User Requirement Capture in Construction IT: The Case of the DIVERCITY Project

Marjan Sarshar[1], BSc (Hons), MSc, PhD, CEng, MIEE, FRSA
Yusuf Arayici[2], BSc (Hons), MSc

Abstract

There is a gap between construction IT (CIT) research and its uptake within the construction industry. One way to reduce this gap is better requirements capture and testing during CIT research.

DIVERCITY was a European consortium of researchers and practitioners from the construction industry who attempted to develop a virtual environment that enables the industry to better undertake the client briefing and design reviews. The project had the acronym DIVERCITY – 'Distributed Virtual Workspace for Enhancing Communication within the Construction Industry.'

This paper explores the user requirements capture of the DIVERCITY project. DIVERCITY has large and evolving requirements, which must consider the perspectives of multiple stakeholders, such as clients, architects and contractors. However, virtual environments such as DIVERCITY are not currently in use in the construction sector. Practitioners are often unsure of the detail of how virtual environments would support the construction process, and how to overcome some of the barriers to the introduction of new technologies. This complicates the requirements capture process.

The paper starts by establishing the importance of requirements capture in the CIT community, as well as explaining some of the current methods and trends in requirements capture. This is followed by an explanation of DIVERCITY's requirements capture process and exploration of some of its strengths & weaknesses. Finally some discussion and lessons are offered on how to improve the requirements capture & testing process in the CIT research community.

1. Background

For many years leading researchers and industrialists have attempted to utilise IT as an enabling technology, in order to reduce the problems of communications and information sharing within the construction industry. More recently, within the area of construction IT, researchers have identified the need for an integrated virtual construction environment (/workspace), which acts as a project repository, during all stages of the life-cycle (Grassi 99, Alshawi 96, Kiviniemi 99, Aish 99, Aouad 07, Sawhney 99). This aims to improve the communication between the different stakeholders and improve productivity. This environment has proved complex to develop and implement. DIVERCITY was an EU funded research project (1999-2001), which attempted to develop innovative workspace technologies for the briefing and design phases of the life cycle. The workspace was designed to be extendable to other phases of the construction life cycle, in an aim to develop a holistic virtual construction environment.

In its early days the CIT research community mainly focused on technology based issues, trying to implement lessons learnt from the IT community in the construction industry. Early prototypes in many research projects investigated data standards and common information models through which heterogeneous computer systems could exchange project information. Many different technologies have been explored. However, in the early projects there was little attention and consideration to the user requirements capture and subsequent implementation of the prototypes (Froese 1995, Tanyer 2003). As a consequence there has been little evidence of industrial uptake of CIT research.

---

[1] M.Sarshar is an Associate Dean at the Faculty of Built Environment, University of West of England, Frenchay Campus, Coldharbour Lane, Bristol BS16 1QY, UK.
[2] Y.Araici is a researcher at the School of Construction Management, University of Salford, The Crescent, Salford, M5 4WT, UK.

Increasingly the researchers are directing more attention to capturing end-user requirements, in an attempt to increase the level of technology uptake. DIVERCITY was a project which allocated significant resource and attention to the requirements capture process. This paper explores some of the lessons learnt in this area.

2. The DIVERCITY Project

DIVERCITY is an abbreviation for a Distributed Virtual Workspace for Enhancing Communication within the Construction Industry (DIVERCITY Handbook 2003). It was funded by the EU commission in order to:

- Create a client-briefing workspace that allows interaction and communication of design ideas between the client and the architect;
- Create an interactive design review workspace which allows multi-disciplinary design reviews involving different stakeholders of a construction project;
- Create a virtual construction workspace that allows the user to assess the constructability of a building, and plan and layout of the construction site;
- Develop a software framework for integrating the above three workspaces and sharing them over networks to support collaboration between geographically distributed project team members;
- Future evolution of the virtual environment to encompass other phases of the construction life cycle, such as facilities management.

These were delivered through a workspace, which composed of six software applications, namely: (i) client briefing; (ii) thermal simulation; (iii) acoustics simulation; (iv) lighting simulation; (v) 4D scheduling; and (vi) site planning. Three further applications, which are transparent to end users support collaboration activities and provide mathematical algorithms for the simulation calculations. The functionality of this workspace is explained in detail in the DIVERCITY Handbook (2003).

This paper focuses on DIVERCITY's approach to capturing the construction industry's requirement. The construction industry is a large multi-disciplinary industry, with multi-faceted perspectives and requirements. DIVERCITY needed to define broad industrial requirements, and expand them into more detail to capture the briefing and design requirements of the industry, and more specifically the requirements of the DIVERCITY specific applications.

3. Requirement's Capture

DIVERCITY is a large-scale, highly innovative and interactive workspace. It is concerned with the development of interactive systems that cannot be treated simply as incremental improvements over existing construction IT solutions on the market. In such cases it is not possible to identify user requirements on the basis of empirical techniques, as there are no instances of the use of the product (or similar products) from which to collect data. Consequently, the developers of innovative products must proceed by envisioning the use of the proposed product (Dearden 1998).

This section explores the need for user centred requirements capture in IT research. It provides some definitions and briefly explores some of the existing approaches, which are used in the software engineering discipline.

Requirements as defined by Dorfman (1997) is a software capability that must be met or possessed by a system or system component to satisfy a contract, standard, specification or other formally imposed document which is needed by the user to solve a problem to achieve an objective.

Pfleeger (1998) describe requirements as a feature of the system or a description of something the system is capable of doing in order to fulfil the system purpose. The purposes of requirements capture are (Pfleeger 1998):

1. To understand how the customer wants the system to work
2. To know what functionality and characteristics the resultant system is to have
3. It enables the test team to know what to demonstrate to convince the customer that the system being delivered is indeed what was ordered.

Depending on the characteristics of the system under development, there are a variety of techniques and methodologies to capture requirements in a user-centred manner. Figure 1 below shows some common methods undertaken in discrete stages of the development lifecycle for user centred design (Carbon IQ, 2001). The bottom part of the diagram shows the phases of the software development process. The top part shows the focus of requirements capture, plus some of the methods to support the activities.

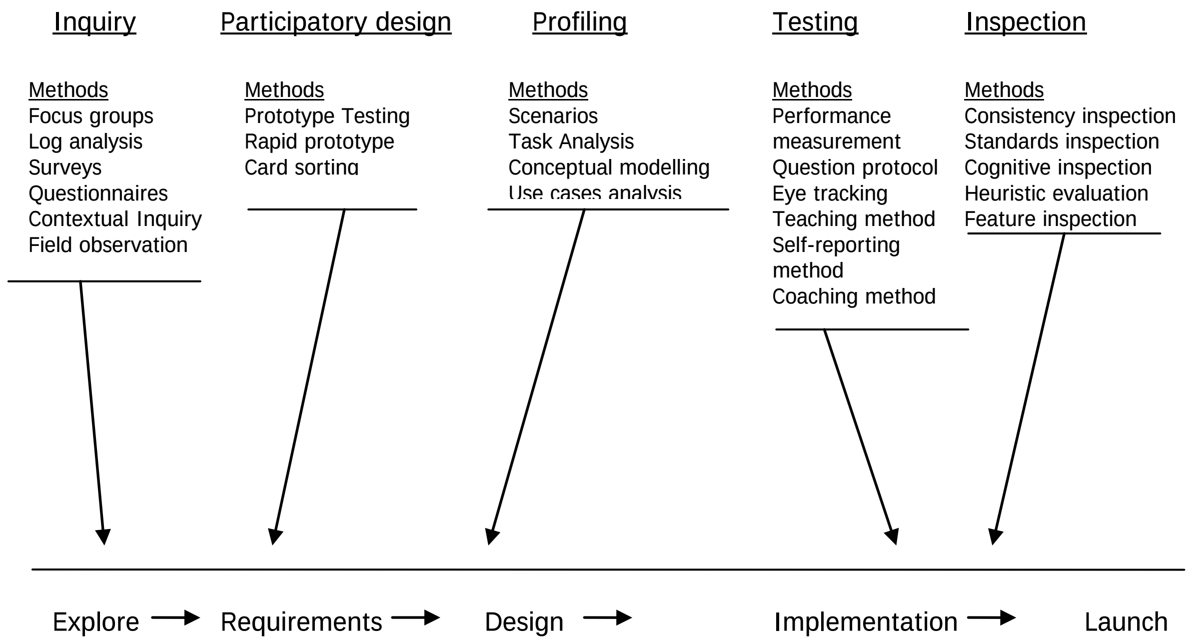

Figure 1:Mapping the common requirements methods on the software development lifecycle (Carbon IQ, 2001)

The figure demonstrates that requirements capture is a continuous process, throughout the software development process. At the start of the process during initial “inquiry” and “participative design”, the requirements elicitation and capture activities ensure that the customer requirements and system’s functionality are understood. Towards the end of the process “testing” and “inspections” convince the users that the system has delivered what was expected. Furthermore the “testing” and “inspections” assist in enriching the initial user requirements.

In innovative systems, such as DIVERCITY end-users are often unsure of what they want and what to expect, at the start of the project. The testing and inspections improve end-user understanding by demonstrating what is possible. The users can then provide more detailed requirements and explore how the systems would fit into their business processes.

4 The Process of Requirements Capture

The process of requirements capture must be planned and managed. There are a number of generic activities to all requirements engineering processes (Sommerville, 2000):

- requirements elicitation
- requirements analysis
- requirements validation
- requirements management

Elicitation is a definition of the system in terms the end user can understand. Analysis is a technical specification of the system in terms the developers can understand. Validation is concerned with showing that the requirements define the system that the users want.

Requirements management is the process of managing changing requirements during the requirements engineering process and system development (Lundh 2002).

The methods listed in figure 1 each have their strengths and weaknesses during the requirements capture process. For example use cases are effective during requirements elicitation and analysis but they are inadequate for capturing holistic requirements of the systems such as non-functional requirements (Regnel 1995). Therefore, researchers must synthesise use cases with additional methods through placing these techniques before or after use case development and analysis.

In a typical systems development project, such as DIVERCITY, the requirements capture process will include a combination of methods, milestones and management techniques. The type of system, and the prior knowledge of the development team influence the detailed process.

4.1 DIVERCITY's Process

The DIVERCITY's requirements capture team consisted of five organisations spread across four EU countries. The team comprised of two universities with construction IT background, a large firm of architects, a medium sized contractor, and a large engineering consultancy firm.

The importance of requirements capture was well understood amongst the project team. However, the processes and methods for requirements capture evolved through the project. Initially team members had individual ideas and perceptions. Different methods were explored concurrently, but independently of each other. It took the team over a year to evolve a shared understanding of the processes and methodologies. There was then a learning period, for all team members to become familiar with all the deployed methods.

This period of evolution was due to three main reason: (i) The background and prior skill base of the team members was different; (ii) The team was geographically disperse; and (iii) The novel nature of the DIVERCITY workspace meant that the team initially did not have a shared understanding of what the workspace will look and feel like.

A key lesson for the construction IT research community is that the requirements capture process is as important as the requirements capture methodology. It is important for the team to agree on the process and the methodology at the start of the research. Training should also be provided, so that team members start from the same basic knowledge and assumptions. This requires resources and facilitation from a requirements capture expert, at the start of any research.

5 DIVERCITY's Methodology

DIVERCITY adopted a requirements capture framework based on the work of Beyer and Holtzblatt, (1998). The framework is depicted in the below figure 2. It has five main stages. These are (i) vision; (ii) storyboard, (iii) user environment design, (iv) use case modelling and (v) object models. The following sections review the definition of each of these stages and provide examples of the DIVERCITY deliverables for each stage.

5.1 Vision

A vision sketches new work without showing how it will happen over time (http://www.incent.com/cd/cdp.html). Katsunosuke Maeda, President of Toray Industries , explains that: "Vision is…a wide and distant vista of the future in which the desirable image of the company is set, directing its course and generating strategies there" (Okazaki-Ward 1998). Fransman (1999) states that vision refers to beliefs about what the world is like, how it changes and what it will be like with and without the believer's interventions.

DIVERCITY adopted an existing vision for construction IT, which was formulated by Sarshar (2000, 2002). This vision was developed as a result of a review of 70 academic papers, two academic workshops and one industrial workshop. The underlying theme is the use of collaborative and integrated environments, through the use of advanced IT. It contained seven key themes, which were taken forward by DIVERCITY, i.e.

1. Model driven as opposed to document driven information management on projects (use of IFC standards).
2. Life cycle thinking and seamless transition of information and process between life cycle phases.
3. Use of past project knowledge (/ information) on new developments.
4. Dramatic changes in procurement philosophies, as a result of the internet.
5. Improved communications at all lifecycle phases, through visualisation.
6. Increased opportunities for simulation and what if analysis.
7. Increased capabilities for change management and process improvement.

The meaning and value of these themes was investigated in the scenario of a typical construction project life cycle (Sarshar 2000).

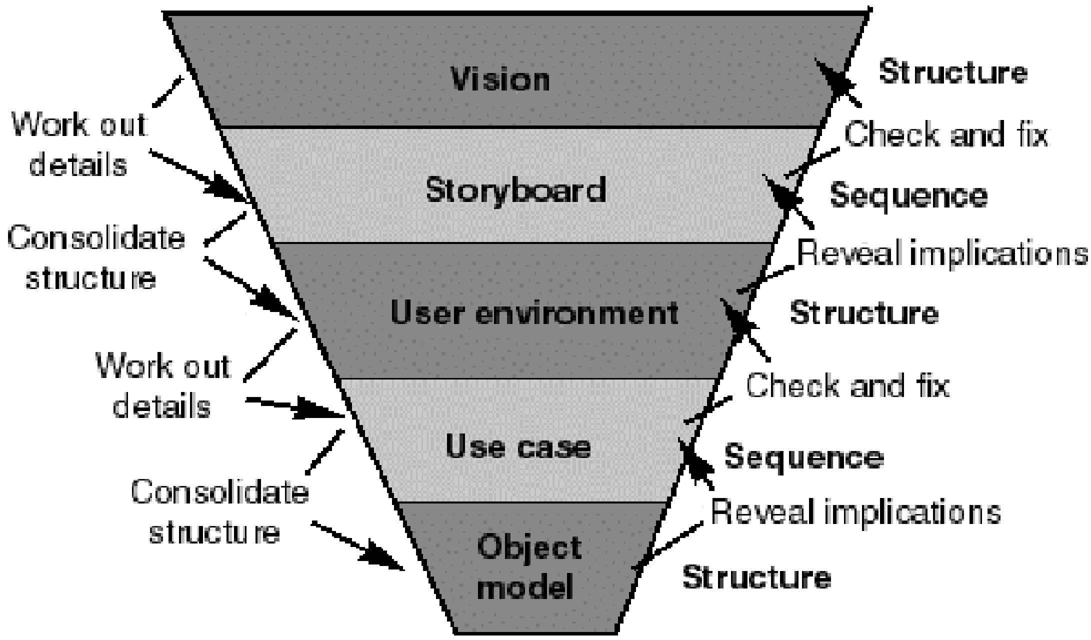

Figure 2: A framework for capturing user requirements (*Beyer and Holtzblatt, 1998)*

5.2 Storyboard

Once a vision definition is completed, the requirements capture team develops the details of the vision in storyboards. This produces 'freeze-frame' sketches, which capture scenarios of how people will work with the new system. Storyboards show how specific tasks will be accomplished in the new world. A storyboard captures the new procedure for doing a task pictorially. Each frame in the storyboard captures a single scene. Storyboards are based on the vision, follow the structure of a consolidated sequence model, and pull implications from other models as necessary (Beyer and Holtzblatt, 1998).

DIVERCITY developed a construction storyboard based on the vision. This storyboard was split into fifteen scenes, each defining how DIVERCITY would interact with a specific construction process. An example of one scene is given below.

Scene 1- Architect collects all data about restrictions to the site
Zone plan: areas reserved for building, parking, traffic etc. maximum and minimum heights, materials, permitted building volume etc. –Local building regulations and standards –Site constraints: geometry, slopes, ground, neighbours, scenery, compass points, climate etc. –Usually very unstructured data! (Text, CAD, sketches, 3d data, hand-drawn pictures)

The scenes where later expanded into use cases, explaining how each stakeholder would interact with the system and what the functionality of the system should be.

5.3 User Environment

The new system must have the appropriate function and structure to support a natural workflow. The User Environment Design captures the floor plan of the new system. It shows each part of the system, how it supports the user's work, exactly what function is available in that part, and how the user gets to and from other parts of the system--without tying this structure to any particular user interface (Beyer and Holtzblatt, 1998).

There are five different types of *Work Models* used in User Environment Design:

- *Flow*, representing communication and co-ordination necessary to do the work (roles, responsibilities, actions/communication topics, and spaces which in DIVERCITY are regarded as project internal or project external memories and virtual/physical spaces).
- *Sequence*, showing the detailed work steps necessary to achieve intent. Sequence models can reveal alternate strategies to achieve the same intent.
- *Artefact*, showing objects created to support the work.
- *Culture,* representing constraints on the work caused by policy, culture or values, formal and informal policy of the organisation, business climate, self-image, feelings and fears of the people in the organization, possibility for privacy.
- *Physical,* showing the physical structure of the work environment as it affects the work.

DIVERCITY focused on flow, sequence and artefact models. The flow and sequence models are combined with the artefact models and synthesised to storyboards. Figure 3 shows a storyboard combining the sequence model and artefact model. Christianssen (2001) provides more detail of DIVERCITY' User Environment diagrams.

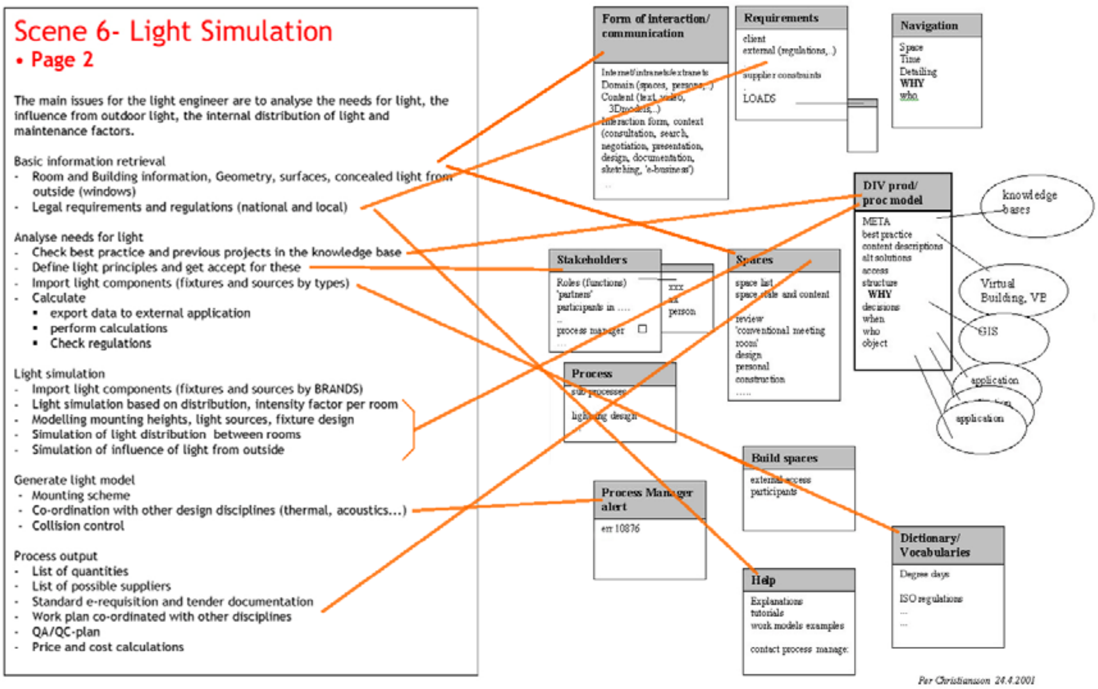

Figure 3: Part of storyboard for lighting design. (Christiansson et. al., 2001).

5.4 Use Cases

Use cases were first developed by Jacobson 1995). A use case describes how the designed software will be used. They describe coordinated activities in which a user accomplishes some purpose in the applications domain. Use cases offer a systematic and intuitive way to capture the functional requirements with particular focus on the value added to each individual user or to each external system. By using use cases, analysts are forced to think in terms of who the users are and what business or mission need can be filled through them. Use cases can help with requirements capture by providing a structured way to go about it:

1) identify the actors
2) for each actor, find out

- What they need from the system: that is, how can the use case add value to their activities
- any other interactions they expect to have with the system, that is, which use cases they might take part in for someone else's benefit (Stevens 2000).

Thus, a use case is precisely a functional test case of an important sequence of operations, but its purpose is to illuminate and sharpen requirements rather than to test the software.

DIVERCITY evolved each scene of the storyboard into more specific use cases. Each use case explores how each actor will interact with the system. An example is given in the table below:

Use case for Scene 6- Light Simulation **Actor: The Lighting Engineer**
The main issues for the light engineer are to analyse the needs for light, the influence from outdoor light, the internal distribution of light and maintenance factors. The light engineer will demand access to list of quantities and technical specifications as result of the simulation. The light engineer is one member of the project team and the work must be co-ordinated with other stakeholders **Basic information retrieval** Get permission to access the model for changing Open the model and evaluate existing conditions - Room and Building information, - Geometry, surfaces, - Required light demands - … **Analyse needs for light** - Check best practice and previous projects in the knowledge base (future) - Define light principles and get accept for these from other relevant stakeholders (architect, HVAC, …) These stakeholders are prompted in the communication module (the process manager) Simulation - Import light components (fixtures and sources by types) … **Light simulation** - Import one room in the Phooka application - Import light components in the model - Distribute the components in the room and measure the intensity factor per room - Modelling mounting heights, light sources, fixture design, concealed light from outside - … **Process output** - Technical calculations, - Mounting scheme - Documentation for saved solutions - …

<u>5.4 Object Modelling</u>

Object modelling is used to translate user requirements into software engineering analysis and design requirements. DIVERCITY is fully IFC compliant. The only object models were the objects used in defining the User Environment, and the IFC object models .

6. Incremental Prototyping with User Tests

Contemporary systems development approaches recognise that for almost all systems it is right and necessary to have some kind of iterative process. Modern development processes take iteration as fundamental and try to provide ways of managing, rather than ignoring, the risks (Stevens and Pooley, 2000).

In DIVERCITY continual validation testing was used as part of its incremental approach to software development. That is to say, testing was not comprehended as a single activity at the end of the project. Nor was testing a phase in the project during which the quality is assessed (Kruchten, 2000). Rather, the technical partners obtained continuous feedback on the evolving system functionality and quality. On the other hand the requirement's capture team continuously evolved their understanding of what the technology could offer, and what the shape and form of the DIVERCITY workspace would be.

The user group tested the broad functionality of early DIVERCITY prototypes as well as the stability, coverage and performance of the software architecture while there was still opportunity and sufficient time to fix it. Furthermore the user team tested and validated the final product to assess its readiness for delivery to end-users, which is called as acceptance testing in the software testing literature.

The DIVERCITY research team undertook three iterative tests during the project. During each iteration of the testing the functional requirements of the applications were expanded and their usability was improved. The tests were performed by all the requirements capture team and the results and experiences were shared in collaborative session. This was a key enabler, which allowed the team to develop a shared understanding of the requirements and the requirements capture processes.

7. Discussion

During the Introduction of this paper it was argued that a key objective of requirement capture in the construction IT research is to increase the level of technology uptake and reduce the gap between research and practice.

DIVERCITY expended much effort in capturing requirements directly from the end user community, who formed the requirements capture team. Many methodologies from the software sector were reviewed and eventually a methodology was adopted and shared among all the end users.

As such DIVERCITY is viewed as a successful EU research project. Its applications are technically advanced and beneficial to construction practitioners.

However, the construction industry in Europe operates on tight operating margins and is slow in adopting new technologies and working methods. The industry is fragmented, with many business interests and parties operating during each stage of the life cycle. Any innovative application needs to take account of requirements at three different levels, i.e:

1 the individual practitioner;
2 the project; and
3 the organisation and business (for each stakeholder within the construction team).

When these requirements overlap and compliment each other it should be possible to introduce innovative solutions to the industry.

The DIVERCITY requirements capture has focused on practitioner requirements, as well as technical project requirements. However, the commercial project requirements, and business needs have been mainly ignored. It is possible that virtual environments such as DIVERCITY could increase project costs, without sufficient benefits, or introduce excessive change in the process and working relationships among stakeholders. A key question is can

DIVERCITY bridge the technology transfer gap? Can some applications within DIVERCITY find their way to the construction industry?

Tanyer (2003) argues that in bridging this gap, the research methodologies used in most construction IT research projects are incomplete. Generally the research stops after the production of a working prototype. It is necessary for all construction IT research to continue with testing on live construction projects. This will allow capturing project and business impact and requirements.

The business requirements, and therefore the software requirements, will vary according to the construction sector (e.g. housing, office, etc.). Therefore researchers must explore the business gains for each specific sector before generalising the results for the whole industry.

Within DIVERCITY it was paramount to explore requirements from the business and financial perspective, alongside the technical requirements. It was essential to have a financial vision, in parallel with the technical and process vision. The framework and methodologies from the software sector, which were reviewed, disregarded these dimensions. Construction IT research needs to tailor or develop requirements' capture methodologies which consider these critical dimensions.

Due to the rigorous capture of end-user requirements, at a practitioner level, some of the DIVERCITY applications have been used in some construction projects. For example, the lighting application has been used in a French museum project, and the collaborative applications have been embedded in software for other industries. However, to date, most DIVERCITY applications have not found their way into the commercial construction market place.

8. Summary

This paper introduced the DIVERCITY research project, which developed a virtual construction workspace for briefing and design. A key challenge for novel environments such as DIVERCITY is the lack of technology uptake by the industry.

The paper focused on the requirements' capture methods and processes, as a means to improve the technology transfer rate. Many early CIT research projects paid little attention to requirements capture. DIVERCITY is one of the few projects, which spent significant time and resource in capturing end user requirements. This paper discussed the methods and processes deployed by the DIVERCITY project. These may prove beneficial for future CIT research projects.

The paper then conducted a critical analysis of the DIVERCITY approach. DIVERCITY has resulted in a virtual workspace, which is functional and technologically advanced. It will improve the effectiveness of some practitioners in a project.

The strengths of Divercity's approach were in adopting an incremental prototyping development cycle, which included practitioners from different backgrounds. However, DIVERCITY's requirements capture process must expand to:

1. Take account of the business benefits and financial impact of this workspace on construction projects as well as individual firms;
2. Includ live piloting on real construction projects, in order to assess the unforeseen impact on businesses and working practices.

Though there has been some uptake of DIVERCITY deliverables within the industry, more could be achieved in bridging the technology transfer gap.

It is important for the CIT community to plan live piloting of research results, as well as the assessment of financial impact of deliverables on businesses, more actively in the research methodology.

9. References

Aish, R., "Migration from an Individual to an Enterprise Computing Model and its Implications for AEC Research", Berkeley-Stanford CE & M Workshop: Defining A Research Agenda, Stanford, 1999.

Alshawi, M. "SPACE: integrated environment", Internal Paper, University of Salford, July 1996.

Aouad, G., Marir F., Child T., Brandon P. and Kawooya A. "Construction Integrated Databases- Linking design, planning and estimating, The OSCON approach"; International Conference on the Rehabilitation and Development of Civil Engineering Infrastructures. American University of Beirut, pp 52-60, June 1997.

Beyer H, Holtzblatt K, 1998, "Contextual Design. Defining Customer-Centered Systems", Morgan Kaufmann Publishers, San Francisco.

Carbon IQ (sited in 2001), User centred design Methods,http://www.carboniq.com/log/archives/00000118.html

Christiansson P, Svidt K, Skjærbæk J O, Aaholm R, 2001, "User requirements modelling in design of collaborative virtual reality design systems". International Conference on Construction Information Technology. Mpumalanga, Soth Africa, 30 May - 1 June 2001. (pp. 40/1 - 40/12)

Dearden, A., Howard, S., 1998, Capturing User Requirements and Priorities for Innovative Interactive Systems, Australasian Computer Human Interaction Conference, November 29, December 12, 1998, Adelaide, S. Australia

Divercity Handbook (2003), "The Divercity Project: A Virtual Toolkit for Constructin Briefing, Design and Management", University of Salford, UK.

Dorfman M. & Thayer R.H. (1997), System and Software Requirements Engineering,(2nd edition), IEEE Computer Society Press

Fransman, M., (1999), "Visions of Innovation", 1st ed., Oxford University Press Inc., New York, 1999, pp57-66

Froese, T. & Paulson, Jr. B.C., 1994, "OPIS: An Object Model-Based Project Information System", Microcomputers in Civil Engineering, **9** (1), pp 13-28.

Grassi, M.D., Zorgno, A.M., "The Virtual Construction Enterprise", Berkeley-Stanford CE&M Workshop, Stanford 1999.

Jacobson, Object –oriented Software Engineering: A Use case Driven Approach, ACM Press, New York, 1992.

Kiviniemi, A., Laitinen, J., Lautanala, M., "Defining a Research Agenda for AEC", Berkeley-Stanford CE&M Workshop, Stanford 1999.

Kruchten, Philippe, "The Rational Unified Process – An Introduction", Second Edition, Addison-Wesley, 2000, ISBN 0-201-70710-1

Lundh, E., Sandberg, M., Time Constraint Requirements Engineering with Extreme Programming -An Experience Report- IEEE Joint International Requirements Engineering Conference, 2002, Essen Germany

Okazaki-Ward, L., (1998), "The Practical Side to Visioning: The Case of Toray Industries Inc." in Kakabadse, Andrew; Nortier, Frédéric and Abramovici, Nello-Bernard (eds.), Success in Sight, Visioning, 1st ed., International Thomson Business Press, London, 1998, pp185-214

Pfleeger Shari Lawrence, (1998), Software Engineering: Theory and Practice, Prentice Hall

Regnell, B., Kimbler, K., Wesslen, (1995) A., *Improving the Use Case Driven Approach to requirements Engineering,* Proceedings of Second IEEE International Symposium on requirements Engineering, York, UK, March 1995.

Sarshar, M. Betts, M.; Abbott, C.; Aouad, G., (2000) "A Vision for Construction IT 2005-2010", RICS (Royal Institute of Chartered Surveyors) Research Series, December 2000, pp1-42

Sarshar, M., Tanyer, A., Aouad, G., Underwood, J., (2002) "A Vision for Construction IT 2005-2010: Two Case Studies", Engineering, Construction & Architectural Management, Issue 2, April, 2002.

Sawhney, A., "Research and Development Plan for the AEC Industry", Berkeley-Stanford CE&M Workshop, Stanford 1999.

Sommerville, I., Sawyer,P. (1997), Requirements Engineering, England, John Wiley&Sons Ltd

Stevens, P. Pooley, R. (2000), Using UML Software Engineering With Objects and Components, Pearson Education Limited 2000,

Tanyer, A.M., 2003," Bridging the Gap Between Research and End-users in Computer Integrated Construction", PhD Thesis, University of Salford, UK.