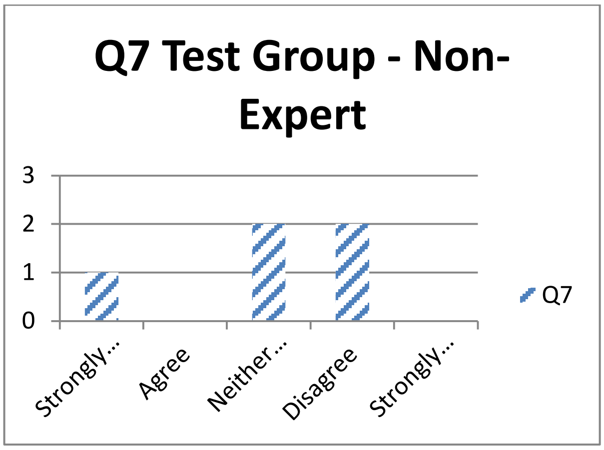


Figure 6-36 Q7 "I found the printed context output helped my analysis of the SBA" Non-Expert (control and test)

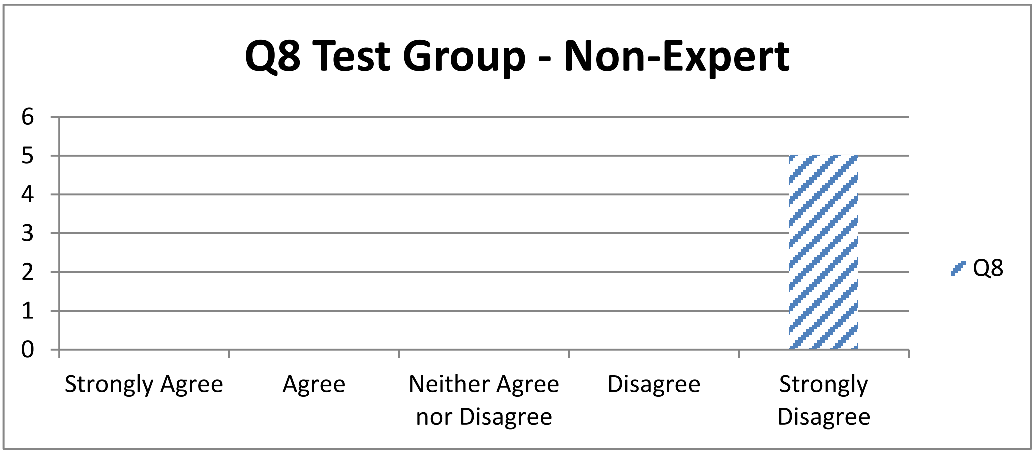


Figure 6-37 Q8 "I thought the visualisation of context did not add much to my analysis of the SBA" Non-Expert

6.5.5.2.2 Understanding

Two questions were asked to ascertain understanding for both control and test groups. These were: **Q9** "What effect did adding the generic uncertainty to Ubisense have on your positioning of the PIR sensors?"**, Q10: "**What effect did adding the uncertainty zone to Ubisense have on your positioning of the PIR sensors?" and one additional question was asked to the test group **Q11:** "How did the visualisation of the simulated context (tagged location) affect your analysis of uncertainty?"

Here we list some of the salient responses (all of which are listed in Appendix F section 12).

Control Group: Q9 (15B): "Closer to the door was better to achieve minimal delay" and Q10: "Locked the door prematurely". Q9 (21B) - "It made me reconsider the position of the sensors so that they would not allow unauthorised access" and Q10:"It made me reconsider position of PIRA due to uncertainty zone." Q9 (24B) "Allowed me to review my initial approach. At first I placed right beside the door area, but decided to move it due to the uncertainty." Q10 (7B) - "Allowed me to define tail-gaiting zones for users "and Q12 - "100% non-expert, found it easy to use and understand."

Test Group: Q9 **(**5A) - "It adds more randomness + real life effects to the experiments to show potential problems if sensors installed in real building" and Q11 (5A)"useful to see as person moving through simulator as textual description would not be clear and it makes it easier to spot potential pitfalls in design". Q9 (22A) – "I needed to move the PIR sensor further away from zone A". Q10 (14A) – "I moved the sensor away from the uncertainty zone " and Q11 "Helped me understand the impact of uncertainty on the application".

6.5.5.2.3 Standard Usability Scale

The SUS returned the following scores: For the test group the SUS score was 76.5 with a standard deviation 5.4. For the control group it was 72 with a standard deviation 8.9.

6.5.6 Findings for Expert Users

6.5.6.1 Observation and Quantitative Metrics

6.5.6.1.1 Correct Positioning of PIRs

Table 6-6 Number of Participants who correctly placed the PIR Sensors: Expert Participants

	Control (No. of participants out of 4)	Test (No. of participants out of 5)
Task 2	0	3
Task 3	3	4
Task 4	2	3
Task 5	2	5

6.5.6.1.2 Times to Complete Tasks

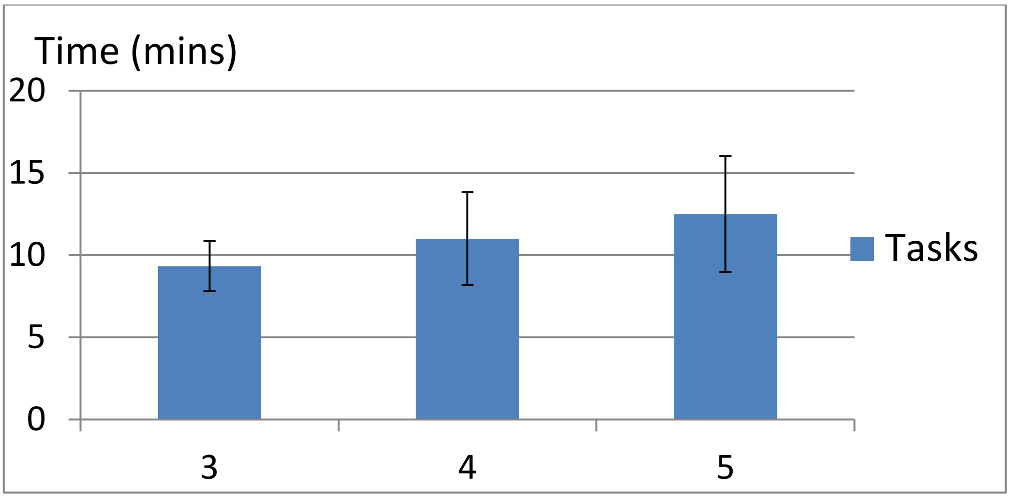


Figure 6-38 Average Time (and Standard Deviation) to Successfully Place PIR and Complete Task: Expert (Control Group)

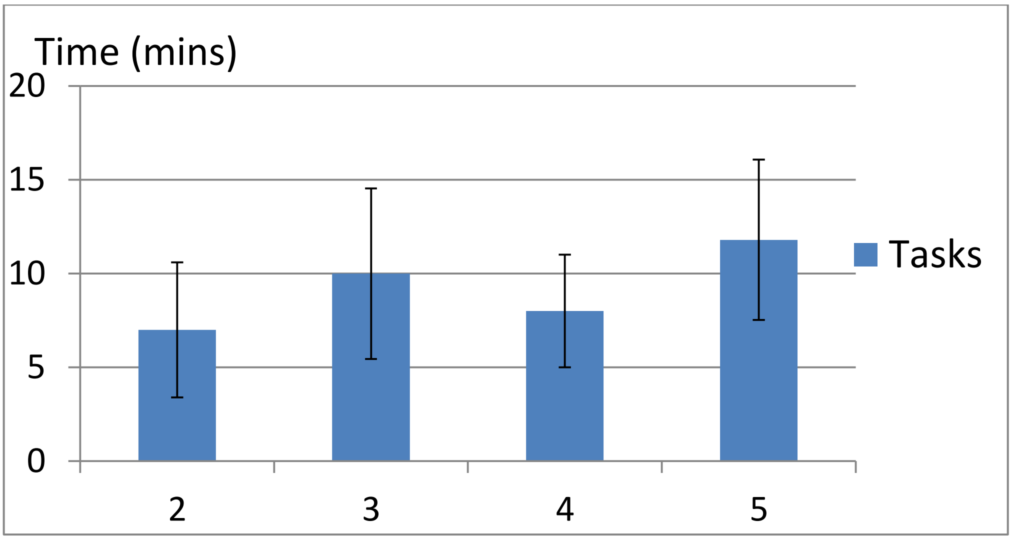


Figure 6-39 Average Time (and Standard Deviation) to Successfully Place PIR and Complete Task: Expert (Test Group)

Total average time for control was 43 minutes with a standard deviation of 5.6. Total time for the test group was 52 minutes, with a standard deviation of 6.5.

6.5.6.1.3 Significant Errors and Comments

There were no significant errors to report; two users asked about the use of GUID instead of ID. Some noteworthy comments include: 9A – “Ubisense is not detecting at right place" 11A – "Why did he stop, means he was falsely identified by the PIR, no?" and "Is there a false positive with the PIR zone?"

6.5.6.2 Post-Questionnaire

6.5.6.2.1 Specific Questions about SimCon Tool

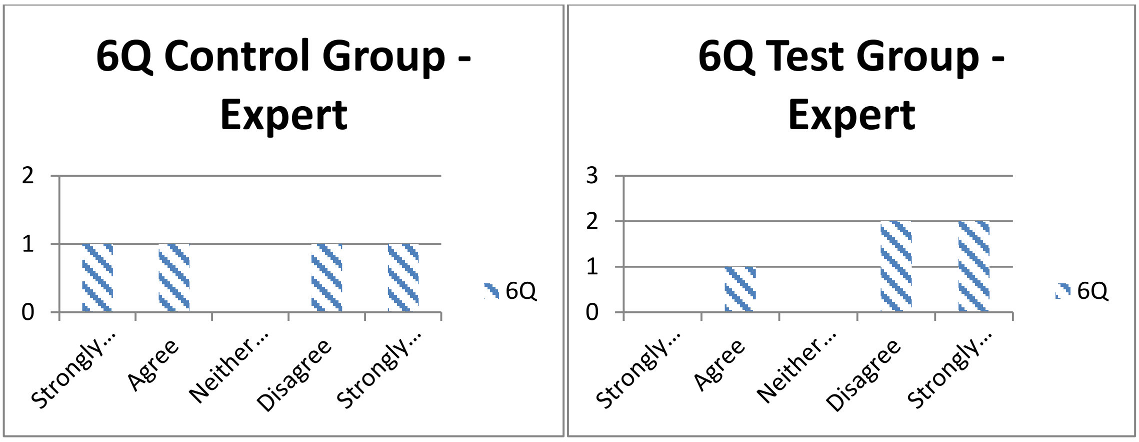


Figure 6-40 Q6 “Configuring an uncertainty area was a challenge” Expert (control and test)

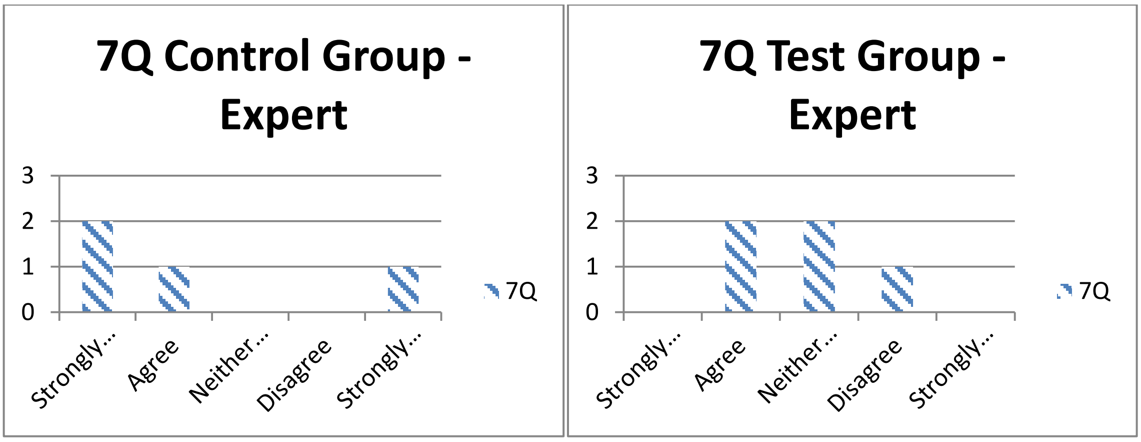


Figure 6-41 Q7 “I found the printed context output helped my analysis of the SBA” Expert (control and test)

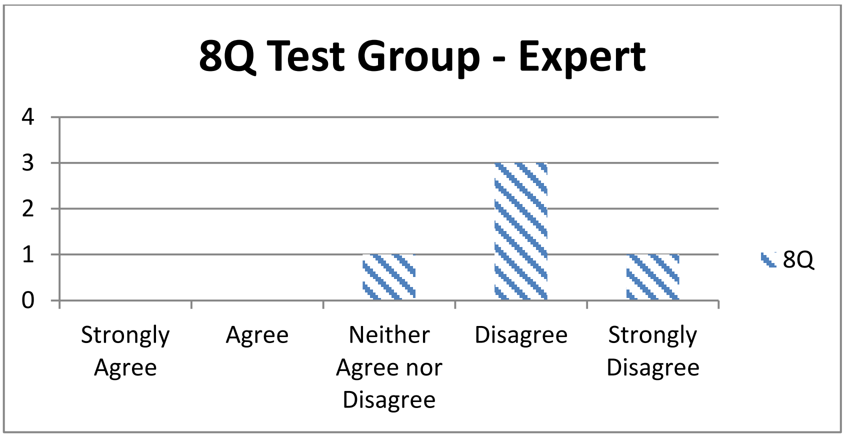


Figure 6-42 Q8 “I thought the visualisation of context did not add much to my analysis of the SBA” Expert

6.5.6.2.2 Understanding

Control: Q10 (6B) “Reinforced my belief that I had placed PIRA properly”. Q9 (8B) – “Uncertainty has to be factored in when placing the PIRSs”. Q9 (17B) – “Introduces challenges with respect to quality of service”

Test: Q11 (19A) - “Real-time shows how much the positions can vary during time user is within zone” and Q12 (19A) “Excellent 3D viz. Perhaps integrate GUI with uncertainty graphs. A time compression factor could speed up experiment (allow > 4 runs)”. Q11 (2A) – “Quickly showed how the sensed location varied from the true location.” And Q12 (2A) “Drag and drop zone placement or re-sizing would be very useful. Colour shading for uncertainty zones.”

6.5.6.2.3 Standard Usability Scale

The SUS returned the following scores: For the test group the SUS score was 75 with a standard deviation 11.6. For the control group it was 72.5 with a standard deviation 7.1.

6.5.7 Interpretation

6.5.7.1 Analysis of PIR positions and Time to Complete Tasks

To meet the objective of task 2 (no uncertainty configured for Ubisense), the correct placement for PIRA was x: 19 and PIRB x: 23. Table 6-4 shows that 1 out of 9 of the control group placed the PIRs correctly from purely reading the instructions. The test group saw a higher rate of success with 3 out of 10 placing the PIRs correctly. Comparing the non-experts with experts, we see that of the non-experts only one control correctly placed the PIRs from reading the instructions (Table 6-5). Of the expert, three of the test group correctly place the PIRs (Table 6-6). Examining the time to complete these tasks, we see that of those who successfully placed the PIRs, the experts took marginally longer to correctly place the PIRs than the one control (Figure 6-33 and Figure 6-39). While the test expert group did have two actual experts (as opposed to intermediate participants), there is no clear reason from the data why the expert test group had the largest number of correct placements.

After task 3, which had the participants place two PIR sensors without any uncertainty configured, 8 of the 9 control group and 8 of the 10 experts had correctly placed the PIRs (Table 6-4). This indicates that the visual representation of the situations made the probability of correct placement much higher than without, as they could in real time see the authorised user wait at the door until the non-authorised user walked past PIRB. The participants also had longer to consider the different situations at this point, which may also have contributed. It also indicates that the visualisation of context with no uncertainty made no noticeable difference between the test and control groups. Comparing the non-experts with the experts (Table 6-5 and Table 6-6), we see that the non-expert control group had only a marginally higher rate of success than the expert (9 out of 10 for the non-expert and 7 out of 9 for the expert) which would indicate the tool is effective for both expert and non-experts for evaluating certain aspects of SBA behaviour like visually analysing the effect of sensor placement with respect to the aforementioned issue of the authorised user waiting for the non-authorised to trigger PIRB.

Task 3 took the non-experts who successfully achieved the task approximately 9 minutes each to complete, with no great variance between the standard deviations of the control and test groups (Figure 6-23 and Figure 6-24). The average time between the expert test and control group was also very close (approx. 10 mins) although the standard deviation for the test was much greater than the control (4.5 mins as opposed to 1.5 mins, Figure 6-38 and Figure 6-39). The cause of this variance is hard to determine and may be related to the higher rate of success for the expert test group in task 2. But no definitive conclusion can be drawn.

Task 4 had both groups configure a generic uncertainty for the entire Ubisense cell and then decide whether the PIRs should be re-positioned. The correct positions did not change from task 3 as the uncertainty was equally applied to the entire Ubisense cell. 5 of the control group and 7 of the test group correctly placed the PIRs for this task (Table 6-4). The control group therefore saw 3 participants change the PIR position to address the additional uncertainty, two non-experts and one expert (Table 6-5). The test group saw only one participant, an expert, change the PIR position to address the uncertainty (Table 6-6). These participants were incorrectly reacting to the additional uncertainty.

The average time to complete the tasks was once again very similar between the non-expert control and test group, at 10 mins and 11 mins respectively, although the standard deviation was slightly larger in the test group, being 4.5 mins as opposed to 2.6 (Figure 6-33 and Figure 6-34). The expert control group took 11 mins with a 2.8 mins standard deviation, whereas the test group took on average only 8 mins with a 3 mins standard deviation (Figure 6-38 and Figure 6-39). The less time taken by the expert control group could be a result of the additional visualisation of context, although as the number of correct placements for the expert test group was lower than that of the non-expert, this would then indicate that the visualisation may have contributed to wrong placement. But with only 1 participant changing placement, no conclusive conclusion should be drawn from this.

Task 5 had both groups configure an additional uncertainty zone on the left side of the zone A. Significantly, 9 of the 10 test group positioned the PIRs correctly, whereas only 3 of the 9 control group positioned it correctly (Table 6-4). This suggests that the visualisation of context made it very obvious that the zone with additional uncertainty was the cause of the security application misidentifying the authorised users. The non–expert control group now saw the least number of correct placements (1) (Table 6-5), whereas two of the expert control correctly placed the PIRs in the control group (Table 6-6). Of the test group, all the experts now correctly placed the PIRS, with the 4 non-experts still correctly placing them.

Comparing the average times of control and test groups for participants who correctly placed the PIRs, we see that the control group completed the task in 16 mins with a standard deviation of 7.6, and the test group completed it in 12 mins with a standard deviation of 4 (Figure 6-23 and Figure 6-24). This would indicate that as well as improving the placement of the PIRs, the visualisation of context also improved the time to complete the task, although as only a small number of the control group (3) completed the task, no definitive conclusions should be drawn regarding improvements in time.

Comparing the time to complete tasks with earlier experiments, we see that on average task 2-4 took 10 mins or less, regardless of whether there was additional visualisation. Task 3 is similar to task 4 in evaluation 2A which also looked at configuring and simulating uncertainty, and evaluating an SBA, and which took on average 8 min. The final task, which had the participants create and configure an additional uncertainty zone and also evaluate the SBA saw times between 11 and 12 min. This task is unique to this experiment, and the additional time over previous tasks must be seen in the context of the additional complexity of the task.

The most significant finding was related to the research question and a positive finding for **$H1_a$** in that the number of users who placed the simulated context sources correctly in the test group was over three times higher than that of the control group.

6.5.7.2 Analysis of Post Questionnaires

6.5.7.2.1 Specific Features of SimCon

The post questionnaires set out to first evaluate a range of statements which were specific to the tasks of the participants. For example, the statement “Creating a SimCon source was challenging”, had two of the control group agree and four strongly disagree. In the test group no participants agreed or strongly agreed with this statement, but only one strongly disagreed, with seven disagreeing. Of particular interest was how the participants found configuring uncertainty. Here the range of answers varied again between the non-expert and expert participants. The non-experts saw one participant strongly agreeing that “Configuring an uncertainty area was a challenge” and two agreeing, whereas only one expert agreed with this statement. This suggests that the configuration of uncertainty presented more of a challenge for non-expert users. The other statements saw all participants generally side on the positive with respect to different features and these answers can be found in the Appendix F section 14.5.1.

Another feature which was of particular interest was the visualisation of context. The control group only had access to textual outputs of context, as opposed to both for the test groups. The number of participants in the test group who disagreed with the statement “I found the printed context output helped my analysis of the SBA” was two, whereas no participant in the control group disagreed with this. Six participants (five non experts) strongly disagreed with the statement “I thought the visualisation of context did not add much to my analysis of the SBA”, which suggests that the visualisation was viewed more positively by non-experts than experts, who rated the textual output highly in the control group.

6.5.7.2.2 Understanding and Comments

Two questions set out to ascertain the level of understanding of the participants. In the non-expert control group the comments were indicative of their level of understanding, statements like “Allowed me to define tail-gaiting zones for users“ demonstrated that one user in the control group did not fully understand the experiment. Some participants chose to leave the comments completely blank, which may also be indicative of a lack of understanding, and the number of participants to leave Q10 blank in the control group was 3 and one “not sure”. Only one participants in the test group left Q10 blank. Others within the control group did demonstrate understanding and the participants who placed the PIR on position x: 15 commented “Allowed me to review my initial approach. At first I placed right beside the door area, but decided to move it due to the uncertainty.” The majority of statements though reflected the fact that the non-expert control group largely got the placement of the PIR wrong in Task 5 (Appendix F section 14.5.2).

The test group made statements which reflected an improved understanding of the effect uncertainty was having on the SBA with statements like “It adds more randomness + real life effects to the experiments to show potential problems if sensors installed in real building” in response to

Q9. The test group also had the ability to commented on the visualisation in Q11, and statements were all positive, for example "useful to see as person moving through simulator as textual description would not be clear and it makes it easier to spot potential pitfalls in design" and "Helped me understand the impact of uncertainty on the application". Comments during the experiment also indicate that the visualisation was providing clues as to the effect of uncertainty with one test participants commenting "Why is the authorised user moving around so much" in reference to the visualisation of the Ubisense location context and the effect of uncertainty.

In the expert group the questionnaires on understanding were also revealing. For example, one intermediate user in the control group placed both PIRs on 17 and 25 from the beginning, and never moved them. It is hard to understand the reasoning behind this, although they did comment for Q10 "Reinforced my belief that I had placed PIRA properly". Other participants in the control group demonstrated understanding of the issues, for example "Introduces challenges with respect to quality of service" and "Uncertainty has to be factored in when placing the PIRs", but the statements did not specifically say what was required with respect to the placement of the PIRs.

In the expert test group statements once again reflected the improved understanding due to the visualisation with comments like "Real-time shows how much the positions can vary during time user is within zone" and "Quickly showed how the sensed location varied from the true location." One participant with a background in Ubicomp also gave some recommendations for improvement "Excellent 3D viz. Perhaps integrate GUI with uncertainty graphs. A time compression factor could speed up experiment (allow > 4 runs)". Another participant also suggested "Drag and drop zone placement or re-sizing would be very useful. Colour shading for uncertainty zones."

6.5.7.2.3 Standard Usability Scale Scores

The average SUS for all control participants was 72 and for the test group 75 with a similar standard deviation between scores of 8.6 and 7.6 respectively. According to Bangor et al. systems which score above 70 on the SUS scale can be classified as having good usability (Bangor, Kortum et al. 2009).

This indicates that both the control and test had a high rate of satisfaction with SimCon, with only a 3 percent difference between those who had access to the visualisation and those who did not. The non-expert test users had a marginally higher rate of satisfaction than the control, 76.5 to 72 and 75 to 72.5, although the deviation is also slight higher at 8.9 to 7.1 This could reflect the higher rate of satisfaction the non-expert users had with the visualisation of context (Q8 above).

6.5.8 Conclusion

The major contribution of this evaluation is that it has demonstrated that the implemented components of the SimCon model, which are core to the research objectives of this thesis, e.g. the SimConfig to support configuration of uncertainty in simulated context sources and the SimConGen and the SimConViz which support the evaluation of SBAs, are satisfying for SBA developers, support efficient configuration and evaluation and are finally effective for improving the SBA design. Satisfaction has been demonstrated through the evaluation of the use of the SimCon tools by a range of SBA developers with different skills, who then rated their experience using the SUS and other questionnaires related to satisfaction. The level of efficiency of the tools has also been evaluated through metrics on the time to complete tasks related to configuring and evaluating. This demonstrates that between users who successfully achieve the tasks, those with additional visualisation completed them in less time on average than those who don't, although due to the small number in the latter category, no definitive conclusions about efficiency can be made. It also demonstrated that as the complexity of the task of configuring and evaluating increased there was a corresponding drop in the efficiency for all participants on average. Finally, the effectiveness of the approach to simulating uncertainty, in particular, with respect to the visualisation of the simulated context, has shown that using the SimCon tools resulting in a threefold improvement of the placement of simulated context sources to address the issue of varying levels of uncertainty in location context.

6.6 Comparison of Evaluation Results against Objectives

This chapter set out to address objective 3 of this thesis which is presented here:

3. Implement and evaluate the resulting model to determine if it has met the key requirement of being usable by SBA developers during configuration of user driven simulated context sources and during the evaluation of SBA's when faced with varying levels of uncertainty. The usability evaluation is further broken down into four sub-objectives. These are that the context simulation model supports:

    SO1: Usable configuration of uncertainty in user driven simulated context sources.

    SO2: Evaluation of SBA through simulation and visualisation of uncertainty in context.

    SO3: Usable configuration of user driven heterogeneous context sources.

    SO4: Evaluation of SBA through simulation and visualisation of heterogeneous contexts.

With respect to SO3, it has been shown through the formative and summative evaluation that SImConfig supports SBA developers in the task of creating, placing and configuring three types of simulated context sources for generating user driven location context (presence, proximity and coordinate). Efficiency was, on average, shown to improve in evaluation 1B, 2A and 2B when repeating similar tasks. Satisfaction was also high with respect to specific aspects of the configuration process, with users generally rating the tasks as easy or very easy in Evaluation 1B. Evaluation 3 users on average either disagreed or strongly disagreed that this task was challenging with the test group rating it on average better than the test. The average SUS rating above 70 also indicates a level of satisfaction with the overall approach. Evaluation 2C reported difficulty when creating SimCon sources and the need for technical assistance to complete tasks and this is reflected in a much lower average SUS score. As SimCon is intended for SBA developers with software development skills, this objective is said to have been met.

With respect to SO1, the configuration of uncertainty alone was shown in evaluations 2A and 2B to not reduce efficiency on average, although tasks of configuring uncertainty took place after tasks which did not involve configuring uncertainty, so familiarity with the tool may have had an influence on the efficiency for those tasks. In evaluation 3 it was found that when configuring additional zones of uncertainty, efficiency was reduced on average. The additional complexity of this task, which involved also evaluating the SBA, is the most probable cause of the reduced efficiency and cannot be solely attributed to the configuration process. Evaluation 1B demonstrated that users had no great difficulty configuring uncertainty, rating the process as either easy or very easy and similarly in evaluation 3 13 of the 19 users disagreed or strongly disagreed that configuring uncertainty was challenging, again with a slight difference between the test and control groups. Combined with the SUS score above 70, this objective is also said to be met.

With respect to SO2 and sO4, Evaluation 2B, 2C and 3 have shown that SBA developers have been able to use simulated context from multiple SimCon sources to evaluate an SBA and that the approach taken to simulating and visualising uncertainty in context demonstrates increased understanding of the effect of uncertainty in context on an SBA under test. This is particularly evident in evaluation 3, where users with additional visualisation of context with uncertainty exhibited a threefold improvement in the placement of simulated context sources to meet the requirements of the SBA under test. Combined with the average SUS score above 70 for the final summative evaluation, both of these objectives are also said to be met.

6.7 Summary and Conclusion

This chapter described the evaluation of the SimCon simulated context model, to determine its level of usability for SBA developers when evaluating SBAs faced with varying levels of uncertainty in context. The final summative evaluation and the previous formative evaluations have indicated that the SimCon model provides levels of efficiency which improve with familiarity, when creating, placing and configuring heterogeneous simulated context sources which generate heterogeneous contexts for SBA developers. The final summative evaluation has shown that the visualisation of the simulated context has also been shown to improve the efficiency when addressing the issue of uncertainty in SBA design over users who did not have this visualisation. The approach has also shown high levels of satisfaction with an average score above 70 in the final summative evaluation. The summative evaluation further validated this claim by demonstrating a threefold improvement improvement of sensor placement to meet the requirements of an SBA developer, which is attributable to an improved understanding by the SBA developer of the issue of uncertainty in location context due to the SimCon approach to visualising uncertainty in context. The SimCon model has therefore met the research objectives outlined in section 1.3.3 and contains feature which advance the state of the art. The following chapter concludes this thesis by outlining the specific contributions that have been made and explores future work to extend this research.

7 Conclusion

This chapter examines the extent to which this thesis has met the objectives and addressed the thesis statement set down in Chapter 1. This is followed by a description of the specific contributions of this work and an exploration of possible future work.

7.1 Objectives and Achievements

The research question this thesis set out to evaluate was *"What level of usability is achievable in a VR- based context simulator that supports creation and configuration of simulated context sources, and their associated uncertainty, for the evaluation of SBAs in a manner that is flexible, extensible and leverages other models in the building life cycle."*

The following objectives stemmed from this question:

7.1.1 Objective 1:

Conduct a state of the art review of current approaches to supporting SBA developers to evaluate SBAs through the use of simulation, in particular when faced with varying levels of uncertainty in context information. From this review, identify the key requirements for such a platform through analysis of the strengths and weaknesses of current approaches.

7.1.2 Objective 2:

Design a context simulation model which supports the creation and configuration of simulated context sources that generate user driven simulated context and which supports SBA developers evaluate SBA's when faced with varying levels of uncertainty in context. The model should be easily understood and manipulated for this task by SBA developers. The model must also be interoperable with existing standards to support integration with other tools involved in the building life cycle and thereby to leverage the models produced with these tools.

7.1.3 Objective 3:

Implement and evaluate the resulting model to determine if it has met the key requirement of being usable by SBA developers during configuration of user driven simulated context sources and during the evaluation of SBA's when faced with varying levels of uncertainty. The usability evaluation is further broken down into four task-based sub-objectives. These are that the context simulation model supports:

- SO1 Configuration of a variety of heterogeneous user driven simulated context sources.
- SO2 Evaluation of SBA using the simulation and visualisation of heterogeneous context sources.
- SO3 Configuration of uncertainty in user driven simulated context sources.
- SO4 Evaluation of SBA using simulation and visualisation of context sources with various uncertainty configurations.

Objective 1 was addressed in Chapter 2 and Chapter 3. Chapter 2 examined some of the key concepts related to this thesis and the background of these areas. This included the concept of context and the uncertainty in context information introduced during context acquisition. It also explored the area of building and sensor modelling, as both are related to smart buildings and context simulation. Finally, it gave an overview of usability evaluation techniques as background to the evaluation methodology adopted. In Chapter 3, a set of criteria were developed by which to analyse the state of the art of context simulators. These included the need for a flexible model in order to support the variety of potential situations an SBA may be required to function over, i.e. the range of buildings and the range of types of context generated by low level context sources. It also highlighted the need for context simulators to be easily configurable by the SBA developer so that they can vary the level of uncertainty in simulated context during an SBA evaluation project. The need for context simulators to support the evaluation process through easy-to-use data visualisation and analysis tools was also identified. Finally, due to the complexity of modelling the building to support interactive context simulation, the need to an approach which integrates the context simulators into the building life cycle process via standardised models was highlighted as a novel requirement that would boost the usefulness of context simulators for SBA development required.

A range of context simulators from the literature were then examined in detail against these criteria. These were classified as either interactive, i.e. ones which support user interaction with a simulated smart building and simulated sensors, or non-interactive simulators where human behaviour was solely simulated by pre-programmed automatons or bots. From this analysis, and building upon a framework developed by O'Neill (O'Neill 2011), a set of features of both interactive and non-interactive context simulators were identified culminating in a breakdown of the strengths and weaknesses of the different approaches against these criteria. This revealed that the strengths of using a context simulator for evaluating context-aware applications SBA are that they are flexible and extensible, they are scalable, and they are less expensive than field based-evaluation techniques.

The weaknesses of current context simulators are the lack of evaluation of the level of usability of the context simulators for application SBA developers; the lack of analysis tools to support evaluation of the issue of uncertainty; the lack of leverage of use of standardised building models and the lack of flexibility in interactive simulators for supporting different context simulation approaches. From these findings the following list of requirements were developed for a context simulation model that addressed Objective 2 of this thesis:

- **Requirement R1:** The context simulation model should be flexible, so as to support different context simulation approaches to simulating heterogeneous context source types which generate heterogeneous types of context, and extensible, so as to support context simulation with additional uncertainty.

- **Requirement R2**: The context simulation model should be usable by SBA developers to achieve the goal of evaluating SBAs which must deal with varying levels of uncertainty in context. This requires that the model be both usable during configuration and also during the evaluation process by providing tools to support SBA analysis.

- **Requirement R3**: The context simulation model should be interoperable with existing standards in building and sensor modelling.

The SimCon model was then designed, therefore, to address these requirements, thereby bringing several novel characteristics into the state of the art in context simulators. The features will now be explored with specific reference to the state of the art.

7.1.4 Comparison to State of the Art

With respect to **R1**, the SimCon model through its design and implementation has demonstrated itself to be flexible by supporting many of the existing features of both interactive and non-interactive context simulators. Table 7-1 gives a comparison of the features supported by SimCon compared to other interactive simulators. It should be noted than not all of these features have been evaluated with respect to usability (only those addressed in Chapter 6), with many being demonstrated through their application as detailed in the implementation in Chapter 5.

Table 7-1 Comparison Framework – SimCon v Interactive Simulators

Interactive Context Simulators		2001	2002	2005	2007	2008	2009	
		QuakeSim	UbiWise	3DSim	eHomeSimulator	V-PlaceSims	Simulation Framework in Second Life	SimCon
Context Simulation	Clocking Mechanism	i	i	i		i	i	i
	Extensible Modular/Layered Architecture	i	i	i	i	i	i	i
	Grid Based				i			i Zone
	Play Backs						i	i
	Statistical Model							i
	Mathematical Formula						i	i

	Parametric Simulation					c				i
	Bot driven simulation							i		i
	*User Driven Simulation			i	i	i	i	i	i	i
	*Simulated within VR Engine			i	i	i		i	i	
	Uncertainty Simulation	Communication								i
		Value (Precision)							i	i
	Hybrid Simulation			i	i	i				i
	Network Simulation									
	Integrate Models from Other Simulation Tools									i
Simulated Context Sources Modelling	Type			i	i	i	i	i	i	i
	Value			i	i		i	i	i	i
	Rate								i	i
	Placement					i	i	i	i	i
	Variation	Time Series/Mathematical Formula								i
		Probability Distribution								i
		Uncertainty								i
	Sensor Specific Formats									i
Building Model	Walls			i	i	i		i	i	i
	Zones/Areas					i	i	i		i
	Others			i	i	i		i	i	i
	Purely a Visual Model			i	i	i	i		i	i
Deployment Environment	Scalability	Building		i	i		i	i	i	I
		Context Sources		i	i		i	i	c	i
		Devices					i	i		i
		Bots						i		i
	Heterogeneity	Building		i	i		i		i	
		Simulated Context Sources	Low Level	i	i		i		i	i
			High Level	i	i	i		i		
			Location	i	i	i	i	i		i
			Other			i	i			i
		Devices			i	i	i	i		i
		Bots								i
	Low Investment (Compared to physical)									
Example Models Provided	(Generated) Context Model							m		m Con-tum
	Context Source/Sensor Model					s		m		s
	World/Building Geometry Model							m		s
	Environment/Error Model									s
	Agent/Bot Model							m		m
	Device/Service Model					s	i	m		

Configuration Tools	Building	Walls	i	i			i	i	
		Zones/Areas				i	i		
		Other (windows, etc.)	i	i			i	i	
	Context Sources	Type				i	i	i	i
		Rate						i	i
		Placement				i	i	i	i
		Transmitters/Receivers					i	i	i
		Time Series							i
		Uncertainty							i
	Devices			i		i	i		
	Bots		c	c			i		
Testing	Repeatability				i				i
	Scalability		i	i					i
	Environment Visualisation	2D				i			i
		3D	i	i	i		i	i	i
Analyses Tools	Context Monitoring	Real Time Visualisation			i			i	i
		Graph							
		Text				i	i		i
		Streamed Events							
	Logging								
Simulator Evaluation	Performance								
	Usability								e
	Realism/Fidelity								
	Case Study		e	e		e	e	e	e
Legend									
Not provided/discussed									
Conceptual		c							
Modelled (based on standard model)		m(s)							
Implemented		i							
Evaluated (partially)		e(p)							

As can be seen from the table, SimCon models all of the context source properties of existing context simulators, i.e. context source type, values sensed, context acquisition rate and physical placement of context source (see also Table 3-1). It also supports the simulation of a number of different types of simulated context, including three types of location context (presence, proximity and coordinate) and also temperature. Only InSitu has explicitly looked at simulating this range of location context, and SimCon is unique in interactive context simulators in demonstrating its support for simulating other types of context, like temperature.

SimCon also demonstrates flexibility in the number of approaches to modelling variation in the generated values of context information (for example, in location coordinates, or temperature) through the use of different simulation model. These include playback and statistical models based on physical readings, the use of mathematical functions, and bot and user driven variation in location. It also supports generation of context values using external simulation models, which is an implemented feature unique to SimCon and is discussed only at a conceptual level for the Context Simulation Tool (Reynolds, Cahill et al. 2006). Supporting this range of contexts and context simulation approaches gives SBA developers great flexibility in the types of SBAs they can evaluate, and also between methods which require different levels of investment of time and resources to undertake, for example, using a library of playback values to generate simulated context or developing an external simulation model for generating simulated context. This way the SBA developers can choose the level of fidelity of the simulation as appropriate to the evaluation task. The SBA developers can also choose between using interactive simulation and non-interactive simulation.

Also with respect to **R1,** the SimCon model addresses modelling uncertainty in low level context to support simulation, i.e. the context precision and timeliness properties. Precision is modelled through the use of probability density functions, timeliness through a combination of rate and also delays. SimCon also demonstrates the use of Gaussian distribution to introduce variation to simulate uncertainty in precision of sensors, an approach which is discussed only at a conceptual level by other context simulators (Reynolds, Cahill et al. 2006). Also, SimCon supports configurable zones of uncertainty for context sources which can be used to introduce uncertainty into wireless sensors used for determining location. This goes beyond the current state of the art for interactive context simulators.

With respect to **R2** SimCon has addressed usability both at the configuration stage and also at the evaluation stage. The SimCon model has been developed to support SBA developers in the task of creating, placing and configuring simulated context sources with uncertainty. Therefore, through its design it has made the process of configuration as efficient and as satisfying as possible. Currently very few context simulators have looked at developing tools for this purpose, and while some have commented on its necessity either explicitly (Reynolds, Cahill et al. 2006) or that they were aiming to develop such tools (Prendinger, Brandherm et al. 2009) only SimuContext has presented a tool which supports configuration of uncertainty in simulated context and as it is a non-interactive context simulator, little consideration is given about how the configuration of context source uncertainty is linked to its placement in the environment. SimCon has addressed these issues through the implemented interface "SimConfig", introduced in Chapter 5.

The SimCon model must also be effective in supporting the evaluation of SBAs faced with varying levels of uncertainty. SimCon supports the simulation of context for this purpose using the aforementioned simulation models. It also support analysis of simulated context through the modelling of visual representations of context, and the ability to record and playback the movements of avatars and the simulated context generated during an evaluation run. By integrating uncertainty simulation into this process, SimCon is unique among interactive context simulators in providing this type of experimental analysis of visualised location context with uncertainty. The SimCon model also supports analysis by providing a representation model to visualise external simulations which have been integrated with SimCon to support interaction of users. No current interactive context simulator has looked at integrating external radio propagation models with the simulation of tags to provide both simulation of changing radio propagation and also its visualisation. This type of visual feedback can give developers insights into how received signal strength changes as a transmitter moves about an environment.

Finally, to address **R3** the SimCon model has been based on two existing standards for describing buildings and sensors, IFC and sensorML respectively. By addressing interoperability with IFC the potential to import existing IFC models for interactive simulation has the potential to reduce the time and effort involved in constructing such a model solely for this purpose. It also has the potential to further support parametric simulations as properties modelled in IFC, for example materials and wall thickness, which can be used as inputs into simulation of context values like received signal strength between transmitters and receivers as these are influenced by these properties. Languages like VRML, supported by UbiREAL, do not contain such properties. This type of integration is seen as a necessary step towards developing a smart building context simulator as the process of simulating such environments is complex, and placing all the onus of modelling on the SBA developers may be unrealistic. SimCon as such is the first context simulator to consider its integration into the wider building life cycle. Also, as IFC does not currently provide enough detail in its modelling of sensors, the SimCon model has been developed using the information delivery manual, a standard used in the building industry for defining information exchanges that need to be supported by building information models (BIM), such as IFC. This work therefore offers a route to integrating the SimCon model and the approach taken to context simulation into a mainstream BIM.

In Chapter 5 the SimCon model was implemented to evaluate both **R1** and **R2**. The SimCon implementation has three main components, SimConfig, SimConGen and SimConViv. In particular, these were developed to support the evaluation of the SimCon model usability. Chapter 6 presented this evaluation of the SimCon implementation against the stated sub objectives of Objective 3. There were six evaluations in all, the first five (evaluation 1A, B and 2 A, b and C) being formative and resulting in the development of version 3 of SimCon and a final summative evaluation (evaluation 3) of that version. Each evaluation set out to ascertain the efficiency with which users completed tasks using SimCon, the effectiveness of SimCon (i.e. could they achieve the goals of the tasks) and level of satisfaction of users with the SimCon approach. The majority of evaluations (all but Evaluation 2C) examined SBA developers with the basic technical skills necessary to create an SBA (i.e. software development skills). Within this cohort, the SBA developers ranged from those with no experience with either or both context-aware application development and location systems, to those with expert knowledge of either or both of these technologies. Evaluation 2C looked at non-technical users who studied architecture.

These evaluations are the first systematic attempt to assess usability of a context simulator for SBA developers, or indeed any class of context aware application developer. This work therefore provides a novel usability assessment framework and a baseline against which other context simulators can now be compared for the purpose of SBA development, and potentially other context simulator applications.

The evaluations have also contributed to the understanding of the usability to SimCon model through assessment of three factors: effectiveness, user satisfaction and efficiency. The evaluations have demonstrated the effectiveness of SimCon for supporting context simulators by potential SBA developers with a range of skills, from those with no experience with either or both context-aware application development and location systems, to those with expert knowledge of either or both of these technologies. The summative evaluation has shown that the configuration of context source uncertainty and visualisation of that uncertainty made SBA developers better able to position simulated context sources to meet the requirements of a specific SBA than otherwise. The implementation and evaluation of the SimCon model therefore demonstrates that context simulation will provide benefits to users in clearly informing them about SBA design and configuration issues related to uncertainty in context, something which no other context simulator has yet demonstrated.

Secondly, SimCon has been evaluated to determine the levels of user satisfaction with the tool. By addressing issues identified in formative evaluations as the source of lower levels of satisfaction, the final summative evaluation of the SimCon tool demonstrated high SUS scores for SBA developers.

Thirdly, SimCon has been evaluated to determine levels of efficiency to complete tasks. The formative evaluations suggest that repeating similar tasks related only to configuration improves efficiency for users. The final summative evaluation revealed that the difference in times between a task which involved no configuration of uncertainty and one that required a generic uncertainty to be configured were only marginally different in average time to completion. When the task required configuring an additional uncertainty zone and then evaluation of its effect, the average time increased correspondingly, thus efficiency is reduced as task complexity increased. When comparing those who achieved the task and had additional visualisation of context to those who did not have this visualisation, there was an improvement in the level of efficiency.

7.2 Contribution to the State of the Art

The major contribution of this thesis is the SimCon model which addresses the complexity of developing and evaluating SBAs when faced with varying levels of uncertainty in context generated by sensors. The SimCon model addressed this complexity through the development of models which abstract from the underlying complexity of the causes of uncertainty, so that SBA developers may configure the properties of the simulated context sources and then evaluate their SBA in a manner which is satisfying and most importantly effective. Only one other existing context simulator, InSitu (section 3.4.1.9), has been examined to determine its level of usability, but as its focus is on specifying situations to determine unwanted behaviour, it does not address the range of SBA developer tasks assembled here from the literature. Conducting usability evaluation of the subsets of tasks that the SimCon model addresses is an important contribution since improving the level of usability of context simulators, in particularly when dealing with practical problems around context uncertainty, is key to integrating SBA development as a mature part of the future smart building lifecycle. In particular, the SimCon model has contributed to this in two key respects.

The first is in the task of creating, placing and configuring simulated context source with varying levels of uncertainty. The causes of uncertainty in context-aware system behaviour are often difficult to ascertain, even for experts in sensing systems. For SBA developers who may not be experts in these systems, and who must develop SBAs which are robust in the face of varying levels of uncertainty, tool support for the configuration of uncertainty is a key requirement. The SimCon model allows the configuration of the uncertainty associated with a context source model in a way that enables the precision characteristic of that uncertainty to be replicated by the output of the simulated sensor during an interactive evaluation. It also supports configurable zones of differing imprecision for location context sources and a toolset for creating, configuring and placing of these zones, so that they can be related to the building geometry and the position of avatars when generating user driven location context. While a number of non-interactive context simulators have discussed modelling uncertainty, for example, the Generic Simulation Tool (section 3.4.1.6), and for configuring uncertainty, for example, SimuContext (section 3.4.1.4) neither of these have been evaluated in use by SBA developers to determine their level of usability. The only interactive context simulator to discuss uncertainty is the Simulation Framework in Second Life (section 3.4.2.6), but does not provide any models and only discusses the requirement for usable tools to support the SBA developer with configuring the simulator. It also presents no discussion of additional tools to support the evaluation process.

The second part of the contribution is the support provided by applying the SimCon model for the task of evaluating the SBA. Through the combination of the SimConGen simulated context generator, and the SimConViz context visualisation tool, the SimCon model has been shown to support SBA developers during the process of SBA evaluation by generating simulated context with varying, configured levels of precision and visualising the resulting uncertainty in context. A key finding of the research, through the summative evaluation, is that SBA developers ranging from novice to expert who were presented with the visualisation of varying levels of precision in location context, when placing simulated context sources to meet the requirements of an SBA under evaluation, exhibit a threefold improvement of placement over users who did not have this visualisation. This provides evidence on the importance of visual feedback in SBA development, whereas evaluation in providing such feedback during SBA usage indicates that in that role it is more of a distraction (Rukzio, Hamard et al. 2006; Lim and Dey 2011).

Finally, the SimCon model has demonstrated greater flexibility in the number of approaches it supports to simulating context, enabling SBA developers choose between the best suited context simulation approaches to meet the needs of their particular evaluation. To date, the most flexible simulator is DiaSim (section 3.4.1.8), which supports a number of the reviewed approaches to context simulation. It does not address interactive context simulation though and no existing interactive context simulator presents methods for simulating context other than location context. Also, only the Context Simulation Tool discusses at a high level the use of external models to support simulation, and SimCon is unique in its support for this as shown in its implementation.

The minor contribution of this thesis is the integration of the context simulation model into a standard model for describing buildings called Industry Foundation Classes. This is important because the user driven simulation of context in buildings is a complex task which can require visual representations of the building, properties of building elements and models of sensors which support context simulation. The main advantage of this is that the potential exists for importing existing building models directly into SimCon, reducing the time to develop interactive VR environments and improving consistency with other modelling activities in the BLC. IFC also supports modelling aspects of the building which may affect the value of simulated contexts. This type of parametric simulation can result in context simulation which is a closer match to how the physical system being simulated behaves in a particular environment. Therefore, integrating IFC into the context simulation approach has the potential of improving the realism of the simulations, i.e. how true the simulation is to how the physical systems behave.

7.2.1 Peer Reviewed Publications

This work has been peer-reviewed and has been accepted to a journal and a number of peer review conferences. A list of relevant author publications appears at the front of this thesis, the most significant of which are noted here:

McGlinn, K., O'Neill, E., Gibney, A., O'Sullivan, D. Lewis, D.(2010). "*SimCon: A Tool to Support Rapid Evaluation of Smart Building Application Design using Context Simulation and Virtual Reality*." Journal of Universal Computer Science. V 16(15): pages 1992-2018, 2010.

> Describes the SimCon model (Version 2) and the formative evaluation (1B, 2A and 2B). Also describes the integration of the SimCon model into IFC.

McGlinn, K., R. Brennan, O'Sullivan, D. Lewis, D. (2011). "*The SimCon Generator: An Interactive Context Simulator for Rapid Evaluation of Smart Building Applications using Virtual Reality*." 8th International Workshop on Managing Ubiquitous Communications and Services, pages 50-55, Seattle, USA, IEEE, 2011.

> Describes the exploration of the Ubisense system, identifying an approach to simulating uncertainty for the purposes of rapidly modelling for simulation.

McGlinn, K., E. Corry, O'Neill, E. Keane, M. Lewis, D. O'Sullivan D. (2010). "*Monitoring Smart Building Performance Using Simulation and Visualisation*". In Proceedings of Ubiquitous Computing for Sustainable Energy (UCSE2010), Ubicomp 2010 Workshop, Copenhagen, Sept 25th, 2010.

> Describes the use of SimConViz for supporting evaluation of building performance.

McGlinn, K., E. O'Neill, Lewis, D. (2008). SimCon: A Tool for Modelling Context Sources for Rapid Evaluation of Pervasive Applications using Virtual Reality. Proceedings of the 5th Annual

International Conference on Mobile and Ubiquitous Systems: Computing, Networking, and Services (Mobiquitous '08), pages 1-2, ICST, Trinity College Dublin, Ireland, July 21-25, 2008.

Describes features of SimCon Version 2.

McGlinn, K., O'Neill, E., Lewis, D., `Modelling of Context and Context-Aware Services for Simulator Based Evaluation', In Proceedings of 4th International Workshop on Managing Ubiquitous Communications and Services (MUCS ’07), Multicon Lecture Notes, Munich, Germany, May 25, 2007.

Describes early modelling of SimCon and evaluation 1A.

7.3 Further Work

During the course of this work several avenues for further research and development have been considered. Here we examine these potentials under two distinct headings. The first examines the context simulation process itself and areas of further research directly related to findings in the state of the art of context simulators. The next section looks at how this work is being applied and extended to meet the requirements or a Framework Seven European Project (FP7). In particular it explores extensions to the configuration and visualisation tools, further development of IFC to meet the requirements of smart building design and evaluation and finally, the development of building occupant simulation.

7.3.1 Context Simulation

A number of interesting avenues of research with respect to context simulation were identified during the course of this research. These are discussed here.

7.3.1.1 Heterogeneity of Contexts

Heterogeneity of context types is required to support the potentially wide range of SBAs that can be developed in smart buildings. SimCon evaluated the use of three different types of location context (presence, proximity and coordinate) and the generation of temperature context was also implemented. The use of a wider range of simulated contexts for SBAs should be explored for the evaluation of other types of SBA. For example, combining simulated coordinate contexts to simulate higher level contexts like walking or running (by calculating the distance travelled between two points in time) to develop applications which can identify if a person is exercising, like that described by Lim et al. (Lim, Dey et al. 2009). Other types of simulated context may also be explored which involve the use of user models for avatars. By assigning properties to avatars, like height, weight, gait and facial characteristics, a wider range of potential simulated context sources could be developed, which range from smart floors, which identify users depending on their gait (Orr and Abowd 2000), to facial recognition systems which require that the simulated context source be facing an avatar who is also orientated towards it to enable identification (Wheeler, Weiss et al. 2010).

7.3.1.2 Simulation and Realism

An area identified in the state of the art as lacking in evaluation is the "realism" of the simulation approaches, i.e. how true is the simulation to how the physical systems behave. Only SENS looked at comparing its simulation against an equivalent physical system. An interesting question for SimCon is how accurately the simulated values can be made to reflect the context generated by a physical context source in the same environment. For example, by placing a simulated Ubisense tag in a simulated Ubisense cell in a room in a VR building, and then placing a physical Ubisense tag in the same position, in a Ubisense cell, in the physical building the VR model represents, what level of modelling is required before the two sources become indistinguishable? Each of the aforementioned methods for simulating context could be examined and compared. This would result in a breakdown of the particular benefits of each of these approaches in terms of required investment of resources and level of fidelity (i.e. realism). This could also be applied to the simulation of uncertainty, to examine the level of realism of the uncertainty models and the uncertainty modelling approach. Some work with this regard has been conducted with respect to the received signal strength between two radio frequency transceivers using statistical models, mathematical models and the use of external models. These results have yet to be fully collated and as such are not presented in this thesis.

7.3.1.3 Visualisation and Analysis Tools

The visualisation of uncertainty proved itself to provide benefits to SBA developers when designing SBAs which must deal with varying levels of uncertainty. The approach taken was to visualise the simulated context as a visual representation in the 3D environment. This location could then be compared to the actual location of the avatar, quickly highlighting the difference between the simulated context location and the avatar location. Other methods for visualising uncertainty may provide additional benefits, for example those explored by King et al. (Lemelson, King et al. 2008). They examined traffic light systems, analogue round and bar gauges, error distributions and shaded areas to represent the probability of a location falling in a certain area. These types of visualisation may prove to also be effective for improving SBA developers understanding of precision in location sensing and whether they prove more effective, or combinations of approaches prove more effective is a potential avenue of research.

Evaluation may be further supported through the use of tools which support visualisation of historical context, either graph based or as a time line. Cass (Fahy and Clarke 2004) looked at developing a time line to support this kind of analysis, and a similar method to visualising context events as they have occurred and as they are occurring could provide useful insights into the behaviour of one or more context sources over time. Similarly, graph based visualisation like those of GLS (Sanmugalingam and Coulouris 2002) and SENS (Sundresh, Kim et al. 2004) could provide similar insights.

7.3.1.4 Scalability and Performance

The SimCon model was evaluated using a small number of simulated context sources and so performance was not an issue. For smart buildings which require large numbers of simulated context sources, an exploration of scalability is required. A number of existing simulators have specifically looked at performance (GLS, SENS, UbiREAL, InSitu). As SimConGen is a standalone process, a distributed approach could provide benefits in performance, but must be implemented with consideration of the SBA developer, who may have limited resources. Further exploration of the performance of SimCon could provide insights into its scalability, which is particularly relevant when dealing with large buildings with potentially many simulated context source.

7.3.1.5 Reducing Energy Usage in Buildings

The work presented in this thesis has led to the granting of funding as part of a Framework Seven European Project (FP7) called "Knowledge-based Energy management for Public Buildings through Holistic Information Modelling and 3D Visualization" (KnoHolEM). The KnoHolEM project aims to develop an intelligent energy management solution for energy efficient buildings and spaces of public use, both by systematically avoiding energy wasting in buildings and by knowledge-based holistic optimization of energy consumption. The solution proposed integrates previously developed intellectual property into a holistic intelligent energy management solution. SimCon is providing simulation and visualization capabilities in the project.

The project brings together a diverse set of academic and industrial partners from Germany, Great Britain, Ireland, Italy, the Netherlands and Spain. The industrial partners are facility management and construction companies like ISOTROL (ISOTROL), BDigital (B-Digital), the Woningstichting de Zaligheden WSZ (WSZ) in cooperation with Smart Homes (SH) and Haagse Hoggeschool (HHS). These have provided four demonstration objects, the BlueNet building (Seville), the Media-TIC (Barcelona), the Forum building (Eindhoven) and HHS (the Hague), respectively. These building vary in design, scale and use, ranging from those which have a state of the art sensors (HSS and Media-TIC), for example passive infrared and CO2 sensors covering all rooms and corridors, to those which currently have limited sensing capabilities (BlueNet and the Forum buildings), which only have limited instalments of PIR sensors. Together these provide a unique opportunity to test and validate the proposed solution.

7.3.2 Extending SimCon to Support Building Energy Analysis

A number of extensions to the SimCon model have been considered to support the configuration of different aspects of smart buildings, in particular, those which are related to the energy consumption of the building.

7.3.2.1 WebGL Version

One issue with the SimConfig and SimConViz tool are that they currently require installation on a PC before they can be used. JQuery (Bibeault and Katz 2008) and WebGL (WebGL-OpenGL 2011) are JavaScript libraries which can support the same type of configuration and visualisation that SimCon supports, but through a browser rather than a standalone application. SimConfig and SimConViz are currently being ported to a web based interface so that it will be easier for building users to access. The WebGL version will support the configurations described in the next two sections.

7.3.2.2 Sensor and Device Configuration, Visualisation and Simulation

The SimCon model has demonstrated that SBA developers can use it to configure aspects of sensors. SimCon will continue in this line of research by exploring a larger range of types of sensors to meet the demands of the various building demonstration objects. Cass, UbiREAL, UbiWise, eHomSimulator, InSitu and V-PlaceSims also examined the use of configuration tools for configuring devices. Here, extensions to the SimCon model will be developed and will be evaluated when used by building facility managers who may wish to model sensors and devices for the purposes of simulation or to configure aspects of the physical building, like the rate of sensors or the state of devices. Suitably generic interfaces must also be developed to support the wide range of different sensors and devices which enable the various smart buildings.

Visualisation must also be considered for representing the data from these devices, both real time visualisation and also graph based, for example, giving energy consumption for devices over specific time periods. The extension to the SimCon model will explore visualisation of different data related to the energy consumption of the building at different granularities. Examples of this type of visualisation are energy consumption readings at metered points on the electrical grid, either at each plug, or a zone or floor level. Alternatively, where actual metered data is not available, known energy consumption of devices can be displayed, where this data is available. Similarly, different visualisations of sensor data will be made available through the SimConViz interface, ranging from data on occupancy, temperature, humidity and CO2 levels. This data can either be presented as they are measured or where data is stored, presented in graph form. This proposed extension to SimCon will be implemented and evaluated as part of the KnoHolEM project.

Experiment 3 also looked at a very simple simulation of an automated door which could be opened and closed. Incorporating devices and the states of devices into SimCon will enable the evaluation of a wider range of SBAs, for example, those which control building devices. Also, as models are developed which maintain the state of building devices, these states can be further used as inputs into the simulation of context. For example, where a heating unit is turned on, this will affect the measured value at a temperature sensor close to the heater.

7.3.2.3 Occupant Activity Modelling and Simulation

User behaviour has a large impact on the energy consumption of a building during operation (Hoes, Hensen et al. 2009). User behaviour can be defined as user presence, user activities and user controls of building systems, where these affect the indoor environment (Tabak and De Vries 2010). To understand how user's behaviour influences the energy consumption of a building, the SimCon model is being extended to enable the capture of user movements, activities and interactions with the building and its systems. Currently, a number of the existing context simulators document support for modelling bots (Cass, UbiREAL, V-PlaceSims, InSitu), although the Quake III games engines also support modelling bots. Only InSitu has looked at modelling the behaviour of building occupants to support occupant simulation. SimCon will look at extending this work to develop models which will then be used as inputs into building performance simulation tools. Building performance here is related not only to energy consumption, but also user satisfaction and job performance, factors which must be taken into consideration when developing energy reducing strategies (Zimmerman 2007). Building performance simulation tools can then inform those responsible for building energy management about their decisions regarding new policies and building configurations (Shen, Shen et al. 2011).

An extension to SimConfig will also be developed for supporting modelling of user activities, which include their movements in the building and their interactions with the building and building devices. These models can then be used as inputs into SimConGen for simulating context sensors which are affected by the presence of occupants, for example, CO2, humidity, temperature and passive location systems like infrared sensors. These simulated context sources will also make use of the same uncertainty model defined in this thesis for introducing uncertainty to evaluate the robustness of the smart building systems. These models will also be used for performance simulation and a number of researchers have looked at developing user behaviour models for the purpose of performance simulation. These have ranged from activity models to predict lighting energy performance (Reinhart 2004), to user interactions with windows and its effect on thermal comfort and energy use (Rijal, Tuohy et al. 2007), to larger sets of interactions including (in addition to aforementioned windows and lights) heaters and fans (Nicol 2001) and additional activities like going to lunch, getting a drink and having a smoke (Tabak and De Vries 2010).

It is intended that the aforementioned developed models will be validated against the data collected from the building objects. Collectively, this further development and the required evaluation will result in a yet more flexible SimCon model which can be applied to the domain of energy efficiency in buildings, and can potentially result in reducing energy consumption across Europe.

8 Bibliography

Aberer, K., M. Hauswirth and A. Salehi, "*A Middleware For Fast And Flexible Sensor Network Deployment*". Proceedings of the 32nd International Conference on Very Large Data Bases (VLDB '06), ACM, Seoul, Korea, pages 1199-1202, Sept 12-15, 2006.

Aberer, K., M. Hauswirth and A. Salehi. "*The Global Sensor Networks Middleware for Efficient and Flexible Deployment and Interconnection of Sensor Networks*". Technical Report LSIR-REPORT-2006-006. Ecole Polytechnique Fdrale de Lausanne (EPFL).

Addlesee, M. D., A. Jones, F. Livesey and F. Samaria (1997). "*The ORL active floor [sensor system]*". Personal Communications, IEEE **4**(5): 35-41. 1997.

AeroScout (2012). "*WIFI Based RTLS - AeroScout.*" Retrieved 05/07/2012, from http://www.aeroscout.com/.

Agarwal, Y., B. Balaji, R. Gupta, J. Lyles, M. Wei and T. Weng, "*Occupancy-driven energy management for smart building automation*". Second ACM Workshop on Embedded Sensing Systems For Energy-Efficiency In Buildings (SenSys 2010), ACM, Zürich, Switzerland, pages 1-6, November 2, 2010.

Anagnostopoulos, C., O. Sekkas and S. Hadjiefthymiades, "*Context fusion: dealing with sensor reliability*". IEEE, pages 1-6, 2007.

Antifakos, S., N. Kern, B. Schiele and A. Schwaninger, "*Towards Improving Trust in Context-aware Systems by Displaying System Confidence*". In Proceedings MobileHCI 2005, ACM, University of Salzburg, Austria, pages 9-14, 19 - 22 Sept, 2005.

Antony Guinard, A. M. G., Dirk Pesch, "*A Wireless Sensor Network Design Tool to Support Building Energy Management*". Proceedings of the First ACM Workshop on Embedded Sensing Systems for Energy-Efficiency in Buildings, BuildSys '09, ACM, Berkeley, California, pages 25-30, Nov 4-6, 2009.

Armac, I. and D. Retkowitz. "*Simulation of Smart Environments*". IEEE International Conference on Pervasive Services (ICPS'07). Istanbul, Turkey, IEEE**:** 257- 266

Autodesk (2012). "*AutoCAD.*" Retrieved 04/02/2013, from http://usa.autodesk.com/autocad/.

B-Digital. "*B-Digital Web Page.*" Retrieved 05/07/2012, from http://www.bdigital.org/en/Pages/Home.aspx.

Bahl, P. and V. N. Padmanabhan, "*RADAR: An In-building RF-based User Location and Tracking System*". Proceedings of the Nineteenth Annual Joint Conference of the IEEE Computer and Communications Societies (INFOCOM '00) IEEE, Tel Aviv, Israel, pages 775-784, March 26-30, 2000.

Baldauf, M., S. Dustdar and F. Rosenberg (2007). "*A Survey on Context-Aware Systems*". International Journal of Ad Hoc and Ubiquitous Computing **2**(4): 263-277. 2007.

Bangor, A., P. Kortum and J. Miller (2009). "*Determining what individual SUS scores mean: Adding an adjective rating scale*". Journal of Usability studies **4**(3): 114-123. 2009.

Barclay, L. W. "*Propagation of radiowaves*", Institution of Electrical Engineers. 2003.

Barton, J. J. and V. Vijayaraghavan. "*UBIWISE, A Simulator for Ubiquitous Computing Systems Design"*, Hewlett Packard Laboratories, Palo Alto.

Battestini, A., A. Flanagan and F. Helsinki, "*Modelling and Simulating Context Data in a Mobile Environment*". Workshop on Context Awareness for Proactive Systems (CAPS), IEEE, University of Surrey, UK, pages 127–136, June 18-19, 2005.

Bazire, M. and P. Brézillon, "*Understanding Context Before Using It*". Proceedings of the 5th international conference on Modeling and Using Context (CONTEXT'05) Springer-Verlag, Paris, France, pages 29-40, July 5-8, 2005.

Bettini, C., O. Brdiczka, K. Henricksen, J. Indulska, D. Nicklas, A. Ranganathan and D. Riboni (2010). "*A Survey of Context Modelling and Reasoning Techniques*". Pervasive and Mobile Computing. Context Modelling, Reasoning and Management. **6**(2): 161-180. 2010.

Bevan, N., "*Common Industry Format Usability Tests*". Proceedings of Usability Professionals Association Conference, Citeseer, Scottsdale, Arizona, 1999.

Bibeault, B. and Y. Katz. "*jQuery in Action*", Manning Publications Co., 2008.

Björk, B. C. and M. Laakso (2010). "*CAD Standardisation in the Construction Industry--A Process View*". Automation in Construction, Building Information Modeling and Interoperability **19**(4): 398-406. 2010.

Blip (2012). "*Blip Systems.*" Retrieved 05/07/2012, from http://www.blipsystems.com/.

Botts, M. "*Sensor Model Language (SensorML) for In-situ and Remote Sensors"*, Open Geospatial Consortium (OGC).

Broens, T. and A. Halteren, "*SimuContext: Simply Simulate Context*". In Proceedings of the International Conference on Autonomic and Autonomous Systems (ICAS ’06), IEEE Computer Society, Silicon Valley, USA Marriott Hotel, Santa Clara., pages 45, July 19-21, 2006.

Brooke, J. "*SUS-A Quick and Dirty Usability Scale*". Usability Evaluation in Industry In P. W. Jordan, B. Thomas, B. A. Weerdmeester, & A. L. McClelland.**:** pages 189-194, 1996.

Brown, P. J. "*The Stick-e Document: A Framework for Creating Context-aware Applications"*. In Proceedings of Conference on Electronic Publishing and Document Manipulation (EP'96). Xerox Palo Alto Research Center (PARC) in Palo Alto, California., Wiley. **8:** 259-272.

Brumitt, B., B. Meyers, J. Krumm, A. Kern and S. Shafer, "*Easyliving: Technologies for Intelligent Environments*". Proceedings of the 2nd international Symposium on Handheld and Ubiquitous Computing (HUC '00), Springer, Bristol, UK, pages 97-119, Sept 25-27, 2000.

Buchholz, T., A. Küpper and M. Schiffers, "*Quality of context: What it is and why we need it*". In Proceedings of the 10th Workshop of the OpenView University Association: OVUA'03, University of Geneva, Switzerland, pages July 6-9, 2003.

Bylund, M. and F. Espinoza. "*Testing and Demonstrating Context-aware Services with Quake III Arena"*. Communications of the ACM - Internet Abuse in the Workplace and Game Engines in Scientific Research. New York, NY, USA ACM. **45:** 46 - 48

Cao, J., N. Xing, A. T. S. Chan, Y. Feng and B. Jin, "*Service Adaptation Using Fuzzy Theory in Context-aware Mobile Computing Middleware*". Proceedings of the 11th IEEE International Conference on Embedded and Real-Time Computing Systems and Applications (RTCSA '05), IEEE Computer Society, Hong Kong, China, pages 496-501, Aug 17-19, 2005.

Carter, S. and J. Mankoff (2005). "*Prototypes in the Wild: Lessons from Three Ubicomp Systems*". IEEE Pervasive Computing **4**(4): 51-57. 2005.

Cassou, D., J. Bruneau and C. Consel, "*A Tool Suite to Prototype Pervasive Computing Applications*". 8th IEEE International Conference on Pervasive Computing and Communications Workshops (PERCOM Workshops), Mannheim, Germany pages 820-822, March 29 - April 2, 2010.

Chalmers, D. "*Sensing and Systems in Pervasive Computing: Engineering Context Aware Systems*". New York, Springer-Verlag, 2011.

Chen, G. and D. Kotz. "*A Survey of Context-aware Mobile Computing Research"*. Technical Report TR2000-381, Dept. of Computer Science, Dartmouth CollegeDartmouth College Hanover, NH, USA.

Chen, H., T. Finin and A. Joshi. "*The SOUPA Ontology for Pervasive Computing*". Ontologies for Agents: Theory and Experiences, Birkhäuser Basel**:** pages 233-258, 2005.

Cheung, R., "*An Adaptive Middleware Infrastructure Incorporating Fuzzy Logic for Mobile Computing*". International Conference on Next Generation Web Services Practices (NWeSP'05), IEEE, Seoul, Korea, pages 3 pp., Aug 22-26, 2005.

Chin, J. P., V. A. Diehl and K. L. Norman, "*Development of an Instrument Measuring User Satisfaction of the Human-computer Interface*". Proceedings of the SIGCHI conference on Human factors in computing (CHI '88), ACM, Omni-Shoreham Hotel, Washington, DC, USA, pages 213-218, May 15-19, 1988.

Clear, A. K., S. Dobson and P. Nixon, "*An Approach to Dealing with Uncertainty in Context-aware Pervasive Systems*". Proceedings of the UK/IE IEEE SMC Cybernetic Systems Conference IEEE, University College Dublin, Ireland, pages Sept 6-7, 2007.

Clements-Croome, D. "*Intelligent Buildings: Design, Management and Operation*", Thomas Telford Books, 2004.

Consel, C., "*DiaSuite: A Paradigm-oriented Software Development Approach*". Proceedings of the 20th ACM SIGPLAN Workshop on Partial Evaluation and Program Manipulation (PEPM '11), ACM, Austin, Texas, USA., pages 77-78, Jan 26-28, 2011.

Consolvo, S., L. Arnstein and B. Franza, "*User Study Techniques in the Design and Evaluation of a Ubicomp Environment*". Proceedings of the 4th International Conference on Ubiquitous Computing (UbiComp '02), Springer, Lecture Notes in Computer Science, pages 73-90, 2002.

Cook, D. and S. Das. "*Smart Environments: Technology, Protocols and Applications*", Wiley-Interscience, 2004.

Criterion (2012). "*Renderware.*" Retrieved 05/07/2012, from http://www.moddb.com/engines/renderware.

Damián-Reyes, P., J. Favela and J. Contreras-Castillo (2011). "*Uncertainty Management in Context-Aware Applications: Increasing Usability and User Trust*". Wireless Personal Communications, Trustworthy and Intelligent Services for Ubiquitous Computing **56**(1): 37-53. 2011.

Davies, N., J. Landay, S. Hudson and A. Schmidt (2005). "*Rapid Prototyping for Ubiquitous Computing*". IEEE Pervasive Computing **4**(4): 15-17. 2005.

Deak, G., K. Curran and J. Condell (2012). "*A Survey of Active and Passive Indoor Localisation Systems*". Computer Communications **35**(16): 1939-1954. 2012.

Dearman, D., A. Varshavsky, E. De Lara and K. N. Truong, "*An Exploration of Location Error Estimation*". Proceedings of the 9th International Conference on Ubiquitous Computing (UbiComp '07), Springer-Verlag, Innsbruck, Austria, pages 181-198, Sept 16-19, 2007.

DeFanti, T. A., D. Acevedo, R. A. Ainsworth, M. D. Brown, S. Cutchin, G. Dawe, K. U. Doerr, A. Johnson, C. Knox and R. Kooima (2011). "*The Future of the CAVE*". Central European Journal of Engineering **1**(1): 16-37. 2011.

Delir Haghighi, P., S. Krishnaswamy, A. Zaslavsky and M. Gaber. "*Reasoning about Context in Uncertain Pervasive Computing Environments"*. Proceedings of the 3rd European Conference on Smart Sensing and Context (EuroSSC '08). Zurich, Switzerland.**:** 112-125.

Dey, A. K. "*Providing Architectural Support for Building Context-aware Applications"*. College of Computing, Georgia Institute of Technology. **Ph.D. Thesis**.

Drools (2012). "*Drools -The Business Logic integration Platform.*" Retrieved 05/07/2012, from http://www.jboss.org/drools/.

Duffy, R. A. (2012). "*QERadiant* ". Retrieved 05/07/2012, from http://www.blender.org/documentation/intranet/docs/develop/ui/qeradiant.html.

Eastman, C., Y. S. Jeong, R. Sacks and I. Kaner (2010). "*Exchange Model and Exchange Object Concepts for Implementation of National BIM Standards*". Journal of Computing in Civil Engineering **24**: 25-34. 2010.

Eastman, C., P. Teicholz, R. Sacks and K. Liston. "*BIM Handbook: A Guide to Building Information Modeling for Owners, Managers, Designers, Engineers and Contractors*", Wiley, 2011.

Eastman, C. M. "*Building Product Models: Computer Environments, Supporting Design and Construction*", CRC Press Book, 1999.

Eclipse (2010). "*Graphical Modelling Framework.*" Retrieved 05/07/2012, from http://www.eclipse.org/modeling/gmf/.

Elnahrawy, E. and B. Nath, "*Cleaning and Querying Noisy Sensors*". Proceedings of Second ACM International Workshop on Wireless Sensor Networks and Applications (co-located with Mobicom 2003 Conference) WSNA'03, ACM, San Diego, CA, USA pages 78-87, Sept 19, 2003.

ERI (2011). "*Environmental Research Institute Cork.*" from http://www.ucc.ie/en/eri/.

Espedokken, K. (2008, Jan 11, 2008). "*Information Delivery Manual - Confluence.*" Retrieved 28/05/2012, from http://idm.buildingsmart.no/confluence/display/IDM/Home.

Espinoza, F., P. Persson, A. Sandin, H. Nystrom, E. Cacciatore and M. Bylund. "*GeoNotes: Social and Navigational Aspects of Location-Based Information Systems* ". Proceedings of the 3rd international conference on Ubiquitous Computing (UbiComp '01). Atlanta, Georgia, USA.**:** 2-17.

Fahy, P. and S. Clarke, "*CASS - Middleware for Mobile Context-Aware Applications*". Workshop on Context Awareness at The Second International Conference on Mobile Systems, Applications, and Services (MobiSys 2004) Citeseer, Boston, Massachusetts, USA, June 6, 2004.

Fall, K. and K. Varadhan (2007). "*The Network Simulator - NS -2.*" Retrieved 05/07/2012, from http://www.isi.edu/nsnam/ns/.

Foundation, P. S. (2012). "*Python Programming Language.*" Retrieved 19th June, 2012, from http://www.python.org/.

GentleWare (2012). "*Poseidon for UML.*" Retrieved 06/06/2012, from http://www.gentleware.com/products.html.

Graphisoft (2012). "*ArchiCAD* ". Retrieved 05/07/2012, from http://www.graphisoft.com/products/archicad/.

Gray, P. and D. Salber. "*Modelling and Using Sensed Context Information in the Design of Interactive Applications*". Proceedings of the 8th IFIP International Conference on Engineering for Human-Computer Interaction (EHCI '01). Toronto, Canada**:** 317-335.

Group, O. M. (2009). "*Business Process Model and Notation (BPMN).*" Retrieved 05/07/2012, from http://www.omg.org/spec/BPMN/2.0/.

Gu, T., H. K. Pung and D. Q. Zhang, "*A Bayesian Approach for Dealing with Uncertain Contexts*". In Proceedings of the 2nd International Conference on Pervasive Computing Springer, Vienna, Austria, pages April 21-23, 2004.

Harter, A. and A. Hopper (1994). "*A Distributed Location System for the Active Office*". Network, IEEE **8**(1): 62-70. 1994.

Harter, A., A. Hopper, P. Steggles, A. Ward and P. Webster (2002). "*The Anatomy of a Context-Aware Application*". Wireless networks **8**(2): 187-197. 2002.

Henderson, T. R., M. Lacage, G. F. Riley, C. Dowell and J. Kopena, "*Network Simulations with the ns-3 Simulator*". Special Interest Group on Data Communications, Demonstration (SIGCOMM '08), ACM, Seattle, WA, USA, , Aug 17-22, 2008.

Henricksen, K., J. Indulska and A. Rakotonirainy, "*Generating Context Management Infrastructure from High-Level Context Models*". 4th International Conference on Mobile Data Management (MDM) - Industrial Track, IEEE, Melbourne, Australia, pages 1-6, Jan 21-24, 2003.

Henricksen, K., S. Livingstone and J. Indulska, "*Towards a Hybrid Approach to Context Modelling, Reasoning and Interoperation*". Proceedings of the First International Workshop on Advanced Context Modelling, Reasoning and Management., Nottingham, The United Kingdom, pages 54-61, Sept 7, 2004.

HHS. "*Haagse Hoggeschool Web Page.*" Retrieved 05/07/2012, from http://www.dehaagsehogeschool.nl/.

Hightower, J., R. Want and G. Borriello, "*SpotON: An Indoor 3D Location Sensing Technology based on RF Signal Strength*". UW CSE 00-02-02, University of Washington, Department of Computer Science and Engineering, Seattle, WA, 2000.

Himanen, M. "*The Intelligence of Intelligent Buildings. The Feasibility of the Intelligent Building Concept in Office Buildings*". Department of Surveying. VTT Publications, Aalto University.

HL2 (2012). "*Valve Source Engine SDK.*" Retrieved 00/07/2012, from https://developer.valvesoftware.com/wiki/Main_Page.

Hoes, P., J. L. M. Hensen, M. G. L. C. Loomans, B. de Vries and D. Bourgeois (2009). "*User behavior in whole building simulation*". Energy and Buildings **41**(3): 295-302. 2009.

Hofer, T., W. Schwinger, M. Pichler, G. Leonhartsberger, J. Altmann and W. Retschitzegger, "*Context-Awareness on Mobile Devices - The Hydrogen Approach*". Proceedings of the 36th Annual Hawaii International Conference on System Sciences (HICSS'03), IEEE, Waikoloa, Hawaii, pages 10 pp., Jan 6-9, 2003.

Hong, J., E. Suh and S. J. Kim (2009). "*Context-aware Systems: A Literature Review and Classification*". Expert Systems with Applications **36**(4): 8509-8522. 2009.

Hu, P., R. Robinson and J. Indulska, "*Sensor Standards: Overview and Experiences*". In Proceedings of The 3rd International Conference on Intelligent Sensors, Sensor Networks and Information Processing (ISSNIP 2007), Melbourne, Australia, pages 3-6, Dec 3-6, 2007.

Hunter, J. and R. Lear (2000). "*JDOM.*" Retrieved 05/-7/2012, from http://www.jdom.org/.

ISO. "*ISO:9241-11 Ergonomic Requirements for Office Work with Visual Display Terminals (VDTs)* ". Guidance on Usability, Internation Organisation for Standards.

ISO. "*ISO/PAS 16739:2005*". Industry Foundation Classes, Release 2x, Platform Specification (IFC2x Platform), International Organization of Standardization.

ISO. "*ISO/IEC:25062 Software Engineering — Software Product Quality Requirements and Evaluation (SQuaRE)* ". Common Industry Format (CIF) for Usability Test Reports, ISO.

ISO (2012). "*International Organisation for Standardisation Web Site.*" Retrieved 05/07/2012, from http://www.iso.org/iso/home.html.

ISOTROL. "*ISOTROL Web Page.*" Retrieved 05/07/2012, from https://www.isotrol.com/web/.

Jang, S., E. J. Ko and W. Woo. "*Unified User-centric Context: Who, Where, When, What, How and Why*". UbiComp Workshop on Personalized Context Modeling and Management for Ubicomp Applications (ubiPCMM). Tokyo, Japan. **149:** 26–34.

Jang, S. and W. Woo, "*Unified context representing user-centric context: Who, where, when, what, how and why*". pages 26-34, 2005.

Jokela, T., N. Iivari, J. Matero and M. Karukka, "*The Standard of User-centered Design and the Standard Definition of Usability: Analyzing ISO 13407 against ISO 9241-11*". Proceedings of the Latin American conference on Human-computer interaction CLIHC '03 ACM, Rio de Janeiro, Brasil, pages 53-60, Aug 17-20, 2003.

Jouve, W., J. Bruneau and C. Consel. "*DiaSim: A Parameterized Simulator for Pervasive Computing Applications*". Proceedings of the IEEE International Conference on Pervasive Computing and Communications PERCOM '09 San Diego, California, USA.

Kaplan, E. D. and C. J. Hegarty. "*Understanding GPS: Principles and Applications*", Artech House Publishers, 2006.

Keller, M., J. O'Donnell, K. Menzel, M. Keane and U. Gokce. "*Integrating the Specification, Acquisition and Processing of Building Performance Information*". 12th International Conference on Computing in Civil and Building Engineering. Beijing, China. **13:** 1-6.

Kidd, C., R. Orr, G. Abowd, C. Atkeson, I. Essa, B. MacIntyre, E. Mynatt, T. Starner and W. Newstetter, "*The Aware Home: A Living Laboratory for Ubiquitous Computing Research*". Proceedings of the Second International Workshop on Cooperative Buildings, Integrating Information, Organization, and Architecture (CoBuild '99), Springer-Verlag, London, UK, pages 191-198, Oct 1-2, 1999.

Kirakowski, J. and M. Corbett (1993). "*SUMI: The Software Usability Measurement Inventory*". British Journal of Educational Technology **24**(3): 210-212. 1993.

Klyne, G., F. Reynolds, C. Woodrow, H. Ohto, J. Hjelm, M. H. Butler and L. Tran (2004). "*Composite Capability/Preference Profiles (CC/PP): Structure and Vocabularies 1.0. W3C Recommendation.*". Retrieved 05/07/2012, from http://www.w3.org/TR/2001/WD-CCPP-struct-vocab-20010315/.

Korpipää, P. and J. Mäntyjärvi. "*An Ontology for Mobile Device Sensor-based Context Awareness"*. Proceedings of the 4th International and Interdisciplinary Conference on Modeling and Using Context (CONTEXT'03). Stanford, CA, USA, Springer. **2680:** 451–458.

Koyuncu, H. and S. H. Yang (2010). "*A Survey of Indoor Positioning and Object Locating Systems*". International Journal of Computer Science and Network Security (IJCSNS) **10**(5): 121-128. 2010.

LaMarca, A., Y. Chawathe, S. Consolvo, J. Hightower, I. Smith, J. Scott, T. Sohn, J. Howard, J. Hughes and F. Potter (2005). "*Place Lab: Device Positioning Using Radio Beacons in the Wild*". Pervasive Computing: 301-306. 2005.

Landauer, T. K. "*The Trouble with Computers: Usefulness, Usability, and Productivity*", MIT Press, Cambridge, MA., 1995.

Lee, S., K. N. Ha and K. C. Lee (2006). "*A Pyroelectric Infrared Sensor-based Indoor Location-aware System for the Smart Home*". Consumer Electronics, IEEE Transactions on **52**(4): 1311-1317. 2006.

Lei, H., D. M. Sow, J. S. Davis II, G. Banavar and M. R. Ebling (2002). "*The Design and Applications of a Context Service* ". SIGMOBILE Mobile Computing and Communications Review ACM **6**(4): 45-55. 2002.

Lemelson, H., T. King and W. Effelsberg, "*A Study on User Acceptance of Error Visualization Techniques*". Proceedings of the 5th Annual International Conference on Mobile and Ubiquitous Systems: Computing, Networking, and Services (Mobiquitous '08), ACM, Trinity College Dublin, Ireland, pages 53, July 21-25, 2008.

Lertlakkhanakul, J., J. W. Choi and M. Y. Kim (2008). "*Building Data Model and Simulation Platform for Spatial Interaction Management in Smart Home*". Automation in Construction **17**(8): 948-957. 2008.

Li, Y., J. I. Hong and J. A. Landay (2007). "*Design Challenges and Principles for Wizard of Oz testing of Location-enhanced Applications*". IEEE Pervasive Computing **6**(2): 70-75. 2007.

Liao, J., Y. Bi and C. Nugent (2011). "*Using the Dempster-Shafer Theory of Evidence with a Revised Lattice Structure for Activity Recognition*". IEEE Transactions on Information Technology in Biomedicine **15**(1): 74-82. 2011.

Liao, L., D. J. Patterson, D. Fox and H. Kautz (2007). "*Learning and Inferring Transportation Routines*". Artificial Intelligence **171**(5): 311-331. 2007.

Lim, B. Y., A. K. Dey and D. Avrahami, "*Why and why not explanations improve the intelligibility of context-aware intelligent systems*". Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (CHI '09). ACM, New York, Boston, USA, pages 2119-2128, April 4th-9th, 2009.

Lim, B. Y. and A. K. Dey. "*Investigating Intelligibility for Uncertain Context-aware Applications"*. Proceedings of the 13th international conference on Ubiquitous Computing. Beijing, China, ACM**:** 415-424.

Lo´ pez de Ipin˜ a, D., P. R. S. Mendonça and A. Hopper (2002). "*TRIP: A Low-cost Vision-based Location System for Ubiquitous Computing*". Personal and Ubiquitous Computing **6**(3): 206-219. 2002.

Loke, S. W., "*On representing situations for context-aware pervasive computing: six ways to tell if you are in a meeting*". Proceedings of the 4th annual IEEE international conference on Pervasive Computing and Communications Workshops (PERCOMW '06), IEEE, Pisa, Italy, pages 35, March 13-17, 2006.

Lund, A. M. (2001). "*Measuring Usability with the USE Questionnaire*". Usability Interface: The Usability SIG Newsletter of the Society for Technical Communications **8**(2): 8. 2001.

Mara, P., K. McGlinn, R. Brennan, D. O'Sullivan, M. Keane and J. O'Donnell. "*Pervasive Knowledge-Based Networking for Maintenance Inspection in Smart Buildings"*. International Conference on Autonomic Computing Proceedings of the 6th International Workshop on Managing Ubiquitous Communications and Services (ICAC '09). Barcelona, Spain, ACM, New York, USA.**:** 59-66

Mc Gibney, A., M. Klepal and D. Pesch, "*User Demand Based WLAN Design and Optimisation*". Proceedings IEEE Vehicular Technology Conference (VTC2007), IEEE, Dublin, Ireland, pages 1101-1105, April 22-25, 2007.

McGlinn, K., E. O'Neill, A. Gibney, D. O'Sullivan and D. Lewis (2010). "*SimCon: A Tool to Support Rapid Evaluation of Smart Building Application Design using Context Simulation and Virtual Reality*". Journal of Universal Computer Science **16**(15): 1992-2018. 2010.

McGlinn, K., R. Brennan, D. O'Sullivan and D. Lewis. "*The SimCon Generator: An Interactive Context Simulator for Rapid Evaluation of Smart Building Applications using Virtual Reality"*. 8th International Workshop on Managing Ubiquitous Communications and Services. Seattle, USA, IEEE.

Meier, W. (2012). "*eXistDB.*" Retrieved 05/07/2012, from http://exist-db.org/exist/index.xml.

Miquel, M. (2012). "*Siafu - An Open Source Context Simulator.*" Retrieved 05/07/2012, from http://siafusimulator.sourceforge.net/.

NASA. "*SWEET Ontology.*"_Semantic Web for Earth and Environmental Terminology (SWEET) Retrieved 05/07/2012, from http://sweet.jpl.nasa.gov/ontology/sensor.owl.

Ni, L. M., Y. Liu, Y. C. Lau and A. P. Patil (2004). "*LANDMARC: Indoor Location Sensing using Active RFID*". Wireless networks **10**(6): 701-710. 2004.

Nicol, J. F., "*Characterising occupant behaviour in buildings: towards a stochastic model of occupant use of windows, lights, blinds, heaters and fans*". pages 1073-1078, 2001.

Nielsen, J. "*Usability Engineering*", Morgan Kaufmann, 1993.

Nielsen, J. and T. K. Landauer, "*A Mathematical Model of the Finding of Usability Problems*". Proceedings of the INTERACT '93 and CHI '93 conference on Human factors in computing systems (CHI '93) ACM New York, NY, USA, Amsterdam, The Netherlands, pages 206-213, April 24-29, 1993.

Nielsen, J., "*Usability Inspection Methods*". Conference Companion on Human Factors in Computing Systems CHI '94 ACM, Boston, Massachusetts, USA, pages 413-414, April 24-28, 1994.

NIMBUS (2012). "*Nimbus Embedded Systems Research Centre.*" Retrieved 05/07/2012, from http://nimbus.cit.ie/.

Nishikawa, H., S. Yamamoto, M. Tamai, K. Nishigaki, T. Kitani, N. Shibata, K. Yasumoto and M. Ito. "*UbiREAL: Realistic Smartspace Simulator for Systematic Testing*". 8th International Conference on Ubiquitous Computing (UbiComp2006) Orange County, California, USA**:** 459-476.

Norbisrath, U., C. Mosler and I. Armac, "*The eHome Configurator Tool Suite*". Proceedings of the 2006 international conference on On the Move to Meaningful Internet Systems: AWeSOMe, CAMS, COMINF, IS, KSinBIT, MIOS-CIAO, MONET - OTM'06 Springer, Montpellier, France, pages 1315-1324, Oct 29 - Nov 3, 2006.

Northover, S. and M. Wilson. "*SWT: The Standard Widget Toolkit*", Addison-Wesley, 2004.

O'Neill, E., D. Lewis, K. McGlinn and S. Dobson. "*Rapid User-centred Evaluation for Context-aware Systems*". Proceedings of the 13th International Workshop on Design. Specification and Verification of Interactive Systems (DSVIS'06). Trinity College Dublin, Ireland, Springer LNCS. **4323:** 220.

O'Neill, E., D. Lewis and O. Conlan, "*A Simulation-based Approach to Highly Iterative Prototyping of Ubiquitous Computing Systems*". Proceedings of the 2nd International Conference on Simulation Tools and Techniques (Simutools '09), Rome, Italy, pages 1-10, March 2-6, 2009.

O'Neill, E. "*Situation-Based Testing for Ubiquitous Computing Systems*". School of Computer Science and Statistics, Trinity College Dublin. **Ph.D. Thesis:** 2011.

O'Neill, E., O. Conlan and D. Lewis (2011). "*Modeling and simulation to assist context aware system design*". Simulation **87**(1-2): 149-170. 2011.

O'Neill, E., M. Klepal, D. Lewis, T. O'Donnell, D. O'Sullivan and D. Pesch. "*A Testbed for Evaluating Human Interaction with Ubiquitous Computing Environments"*. Proceedings of First International Conference on Testbeds and Research Infrastructures for the DEvelopment of NeTworks and COMmunities (TRIDENTCOM 2005). Trento, Italy.

O'Sullivan, B. and M. Keane, "*Specification of an IFC Based Intelligent Graphical User Interface to Support Building Energy Simulation*". Proceedings of the Ninth International Building Performance Simulation Association Conference (IBPSA), Montreal, Quebec, Canada, pages 247, Aug 15–18, 2005.

O'Sullivan, D., M. Keane, D. Kelliher and R. Hitchcock (2004). "*Improving Building Operation by Tracking Performance Metrics throughout the Building Lifecycle (BLC)*". Energy and Buildings **36**(11): 1075-1090. 2004.

Ochoa, C. E., M. B. C. Aries and J. L. M. Hensen (2011). "*State of the Art in Lighting Simulation for Building Science: A Literature Review*". Journal of Building Performance Simulation **5**(4): 209-233. 2011.

Oh, Y., A. Schmidt and W. Woo, "*Designing, Developing, and Evaluating Context-aware Systems*". International Conference on Multimedia and Ubiquitous Engineering (MUE '07) IEEE, Korean Bible University, Seoul, Korea, pages 1158-1163, April 26-28, 2007.

Oracle (2012). "*MySQL.*" Retrieved 05/07/2012, from http://www.mysql.com/.

Orr, R. J. and G. D. Abowd, "*The Smart Floor: A Mechanism for Natural User Identification and Tracking*". Extended Abstracts on Human factors in computing systems (CHI EA '00 CHI '00), ACM, The Hague, The Netherlands, pages 275-276, April 1-6, 2000.

Padovitz, A., S. W. Loke and A. Zaslavsky, "*On Uncertainty in Context-aware Computing: Appealing to High-level and Same-level Context for Low-level Context Verification*". Proceedings of the 1st International Workshop on Ubiquitous Computing (IWUC 2004), INSTICC Press, Porto, Portugal, pages 62–72, April 13-14, 2004.

Padovitz, A., S. Loke, A. Zaslavsky, B. Burg and C. Bartolini. "*An Approach to Data Fusion for Context Awareness"*. Proceedings of the 5th international conference on Modeling and Using Context (CONTEXT'05). Paris, France**:** 9-63.

Papamichael, K., H. Chauvet, J. LaPorta and R. Dandridge (1999). "*Product Modeling for Computer-aided Decision-making*". Automation in Construction **8**(3): 339-350. 1999.

Pazlar, T. and Z. Turk (2008). "*Interoperability in Practice: Geometric Data Exchance using the IFC Standard*". ITcon **13**(Special Issue Case studies of BIM use): 362-380. 2008.

Pfeifer, T. (2005). "*Redundant Positioning Architecture*". Computer Communications **28**(13): 1575-1585. 2005.

Popov, V., D. Migilinskas, V. Juocevi ius and S. Mikalauskas. "*Application of Building Information Modelling and Construction Process Simulation Ensuring Virtual Project Development Concept in 5D Environment*". The 25th International Symposium on Automation and Robotics in Construction (ISARC-2008). Vilnius, Lithuania, Vilnius Gediminas Technical University Publishing House "Technika". 2008.

Prendinger, H., B. Brandherm and S. Ullrich (2009). "*A Simulation Framework for Sensor-based Systems in Second Life*". Presence: Teleoperators and Virtual Environments **18**(6): 468-477. 2009.

Priyantha, N. B., A. K. L. Miu, H. Balakrishnan and S. Teller, "*The Cricket Compass for Context-aware Mobile Applications*". 7th ACM Annual International Conference on Mobile Computing and Networking (MOBICOM 01), ACM, Rome, Italy, pages 1-14, July 16-21, 2001.

Ranganathan, A., J. Al-Muhtadi and R. H. Campbell (2004). "*Reasoning About Uncertain Contexts in Pervasive Computing Environments*". Pervasive Computing, IEEE **3**(2): 62-70. 2004.

Reilly, D., D. Dearman, M. Welsman-Dinelle and K. Inkpen (2005). "*Evaluating Early Prototypes in Context: Trade-offs, Challenges, and Successes*". Pervasive Computing, IEEE **4**(4): 42-50. 2005.

Reinhart, C. F. (2004). "*Lightswitch-2002: a model for manual and automated control of electric lighting and blinds*". Solar Energy **77**(1): 15-28. 2004.

Reynolds, V., V. Cahill and A. Senart, "*Requirements for an Ubiquitous Computing Simulation and Emulation Environment*". Proceedings of the First International Conference on Integrated Internet Ad Hoc and Sensor Networks, ACM Press, Nice, France, pages May 30-31, 2006.

Rijal, H., P. Tuohy, M. Humphreys, J. Nicol, A. Samuel and J. Clarke (2007). "*Using results from field surveys to predict the effect of open windows on thermal comfort and energy use in buildings*". Energy and Buildings **39**(7): 823-836. 2007.

Ronzani, D. (2009). "*The Battle of Concepts: Ubiquitous Computing, Pervasive Computing and Ambient Intelligence in Mass Media*". Ubiquitous Computing and Communication Journal. **4**(2). 2009.

Roy, N., T. Gu and S. K. Das (2010). "*Supporting Pervasive Computing Applications with Active Context Fusion and Semantic Context Delivery*". Pervasive and Mobile Computing **6**(1): 21-42. 2010.

Rukzio, E., J. Hamard, C. Noda and A. De Luca, "*Visualization of Uncertainty in Context Aware Mobile Applications*". Proceedings of the 8th conference on Human-computer Interaction with Mobile Devices and Services (MobileHCI '06), ACM, Espoo, Finland, pages 247-250, Sep 12-15, 2006.

Salber, D., A. K. Dey and G. D. Abowd, "*The context toolkit: Aiding the Development of Context-enabled Applications*". Proceedings of the SIGCHI Conference on Human Factors in Computing Systems: the CHI is the Limit (CHI '99), ACM, New York, NY, Pittsburgh, Pennsylvania, USA, pages 434-441, May 15 – 20, 1999.

Sanchez, D., M. Tentori and J. Favela (2008). "*Activity Recognition for the Smart Hospital*". Intelligent Systems, IEEE **23**(2): 50-57. 2008.

Sanmugalingam, K. and G. Coulouris, "*A Generic Location Event Simulator*". Proceedings of the 4th International Conference on Ubiquitous Computing (UbiComp '02), Springer-Verlag London, UK, Gothenburg, Sweden, pages 308-315, Sept 29 - Oct 1, 2002.

Sauro, J. and E. Kindlund, "*A Method to Standardize Usability Metrics Into a Single Score*". Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (CHI '05) ACM, Portland, Oregon, USA, pages 401-409, April 2-7, 2005.

Schilit, B., N. Adams and R. Want, "*Context-aware computing applications*". Proceedings of the 1994 First Workshop on Mobile Computing Systems and Applications WMCSA '94 IEEE Computer Society, Santa Cruz, California, USA, pages 85-90, Dec 8-9, 1994.

Schilit, B. N. and M. M. Theimer (1994). "*Disseminating Active Map Information to Mobile Hosts*". IEEE Network **8**(5): 22-32. 1994.

Schmidt, A. "*Ubiquitous Computing - Computing in Context.* ". Computing Department. , Lancaster University, England, U.K. **Ph.D. Thesis**.

Scholtz, J., A. Wichansky, K. Butler, E. Morse and S. Laskowski, "*Quantifying Usability: The Industry Usability Reporting Project*". Proceedings of the Human Factors and Ergonomics Society Annual SAGE Publications, Baltimore, Maryland, USA, pages 1930-1934, Sept 29 - Oct 4, 2002.

Scholtz, J. and S. Consolvo (2004). "*Toward a Framework for Evaluating Ubiquitous Computing Applications*". IEEE Pervasive Computing **3**(2): 82-88. 2004.

Sebastian, R. (2005). "*Field Simulation for RFID-based in Door Location Sensing Algorithm.*" Retrieved 05/07/2012, from http://www.google.ie/url?sa=t&rct=j&q=&esrc=s&source=web&cd=2&ved=0CE0QFjAB&url=http%3A%2F%2Fciteseerx.ist.psu.edu%2Fviewdoc%2Fdownload%3Fdoi%3D10.1.1.111.3710%26rep%3Drep1%26type%3Dpdf&ei=2Cf4T92IBtGChQfmj5TZBg&usg=AFQjCNFtFrqukgNOe29fmQPmq122yI3UJw&sig2=IvT4gJM3TMZetb5XBIDOmg.

Seco, F., C. Plagemann, A. R. Jiménez and W. Burgard, "*Improving RFID-based Indoor Positioning Accuracy using Gaussian Processes*". International Conference on Indoor Positioning and Indoor Navigation (IPIN '10), IEEE, Campus Science City, Zurich, Switzerland, pages 1-8, Sept 15-17, 2010.

See, R. (2012). "*BLIS (Building Lifecycle Interoperable Software).*" Retrieved 05/07/2012, from http://www.blis-project.org/.

SH. "*Stichting Smart Homes Web page.*" Retrieved 05/07/2012, from http://www.smart-homes.nl/.

Shen, W., Q. Shen and Q. Sun (2011). "*Building Information Modeling-based user activity simulation and evaluation method for improving designer-user communications*". Automation in Construction. 2011.

Sheng, Q. Z. and B. Benatallah, "*ContextUML: A UML-based Modeling Language for Model-driven Development of Context-aware Web Services*". The 4th International Conference on Mobile Business (ICMB '05), IEEE, Sydney, Australia, pages 206-212, July 11 - 13, 2005.

Shirehjini, A. A. N. and F. Klar, "*3DSim: Rapid Prototyping Ambient Intelligence*". Proceeding of the 2005 Joint Conference on Smart objects and Ambient Intelligence: Innovative Context-aware Services: Usages and Technologies (sOc-EUSAI), ACM, Grenoble, France, pages 303-307, Oct 12-14, 2005.

Shreiner, D. "*OpenGL Reference Manual: The Official Reference Document to OpenGL, Version 1.2*", Addison-Wesley Longman Publishing Co., Inc., 1999.

Software, I. (2012). "*Quake III Arena.*" Retrieved 05/07/2012, from http://www.idsoftware.com/.

Song, B., H. Choi and H. S. Lee, "*Surveillance Tracking System Using Passive Infrared Motion Sensors in Wireless Sensor Network*". International Conference on Information Networking (ICOIN '08), IEEE, Busan, Korea, pages 1-5, Jan 23 - 25, 2008.

Staroswiecki, M., "*On Fault Tolerant Estimation in Sensor Networks*". Proceedings of European Control Conference (ECC'03), University of Cambridge, UK, UK, pages Sept, 2003.

States, T. N. B. S.-U. (2012). "*National BIM Standard.*" Retrieved 05/07/2012, from http://www.nationalbimstandard.org/.

Steggles, P. and S. Gschwind, "*The Ubisense Smart Space Platform*". Proceedings of the Third International Conference on Pervasive Computing, University of Texas, Arlington, USA, pages March 12, 2005.

Strang, T. and C. Linnhoff-Popien, "*A Context Modeling Survey*". The Sixth International Conference on Ubiquitous Computing.Workshop on Advanced Context Modelling, Reasoning and Management, UbiComp 2004. , Springer, Nottingham/England, pages Sept 7-10, 2004.

Sundresh, S., W. Kim and G. Agha, "*SENS: A Sensor, Environment and Network Simulator*". The 37th Annual Simulation Symposium (ANSS '37), IEEE, Arlington, Virginia, USA, pages April 18 - 22, 2004.

Tabak, V. and B. De Vries (2010). "*Methods for the prediction of intermediate activities by office occupants*". Building and Environment **45**(6): 1366-1372. 2010.

Tang, L., Z. Yu, X. Zhou, H. Wang and C. Becker (2010). "*Supporting Rapid Design and Evaluation of Pervasive Applications: Challenges and Solutions*". Personal and ubiquitous computing **15**(3): 253-269. 2010.

TCD (2012). "*Lloyd building - Trinity College Dublin.*" Retrieved 05/07/2012, from http://www.tcd.ie/disability/accessibility/Building-check/lloyd.php.

Truong, B. A., Y.-K. Lee and S.-Y. Lee. "*Modeling and Reasoning about Uncertainty in Context-Aware Systems*". IEEE International Conference on e-Business Engineering (ICEBE'05). Beijing, China, IEEE Computer Society**:** 102-109.

Tsai, C. Y., S. Y. Chou and S. W. Lin (2010). "*Location-Aware Tour Guide Systems in Museum*". Scientific Research and Essays **5**(8): 714–720. 2010.

Tulke, J., J. Tauscher, M. Theiler and T. Riedel (2012). "*Open IFC Tools.*" Retrieved 05/06/2012, from http://www.openifctools.org/Open_IFC_Tools/Home.html.

Tullis, T. S. and J. N. Stetson, "*A Comparison of Questionnaires for Assessing Website Usability*". Usability Professional Association Conference (UPA '04), Minneapolis, Minnesota, USA, pages June 7-1, 2004.

Ubisense (2012). "*Ubisense System.*" Retrieved 05/07/2012, from http://www.ubisense.net/en/.

Want, R., A. Hopper, V. Falcao and J. Gibbons (1992). "*The Active Badge Location System*". ACM Transactions on Information Systems (TOIS) **10**(1): 91-102. 1992.

WebGL-OpenGL, E. (2011). "*2.0 for the Web.*" Retrieved 28/05/2012, from http://www.khronos.org/webgl/.

Webster, J. G. "*The Measurement, Instrumentation and Sensors Handbook*", Springer, 1999.

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

Weiser, M. (1999). "*The Computer for the 21st Century*". Mobile Computing and Communications Review ACM SIGMOBILE **3**(3): 3-11. 1999.

Wheeler, F. W., R. L. Weiss and P. H. Tu, "*Face recognition at a distance system for surveillance applications*", 2010.

White, S. A. (2004). "*Introduction to BPMN.*" Retrieved 05/07/2012, from http://yoann.nogues.free.fr/IMG/pdf/07-04_WP_Intro_to_BPMN_-_White-2.pdf.

Wix, J. (2007, Feb 2010). "*HVAC Engineering A - Process Maps (PM).*" Retrieved 05/07/2012, from http://idm.buildingsmart.no/confluence/display/IDM/HVAC+Engineering+(PM).

Wood, A., J. Stankovic, G. Virone, L. Selavo, Z. He, Q. Cao, T. Doan, Y. Wu, L. Fang and R. Stoleru (2008). "*Context-aware Wireless Sensor Networks for Assisted Living and Residential Monitoring*". Network, IEEE **22**(4): 26-33. 2008.

Woolrych, A. and G. Cockton, "*Why and When Five Test Users Aren't Enough*". Proceedings of IHM-HCI 2001 Conference, Lille, France, pages 105-108, Sept, 2001.

WSZ. "*Woningstichting de Zaligheden Web Page.*" Retrieved 05/07/2012, from http://www.wsz.nl/Home/.

Wu, H., M. Siegel, R. Stiefelhagen and J. Yang, "*Sensor Fusion Using Dempster-Shafer Theory*". Proceedings of the 19th IEEE Instrumentation and Measurement Technology Conference (IMTC'02) IEEE, Anchorage, AK, USA, pages 7-12 May 21-23, 2002.

Yamazaki, T., "*Ubiquitous Home: Real-life Testbed for Home Context-Aware Service*". Proceedings of the First International Conference on Testbeds and Research Infrastructures for the DEvelopment of NeTworks and COMmunities, IEEE Computer Society, Trento, Italy, pages 54-59, Feb 23 - 25, 2005.

Ye, J., L. Coyle, S. Dobson and P. Nixon, "*Using Situation Lattices to Model and Reason about Context*". In Proceedings of the Fourth International Workshop Modeling and Reasoning in Context (MRC '07), Roskilde, Denmark, pages Aug 20-24, 2007.

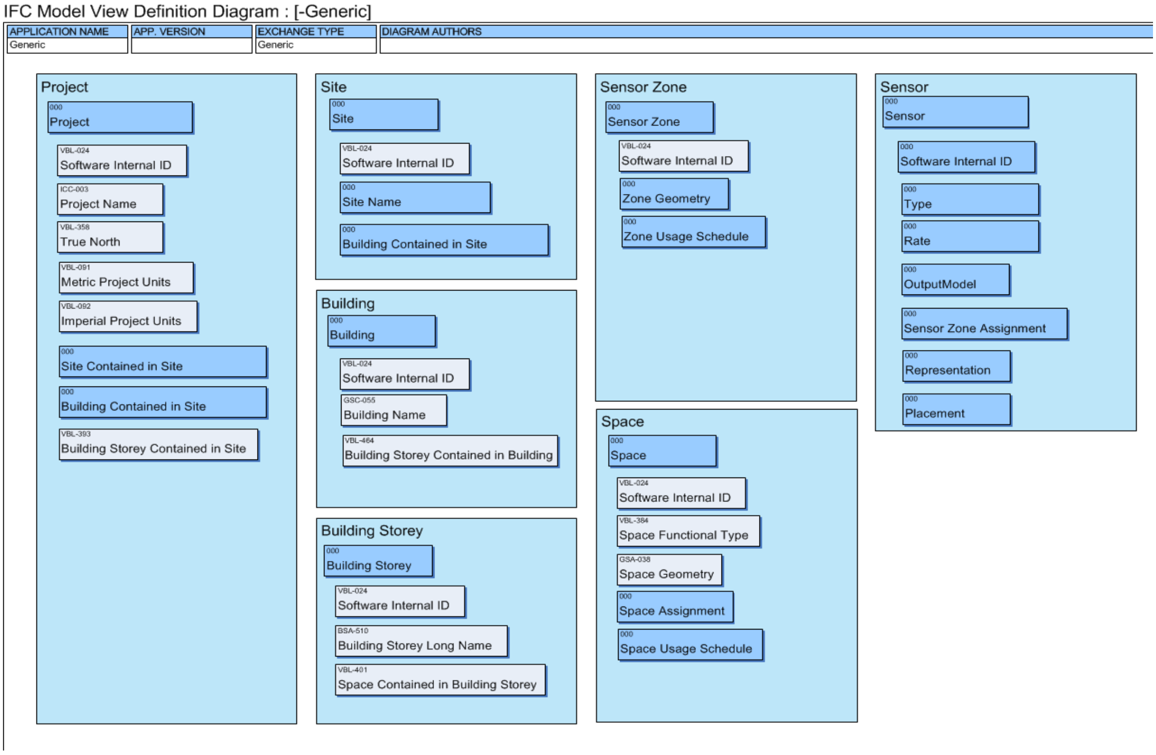
Ye, J., S. McKeever, L. Coyle, S. Neely and S. Dobson, "*Resolving Uncertainty in Context Integration and Abstraction: Context Integration and Abstraction*". Proceedings of the International Conference on Pervasive Services (ICPS 2008), ACM, Dublin Institute of Technology, Ireland, pages 131-140, July 6-10, 2008.

Zhang, D., T. Gu and X. Wang (2005). "*Enabling Context-aware Smart Home with Semantic Web Technologies*". International Journal of Human-friendly Welfare Robotic Systems **6**(4): 12-20. 2005.

Zimmerman, G. "*Modeling and simulation of individual user behavior for building performance predictions"*. Proceedings of the 2007 summer computer simulation conference. San Diego, California, Society for Computer Simulation International**:** 913-920.

Zlatanova, S. and E. Verbree (2005). "*The Third Dimension in LBS: The Steps to Go*". Geowissenschaftliche Mitteilungen(74): 185-190. 2005.

9 Appendix A: BLIS Model View Definition



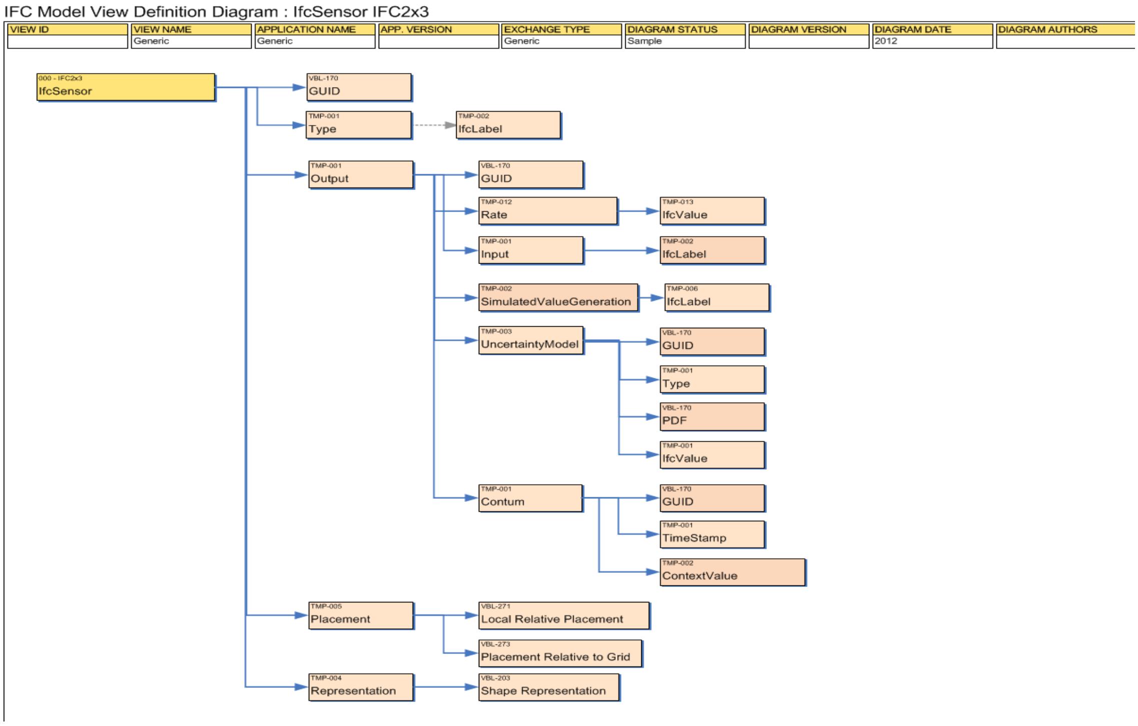


Figure 9-1 IFC-Independent MVD - Generic Sensor Model (Above)

IFC-Dependent (Ifc2x4) MVD - Sensor Model (Below)

10 Appendix B (Data Models)

10.1 SimConXML Description Ubisense Cell


```
<simconXML>
    <guid>0x_dBXGqr63eY2wgUMqApB </guid>
    <type>Ubisense</type>
    <contum>Coordinate</contum>

    <position>
        <areaID>0x_dBXGqr63eY2wgUMqApB </areaID>
        <origin>
            <x>23.0</x>
            <y>7.45</y>
            <z>1.5</z>
        </origin>
        <orientation>n/a</orientation>
        <zone>
            <coordA>
                <x>13.8</x>
                <y>6.8</y>
                <z>0.0</z>
            </coordA>
            <coordB>
                <x>28.2</x>
                <y>8.7</y>
                <z>3.0</z>
            </coordB>
        </zone>
    </position>

    <position>
        <areaID>3S8FaS899BVQ$Ontje4RZC</areaID>
        <origin>
            <x>23.0</x>
            <y>7.45</y>
            <z>1.5</z>
        </origin>
        <orientation>n/a</orientation>
        <zone>
            <coordA>
                <x>18.0</x>
                <y>6.8</y>
                <z>0.0</z>
            </coordA>
            <coordB>
                <x>20.0</x>
                <y>8.7</y>
                <z>3.0</z>
            </coordB>
        </zone>
    </position>
```

```
        <transmitter>
                <avatar>exp1</avatar>
                <rate>0.1</rate>
                <type>Ubisense</type>
                <sourceGUID></sourceGUID>
        </transmitter>


        <uncertainty>
                <sourceID>0x_dBXGqr63eY2wgUMqApB </sourceID>
                <target>Output</target>
                <distribution>Normal</distribution>
                <standardDeviation>0.15</standardDeviation>
                <mean>0.0</mean>
        </uncertainty>

        <uncertainty>
                <sourceID>3S8FaS899BVQ$Ontje4RZC</sourceID>
                <target>Zone</target>
                <distribution>Normal</distribution>
                <standardDeviation>0.3</standardDeviation>
                <mean>0.0</mean>
        </uncertainty>

</simconXML>
```

10.2 SensorML Descriptions

10.2.1 Ubisense Cell


```
<?xml version="1.0" encoding="utf-8"?>
<sml:SensorML xmlns:sml="http://www.opengis.net/sensorML/1.0"
        xmlns:swe="http://www.opengis.net/swe/1.0"
        xmlns:gml="http://www.opengis.net/gml"
        xmlns:xlink="http://www.w3.org/1999/xlink"
        xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:schemaLocation="http://www.opengis.net/sensorML/1.0
http://schemas.opengis.net/sensorML/1.0.0/sensorML.xsd" version="1.0">
        <member xlink:role="urn:ogc:def:role:OGC:sensorSystem">
<!--~~~~~~~~~~~~~~~~~~~~~~~~~~ -->
<!-- System Begins -->
<!-- ~~~~~~~~~~~~~~~~~~~~~~~~ -->
<System gml:id=" 0n0YGP2oTBJh_EeNTHMMiH">
        <gml:name>Ubisense</gml:name>
        <!-- Contact -->
        <sml:contact xlink:arcrole="author">
                <sml:ResponsibleParty>
                        <sml:individualName>Kris McGlinn</sml:individualName>
                        <sml:organizationName>Trinity College Dublin</sml:organizationName>
                        <sml:contactInfo>
                                <sml:address>

        <sml:electronicMailAddress>Kris.McGlinn@cs.tcd.ie</sml:electronicMailAddress>
                                </sml:address>
                        </sml:contactInfo>
                </sml:ResponsibleParty>
        </sml:contact>
<!--~~~~~~~~~~~~~~~~~~~~~~~~~~ -->
<!-- System Coordinate Frame -->
<!-- ~~~~~~~~~~~~~~~~~~~~~~~~ -->
<spatialReferenceFrame>
        <gml:EngineeringCRS gml:id="0x_dBXGqr63eY2wgUMqApB">
                <gml:srsName>Undefined</gml:srsName>
                <gml:usesCS xlink:href="urn:ogc:cs:OGC:xyzFrame"/>
                <gml:usesEngineeringDatum>
                        <gml:EngineeringDatum gml:id="Undefined">
                                <gml:datumName>Undefined</gml:datumName>
                                <gml:anchorPoint>origin is at the bottom left of the building;
                                        x and y are orthogonal to z but undetermined;
                                        z is along the axis of the mounting pole - typically
vertical
                                </gml:anchorPoint>
                        </gml:EngineeringDatum>
                </gml:usesEngineeringDatum>
        </gml:EngineeringCRS>
</spatialReferenceFrame>
<!--~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~-->
<!-- Ubiisense Placement -->
<!--~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~-->
<positions>
        <PositionList>
                <!--===============================================-->
                <!-- Placement of Ubisense in Environmental Research Institute Ref Frame -->
                <!--===============================================-->
                <position name=" 1JsdxtTxP7nPm31ZgKLFlA">
                        <swe:Position localFrame="Unspecified"
                                referenceFrame="Unspecified">
                                <swe:location>
                                        <swe:Vector
definition="urn:ogc:def:property:OGC:location">
                                                <swe:coordinate name="x">
                                                                <swe:Quantity
definition="urn:ogc:def:property:OGC:distance"
                                                                        axisID="X">
                                                                        <swe:uom code="m"/>
                                                                <swe:value>42.2</swe:value>
                                                        </swe:Quantity>
                                                </swe:coordinate>
                                                        <swe:coordinate name="y">
```

```
                                                            <swe:Quantity
definition="urn:ogc:def:property:OGC:distance"
                                                                    axisID="Y">
                                                                    &lt;swe:uom code="m"/&gt;
                                                            &lt;swe:value&gt;4.0&lt;/swe:value&gt;
                                                    &lt;/swe:Quantity&gt;
                                            &lt;/swe:coordinate&gt;
                                                    &lt;swe:coordinate name="z"&gt;
                                                            <swe:Quantity
definition="urn:ogc:def:property:OGC:distance"
                                                                    axisID="Z">
                                                                    &lt;swe:uom code="m"/&gt;
                                                            &lt;swe:value&gt;7.3&lt;/swe:value&gt;
                                                    &lt;/swe:Quantity&gt;
                                            &lt;/swe:coordinate&gt;
                                    &lt;/swe:Vector&gt;
                            &lt;/swe:location&gt;
                    &lt;/swe:Position&gt;
            &lt;/position&gt;
        &lt;/PositionList&gt;
&lt;/positions&gt;
&lt;swe:Envelope referenceFrame="#ENVIRONMENTAL_RESEARCH_INSTITUTE_FRAME"&gt;
        &lt;swe:lowerCorner&gt;
                &lt;swe:location&gt;
                        &lt;swe:Vector definition="urn:ogc:def:property:OGC:location"&gt;
                                &lt;swe:coordinate name="x"&gt;
                                                <swe:Quantity
definition="urn:ogc:def:property:OGC:distance"
                                                        axisID="X">
                                                        &lt;swe:uom code="m"/&gt;
                                                &lt;swe:value&gt;42.2&lt;/swe:value&gt;
                                        &lt;/swe:Quantity&gt;
                                &lt;/swe:coordinate&gt;
                                        &lt;swe:coordinate name="y"&gt;
                                                <swe:Quantity
definition="urn:ogc:def:property:OGC:distance"
                                                        axisID="Y">
                                                        &lt;swe:uom code="m"/&gt;
                                                &lt;swe:value&gt;2.1&lt;/swe:value&gt;
                                        &lt;/swe:Quantity&gt;
                                &lt;/swe:coordinate&gt;
                                        &lt;swe:coordinate name="z"&gt;
                                                <swe:Quantity
definition="urn:ogc:def:property:OGC:distance"
                                                        axisID="Z">
                                                        &lt;swe:uom code="m"/&gt;
                                                &lt;swe:value&gt;7.2&lt;/swe:value&gt;
                                        &lt;/swe:Quantity&gt;
                                &lt;/swe:coordinate&gt;
                        &lt;/swe:Vector&gt;
                &lt;/swe:location&gt;
        &lt;/swe:lowerCorner&gt;
        &lt;swe:upperCorner&gt;
                &lt;swe:location&gt;
                        &lt;swe:Vector definition="urn:ogc:def:property:OGC:location"&gt;
                                &lt;swe:coordinate name="x"&gt;
                                                <swe:Quantity
definition="urn:ogc:def:property:OGC:distance"
                                                        axisID="X">
                                                        &lt;swe:uom code="m"/&gt;
                                                &lt;swe:value&gt;46.0&lt;/swe:value&gt;
                                        &lt;/swe:Quantity&gt;
                                &lt;/swe:coordinate&gt;
                                        &lt;swe:coordinate name="y"&gt;
                                                <swe:Quantity
definition="urn:ogc:def:property:OGC:distance"
                                                        axisID="Y">
                                                        &lt;swe:uom code="m"/&gt;
                                                &lt;swe:value&gt;5.8&lt;/swe:value&gt;
                                        &lt;/swe:Quantity&gt;
                                &lt;/swe:coordinate&gt;
                                        &lt;swe:coordinate name="z"&gt;
                                                <swe:Quantity
definition="urn:ogc:def:property:OGC:distance"
                                                        axisID="Z">
                                                        &lt;swe:uom code="m"/&gt;
```

```
                                        <swe:value>9.0</swe:value>
                                </swe:Quantity>
                        </swe:coordinate>
                </swe:Vector>
        </swe:location>
    </swe:upperCorner>
</swe:Envelope>

<sml:parameters>
    <sml:ParameterList>
        <!--~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~-->
        <!-- Rate -->
        <!--~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~-->
        <parameter                                              name="resolution"
xlink:arcrole="urn:ogc:def:property:OGC:resolution">
                <ConditionalValue>
                        <data>
                                <swe:Quantity
definition="urn:ogc:def:property:OGC:duration">
                                        <swe:uom code="s"/>
                                        <swe:value>0.1</swe:value> //default value
                                </swe:Quantity>
                        </data>
                </ConditionalValue>
        </parameter>
        <!--~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~-->
        <!-- Latency -->
        <!--~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~-->
        <parameter name="latency" xlink:arcrole="urn:ogc:def:property:OGC:latencyTime">
                <ConditionalValue>
                        <data>
                                <swe:Quantity
definition="urn:ogc:def:property:OGC:duration">
                                        <swe:uom code="s"/>
                                        <swe:value>0</swe:value> //default value
                                </swe:Quantity>
                        </data>
                </ConditionalValue>
        </parameter>
        <!--~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~-->
        <!—Uncertainty Model -->
        <!--~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~-->
        <sml:parameter                                            name="accuracy"
xlink:arcrole="urn:ogc:def:property:OGC:accuracy">
                <swe:NormalizedCurve                                     fixed="true"
definition="urn:ogc:def:property:OGC:absoluteError">
                        <swe:function>
                                <swe:Curve>
                                        <swe:elementCount>
                                                <swe:Count>
                                                        <swe:value>1</swe:value>
                                                </swe:Count>
                                        </swe:elementCount>
                                        <swe:elementType>
                                                <swe:SimpleDataRecord>
                                                        <swe:field name="distance">
                                                                <swe:Quantity
definition="urn:ogc:def:property:OGC:distance">
                                                                        <swe:uom code="m"/>
                                                                </swe:Quantity>
                                                        </swe:field>
                                                        <swe:field name="uncertainty">
                                                                <swe:Quantity
definition="uncertainty">

                                                                </swe:Quantity>
                                                        </swe:field>
                                                </swe:SimpleDataRecord>
                                        </swe:elementType>
                                        <swe:encoding>
                                                <swe:TextBlock          tokenSeparator=","
blockSeparator=" " decimalSeparator="."/>
                                        </swe:encoding>
                                        <swe:values>
                                                0.15 //default value for all cells
                                        </swe:values>
                                </swe:Curve>
```

```
                              </swe:function>
                      </swe:NormalizedCurve>
              </sml:parameter>
       </sml:ParameterList>
</sml:parameters>


              </sml:System>
              <!-- ~~~~~~~~~ End of System ~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~ -->
       </sml:member>
</sml:SensorML>
```

10.2.2 Tyndall ZigBee Transceiver

```
<?xml version="1.0" encoding="utf-8"?>
<sml:SensorML xmlns:sml="http://www.opengis.net/sensorML/1.0"
       xmlns:swe="http://www.opengis.net/swe/1.0"
       xmlns:gml="http://www.opengis.net/gml"
       xmlns:xlink="http://www.w3.org/1999/xlink"
       xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"

xsi:schemaLocation="http://www.opengis.net/sensorML/1.0

http://schemas.opengis.net/sensorML/1.0.0/sensorML.xsd" version="1.0">
       <member xlink:role="urn:ogc:def:role:OGC:sensorSystem">
<!--~~~~~~~~~~~~~~~~~~~~~~~~~ -->
<!-- System Begins -->
<!-- ~~~~~~~~~~~~~~~~~~~~~~~ -->
<System gml:id="1">
       <gml:name>TyndallZigBeeTransceiver</gml:name>
       <!-- Contact -->
       <sml:contact xlink:arcrole="author">
              <sml:ResponsibleParty>
                     <sml:individualName>Kris McGlinn</sml:individualName>
                     <sml:organizationName>Trinity College Dublin</sml:organizationName>
                     <sml:contactInfo>
                            <sml:address>


<sml:electronicMailAddress>Kris.McGlinn@cs.tcd.ie</sml:electronicMailAddress>
                            </sml:address>
                     </sml:contactInfo>
              </sml:ResponsibleParty>
       </sml:contact>
<!--~~~~~~~~~~~~~~~~~~~~~~~~ -->
<!-- System Coordinate Frame -->
<!-- ~~~~~~~~~~~~~~~~~~~~~~~ -->
<spatialReferenceFrame>
       <gml:EngineeringCRS gml:id="999">
              <gml:srsName>Undefined</gml:srsName>
              <gml:usesCS xlink:href="urn:ogc:cs:OGC:xyzFrame"/>
              <gml:usesEngineeringDatum>
                     <gml:EngineeringDatum gml:id="Undefined">
                            <gml:datumName>Undefined</gml:datumName>
                            <gml:anchorPoint>origin is at the bottom left of the building;
                                   x and y are orthogonal to z but undetermined;
                                   z is along the axis of the mounting pole - typically
vertical
                            </gml:anchorPoint>
                     </gml:EngineeringDatum>
              </gml:usesEngineeringDatum>
       </gml:EngineeringCRS>
</spatialReferenceFrame>
<!--~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~-->
<!-- ZigBee Position -->
<!--~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~-->
<positions>
       <PositionList>
              <!--=============================================-->
              <!-- Position of Receiver in Environmental Research Institute Ref Frame -->
              <!--=============================================-->
              <position name="zigbee01Position">
                     <swe:Position localFrame="Unspecified"
                            referenceFrame="Unspecified">
```

```
                              <swe:location>
                                     <swe:Vector
definition="urn:ogc:def:property:OGC:location">
                                            <swe:coordinate name="x">
                                                   <swe:Quantity

definition="urn:ogc:def:property:OGC:distance"
                                                          axisID="X">
                                                          <swe:uom code="m"/>
                                                   <swe:value>42.2</swe:value>
                                            </swe:Quantity>
                                     </swe:coordinate>
                                            <swe:coordinate name="y">
                                                   <swe:Quantity

definition="urn:ogc:def:property:OGC:distance"
                                                          axisID="Y">
                                                          <swe:uom code="m"/>
                                                   <swe:value>4.0</swe:value>
                                            </swe:Quantity>
                                     </swe:coordinate>
                                            <swe:coordinate name="z">
                                                   <swe:Quantity

definition="urn:ogc:def:property:OGC:distance"
                                                          axisID="Z">
                                                          <swe:uom code="m"/>
                                                   <swe:value>7.3</swe:value>
                                            </swe:Quantity>
                                     </swe:coordinate>
                              </swe:Vector>
                       </swe:location>
                </swe:Position>
         </position>
  </PositionList>
</positions>
<swe:Envelope referenceFrame="#ENVIRONMENTAL_RESEARCH_INSTITUTE_FRAME">
       <swe:lowerCorner>
              <swe:location>
                     <swe:Vector definition="urn:ogc:def:property:OGC:location">
                            <swe:coordinate name="x">
                                   <swe:Quantity
definition="urn:ogc:def:property:OGC:distance"
                                          axisID="X">
                                          <swe:uom code="m"/>
                                   <swe:value>42.2</swe:value>
                            </swe:Quantity>
                     </swe:coordinate>
                            <swe:coordinate name="y">
                                   <swe:Quantity
definition="urn:ogc:def:property:OGC:distance"
                                          axisID="Y">
                                          <swe:uom code="m"/>
                                   <swe:value>2.1</swe:value>
                            </swe:Quantity>
                     </swe:coordinate>
                            <swe:coordinate name="z">
                                   <swe:Quantity
definition="urn:ogc:def:property:OGC:distance"
                                          axisID="Z">
                                          <swe:uom code="m"/>
                                   <swe:value>7.2</swe:value>
                            </swe:Quantity>
                     </swe:coordinate>
              </swe:Vector>
       </swe:location>
  </swe:lowerCorner>
  <swe:upperCorner>
       <swe:location>
              <swe:Vector definition="urn:ogc:def:property:OGC:location">
                     <swe:coordinate name="x">
                            <swe:Quantity
definition="urn:ogc:def:property:OGC:distance"
                                   axisID="X">
                                   <swe:uom code="m"/>
                            <swe:value>46.0</swe:value>
```

```
                                        </swe:Quantity>
                                </swe:coordinate>
                                        <swe:coordinate name="y">
                                                <swe:Quantity
definition="urn:ogc:def:property:OGC:distance"
                                                        axisID="Y">
                                                        <swe:uom code="m"/>
                                                <swe:value>5.8</swe:value>
                                        </swe:Quantity>
                                </swe:coordinate>
                                        <swe:coordinate name="z">
                                                <swe:Quantity
definition="urn:ogc:def:property:OGC:distance"
                                                        axisID="Z">
                                                        <swe:uom code="m"/>
                                                <swe:value>9.0</swe:value>
                                        </swe:Quantity>
                                </swe:coordinate>
                        </swe:Vector>
                </swe:location>
        </swe:upperCorner>
</swe:Envelope>

<sml:parameters>
        <sml:ParameterList>
                <!--~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~-->
                <!-- Rate -->
                <!--~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~-->
                <parameter                                                    name="resolution"
xlink:arcrole="urn:ogc:def:property:OGC:resolution">
                        <ConditionalValue>
                                <data>
                                        <swe:Quantity
definition="urn:ogc:def:property:OGC:duration">
                                                <swe:uom code="s"/>
                                                <swe:value>0.5</swe:value>
                                        </swe:Quantity>
                                </data>
                        </ConditionalValue>
                </parameter>
                <!--~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~-->
                <!-- Latency -->
                <!--~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~-->
                <parameter name="latency" xlink:arcrole="urn:ogc:def:property:OGC:latencyTime">
                        <ConditionalValue>
                                <data>
                                        <swe:Quantity
definition="urn:ogc:def:property:OGC:duration">
                                                <swe:uom code="s"/>
                                                <swe:value>0.0</swe:value>
                                        </swe:Quantity>
                                </data>
                        </ConditionalValue>
                </parameter>
                <!--~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~-->
                <!-- Response Curve Steady State (Input to Output) -->
                <!--~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~-->
                <sml:parameter                                               name="steadyState"
xlink:arcrole="urn:ogc::def:property:OGC:calibration">
                        <swe:NormalizedCurve                                          fixed="true"
definition="urn:ogc:def:property:OGC:steadyState">
                                <swe:function>
                                        <swe:Curve>
                                                <swe:elementCount>
                                                        <swe:Count>
                                                                <swe:value>4</swe:value>
                                                        </swe:Count>
                                                </swe:elementCount>
                                                <swe:elementType>
                                                        <swe:SimpleDataRecord>
                                                                <swe:field name="dist">
                                                                        <swe:Quantity

definition="urn:ogc:def:property:OGC:distance">
                                                                                <swe:uom code="m"/>
                                                                        </swe:Quantity>
```

```
                                                      </swe:field>
                                                      <swe:field name="distance">
                                                             <swe:Quantity
definition="rss">
                                                                    <swe:uom code="m"/>
                                                             </swe:Quantity>
                                                      </swe:field>
                                               </swe:SimpleDataRecord>
                                        </swe:elementType>
                                        <swe:encoding>
                                               <swe:TextBlock         tokenSeparator=","
blockSeparator=" "

decimalSeparator="."/>
                                        </swe:encoding>
                                        <swe:values>
                                               0.6,24 1.2,-9 1.8,-12 2.4,-19
                                        </swe:values>
                                 </swe:Curve>
                          </swe:function>
                   </swe:NormalizedCurve>
            </sml:parameter>
            <!--~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~-->
            <!—Uncertainty Model (Variation in Steady State Output) -->
            <!--~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~-->
            <sml:parameter                                              name="accuracy"
xlink:arcrole="urn:ogc:def:property:OGC:accuracy">
                   <swe:NormalizedCurve fixed="true"

definition="urn:ogc:def:property:OGC:absoluteError">
                          <swe:function>
                                 <swe:Curve>
                                        <swe:elementCount>
                                               <swe:Count>
                                                      <swe:value>4</swe:value>
                                               </swe:Count>
                                        </swe:elementCount>
                                        <swe:elementType>
                                               <swe:SimpleDataRecord>
                                                      <swe:field name="distance">
                                                             <swe:Quantity

definition="urn:ogc:def:property:OGC:distance">
                                                                    <swe:uom code="m"/>
                                                             </swe:Quantity>
                                                      </swe:field>
                                                      <swe:field name="error">
                                                             <swe:Quantity
definition="error">
                                                             </swe:Quantity>
                                                      </swe:field>
                                               </swe:SimpleDataRecord>
                                        </swe:elementType>
                                        <swe:encoding>
                                               <swe:TextBlock         tokenSeparator=","
blockSeparator=" "

decimalSeparator="."/>
                                        </swe:encoding>
                                        <swe:values>
                                               0.6,6 1.2,6 1.8,6 2.4, 5
                                        </swe:values>
                                 </swe:Curve>
                          </swe:function>
                   </swe:NormalizedCurve>
            </sml:parameter>
     </sml:ParameterList>
</sml:parameters>


            </sml:System>
            <!-- ~~~~~~~~~ End of System ~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~ -->
     </sml:member>
</sml:SensorML>
```

10.3 Contums

10.3.1 Coordinate (Ubisense)


```
<contum>
<type>coordinate</type>
<source>Ubisense</source>
<cellguid>2BqxkWxMfCovB_kH0rCosr</cellguid>
<guid>19$fPK0jz009xTHK7Hm7C2</guid>
<coordinate>
        <x>21.50316556043252</x>
        <y>6.84301621144843</y>
        <z>1.004022379100966</z>
</coordinate>
<timeStamp>1342024230206</timeStamp>
```


10.3.2 Presence (Generic Passive Infrared Sensor)


```
<contum>
        <type>Passive Infrared</type>
        <guid>3PRlf466nFIOjTMRGO4IYM</guid>
        <event>1</event>
        <timeStamp>1340355959123</timeStamp>
</contum>
```


10.3.3 Proximity (ZigBee Tyndall Mote)


```
<contum>
        <type>ZigBee</type>
        <guid> 0R_DIVIq15SRZxcmYGI6xH</guid>
        <rss> -25</rss>
        <time>1340356426829</time>
</contum>
```


10.3.4 Temperature


```
<contum>
        <type>tmote</type>
        <guid>2EfPsmpzv7tuy4jlXEI3Yc</guid>
        <temperature>20</temperature>
</contum>
```

11 Appendix C: Usability Evaluations Figures

11.1 Figures Usability Evaluation 1

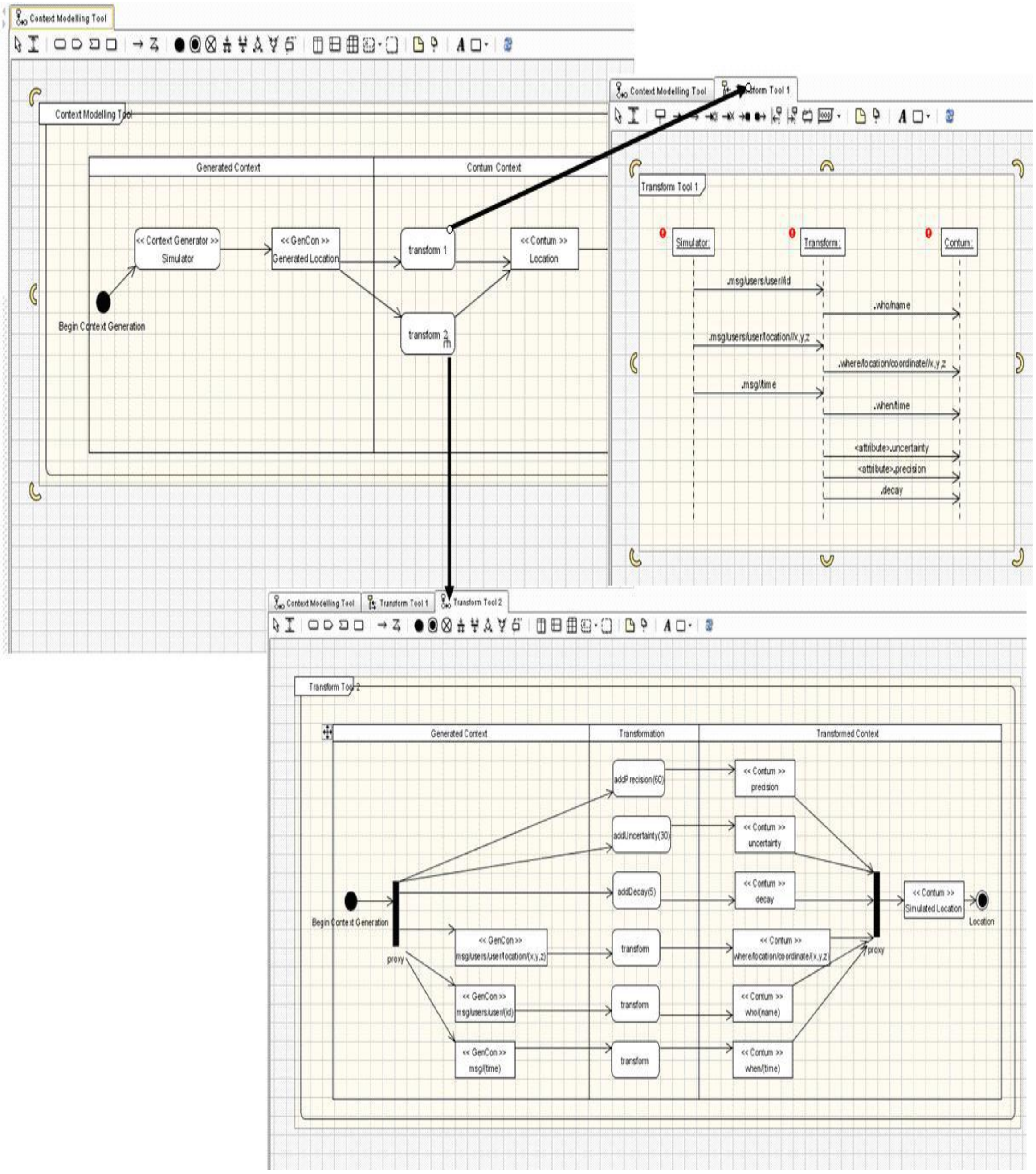


Figure 11-1 Overview of SimCon Prototype, Upper right Sequence approach, bottom right Activity approach.

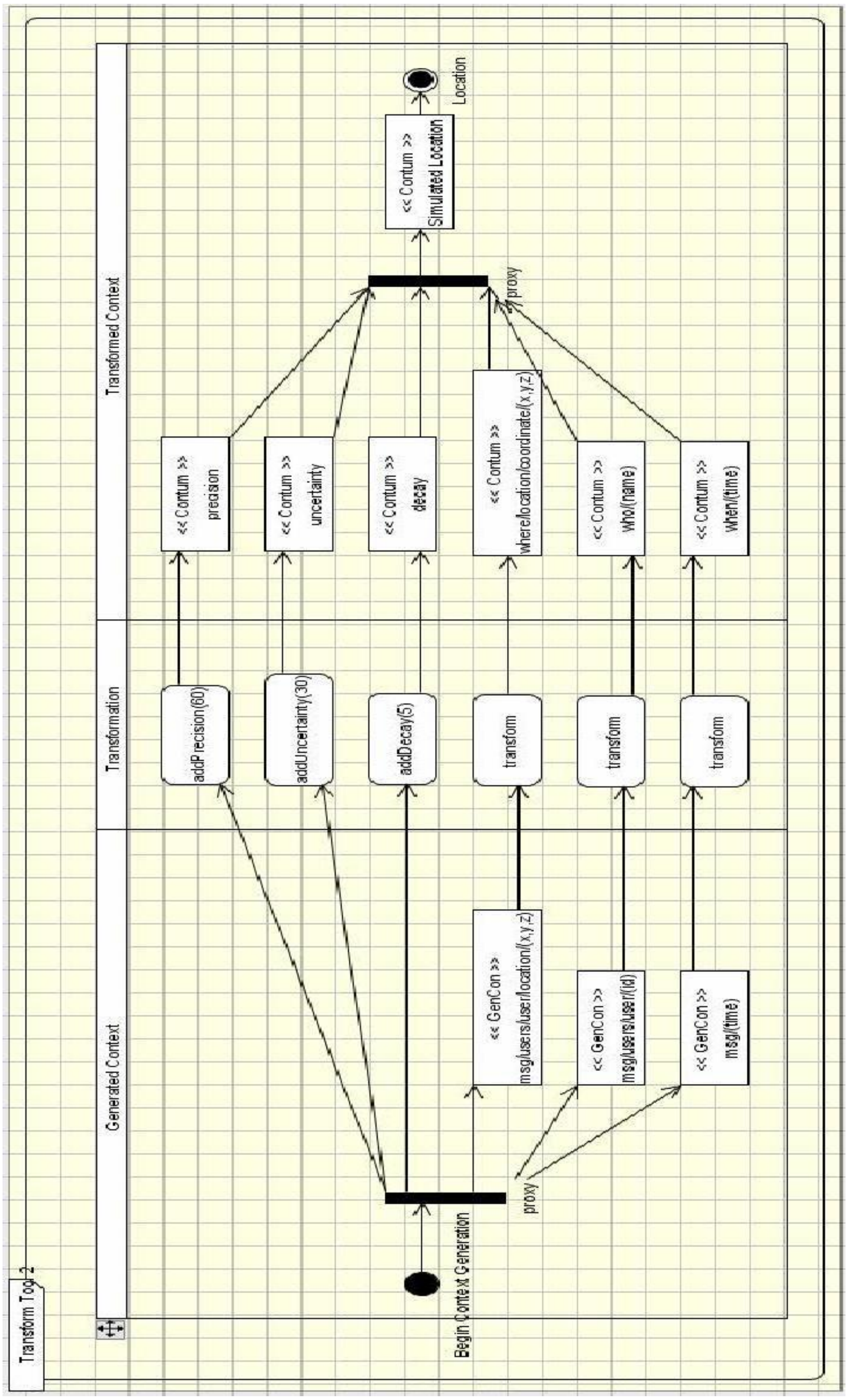


Figure 11-2 Prototype SimCon Tool (Activity Diagram)

11.1.1 Figures Usability Evaluation 1 SimCon V1

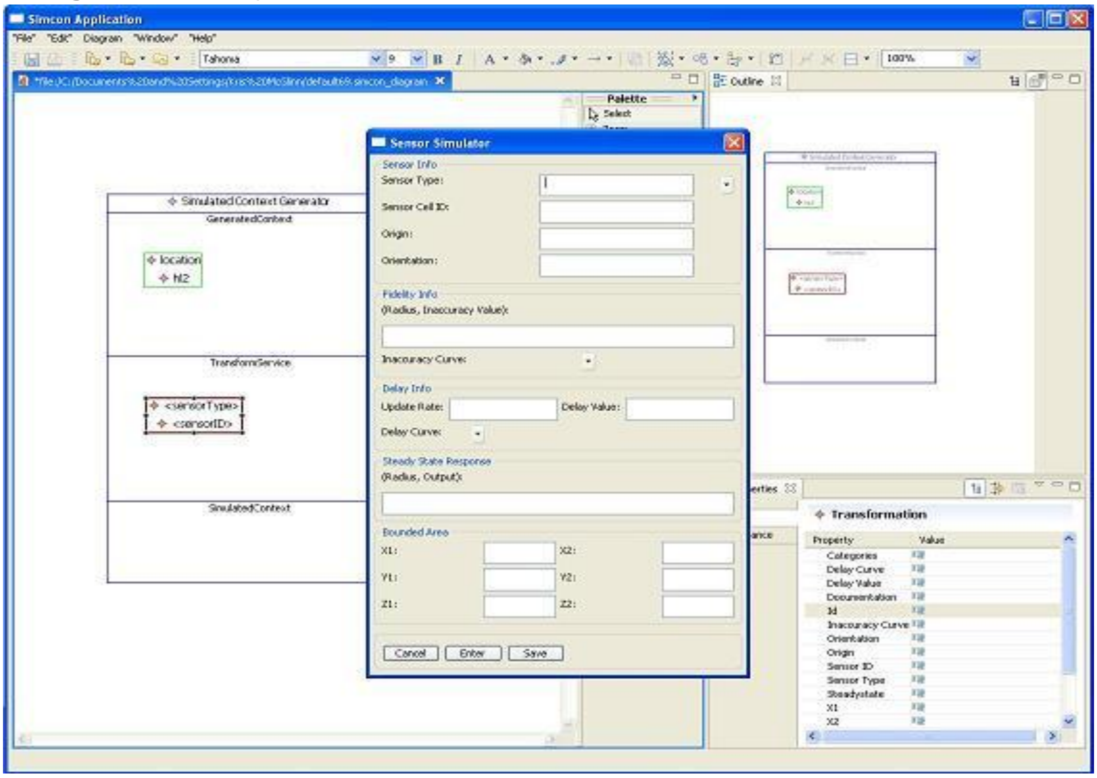


Figure 11-3 SimCon Configuration Tool Interface, Iteration 1

11.1.2 Figures Usability Two: SimCon Version 2 Evaluation 1

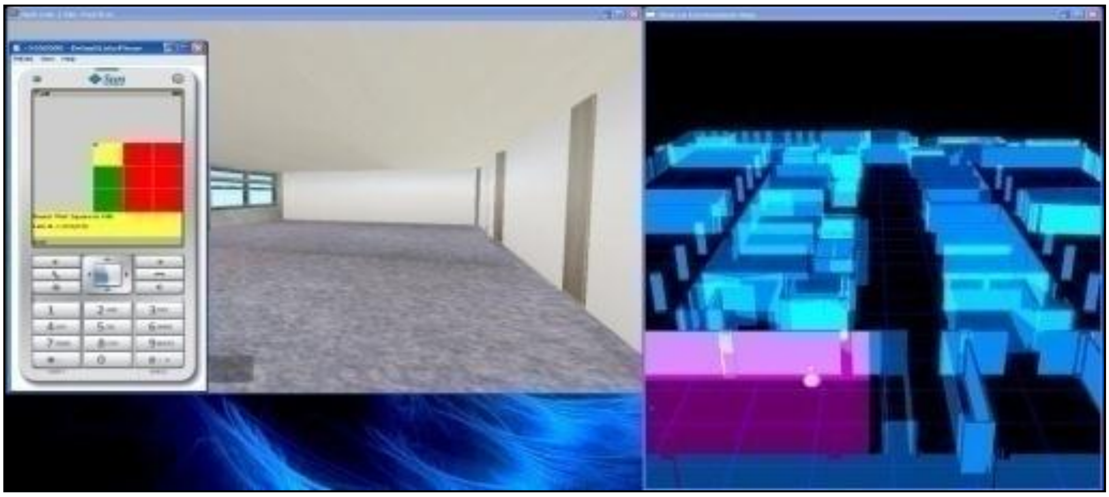


Figure 11-4 Maze Game and 3rd Person Environment and SimConViz Tool

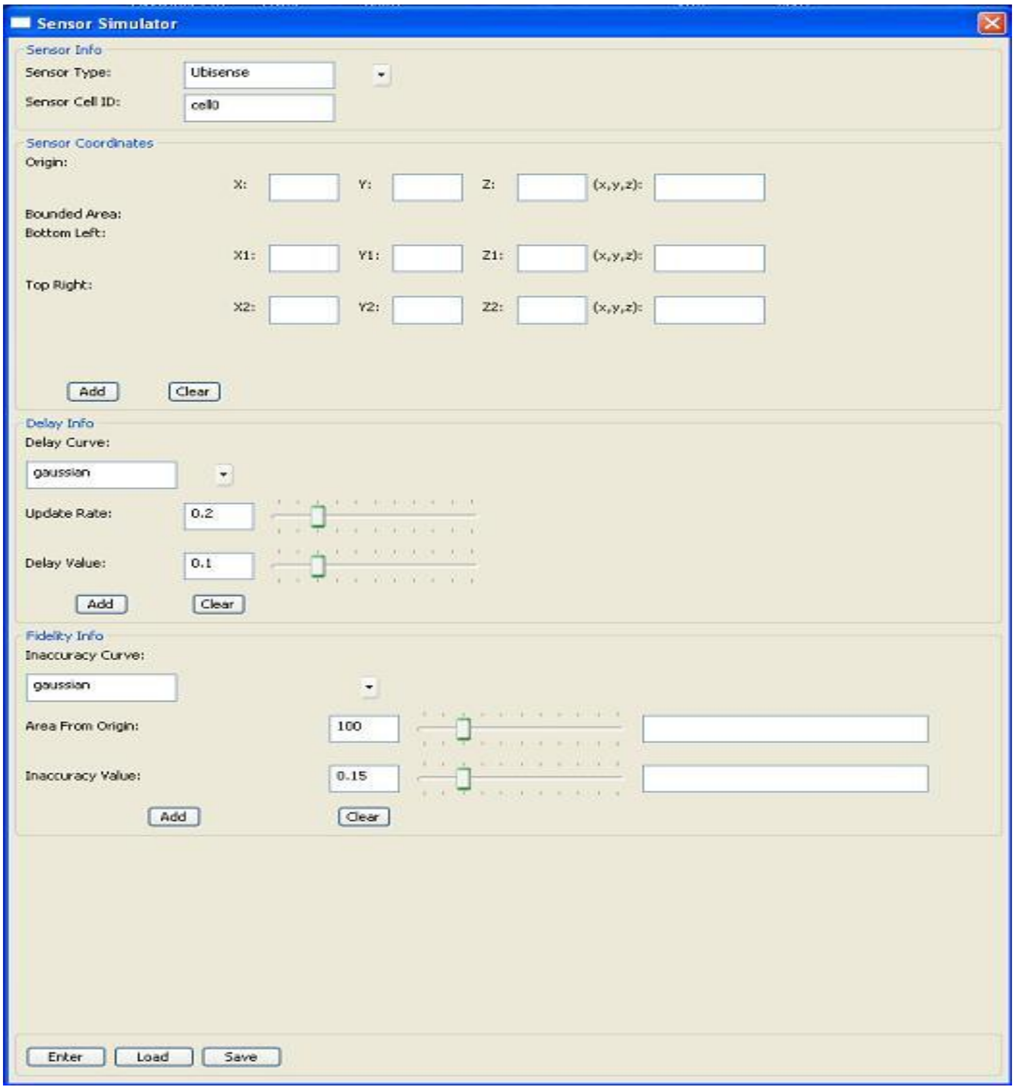


Figure 11-5 SimConfig tool bottom. SimConViz Tool.

11.1.3 Usability Two: SimCon Version 2 Evaluation 2

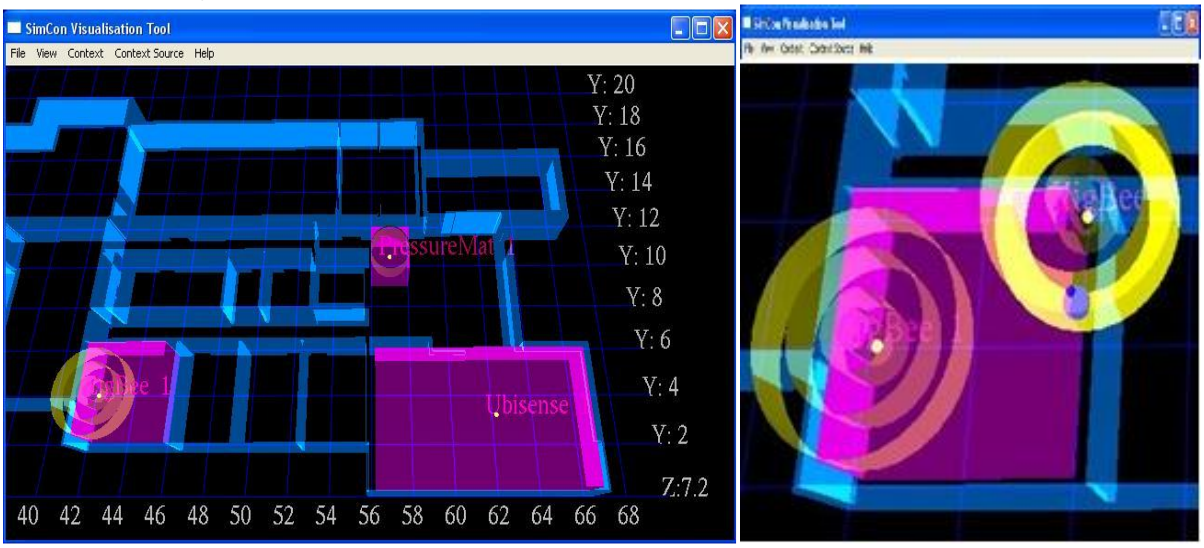


Figure 11-6 SimConfig SimCon source Visualisation and SimConViz Contum Visualisation

12 Appendix D: Pre-questionnaires and Backgrounds

12.1 Evaluation 1A

12.1.1 1A Pre-Questionnaire Findings

The pre-questionnaire consisted of the following questions and results (Figure 12-1). Q1 asked about their backgrounds and area of study (answered above).

- Q2: How would you rate your knowledge of context in relation to context-aware systems? (none, novice, intermediate, expert).
- Q3: How would you rate your knowledge of UML diagrams? (none, novice, intermediate, expert).
- Q4: Have you worked on designing a context-aware system? (never, infrequently, frequently, very frequently).
- Q5: Have you worked on evaluating a context-aware system? (never, infrequently, frequently, very frequently).

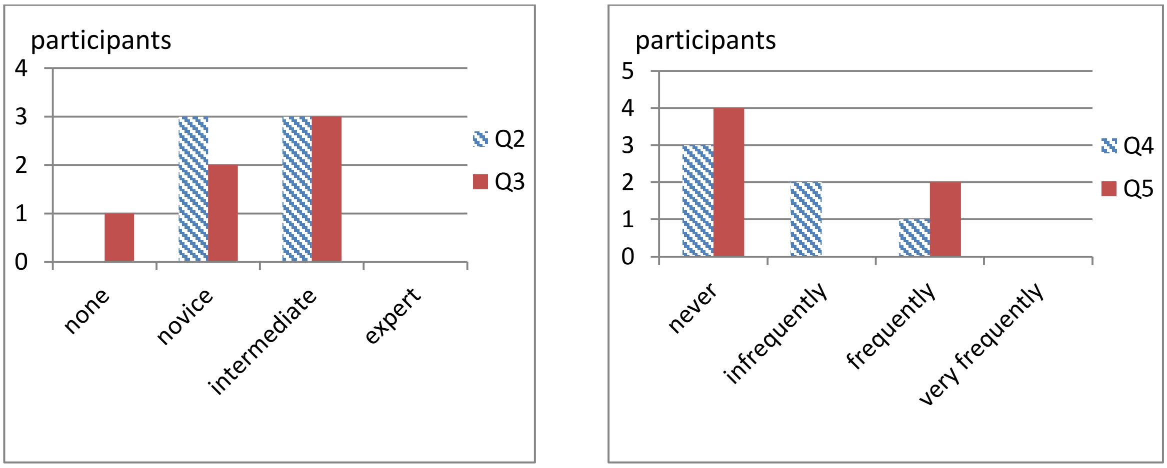


Figure 12-1 1A Pre-Questionnaire Answers

12.2 Evaluation 1B

12.2.1 Pre-Questionnaire

Table 12-1 1B Pre-questionnaire

Question	Yes	No
Q1: Do you or have you in the past conducted research in the area of development and/or evaluation of context-aware computer systems, or a related area (pervasive, ubiquitous computing, ambient intelligence etc.)?	2	3
Q2: Do you or have you in the past conducted research using sensor systems?	1	4

12.3 Evaluation 2A

12.3.1 2A Pre-Questionnaire

The pre-questionnaire consisted of the following questions and results (Figure 12-2).

- Q1: Have you studied indoor location sensor systems? (yes, no).
- Q2: Have you worked with indoor location sensor systems? (yes, no).
- Q3: Do you understand the term "location aware"?(yes, no).
- Q4: How would you rate experience with regard to indoor locations sensor systems? (expert, intermediate, novice, none).

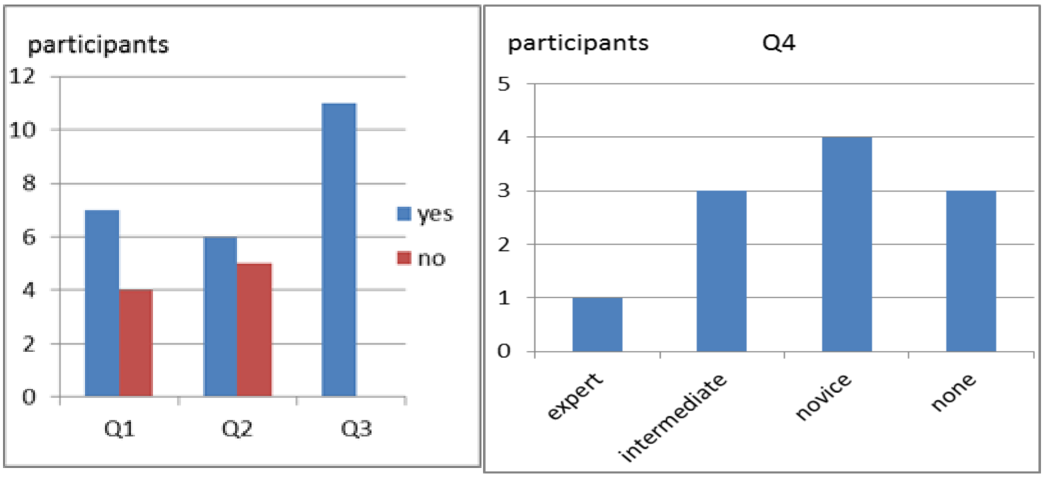


Figure 12-2 2A Pre-Questionnaire Answers

12.4 Evaluation 2B

12.4.1 2B Pre-Questionnaire

The pre-questionnaire consisted of the following questions.

- Q1: Have you worked with indoor location tracking sensor systems? (3 yes, 0 no).
- Q2: Have you worked with applications which use indoor location tracking sensor systems? (3 yes, 0 no).
- Q3: How would you rate your knowledge of indoor location tracking sensor systems? (none, novice, intermediate, expert).
- Q4: How would you rate your knowledge of applications which make use of indoor location tracking sensor systems? (none, novice, intermediate, expert).

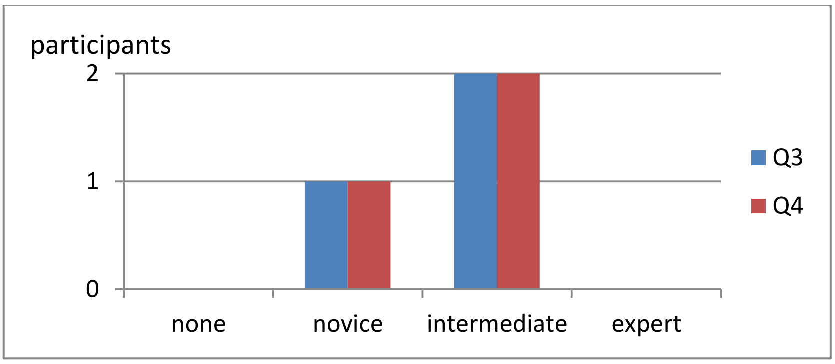


Figure 12-3 2B Pre-Questionnaire Answers

12.5 Evaluation 2C

12.5.1 Pre-Questionnaire

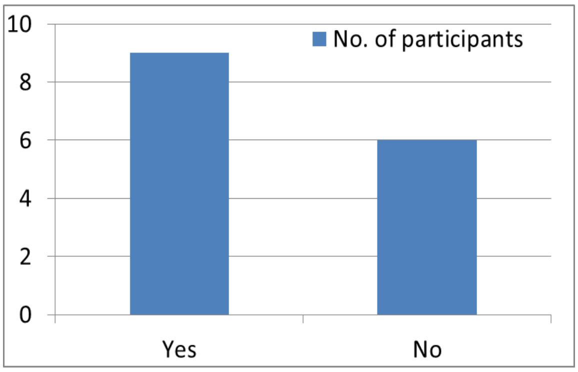


Figure 12-4 Have you ever used a Computer Aided Design (CAD) tool?

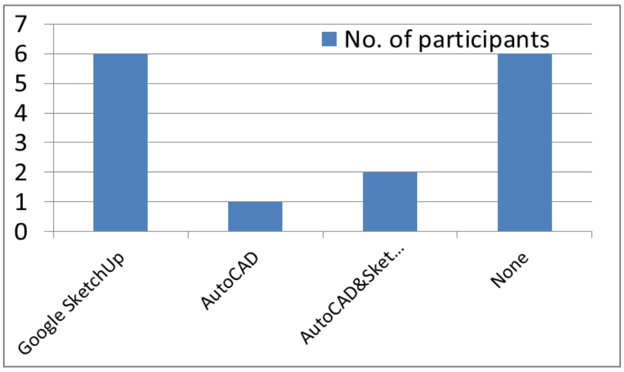


Figure 12-5 Can you name the tool(s)?

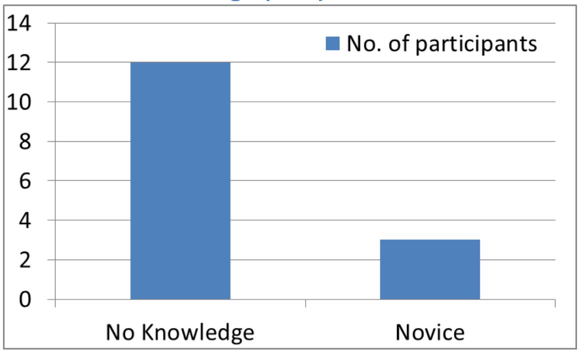


Figure 12-6 How would you rate your level of knowledge of sensor systems for detecting location?

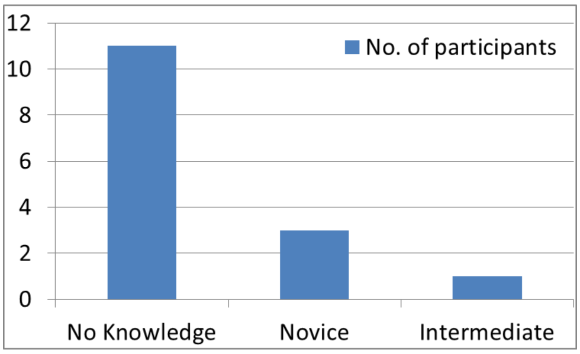


Figure 12-7 How would you rate your level of knowledge of SBAs?

12.5.2 Evaluation 3

12.5.2.1 Participant Backgrounds

Table 12-2 Background - No Experience

	15	24	1	16	22	14
Group (c = control, t = test)	c	c	c	t	t	t
Location-aware Applications	0	1	1	1	0	0
Location Sensors	0	0	0	0	0	1
Sensor and context simulation	0	0	0	1	1	1
Uncertainty and Uncertainty Simulation	0	0	0	0	0	0
Probability and Stochastic modelling	0	1	1	4	0	0

Table 12-3 Background - Novice

	21	7	3	5	13
Group (c = control, t = test)	c	c	t	t	t
Location-aware Applications	3	3	3	2	3
Location Sensors	1	3	3	1	3
Sensor and context simulation	0	2	2	1	2
Uncertainty and Uncertainty Simulation	0	2	1	2	3
Probability and Stochastic modelling	2	2	4	1	3

Table 12-4 Background - Intermediate

Intermediate	8	12	17	9	2	11
Group (c = control, t = test)	c	c	c	t	t	t
Location-aware Applications	5	6	6	6	5	5
Location Sensors	4	5	2	2	3	5
Sensor and context simulation	2	4	1	2	4	3
Uncertainty and Uncertainty Simulation	2	2	0	2	4	6
Probability and Stochastic modelling	0	3	1	2	4	6

Table 12-5 Background - Expert

Expert	18	6	19
Group (c = control, t = test)	t	c	t
Location-aware Applications	8	9	9
Location Sensors	5	3	1
Sensor and context simulation	5	0	3
Uncertainty and Uncertainty Simulation	2	0	2
Probability and Stochastic modelling	3	5	4

Table 12-6 Background Questionnaire - Questions and Answers (No experience = 0, Novice = 1, Intermediate = 2 Expert = 3)

	3	5	2	6	12	13	1	15	9	8	11	17	7	18	19	21	22	16	24

Group (c = control, t = test)	t	t	t	c	c	t	c	c	t	c	t	c	c	t	t	c	t	t	c
How would you rate your knowledge in location-aware applications?	1	1	2	3	2	1	0	0	2	2	2	2	1	2	3	1	0	1	0
How would you rate your knowledge in design of location-aware applications?	1	1	2	3	2	1	1	0	2	1	1	2	1	3	3	1	0	0	0
How would you rate your knowledge in evaluation of location-aware applications?	1	0	1	3	2	1	0	0	2	1	2	2	1	3	3	1	0	0	0
Location-aware Applications	3	2	5	9	6	3	1	0	6	5	5	6	3	8	9	3	0	1	0
How would you rate your knowledge in location sensors?	1	1	1	2	2	1	0	1	1	2	2	1	1	2	1	1	0	0	1
How would you rate your knowledge in design of location sensors?	1	0	1	0	2	1	0	0	0	1	2	0	1	1	0	0	0	0	0
How would you rate your knowledge in evaluation of location sensors?	1	0	1	1	1	1	0	0	1	1	1	1	1	2	0	0	0	0	0
Location Sensors	3	1	3	3	5	3	0	1	2	4	5	2	3	5	1	1	0	0	1
How would you rate your knowledge in sensor simulation?	1	1	2	0	2	1	0	0	1	1	3	1	1	3	3	0	1	0	0
How would you rate your knowledge in location context simulation?	1	0	2	0	2	1	0	1	1	1	0	0	1	2	0	0	0	1	0
Sensor and context simulation	2	1	4	0	4	2	0	1	2	2	3	1	2	5	3	0	1	1	0
How would you rate your knowledge in sensor/context uncertainty/error modelling?	0	1	2	0	1	2	0	0	1	1	3	0	1	2	2	0	0	0	0
How would you rate your knowledge in sensor/context uncertainty simulation?	1	1	2	0	1	1	0	0	1	1	3	0	1	0	0	0	0	0	0
Uncertainty and Uncertainty Simulation	1	2	4	0	2	3	0	0	2	2	6	0	2	2	2	0	0	0	0
How would you rate your knowledge of probability distributions?	2	1	2	2	2	2	1	0	1	0	3	1	1	3	2	2	0	2	1
How would you rate your knowledge of stochastic modelling?	2	0	2	3	1	1	0	0	1	0	3	0	1	0	2	0	0	2	0
Probability and Stochastic modelling	4	1	4	5	3	3	1	0	2	0	6	1	2	3	4	2	0	4	1

13 Appendix E: Significant Errors and Times

13.1 Errors

13.1.1 Evaluation 1A

Interface: Trouble navigating and using the interface. There were a number of issues with the interface. First and foremost was the layout of the swim lanes which caused confusion. Also, the additional functionality provided by Poseidon above that which was required to complete the tasks caused confusion. It was also observed that one user who had a background in UML found the non-adherence to the UML conventions for the diagrams confusing.

Terminology: The term contum was indicated to need further explanation.

13.1.2 Evaluation 1B

Significant errors occurred for the majority of participants related to the interface (position of buttons, tab features not being enabled). For example, to access the widget for placing and configuring the SimCon source required double clicking on free space within a “SimCon” node in the GMF interface. Often the participants would click on what appeared to be empty space, select the node, and when the SimCon source widget did not pop up, they would naturally assume there was another method to achieving this. This required the evaluator to step in and explain. The GMF prototype also proved to have a number of bugs. An early bug caused the GMF tool to slow down, making entries into the tool (of data) not save properly, which meant it would require re-entry. This time was recorded and was deducted from the overall time recorded for completing that task.

Some errors were also reported with the usability test approach: Two of the participants had issue with the map and related information being too small to read. The term “fidelity” in reference to SimCon source error distributions and delays was not clear to the participants.

13.1.3 Evaluation 2A

Using the interface. The issue with the GMF interface once again arose and was remedied during these evaluations. This had a significant impact on two of the usability tests as the evaluator was once again forced to intervene. A number of participants commented on the number of windows they were expected to navigate and this caused some confusion when attempting the tasks.

Other errors: Some features of the GMF tool caused problems with participants. For instance, when double clicking a SimCon source to open the pop up widget. While the need to click on empty space is been emphasised in the instructions, not all participants realise this, often resulting in some confusion. Once they succeed the issue is no longer a problem. Also, when connecting a SimCon source to its input and output in the GMF interface some participants have difficulty due to the particular type of drag feature on the "connector". These, it is hypothesised, are simply a result of the participant's own mental model which has been constructed through interactions with different operating systems and applications.

13.1.4 Evaluation 2B

Using the interface: Two participants had difficulty seeing the Z coordinate on the visualization tool and required consultation with the instructions.

Other errors: A number of functions with the user interface still cause problems for participants when they are first encountered. These include double clicking to access the configuration widget and the process by which a "connector" connects the "activities" which involves dragging the connector.

13.1.5 Evaluation 2C

The Virtual Reality interface caused a number of participants difficulty. It was discovered that the inverted mouse look made navigation around the VR environment particularly difficult for 2 participants. Errors with respect to the SimConfig interface continued with the GMF interface and issues which have been noted in earlier evaluations of SimCon Version 2, like clicking on SimCon sources to access the SWT widget for configuration. The number of screens remains an issue as participant must move between the GMF interface, the SimConViz and VR environment. It was observed that the need to both move the avatar and observe the simulated context in two separate windows was an issue for some users.

13.2 Times

13.2.1 Evaluation 1B

Table 13-1 Average time per task and standard deviation.

Participants	P1	P2	P3	4	5	Average Time	Standard Deviation
Task 1	2	1	2	6	1	**2**	**1.9**
Task 2	2	1	1	1	2	**1**	**0.5**
Task 3	13	16	11	13	19	**14**	**2.8**
Task 4	10	4	7	16	11	**10**	**4.0**
Task 5	12	11	6	21	10	**12**	**4.9**
Total Time	**39**	**33**	**27**	**57**	**43**	**40**	**11.4**

13.2.2 Evaluation 3

Table 13-2 Time to Complete Tasks: All Participants

	Intro + Task 1		Task 2		Task 3		Task 4		Task 5		Total Time	
	Avg.	Std.	Avg.	Std.	Avg.	Std.	Avg.	Std.	Avg.	Std.	Avg.	Std.
Control	9	1.9	5	1.7	8	2.5	10	2.4	11	3.1	43	5.5
Test	14	3.4	8	2.7	9	4.3	9	2.5	12	4.2	52	6.5

Table 13-3 Time to Complete Tasks: Non-Expert participants

	Intro + Task 1		Task 2		Task 3		Task 4		Task 5		Total Time	
	Avg.	Std.	Avg.	Std.	Avg.	Std.	Avg.	Std.	Avg.	Std.	Avg.	Std.
Control	14	2.6	5	0.4	9	4	10	2.1	13	7.0	51	6.4
Test	16	3.3	6	1.7	10	2.3	10	3.8	12	3.8	54	3.3

Table 13-4 Time to Complete Tasks: Expert participants

	Intro + Task 1		Task 2		Task 3		Task 4		Task 5		Total Time	
	Avg.	Std.	Avg.	Std.	Avg.	Std.	Avg.	Std.	Avg.	Std.	Avg.	Std.
Control	9	1.9	5	1.7	8	2.5	10	2.4	11	3.1	43	5.6
Test	14	3.4	8	2.7	9	4.3	9	2.5	12	4.3	52	6.5

14 Appendix F: Post-Questionnaires

14.1 Evaluation 1A

- Q1: Understanding the SimConfig Tool was? (very easy, easy, difficult, very difficult).
- Q2: How would you describe the SimConfig Tool interface? (very easy to use, easy to use, difficult to use, very difficult to use).
- Q3: Creating a new contum using activity diagrams was? (very easy, easy, difficult, very difficult).
- Q4: Creating a new contum using sequence diagrams was? (very easy, easy, difficult, very difficult).

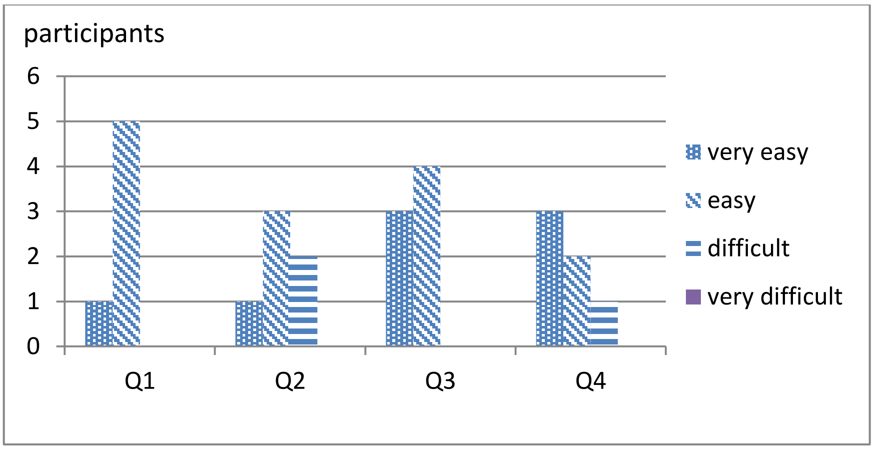


Figure 14-1 Post Questionnaire

14.2 Evaluation 2C

Table 14-1 Specific Usability Questions for Part 2 Participants

Visualisation and Uncertainty	14	11	6	13
How helpful was sensor data (context) visualisation for understanding how the security application views user location?	Helpful	Helpful	Helpful	Very Helpful
What effect do you think uncertainty in user location has on the behaviour of the security application?		May not know when to act accordingly		Difficulty distinguishing friend or foe entering building
How helpful was user location visualisation for understanding how uncertainty impacts on the application for detecting intruders?	Helpful	Helpful	Helpful	Neither Helpful nor Unhelpful

Configuration				
How easy was it to create a SimCon source type?	Neither easy nor difficult	Difficult	Neither easy nor difficult	Difficult
How easy did you find configuring SimCon uncertainty?	Easy	Neither easy nor difficult	Neither easy nor difficult	Neither easy nor difficult
How easy did you find visualising sensor data (context)?	Difficult	Neither easy nor difficult	Neither easy nor difficult	Easy

14.3 Evaluation 1B

1. Q1: How did you find the prerequisite information to understand? (very easy, easy, difficult, very difficult).
2. Q2: Downloading and opening the SimConfig Tool was? (very easy, easy, difficult, very difficult).
3. Q3: Creating a SimCon Generator was? (very easy, easy, difficult, very difficult).
4. Q4: Familiarising yourself with the SimConfig tool was? (very easy, easy, difficult, very difficult).
5. Q5: Creating a SimCon Ubisense Cell was? (very easy, easy, difficult, very difficult).
6. Q6: Setting the SimCon source area, delay and accuracy was? (very easy, easy, difficult, very difficult).
7. Q7: Configuring a SimCon source error distribution was? (very easy, easy, difficult, very difficult).
8. Q8: Configuring a SimCon source response curve was? (very easy, easy, difficult, very difficult).

14.4 Evaluation 2B: SUS Questions

- S1: I think that I would like to use this product frequently.
- S2: I found the product unnecessarily complex.
- S3: I thought the product was easy to use.
- S4: I think that I would need the support of a technical person to be able to use this product.
- S5: I found the various functions in this product were well integrated.
- S6: I thought there was too much inconsistency in this product.
- S7: I would imagine that most people would learn to use this product very quickly.
- S8: I found the product very awkward to use.
- S9: I felt very confident using the product.
- S10: I needed to learn a lot of things before I could get going with this product.

14.5 Evaluation 3

14.5.1 SimCon Tool Specific Questions

1) I found the instructions easy to understand?
2) Familiarising myself with SimConfig tool was difficult
3) Creating a SimCon source was challenging
4) I found placing a SimCon source was not difficult
5) I found visualising the SimCon sources position and boundaries an easy task
6) Configuring an uncertainty area was a challenge
7) I found the printed context output helped my analysis of the SBA

Additional Question (Test Group)

8) I thought the visualisation of context did not add much to my analysis of the security application

SimCon Tool Understanding (Both Groups) -

9) What effect did adding the generic uncertainty to Ubisense have on your positioning of the PIR sensors?
10) What effect did adding the uncertainty zone to Ubisense have on your positioning of the PIR sensors?

Additional Question (Test Group)

11) How did the visualisation of the simulated context (tagged location) affect your analysis of uncertainty?

12) Further Comments (Both Groups):

Table 14-2 SimCon Specific Questions Control Group: All Participants

	Strongly Agree	Agree	Neither Agree nor Disagree	Disagree	Strongly Disagree
1	1	7	1		
2		1		5	3
3		2	1	2	4
4	3	6			
5	3	6			
6	1	2	1	3	2
7	2	3	3	1	

Table 14-3 SimCon Specific Questions Control Group: Non-Expert

	Strongly Agree	Agree	Neither Agree nor Disagree	Disagree	Strongly Disagree
1	0	4	1		
2				3	2
3			1	2	2
4	1	4			
5	1	4			
6		1	1	2	1
7		2	3		

Table 14-4 SimCon Specific Questions Control Group: Expert

	Strongly Agree	Agree	Neither Agree nor Disagree	Disagree	Strongly Disagree
1	1	3			
2		1		2	1
3		2		2	
4	2	2			
5	2	2			
6	1	1		1	1
7	2	1		1	

Table 14-5 SimCon Specific Questions Test Group: All Participants

	Strongly Agree	Agree	Neither Agree nor Disagree	Disagree	Strongly Disagree
1	4	4	2		
2		1		8	1
3			1	7	2
4	3	7			
5	4	6			
6		1	1	5	3
7	1	2	4	3	
8			1	3	6

Table 14-6 SimCon Specific Questions Test Group: Non-Expert

	Strongly Agree	Agree	Neither Agree nor Disagree	Disagree	Strongly Disagree
1	2	3			
2		1		4	
3			1	3	1
4	2	3			
5	2	3			
6			1	3	1
7	1		2	2	
8				3	1

Table 14-7 SimCon Specific Questions Test Group: Expert

	Strongly Agree	Agree	Neither Agree nor Disagree	Disagree	Strongly Disagree
1	2	1	2		
2				4	1
3				4	1
4	1	4			
5	2	3			
6		1		2	2
7		2	2	1	
8					5

14.5.2 Evaluation 3: Post Questionnaire Comments

Q9 Control: What effect did adding the generic uncertainty to Ubisense have on your positioning of the PIR sensors?

13(B) - Not sure
6(B) - None - I had positioned them cautiously, allowing a trade-off between immediacy and ensuring fulfilling objective 1
7(B) - Allowed me to model a more natural scenario where users did not walk at the same speed or with the same pace in each situation
21(B) - It made me reconsider the position of the sensors so that they would not allow unauthorised access
8(B) - Uncertainty has to be factored in when placing the PIRs
17(B) - It introduces challenges with respect to guarantees on service.
12(B) - Blank
15(B) - Closer to the door was better to achieve minimal delay
24(B) - Allowed me to review my initial approach. At first I placed right beside the door area, but decided to move it due to the uncertainty.
1(B) - Gives an unpredictable result

Q9 Test: What effect did adding the generic uncertainty to Ubisense have on your positioning of the PIR sensors?

16(A) - No effect
11(A) – Nothing
19(A) - None, because I guessed well after reading the introduction thoroughly.
22(A) - I needed to move the PIR sensor further away from zone A
9(A) - I place the sensors away from doors
5(A) - It adds more randomness + real life effects to the experiments to show potential problems if sensors installed in real building
3(A) - I needed to think about the application in real world scenarios
14(A) - I saw the uncertainty in the zone would have no impact on my placement
18(A) - Blank
2(A) - Needed more space between the zones

Q10 Control: What effect did adding the uncertainty zone to Ubisense have on your positioning of the PIR sensors?

13(B) - Not sure
6(B) - Reinforced my belief that I had placed PIRA properly
7(B) - Allowed me to define tail-gaiting zones for users
21(B) - It made me reconsider again the position of PIRA due to uncertainty zone.
8(B) - As above…
17(B) - Blank
12(B) – Blank
15(B) - Moved sensor into uncertainty zone to see if that would improve accuracy. However I returned it after running the simulation.
1(B) – Blank

Q9 Test: What effect did adding the generic uncertainty to Ubisense have on your positioning of the PIR sensors?

16(A) - No effect
22(A) - I needed to move the PIR sensor further away from zone A
9(A) - I place the sensors away from doors
5(A) - It adds more randomness + real life effects to the experiments to show potential problems if sensors installed in real building
3(A) - I needed to think about the application in real world scenarios
11(A) – Nothing
19(A) - None, because I guessed well after reading the introduction thoroughly.
14(A) - I saw the uncertainty in the zone would have no impact on my placement
18(A) - Blank
2(A) - Needed more space between the zones

Q10 Test: What effect did adding the uncertainty zone to Ubisense have on your positioning of the PIR sensors?

16(A) - Move left sensor a bit further from the door.
22(A) – Moved the sensor away from it
9(A) - I place the sensors away from the zone
5(A) - I'd like to see an outline/halo outside of bounding box to show where potential events may be triggered
3(A) - Added complexity to the design which I had to account for in my positioning
11(A) – I re-positioned the sensor
19(A) - none, as above
14(A) - I moved the sensor away from the uncertainty zone
18(A) – Blank
2(A) - The application stopped working because of it, so I moved the sensor

Q11 Test: What effect did adding the uncertainty zone to Ubisense have on your positioning of the PIR sensors?

16(A) - The flickering blue ball kind of gives me the impression of uncertainty, which is useful.
22(A) – Blank
9(A) - It helped with my decision on the placement of the sensor
5(A) - useful to see as person moving through simulator as textual description would not be clear and it makes it easier to spot potential pitfalls in design
3(A) – Positively, due to seeing the zones…
11(A) – The jumping indicates there is some uncertainty, but it might be better to visualise as blurred sphere - give better impression of uncertainty.
19(A) - Real-time shows how much the positions can vary during time user is within zone
14(A) - Helped me understand the impact of uncertainty on the application
18(A) – Blank
2(A) - Quickly showed how the sensed location varied from the true location.

Q12 Comments:

Control:

13(B) – Blank
6(B) - Modification arrows to tweak positions rather than entering them
7(B) - 100% non-expert, found it easy to use and understand.
21(B) - Blank
8(B) - Blank
17(B) - Blank
12(B) – Blank
15(B) – Blank
1(B) – line numbers in xml view

Test:

16(A) - Blank
22(A) - Blank
9(A) - Blank
5(A) - Would like to have seen purple box move when entering coordinates
3(A) - Blank
11(A) – Blank
19(A) - Excellent 3D viz. Perhaps integrate GUI with uncertainty graphs. A time compression factor could speed up experiment (allow > 4 runs)
14(A) - Blank
18(A) – Blank
2(A) - Drag and drop zone placement or re-sizing would be very useful. Colour shading for uncertainty zones.

14.5.3 Standard Usability Scale

(0 – Strongly Disagree, 1 – Disagree, 2 – Neither Agree nor Disagree, 3 – Agree, 4 – Strongly Agree)

Table 14-8 SUS All Control Group

	6(B)	7(B)	21(B)	8(B)	17(B)	12(B)	15(B)	24(B)	1(B)
1	2	3	3	2	4	3	3	2	4
2	1	1	1	1	0	1	2	1	2
3	3	4	3	3	4	3	3	2	3
4	0	2	0	1	0	1	2	0	3
5	3	4	3	3	4	3	2	1	4
6	1	0	1	1	0	1	2	0	2
7	4	3	3	3	4	2	1	3	4
8	0	0	0	1	0	1	1	1	1
9	3	2	3	2	3	3	3	4	3
10	0	2	1	2	0	1	2	0	2

Table 14-9 SUS All Test Group

	16(A)	11(A)	19(A)	22(A)	9(A)	5(A)	3(A)	14(A)	18(A)	2(A)
1	3	1	3	3	3	4	3	3	4	3
2	0	2	2	0	1	1	1	1	1	0
3	3	2	4	3	3	4	3	3	3	3
4	3	2	1	1	0	1	3	1	1	0
5	3	2	4	3	3	3	3	3	2	3
6	1	1	0	0	1	1	0	1	1	0
7	3	3	3	3	3	3	4	3	3	3
8	1	2	1	0	1	1	0	1	1	0
9	2	2	4	3	3	3	3	4	3	3
10	2	1	1	1	0	2	1	1	1	2

15 Appendix G: Evaluation 3 Position of PIRs

15.1 All Participants

Table 15-1 Position of PIR Sensors Control Group: All participants

	PIRA			PIRB		
	15	17	19	23	25	27
Task 2	3	5	1	3	5	1
Task 3		1	8	8	1	
Task 4		3	6	6	2	1
Task 5		3	6	6	2	1

Table 15-2 Position of PIR Sensors Test Group: All participants

	PIRA			PIRB		
	15	17	19	23	25	27
Task 2	2	3	5	8	2	
Task 3		2	8	8	2	
Task 4		3	7	8	2	
Task 5	1	8	1	8	2	

15.2 Non-Experts

Table 15-3 Position of PIR Sensors Control Group: Non-Expert participants

	PIRA			PIRB		
	15	17	19	23	25	27
Task 2	2	2	1	3	1	1
Task 3			5	5		
Task 4		1	4	4		1
Task 5		1	4	4		1

Table 15-4 Position of PIR Sensors Test Group: Non-Expert participants

	PIRA			PIRB		
	15	17	19	23	25	27
Task 2	1	2	2	2	2	1
Task 3		1	4	4	1	
Task 4		1	4	4	1	
Task 5	1	3	1	3	2	

15.3 Experts

Table 15-5 Position of PIR Sensors Control Group: Expert Users

	PIRA			PIRB		
	15	17	19	23	25	27
Task 2	1	3			4	
Task 3		1	3	3	1	
Task 4		2	2	2	2	
Task 5		2	2	2	2	

Table 15-6 Position of PIR Sensors Test Group: Expert Users

	PIRA			PIRB		
	15	17	19	23	25	27
Task 2	1	1	3	4	1	
Task 3		1	4	4	1	
Task 4		2	3	4	1	
Task 5		5		5		

16 Appendix H: Experimental Instructions

16.1 Evaluation 2C

16.1.1 Part 1

16.1.2 Introduction

You are a building manager. You wish to create a system to track both authorised and unauthorised users coming in and out of the building. You wish to see what type of sensor data (context) best meets your requirements. In this experiment you will examine a pressure mat to detect when a user moves in and out of the building. This system will not provide identification of the user; it will simply generate an alert when someone walks over it. It is your task to use SimCon to create, place and configure the pressure mat and then examine the output.

16.1.3 Tasks

You have in front of you three screens. One screen is a Virtual Reality model of the NIMBUS building in Cork. The second screen is the SimConViz tool for visualising a building, simulated context sources and also simulated context. The third screen is the SimConfig tool for creating, placing and configuring simulated context sources. Could you please now do the tasks on the following pages (instructions are provided).

16.1.3.1 *Task 1 - Familiarise yourself with the VR building and SimConViz tools*

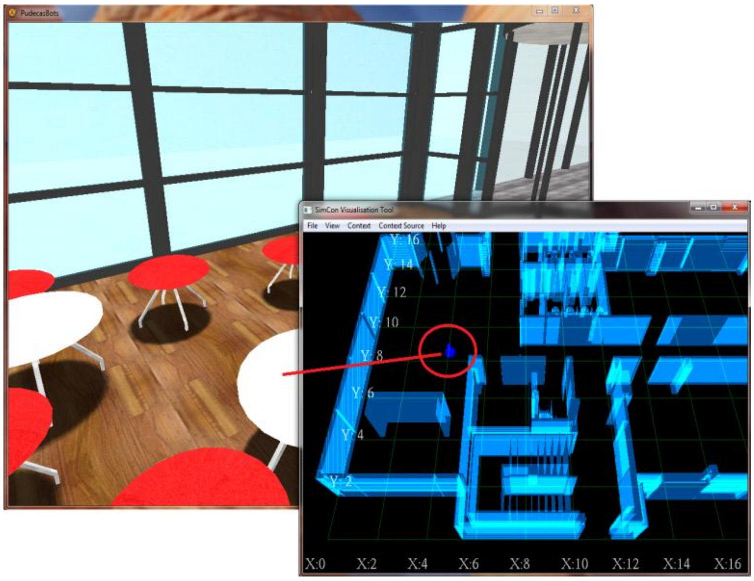
In this task you will familiarise yourself with the VR building and SimConViz tool.

Task 1 Instructions

You can move around the VR building using the following keys "w" = forward (in the direction you are looking), "s" = backward, "a" = strafe left, "d" = strafe right. Familiarise yourself with moving around. You will also notice in the SimConViz tool a small blue shape representing your avatar moving around. This is your actual location in real time. **PLEASE STICK TO THE ENTRANCE AND THE ROOM WITH THE CHAIRS AS YOU WILL BE TRACKING PEOPLE USING THIS ENTRANCE.**

You will also notice in the SimConViz tool a small blue shape representing your avatar moving around. This is your actual location in real time.

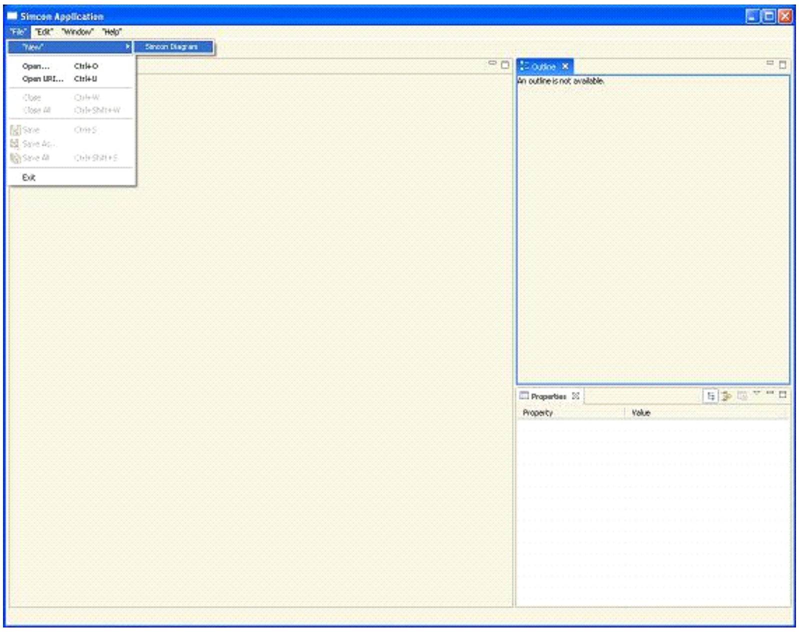
The SimConViz tool will allow you to see where you are placing context sources inside the Virtual Environment. Use the mouse wheel to zoom in and out of the map, hold the right click to move the map left right and up and down, and hold the left mouse button to rotate the map.



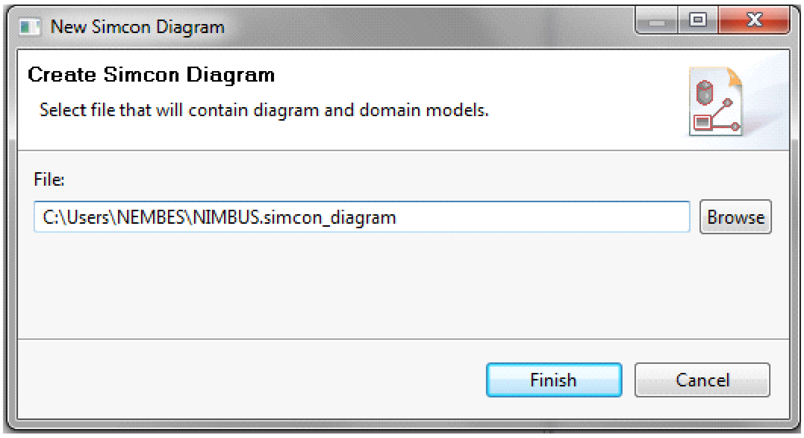
16.1.3.2 Task 2 - Familiarise yourself with SimCon Configuration Tool

You will now familiarise yourself with the SimConfig tool for creating, placing and configuring simulated context sources in a VR building. Click on the SimConfig tool.

Go to File->new->SimCon Diagram.



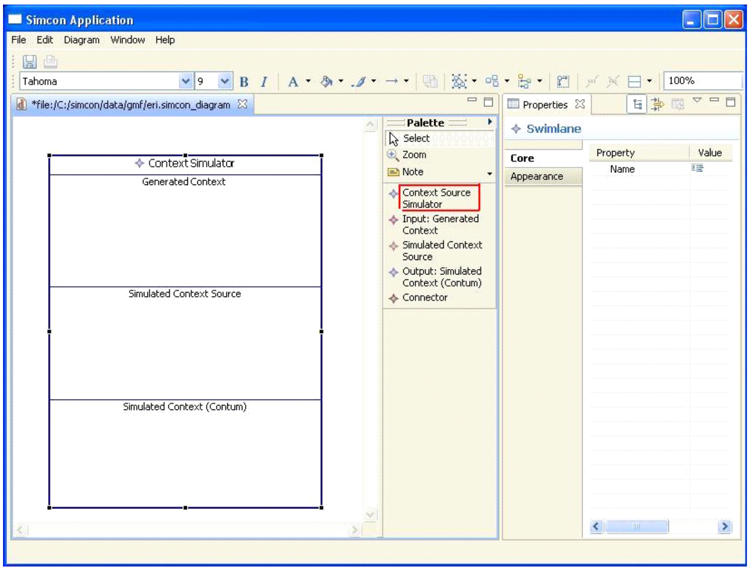
Call the file NIMBUS.simcon_diagram and save it in "C:/simcon/data/gmf/". (The file should already exist, so replace it).



Click Finish to create.

You will now see a window opened up in front of you with a menu on the right hand side.

Select "Context Source Simulator" from the menu and click anywhere on the canvas.



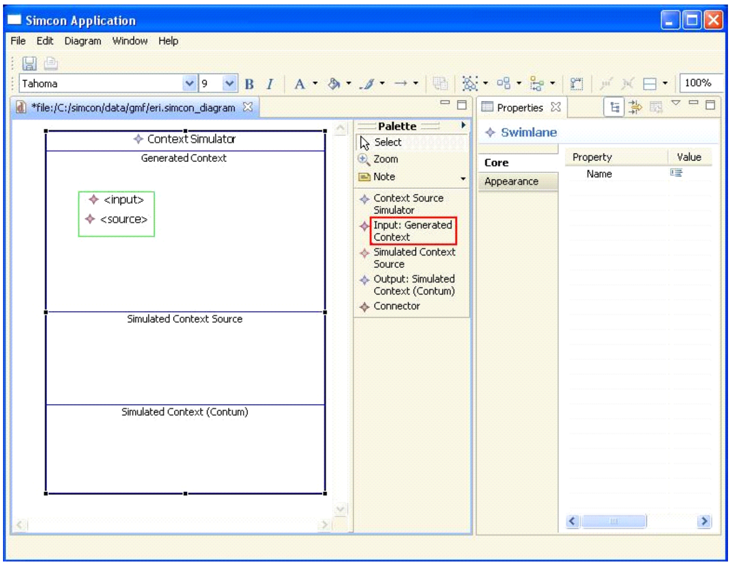
Familiarise yourself with the layout. There is a "Swimlane" which consists of three boxes: "Inpuut: Generated Context", "Simulated Context Source" and "Output: Simulated Context (Contum)". You can adjust the size of the "Context Source Simulator" if required by selecting its edges.

16.1.3.3 Task 3 - Create a pressure mat at the door of the NIMBUS building.

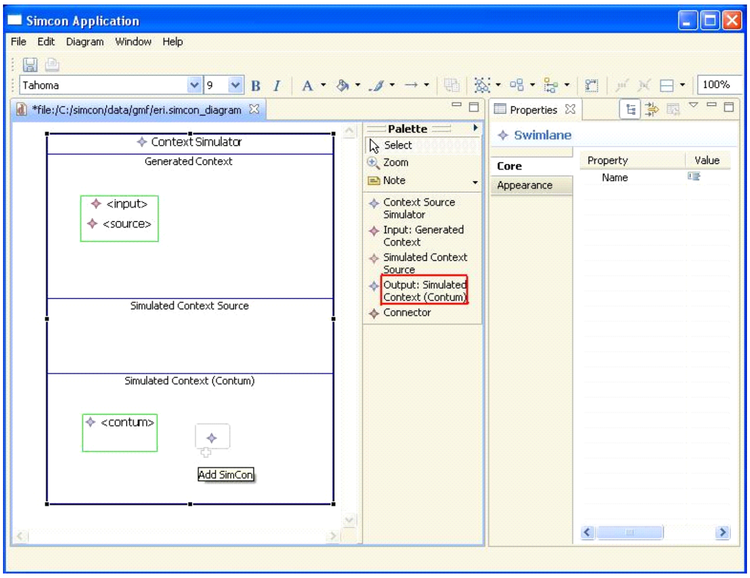
Create a pressure mat using the SimConfig tool at the door of the building in order to track when someone enters and leaves the building. **When you have created and placed the SimCon Source, alert the instructor**. The instructor will start up the Simulated Context Generator for you and you can then observe the context as it is generated by the SimCon Source.

Task 3 Instructions

Begin by creating an "Input: Generated Context" by clicking on this in the menu on the right (marked in red in fig 1). Now press inside the "Generated Context" box in the main canvas where you created your Context Simulator. A new bubble should appear (as in fig 1).

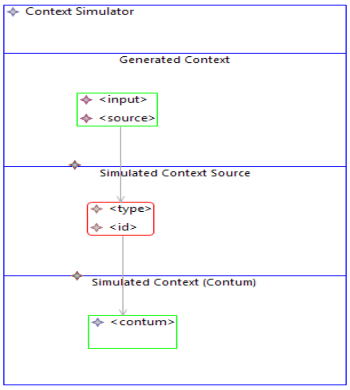


Now create an "Output: Simulated Context" bubble in the "Simulated Context (Contum)" box. You need not fill in any values for this.

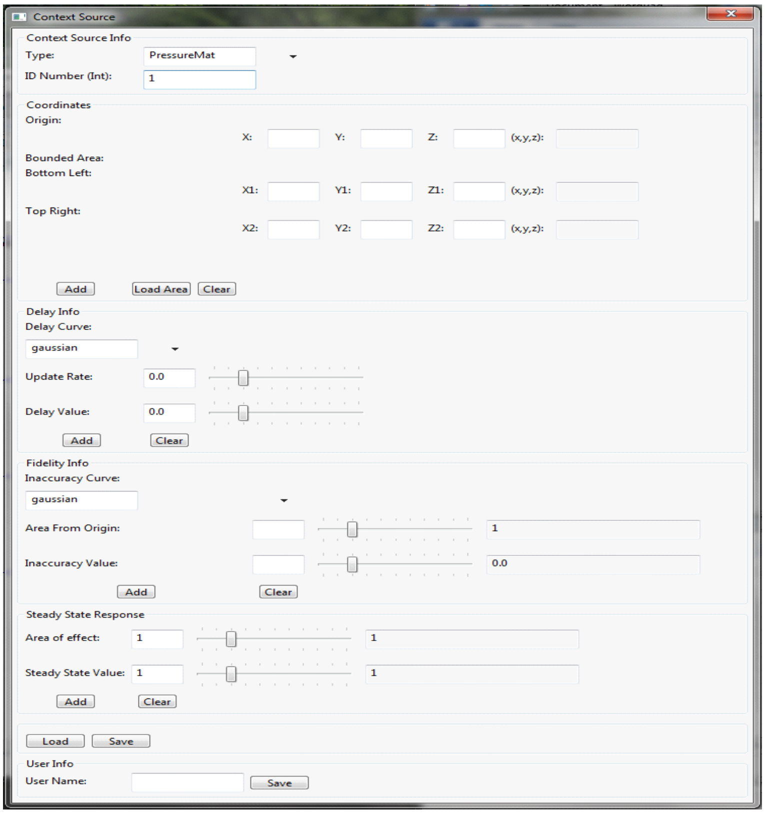


Now create a "Simulated Context Source" in the "Simulated Context Source" box in the same manner.

Now press the connector button in the menu, click on the Input you placed in the "Generated Context" box and keeping your finger pressed on the mouse button, drag this connector to the Context Simulator bubble. Repeat this procedure to connect the Context Simulator bubble to the Simulated Context bubble. It should now look like this:



Now double click on some empty space in the centre Context Source bubble you created. The following window should appear. (You need to click on an empty space in the bubble, so you may need to expand the bubble to do this.) Now select Pressure Mat from the drop down menu beside "Type". Default values will be placed into a number of boxes. Ignore all of these except for the Coordinates.

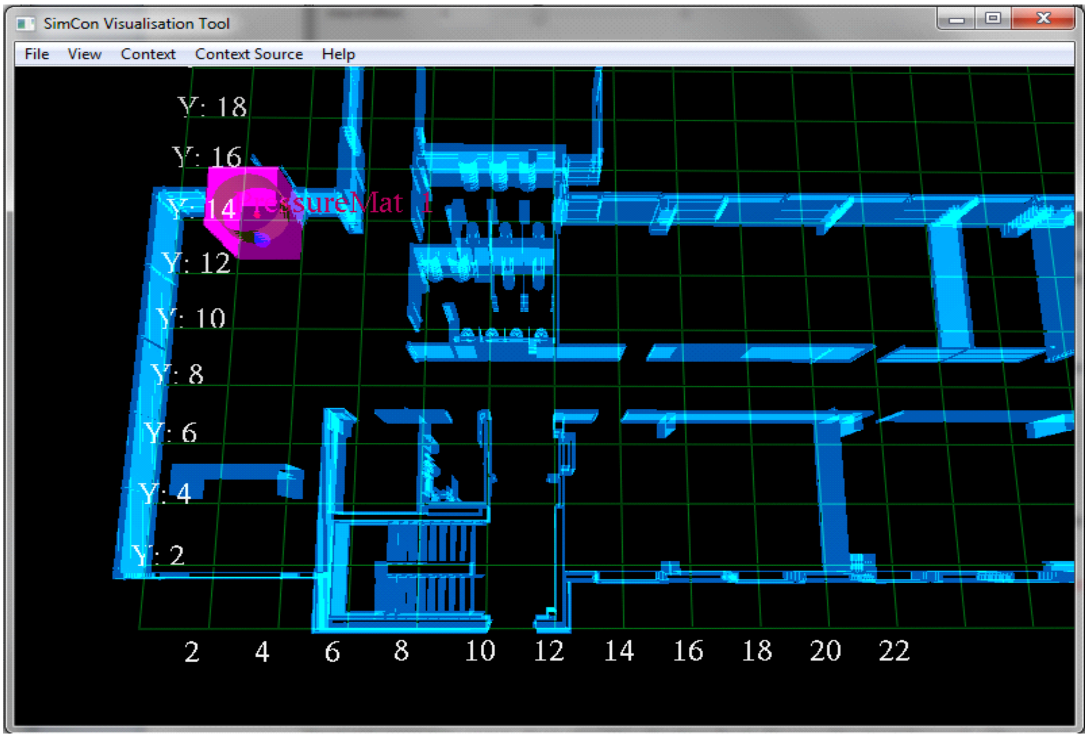


Using the SimCon Visualisation Tool examine the grid coordinates X and Y. The X runs along the bottom of the screen, Y along the side. These represent distance in meters from the origin; the floor height here is 0 meters (Z is 0).

Now choose an area of the building and enter values for lower bounded area and upper bounded area into the appropriate boxes and click "Add". You can see by the coordinates on the map, the front door of the NIMBUS building is roughly at point X = 3 and Y = 13. You need to define the "bounded area" of the Pressure Mat, so try the values X1 = 2, X2 = 4, Y1 = 12, Y2 = 14 and Z 1 = 0, and Z2 = 3.

If you do not set an origin, it will be calculated for you as the centre of the bounded area.

Now click "Add" for the delay and fidelity values and then click the "Save" button and save the sensorXML description as PressureMat1.xml in c:\simcon\data\sensorXML\. You will now see your SimCon source in the SimConViz tool.



You have now returned to the Main SimConfig screen. Click the save icon in the top left hand corner. Now inform the instructor you have finished, and he will begin SimCon so that you can analyse the output.

Walk to the pressure mat and stand in the area. Notice that when you walk into the doorway where you placed the pressure mat, in SimConViz it becomes highlighted to indicate you have alerted the system to your presence. (You may need to click back onto the SimConViz tool to get the visualisation. You can do this by alt tabbing and clicking on. If you have any troubles ask the instructor).

16.1.4 Part 2 (Group B)

16.1.5 Introduction

In the first part of this experiment you created and placed a Pressure Mat SimCon Source in a Virtual Reality building and you examined the output of that context source. In order for the security application to identify unauthorised users, it must be able to detect when an authorised user is triggering the pressure mat. In this exercise you wish to analyse sensor data (context) which tracks and identifies a user in order to track both authorised and unauthorised users and determine whether it meets your requirements. **YOU WILL PAY SPECIAL ATTENTION TO HOW UNCERTAINTY IN USER LOCATION WILL EFFECT THE APPLICATION.**

It is your task to use SimCon to create, place and configure a Ubisense Real Time Location SimCon Source which identifies a user and gives their 3D coordinates. You will also examine the output and see whether it meets your requirements.

16.1.6 Tasks

Using the previous set up (VR, SimConfig and SimConViz) could you please now do the tasks on the following pages.

16.1.6.1 Task 1 - Create a ubisense cell at the entrance of the NIMBUS building.

Create a ubisense cell using the SimConfig tool at the entrance of the building in order to track who is entering and leaving the building. You will notice that the pressure mat you created before is still there. Create the ubisense cell alongside in a similar manner. When you have created and placed the SimCon Source, alert the instructor. The instructor will start up the Simulated Context Generator for you and you can then observe the context as it is generated by the SimCon Source.

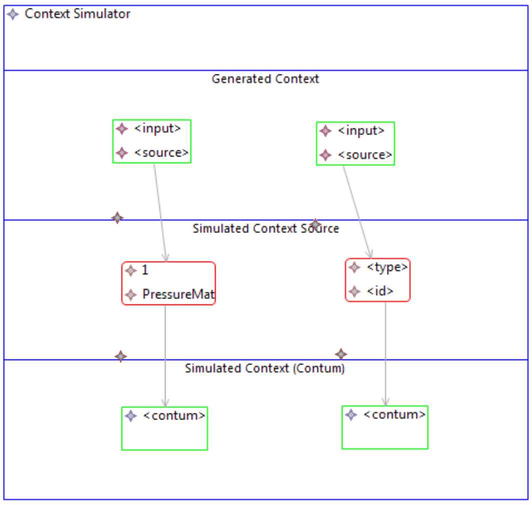
Instructions Task 1

Begin by creating an "Input: Generated Context" by clicking on this in the menu on the right (marked in red in fig 1). Now press inside the "Generated Context" box in the main canvas where you created your Context Simulator. A new bubble should appear (as in fig 1). You need not fill in any values for this.

Now create an "Output: Simulated Context" bubble in the "Simulated Context (Contum)" box. Now create a "Simulated Context Source" in the "Simulated Context Source" box in the same manner.

Now press the connector button in the menu, click on the Input you placed in the "Generated Context" box and keeping your finger pressed on the mouse button, drag this connector to the Simulated Context Source bubble.

Repeat this procedure to connect the Simulated Context Source bubble to the Simulated Context (Contum) bubble. It should now look like this:

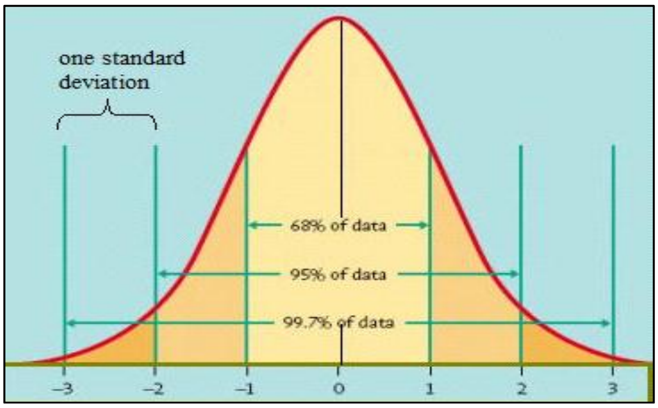


Now double click on some empty space in the centre Context Source bubble you created. The following window should appear. (You need to click on an empty space in the bubble, so you may need to expand the bubble to do this.) Now select Ubisense Cell from the drop down menu beside "Type". Default values will be placed into a number of boxes. Ignore all of these for the minute except for the Coordinates.

You may notice that the previous coordinates for the Pressure Mat are still in the SimCon Source Position. If so press “Clear”. Now choose an area of the building and enter values for lower bounded area and upper bounded area into the appropriate boxes and click "Add". You need to define the "bounded area" of the Ubisense cell, so try the values X1 = 0, X2 = 8, Y1 = 7, Y2 = 14, Z 1 = 0, and Z2 = 2.

If you do not set an origin, it will be calculated for you as the centre of the bounded area.

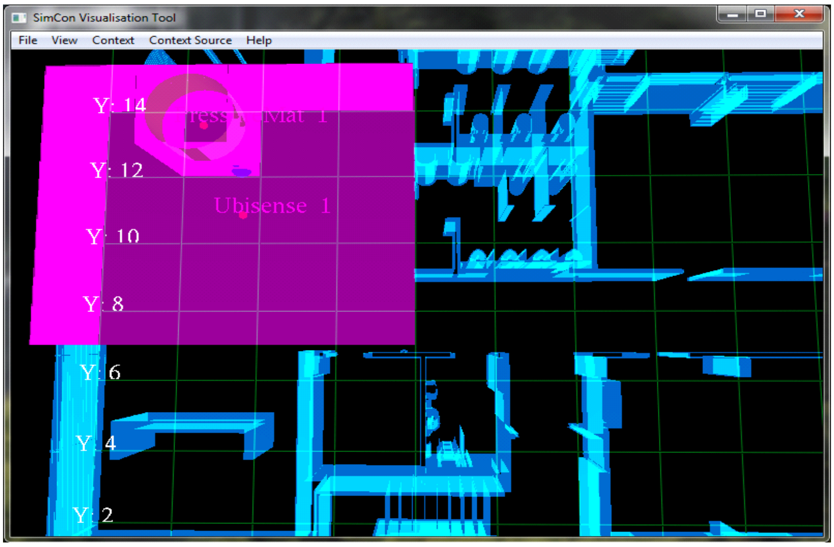
Now look at the Measurement rate. The Ubisense cell gives a measurement rate of 0.2 which means it will generate messages every fifth of a second. There is also 0.1 delay which has Gaussian (normal) probability density function which means that there will be an added delay of up to a tenth of a second 68% of the time, up to a fifth of a second 95% of the time and up to .3 of a second 98% of the time.



Now look at the Uncertainty. The area of effect indicates the distance from the SimCon source origin where uncertainty will be introduced. Once again a variance is introduced using a Gaussian distribution, meaning that the location will fall up to 0.15 meters from its true location 98% of the time, etc.

Now click both "Add" buttons to enter the delay and uncertainty values into the model.

Once you have completed this, click the "Save" button and save the sensorML description as Ubisense1.xml in c:\simcon\data\sensorXML\.

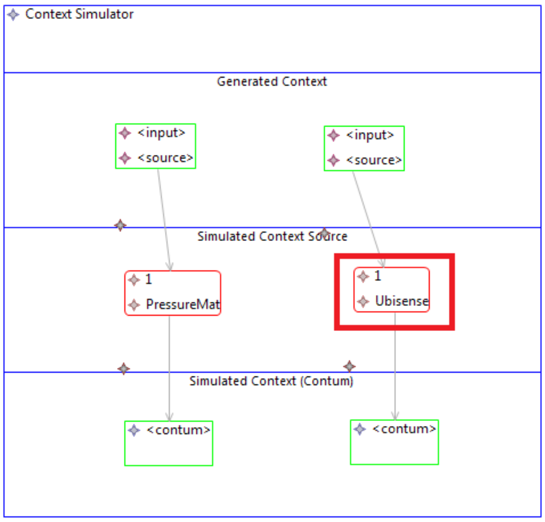


You have now returned to the Main SimConfig screen. Click the save icon in the top left hand corner and inform the instructor who will start up SimCon. You can now analyse the Ubisense output.

16.1.6.2 Task 2 - Examine the ubisense output with respect to the security application and reconfigure the uncertainty

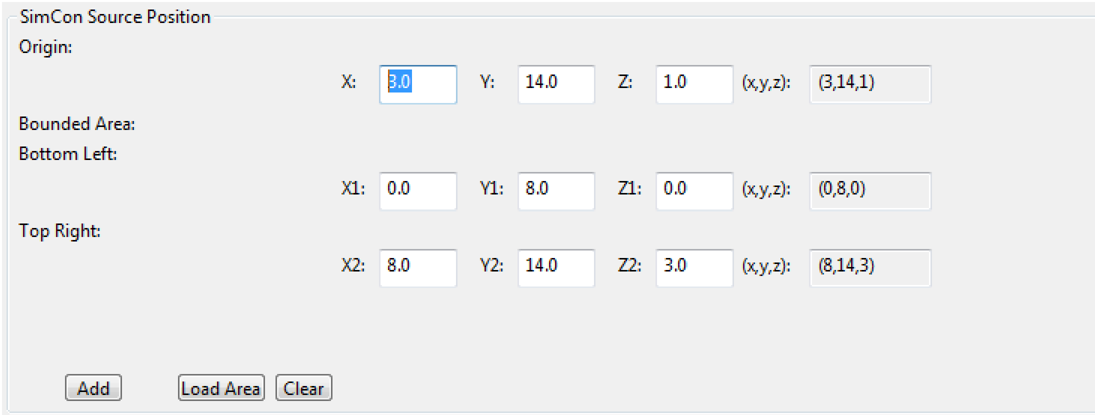
In order for the security application to identify unauthorised users, it must be able to detect when an authorised user is triggering the pressure mat. Examine the ubisense output and determine whether it meets this requirement. You will notice a small green man indicating your location and given an identification. You will notice this also jumps. **MOVE TO THE AREA WHERE THE PRESSURE MAT IS.** (For a better view you may turn off the bounded areas using SimConViz: Context Source –> Bounded Area). Notice how the location is shaky due to **uncertainty** in the location sensed. You will now increase the uncertainty exhibited around the door.

Go back to the SimConfig tool and double click on the Ubisense Simulated Context Source



Now create an area of increase uncertainty within the Ubisense cell. To do this you must specify the area where you wish to increase the uncertainty. Let's examine how an increase in uncertainty, close to the door (as the door is glass), will impact on the security application.

This is done by moving the origin of the sensor. Change the origin of the sensor to the following values X = 3, Y = 14 and Z = 1. It should now look like this -



Click "Add".

Next you must adjust the area of **uncertainty**. Create an area of increased uncertainty within two meters of the new origin (covering the area around the door). First go to the section which says “**Uncertainty**”. Now press "Clear". Next put the value "2" beside the Area of Effect. Next put the value "0.3" inside the box for "Inaccuracy Value". Choose a Gaussian probabilistic distribution. Press "Add". This means that when the avatar is within two meters of the sensor origin, the coordinate returned will vary by up to 0.3 meters from its actual location 68 percent of the time.

You must also configure the uncertainty for the rest of the Ubisense cell. Now enter a new value of "20" beside the same Area of Effect. Next put the value "0.15" inside the box for "Inaccuracy Value". Press "Add".

Now look at the Measurement rate. First press the "Clear" button. Enter the value 1.0 for the measurement rate. This means it will generate messages every second. Increase the delay to 0.5 using a Gaussian (normal) probability density function. This which means that there will be an added delay of up to 0.5 seconds 68 percent of the time.

Once you have completed this, click the "Save" button and save the sensorML description as Ubisense1.xml in c:\simcon\data\sensorXML\.

You have now returned to the Main SimConfig screen. Click the save icon in the top left hand corner and inform the instructor who will start up SimCon.

You can now analyse the Ubisense output again using SimConViz. **WALK TO THE PRESSURE MAT AND EXAMINE THE OUTPUT**. Notice how the location is shaky due to **uncertainty** in the location sensed. Take some time to consider how this uncertainty (inaccuracy in the location) will affect your security application?

Now please ask the instructor for the final questionnaire. Thank you.

Take some time to consider how this uncertainty (inaccuracy in the location) will affect your security application?

Now please ask the instructor for the final questionnaire. Thank you.

16.2 Evaluation 3

16.2.1 Introduction

You are a building manager. You are installing a security system to monitor a secure entrance to a building area. Ubisense has been installed in a corridor to track tagged (authorised) users (Figure 6-20). Ubisense is a Real Time Tracking System which through the use of tags (a mobile transmitter which can be carried by a person) returns a 3D coordinate for any tag in a Ubisense "cell" or zone.

The Ubisense Zone covers ten meters of a corridor, the area from x = 16 to x = 26 (Figure 6-20). An area of the Ubisense Zone labelled **zoneA** covers two meters in front of the door (x = 20 and x = 22). When a tagged user (**userA**) enters this zone an automatic door (**doorX**) opens. Unfortunately, there is 0.8 second window in which a non-tagged (non-authorised, **user?**) could pass through the door before it closes once **userA** has left **zoneA**. Non-tagged users have; as a result, been able to gain access by tailgating (following closely behind) tagged users.

Your task is to design the placement of two Passive Infrared (PIR) motion sensors (on either side of the door) to determine if a non-tagged user is tailgating when a tagged user opens the door. A PIR is an electronic device that measures infrared (IR) light radiating from objects in its field of view, so that a change in the amount if IR (e.g. due to a moving person's body heat) will generate an event. You will test your approach using a context simulator called "SimCon" which provides tools for placing, configuring and visualising simulated context sources (like sensors). Only one tagged **userA** and one non-tagged **user?** must be considered in this simulation scenario.

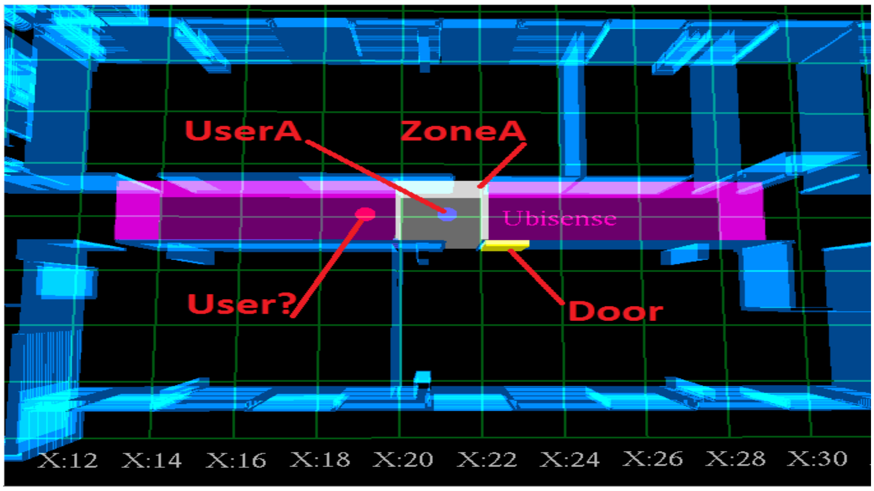


Figure 16-1 The corridor with a Ubisense cell (purple), zoneA (white) in front of the open door (yellow) and user? (red) and userA (blue).

Figure 6-22 gives an overview of four situations which your placement of the PIR sensors must account for:

1) **userA** walks through **door**.
2) **user?** walks past **door**.
3) **userA** enters **door** and **user?** does not attempt to enter **door**.
4) **userA** attempts to enter **door** and **user?** also attempts to enter **door**.

Ask your instructor to now run these four scenarios. You will see them displayed through the SimConViz tool. You will also be shown the outputs from the Ubisense system and the state changes of the Security Application* as it behaves before the PIR sensors are placed. **Notice that Ubisense outputs are only generated inside the Ubisense zone and that that an authorised user needs to be in Zone A for the door to open.** Familiarise yourself with these situations.

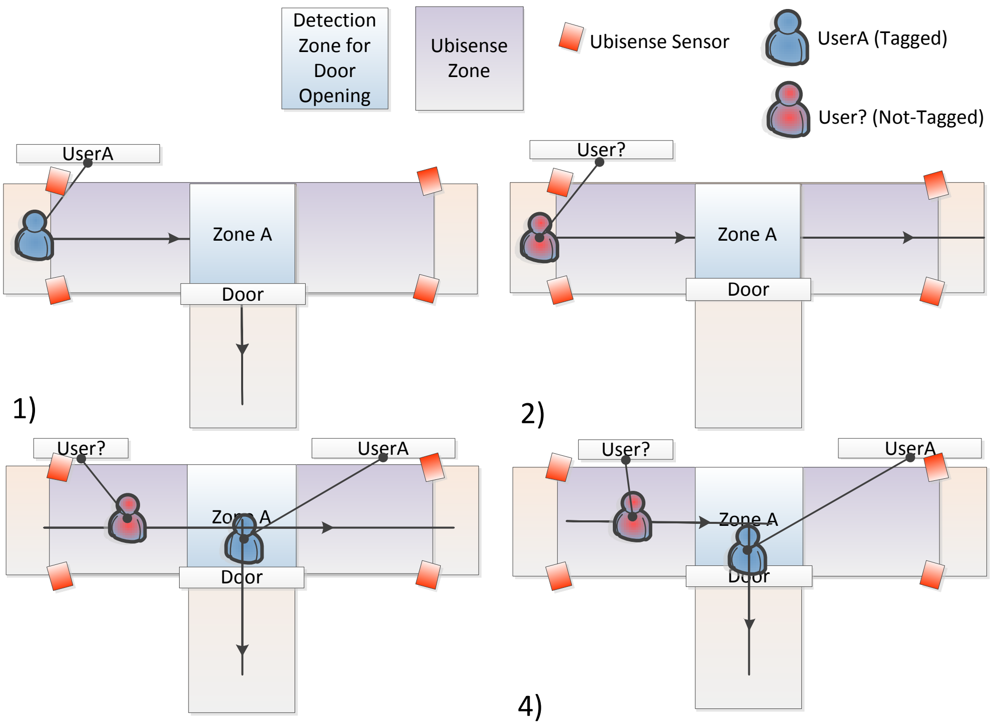


Figure 16-2 Experiment Situations

*The **Security Application** gives the following output when someone is detected in **zoneA**:

"**Authorised user in Zone**" and "**Authorised user not in Zone**"

When they are in the zone, the following message will be displayed "**Door Opening**"

When they have left the zone the following message is displayed "**Door Closing**"

It will then take 0.8 seconds for the door to fully close "**............Door Closed**"

The PIR sensors you will be placing each cover a 2x2 meter PIR zone. Both PIR sensors generate events when a user (tagged or not) walks into a PIR zone. The security system sets a PIR zone as **triggered** only when a non-tagged user is in the PIR zone and no tagged user is in the same zone. When a tagged user is in this zone and a PIR sensor event is generated, the PIR will **not** be **triggered**.

Once one PIR is **triggered** the door will not open even if **userA** is in **zoneA.** Both PIR sensors (**PIRA** and **PIRB**) must be placed on either side of the door; so that once the second PIR sensor is **triggered** it is assumed that **user?** has walked past the door. This results in both PIR sensors being reset to a "**not triggered**" state. This logic is captured as a finite state machine in Figure 16-3 (for the door and the PIR sensors).

It should be noted here that you will only need to choose between three positions for each PIR sensor and that neither PIR sensor should be placed in **zoneA** as it is assumed that the tagged user (**userA)** would be aware of the tailgating at this distance and would act accordingly. It should also be noted that for this experiment **userA** and **user?** will always come from the left and **will not reverse direction**. The user movement is covered in more detail in Task 1 description.

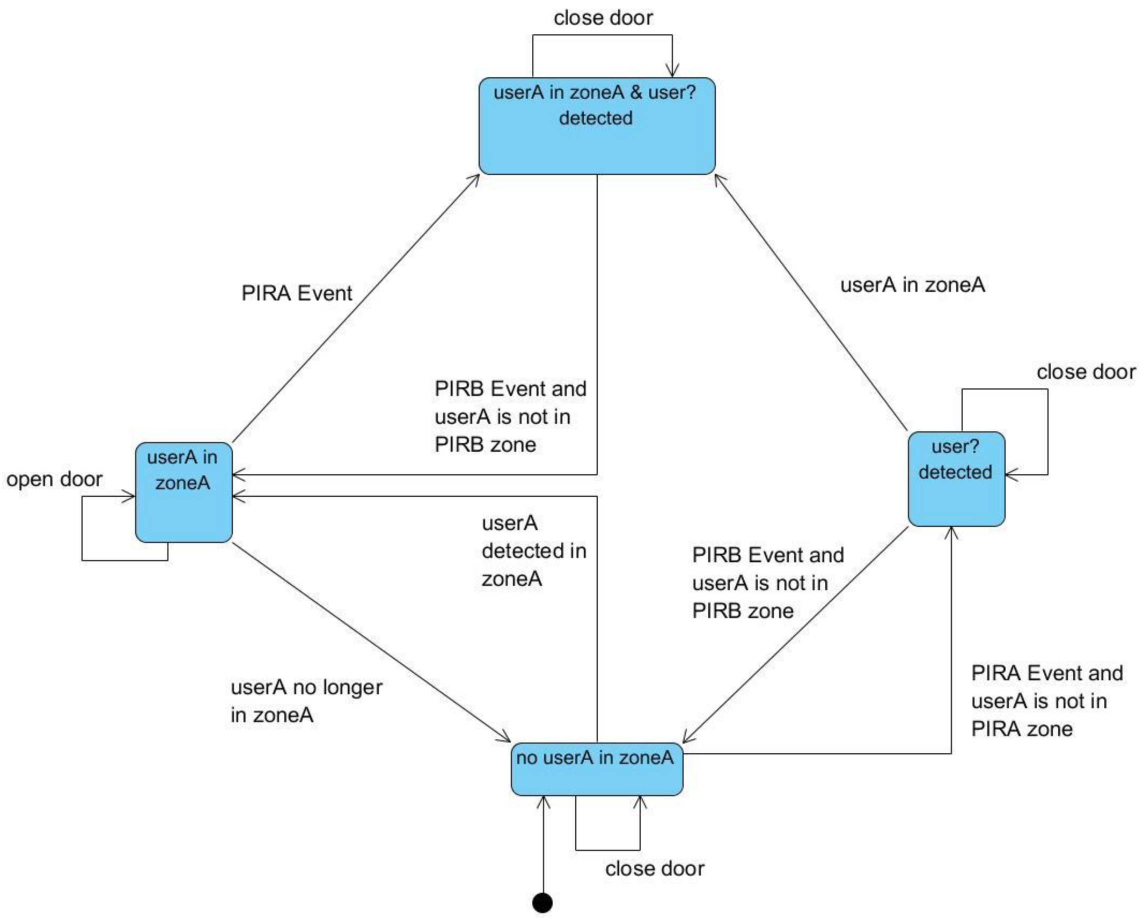


Figure 16-3 Automatic Door Finite State Machine

Table 16-1 Door States

Conditions			
(UserA inZoneA) & (BOTH (PIRA AND PIRB) NOT Triggered)	(UserA inZoneA) & (PIRA Triggered)	(UserA inZoneA) & (PIRB Triggered)	(UserA NOT inZoneA)

Open	Closed	Closed	Closed

You will also be required to evaluate the impact of uncertainty experienced by the Ubisense system. Uncertainty here describes the amount of noise or variation of the sensed position of the tag (carried by **userA**) and the actual location of the tag. You will model additional noise through the use of a Probability Density Function (PDF) which captures the offset in the x, y and z coordinate (from the actual x, y, z of the tag) using a Normal (Gaussian) distribution (Figure 16-4). A PDF modelled as 0.5 meters at one standard deviation will therefore result in an offset 0.5 meters 68% of the time, 1 meter 95% of the time and 1.5 meters 99.7% of the time.

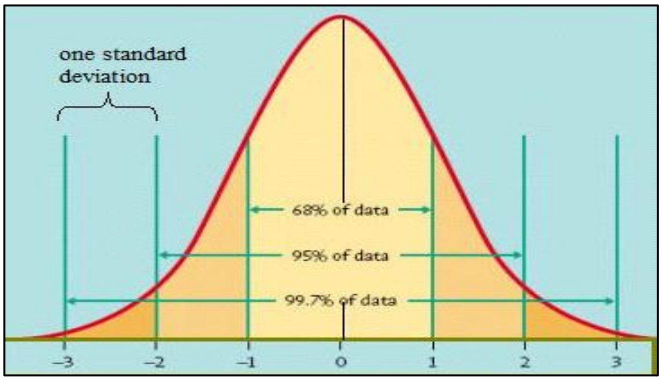


Figure 16-4 A Normal PDF

You have **two main objectives** when placing the two simulated PIR sensors both before and after uncertainty has been added to the Ubisense system. The first and most important objective is:

- **To avoid incorrect identification of an authorised user** (resulting in the door not opening for the authorised user when there is no risk of a non-authorised user gaining access)

The second and of less priority:

- **To minimise the time taken for the door to be unable to open due to a PIR being triggered** (and thus avoid the inconvenience for tagged users having to wait for non-tagged users to trigger the second PIR and allow the door to open)

16.2.2 Tasks

16.2.2.1 Task 1 Familiarise yourself with the SimConfig Tool

In this task you will familiarise yourself with the SimConfig tool which is on the screen in front of you. Below are four diagrams. Each will explain certain aspects of the SimConfig tool. On the left of the tool is where the configuration takes place. There are four tabs which you will be concerned with in these tasks. For **Task 3** Figure 16-5 and Figure 16-6 will suffice. For **Task 5** Figure 16-7 is required. On the right of the SimConfig tool is the SimConViz tool. This supports the visualisation of the simulated context source (e.g. the simulated PIR sensor zones) zones to help with their placement. It displays the building on a floor by floor basis. Here the building being visualised is the NIMBUS centre in Cork.

16.2.2.1.1 Source Info

Figure 16-5 shows the SimCon (Simulated Context) Source Info tab. This is where new generic SimCon sources can be created from the drop down menu. Type defines the SimCon source type (e.g. Passive Infrared). G.U.I.D indicates the SimCon sources unique i.d. Context Type indicates the type of context the SimCon source generates. The Passive Infrared generates "Presence Events". Once a SimCon source is chosen, you must click "Add" to add the current selection to the SimCon model. It will then appear in xml below.

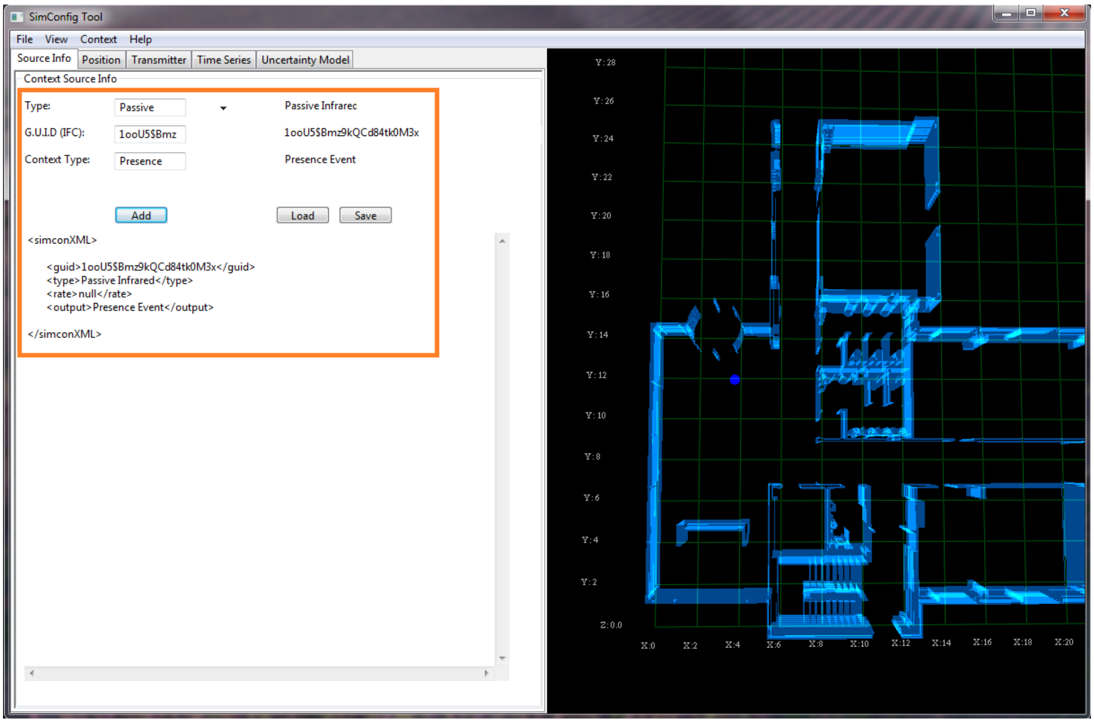


Figure 16-5 SimCon Model: (Type, GUID and Context)

16.2.2.1.2 Position

Figure 16-6 shows the position tab. This is where the coordinates of a SimCon source zone are to be configured. Anything within this zone can have an impact on the SimCon source. A zone here is defined as two coordinates which define the size of the zone. The right screen gives the coordinates of the building in meters and these can be used along with the grid system to place the SimCon source. For the PIR sensors you need only concern yourself with the origin of the sensor as the zone of the PIR is fixed at 2 meters squared (this will be calculated automatically).

Once coordinates are defined here they must be added to the SimCon model by clicking the "Add" button. It should be noted that the position ID must be the same as that of the SimCon source id. Once this has been done you will see the position data in xml by clicking on the "SimCon Info" tab .

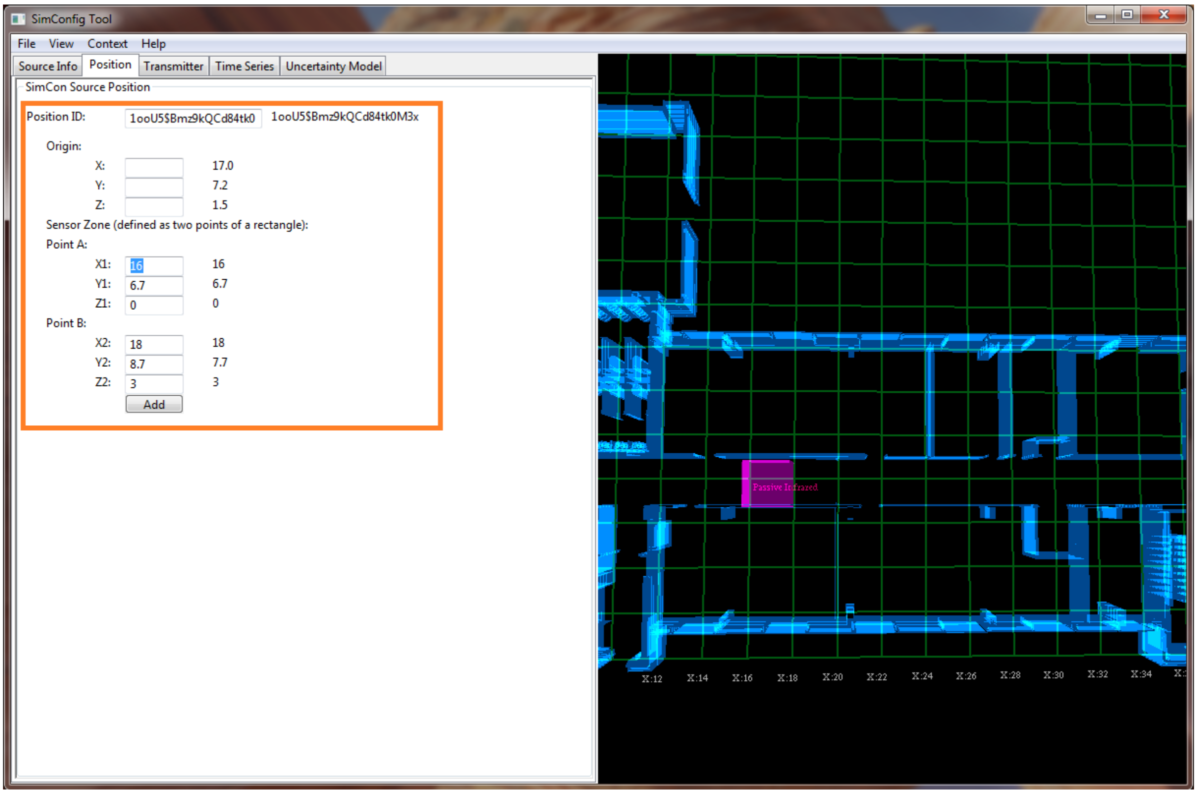


Figure 16-6 SimCon Model: (Position ID, Coordinate)

Figure 16-7 shows how to define uncertainty for a SimCon source (e.g. Ubisense) using a Probability Density Function (PDF). Here you can add uncertainty directly to the output of a SimCon source (for instance defining the variance for all tags within a Ubisense cell) or for an area (defining the variance for tags within a specific area of a Ubisense cell) or for the rate (introducing additional pseudo-random delays into the data generated by a SimCon source). For Task 5 you will be interested in defining two Uncertainty Zones for the Ubisense cell at both ends. This will be explained in greater detail in Task 4 and 5.

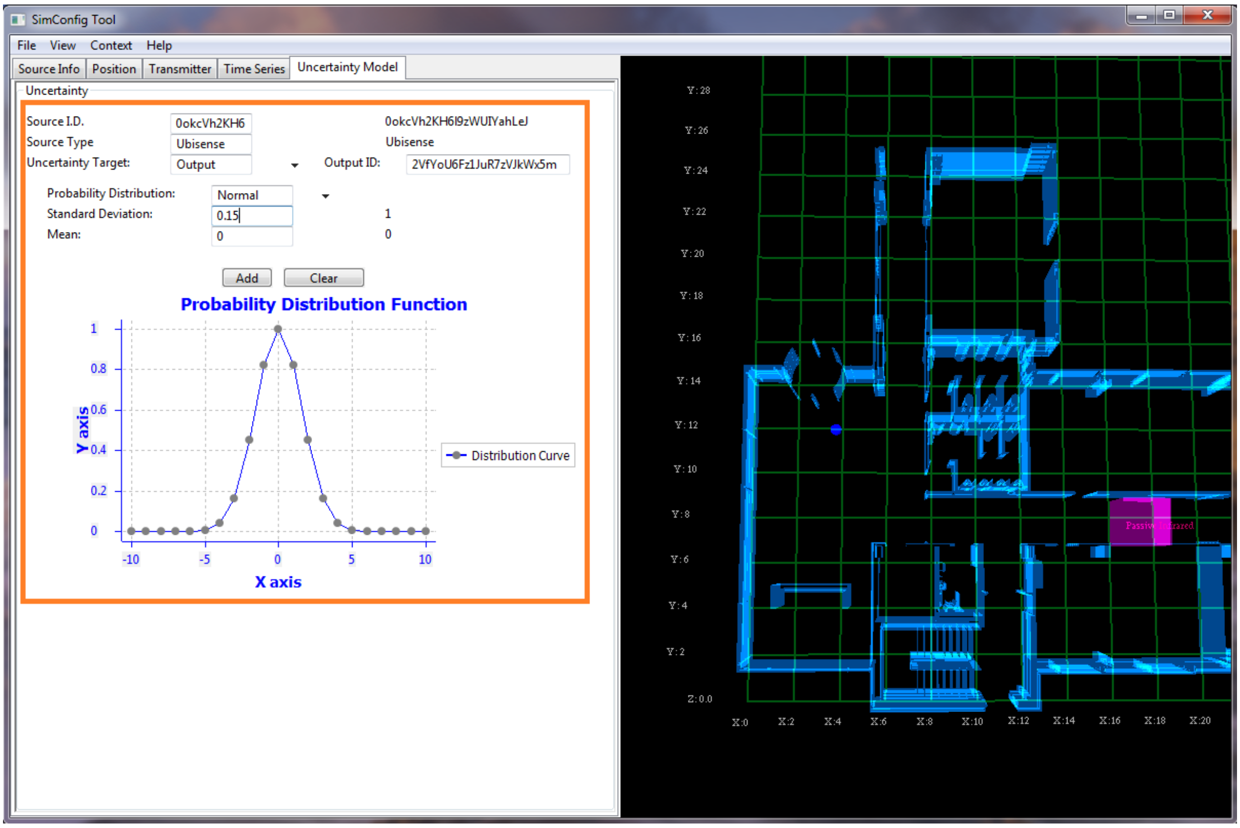


Figure 16-7 SimCon Model: (Source ID and Type, Target, Uncertainty ID, Distribution, Standard Deviation))

16.2.2.2 Task 2 Place Two PIR SimCon sources.

In this task you will create and place two PIR SimCon sources where you believe they will best achieve the objectives outlined above and reiterated now.

The first and most important objective is:

- **To avoid incorrect identification of an authorised user** (resulting in the door not opening for the authorised user when there is no risk of a non-authorised user gaining access)

The second and of less priority:

- **To minimise the time taken for the door to be unable to open due to a PIR being triggered** (and thus avoid the inconvenience for tagged users having to wait for non-tagged users to trigger the second PIR and allow the door to open)

Go back to the SimConfig tool and create a PIR sensor.

- Click on the "Source Info" tab. Select Passive Infrared in the drop down menu. Change the I.D. to "**PIRA**". Add it to the SimCon model using the Add button.
- Now click on the Position tab. Here you can set the position of the PIR sensor. Set the Z position as zero and the Y position as 7.75. These will always be this value as you need only concern yourself with the X position. Set the X position for the PIRA as one of the following three values: 15, 17 or 19. These three values specify the three positions you can place the PIRA Sensor. You need not concern yourself with the Sensor Zone coordinates as these are calculated automatically.
- Click "Add". Now click on the "Source Info" tab and press "Save". If you cannot see it, try clicking on the drop down menu at the top of the screen "View" and select "Context Source" -> "Zones" and Click "Ubisense" to turn off the Ubisense Zone visualisation.

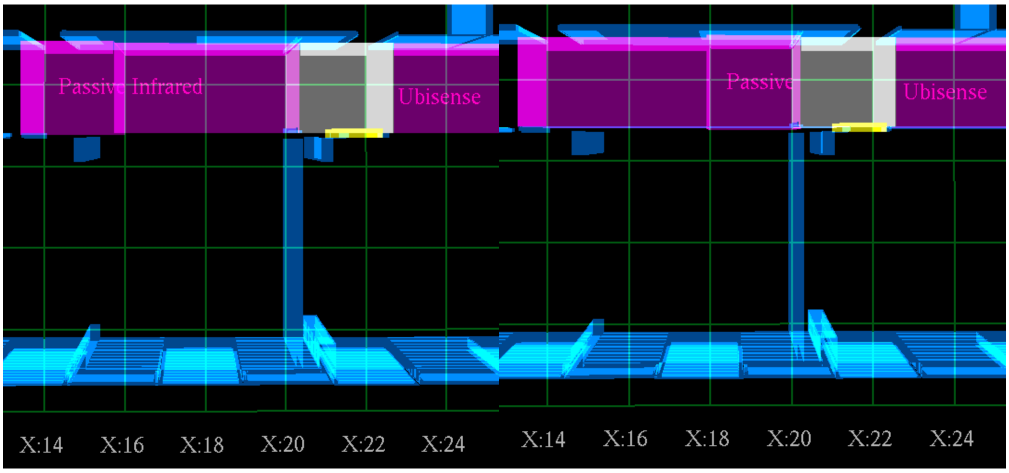


Figure 16-8 Left: PIRA at X:15, Right: PIRA at X:19

- Repeat for the second PIR SimCon source. This time, give it a unique I.D. “**PIRB**”. Once again set the position through the Position tab. Set the Z position as zero and the Y position as 7.75. These will always be this value as you need only concern yourself with the X position. Set the X position for the PIRB as one of the following three values: 23, 25 or 27. Make sure you click “Save” if you wish to save your changes. Once you have placed the PIR simulated sensors, proceed to Task 3.

16.2.2.3 *Task 3 Evaluate the SimCon source Outputs with respect to the requirements of the SBA.*

In this Task you will evaluate the outputs from your placed PIR sensors while also examining the state changes of the Security Application. You must **play** each scenario **4 times** (situation1.bat – situation4.bat)**.** You must **evaluate** your SimCon placement in this time. After this time you may re-run each scenario as many times as you please. **First** start the **Security Application** securityApp.bat in the bottom right of the screen (ask the Instructor if you require assistance).

You can examine the outputs from the SimCon PIR sensors, Ubisense and the Security Application* to evaluate your placement. You can examine the text output or the **visual output** on the screen.

Evaluate the placement with respect to the objectives of the SBA. The first and most important objective is:

- **To avoid incorrect identification of the user** (resulting in the door falsely locking)

The second and of less priority:

- **To minimise the time taken for the locked door to unlock** (and thus allow the door to open)

Once you are satisfied with your placement inform the Coordinator who will record the placement. Then, continue on to Task 5.

IF THE AUTHORISED USER FALSELY TRIGGERS A PIR SENSOR AND THE DOOR BECOMES PERMANENTLY LOCKED (i.e. the authorised user cannot move) YOU MUST RESTART THE SECURITY APPLICATION.

*When a PIR is triggered, the door will no longer open (causing any user who wishes to enter to have to wait). The Security Application gives the following outputs when the PIR sensors are triggered:

PIRA Triggered, door will now close and not re-open until second PIR triggered
----
PIRB Triggered, door can now open.

16.2.2.4 Task 4 Configure Ubisense to include uncertainty as defined in the Ubisense specifications

Ubisense specifications give Ubisense an accuracy of 0.15 meters. That is, there can be an offset in the sensed location of up to 0.15 meters from the actual location of the tag. In this Task you will introduce this uncertainty into the Ubisense cell using the SimConfig interface. Now do the following:

- Select the SimConfig tool. Press the "Load" button and load "UbisenseUbi1.xml". You can now see the simconXML description of this sensor cell at the bottom left of the screen. Here you see the info, position, transmitter data regarding the SimCon source.
- Click on the "Uncertainty Model" tab at the top of the SimConfig tool. The Source I.D. and type data is already provided. This data links this uncertainty model to the Ubi1 cell.
- Under "Uncertainty Target" select "Output".
- Under "Probability Distribution" select "Normal" and put "0.05" (meters) as the Standard Deviation. This means at one standard deviation there will be an offset of up to 0.025 meters 68% of the time, 0.05 meters 95% of the time and 0.1 meters 99.7% of the time. **Click "Add".**
- Under Source Info the new Uncertainty model should now be at the bottom of the simconXML. **Now click on "Save"** and save the file as UbisenseUbi1.xml.

Now repeat Task 2 and 3.

16.2.2.5 Task 5 Introduce additional uncertainty into the Ubisense system.

Reflective surfaces (glass walls) on the left of the door in the corridor are a likely to cause multipath in the Ubisense system which will result in increased uncertainty in the Ubisense positioning. Therefore you wish to evaluate the impact of additional uncertainty in this area. In this Task you will introduce an additional uncertainty zone into the Ubisense system at that location using the SimConfig interface. Now do the following:

- Select the SimConfig tool. Press the "Load" button and load "UbisenseUbi1.xml". You can now see the simconXML description of this sensor cell at the bottom left of the screen. Here you see the info, position, transmitter and uncertainty data regarding the SimCon source.
- Click on the "Uncertainty Model" tab at the top of the SimConfig tool. The Source I.D. and type data is already provided. This data links this uncertainty model to the Ubi1 cell.
- Under "Uncertainty Target" select "Zone". A Unique I.D. will be created. This I.D. will link the Uncertainty Zone to the "Position" data describing the placement of the Zone.
- Under "Probability Distribution" select "Normal" and put "0.3" (meters) as the Standard Deviation. Leave the mean value as "0". **Click "Add".**
- Under Source Info the new Uncertainty model should now be at the bottom of the simconXML.
- You must now define a position for the Uncertainty Zone. Go to the "Position" tab. Here you will see the position I.D. which corresponds to the Uncertainty "Zone I.D."
- Enter the following coordinates into the Uncertainty Zone: – X1:18, Y1:6.8, Z1:0, X2:20, Y2:8.7, Z2:3.
- **Click "Add".** This Uncertainty Zone will now appear as a position in the simconXML under the Info tab. You will notice that both I.D.'s are the same. **Now click on "Save"** and save the file as UbisenseUbi1.xml.

You can now visualise the "Uncertainty Zone" by de-selecting the visualisation of the Ubisense cell zone. This can be done by selecting "View" -> "Context Source" -> "Zones" and clicking on the items you wish not to be displayed (i.e. "Ubisense" and "Passive Infrared"). The Uncertainty Zone is displayed in green.

Now repeat Task 2 and 3.

17 Appendix I: Detailed Description of Experiment Tasks

17.1 Evaluation 1A Tasks

The usability test began by briefing the participant. Tasks were presented in a counter balanced order to prevent ordering effects.

1. Task 1 required the participants to use the SimConfig prototype to create a SimCon source to transform location information generated in the VR building into a contum using either a sequence diagram approach or an activity diagram approach.

2. Task 2 required them to repeat this process using the alternative approach. (Half of participants started with the activity approach and half with the sequence, so as to remove any bias that might be introduced by the order in which they are taken).

In both cases the participant began by configuring the source of the incoming message. They then configure the xml format of the outgoing message. The tool also allows the configuration of uncertainty to be associated with a particular piece of context.