Steve North

Virtual Reality Centre for the Built Environment

The Bartlett School of Architecture, Planning
and Construction Management

University College London

PhD Thesis

First Supervisor: **Prof. Graham M. Winch**
Second Supervisor: **Alan Penn**

"Clients' 3D visualisation of construction performance: from s-curves to data surfaces"

ABSTRACT

Problem definition:

Construction clients are currently without adequate software tools for simplifying the complexity of project performance.
Can 3D Information Visualisation provide construction clients with informative performance reports?

Contributions to knowledge:

A conceptual 3D framework for the interpretation of construction performance parameters
A practical example of implementing the 3D framework
A protocol for prototyping and evaluating the developed software

Construction project management traditionally uses two dimensional visualisation techniques to analyse project performance. These techniques are usually graphically represented as an 's-curve', so called because of the characteristic plot shape (generated by resource take-up in many projects). S-curves are often limited by the amount of information that can be simultaneously displayed.

It is proposed that a more comprehensive performance measurement system might represent project progress as a three-dimensional data surface. This research considers the use of 'desk-top' Virtual Reality as an alternative to traditional project performance measurement systems. A conceptual three-dimensional framework for the representation of non-physical construction industry data is outlined. This framework led to the development of Procession, a three-dimensional Information Visualisation software tool. Procession uses a three-dimensional data surface as an abstract representation of the described framework.

The target group for Procession was social housing project clients, or Registered Social Landlords (RSLs). In the UK, the catchall term RSL describes Housing Associations, Housing Cooperatives and Local Housing Companies. Field experiments were undertaken to compare the information quality and usability of Procession, with progress reporting methods currently used in the construction industry.

The most important finding was that RSL clients were better able to assess project status with the developed software, than with traditional reporting methods. Secondly, there was found to be a correlation between users' satisfaction with the software, and the technical complexity of the method currently providing them with project progress reports. Finally, this research proposes some possible directions for future work.

CONTENTS

Enclosures:

Compact Disc of Procession v1.2 for Microsoft Windows

FIGURES

Acknowledgements

Alan Penn for academic and organisational support.

Arthur, Beth, Kim, Lucy and Sarah North for familiar understanding.

Derek Elsley at United House Construction for helping
to locate live project data.

Prof. Graham M. Winch for his ideas, supervision and support.

Dr Ruth Conroy Dalton for doctoral sympathy.

Dr. Ihsan Faraj at Building Research Establishment for help in connection
with the WISPER project.

Eileen Stewart for being endlessly perfect.

Mary Rawes for dog-sitting and making it possible.

NIHHA (New Islington and Hackney Housing Association) for evaluating the
Procession software tool.

okupi Ltd for 3D/Virtual Environments inspiration and allowing me to work
flexibly during the initial years.

Philip Pamment at SLD (South London Design) for helping
to locate live project data.

Professor Anthony Finkelstein for advice and feedback on requirements
capture, software evaluation and development.

Professor Mustafa Alshawi at the University of Salford for help in connection
with the WISPER project.

Sem and Goblin (RIP) for helping as only canines can.

Sheep (aka Nick Conroy Dalton) for having 'computer guru' status.

Southern Housing Group for evaluating the Procession software tool.

This work was supported by a UK Office of Science & Technology Foresight
Challenge Award. The Virtual Reality Centre For The Built Environment is an
interdisciplinary initiative between University College London, Imperial
College and a consortium of UK construction industry partners.

Chapter 1: Introduction

> "Good design can only emerge if there is a deep understanding of the real needs of the client. Ignore those needs or spend too little time in coming to terms with them and the result will be a building that fails to satisfy." (Construct IT Centre of Excellence 1996, p.6)

The client's role is central to construction. However, this is not always reflected in the availability of software tools for them to participate in the project. When clients remain ignorant of a project's day-to-day performance, the surprise of an unexpected negative outcome (in terms of delays and cost overruns) can be very alarming. Winch et al. suggest that putting "the minimisation of client surprise at the heart of the assessment of project success" (1998) would help to reinforce the client's role as the key member of the project. Winch et al. have gone on to propose that 'gap analysis', a service quality management concept, is readily applicable to the study of project performance. They define these 'gaps' as spaces between "the service that consumers expected that they were going to get, and their perception of what they actually got" (1998).

This research proposes that 'gaps', client ignorance and resulting surprise can be minimised by providing summarised project performance reports. These can be made available to clients remotely (via the Internet) and at a time of their choosing. The software tool presented in this thesis offers clients a simple 3D visualisation of project performance, in the form of a data surface. Existing project performance measurement tools (for example the s-curve) are not specifically intended for client use. The quantitative values that existing tools provide are the same ones used here for the 3D tool (basically variances in cost and time). However, accurate interpretation of s-curves is not intuitive for the amateur. There are no obvious graph peaks to indicate problems. Several curves must be compared to obtain performance measurements. In contrast, the 3D tool proposed here improves on this situation. It only reports by exception, displaying a flat 3D data surface if all is well. In addition, traditional performance measurement techniques describe all tasks simultaneously. Such techniques report the performance of the entire project. Therefore, it is not possible to identify specific project tasks that are the cause of any underperformance. It is true that individual s-curves can be produced for each task, but this then results in multiple information views, or sheets of paper. Adding a third dimension, allows all of the tasks (or at least the summary ones) to be represented in one view. Combining this with reporting by exception means that even a client unfamiliar with performance measuring techniques, can instantly identify problematic tasks.

This thesis will now introduce a clear set of research aims and objectives, which will be revisited in the conclusions. These are as follows:

Aim:

- To investigate whether 3D Information Visualisation can provide construction clients with informative performance reports.

Objectives:

- To develop a conceptual 3D framework that decomposes construction project performance into axial dimensions.

- To develop a software tool for construction clients that implements the conceptual 3D framework.

- To develop and apply a protocol for evaluating the developed software tool.

With this underlying structure in place, the actual research sequence will now be described. This will first be introduced in chronological order and then represented in terms of the chapter structure i.e. where to find each step documented. The first element of the research was a review of current literature and existing construction software. Of primary interest, was the availability (or not) of software for construction clients to obtain project performance reports.

Next, a requirements capture methodology was required, to ensure the appropriateness of the developed software. It was decided to focus on a sub-group of construction clients, those who work with social housing. This will be discussed in more detail later in this chapter. The Researcher's own experiences of working within this client sub-group, coupled with initial feedback from potential research subjects, suggested that they would prove technically inexperienced. Therefore, a requirements capture and development methodology was selected that allowed for them to be shown an initial prototype. If was decided that without this prototype example, the subjects might not be able to fully appreciate the potential of an Information Visualisation reporting tool. The chosen prototyping methodology was only arrived at after a review of the requirements engineering field and differing approaches to prototyping. Evaluation of the software prototypes was achieved through the statistical analysis of three questionnaires.

The evaluation methodology and the initial prototype had to be developed in advance of the first evaluation visit to the subjects. This meant that certain decisions had to be made, that might prove hard to reverse at a later stage in the research. The first of these decisions was the assumption that construction clients were familiar with and regularly used performance measurement techniques, such as the s-curve. This seemed a reasonable conclusion, given the inclusion of these techniques in standard textbooks and literature. When contact was first made with the subjects, the Researcher informally enquired as to any professional training that they had undertaken. Those subjects who had attended university, indicated at least a *theoretical* familiarity with performance measurement techniques. Evaluation of the prototypes partially depended on the subjects being familiar with s-curves, in order to provide a basis for comparison with the Information Visualisation data surfaces. Therefore, there was potential for the results to be adversely affected if the subjects were unfamiliar with s-curves. In order to prepare for this eventuality, the Researcher gave each subject an introduction to s-curve techniques (see Appendix 4).

A second assumption was that different types of construction client are equally interested in all of the possible performance deviation parameters (time, cost and quality). The implication of this was that the initial software prototype needed to implement and visualise all of the performance parameters. In the initial prototype, the chosen dimensions for the conceptual 3D visualisation space reflected this need to represent all of the parameters. However, the first visit questionnaires were designed to allow feedback from the subjects. This allowed modification of the parameters visualised in the operational (or second) prototype.

Research revealed that the construction industry already had project performance measurement techniques. Some of these made use of two-dimensional graphics. It was necessary to consider the applicability of these techniques to the proposed Information Visualisation tool. It was important to consider whether greater project insight could actually be provided by a third visual dimension. Great importance was placed on the extra visual dimension adding value to the existing two-dimensional performance graphic, for example the s-curve.

As 3D Information Visualisation was being proposed as a solution for a client software tool, an introduction to this field was prepared. Having considered existing project performance measurement techniques, morphological analysis was used to develop a new conceptual 3D framework. Morphological analysis allows a system, or problem, to have its fundamental constituents identified and represented in a conceptual space. Using this method, the elements of construction project performance were decomposed into axial dimensions.

Once the initial software prototype had been developed, the Researcher visited the subjects. Before showing the subject the first prototype, a five-point scale interview was undertaken to capture the subject's current performance reporting method and requirements. Next a fictional project scenario was shown to the subject in the form of a single 2D s-curve. The project was punctuated by seven milestones separated by quarterly (three monthly) intervals. The subject was asked to assess the project status at milestone four. The accuracy of this assessment was later scored on a five-point scale. After completion, the same scenario was repeated using the prototype at each milestone. At milestone four, the subject was again asked to assess project status. Immediately after the second run through the scenario, questions were asked about the information and usability of the software. The results of this first visit were collated and the outcomes applied to the development of a second prototype. On a second visit to each subject, a different fictional scenario was used to evaluate the second (operational) prototype.

With the research sequence described, this introduction will now detail how this process is documented on a chapter-by-chapter basis. Chapter 2 provides a contextual overview of current thought and literature in the areas of construction software, interoperability, standards and construction process modelling.

As stated, the research documented in this thesis is intended to investigate techniques for providing a reliable source of project performance information to the construction client. Specific focus will be on clients who are building (or renovating) social housing. The current term applied to such clients is Registered Social Landlords (RSLs). At an early stage in this research, it was observed that many construction software projects seem over-focused on the detail of the enabling technologies. However, they fall down when it comes to identifying a genuine requirement for the product and then evaluating it against that need. With this in mind, it was decided that a significant element in the contribution to knowledge should be a protocol for prototyping and evaluating the developed software. Chapter 3 investigates current thinking in the field of Software Requirements Engineering and specifically prototyping.

Humans are good at interpreting visual representations. Many disciplines have made use of this for the interpretation of mathematical and statistical data. This often entails physically visualising data sets that may be *intrinsically* multidimensional. The construction industry traditionally uses 2D s-curve type visualisations to analyse project performance. Chapter 4 describes the various analytical methods currently available and then briefly introduces the field of 3D Information Visualisation as a route to more informative performance reporting.

While the construction industry does use 2D graphing techniques to view information, little or no thought seems to have been given to more conceptual frameworks for representing its processes. Researchers, in non-construction fields, have suggested the application of theoretical dimensions to assist in the morphological analysis of complex data. In Chapter 5, a 3D morphological framework for visualising construction performance is proposed.

The software tool developed as a deliverable for this research is called 'Procession'. This name reflects its ability to report on the performance of a linear sequence of (sometimes concurrent) construction tasks. Building further on the prototyping methodology selected in Chapter 3, Chapter 6 documents the technologies underlying the three Procession prototypes (v.1.0, v1.1 and v1.2). In order to answer the question, "can 3D Information Visualisation provide construction clients with informative performance reports?", a number of hypotheses and a correlation check were formulated. Evaluating the software prototypes required a protocol and a data set. Chapter 7 details the theoretical process of obtaining example data from a real construction project, presenting the software prototypes to RSL Informants (this term is used from now on for individual subjects employed by an RSL) and capturing their requirements with interviews. Also documented are the statistical techniques used to calculate the significance of the interview scores and hence the hypothetical outcomes.

Chapter 8 describes the specific case of applying Chapter 7's methodology to the 'live' project data obtained for this research. As the data set contained only pre-construction baseline values, risk analysis software was used to predict likely problem areas during execution. The evaluation methodology required that two prototypes be produced and presented to RSL Informants, on two separate occasions. To avoid the subject becoming familiar with the program sequence of the example data, two separate scenarios were generated, based on the results of risk analysis. After the evaluation of the first prototype, it was possible to collate and consider RSL Informants' requirements for the operational version. Chapter 9 presents the final results, after the second visit to the RSL Informants. The hypothetical outcomes and their statistical significance are recorded.

Chapter 10 describes possible future research directions identified during the course of this study. Chapter 11 reviews the results of this research and provides a final summary.

Chapter 2: Research context and scope

2.1 Chapter introduction

The introductory chapter described the problem that provided the focus for this research and then presented the methodologies used to address it. The research problem is: "can 3D Information Visualisation provide construction clients with informative performance reports?". The purpose of this chapter is to provide context and boundaries for an otherwise open-ended question.

First of all, it is necessary to specify the nature of the information that is being reported to the client. For this purpose, some clarification of the terms *product*, *process* and *performance* is provided. It is then necessary for the reader to have an introduction to current practice in construction industry Information and Communication Technology (ICT). This includes any standards that might be applicable to the software tool Procession and the means by which existing applications communicate and exchange information. When considering ICT solutions, the availability of existing software for construction clients is of interest. Also relevant is the nature of the information that any such tools provide to clients and the reporting method (text-based, graphical, etc.) that they use.

With overall context provided, this chapter will then provide some tighter boundaries for the development and evaluation of Procession. For example, it was decided to focus on a sub-set of all construction clients and this group are introduced. In addition, reporting to clients was limited to one phase in the construction project lifecycle. In the light of Section 2.3, technical decisions were made about the source for Procession's information.

With a clear scope for the research having been established at this stage in the thesis, later chapters will further explore the methodologies used to achieve the stated aim and objectives.

2.2 What information is being visualised (the distinction between construction product, process and performance)?

It is important to understand the nature of the information being visualised by the Procession tool. The term used here for the source data is construction project 'performance' information. If should be noted that this is distinct from 'technical performance' information, a term sometimes used in the construction industry to describe the behaviour of building components or structures. Performance in the context of this thesis refers to the success or failure of a project, in relation to planned cost and schedule. It is common practice in project management to view a project in two ways. The physical components that contribute to the final product are described as the Product Breakdown Structure (PBS). The activities or tasks that constitute the process of building the product are known as the Work Breakdown Structure (WBS). Another way to express this is that PBS=product and WBS=process. Performance can be thought of as how well the process delivers the product, or to express it more formally:

> **Project performance is a quantitative measure of the processes' (WBS-Work Breakdown Structure) *actual* delivery of the product (PBS-Product Breakdown Structure) relative to a *planned* project baseline.**

The performance measurement parameters are time, cost and quality. Measurement of quality is more complex and subjective than for the more quantitative parameters of time and cost. Initial developments of the Procession tool focus on the time and cost parameters.

2.3 An overview of current construction software

This section provides an overview of construction software. The intention is to provide context for the proposed 3D visualisation tool. Winch (2002, p.345) describes Information Communications Technology (ICT) tools for construction as falling into two categories. Firstly, there are Engineering Information Management Systems (EIMS), which deal with the tools used to create and manipulate construction product data (for example, architectural CAD, etc.). Secondly, there are Enterprise Resource Management Systems (ERMS), which is a catchall name for systems that manage the process data surrounding a project (for example, payroll, document management, etc.).

Winch (2002) goes on to propose a third sub-category, Project Management Information Systems (PMIS). This sub-category intersects with both EIMS and ERMS and includes project planning tools. Examples of popular project planning tools include: Microsoft Project, PowerProject, Pertmaster and Primavera (Heesom & Mahdjoubi 2001). The tool that was developed for this research requires performance data. Information of this type is generated by project planning software. PMIS are not purely product (EIMS) related, nor do they only relate to process (ERMS). For example, project planning tools (as components of PMIS) are concerned with the sequencing of product 'on site' (utilising EIMS components) but are fundamentally describing the construction process (in the ERMS domain).

EIMS can be seen as frameworks for the software tools in a project that actually create data relating to product. The individual elements of an EIMS are called Engineering Information Creation Systems (EICS). Each element represents a specific project profession and the tools that they use. The primary EICS in an EIMS is likely to be the architect and the architect's Computer Aided Design (CAD) application, for example AutoCAD. Secondary EICS might include a Structural Engineer with a Finite Element Analysis (FEA) tool, or the architect's Ray Tracing or Virtual Reality applications.

EIMS is the 'glue' that holds its constituent EICS elements together and allows them to share information. The manner in which data is exchanged by EICS elements within an EIMS, can be anything from high level interoperability to a primitive so-called 'sneaker net' (i.e. walking over to a colleague's desk and exchanging CDs, floppy disks or print-outs). The applications within ERMS focus on tasks such as payroll, estimation, accounting and the exchange of documents relating to the project. ERMS can be considered one instance of Enterprise Resource Planning Systems (ERPS), sometimes also known as Enterprise Systems. Whereas Enterprise systems are designed to automate the flow of information with a company, ERMS provide similar tools on a project-specific basis.

Enterprise Systems tend to be concerned with unifying a single organisation. The most well known commercial implementation of an Enterprise System is SAP (Systems, Applications and Products). SAP provides an integrated package of business applications and is widely used in the oil, chemical and hi-tech industries. The SAP package for construction is known as SAP E&C and it features sub-applications such as: business planning, budgeting and contract verification (SAP 2003). The main drawback with SAP would seem to be high set-up costs and its reliance on proprietary applications.

As previously mentioned, CAD often forms the primary EICS element in an EIMS. A recent report on construction industry ICT in the US noted that one of the particular features of architectural design development which constantly provokes discussion, "is the use of apparently visual-only methods- notably drawing- when the phenomena under investigation may be only partly determined by appearance or other visual considerations" (Gann et al. 1996, p.12). At several points in this chapter it is noted that the architectural (or CAD) view tends to dominate construction ICT, both in terms of interoperability and standards.

Architectural CAD (particularly when it is 3D) is often referred to as 'visualisation'. It is this representation of product that is usually meant by the term 'Architectural Visualisation'. This differs from the intended meaning of 3D Information Visualisation, as used in this thesis. In fact, one recent paper uses Information Visualisation to describe advanced CAD (Aouad et al. 2000). This usage seems contrary to definitions commonly provided by practitioners in the visualisation field. Information Visualisation is widely described as relating to 'abstract' and 'non-physical' data sets (see Chapter 4 and Strothotte 1998). As such, CAD may be visualisation but it does not meet the criteria for Information Visualisation.

When time is added to 3D CAD, this is often referred to as 4D CAD- X,Y,Z and time (Mckinney et al. 1996, pp.383-389 and Mckinney & Fischer 1998, pp. 433-447). For example, in his 1997 paper 'Visualisation Technologies' Fischer states that CIFE's (Center for Integrated Facilities Engineering) 4D CAD system "captures space and time and puts them into one common model" (Construct IT Centre of Excellence 1997b, p.80). At its most basic level, 4D CAD allows evaluation of changes to the project schedule. When started, the 3D model builds itself in accordance with the planned programme. Other uses for 4D CAD include; spatial conflict resolution on building sites (North & Winch 2002), plant movement rehearsal and fire spread simulation (Mckinney & Fischer 1998).

The pace of technological change is likely to add new software types to those previously identified. Here are several examples:

- VR immersion techniques such as VR_Systems 'sphere' (1998). This is a ball with 360 degree projection on all its internal surfaces. A single user enters the sphere through a door and can then 'walk' inside the VR projection. The sphere actually rotates in response to the users walking motion, changing the projected view accordingly.
- Augmented reality- the superimposing of a VR image onto a real object. For instance, an engineer working 'on-site' could view the plans of an electrical installation overlaying the actual wiring.
- 3D scanning and reality acquisition of architectural components and site terrain- it is now possible to scan any 3D object into a computer. The

technology is rapidly becoming smaller and cheaper. It seems likely that its use will become more common.

- Tele-robotics: radio-controlled robots equipped with cameras can be used for site overviews/aerial observation, removal of dangerous contaminants, tunnelling etc..

It is now useful to discuss the type of report to be provided by Procession. Construction performance reports are currently generated by components of PMIS, such as Microsoft Project. Such reports often use existing performance measurement techniques, such as Earned Value Analysis s-curves (see Chapter 4). The reports may also take the form of a written document, describing performance and progress. Reports (particularly s-curves) are frequently intended only for the use of project managers but may also be circulated to the entire project coalition, including the client.

Tools specifically for construction clients are thin on the ground. The few examples that do exist are remote interfaces to a project planning tool, or commercial PMIS. For example, Spider is a Web-based tool that works with the planning application Open Plan (Welcome 2002). Such tools will often allow the client to assess project performance by allowing access to the current status of the planned programme. However, where such client interfaces are found, they differ from Procession as follows:

- The view is not designed specifically for the client and may be over complex.
- They do not use 3D Information Visualisation.
- They do not usually provide a summary of performance.
- They do not report by exception allowing 'at-a-glance' statusing.

Tools for construction clients might be described as Decision Support Systems. One definition of a Decision Support System for construction is a system that can "aid in the understanding of building construction and performance through visualisation and different forms of representation"

(Gann et al. 1996, p.5). Writing in *Information Technology Decision Support in the Construction Industry: Current Developments and Use in the United States*, Gann et al observe that:

> "Emerging construction processes utilising IT-based decision support systems are quite different in character from conventional approaches, for example:
> c) they facilitate a new notion of client/user involvement in the *total construction process*. Clients can be brought closer to the total process, whereas hitherto they tended only to be involved in specific aspects of design briefing and specification. The transformation in client/designer/constructor relationship is made possible in part by a number of technologies providing visualisation, auralisation, Virtual Reality visualisations and other realisations- not just to display ideas and proposals, but actually to enable clients and designers to think about them in fundamentally different ways." (Gann et al. 1996, p.9)

It is possible to conclude that there is very little, if anything, currently available specifically for the client.

2.4 Modelling, communicating and sharing construction information

The software developed for this research is dependent on information from existing construction software tools. As such, it is important to consider the current status of technologies that enable communication and data sharing within the construction industry. The primary motivation for this was the identification of potential standards or architectures that might have provided an interface for Procession.

The first area of consideration is 'modelling' in the construction industry. This term describes a variety of related but not identical approaches. Process Modelling provides an array of tools and techniques for analysing activities within the project life-cycle (Karsila et al. 2000, pp.179-186). The most basic techniques available are generic process mapping tools, providing little more than a library of symbols and graphic functionality. These are often used to analyse and document existing projects. Construction processes can be modelled at many different scales (or levels of granularity) and from many perspectives.

One of the most influential activities in the area of process modelling is the attempt to provide a generic chart of the relationships within all projects. The University of Salford's GDCPP (Generic Design and Construction Process Protocol) Group have attempted this with their Process Protocol Map (Kagioglou et al. 1998 and Fleming & Cooper 2000, pp.187-192). This is a widely adopted attempt to analyse and model construction processes, throughout the project life-cycle. Some of the phases in the map can be conducted concurrently. This is signified by the use of 'hard' and 'soft' gates. Phases can only overlap temporally, if they are joined by a soft gate.

In addition to the generic approach, process modelling can also be applied to specific projects. The performance of a project is directly related to the performance of its participants. As projects increase in size, stakeholders may increasingly need evaluation criteria and indicators for application both to themselves and other project actors. Kumaraswamy et al. suggest the use of flow charting to select participants for future projects, based on their past performance (1996a, pp.34-39). Baldwin et al. propose the use of data flow diagrams to provide a better understanding of information flow between project actors during a project's design phase (1999, pp.155-167). The IES (Integrated Engineering Solutions) Project uses a single graphical model to describe all stages of the design, construction and lifetime use of the building (Construct IT Centre of Excellence 1997a, p.61).

While process modelling has inevitable crossover with Procession (the use of graphics etc.), it does not provide any techniques that are directly applicable. Procession reports on the *performance* of processes, not the network of processes themselves (whether generic or project-specific). Another branch of construction modelling is potentially more relevant. This is known as Information Modelling, although it may sometimes be grouped under the process-modelling banner. Information modelling attempts to model some or all of the products and processes in the construction industry.

The models in themselves are not a file format, but they offer the potential for software interoperability by creating a common view of the project. Information models usually consist of text or graphical representations of individual elements, their properties, functionality (methods) and relationships. For example, an object such as a brick has size, weight, colour, material etc. Having this single view of project elements allows the simple creation of computing interfaces that conform to this agreed model.

The most well-known information model for construction has been developed under the general banner of ISO (International Standards Organisation) standard 10303 STEP (STandard for the Exchange of Product model data). STEP covers the exchange of data between software applications in many major industries including manufacturing and construction.

> "The intention is to create a single ISO standard that enables the capture of information needed to represent a computerised product model in a neutral form throughout the lifecycle of the product without loss of completeness or integrity." (Gann et al. 1996, p.92)

ISO 10303 provides a neutral mechanism capable of describing product data throughout the life cycle of that product. It is independent from any particular software application or computer platform. The format of the product description is suitable for neutral file exchange and also as a basis for implementing product databases or data archiving. STEP consists of many individually documented 'parts'. Each of these parts is referenced using the STEP prefix, followed by a unique part number. For example, part 11 (ISO 10303-11) describes the modelling language EXPRESS, which itself is used to write all of the other parts. STEP parts 40-49 (ISO 10303-40 to ISO 10303-49) are known as Integrated Resources. These provide the generic building blocks to describe any product. Part 49 describes processes and therefore provides a foundation for any product model capable of containing associated activity planning data. Parts 40-49 are never directly implemented. Instead, the relevant resources are used to build STEP

Application Protocols (APs). These are product models for a particular industry and have the part numbers 201-209 (ISO 10303-201 to ISO 10303-209). It should be noted that this is sometimes written as AP201-AP209. AP225 (ISO 10303-225) defines structural building elements in the construction industry. AP225 includes process functionality through its inclusion of Integrated Resource part 49 (as mentioned earlier).

AP225 has undergone many drafts, revisions and extensions since its first incarnation in the 1990s. Ultimately, though its success still seems dependent on adoption by the major construction software companies (Haas 2000, pp.25-31). Implementation of each industry specific Application Protocol (including construction's AP225) is provided by STEP parts 22-26 (ISO 10303-22 to ISO 10303-26). These parts include programming interfaces (for example, SDAI part 22) and the widely adopted part 21 (ISO 10303-21), a text-based, physical file format.

A complementary product information model for construction is provided by the IAI (International Alliance for Interoperability), a US group that launched its UK chapter in 1996. Its intention is to define a common way in which construction project objects are written and their attributes defined. The IAI calls its model Industry Foundation Classes (IFCs). IFCs are based on STEP AP225 and implemented using EXPRESS, although XML has recently been offered as an alternative modelling language. The file format offered by STEP part 21 has also been used in IFCs. There are now minor differences between the construction product model as originally represented in AP225 and the IFC model. There have been recent attempts by the two committees to address these differences and unify IFCs with AP225. The ability to model scheduled tasks is included in IFCs. This builds on the STEP part 49 resource found in AP225. The IfcControlExtension schema in the IFC core layer declares basic classes for project resources, tasks, allocations and milestones.

Autodesk (makers of the industry standard AutoCAD software) is a founder and key player in IAI. Microsoft also has a limited commitment to writing software that will utilise IFCs. The UK chapter of the IAI includes members such as: BAA, Lloyds, Ove-Arup, BDP, WS Atkins, Taylor Woodrow and Kvaerner. IAI also develops CAD standards in conjunction with the Construction Round Table, BAA, British Airways and Smith Kline Beecham. Despite this impressive list of contributors, Gann et al comment that, "IAI must define scope and practical working methods if it is to stand a chance of succeeding" (1996, p.17). One of the findings of Gann et al's US study was that non-Architectural project actors (for example engineers) are concerned that the IAI is leaning towards a reliance on CAD as a model: "The need for a neutral format was stressed so as not to lock information in to proprietary CAD standards" (1996, p.24).

A third model is UNICLASS (UNIfied CLASSification for the construction industry), which is intended for organising library materials and for structuring product literature and project information (Building Project Information Committee 1998). It consists of fifteen tables, each of which represents a different facet of construction information. Each table can be used on its own to classify a particular type of information. Terms from different tables can also be combined to classify complex subjects.

All three standards have in common a reliance on an architectural view of the project, rather than that of (say) a structural engineer, project manager or client. Where processes are included in the model, they are frequently represented as adjuncts to products. The product, rather than the task, is predominantly used as the fundamental unit. Gann et al noted that, "we have poor methods of representing construction processes, so that architects perforce rely extensively on visual representation as a means of maintaining a holistic project " (Gann et al 1996, p.13). STEP and similar data models may be seen as the potential 'glue' which may one day hold Engineering Information Management Systems (EIMS) together (see Section 2.3).

In relation to the development of the Procession tool, the key observation is that none of the current information models have the ability to describe performance. STEP, IFCs and UNICLASS all provide some level of support for processes (even if it is just a basic WBS). However, they do not yet consider the cost and time requirements of tasks and so performance measurement is excluded from the models. As such, while their introduction provides relevant context, none of these standards offer potential for interfacing Procession to other construction software.

In addition to work on standards such as STEP, another popular area of research in construction ICT are Integrated Project Databases (IPDBs). Sometimes IPDBs are referred to by other names. For example, 'the single project database' is a favourite. Amor and Faraj (2001) describe the many different ideas that constitute IPDBs and conclude that the construction industry is still a long way from achieving true interoperability. Amongst the IPDB variants documented are: a 3D CAD model acting as a gateway to project information, a centralised database for all project participants, a distributed project database using a standard programming interface (for example CORBA- see Object Management Group 2003), a STEP type information model and proprietary examples, such as Bentley's ProjectBank (Bentley 2002). Amor and Faraj suggest that IPDBs are not currently viable for several reasons. They argue that Object-oriented (OO) approaches such as STEP, IFC etc. are not well suited to representing large systems. Amor and Faraj also conclude that 'real world' construction is not OO in the sense used by computer science. Notions such as property inheritance within object classes are virtually non-existent in real-world building components.

Amor and Faraj (2001) support an opinion frequently revisited in this chapter, that CAD is often assumed to be the default view for determining information models and database structure. Many researchers working with IPDBs seem to assume that the views of different project participants are reconcilable. This is reflected in the prevalence of CAD and product dominated standards and system prototypes.

It is useful to reflect upon whether interoperability and IPDBs are important factors for the client. Unfiltered access to all project data is unlikely to offer a client any meaningful overview of current status. Clients (experienced or otherwise) seem unlikely to require the detailed information that would be provided by software tools designed for individuals actively involved in specific project area, such as the architect or project manager. Instead, it is extremely plausible that what clients really require is a simple, user-friendly software application that offers the client a 'window' on the project. The client may not actually care whether the data for a performance report is sourced from a stand-alone application, such as Microsoft Project, or an IPDB. However, even without directly using standards such as STEP, Procession is capable of working with either source.

For example, the project now detailed uses a STEP/IFC product model on the server, but still provides a Web-based interface compatible with Procession. In this example, server-side programming converts planning data from the IFC model into CSV (Comma Separated Value) format. The CSV data is returned in response to HTTP requests from a browser. The University of Salford's Automation and Integration in Construction (AIC) Research Group's PIT/WISPER Project (Faraj et al. 1998a) project was sponsored by the DETR under reference 39/3/336(CC1109) and led by John Laing, W.S. Atkins, Matthew Hall Ltd, and J. Wix Consultants. The name of the project was initially PIT but later changed to WISPER (Web-based IFC Shared Project EnviRonment). This is an Internet-based STEP/IFC project that utilises a three-tier client/server design, providing interfaces to VRML, Microsoft Excel and Microsoft Project. The lowest tier of this architecture consists of the project database. This is a single project database, as opposed to a distributed source. In a paper that describes WISPER (Faraj et al. 1998b), the AIC Group do not discuss the potential for distributing the data repositories and so it is assumed that this is not a possibility with this implementation. The significance of this is that it remains unclear as to the proposed location for such an architecture. Who would house and maintain the server? Is it suggested that there should be a new actor whose role is the provision of data management services? WISPER is capable of handling many concurrent projects. It is possible that data storage space could be purchased on a project-specific basis. There are certainly issues of trust and security to be considered. WISPER is described as allowing restrictions to be placed on levels of access and permissions to modify.

The second tier of the architecture contains a standard web-server, augmented with CORBA to handle information requests over the Internet. Both the first and second tiers are server-side. Project actors interact with the project database through a World Wide Web browser, the third tier. Architectural drawings, project plans etc. can be set-up and remotely modified over the Internet. In addition to browser interaction, certain common formats (Microsoft Excel and Project, for example) can be requested as downloadable files. This is the interface level that provides access for Procession to Microsoft Project data.

Having discussed potential architectures for interfacing Procession, the next section will move from context to scope. This will start by introducing the sub-group of construction clients who will be the target for evaluating the software.

2.5 Social housing and the Registered Social Landlord (RSL)

A simple definition of Social Housing (SH) might be that it is housing built and/or maintained by state funds. An SH construction client is likely to be one of the following; either a local authority (LA or local council) or a Registered Social Landlord (RSL), which is the new catchall term for registered Housing Associations, Housing Co-operatives and Local Housing Companies. There are several factors that make RSLs unique:

- They are experienced clients usually with 'in-house' development expertise.
- The nature of public grant funding (coupled with an increasing reliance on private finance) has several impacts-
    - It makes the number of abortive projects high.

- It often greatly increases the lead-in time from feasibility to grant confirmation (sometimes to years).
- It places specific restrictions in terms of project time-scale and component standards that often far exceed statutory requirements.
- Many of the projects will be 'new build' from an 'off the shelf' design tried and proved many times in the past.

From the software perspective, it seems likely that many RSLs (or at least their Employer's Agents) will be familiar with the popular low-end software packages for project management (Microsoft Project, for example).

2.6 Focus of the research

In order to provide some boundaries for the development of Procession and its subsequent evaluation, several constraints were imposed:

- The client group (subject population for evaluation) was limited to RSLs (see Section 2.5).

- Procession only reports performance during the 'on-site' phase of the project (i.e. Salford Process Protocol Map, Phase 8). This is the stage of the project when, anecdotal evidence suggests, clients need the most information but receive the least. It is possible that project actors understand a client's need for information during the pre-project and pre-construction phases, but assume that once most of the key decisions have been made, this need diminishes.

- The source for Procession's information will be Microsoft Project. This is because it is claimed to be one of the two most popular project management packages in the construction industry (Meridith and Mantel 1995, p.469, Knight and Kaka 1998, p.289), with a worldwide user base of approximately five million (Heesom & Mahdjoubi 2001, p.28). In addition, STEP/IFC projects are starting to provide interfaces to it. For example, the WISPER project as described in Section 2.4.

2.7 Chapter summary and contextualisation

This chapter has identified the subject of Procession's client reports as project 'performance' data. Project performance was described as a quantitative measure of the processes' *actual* delivery of the product relative to a *planned* project baseline. The parameters of performance to be included in Procession's visual reports were presented as 'time' and 'cost'.
The current state of ICT in the construction industry was introduced, along with attempts at standardisation. It was noted that many attempts at providing a unified ICT model for construction, assume that all project actors will be satisfied with the same 'view'. A common extension of this belief is that CAD should provide that primary view, or at least provide the underlying structure for any unified model.

The data source for Procession's visualisations was selected as the widely adopted Microsoft Project. This was chosen because attempts to develop a *de jure* data standard (such as STEP) are still far from accepted and currently fail to model information relating specifically to *performance*. In addition, standard models are very likely to continue providing interfaces for popular *de facto* formats such as Microsoft Project, as was found with the STEP-based WISPER project (Faraj et al. 1998a). At the point of selection, Microsoft Project appeared the most 'future proof' choice, even if not ideal in terms of it being a proprietary format.

In order to make the research described here more practical, certain constraints were applied to the scope. Procession's target end-users for evaluation were identified as Registered Social Landlords (RSLs), a sub-group of construction clients. Therefore the type of project was also restricted to the Social Housing (SH) domain. Further to this, it was finalised that Procession would only report to RSLs during the 'on-site' phase of the project (i.e. Salford Process Protocol Map, Phase 8).

It would seem that little or no software is currently available for construction clients. Even less of what is available provides a simple approach to performance monitoring. The application of Information Visualisation to performance reporting for clients seems unique. It may be concluded that construction clients are not used to having software tools developed specifically to meet their needs. As this research required the development of a new software application for a technically inexperienced client group, the next chapter will consider existing methods (and sequences) for capturing requirements. It will then propose an approach suitable for this research.

Chapter 3: Selecting a software development methodology

3.1 Chapter introduction

In the last chapter, it was explained that RSLs are inexperienced with advanced ICT solutions. Therefore, the method used to capture their requirements needed to allow for this inexperience. The detailed methodology (questionnaires etc.) used to actually capture the RSL Informants' requirements is described in Chapter 7. This chapter is more concerned with the timing of the software development. The selection of a research sequence that would enable the identification of the RSL Informants' informational needs was paramount.

In order to select an appropriate methodology, this chapter describes current practice and thinking in the phase of software development known as Requirements Engineering (RE). Any well-planned project might be expected to include a comprehensive planning phase. Such a project stage would concern itself with a thorough interpretation of the problem to be resolved. In software development, this stage is known as Requirements Engineering. In Section 3.2 a more detailed introduction will be presented. Having contextualised the area of discussion, the sections following this will consider both traditional and more recent approaches to software development and requirements capture. The final section describes the methodology to be used for this research.

3.2 Requirements Engineering

Requirements Engineering (RE) is the name given to the stage of a software development project concerned with identifying an end-user's problems and the mapping out of a strategy to address these problems. This process is sometimes also referred to as Requirements Analysis, or Software Systems Analysis, and therefore its practitioners are termed 'analysts'. If the focus of software engineering is accepted as constructing and maintaining computer-based systems, construction can typically be expected to commence with RE.

As will be discussed in the next section, the software development process has traditionally been viewed in linear terms. As such, RE would once have been seen as a time-based stage in the process, rather than as a task, which might be scheduled in a logical rather than temporal manner. Traditional RE tasks might once have been defined as follows:

- Problem recognition (requirements capture or elicitation)
- Evaluation and synthesis
- Modelling
- Specification
- Review

The 'review' phase, while presenting the opportunity for the process to loop back on itself, actually represented a 'hard gate'. Passing beyond this temporal point in the project indicated the end of the RE stage. With the introduction of methodologies that adopt concurrent and incremental development strategies, there seems increasing interest in the concept of 'evolving requirement'. Therefore, a revised consideration of RE categories might include a stage for handling changing requirement throughout the project. For example, Zowghi proposes the following task divisions:

> "1. Requirements elicitation which is the process of exploring, acquiring, and reifying user requirements through discussion with the problem owners, introspection, observation of the existing system, task analysis and so on. 2. Requirements modelling where alternative models for a target composite system are elaborated and a conceptual model of the enterprise as seen by the system's eventual users is produced. This model is meant to capture as much of the semantics of the real world as possible and is used as the foundation for an abstract description of the requirements. 3. Requirements specification where the various

components of the models are precisely described and possibly formalised to act as a basis for contractual purposes between the problem owners and the developers. 4. Requirements validation where the specifications are evaluated and analysed against correctness properties (such as completeness and consistency), and feasibility properties (such as cost and resources needed). 5. Requirements management refers to the set of procedures that assists in maintaining the evolution of requirements throughout the development process. These include planning, traceability, impact assessment of changing requirements and so on." (Zowghi 1996)

It can be seen from this that the task 'requirements management' has been added to describe the process of on going requirement assessment. As will be shown in this chapter, there are approaches, which handle requirement as an evolutionary process. While some offer more rigorous decision-making points than others, there is a danger that such methodologies may always compromise a project manager's ability to successfully predict and manage resource-requirement.

3.3 Traditional methods for software development

The development of software has traditionally been seen as a sequential process in which there is no distinction between the logical ordering of tasks and their actual occurrence in time. With this approach, requirement analysis is performed early on in the project and then remains unchanged for its duration. This methodology is variously known as the 'linear', 'top-down', 'lifecycle' or 'waterfall' strategy (see Figure 3.1, no. 1). The transitional points between project stages (i.e. from requirement to construction) are known as 'milestones'. Once a transition has been successfully made from one stage to the next, the work leading up to the milestone is considered to be 'approved'. It is then referred to as a 'baseline' for subsequent development. The Waterfall methodology is defined by a hard and fast rule of no 'backtracking'. Once a decision has been made it is unchangeable (Pressman 1992, p.24).

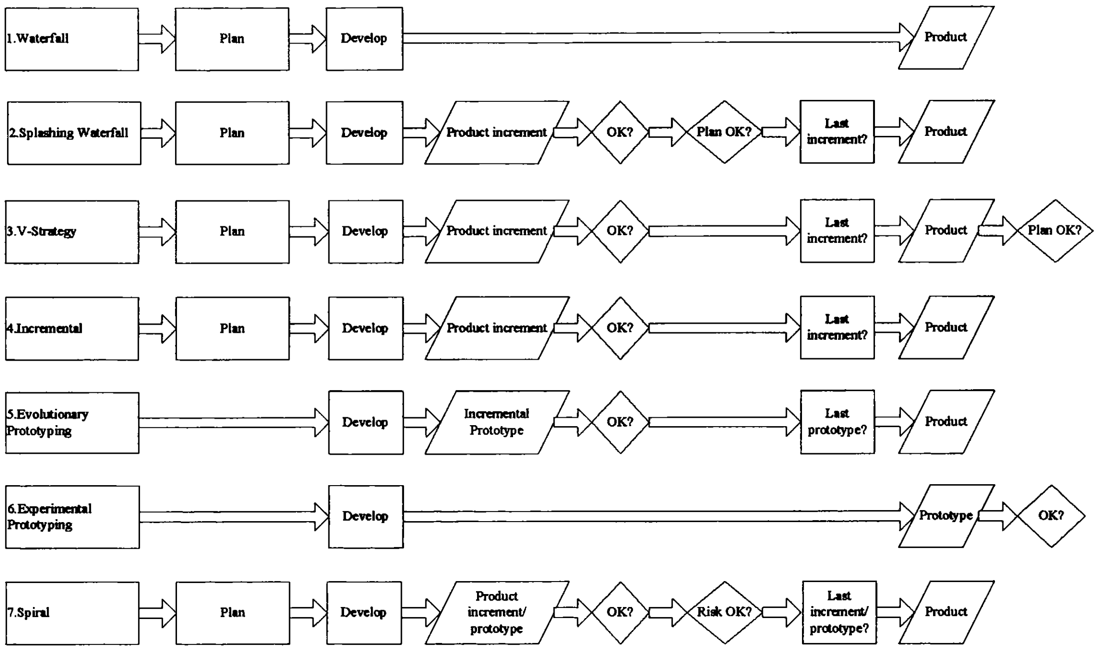


Figure 3.1 Comparing software development sequences, source: author.

3.4 New methods for software development

Recent methodologies tend to have abandoned the linear Waterfall model, discussed earlier. These 'non-linear' approaches can be broadly classified into; Waterfall Variants and Phased Development Models (including Prototyping).

Waterfall variants:

- In the Splashing Waterfall (see Figure 3.1, no. 2), each stage is permitted to 'splash back' to the previous one. Anecdotally, this methodology is very popular with programmers but hated by project managers. This is because milestones/baselines become meaningless and delivery dates cannot be guaranteed.

- The V-Strategy (see Figure 3.1, no. 3) allows baselines to be changed after implementation (Wieringa 1996, p.357).

The difference between these two is that the V-Strategy only allows backtracking at the evaluation stage, not at every stage.

Phased development models:

There are four models within this category: Incremental, Evolutionary, Experimental and Spiral.

- Incremental Development (see Figure 3.1, no. 4) uses the waterfall model for each of its steps. Requirements are base-lined at the start of development. Completed sections of the system are delivered to the client. The incremental order and therefore delivery is determined by a cost/ benefit analysis. The product becomes increasingly functional as elements are delivered.

The Evolutionary and Experimental methodologies utilise prototyping. The term 'prototyping' is used in engineering for the quick testing of technical and conceptual product models. These are known as 'throw-away' or 'rapid' prototypes and essentially conform to the Experimental development method, described below. They are usually delivered as a complete model for evaluation, rather than delivery as one increment towards a final product. To quote Brooks, "plan to throw one away; you will, anyhow" (1995, p. 116).

- For Evolutionary Development (see Figure 3.1, no. 5) the requirements are not base-lined and can be changed throughout development. Delivery to the client is incremental. In Figures 3.2 and 3.3, a comparison is made between resource-use in Evolutionary and Waterfall projects.

- For Experimental Development (see Figure 3.1, no. 6), requirements are not base-lined. The difference between Evolutionary and Experimental is that the former delivers a final product (albeit in reviewed stages), while the latter delivers a prototype to be reviewed once and then used. In the Evolutionary method, each increment of the product has individually passed through the traditional linear lifecycle, whereas Experimental prototypes are only reviewed retrospectively. Traditionally, this has been mainly used for research environments.

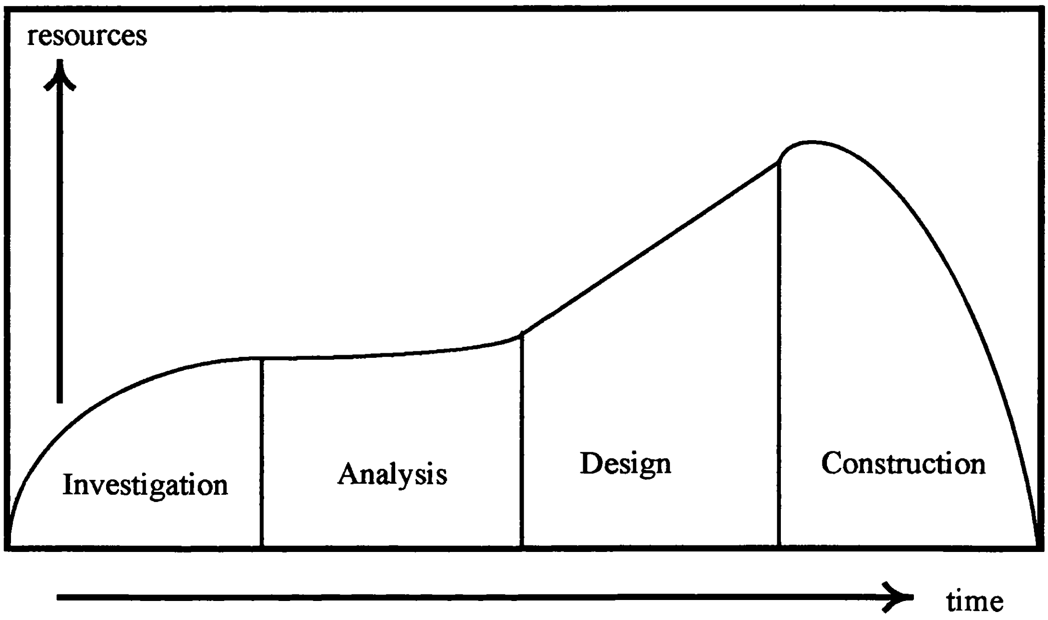


Figure 3.2 Traditional software development, source: Crinnion 1991, p.23.

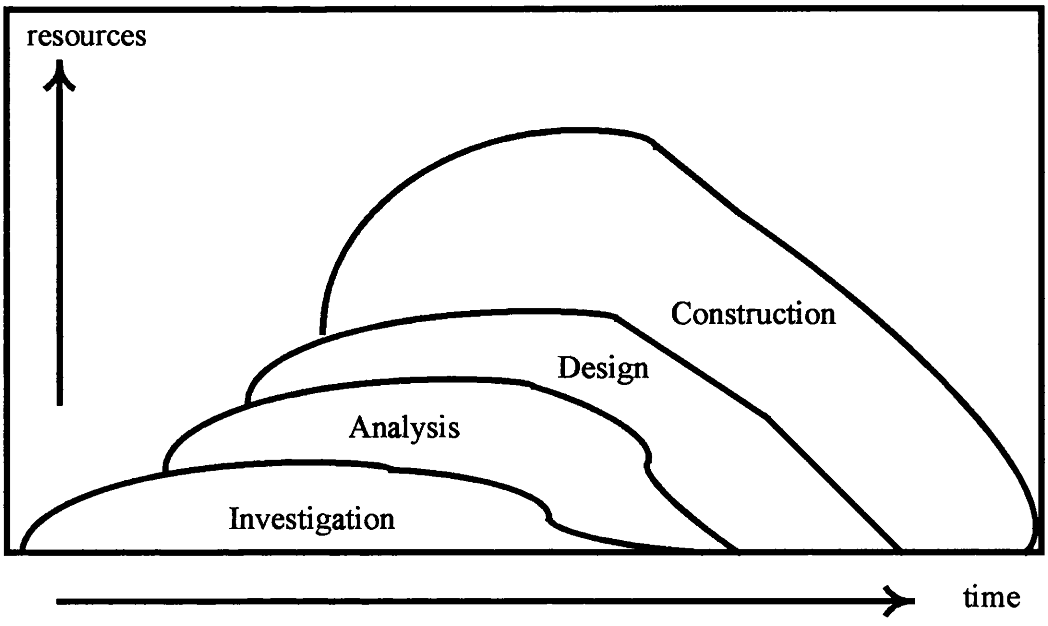


Figure 3.3 Evolutionary software development, source: Crinnion 1991, p.23.

- Another popular evolutionary paradigm is the 'Spiral Model' (see Figure 3.1, no. 7). This is a hybrid of the Waterfall method and prototyping (Pressman 1992, p.29). The key focus of the Spiral Model is on risk analysis. It can react to risks throughout the project. Wieringa defines risk as, " the possibility of not getting sufficient benefits from the product to justify the cost of developing it" (Wieringa 1996, p.377).

A transition to Evolutionary prototyping might help to enable greater client participation in the process. Pressman says, "it is likely that customers and users will become much more involved in the software engineering process as we move from a sequential view to an evolutionary view" (1992, p.768).

A final reference should be made to a new method called Extreme Programming (XP). This approach allows multiple opportunities for customer feedback during the development cycle. Formal software testing starts on the first day of development and prototype systems are delivered to customers as early as possible (Extreme Programming 2003).

3.5 The selected development methodology

Evolutionary Prototyping has been selected as a basis for the methodology, instead of the Waterfall approach. An explanation now follows. At the commencement of this research, a commitment to requirements capture and focusing on client need rather than technology had been stated. Initial discussion with potential RSL volunteers revealed a relatively low awareness of computer technologies and in particular 3D Information Visualisation (see Section 4.7). The use of Evolutionary Prototyping offered an approach that would allow for requirements capture but also allowed the Researcher to present the subject with an initial prototype to demonstrate 'ball park' functionality. As Brooks says, "the very existence of a tangible object serves to contain and quantize user demand for change" (1975, p.117).

Figures 3.4 and 3.5 provide a comparison between the Evolutionary Prototyping method and a revised method proposed for the software developed during this research. The primary modification to Jenkins' methodology was that the development timescale only allowed for two loops of the prototyping cycle. The red-outlined decision symbol on each of the figures highlights the difference between the two methodologies.

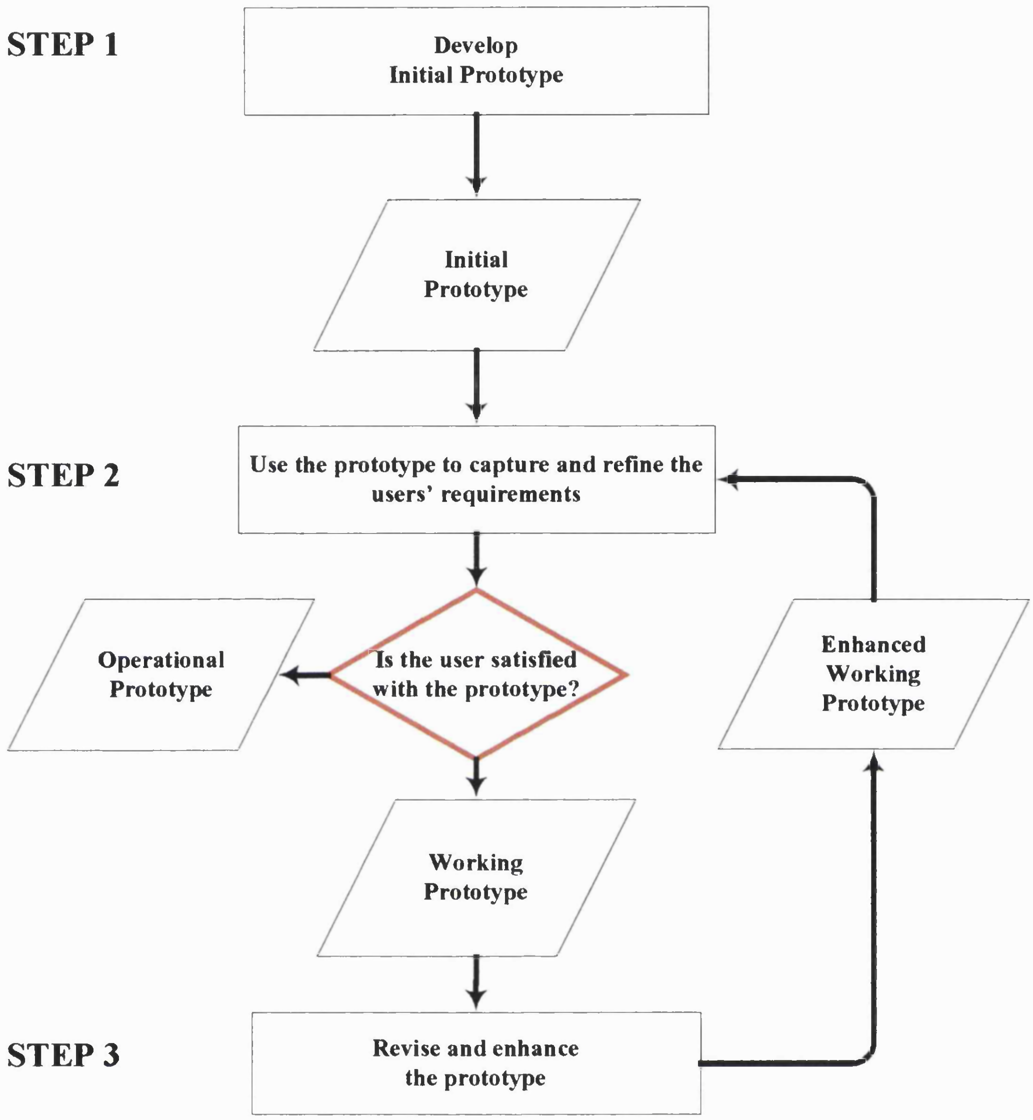


Figure 3.4 An adaptation of Milton Jenkins' Evolutionary Development Methodology, source: Crinnion 1991, p.32.

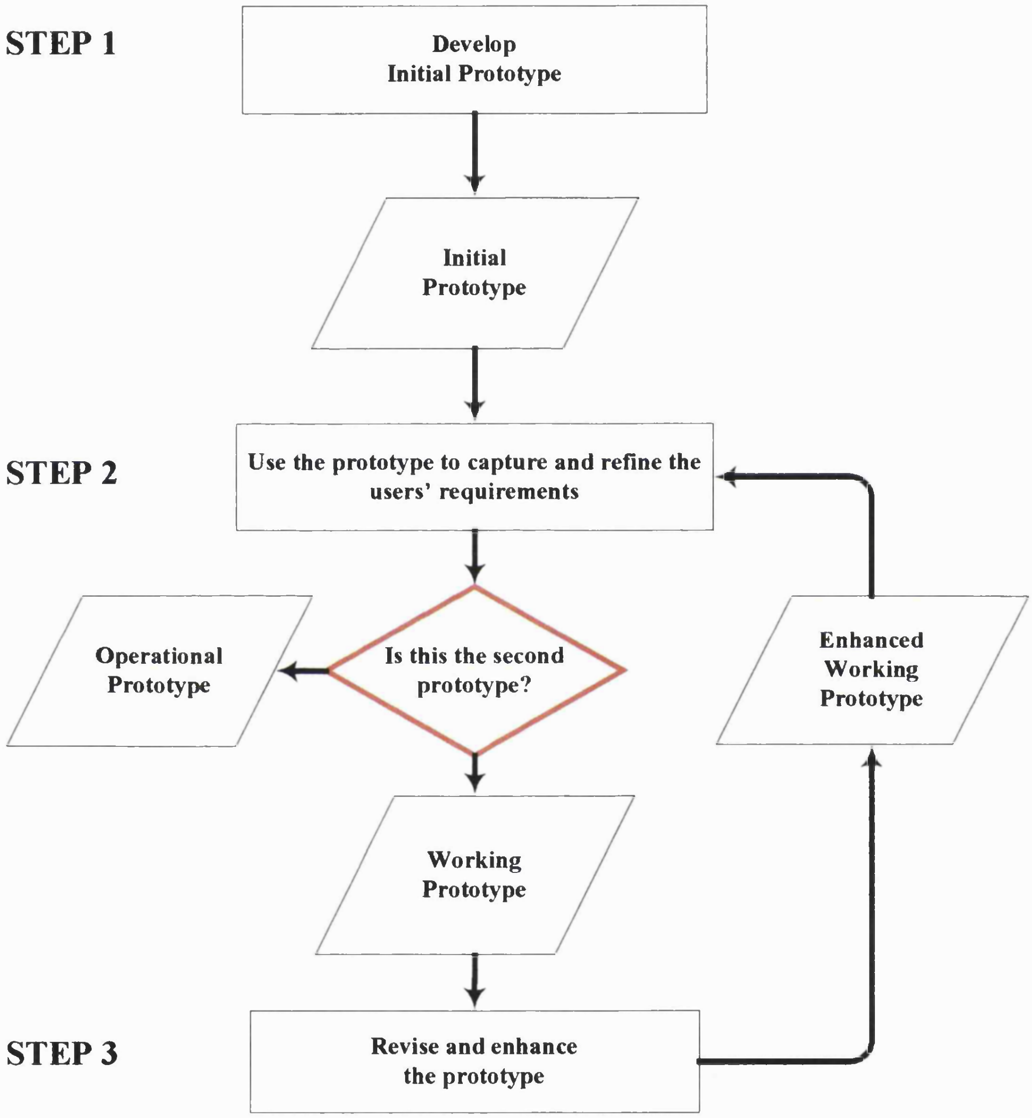


Figure 3.5 Prototyping methodology flow for this research: adapted from Jenkins' 1985 evolutionary model (Crinnion 1991, p.32), source: author.

3.6 Chapter summary

This chapter has provided a brief introduction to current practice in Requirements Engineering and its various sequential models. These range from the linear Waterfall model and its less linear variants, to the Phased Development Models including prototyping. An approach for developing the software tool has now been outlined. This is a modified version of Evolutionary Prototyping. Requirements are not base-lined and can be changed throughout development. Delivery to the client is incremental. The modification is that only two prototyping cycles are completed before the operational version of Procession is reached.

The intention is now to move from the software development sequence to considering the nature and design of the 'content' that will be visualised by the tool. The next chapter will discuss current techniques for measuring project performance. Discussion of current reporting methods will then lead into how the new tool's 3D Information Visualisation capability might enhance data intelligibility.

Chapter 4: Methods for measuring and visualising project performance (current and potential)

4.1 Chapter introduction

The evaluation of the software tool Procession considers whether it is more useful to RSL Informants than their current performance reporting methods. This chapter starts by describing traditional performance reporting methods. Information Visualisation is then introduced as an alternative approach. This chapter provides a basis for the migration from 2D single view performance reporting to a 3D multiple view approach. In order to develop an initial software prototype using 3D visualisation, it is useful to understand both current techniques for 2D project analysis and the 3D technology that Procession utilises.

4.2 Current method: s-curves

The traditional means of reporting construction project performance is the two-dimensional s-curve (Meredith and Mantel 1995, pp.457-459, Turner 1993, pp.203-204), with time along the x axis and cost on the y axis. Researchers have already proposed methods for detailed interpretation of the s-curve as a measure of construction performance (Khosrowshahi 1998 and 1999).

S-curves are so named because of the characteristic plot shape, generated by resource application take-up in many projects. The name refers to a phenomenon common in many activities, they take some time to accelerate at the start and then slow down towards the end. Although of considerable value, this approach is limited by the level of detail that can be displayed. The building blocks of projects are 'tasks'. Deeper interpretation of a project might require studying individual s-curves for each task. Indicative patterns can go unrecognised in this mass of unfiltered visual data. Recent observers have recognised the impact of small, task-level performance deviations on overall project performance. For example, Shi et al. (2001) have discussed methods for calculating the complex relationship between individual task duration variances and overall project delay.

Analysing project performance in this manner provides a 'single view' guide to data trends. Additionally, it is limited by the number of performance parameters that can simultaneously be represented, at a given point of observation. S-curves are generally utilised to analyse performance against budget in one of the three ways discussed in subsequent sections.

4.3 Current method: Earned Value Analysis

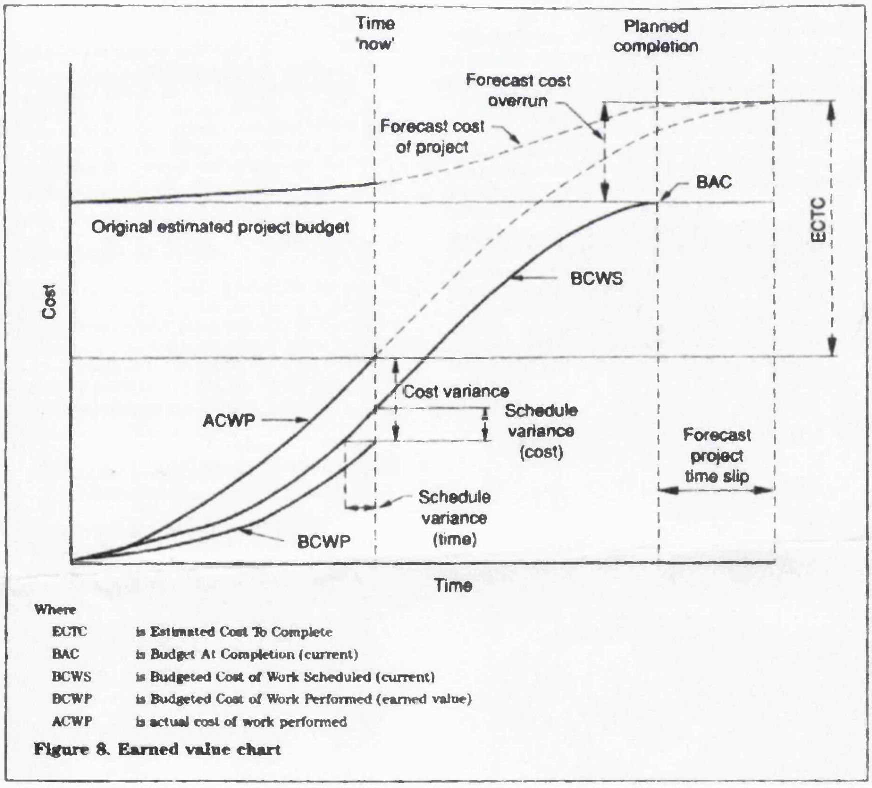


Figure 4.1 BS6079: Earned Value Analysis chart, source: British Standards Institute 2000.

Figure 4.1 shows the British standard definition for Earned Value Analysis (EVA). This is the most common of the three s-curve techniques. EVA was developed in the 1960s by the US Department of Defense and NASA (1962, 1967). It provides the total budgeted cost of completing the tasks actually completed at an observation point (time 'now'). This sum is the 'earned value'. Comparing this with the amount actually spent is a good indicator of project performance (Forsberg et al. 1996, p. 233). There is an acronym for each of the three curves plotted against each other for earned value analysis:

- ACWP (Actual Cost of Work Performed. Actual cost of performing the completed tasks).
- BCWS (Budgeted Cost of Work Scheduled. The planned budget for the scheduled tasks).
- BCWP (Budgeted Cost of Work Performed. The planned budget for the tasks that were actually completed- the earned value).

Interpretation of the generated s-curve is now described. All three curves should correspond if all is well with the project. This indicates that the budgeted amount of money has been spent to achieve the budgeted amount of work. If ACWP is greater than BCWP at point of observation, the difference is the Cost Variance. This means that the work completed is over-budget. If BCWP is less than BCWS, the difference is the Schedule Variance. This indicates schedule slippage, because it uses the budgeted costs as a progress indicator i.e. if the budget for painting a wall is 2 days and £10, at the end of day 1, BCWP and BCWS should both be £5. If BCWP is less than BCWS (£5), then the schedule is behind. Cost and Schedule Variance are performance deviation parameters, as described in Section 5.3.

4.4 Current method: Completion Cost Analysis

In Completion Cost analysis, an s-curve plots three lines against each other to compare the original budgeted cost of the project and the revised cost, from the point of observation. There are three acronyms for these curves:

- BAC (Budget At Completion. The planned budget).
- EAC (Estimate At Completion. The cost at completion revised as project progresses)

=ACWP+((BAC-BCWP) x (ACWP/BCWP))

- ETC (Estimate To Complete. The cost required to complete the project as at time 'T')

= (BAC-BCWP) x (ACWP/BCWP)

4.5 Current method: Performance Indices

Performance Indices provide a third approach to analysis. In this method, two curves are plotted. One represents cost and the other schedule. For both plots, a flat horizontal line at £1 indicates that £1 expended is earning £1 of completed project. Values below £1, suggest that the project is either over-budget or behind schedule. In addition to providing 'to date' progress, the performance indices also indicate the underlying future trends in the project. The two curves represent:

- CPI (Cost Performance Index)

= BCWP/ACWP

- SPI (Schedule Performance Index)

= BCWP/BCWS

4.6 Current method: Project statusing

The term 'project statusing' is increasingly used in the construction industry to describe the "timely comprehensive measurement of project progress against the plan to identify variances and the seriousness of the variances if not controlled by corrective action" (Wideman 1999). Project statusing is a catchall term for both the performance measurement techniques so far described in this chapter and any other form of project progress reporting. Therefore, the purpose of the software tool developed for this research might be described as being statusing. This phrase does not specifically imply *who* is being reported to. However, the Wideman Comparative Glossary of Common Project Management Terms (1999) describes the preparation of a 'status report' as follows:

> "A written report given to both the project team and to a responsible person on a regular basis stating the position of an activity, work package, or whole project. Status Reports should be used to control the project and to keep management informed of project status...a report issued by the project team that provides an overview on the status of the project. Usually included are scope, quality, risk, schedule, staffing effort and spending status. Project progress, variance, and corrective actions are summarized by project mangers in brief status reports. These are collated by the program support office into a program progress report, which is issued prior to each program executive meeting".

The implication would seem to be that a status report is not intended for the client. A possible explanation for the client's exclusion from Wideman's definition can be found in Levine's paper Real-time Status vs. Period Data (1999). Levine compares the traditional method of project reporting with contemporary statusing, which is now possible in real-time. Of the older, paper-based systems Levine states: "this process could often take up to three weeks, by which time much of the data was aged and the ability to react was impaired" (1999, p.1). By way of contrast, Levine recounts a colleague who wanted access to a centralised database, with data never more than twenty-four hours old.

This type of real-time statusing is now technically possible, but is also vulnerable to the GIGO (Garage In Garbage Out) phenomenon, in common with other information systems. Which is to say, status reports are only as reliable as their source data. Reliability can be affected by a variety of factors. For example, differing 'as of' dates i.e. team members have not entered data up to the same date. Even where a deadline for data entry has been specified, with a project 'on-site' it is not always logistically possible for team members to prioritise administration.

It is also possible for temporary data mistakes to be made. This could occur where a team member incorrectly enters data. This is then remotely viewed by other participants, before the mistake is noticed. There are knock-on effects of statusing conducted on inaccurate or incomplete data. These can include; unnecessary team stress and wasted person hours, as members communicate with each other and take unnecessary action to rectify non-existent problems.

Levine (1999) identifies one of the major problems with real-time statusing as being that the client will find out about problems, before corrective action can be decided. Earlier in this section, Wideman's definition of statusing was introduced. Concern over clients having unlimited access to project data, may explain the non-inclusion of clients as recipients of statusing data in Wideman's definition. Levine concludes by suggesting that even when real-time statusing is possible, periodic freezing and some form of editorial process may be required. He agues that the potential of real-time data should not be ignored, but that this data may be inaccurate without some form of moderation.

In terms of the research presented in this thesis, an assumption has been made that the input data is reliable, through moderation or otherwise. The Procession software tool is capable of accessing data from a single application or database. However, in practice most anticipated use scenarios would feature a single user entering all of the required data. While this does not completely remove the potential for misleading data, it is greatly reduced when compared with multi-user environments.

4.7 Potential method: 3D Information Visualisation

Information visualisation is a new and rapidly developing field, emerging from the well-established areas of data visualisation and scientific visualisation (Chen 1999 & 2002). Card et al. define Information Visualisation as, "the use of computer-supported, interactive, visual representations of abstract data to amplify cognition" (1999, p.7). As a technique, it can be applied to finance, database structure, business processes and project planning (Spence 2001; Cleveland 1993). As virtual environment techniques become readily available on a wide range of computer platforms, they are increasingly applied to the interpretation and analysis of multi-dimensional, electronic based information. The primary function of Information Visualisation is arguably to enhance the intelligibility of complex data sets, by revealing emergent patterns and relationships.

By definition, Information Visualisation is applied to abstract and generally non-physical data sets (Strothotte 1998). Project performance data is abstract in the sense that it is purely numerical and has no intrinsic spatial form. It is possible for abstract data to be represented in conjunction with a physical visual structure (Chi et al., pp.30-38). For example, colour scaling might be superimposed on a 3D building model, to represent levels of construction cost. This is a hybrid solution that has not been selected for this research. Instead, an attempt has been made to combine abstract and non-physical visualisation, so that the data can be spatially explored in isolation from its built product (North 2000a).

By way of further definition, Shroeder et al state that :

> "visualization is the transformation of data or information into pictures. Visualization engages the primary human sensory apparatus, *vision*, as well as the processing power of the human mind. The result is a simple and effective medium for communicating complex/or voluminous information." (1998, p.1)

Shneiderman (1996) describes the subject of his Information Visualisation taxonomy as follows:

> "Abstract information visualization has the power to reveal patterns, clusters, gaps or outliers in statistical data, stock-market trades, computer directories, or document collections"

Recent work by Chuah and Eick looks at the individual graphic elements within a visualisation, naming them 'Glyphs' (1998). They define glyphs as "graphical objects or symbols that represent data through visual parameters that are either spatial (positions x or y), retinal (colour and size), or temporal" (Chuah & Eick, 1998). This concept is then applied to the interpretation of large software projects, using both 2D and 3D methods. Chuah and Eick propose an almost linguistic approach to the intelligibility of project data. The work suggests a method of presenting data, which is easier for interpretation by the human eye. For example, making use of our natural ability to detect asymmetry, to reveal data deviation.

Other commentators have suggested that many of the new data sets made available by increases in computing power are inherently abstract. Foley (2000) states:

> "Consider the total amount of computer-processable information. Very little of it is about geometry and very little has an inherent geometry. Most of it concerns the stuff of which information visualizations are made, such as entities, relationships amongst entities, and properties thereof."

There is some Information Visualisation research that (at least superficially) has similarities to Procession. Zhang (1996, 1997 & 2000) describes the sub-field of Business Information Visualisation. Zhang has even applied Business Information Visualisation to project management and scheduling (1997). Despite the apparent relevance of Zhang's work to this thesis, it is actually concerned with preliminary planning in a manufacturing plant. Unlike Procession, Zhang's tool, VIZ_planner, does not report retrospectively on performance. Nor does it report by exception. In another piece of related research, Khosrowshahi (1998 & 1999) discusses s-curves in the context of Information Visualisation.

4.8 Chapter summary and contextualisation

S-curves are the primary method used in the construction industry for measuring project performance. There are three main types of 2D visualisation that utilise s-curves: Earned Value Analysis, Completion Cost Analysis and Performance Indices.

Information Visualisation is a field for representing abstract and generally non-physical data sets, such as project performance. 3D Information Visualisation provides a mechanism for expanding the single view of an s-curve into the multiple view of a data surface. It should be noted that the performance parameters reported by Procession (and described in the next chapter) are derived from measurements used in the Earned Value Analysis s-curve. For example, cost variance and schedule variance.
The difference is that that the EVA s-curve illustrates all performance (good or bad) and deviations from the baseline are measured relative to this. An EVA chart is likely to be curved, even if performance is as planned. Procession only reports by exception and only displays the deviations from the baseline. If the project is proceeding as planned, then Procession's data surface will appear flat.

An important part of Information Visualisation is deciding how raw data is mapped to the visual structures. The next chapter explains how the performance parameters of a project were mapped onto 3D space, as a morphological framework.

Chapter 5: A 3D framework for measuring construction planning performance

5.1 Chapter Introduction

In the last chapter, 3D Information Visualisation was presented as a method for reporting project performance. It was pointed out that the process of mapping data dimensions to visual dimensions requires careful consideration and justification. 3D views are not inherently superior to 2D views unless value is added to the information presented. A good starting point for the design of visual dimensions was thought to be the consideration of fields where work has already been done on conceptual frameworks.

One of the main claims for a contribution to knowledge made by this work is that it will provide a conceptual 3D framework for the interpretation of construction performance parameters. This chapter starts out by describing the 'morphological framework', a technique developed in software engineering for the conceptual representation of a system. The system is decomposed into its fundamental elements and mapped to spatial dimensions.

In this research, this approach is applied to the processes of construction performance. This usage is initially conceptual, but in later chapters the resulting 3D framework for measuring construction planning performance is utilised as a literal set of dimensions for Procession's data surface.

5.2 Morphological Frameworks

The field of computer systems engineering has a long tradition of employing three-dimensional frameworks to analyse its constituent processes. For example, researchers in the field of Requirements Engineering have proposed various conceptual three-dimensional frameworks, intended to aid the intelligibility of the software project process. Underlying this is the deconstruction of morphological form to reveal inherent properties. This approach has much in common with the cognitive amplification methods used in Information Visualisation. An early systems engineer, Hall describes morphological analysis as decomposing "a general problem or system into its basic variables, each variable becoming a dimension on a morphological box" (Hall 1969).

When a third dimension has been added to Requirements Engineering frameworks using 'time' and 'logic', it has tended to be either 'aggregation' (observation from different scales, or levels in a hierarchy e.g. social, computer system, software, software component) or 'aspect' (e.g. organisational, ergonomic, legal). It has even been suggested that both aggregation and aspect could be used in conjunction, making the framework four-dimensional. Wieringa states that:

> "A fourth dimension would be *aspect*. Important system aspects that may have to be [the] subject of development include the organizational, social, ergonomic and legal aspects." (1996, p.62)

Frameworks adopting dimensions of this type are usually described as being *method-orientated*. Software project frameworks using three dimensions can also be *specification-orientated*. For example, one framework uses the dimensions 'specification', 'representation' and 'agreement' (Wieringa 1996, p.69; Pohl 1994). In Pohl's framework, the purpose is to plot the transition of the requirements capture phase, from informal personal understanding to formal collective agreement. The specification process is imagined as starting at 0,0,0 on the three axes. At this point the specification is opaque, the level of agreement is personal and the representation is informal. The conceptual space cube formed by the three axes then represents the current state of the specification. The idealised project would produce a straight plotted line (X=Y=Z) i.e. the levels of formality, understanding and clarity increase equally.

While Pohl's approach is interesting, Hall's *method-orientation* also requires consideration. For his three dimensions, Hall uses logic ("problem solving procedure"), time ("problem solving process") and aspect ("knowledge of the developed system") (Hall 1969; Wieringa 1996, p.68). Figure 5.1 shows an example of a Hall framework. Hall's work in the 1960s was the forerunner of many later two and three-dimensional frameworks making use of time and logic axes (Wieringa 1996, p.368).

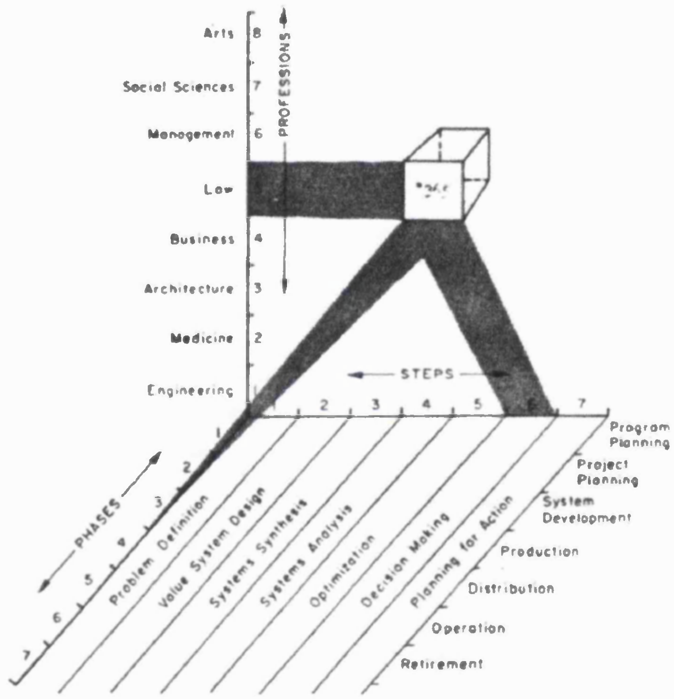


Figure 5.1 An example of a Hall three-dimensional morphological framework, source: Hall 1969.

Generally, construction industry process modelling seems not to have adopted morphological frameworks (see Section 2.4). However, recent commentators on project management (for example, Forsberg et al. 1996) have described a project in terms of three 'aspects'.

The use of 3D graphics, only seems justifiable if it adds to the intelligibility of a data set. An approach for construction process data will now be proposed. It is suggested that the concepts used for Hall's three-dimensional systems engineering framework may prove readily applicable to the interpretation of construction processes. This is suggested not only in the conceptual sense (as with systems engineering) but also for the more literal representation of processes through Information Visualisation. Figure 5.2 shows the proposed conceptual framework, with 'deviation parameters', 'deviation level' and 'tasks' as its three dimensions. Time is not included in the framework, but might later be represented by animating the data surface. Deviation level represents units of time or money, which deviate from the flat terrain of the project baseline.

In Section 4.2, s-curves were introduced as a monitor of overall project 'health'. It is proposed that the three-dimensional framework for construction planning performance developed for this research, enables a move from s-curves to data surfaces.

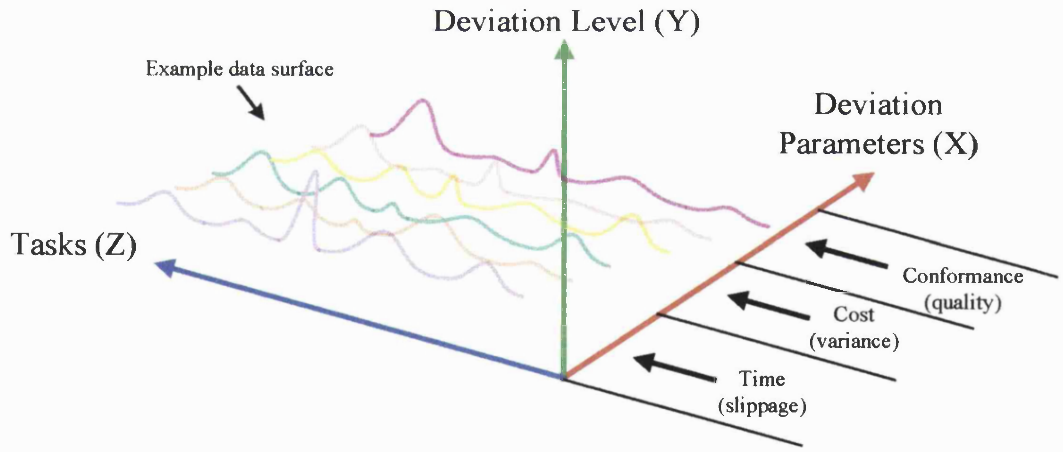


Figure 5.2 A 3D framework for measuring construction planning performance, source: author.

5.3 Performance deviation parameters and sub-parameters

Before commencing the 'on-site' phase of construction, a planning 'baseline' is usually developed from the project program. This consists of predicted durations and costings for tasks in the project lifecycle. A project baseline builds upon the previous experience of construction project managers to create a plan that represents their expectations for the project.

As construction progresses, the baseline is used for each project task, to compare the original plan for the project with its actual course. It is possible to see which tasks started earlier or later than planned, exceeded their original budget, took longer than planned, and so on.

There are actually three types of planning values: baseline (predictive and pre-project), estimated (revisions to baseline values made during a project) and actual. Before a project starts, all of its values are of the type 'baseline'. After commencement, uncertainty is slowly replaced with certainty. Values before any given project time 'T' are gradually transformed into the type 'actual'. Values after project time 'T' move towards 'estimated', as the baseline is revised. It is almost necessary to suggest that all baseline values become estimated values, at project commencement. Eventually, all values become actual values. Project progress reports are usually based on either actual *and* estimated data or just actual data. Reports combining actual and estimated data may be termed 'at completion'. They predict the outcome at completion of the project, based on current status. Reports using only actual data may be called 'to date', as they only consider progress up to project time 'T'. Project planning is essentially concerned with project data concerning 'time' and 'cost'. Reports generated for either of these parameters may be of the type 'at completion' or 'to date'.

Time and cost may be categorised as performance 'deviation parameters'. That is to say, they can be used to measure deviations from the project baseline. These can be seen on the three-dimensional framework for construction planning performance, proposed in the last section.

Further to this, there are a variety of measurements that can be made for each of the deviation parameters. Most of these correspond to elements of the standard Earned Value Analysis diagram (see Section 4.3 and Figure 4.1).

The most frequently used and relevant of these 'deviation sub-parameters' are-

Cost deviation parameter/reporting to date:

- COST VARIANCE, units £s. At this point in the project, is completed work on this task over budget (see Cost Variance in Figure 4.1)?
- SCHEDULE VARIANCE, units £s. At this point in the project, how are we doing on this task compared with the estimated spend (see Schedule Variance in Figure 4.1)?

Cost deviation parameter / reporting at completion:

- VARIANCE AT COMPLETION, units £s. At the end of the project, what is the difference predicted to be between the current estimate of the total cost for this task and the original estimate (shown as Forecast Cost Overrun in Figure 4.1)?

Time deviation parameter/ reporting to date:

- SCHEDULE SLIP, units days. At this point in the project, what is the difference between the elapsed time estimated and the actual elapsed time (shown as Schedule Variance -time- in Figure 4.1)?

Time deviation parameter/ reporting at completion:

- DURATION VARIANCE, units days. In Earned Value Analysis terminology, this is sometimes called Projected Program Delay. At the end of the project, what is the difference predicted to be between the current estimate of the total time to complete this task and the original estimate (shown as Forecast Project Time Slip in Figure 4.1)?

- WORK VARIANCE, units hours. At the end of the project, what is the difference predicted to be between the current estimate of total number of person hours to complete this task and the original estimate (not shown in Figure 4.1)?

It should be noted that a third deviation parameter might be described as 'conformance' or 'quality'. Objective measurement of this parameter is not a simple matter. As yet, conformance has not been implemented in the Procession software tool. One possible approach to measuring quality might be a quantitative assessment of 'snags' within a given time period. Whilst recognising its importance to construction projects, it has been decided that this is beyond the scope of this research (see Section 10.3 Implementing 'quality' as a deviation parameter).

Figure 5.3 presents the 3D framework with the addition of the deviation sub-parameters.

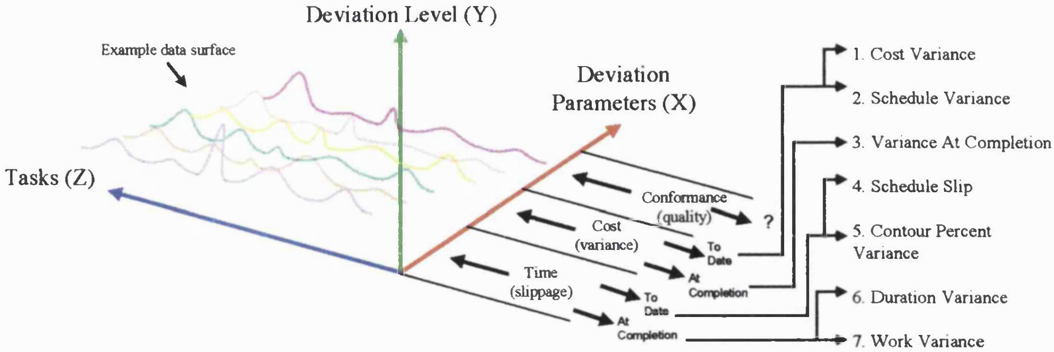


Figure 5.3 A 3D framework for measuring construction planning performance (including deviation sub-parameters), source: author.

5.4 Chapter summary and contextualisation

In Chapter 4, Earned Value Analysis was introduced as one of the main systems for 2D project performance reporting (see Section 4.3). This chapter has proposed a 3D morphological framework for construction planning. The author has described this in more detail elsewhere (North 2000b, pp.577-582).

The 3D framework has the following dimensions: X= deviation parameters, Y= deviation level and Z= tasks. The X dimension (deviation parameters) has the following deviation sub-parameters: Cost Variance, Schedule Variance, Variance At Completion, Schedule Slip and Duration Variance. It should be noted that the performance deviation sub-parameters (explained in Section 5.3) are derived from standard measurements utilised for Earned Value Analysis (see Section 4.3). It should be noted that Contour Percent Variance is a non-standard sub-parameter, developed by the Researcher and not relevant to this thesis. In the next chapter, the 3D framework is implemented as a data surface in the Procession software tool.

Chapter 6: Developing the software tool

6.1 Chapter introduction

This chapter documents the technologies used to develop the software tool Procession. The first section describes the three ways that Procession might be used in everyday practice. This description is provided to contextualise certain elements of Procession's functionality (interfacing, file formats etc), as discussed later in the chapter. Further sections sequentially introduce each of the three Procession prototypes and the two graphics programming technologies that were used in their development. After the first prototype (v1.0), a decision was made to change from one graphics technology to another. This was driven by the recognition of inherent technical limitations in this graphics technology, rather than user requirements. The second (v1.1) and third (v1.2) prototypes were the subject of evaluation for this research. The changes made to v1.2 were in response to user feedback after evaluation of v1.1. This chapter mainly focuses on the technologies underlying each prototype and the impacts of changes made between each iteration. More detailed information on the revisions made to the third prototype (v1.2), in response to user feedback, can be found in Appendix 10.

It should be noted that all versions of Procession could have been implemented directly using an industry standard 3D graphics library, such as OpenGL (or, alternatively Microsoft's Direct 3D). OpenGL is a cross-platform standard for 3D. The software runtime library is a standard inclusion with all Windows, MacOS, Linux and Unix systems. OpenGL's Architecture Review Board (ARB), an independent consortium formed in 1992, governs its specification. Members of the ARB include leading graphics vendors such as: 3Dlabs, Apple, ATI, Dell Computer, Evans & Sutherland, Hewlett-Packard, IBM, Intel, Matrox, NVIDIA, Microsoft, SGI and Sun (OpenGL home page

2002). This non-proprietary approach to OpenGL's definition and enhancement makes it a popular choice for 3D programmers.

It is important to realise that, although Procession was not programmed *directly* in OpenGL, both of the libraries used (VTK and TeeChart) are extensions built on top of an OpenGL foundation. Effectively, Procession's 3D functionality (in all versions) is provided by OpenGL. The justification for using the VTK and TeeChart extended libraries is simply that they provide pre-coded charting and visualisation functions. These allowed the developer to concentrate on programming related directly to the research question. Without these libraries, a great deal of time would have been spent 'reinventing the wheel', producing code for rendering and managing axes, scale increments, chart labels, graph types etc.

Chapter 3 described the prototyping software development methodology selected for this research. However, it is useful to very briefly discuss the approach taken with regard to two common development issues: Regression Testing and Version Control. Regression Testing is a commonly used technique in software development (Pressman 1992). This approach ensures that changes to new software versions, do not accidentally introduce errors into existing, working sections of code. Testing for both of the evaluated prototypes (v1.1 and v1.2) was conducted using the same data. This data was obtained from a 'live' construction project (see Chapter 8) and then used for evaluation of the prototypes, as described in Chapter 3. Using the same data removed the need for Regression Testing, as retrospective software faults would have been immediately apparent.

Procession only required a simple, manual approach to software Version Control (Pressman 1992). With only one programmer working on the software, it was possible to implement a system of regular backups combined with a standardised file naming procedure. Great care was taken to start a new 'draft' each time a subjective functional milestone was achieved. The previous draft was archived with a log file containing a list of 'previous problems fixed', 'previous problems unfixed' and 'new problems identified'.

6.2 Using Procession

There are three different ways that a development worker can use Procession. In the first scenario, the employer's agent exports a local data file from Microsoft Project and physically sends it to the development worker (using, for example, the postal service, email, personal delivery etc.). The second method sees the employer's agent up-load the data file from Microsoft Project to a traditional web server. This could be achieved with a standard File Transfer Protocol (FTP) tool or possibly using Microsoft Project's built-in ability to 'publish' on the Internet. The final approach is to make HTTP requests to a standard web server. Server-side programming would then query a database, structured in accordance with the STEP/IFC product model. For example, the University of Salford's Web-based IFC Shared Project EnviRonment – WISPER (Faraj et al. 1998 and Section 2.4 of this thesis).

6.3 Procession v1.0: The Visualisation ToolKit C++ libraries

The first version of Procession (version 1.0) had its 3D graphics functionality provided by VTK (the Visualization ToolKit), an 'open source' system providing a C++ class library, and a choice of Tcl/Tk, Java, and Python as interpreted languages. VTK runs on both the Unix and Windows platforms and its design has been driven by the principles of OO (Object-Orientation). In addition to its graphical functionality, VTK also provides a useful library of manipulative algorithms including scalar, vector, tensor, texture, and volumetric methods. VTK features advanced modelling techniques such as implicit modelling, polygon reduction, mesh smoothing, cutting, contouring, and Delaunay triangulation.

Early experiments showed that it was relatively easy to derive new C++ classes from VTK. In terms of negative observations, VTK trades off achieving the highest graphical speeds in return for its platform independence. The outcome of this is that it requires a reasonably fast computer to make best use of its full graphics capability. This version of Procession's data surface, as illustrated in Figure 6.1, is actually a 'carpet plot', generated by a scalar algorithm.

The three dimensions were achieved from a two-dimensional set of points, warped by scalar values in the direction of the surface normal. The amount of warping is controlled by the scalar value. The point set is determined by the 3D framework for measuring construction planning performance's 'tasks' and 'deviation parameters' dimensions (see Section 5.2). The scalar (or height value) is provided by the framework's 'deviation level' dimension. Early attempts at applying AI techniques created a fourth and non-spatial dimension, provided by colour mapping the data surface (see Appendix 16).

In this version of Procession, users can navigate the 3D data surface with a standard mouse. It is possible to:

- 'examine' the data surface (left mouse button and drag)
- move forwards and backwards (right mouse button and drag) or
- slide in any direction (shift key and the left mouse button).

Procession v1.0's menu options provide the following functionality:

- print current view,
- export a 3D report to Virtual Reality Modelling Language (VRML-version 2.0/97),
- take and save a 2D snapshot of a current view in the Windows bitmap (BMP) format,
- animate the data surface on a pre-set path and
- turn detailed performance alerts on or off.

Procession v1.0's rendering can be switched from solid, which is the default, to stereoscopic (requiring red and blue coloured glasses) or wire-frame. Procession v1.0 was designed as a partner application for Microsoft Project. Before Procession v1.0 can process Microsoft Project data files, they must be converted to a different format. Procession v1.0 imports data in the Comma Separated Value (CSV) format. Files of this type have the extension .CSV and are commonly available as export options from database and spreadsheet applications, including Microsoft Project. CSV uses an American Standard Code for Information Interchange (ASCII) text encoding structure, with individual values followed by a comma.

The final version of Procession v1.0 included example template files for Microsoft Project. These contained the CSV export map and Visual Basic for Applications (VBA) Macros, for automatically exporting projects to Procession and publishing CSV files to the Internet. Some early attempts were made to add Artificial Intelligence (AI) functionality to v1.0 (see Appendix 16). These ideas are revisited in Chapter 10, as a possibility for future work.

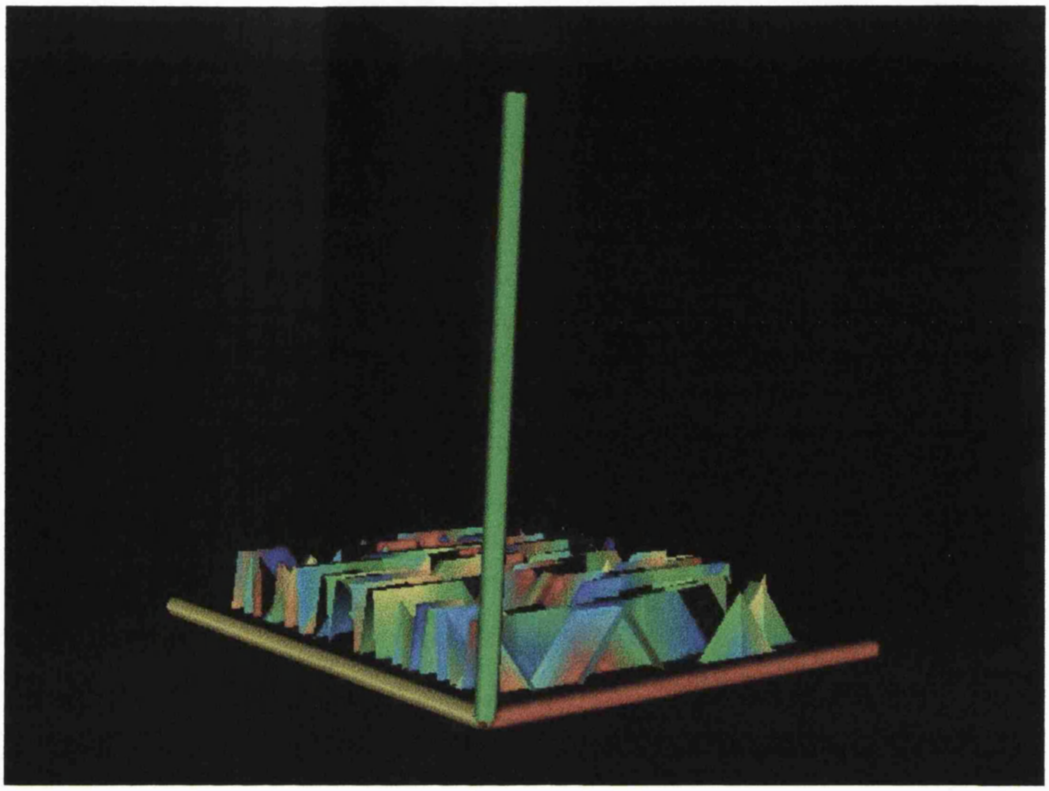


Figure 6.1 An illustrative screenshot of a Procession v1.0 data surface, source: author.

6.4 Procession v1.1 and v1.2: The TeeChart ActiveX object with C++

As described in the last section, Procession v1.0 was constructed using the Visualisation ToolKit. In many ways this proved to be successful but, before the development of Procession v1.1 (the Initial Prototype), a decision was made to switch 3D libraries. The main reasons for not using VTK were:

- It did not include in-built functionality for detecting and converting 2D mouse positions to data points (vital for using the mouse to explore the surface and display values).
- It did not provide functions for automating the labelling and scaling of axes.

Both of these were present in the TeeChart Active X control and so development was switched to these libraries.

However, as a result of changing from VTK to TeeChart, there were definite disadvantages to the visual appearance of the graphics. As previously described, Procession v1.0 generated a continuous carpet plot or data surface (see Figure 6.1). Individual points were joined to form a seamless surface. It had been intended that this metaphor be continued in the TeeChart version. It rapidly became apparent that this would not be possible. The nature of TeeChart's data structures made it easier to construct a data surface from individual ribbon plots, set side-by-side (see Figures 6.2 and 6.3). The end result is the same- in terms of dimension, interpretation and meaning but is possibly not so readily intelligible. On a more positive note, dividing the surface into separate ribbons, made it easier for users to discriminate between the performance deviation parameters. It seems likely that this might anyway have become an issue at the evaluation stage. Future work may allow a better-resourced software development programme and the reintroduction of the continuous carpet plot may then become possible. In the meantime, the term 'data surface' is used in reference to both ribbon-form and continuous carpet plots.

Transfer of the project from VTK to TeeChart proved to be relatively straightforward (particularly as no attempt was made to replicate the Artificial Intelligence elements – see Chapter 10 and Appendix 16 for detailed discussion). Procession v1.1 and v1.2 were both written in C++, using the Microsoft Visual C++ compiler and MFC (Microsoft Foundation Classes). Whereas, Procession v1.0 had a MDI (Multiple Document Interface), this was considered unnecessary for v1.1. and v1.2, which were both built on a standard Windows dialog. The important elements of the code for v1.2 can be found in Appendix 2. Unresolved technical problems with v1.2 are recorded in Appendix 17 for possible future work.

6.5 Changes in the visual design of the Procession prototypes from v1.1 to v1.2

Figures 6.2 and 6.3 show example data surfaces generated respectively by Procession v1.1 and v1.2. It is obvious from these figures that substantial changes were made to the interface between these two versions. These alterations were made in response to RSL Informants' feedback on Procession v1.1. More information on the specific changes made between version v1.1 and v1.2 can be found in Appendix 10.

This section will now consider the changes made to Procession's visual design between v1.1 and v1.2. This should be considered in conjunction with Figures 6.2 and 6.3. In Procession v1.1, a different coloured ribbon represents each performance parameter. The onscreen key provides a mapping between performance parameters and colours. For example, white is Cost Variance. The individual ribbons make up a single data surface. Time flow is not represented. The v1.1 view represents a snapshot in time. The base axis defines individual project tasks. Figure 6.2, uses simulated data and the task names along the base axis are replaced with automatically generated numbers. With real data, the task names would appear as text rotated 90 degrees to horizontal. This can be better seen in Section 8.8, where s-curves are shown paired with corresponding Procession data surfaces. The height dimension represents increments of deviation from planned performance.

Reporting is by exception and, therefore, a flat surface indicates performance is according to plan. Upward peaks in the surface indicate elements of performance that may be negatively impacting the project. Therefore, the user can equate height with poor performance. For example, where the white ribbon (Cost Variance) is positive in height this represents over-budget expenditure. It is important to understand that increments on the height scale may represent different types of unit for each ribbon/performance parameter. For example, the white Cost Variance ribbon is showing £ sterling. In contrast, green (Duration Variance) is reporting days of slippage. Therefore, no cross-ribbon comparisons can be drawn between equivalences of surface height. Individual ribbon height simply indicates a problem for that particular performance parameter. Its actual level of deviation must be analysed in isolation from other parameters with spikes.

The most noticeable modification in v1.2 (Figure 6.3) is the change of background colour from black to white. This was done in response to questionnaire feedback to v1.1. Each task is given an automatically generated colour. These colours have no significance other than providing distinction between individual project tasks. This was partly introduced to replace the use of on-screen task names. Users reported that long task names resulted in screen clutter. Truncating task names was not appropriate. Microsoft Project files often use long task names where the end of the string is the only unique identifier. Therefore, it was decided that v1.2 would use a mouse-rollover, with task information displayed on Procession's status bar. A different colour for each task allows the user to more accurately position the mouse. Using colour for individual tasks (instead of for individual performance parameters as in v1.1) had another impact on the visual design. As one performance parameter now had to contain multiple colours, using the same ribbon approach was impossible. The selected graphics library only allowed ribbons to be a single colour. Therefore, a filled surface was used as an alternative.

Multiple ribbons are no longer used in v1.2 because user feedback had required the limiting of reported parameters to one: Duration Variance. A fuller discussion of the implications of this is provided in Section 11.2 (Limitations of Research). As v1.2 has only one performance parameter, it was possible to make the labels on the height axis much more explicit. The parameter Duration Variance is measured in units of 'Days at Completion'. Clearly labelling this as 'Ahead' or 'Behind' the planned schedule provides the user with 'at-a-glance' feedback on project status.

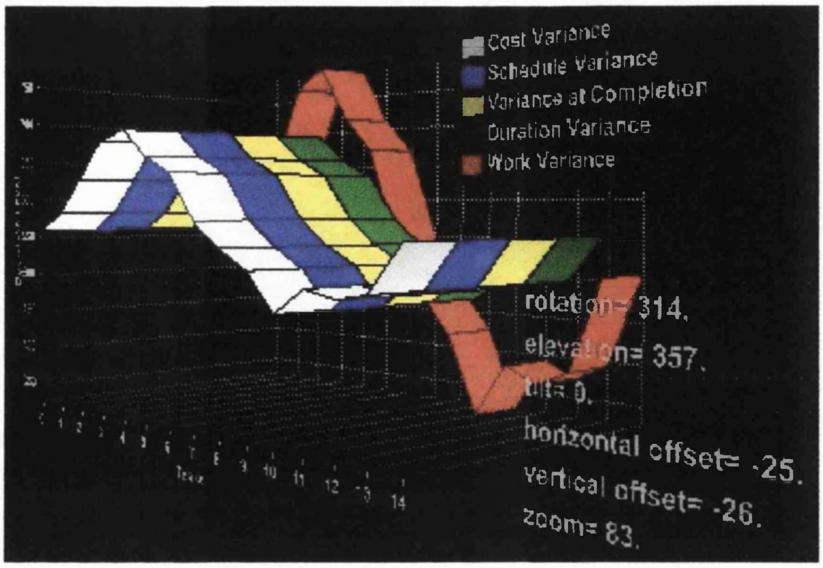


Figure 6.2 Procession v1.1: example data surface 3, source: author.

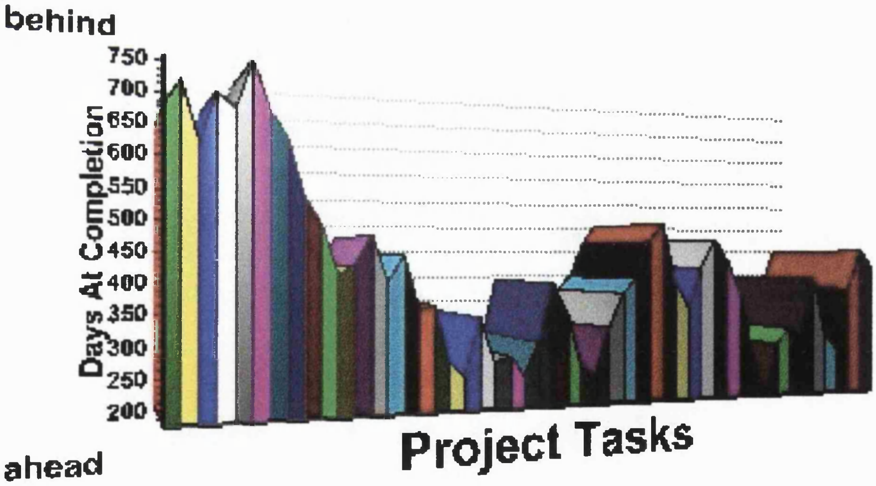


Figure 6.3 Procession v1.2: example data surface 2, source: author.

6.6 User documentation for Procession

It should be noted that a detailed help file was drafted in conjunction with the prototypes and can be found in Appendix 14. In Frederick Brooks' popular book on software development, *The Mythical Man-Month*, he recognises the importance of user documentation. He describes the manual as, "the external specification of the product" (1995, p.62). Brooks goes on to state that, "manual writers...must define what is not prescribed as carefully as what is" (1995, p.63).

6.7 Chapter summary and contextualisation

This chapter has described the practical development of the software tool Procession. Appendix 10 provides greater detail of the technical revisions made during the prototyping cycles. All versions of Procession were developed for the Microsoft Windows platform in C++. It should be noted that before the initial prototype (v1.1) was shown to RSL Informants, there was an earlier version (v1.0) constructed using a different 3D graphics library. Procession v1.0 was developed using the VTK graphics libraries, v1.1 and v1.2 used Teechart. Both of these libraries are built on a foundation of OpenGL.

With an initial prototype of the software now ready for demonstration, the next chapter provides an original protocol for this evaluation. This requirements capture and evaluation protocol builds on Chapter 3's selection of a modified Evolutionary Prototyping methodology. With the chosen software development methodology, requirements are not base-lined and can be changed throughout development. Delivery to the client is incremental. For the purposes of this research, only two prototyping cycles are completed before the operational version of Procession is reached. Chapter 7 discusses the specifics required to apply the methodology outlined in Chapter 3.

Chapter 7: Requirements capture and prototype evaluation methodology

7.1 Chapter introduction

In order to evaluate Procession, comparative research was conducted. The intention was to determine a selected RSL Informant's level of satisfaction with the quality and format of the project performance information provided. In Chapter 3, a new development cycle was proposed using Evolutionary Prototyping as a basis. In this chapter, a series of null hypotheses are proposed to determine the suitability of the software solution (see Section 7.3). This chapter describes the entire process of evaluating Procession and how statistical analysis was applied to the results.

7.2 The evaluation methodology

In order that Procession's informational provision improves on the current model, a requirements capture stage was undertaken. This made use of rapid prototyping to present the RSL Informant with an initial version for evaluation. The Researcher obtained a data set for a 'live' RSL project (see Chapter 8). This data came from an RSL not involved in the prototype evaluation. The data was originally in the Power Project file format and then converted to Microsoft Project.

The Researcher used the skeleton of this live project to create two project scenarios, with different sequences and outcomes. Monte Carlo Simulation techniques were applied to the example construction programme, in order to identify areas of potential risk (see Chapter 8 for detailed discussion of this process). One of the scenarios was used with an Initial Prototype and the other with an Operational Prototype. The purpose of this was to provide a testing ground for Procession, which had to be neither familiar to the RSL Informant (otherwise there would be no motivation for information requests) nor 'mission critical' to the RSL. A precedent can be found for this approach in Hackos and Redish's description of similar simulations as 'Prototyping Scenarios' (1998, p.396). Each of the project scenarios had seven quarterly milestone files produced, to represent the chronological sequence through the project. These files were stored on the Researcher's laptop computer, in the dBase (.dbf) format.

The quarterly milestone files were used to compress a project of nineteen months into approximately as many minutes. For this research, it was decided that 'real time' simulation of a project was not practical. For example, RSL Informants' need for information concerning the performance of 'on-site' projects is likely to make up a vital, but proportionally small part of their workload. As this information could have been required at unpredictable times, it did not seem a sensible use of resources to attempt direct observations of the RSL Informant's behaviour on a 'live' project.

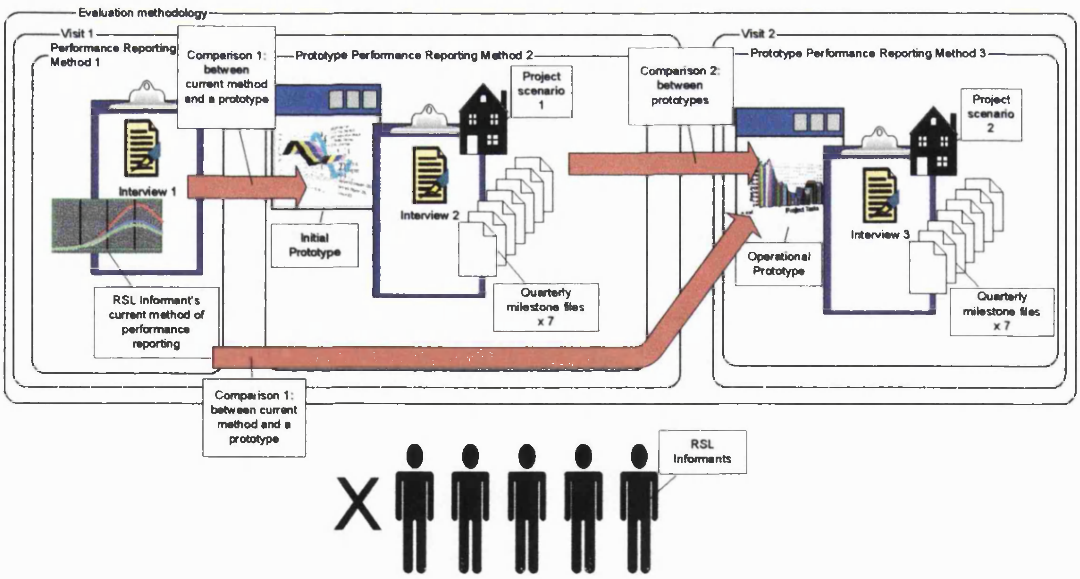


Figure 7.1 Diagram of Evaluation Methodology, source: author.

The prototypes were evaluated by five RSL Informants. It is appreciated that this sample size was very small, but it was all that was possible within the resources available. It should be noted that various efforts were made to increase the sample size, including the publication of an article in a housing publication (see Appendix 11) and presentations at housing events.

Following development of an Initial Prototype, the Researcher visited each of the five RSL Informants and asked them to evaluate the software. The Researcher utilised three interview protocols to evaluate the software-two protocols during the first visit and one for the second. Section A of the Client Information Requirement Protocol (see Appendix 5) considered the quality of information provided by the RSL Informants' current methods of performance reporting, allowing for later comparison with the prototypes. This Protocol also questioned RSL Informants about their general familiarity with computers. Attempts to identify the RSL Informants' current reporting methods focused on two factors: firstly, TECHNICAL COMPLEXITY (verbal>paper>computer) and secondly TYPE OF INFORMATION (descriptive>numerical>graphical).

The questions in Section A (Perceived quality of information provided by CURRENT METHOD of progress reporting) of this protocol were only asked before the RSL Informants had ever seen a prototype, so as not to influence their expectations.

The second and third interview protocols (Prototype Usability and Information Evaluation, Appendices 6 and 7) are identical documents, differing only in the prototype to which they are applied and the s-curve on page 1. Both of these protocols contain the following sections, which were applied at different stages in the evaluation process:

- Section A. Perceived quality of information provided by the prototypes
- Section B. Usability of software.

Interview Section A covered the RSL Informants' general level of satisfaction with the quality of information provided by the prototype. Section

A enabled a comparison with the RSL Informants' current methods of performance reporting. It also provided data for comparison between the results obtained for the two prototypes.

The interview questions in Section B related to functional aspects of the software prototype. This section was completed AFTER both the application of the project scenario and Section A. The purpose of this was to assess not the quality of the information that Procession presents, but the usability of the software. As such, reference was made to standard guidelines for user-interface design, which are widely available (for example Pnueli 1986, pp. 845-858). Section B provided a comparison between the results obtained for the Initial and Operational prototypes. RSL informants were shown each of the seven milestone files using Procession. The level of 'hands on' interaction from the subjects depended on their confidence with computers.

In order to verify RSL Informants' ACTUAL level of informational understanding, they were shown the scenario both as a traditional s-curve generated using Earned Value Analysis (see Chapter 4) and then using Procession. The Researcher had pre-scored the performance of each project scenario at milestone 4, using a list of codes provided at the back of each prototype questionnaire (see Appendices 6 and 7). Informants were asked to choose the code that most accurately described the current project status at milestone 4. The standard five-point scale was used to score each Informant's impression of performance. For the s-curve version, one curve was used and covered with a sheet of paper. This was done to obscure both the later milestones and the overall trend of the curve. The Researcher then progressively uncovered the project milestones.

One possible criticism of this approach is that having seen the milestone 4 s-curve, RSL Informants would be familiar with the project status, before seeing the data surface. The answer to this is that the Researcher did not tell the Informant whether or not the code chosen for the s-curve was correct.

The use of observers to record and analyse how users perform tasks is an accepted methodology within (amongst others) the field of Requirements Engineering (Mumford 1995). It has been argued that, where an observer is physically present, notes taken should be of not only the user's exact behaviour but also of the observer's inferences concerning this behaviour (Hackos and Redish 1998, p.264). Inferences may provide vital insight and so provision was made within the interview format for these to be recorded.

The Researcher next collated the findings from all of the volunteer RSL Informants and developed a second Operational Prototype of Procession. The findings were compared:

- between each RSL Informant's current reporting method and the Initial Prototype, with specific reference to the perceived quality of information provided (Section A, Client Information Requirement Protocol and Section A, Initial Prototype Usability and Information Evaluation Protocol)

In addition, the software usability responses (Section B, Initial Prototype Usability and Information Evaluation Protocol) were analysed. This informed the Operational Prototype development process.

Once the Operational Prototype was considered ready for evaluation, the volunteer RSLs were again individually visited. The second project scenario was substituted for the first (including an s-curve for milestone 4) and the experiment was repeated as previously detailed. Five RSL Informants participated in the research and so this resulted in a total of fifteen completed interviews being available for analysis.

Finally, the Researcher compared the results in two directions (as shown in Figure 7.1):

- between each RSL Informant's current reporting method and the prototype solutions, with specific reference to the perceived quality of information provided (Section A, Client Information Requirement Protocol, Section A, Initial Prototype Usability and Information Evaluation Protocol and Section A, Operational Prototype Usability and Information Evaluation Protocol)

AND

- across prototype versions, in terms of informational requirements and software usability (Section B, Initial Prototype Usability and Information Evaluation Protocol and Section B, Operational Prototype Usability and Information Evaluation Protocol)

A combination of statistical scoring and analysis of the RSL Informants' impressionistic responses made up the final research results.

7.3 Evaluation hypotheses

In order to evaluate Procession and the proposed conceptual 3D framework for measuring construction planning performance (see Chapter 5), an experiment was undertaken to test the following null hypotheses:

Hypothesis 1

Can RSLs assess project status more accurately with software (both prototypes) than with the application of traditional methods?

H0: $M_b \leq M_a$: RSL Informants' MEAN ability to assess project status is the same or worse with both prototypes than with current methods.

Hypothesis 2

Is the second software prototype easier to use than the first prototype?

H0: $M_b \leq M_a$: Software usability GETS WORSE or DOES NOT improve from initial to operational software prototype.

Hypothesis 3

Is the QUALITY of the information perceived to be better in the second prototype than the first prototype?

H0: Mb ≤ Ma: RSL Informants' perceive the quality of project reports to be the same or worse with the operational prototype than with the initial prototype.

Hypothesis 4

Is the RELEVANCE of the information provided perceived to be better in the second prototype than the first prototype?

H0: Mb ≤ Ma: RSL Informants' perceive the relevance of project reports to be the same or worse with the operational prototype than with the initial prototype.

Hypothesis 5

Is the FORMAT of the report perceived to be better in the second prototype than the first prototype?

H0: Mb ≤ Ma: RSL Informants' perceive the format of project reports to be the same or worse with the operational prototype than with the initial prototype.

Hypothesis 6

Is there a negative or positive correlation between users' satisfaction with the software and the technical complexity of the method used to provide their project progress reports?

H0: Correlation r = 0: There is NO negative or positive correlation between RSL Informants' satisfaction with the software and the technical complexity of the method currently providing their project progress reports.

It should be noted that the standard notation 'H0' is used to indicate that each of the above are null hypotheses. The desired outcome for null hypotheses 1-5 was their rejection with 't' values (the significance of the difference between the means of two groups) the same or larger than the value given in the Table of Critical Values of t. For null hypothesis 6, a correlation value (r) not equal to zero would indicate rejection of the null hypothesis. It was further hoped that the value of r would be within the range indicating significance for a sample of this size.

7.4 Method for statistical analysis of interviews

The evaluation experiment was conducted with observations of the same RSL informants at three different points (three interviews- one concerning current reporting method and two on the prototypes, as shown in Figure 7.1). In similar situations, many researchers have applied the approach used here of treating the different measures as independent samples, drawn from the same RSL Informant source population (Everitt and Hay 1992, p.85). Therefore, the same RSL Informant, when in a different sample group, was treated as a discrete observation. The groups a, b and c corresponded to the three interview questionnaires; Client Information Requirement Protocol (a), Initial Prototype Usability and Information Evaluation Protocol (b) and Operational Prototype Usability and Information Evaluation Protocol (c).

It is common practice in an experiment such as this to apply one-way ANOVA (ANalyis Of VAriance) for three independent samples (Everitt and Hay 1992, p.69 and Lowry 2000, chapter 13). While this reveals the level and significance of overall variance between three groups, it fails to offer multiple comparisons between pairs of groups within the population. The nature of this evaluation (two software prototypes and the RSL Informants' existing reporting methods) made multiple comparison essential. Consequently, the chosen analytical method was the paired t-test for the significance of the difference between the means of two groups. All of the t-tests documented were 'directional' or 'one tail'. That is to say, the experimental hypotheses prescribed not just a difference between means, but specifically which of the means would be the greater. The t-tests were also 'univariate', only one variable was compared between groups.

The alpha level α was defined as the standard 0.5 or 5%. This represented the probability that rejection of a null hypothesis was not due to pure chance. Another way of regarding the α significance level is that there is a 95% probability that the results were not random. The assumptions for all calculations are presented here, but the actual analysis was implemented with a statistical software package, NCSS version 6.0.21 (NCSS 2000).

The methods used for calculating correlation will now be discussed (see Hypothesis 6 in Section 7.3). Two common techniques for calculating the correlation between value pairs across two groups are; the Pearson Product-Moment Correlation Coefficient ('r') and the Coefficient of Determination ('r squared'). The scale of r takes values from +1 (a perfect positive correlation) to –1 (a perfect negative correlation). An r value of zero indicates a complete absence of correlation (Lowry 2000, chapter 7). The Coefficient of Determination ('r squared') was then calculated as r squared = r * r. Values for r squared are always positive, ranging from +1.0 to zero. This measurement indicates the strength of the correlation, but not its direction (positive or negative). The correlation assessments carried out for this research required the pairing of each RSL Informant's individual score for one specific question (A0.40 – see Appendix 5), with a value representing the same Informant's overall level of satisfaction with both of the software prototypes. To obtain the second of these values, a mean was calculated for each RSL informant of the question scores relating to the Informant's satisfaction with the Initial Prototype. Then the same was done for the Operational Prototype. Finally, one value (the mean of the two previously obtained values) was calculated for each RSL Informant, representing the Informant's general satisfaction with both prototypes.

7.5 Summary of methodology

- The Researcher built an Initial Prototype of the Procession software.

- The Researcher obtained data from an RSL concerning a live project. This information was then used to run simulated instances of the project.

- The Researcher used the basic live project data to fictionalise two instances of the project. These had different sequential differences and outcomes. One of the simulated projects was used with the Initial Prototype and the other with the Operational Prototype. Two simulated projects were required to ensure that the RSL Informants were not familiar with the project sequence.

- For both of the simulated projects, the Researcher used Monte Carlo Risk Analysis to fictionalise a series of chronological, quarterly milestone files. The changes made to these milestones illustrated alterations that, in a 'real' project, would be made by the Employer's Agent. The nineteen-month project produced seven individual data files.

- The Researcher identified five RSL Informants willing to participate in the evaluation of the software.

- The Researcher visited the offices of each volunteer RSL Informant and asked them to evaluate the Initial Prototype.

- Before the RSL Informant was shown the Initial Prototype, they were asked the questions from Section A of the Client Information Requirement Protocol. These concerned the RSL Informant's perception of the information provided by the Informant's current method of progress reporting. It also considered the RSL Informant's general familiarity with computers.

- The RSL Informant was shown an Earned Value Analysis s-curve for the entire scenario.

- For each of the seven milestones, the Researcher kept the 'estimated' and 'at completion' values covered with a blank sheet of paper. The sheet was progressively moved to the right, uncovering more and more of the s-curve and pausing at milestones one to three for the RSL Informant to consider current status.

- At milestone four, the subject was asked to describe the current status of the project, with reference to a chart of codes provided by the Researcher.

- The sheet was again progressively moved to the right, uncovering more and more of the s-curve and pausing at milestones five to seven.

- The RSL Informant was given a brief introduction to the Initial Prototype.

- The RSL Informant explored the 3D data surfaces produced by the Procession prototype, for milestones one to three.

- At milestone four, the RSL Informant was asked to select a phrase (from a list at the back of the questionnaire), which most closely described the Informant's impression of the project performance (as shown by Procession).

- The RSL Informant explored the remaining 3D data surfaces for milestones five to seven.

- Where appropriate, the Researcher made notes concerning the software usability. Section B of the Initial Prototype Usability and Information Evaluation Protocol provides space for this purpose.

- After the project scenario was complete, the Researcher applied Section A of the Initial Prototype Usability and Information Evaluation Protocol, relating to the RSL Informant's information requirements. This section considered the RSL Informant's general satisfaction with the prototype.

- The Researcher then asked software usability questions from Section B of the Initial Prototype Usability and Information Evaluation Protocol.

- When the Initial Prototype had been evaluated by all participating RSL Informants, the Researcher collated the responses provided in the Protocols. Comparison was made between each RSL Informant's current reporting method and the Initial Prototype, with specific reference to the perceived quality of information provided (Section A, Client Information Requirement Protocol and Section A, Initial Prototype Usability and Information Evaluation Protocol).

- In addition, the software usability responses (Section B, Initial Prototype Usability and Information Evaluation Protocol) were analysed. This informed the Operational Prototype development process.

- The Researcher developed an Operational Prototype, with reference to the RSL Informants' evaluation of the Initial Prototype.

- The Researcher re-visited the offices of each volunteer RSL and asked the relevant RSL Informants to evaluate the Operational Prototype.

- As distinct from the interview conducted for the Initial Prototype, with the Operational version there was no need to ask the RSL Informant questions from Section A of the Client Information Requirement Protocol. These concerned the RSL Informant's perception of the information provided by the Informant's current method of progress reporting and this has already been captured.

- The RSL Informant was shown an Earned Value Analysis s-curve for the entire scenario.

- For each of the seven milestones, the Researcher kept the 'estimated' and 'at completion' values covered with a blank sheet of paper. The sheet was progressively moved to the right, uncovering more and more of the s-curve and pausing at milestones one to three for the RSL Informant to consider current status.

- At milestone four, the subject was asked to describe the current status of the project, with reference to a chart of codes provided by the Researcher.

- The sheet was again progressively moved to the right, uncovering more and more of the s-curve and pausing at milestones five to seven.

- The RSL Informant was given a brief introduction to the Operational Prototype.

- The RSL Informant explored the 3D data surfaces produced by the Procession prototype, for milestones one to three.

- At milestone four, the RSL Informant was asked to select a phrase (from a list at the back of the questionnaire), which most closely described the Informant's impression of the project performance (as shown by Procession).

- The RSL Informant explored the remaining 3D data surfaces for milestones five to seven.

- Where appropriate, the Researcher made notes concerning the software usability. Section B of the Operational Prototype Usability and Information Evaluation Protocol provides space for this purpose.

- After the project scenario was complete, the Researcher applied Section A of the Operational Prototype Usability and Information Evaluation Protocol, relating to the RSL Informant's information requirements. This section considered the RSL Informant's general satisfaction with the prototype.

- The Researcher then asked software usability questions from Section B of the Operational Prototype Usability and Information Evaluation Protocol.

- As distinct from the first collation of responses, two types of comparison were then made.

- Firstly, between each RSL Informant's current reporting method and the prototype solutions, with specific reference to the perceived quality of information provided (Section A Client Information Requirement Protocol, Section A, Initial Prototype Usability and Information Evaluation Protocol and Section A, Operational Prototype Usability and Information Evaluation Protocol).

- Secondly, across prototype versions, in terms of informational requirements and software usability (Figure 7.1 Comparison 2, Section B, Initial Prototype Usability and Information Evaluation Protocol and Section B, Operational Prototype Usability and Information Evaluation Protocol).

- Statistical analysis was applied to the results. Question score means were calculated as required to test the hypotheses (see Section 7.3). The t-test for the significance of the difference between the means of two groups was used to evaluate hypotheses 1 to 5 and the Pearson Product-Moment Correlation Coefficient ('r')/Coefficient of Determination ('r squared') was used for hypothesis 6.

- Final results were compiled and written-up for final delivery.

7.6 Chapter summary and contextualisation

This chapter provided a detailed description of the methodology used to evaluate the software tool Procession. The Researcher first visited five RSL Informants with an initial prototype. Before showing each subject the prototype, a five-point scale interview was undertaken to the capture the subject's current reporting method and requirements. Next a fictional project scenario was shown to the RSL Informant in the form of a single 2D s-curve. The project was punctuated by seven quarterly milestones. The subject was asked to assess the project status at milestone four. The accuracy of this assessment was later scored on a five-point scale. After completion, the same scenario was repeated using Procession at each milestone. At milestone four, the subject was again asked to assess project status. Immediately after the second run through the scenario, questions were asked about the information and usability of the software.

The results of this first visit were collated and the outcomes applied to the development of a second prototype. On a second visit to each RSL Informant, a different fictional scenario was used to evaluate the new prototype. Also in this chapter, a series of null hypotheses were proposed to test whether 3D Information Visualisation can provide construction clients with informative performance reports. The null hypotheses were as follows:

- Can RSLs assess project status more accurately with software (both prototypes) than with the application of traditional methods?
- Is the second software prototype easier to use than the first prototype?
- Is the quality of the information perceived to be better in the second prototype than the first prototype?
- Is the relevance of the information provided perceived to be better in the second prototype than the first prototype?
- Is the format of the report perceived to be better in the second prototype than the first prototype?
- Is there a negative or positive correlation between users' satisfaction with the software and the technical complexity of the method used to provide their project progress reports?

A breakdown of the techniques used to calculate the statistical relevance of the results was then presented. The construction programme from a real project was required to evaluate the software. In the next chapter, the process of applying this concept to an actual 'live' project is documented.

Chapter 8: The 'live' construction project data set

8.1 Chapter Introduction

The program schedule data was obtained for a 'live' construction project. The Researcher used this data as a skeleton to create simulated project scenarios, with different sequences and outcomes. This chapter records this process. The output from this stage of the research was the 2D s-curves and 3D data surface files used to evaluate Procession with the RSL Informants.

One of the claims for a 'contribution to knowledge' made for this research relates to the evaluation protocol. The implementation of this protocol, as described in Chapter 7, requires data from a real construction project. This chapter describes how the obtained data was transformed into both traditional and experimental performance reporting formats, allowing RSL Informants to evaluate the software by comparing the two sets of reports.

8.2 The 'live' project

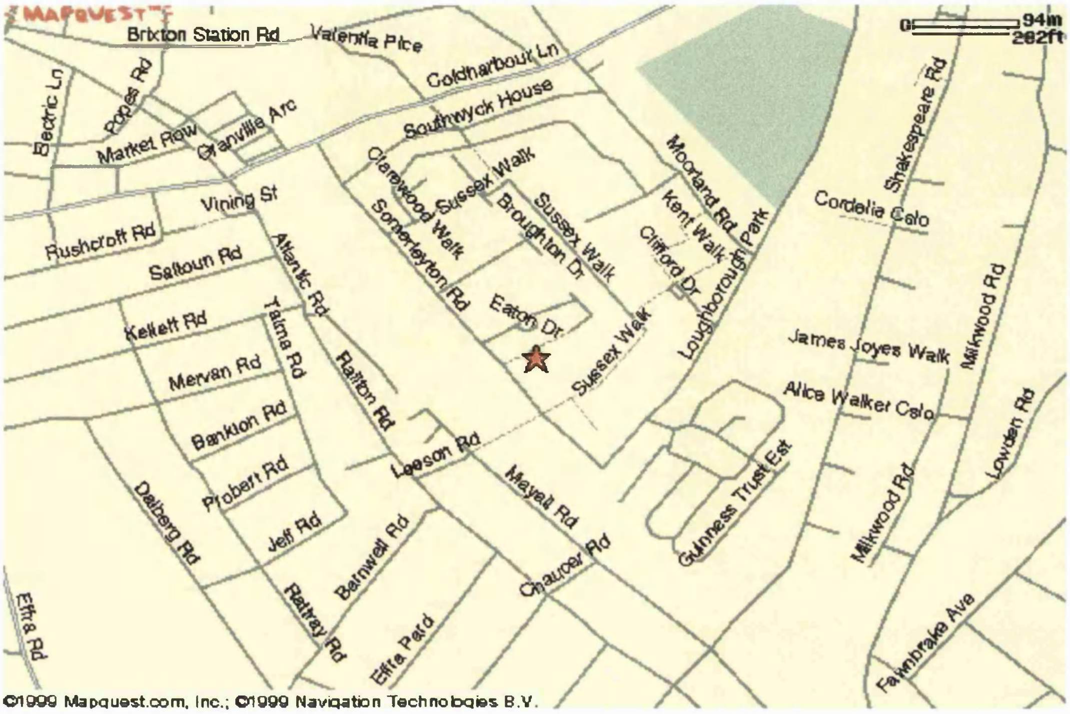


Figure 8.1 Location of Moorlands Estate project in South London, source: Mapquest.com/Navigation Technologies B.V..

This project focuses on the refurbishment of the Moorlands Estate in Brixton, South London (see Figure 8.1). The estate was transferred from the London Borough of Lambeth to Metropolitan Housing Trust, under the ERCF (Estate Renewal Challenge Fund) in the late 1990s. Metropolitan are the client for the current refurbishment. The architects are HTA Limited of London and the principal contractor is Mansell plc (www.mansell.plc.uk). Project management services are being provided by United House Construction Ltd (www.united-house.co.uk), who are based in Kent.

The Moorland Estate project objectives were; to provide new and clearly defined street patterns; to enlarge the individual gardens and to remove crime-ridden pedestrian routes. Following a vote by tenants in the late 1990s, management of the estate was transferred from the local authority to a Housing Association, under the UK government's ERCF (Estate Renewal Challenge Fund). The project represents one phase of the continuing ERCF funded improvements. The total estate consists of 517 housing units (individual houses and flats). The 'live' project focuses on 252 of these units (123 houses and 129 flats). Houses are collected into groups of 10, 8, 7, 6, 5 and 4 houses. Blocks of flats contain 15, 14, 10 or 9 flats. Work on the houses will be conducted with most of the tenants in situ. A unique aspect of the project is the 'porch turnarounds' on the houses. This entails making the back of the house into a new entrance porch. The houses are given enlarged back gardens by gains made from old pedestrian ways. The new front gardens have parking spaces. Flats receive basic internal refurbishment. For example; kitchen fittings, vinyl kitchen flooring and decorations. Extensive landscaping is being conducted on the estate's communal areas. This include turf-laying, tree-planting, re-routing of gas mains and drains, new street patterns and improved estate lighting.

8.3 Transferring data from Power Project to Microsoft Project

Unfortunately, the original 'live' project file was only available in the Power Project 5.0 format, another popular management tool. This file was exported to MPX (Microsoft Project eXchange) 4.0. This is an ASCII, record-based text format used to transfer files between Microsoft Project and other programs that support MPX. Files saved in this format have an '.mpx' extension. There were concerns that Power Project does not support every field available in the MPX file format. This could have meant the loss or modification of some project information. However, no problems were detected with the fields required for this research. After importing the MPX file into Microsoft Project 98, it was noted that the tasks in the original file had no hierarchy. To resolve this, 'summary tasks' were added (see Figure 8.2). Summary Tasks is a term used to describe a high level collection of 'sub-tasks'. For example, 'plumbing' might be the summary task of the sub-tasks 'install sink' and 'install shower'. The original 'live' project durations, dependencies and schedule were left unchanged.

8.4 Assigning costs and producing a baseline

In project management terminology, 'resources' are the materials and labour costs assigned to a specific task. Resources had not been assigned to the tasks in the original Power Project file and so assumptions were made about example resources that could be applied (i.e. the sub-tasks, materials and labour required to achieve the stated tasks). As the durations were pre-determined, resources based on hourly paid rates were not applicable. Instead, the labour and material costs were obtained from an industry standard pricing book (Davis et al. 2000) and calculated on a per house, block or site basis. In the initial version of the project budget, sums were applied as 'fixed cost' resource elements. Testing revealed that Earned Value (see Section 4.3) fields required for Procession seemed not to be functioning as expected when this approach was adopted (i.e. BCWS and BCWP values were not present). Instead, a new baseline file was created with the 'cost' values cut and paste as 'fixed costs' directly attached to tasks (i.e. there were now no resources). As Procession obtains its performance data from task fields, 'fixed costs' have no effect on the data surfaces generated.

Allocation of resources to tasks was done as units, rather than as a percentage. Number of available resource units was set to an arbitrarily high figure (10,000), to make sure that sufficient were available. Each 'fixed cost' was set to 'accrue at prorated'. 'Actual Costs' were calculated automatically by Microsoft Project. A 'Baseline' was saved. The baseline provides a record of all budget values before project commencement. These are then used for later analysis during the project lifecycle. Costs for the 'live' project were estimated from a construction industry standard pricing book (Davis et al.

2000). The total budgeted cost before the project started was estimated as £6,247,849 (see Figure 8.4).

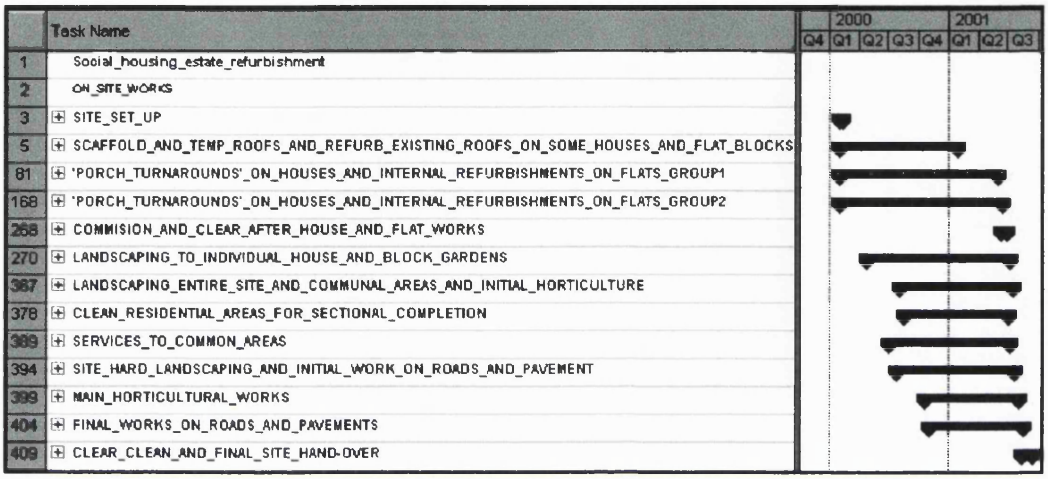


Figure 8.2 'Live' project summary tasks and their scheduled durations before commencement, source: author.

All prices exclude VAT, include labour and materials but exclude profit.

	spon's page	spon's ref	unit type	per unit
SITE_SET_UP				
Temporary works	p.110	A44	0.5% all	£31,960.14
SCAFFOLD_TEMP'_ROOFS_AND_REFURB'_EXISTING_ROOFS_&ELEV'NS				
scaffolding (0.8% of total cost blocks and houses)	p.109	A44	unit	£352.12
temp roof covering	p.110	A44	m2	£3.25
refurb ex. Roof and elevations				
softwood trussed pitched roof structure - 35 degree pitch	p.710	2.3.11	m2	£15.50
concrete interlocking tile coverings	p.710	2.3.18	m2	£68.50
Dormer 'mansard' softwood structure	p.710	2.3.16	each	£421.00
TURN_AROUND'_WORKS_AND_INTERNAL_REFURBISHMENTS				
Turnarounds				
construct brick and glass entrance porch				
Dormer 'mansard' softwood structure	p.170	2.3.16	each	£421.00
zinc pitched roof	p.230	H74	m2	£46.41
0.80 mm thick (23 swg) dormer covering				
sloping 10-50 degrees				
Composite cavity wall:facing brick outer skin;50mm insulation				
and plasterboard and emulsion	p.736	2.5.44	m2	£58.00
2 x double glazed, softwood standard windows	p.742	2.6.2	m2	£231.50
external door and hardware	p.744	2.6.7	each	£617.50
Internals				
New kitchen fittings (all units)	p.772	4.1.2	each	£1,077.50
New kitchen floor covering (vinyl tiling including screed and skirtings)	p.762	3.2.95	m2	£20.60
Decoration (emulsion two coats)	p.770	3.4.1	m2	£1.60
Decoration (gloss primer and two coats)	p.770	3.4.6	m2	£3.20
COMMISION_AND_CLEAR				
25% of (0.2% of £4,181,4793 = £8,362.95 removal of rubbish and all clearing)	p.106	A42	m2	£0.29
LANDSCAPING_TO_'INDIVIDUAL_GARDENS'				
Houses				
Car parking surface over back garden	p.798	6.1.68	each	£950.00
Both Size for parking space 21m2 p.869				
garden walls-one brick thick-1.8m high	p.800	6.1.137	m	£189.00
COMPLETE_LANDSCAPING_FOR_SECTIONAL_COMPLETION				
turfed area (supply,plant and 12 months guar.)	p.796	6.1.16	m2	£4.94
SERVICES_TO_COMMON_AREAS				
Site drainage	p.802	6.3.1	m2	£9.00
External lighting	p.801	6.2.5	m2	£2.25
Gas main (including trenches and excavation)	p.801	6.2.3	m	£35.00
ROAD_BASE_PAVEMENT_AND_HARD_LANDSCAPING				
50% of 200m roads and barriers (two-lane road 7.3m wide urban)	p.799	6.1.85	m	£1,540.50
individual street lights	p.800	6.1.114	each	£199.00
HORTICULTURAL_WORK				
turfed area (supply,plant and 12 months guar.)	p.796	6.1.16	m2	£4.94
shrubbed area including allowance for small trees	p.796	6.1.23	m2	£42.13
ROAD_WEARING_COURSE				
50% of 200m roads and barriers (two-lane road 7.3m wide urban)	p.799	6.1.85	m	£1,540.50
CLEAR_CLEAN_AND_H.O._SITE				
75% of (0.2% of total cost removal of rubbish and all clearing)	p.106	A42	m2	£0.29

Figure 8.3 Pricing the 'live' project, source: author.

myProject baseline budget at summary task level	
SITE_SET_UP	
SCAFFOLD_AND_TEMP_ROOFS_AND_REFURB_EXISTING_ROOFS_ON_SOME_HOUSES_AND_FLAT_BLOCKS	£20,907
'PORCH_TURNAROUNDS'_ON_HOUSES_AND_INTERNAL_REFURBISHMENTS_ON_FLATS_GROUP1	£2,585,299
'PORCH_TURNAROUNDS'_ON_HOUSES_AND_INTERNAL_REFURBISHMENTS_ON_FLATS_GROUP2	£412,588
COMMISION_AND_CLEAR_AFTER_HOUSE_AND_FLAT_WORKS	£532,999
LANDSCAPING_TO_INDIVIDUAL_HOUSE_AND_BLOCK_GARDENS	£2,091
LANDSCAPING_ENTIRE_SITE_AND_COMMUNAL_AREAS_AND_INITIAL_HORTICULTURE	£1,260,290
CLEAN_RESIDENTIAL_AREAS_FOR_SECTIONAL_COMPLETION	£94,535
SERVICES_TO_COMMON_AREAS	£2,509
SITE_HARD_LANDSCAPING_AND_INITIAL_WORK_ON_ROADS_AND_PAVEMENT	£916,083
MAIN_HORTICULTURAL_WORKS	£133,190
FINAL_WORKS_ON_ROADS_AND_PAVEMENTS	£98,316
CLEAR_CLEAN_AND_FINAL_SITE_HAND-OVER	£184,860
	£4,181
	£6,247,849

Figure 8.4 'Live' project summary task budgets before commencement, source: author.

8.5 Generating scenarios with Monte-Carlo risk analysis

In order to generate possible scenarios from the data set, risk analysis was applied to the Microsoft Project file. The Microsoft Project compatible software tool Riskman Professional (RiskMan 2000) was used to identify potential project outcomes. This was undertaken in two stages; identification (risk detection, classification, and description) and evaluation (quantifying the probability of risk occurrence and the resulting cost impact). Riskman can produce simulations from input task uncertainties, costs and overruns, providing an evaluation of probable costs and end dates. The software requires that an Evaluation Mode is selected and, in this case, the Symbolic Evaluation setting was chosen. This provides a way to quickly assess risks, then to assign priorities to them using qualitative terms. The developers of Riskman suggest that this mode of evaluation is most appropriate in the early stages of a project launch, when information is still scarce. Next, Riskman requires the user to evaluate the probable occurrence of each individual risk. The probability of a risk should logically be deduced from the probability of the risk's causes occurring. However, the dependence between causes being unknown, it is also necessary to evaluate the probability of the risk. It is also necessary to define risk impacts. These are the estimated losses (in terms of budget and schedule) in the event of a risk occurring. Riskman evaluates each task's level of risk exposure according to the probabilities and impacts assigned by the user.

Monte-Carlo simulations can be thought of as "statistical simulation methods, where statistical simulation is defined in quite general terms to be any method that utilizes sequences of random numbers to perform the simulation" (CSEP 1995, p.1). First described during World War II's Manhattan Project, Monte-Carlo simulations are now widely used, both in hard science and for predicting games of chance (CSEP 1995, p.1). Riskman uses Monte-Carlo simulations to analyse the likelihood and impact of specified risks.

A project consists of tasks, related by precedence constraints. In a simplified model, each task has a duration and a set of allocated resources. In practice, it is difficult to get accurate information about task durations. This suggests that probabilistic modelling of the assessment uncertainties may be appropriate, using a range of possible values together with an associated probability law. Duration information about the critical path of the project is gathered from the Monte-Carlo simulation, i.e. from computing the duration of the critical path for a random set of possible task duration values, sampled according to their probability laws (planning on a sample-by-sample basis). This results in a set of simulation samples. Riskman allows the user to identify the potential risks present, after running the Monte-Carlo simulations.

Risks are discrete events with a given probability of occurrence and a cost/delay impact. Identified risks are included in the simulation, so that their impact on task scheduling is taken into account. This is done as follows. A random selection of risks is added to each task. Each risk may or may not occur. If it occurs, its delay impact is added to the duration of the impacted tasks. Each time a task is impacted, its duration increases. If resources were allocated to the task, they work on at an unaffected rate and the cost rises. If no resource was allocated to it, its duration will still increase but no extra cost is generated. During each cycle of the simulation, tasks are rescheduled and then the costs of impacts and avoiding actions are added together and divided by the risks calculated in this cycle. This is added to the total cost of the project. At the end of each simulation cycle, each task is noted as on or outside the critical path. After all cycles are completed, Riskman calculates each task's criticality to a successful project outcome. After the completion of the Monte-Carlo cycles, Riskman outputs its results as Microsoft Excel charts and in its customised Microsoft Project 'Simulation View'.

8.6 Results of scenario generation

The Monte-Carlo simulations revealed two summary tasks which were categorised as having attached risks of an 'unacceptable' level. The first of these was the summary task 'scaffolding and temporary roofs and refurbish existing roofs on some houses and flat blocks' which had the following unacceptable risks associated with it. Please note that the Researcher did not write the descriptions of these risks. They are from RiskMan's own library. The risks for the first task were:

- Constraints on availability of external scaffolding components, impact the whole product.
- Requirements provided to scaffolding subcontractors are ambiguous.
- Sub-contracted scaffolding task disturbs project progress.
- Sub-contracted scaffolding task is badly managed and followed up.
- Workloads were under-estimated to gain contract.
- Required security level for the project is higher than usual because some tenants are remaining in occupation. Organisation of key access to occupied properties becomes expensive and time-consuming.
- Project is innovating in its field (porch turnarounds- see Appendix 12).

The second identified summary task was 'site hard landscaping and initial work on roads and pavements' (see Figures 8.3 and 8.4), which had the following unacceptable risks associated with it-

- Feasibility study on impact of road and service re-routing is missing.
- There is a low awareness of possible technical problems.
- When estimating costs, some activities are not quantifiable or have no reference in archived projects.
- Project risks were not assessed or are difficult to assess.
- Economic and political context has an influence on project and can be changed by unilateral decision at a high level. This particular project has to take particular regard of tenant opinion.
- Project environment is evolving.
- Technical difficulties are a real challenge.
- Detailed assessments of cost and time scale are not compatible with project budget and schedule. Planning does not take into account this fact and relies on false hypotheses.

The two unacceptable risks were fictionalised into scenarios (with embedded 'problematic' issues). In scenario 1, the scaffolding sub-contractor failed to remove house scaffolding according to schedule, extending the duration of the roofing tasks and delaying the commencement of the porch turnaround work by almost three months. Scenario 2 sees the estate vehicle access restricted and most of the tenants remaining in residence. When the initial work on roads and pavements began in the third quarter, the situation became unmanageable. All works relating to the communal estate areas (roads, pavements, hard landscaping and horticulture) started to slip and their durations extended.

8.7 Traditional performance measurements (s-curves)

As discussed in the last section, traditional performance measurement techniques were used to analyse the entire sequence. A series of two-dimensional s-curves were produced (see Chapter 4). BAC (Budget At Completion), EAC (Estimate At Completion) and ETC (Estimate To Complete) were plotted against each other up to milestone 7, as a cost/time s-curve. It should be noted that the chosen s-curves have a 3D appearance. In fact, this is a 3D view of a 2D chart. It was chosen because it aided the intelligibility of the s-curve. Although the s-curves appear 3D, they are really 2D plot lines stacked on top of each other. The number of dimensions is still 2 (time x performance/cost) as opposed to Procession's 3 (deviation parameter x task x deviation level).

The expected total cost of the project (before it started) was £6,247,849. By the end of the planned project time-scale for scenario one, £6,169,170.78 had been spent and in order to complete (behind schedule), an additional £14,533,790.31 was required. The total project cost would then have spiralled to £20,781,638.82 (see Figure 8.5). In scenario 2, 100% of the £6,247,849 predicted costs had been spent by the end of the planned project time-scale and in order to complete (behind schedule), an additional £56,751,522.90 was required. The total project cost would then have spiralled to £62,920,693.68 (see Figure 8.6).

CPI (Cost Performance Analysis) was plotted against SPI (Schedule Performance Analysis) as an s-curve to identify trends within the project progress. In Completion Cost Analysis (Section 4.4), a flat line indicates that the baseline is being achieved on a £ for £ basis. Values below £1, suggest that the project is either over-budget or behind schedule.

For scenario 1, the CPI and SPI showed that both the schedule and the budget trends were deteriorating rapidly before the first milestone. In the second quarter, both budget and schedule stabilised. By the third milestone, a slight improvement was seen in both curves, however they quickly dipped back and continued at approximately £0.30 for the rest of the schedule. At completion, the project was both behind schedule and over budget (see Figure 8.7).

Scenario 2 saw both the schedule and the budget plans deteriorating rapidly before the first milestone. In the second quarter, the budget stabilised, while the schedule continued to slip. By the third milestone, the schedule was stabilising and it continued to mirror the budget curve, with a differential of approximately £0.20. This continued until the scheduled completion date, with only a slight budget recovery in the last quarter. At completion, the project was still very behind schedule and over budget. The final CPI value reveals that £0.10 of value was being earned for every £1 spent (see Figure 8.8).

BCWS (Budgeted Cost of Work Scheduled) was plotted against BCWP (Budgeted Cost of Work Performed) and ACWP (Actual Cost of Work Performed) to produce an s-curve of Earned Value (see Section 4.3). For scenario 1, expenditure stayed almost as expected (ACWP and BCWS tracked each other throughout) but earned value (BCWP) slowly deteriorated throughout the sequence. Therefore, the work completed by the planned completion date had cost far more than budgeted and the schedule had proven to be over optimistic (see Figure 8.9). Throughout scenario 2, ACWP almost followed the budget in the baseline schedule, but increased in task durations. This resulted in the scheduled expenditure (BCWS) levelling out to match a longer time-scale. Despite slow improvement through the sequence, the earned value (BCWP) was so low compared with the ACWP, that the situation became irretrievable (see Figure 8.10). As previously stated, RSL Informants were asked to assess project performance at Milestone 4, using Earned Value Analysis. Status codes have been given to the project at Milestone 4, to enable questionnaire scoring.

At Milestone 4 scenario 1 was:
On budget behind schedule- schedule performance deteriorating and cost performance stabilised with a status code of 4.6 (see last pages of Appendix 6, Initial Prototype Usability and Information Evaluation Protocol).

The sequence outcome for scenario 1 was:
Project had run out of time and budget to complete. No cost overruns had been incurred to date but more funds would be required to continue.

At Milestone 4 scenario 2 was:
Over budget behind schedule- schedule performance deteriorating and cost performance deteriorating with a status code of 7.5 (see last pages of Appendix 6, Initial Prototype Usability and Information Evaluation Protocol).

The sequence outcome for scenario 2 was:
Project had run out of time and budget to complete. No cost overruns had been incurred to date but funds totalling many times the original budget would be required to continue.

Finally, two sets of seven milestone Microsoft Project files were generated, one for each scenario. To simulate progress at each milestone, changes were made to the tasks, in terms of their durations and percentages complete. For each scenario, it was necessary to work through the milestone files chronologically. This is because Microsoft Project always passes revised estimates onto the next milestone in the sequence. CSV data files for Procession were exported from each of the Microsoft Project milestone files (i.e. 14 files total).

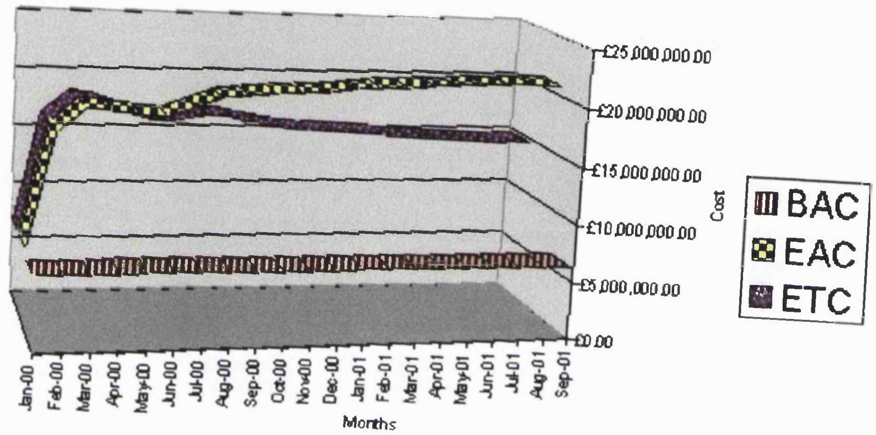


Figure 8.5 Scenario 1: At completion s-curve for entire project, source: author.

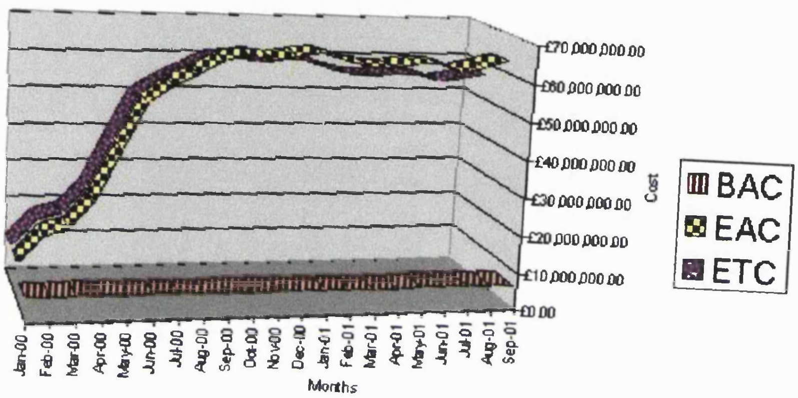


Figure 8.6 Scenario 2: At completion s-curve for entire project, source: author.

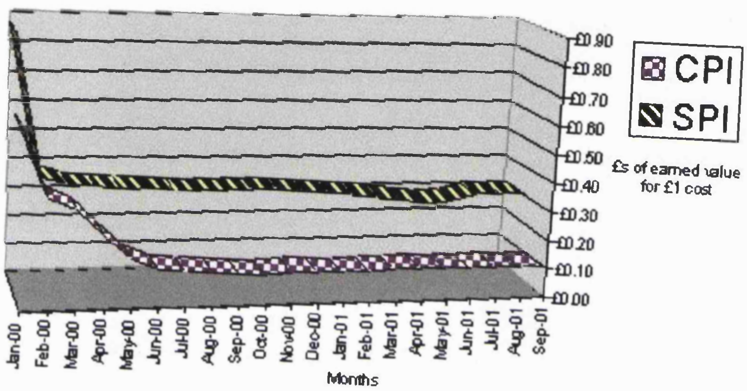


Figure 8.7 Scenario 1: Cost and Schedule indices s-curve for entire project, source: author.

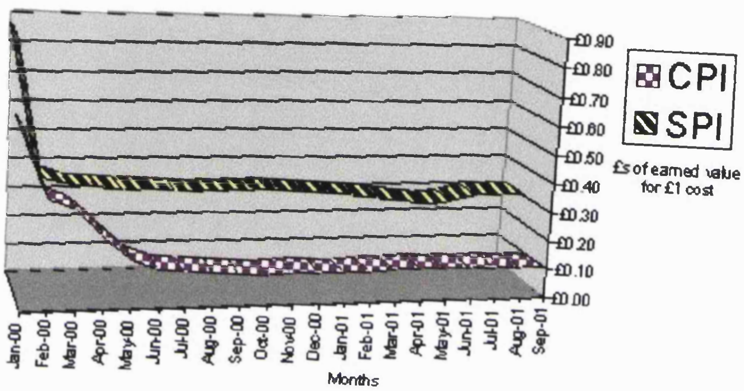


Figure 8.8 Scenario 2: Cost and Schedule indices s-curve for entire project, source: author.

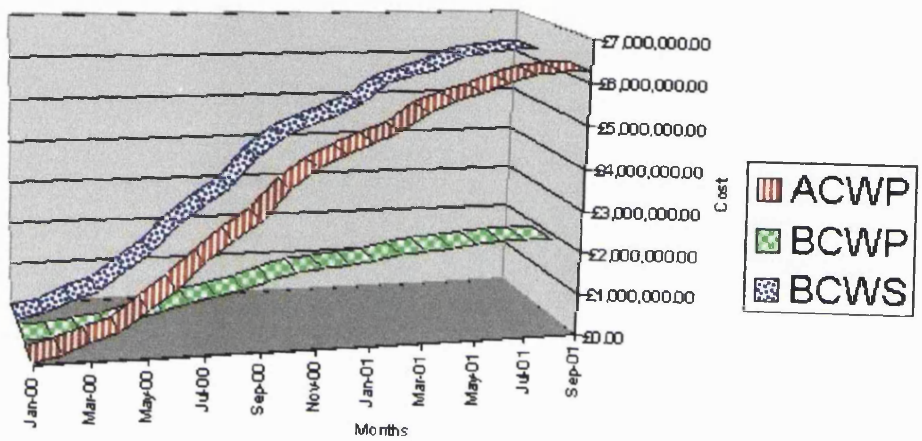


Figure 8.9 Scenario 1: Earned Value s-curve for entire project, source: author.

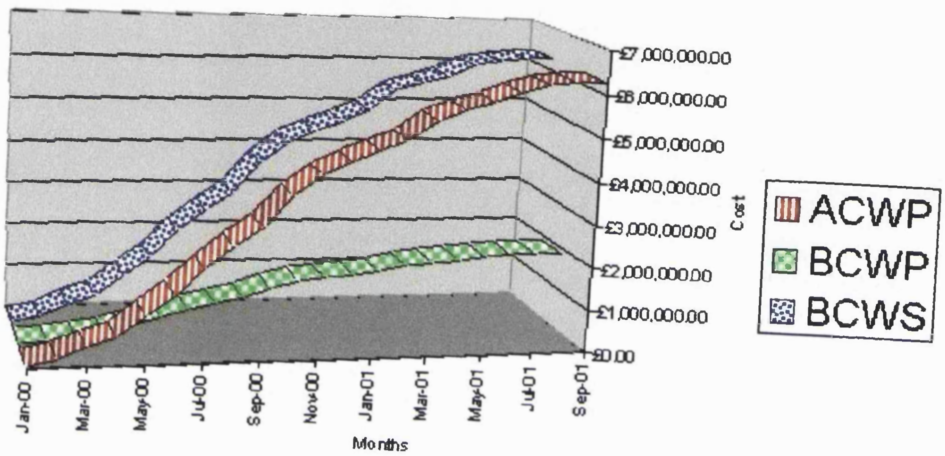


Figure 8.10 Scenario 2: Earned Value s-curve for entire project, source: author.

8.8 Comparative milestones as shown to RSL Informants: s-curves vs. data surfaces

The two figures discussed and presented in this section might be considered the crux of evaluating this research. Comparing the two representations of the data at milestone 4, allowed the Researcher to test the question posed at the start of this thesis: "can 3D Information Visualisation provide construction clients with informative performance reports"?

It is important to remember that these milestone reports were not actually presented to the RSL Informants in the paired format shown in figures 8.11 and 8.12. The RSL Informants did not see both the s-curve and the data surface simultaneously. For each scenario/visit, the s-curves were shown first and then all of the data surfaces. The pairs are shown here in this format only to aid reader comprehension of the comparisons that were made. For example, see Chapter 7, hypothesis 1 which uses the mean questionnaire scores to compare an RSL Informant's ability to assess current status. Effectively, it is each Informant's response to the contrasting pairs shown in Figures 8.11 and 8.12 that determines whether they correctly identified project status at milestone 4. This identification was achieved by informants' selection of the status codes at the end of each prototype questionnaire (see last section and Appendix 9).

For each pair shown, the s-curve is a two-month segment from the entire project Earned Value s-curves from the last section (Figures 8.9 and 8.10). Rather than covering a temporal period, the data surfaces reflect project status at a time 'T'. Therefore, it should be noted that, for each pair, the real comparison is being made between the data surface and the s-curve at its right hand month boundary. It should also be re-stated that the s-curves are showing *overall project status*, whereas the data surfaces present the *performance of individual tasks* (here only at the summary task level). There may be corresponding high or low peaks between the two representations. However, it is also possible for the data surface to peak on an individual project task, with no obvious correlation on the entire project s-curve. It is important to remember that the two visual methods are not directly comparable. Procession only reports exception to the project's baseline cost and time.

Presenting Figures 8.11 and 8.12 in a paired format (s-curve vs. data surface) is not intended to assist the reader in understanding the *data* presented. Rather, it is to aid the reader's appreciation of the differing visual structures that formed the basis for comparison. By definition, 3D data surfaces become more intelligible when a user exploits the added dimension by navigating, examining, rotating etc. For example, text that is difficult to read from one angle may become clearer when zoomed. Therefore, when viewing the 3D data surface element of each pair (Figures 8.11 and 8.12), it may prove difficult to appreciate the intelligibility that was added to the data. Ultimately, any improvement was best reflected by user satisfaction. However, the paired visuals for each scenario are presented here (at milestone 4), in order to compliment the reader's overall grasp of the chosen methodology. In Figures 8.11, the colour scheme of the 3D data surface has been reversed to enhance the limited resolution possible with a screenshot.

At this point, it should be noted that the Earned Value s-curves for the entire project (Figures 8.9 and 8.10), do not show the actual project end. Both of the curves end at milestone seven, the baseline project end date. As there is slippage in both scenarios, complete s-curves would extend well beyond the base-lined timescale.

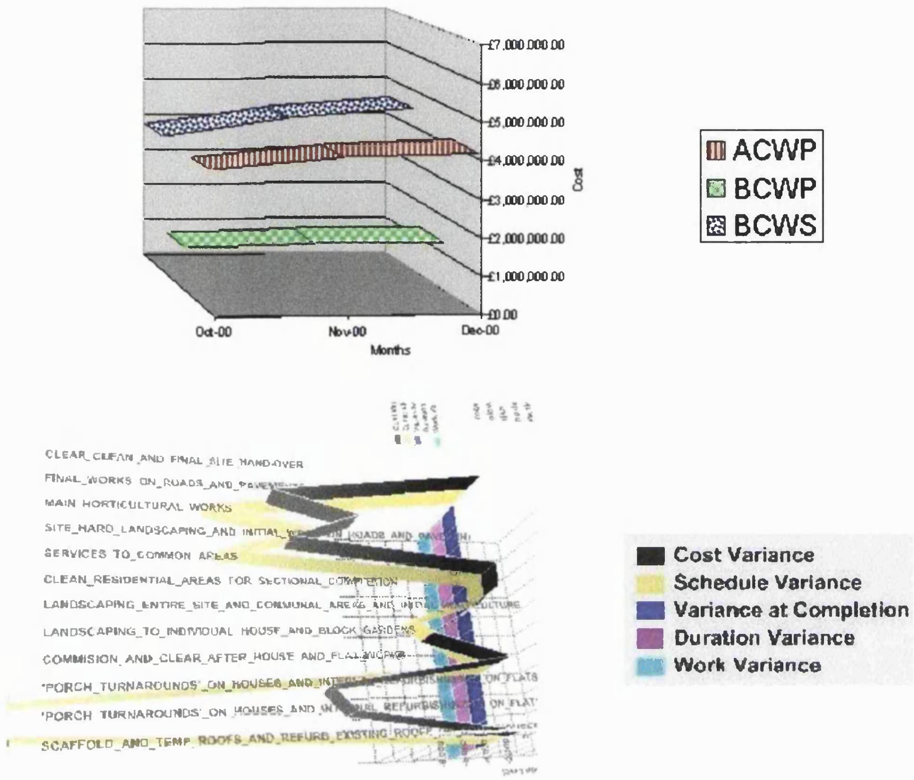


Figure 8.11 Scenario 1: Earned Value s-curve at milestone 4 (top) as compared to Procession Initial Prototype v1.1 data surface (bottom), source: author.

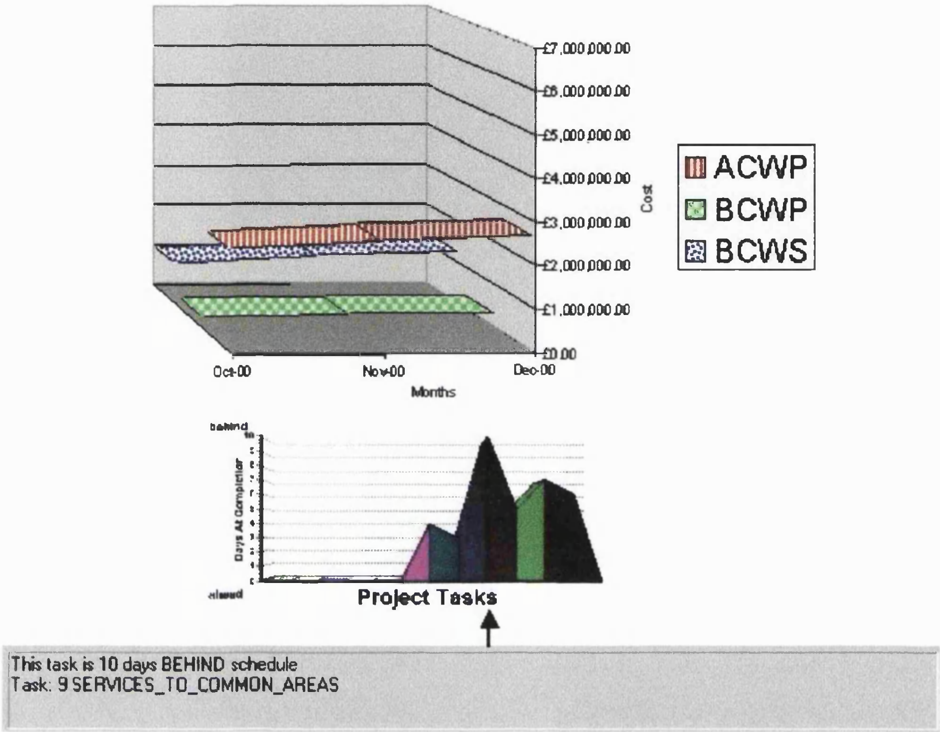


Figure 8.12 Scenario 2: Earned Value s-curve at milestone 4 (top) as compared to Procession Operational Prototype v1.2 data surface (bottom) - surface has been 'mouse-clicked' at arrow, source: author.

8.9 Chapter summary and contextualisation

This chapter has taken the theoretical evaluation methodology outlined in Chapter 7 and populated it with 'live' data for the two scenarios. In this case, preparing the files for Procession was over-complicated by the data not being in the Microsoft Project format. In addition, only basic schedule information was available and estimated costings had to be added. Finaly, Monte-Carlo simulations were used to produce scenarios that seemed statistically likely to result from the data. Traditional s-curves and Procession data files were produced for each scenario.

The completion of the work in this chapter represented the last stage before the evaluation of Procession. The Researcher now had an initial software prototype (Procession v1.1) and two fictionalised project scenarios with related 2D s-curves and 3D data surface files. The next chapter provides the results of the evaluation process, for both prototypes.

Chapter 9: Results

9.1 Introduction

This chapter presents and discusses the results of the evaluation carried out on the software tool Procession. Chapter 7 detailed the process by which this evaluation was conducted. In Chapter 3, a prototyping methodology was proposed for capturing the requirements of RSL Informants. This methodology included a revision of the initial prototype, after the first visit to the RSL Informants. Appendix 10 records the RSL Informants' feedback to the initial prototype. This includes details of the technical changes made, which allowed the development of a second operational prototype.

Section 9.2 presents all of the significant results in a tabulated form. This includes both the results of the t-tests for paired means and the correlation test. Explanation of relevant table fields is provided. In Section 9.3, the results and their implications are discussed. The final section reassesses the hypotheses and explains how each of the analysed values was obtained from the questionnaire scores.

9.2 Tabulated Results

Figure 9.1 presents a summary of those null hypotheses, which were rejected by t-tests with a significance of 95%. The statistical methodology is more fully described in Section 7.4. However, to aid interpretation of Figure 9.1, it should be noted that the t-statistic represents the difference between the *estimated* standard deviation (the standard error) for the two group means and the *actual* difference between the group means, when one is subtracted from the other. This indicates whether the difference between the two group means is statistically significant, or within the realms of chance.

The numbers in the 'null hypothesis' line correspond to the hypothesis numbers used in Chapter 7 and later in this chapter. Figure 9.1's 'report source' line indicates which of the three project reporting sources (traditional s-curve, software prototype one or prototype two) provided the data for each mean. The entry 'p1 & p2' in the 'report source' line shows that this mean has been derived from question responses for both prototypes. The line 'tabulated critical t value' is from the standard 'Critical value of t' tables. For these tests the value was determined by software and then manually checked against the table. This tabulated figure represents the threshold t value between significance and insignificance. Having derived a t-statistic for each mean comparison, this is checked against the 'tabulated critical t value'. These t-tests are 'left-tailed'. That is to say that the null hypotheses not only state that the means are the same, they also specify that one mean will be less than the other.

The values in the line 'degrees of freedom' are an index of the amount of random variability, mere chance coincidence that can be present in each of the sample populations. The number of degrees is calculated by counting the number of scores contributing to one of the means and deducting one. In order to look up a critical value of t in the standard table, it is necessary to know three things; the level of significance required (i.e. 0.05), the number of degrees freedom available and whether the t-test is directional (left or right tailed) or non-directional. The 'result' line in Figure 9.1 indicates whether the null hypothesis has been accepted or rejected. It should be noted that all of the hypotheses in Figure 9.1 have been rejected.

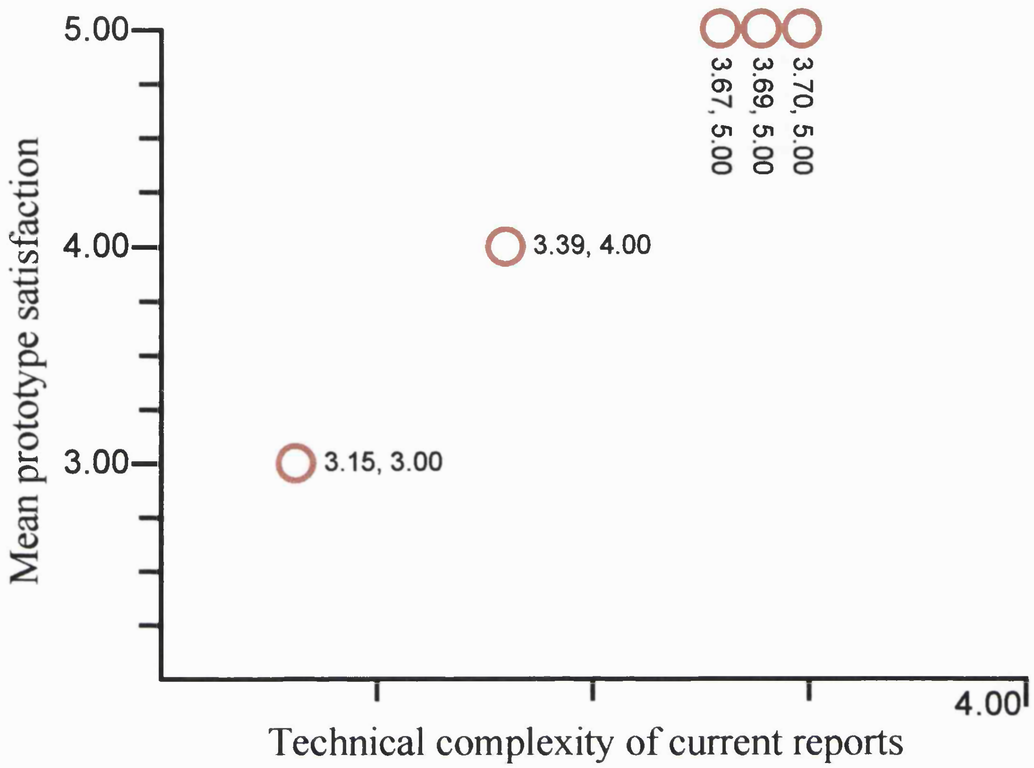
Figure 9.2 tabulates the results of the non-directional correlation test carried out on hypothesis 6 (see Chapter 7). When viewed in conjunction with the raw data in Figure 9.3, it becomes apparent that a positive correlation was identified ($r = 0.99$). Furthermore, the null hypothesis was rejected with a significance level of 95%. This was established by looking up the 'significant r for alpha 5%' with a sample size (N) of five in the standard table for non-directional correlation. Any observed value greater than 0.88 is considered significant for $N = 5$.

null hypothesis	1. Ability to assess status is same or worse		2. Usability is same or worse		3. Report quality perceived same or worse		4. Report relevance perceived same or worse		5. Report format perceived same or worse	
report source (p=prototype)	s-curve	p1 & p2	p1	p2	p1	p2	p1	p2	p1	p2
mean	2.7	3.5	3.28	3.81	3	4	3.2	3.8	2.8	4
tabulated critical t value	1.83		1.74		2.13		2.13		2.13	
degrees of freedom	9		17		4		4		4	
t-statistic	-2.23		-6.13		-3.16		-2.45		-3.21	
result	reject		reject		reject		reject		reject	

Figure 9.1 Results of left-tailed t-tests showing null hypotheses rejected with a significance level of 95%, source: author.

null hypothesis	6. No correlation between satisfaction with prototypes and technical complexity of current method
r	0.99
r^2	0.98
N (sample size)	5
significant r for alpha 5%	>0.88
result	reject

Figure 9.2 Results of non-directional correlation test showing null hypothesis rejected with a significance level of 95%, source: author.



$r = 0.99$

$r^2 = 0.98$

result = positive correlation

Figure 9.3 Correlation scatterplot for hypothesis 6, source: author.

9.3 Results discussed

The results presented earlier in this chapter will now be discussed and summarised. It is interesting to consider the directions of comparison that were eventually applied to the questionnaire scores. Looking back to the evaluation diagram Figure 7.1, the basic comparisons are indicated. The original reporting method is shown being compared to the software prototypes. The prototypes are shown being compared with each other. It is useful to consider which of these comparisons were linked to each hypothesis. Hypothesis 1 (see Figure 9.1) compares both prototypes with the traditional reporting methods. Hypotheses 2 to 5 only compare the prototypes with each other. The implication of this is that these hypotheses are about evaluating the prototyping methodology proposed in Chapter 3, rather than contrasting 'traditional' against 'new' reporting types. Hypothesis 6 contrasts satisfaction with both prototypes against the technical complexity of the subject's current reporting method. This does not reveal much about improving report quality. Hypothesis 6 may provide insight to the rationale behind individual RSL Informant's responses. The confirmation of a positive correlation for hypothesis 6 can be interpreted in a variety of ways. Are technically experienced users more able to appreciate the potential of Procession? Are they more able to assess its failings because they are not distracted by the technology? These questions are beyond the scope of this research but they could prove interesting for future work.

The null hypothesis that RSL Informants' mean ability to assess project status is the same or worse with both software prototypes than with an Earned Value Analysis s-curve, was rejected with a significance of 95% (see Hypothesis 1, Figure 9.1). It is concluded from this that software tools using 3D Information Visualisation provide a viable solution for the delivery of project performance reports to RSL Informants (see Figure 9.4). It could be argued that this result might be distorted by RSL Informants' unfamiliarity with s-curves and Earned Value Analysis. To counter this, it should be noted that all participating RSL Informants were provided with a basic introduction to the principles of these performance measurement techniques (see Appendix 4).

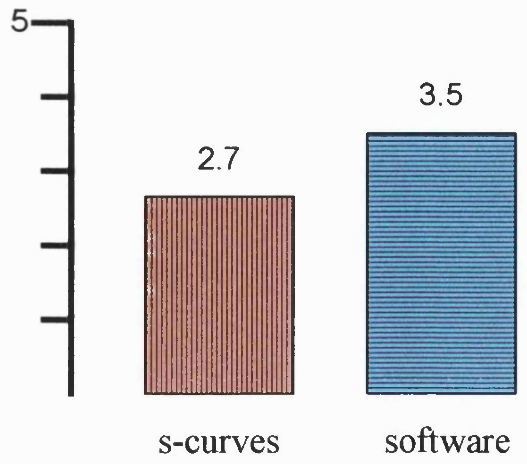


Figure 9.4 Results for RSL Informants' ability to assess project status, with a significance level of 95%, source: author.

The success of the prototype-based development cycle is confirmed by the presence of incremental score improvements. In terms of justifying the validity of the methodology employed, hypotheses relating to the quality of information and usability of the two software prototypes showed significant improvement from initial to operational prototypes (see Hypotheses 2-5, Figure 9.1). It is argued that the evaluation protocol combined with rapid prototyping were successful methods for capturing RSL informant requirements.

Another finding was the correlation between users' satisfaction with the software, and the technical complexity of the method currently providing their project progress reports (see Hypothesis 6, Figure 9.2). It could be concluded that RSL Informants with experience of computer-based solutions, are more able to understand the benefits offered by 3D Information Visualisation software tools.

9.4 Hypotheses- detailed results breakdown

The complete tabulated questionnaire results are presented in Appendix 8 and the stated "V" values relate to the source of values used for mean comparisons (see Appendix 9). The following is a more detailed breakdown of the significant hypotheses that are shown in Figures 9.1 and 9.2:

Hypothesis 1

H0: Mb ≤ Ma: RSL Informants' MEAN ability to assess project status is the same or worse with both prototypes than with current methods.

V6=2.7 V9=3.5

Mb (3.5) was greater than Ma (2.7). Therefore, the null hypothesis H0 is rejected, SIGNIFICANT with an observed t value of **-2.2283** at a reliability level of 95%.

Hypothesis 2

H0: Mb ≤ Ma: Software usability GETS WORSE or DOES NOT improve from initial to operational software prototype.

V37=3.28 V38=3.81

Mb (3.81) was greater than Ma (3.28). Therefore, the null hypothesis H0 is rejected, SIGNIFICANT with an observed t value of **-6.1251** at a reliability level of 95%.

Hypothesis 3

H0: Mb ≤ Ma: RSL Informants' perceive the quality of project reports to be the same or worse with the operational prototype than with the initial prototype.

V16=1.88 V17=2.5

Mb (2.55) was greater than Ma (1.88). Therefore, the null hypothesis H0 is rejected, SIGNIFICANT with an observed t value of **-3.1623** at a reliability level of 95%.

Hypothesis 4

H0: Mb ≤ Ma: RSL Informants' perceive the relevance of project reports to be the same or worse with the operational prototype than with the initial prototype.

V22= 3.2 V23=3.8

Mb (3.8) was greater than Ma (3.2). Therefore, the null hypothesis H0 is rejected, SIGNIFICANT with an observed t value of **-2.4495** at a reliability level of 95%.

Hypothesis 5

H0: Mb ≤ Ma: RSL Informants' perceive the format of project reports to be the same or worse with the operational prototype than with the initial prototype.

V31=2.8 V32=4.0

Mb (4.0) was greater than Ma (2.8). Therefore, the null hypothesis H0 is rejected, SIGNIFICANT with an observed t value of **-3.2071** at a reliability level of 95%.

Hypothesis 6 was tested for correlation rather than for a difference of means:

Hypothesis 6

H0: Correlation r = 0: There is NO negative or positive correlation between RSL Informants' satisfaction with the software and the technical complexity of the method currently providing their project progress reports.

The five RSL Informant's scores for question A0.40 were plotted against the five mean values for each RSL Informant's prototype question scores (mean for both prototypes). Figure 9.3 shows the resulting scatterplot.

A positive correlation was identified with an r value of 0.99 and an r2 value of 0.98. As the value for r is greater than 0.88, the correlation was found to be significant at a reliability level of 95%, for a sample size of five.

9.5 Chapter summary

This chapter has presented and discussed the results from the evaluation carried out for this research. In Section 9.3, the rejected null hypotheses were considered and various observations were made. Firstly, the RSL Informants' ability to correctly interpret project status (i.e. at milestone 4 in both of the scenarios) seemed significantly better using Procession than with earned value s-curves (see Hypothesis 1). This is arguably the most important of the results, because it represents a comparison between the subjects' original reporting methods and Procession.

In terms of justifying the suitability of the prototyping methodology, the rejection of Hypothesis 2 indicates that the usability of the software significantly improved between the Initial and Operational prototypes. In addition, the results for Hypotheses 3-5 suggest that the quality, relevance and format of Procession's performance reports improved significantly between the prototype versions. Finally, the rejection of Hypothesis 6 implies a significant correlation between the RSL Informants' satisfaction with the software and the technical complexity of the method currently providing their project performance reports.

Chapter 11 will draw some broader conclusions from the information presented here. However, before this final summary is made, the next chapter will describe possible future research directions.

Chapter 10: Future research directions

10.1 Chapter introduction

Initial consideration of the research problem suggested the possibility of applying Artificial Intelligence (AI) techniques to Procession. In the end, there was insufficient time to fully develop these ideas for the initial and operational prototypes of Procession. However, considerable effort had already been put into progressing this approach for the VTK prototype v1.0 (see Appendix 16).

One of the main reasons for not implementing the AI options was the difficulty in evaluating them. The author's prototyping methodology was described in an earlier chapter (see Figure 3.5). The timescale of this research only allowed for two prototyping cycles. In order to evaluate a software tool that 'learns' from repeated use, each prototyping loop would have required multiple sub-loops. This in term would have added complexity to the evaluation methodology.

This chapter briefly describes the initial work carried out in connection with applying AI techniques to the Procession tool, leaving open the possibility of further investigation. In addition, several other directions for future development are proposed.

10.2 Artificial Intelligence (AI)

It is suggested that later research versions of Procession might use a Deliberative AI approach, such as Case-Based Reasoning. It is proposed that heuristics, developed from experience on previous projects, would be stored in a 'legacy archive' and used to calculate the relevance of performance deviations from the project baseline. In Procession's VTK prototype v1.0 (see Section 6.3) an early attempt was made to implement this approach, with the Significance of Deviance Algorithm (see Appendix

16). This algorithm calculates the significance of an individual project task to the current total value for a specific deviation parameter in a specific project. The mathematical approach utilised is based on statistical methodologies, such as ANalysis Of VAriance (ANOVA). In Procession v1.0, this value was termed 'the significance of deviance', which is a current snapshot of significance. By using the current value to increase or decrease an on going record of previous significance (the legacy archive), Procession v1.0 'learnt' from its runtime experiences.

Appendix 16 provides a more in depth explanation of the significance calculation. A basic introduction to AI is provided in Appendix 15. This concludes that techniques such as Case-Based Reasoning may be the best choice for an application such as Procession. The author has published further details on Procession v1.0 elsewhere (North 2000c).

10.3 Implementing 'quality' as a deviation parameter

In Chapter 5, a 3D framework for measuring construction planning performance was proposed. One of the deviation performance parameters shown in Figure 5.3 is 'quality' or 'conformance'. However, it was stated that visualising quality was beyond the scope of this research. In the longer term, quality is a matter of great concern to construction clients and its future inclusion in Procession seems desirable. Project planning tools (such as Microsoft Project) do not currently provide any method for measuring quality. It is possible to imagine a quality measurement technique that would baseline anticipated 'snags' for a project type and then report variances.

10.4 Adding 'time' as a dimension

Procession only deals with time in the context of slippage from the planned programme (see Figure 5.3) i.e. duration variance or work variance. Each Procession 3D data surface is a single snapshot at time 'T'. The ability to view time-based changes in the data surface might prove extremely informative for the user. As a step towards this, Procession v1.2 features the button 'Toggle Animate' (see Appendix 14). This simulates the animation of the data surface, representing changes in Deviation Parameter levels over time. Future work might consider how to replace this simulated animation with a real chronological sequence.

10.5 Automating morphological analysis

In Chapter 5, Morphological Frameworks were introduced and one possible decomposition of construction performance into dimensions was proposed (see Figure 5.2). Of course, any such deconstruction of a system is largely subjective. As such, researchers are responsible for justifying their chosen breakdown with existing literature and their own investigations. In the specific case of construction performance (as presented in this thesis), further research might consider the applicability of alternative frameworks and their suitability for mapping to visual structures.

Taking this one stage further, could include the automating of the morphological analysis process. It is possible to envisage a software system capable of analysing a given system or problem and proposing an appropriate set of dimensions. There might be two, three or as many dimensions as specified by the user. Input to such a system might take the form of an information modelling language document, or schema. The primary output from this software might be a simple graphical representation with the dimensions labelled. A secondary software layer might allow the automatic generation of Information Visualisation graphics from the framework.

It is possible to imagine a scenario where a user presented with a data set and its information model, could rapidly generate an insightful Information Visualisation of this data. Initial reading suggests that little or no work has been carried out in this area. In the first edition of the peer-reviewed journal Information Visualization, editor-in-chief Chen (2002) hints at the importance of automatic generation when he writes:

> "the question is whether such geometry is intrinsically derivable from the data or one has to impose it on top of the data. Information Visualization traditionally focuses on finding meaningful and intuitive ways to present non-spatial and non-numerical information to people."

10.6 Chapter summary and contextualisation

This chapter proposed a future direction for the work initiated by this research. Before the development of Procession's Initial Prototype (v1.1), the previous version (v1.0) made use of slightly different concepts and technologies (see Section 6.3). Procession v1.0 experimented with the use of AI techniques. Heuristics were developed from experience on previous projects, stored in a 'legacy archive' and used to calculate the relevance of performance deviations from the project baseline.

Other possibilities for the future include; implementing 'quality' as a deviation parameter, adding 'time' as a dimension and automating morphological analysis. It is suggested that later versions of Procession might utilize and expand the ideas discussed in this chapter.

Chapter 11: Conclusions

11.1 Chapter introduction

In the introduction to this thesis, the underlying research problem was framed. It was stated that construction clients are currently without adequate software tools for simplifying the complexity of project performance. The low questionnaire scores obtained for the information quality of current reporting methods would seem to support this assertion (see Appendix 8). This chapter will now revisit the aim and objectives presented in Chapter 1. The research aim was given as:

- To investigate whether 3D Information Visualisation can provide construction clients with informative performance reports.

Chapter 4 documented the current techniques used for project performance reporting and suggested potential opportunities for applying 3D Information Visualisation to the same information. The null hypothesis that RSL Informants' mean ability to assess project status is the same or worse with both software prototypes than with an Earned Value Analysis s-curve, was rejected with a significance of 95% (see Chapter 9). It is concluded from this that software tools using 3D Information Visualisation provide a viable solution for the delivery of project performance reports to RSL Informants.

It is perhaps useful at this stage to summarise the advantages over traditional 2D performance visualisations provided by Procession's data surfaces. The primary limitation of a 2D s-curve is that it only shows a single summary view of the entire project (both estimated and actual performance). S-curves do not have the dimensional freedom to display a breakdown of individual task contributions to current status. This provides clients with an overall status but not the underlying factors causing that status. Of course, it is possible to generate s-curves for each task in a project. Procession's navigable 3D provides the status of each individual task simultaneously in a one-screen view. In addition, Procession reports 'by exception', showing a flat data surface unless there are performance deviations. This means that clients do not waste time attempting to interpret flat (i.e. on target) areas of the project.

Objective 1:

- To develop a conceptual 3D framework that decomposes construction project performance into axial dimensions.

In order to make use of the extra dimension and navigation offered by 3D graphics, attention was given to previous work on multi-dimensional morphological frameworks (see Chapter 5). A 3D framework for measuring construction planning performance was proposed (Chapter 5), with one of the dimensions utilising the performance parameters from Earned Value Analysis (see Sections 4.3 and 5.3).

Chapter 10 suggests possible improvements to the assignment of dimensions in the 3D framework. Firstly, the Deviation Parameter 'quality' (or 'conformance') was present in the Framework but omitted from all versions of Procession (see Section 5.3). It would seem that there is both justification and scope for its reintroduction to any future development of Procession. Secondly, whilst 'time' is represented in the 3D Framework as the Deviation Parameter 'slippage' (see Figure 5.3), it is not present as an axial dimension. Procession's 3D data surface illustrates a snapshot at time 'T'. The ability to view time-based changes in the data surface might prove extremely informative for the user. As a step towards this, Procession v1.2 features the button 'Toggle Animate' (see Appendix 14). This simulates the animation of the data surface, representing changes in Deviation Parameter levels over time.

Objective 2:

- To develop a software tool for construction clients that implements the conceptual 3D framework.

The software development aspect of this research has proven time-consuming but vital. Chapter 6 documents the technological decisions that carried Procession through three iterations from v1.0 to v1.2. Comments from RSL Informants revealed a striking dichotomy between the desire for a graphical tool that is visually very simple, but still offers low-level analysis and projection when required. One unexpected observation was the subjects' easy adoption of tools using 3D navigation. Both software usability scores and the Researcher's notes, revealed that even technically inexperienced users readily accepted the possibilities offered by graphical 3D.

Objective 3:

- To develop and apply a protocol for evaluating the developed software tool.

This research has successfully combined the software prototyping methodology proposed in Chapter 3 with the approach for evaluation outlined in Chapter 7. This approach has been validated by the inter-prototype improvements in questionnaire scores for 'usability', 'quality', 'relevance' and 'format' (see Figure 9.1).

11.2 Limitations of research

Despite the general success of the methodological design applied to this experiment, experience gained during the fieldwork revealed certain limitations. The primary limitation to this research is the assumption that all clients use similar reporting methods and are interested in the same information. In Chapter 1, it was explained that certain decisions had to be made before the development of the evaluation methodology and the initial prototype. Firstly, the choice of an earned value analysis s-curve for comparison with Procession's data surfaces, during the evaluation. On initial contact with the RSL Informants several of them indicated a working knowledge of performance measurement techniques, such as the s-curve. These techniques are also to be found in the standard project management textbooks and literature.

A second assumption was that different types of construction client are equally interested in all of the possible performance deviation parameters (time, cost and quality). The implication of this was that the initial software prototype needed to implement and visualise all of the performance parameters. All versions of Procession are capable of displaying a full range of deviation parameters (see Section 5.3), both 'to date' and 'at completion'. It should be noted that the Earned Value Analysis s-curves used for Procession's evaluation (see Chapter 7) were only showing 'to date' performance. Curves of this type usually display performance in a changing ratio of 'actual' and 'estimated' (see Section 4.3). For the purposes of the fictional scenarios, the curves consisted of only 'actual' values, requiring data beyond 'time now' to be obscured from the subject.

To make the comparison between the initial prototype and the first scenario valid, Procession v1.1 only used 'to date' performance parameters, such as 'cost variance' and 'schedule variance'. 'At completion' parameters, such as 'duration variance' were disabled, leaving some sections of the data surface permanently flat. In response to client requirements (see Appendix 10), the operational prototype (Procession v1.2) only displayed the 'at completion' performance parameter 'duration variance'. This reduction in performance parameters arguably reduced the value of using 3D. However, multiple tasks were still shown and it should be noted that RSL Informants are a unique group of construction clients. Their informational requirements may not be typical and Procession is capable of being customised to suit a range of performance data needs.

It should also be noted that the RSL Informants' informational requirements were identified by the prototyping methodology, proving the validity of this approach.

A final observation (in terms of limitations) relates to the 3D appearance of the 2D s-curves used for the evaluation (see Sections 8.7 and 8.8). As stated in Chapter 8, this look was chosen because it seemed to increase intelligibility. The number of dimensions was still 2 (time x performance/cost) as opposed to Procession's 3 dimensions (deviation parameter x task x deviation level). However, this choice for the 2D s-curve appeared to cause confusion for some observers and should be avoided in future works.

11.3 Relevant technology rejected as too immature for inclusion at research outset

When work first started on this thesis in 1997, several new technologies and standards (for example STEP- see Chapter 2) were investigated for possible use with Procession. The most important of these was Extensible Markup Language (XML). Development of XML started in 1996 and has been an approved World Wide Web Consortium technology since February 1998 (W3C 2000). XML is a metalanguage which is to say a programming language characterized by the ability to both define a new sub-set language and/or to dynamically extend its own functionality. This involves the definition of new mark-up language "tags", in order to handle data formats unique to a specific industry or application.

Languages such as HTML are static. Its tags are predetermined by a metalanguage (SGML in the case of HTML). XML is a data content language. It allows the definition and manipulation of data separately from its presentation. XML is actually a family of languages, one of which is XML Schema. The latter allows the modeling and verification of information in a similar manner to STEP.

If development on Procession was commencing today for the first time, XML might have a contribution to make. However, XML would not currently make interoperability with other construction industry ICT tools any more straightforward. XML would now enable the definition of an interface for Procession, but this would not be an industry standard interface. New Microsoft Office products make full use of XML and it is likely that an XML link between Procession and Microsoft Project might now be established. There are several attempts to link STEP and IFCs with XML. These include; ifcXML, bcXML and Bentley's aecXML (Amor & Faraj 2001). Ultimately, none of these approaches represents 'performance' data (see Chapter 2) and so provides no advantages in the context of this research.

11.4 Scalability and generalisation of research (both in construction and other domains)

It is important to discuss the applicability of Procession to other data sets. Working on construction industry projects, the current v1.2 release of Procession is 'hard-wired' for a maximum of one thousand tasks. Therefore, it is capable of coping with larger projects than the 'live' project used for the research evaluation. With small revisions, Procession could be modified to work with projects sizes limited only by the performance of its host computer. For extremely large projects, Procession would require additional filtering to remove variations below a given threshold. This might be defined in the specification of the Microsoft Project export map providing Procession's input data. Procession would generalize directly to non-construction projects of various sizes, with the above provisos. With modified performance parameters, Procession could be applied to non-construction data sets, for example; finance information.

In terms of originality, it is believed that Procession is the first 3D Information Visualisation software tool developed specifically for construction clients. In addition, Procession would seem to be the only partner-application of this type for the market-leading Microsoft Project. As originally proposed, this research has delivered a conceptual 3D framework for the interpretation of construction performance and a practical example of implementing this framework. Therefore, this initial demonstration that 3D Information Visualisation can provide construction clients with informative performance reports is offered as a relevant contribution to knowledge. The Researcher has published further details of this research elsewhere (North 1999, 2000a, 2000b, 2000c, 2000d, 2003). It is also referenced in Winch's *Managing Construction Projects: an Information Processing Approach* (2002, pp. 255-256).

References

Amor, R. & Faraj, I. (2001): 'Misconceptions about integrated project databases', *Electronic Journal of Information Technology in Construction (ITcon)*, 6, p.57.

Aouad, G., Ormerod, M., Sun, M., Sarshar, M., Barrett P. & Alshawi, M. (2000): 'Visualisation of construction information: a process view', *International Journal of Computer Intergrated Design & Construction (CIDAC)*, 2 (4), pp.207-215.

Baldwin, A.N., Austin, S.A., Hassan, T.M. & Thorpe, A. (1999): 'Modelling information flow during the conceptual and schematic stages of building design', *Construction Management & Economics*, 17, 2, pp.155-167.

Bentley (2002): ProjectBank, available from: http://www.bentley.com/, [accessed May 2003].

Bertin, Jacques (1983): *Semiology of Graphics: Diagrams, Networks, Maps*, USA: University of Wisconsin Press.

British Standards Institute (2000): *BS6079: A Guide to Project Management*, London.

Brooks, F.P. (1995): *The Mythical Man-Month: Essays on Software Engineering*, 20th Anniversary Edition, UK: Addison-Wesley.

Building Project Information Committee (1998): *Uniclass- Unified classification for the construction industry*, available from The Construction Confederation, London.

Chen, C. (1999): *Information Visualization and Virtual Environments*, USA: Springer-Verleg.

Chen, C. (2002): 'Editorial', *Information Visualization*, 1(1), Palgrave Macmillan, pp.1-4.

Card, S.K., Macklinlay, J.D. and Shneiderman, B. (1999): *Readings in Information Visualization- using vision to think*, USA: Morgan Kaufmann.

Chi, Ed Huai-hsin, Riedl, John, Barry, Philip and Konstan, Joseph (1998):'Principles for Information Visualization Spreadsheets', *IEEE Computer Graphics and Applications,* 18, 4 pp.30-38.

Chuah, Mei C. (1998):'Information rich glyphs for software management data', *IEEE Computer Graphics and Applications,* 18, 4 pp.24-29.

Cleveland, William S. (1993): *Visualizing Data*, USA: Hobart Press.

Construct IT Centre of Excellence (1996): *Benchmarking Best Practice Report: Briefing and Design*, UK: Department of the Environment.

Construct IT Centre of Excellence (1997a): 'Integrated engineering solutions STRATEGY 1996-2000', proceedings of *The Armathwaite Initiative: Global Construction IT Futures International Meeting*, UK: Engineering and Physical Sciences Research Council.

Construct IT Centre of Excellence (1997b): 'Visualisation technologies', proceedings of *The Armathwaite Initiative: Global Construction IT Futures International Meeting*, UK: Engineering and Physical Sciences Research Council.

Crinnion, John (1991): *Evolutionary Systems Development: a practical guide to the use of prototyping within a structured systems methodology*, New York: Plenum Press.

CSEP (1995): *Introduction to Monte-Carlo methods*, USA: The Computational Science Education Project, available from: http://csep1.phy.ornl.gov/CSEP/MC/MC.html, [accessed May 2003].

Davis, Langdon and Everest (2000): *Spon's Architects' and Builders' Price Book*, one hundred and twenty fifth edition, Davis Langdon & Everest (Editor), UK: E. & F.N. Spon.

Everitt, B. and Hay, D. (1992): *Talking about Statistics: a psychologist's guide to design & analysis*, UK: Hodder & Stoughton.

Extreme Programming (2003): *XP Home Page*, available from: www.extremeprogramming.org, [accessed May 2003].

Faraj, I., Alshawi, M., Aouad, G. et al. (1998a): 'The Implementation of the IFC in a distributed computer integrated environment' in *Product and Process Modelling in the Building Industry, the proceedings of ECPPM '98*, R. Amor (ed.), UK: Building Research Establishment, pp.187-198.

Faraj, I., Alshawi, M., Aouad, G., Child, T. and Underwood, J. (1998b): 'An IFC Web-Based Collaborative Construction Computer Environment: WISPER', a working paper.

Fleming, A. L. & Cooper, R. (2000): 'The development of a process mapping methodology for the Process Protocol Level 2' in *Product and Process Modelling in Building and Construction* (the proceedings of ECPPM 2000, the third European conference on product and process modelling in the building and related industries, Lisbon, Portugal, 25th-27th September 2000), Gonçalves, Steiger-Garção and Sherer (eds.), The Netherlands: Balkema, ISBN 90-5809-179-1, pp.187-192.

Foley, J. (2000): 'Getting there: the top ten problems left', *IEEE Computer Graphics and Applications*, 20(1), pp. 66-68.

Forsberg, K., Mooz, H. and Cotterman, H. (1996): *Visualizing Project Management*, USA: John Wiley & Sons.

Gann et al (1996): *Information Technology Decision Support in the Construction Industry: Current Developments and Use in the United States*, UK: Department of Trade and Industry, Science Policy Research Unit, University of Sussex.

Hackos, J.T., Redice, J.C. (1998): *User and Task Analysis for Interface Design*, USA: John Wiley & Sons, Inc..

Hall, A.D.(1969):'Three-dimensional morphology of systems engineering' in *IEEE Transactions on System Science and Cybernetics*, SSC-5(2), pp.156-160.

Haas, W.R. (2000): 'STEP and its application in the construction industry' in *Product and Process Modelling in Building and Construction* (the proceedings of ECPPM 2000, the third European conference on product and process modelling in the building and related industries, Lisbon, Portugal, 25th-27th September 2000), Gonçalves, Steiger-Garção and Sherer (eds.), The Netherlands: Balkema, ISBN 90-5809-179-1, pp.25-31.

Heesom, D. and Mahdjoubi, L. (2002): *Technology opportunities and potential volume 2: project planning software review*, University of Wolverhampton, a VIRCON project report.

Hegazy, T. and Ayed, A. (1998):'Neural network model for parametric cost estimation of highway projects' in *Journal of Construction Engineering and Management*, 124/3, USA: American Society of Civil Engineers, pp.210-218.

Kagioglou, M., Aouad, G., Cooper, R. and Hinks, J. (1998): 'The Process Protocol: Process and IT modelling for the UK construction industry', *Product and Process Modelling in the Building Industry*, the proceedings of ECPPM '98, ed. R.Amor, pp.267-276.

Karsila, K., Karhu, V. & Björk, B.C. (2000): 'Development of process modelling tools for the construction industry' in *Product and Process Modelling in Building and Construction* (the proceedings of ECPPM 2000, the third European conference on product and process modelling in the building and related industries, Lisbon, Portugal, 25th-27th September 2000), Gonçalves, Steiger-Garção and Sherer (eds.), The Netherlands: Balkema, ISBN 90-5809-179-1, pp.179-186.

Kelly, K. (1994): *Out of control: the new biology of machines*, London: Fourth Estate Ltd.

Khosrowshahi, F (1998): 'A visual approach to the analysis of curve profile: a case in the construction industry' in *The Proceedings of IV (Information Visualisation) 1998*, USA: IEEE.

Khosrowshahi, F (1999): 'The mathematics of shape geometry approach to the analysis of curve profile' in *The Proceedings of IV (Information Visualisation) 1999,* USA: IEEE.

Knight, K. and Kaka, A. (1998): 'Application of computer based photogrammetric techniques to building, modelling and costing', *Product and Process Modelling in the Building Industry,* the proceedings of ECPPM '98, ed. R.Amor, pp.289-295.

Kumaraswamy, M.M. and Thorpe, A. (1996a): 'Systematizing construction project evaluations', *Management in Engineering*, New York: American Society of Civil Engineering, 12(1), pp.34-39.

Levine, H. (1999): *Real-time Status vs. Period Data*, USA: Scitor Corporation, available from: http://www.scitor.com/ [accessed: May 2003].

Lowry, R. (2000): *Concepts and Applications of Inferential Statistics*, available from: http://faculty.vassar.edu/~lowry/webtext.html, [accessed May 2003].

Mckinney, K., Kim, J., Fischer, M. & Howard, C. (1996): 'Interactive 4D CAD' in *the proceedings of ASCE (the third congress in computing in civil engineering)*, Anaheim, USA, pp.383-389.

Mckinney, K., & Fischer, M. (1998): 'Generating, evaluating and visualizing construction schedules with 4D-CAD tools', *Automation in Construction*, 7, 6, pp.433-447.

Meredith, Jack R., Mantel, Samuel J. JR. (1995): *Project Management: A Managerial Approach*, USA: John Wiley & Sons Inc..

Minksky, Marvin (1985): *The Society of Mind*, USA: Simon & Schuster.

Miroslaw, J. (1998):'Fuzzy logic for evaluating alternative construction technology', *Journal of Construction Engineering and Management*, 124/4, USA: American Society of Civil Engineers, pp.297-304.

Modin, J. (1995): 'KBS-CLASS: a neural network tool for automatic content recognition of building texts', *Construction Management and Economics*, 13/5.

Mumford, E. (1995): *Effective Systems Design and Requirements Analysis*, UK: MacMillan Press.

NCSS Statistical Software available from: http://www.ncss.com, [accessed May 2003].

North, S. (1999) Enter the Virtual Building Site, *Housing*, April 1999, UK: The Chartered Institute of Housing, p.40.

North, S. (2000a): 'Procession: using intelligent 3d information visualisation to support client understanding during construction projects' *in Visual Data Exploration and Analysis VII*, Robert F.Erbacher, Philip C. Chen, Jonathan C. Roberts, Craig M. Wittenbrink, Editors in *The proceedings of SPIE (The International Society for Optical Engineering) Vol. 3690 (24th-26th January 2000, San Jose)*, USA: SPIE, ISSN 0277-786X, ISBN 0-8194-3578-3, pp.356-364.

North, S. (2000b): 'Towards a 3D Visualisation Framework for Construction Planning with Procession' in *The Proceedings of IV2000 (IEEE International Conference on Information Visualization)*, E.Banissi, M.Bannatyne, C.Chen, F.Khosrowshahi, M. Sarfraz, A. Ursyn, Editors, USA: IEEE Computer Society, ISSN 1093-9547, ISBN 0-7695-0743-3, IEEE order no. PR00743, pp.577-582.

North, S. (2000c): 'Significant decisions: supporting social housing clients through procession's artificially intelligent 3d information visualisation' in *Design & Decision Support Systems in Urban Planning: proceedings of the 5th international conference (Nijkerk, The Netherlands, 22nd-25th August 2000)*, Harry Timmermans and Bauke de Vries, Editors, The Netherlands: Eindhoven University of Technology, ISBN 90-6814-110-4, pp.215-228.

North, S. (2000d): 'Process to procession: visualising construction processes with procession's 3d information visualisation' in *Product and Process Modelling in Building and Construction* (the proceedings of ECPPM 2000, the third European conference on product and process modelling in the building and related industries, Lisbon, Portugal, 25th-27th September 2000), Gonçalves, Steiger-Garção and Sherer (eds.), The Netherlands: Balkema, ISBN 90-5809-179-1, pp.367-374.

North, S. (2003): 'Procession: a three-dimensional information visualisation tool for construction project statusing', Special issue on product and process modelling in construction and related industries, Ricardo Goncalves & Adolfo Steiger-Garcao (guest eds.), *International Journal of Internet and Enterprise Management* (IJIEM), ISSN:1476–1300, Vol.1, No.2 Spring (April-June) 2003, UK: Inderscience.

North, S. & Winch, G.M. (2002): 'VIRCON: a proposal for critical space analysis in construction planning' in *eWork and eBusiness in Architecture, Engineering and Construction, the proceedings of the fourth European conference on product and process modelling in the building and related industries (ECPPM 2002)*, Turk & Scherer (eds.), Portorož / Slovenia, Rotterdam: Balkema, ISBN 90 5809 507 X, pp.359-364.

Nwana, Hyacinth S. (1996): "Software agents: an overview", *Knowledge Engineering Review*, Vol. 11 (No. 3) pp.74-81.

Object Management Group (2003): CORBA home page, available from: http://www.corba.org, [accessed May 2003].

OpenGL home page, available from: http://www.opengl.org/ [accessed May 2003].

Pnueli, A. (1986): 'Specification and development of reactive systems', *Information Processing '86*, H. Kugler, ed., pp. 845-858.

Pohl, K. (1994): 'The three dimensions of requirements engineering: a framework and Its applications', *Information Systems* 19, 3, pp.243-258.

Pressman, S. (1992): *Software Engineering A Practitioner's Approach*, third edition, Singapore: McGraw-Hill.

Riskman (2000): available from: www.riskdriver.com/riskmantool/, [accessed August 2001- link unavailable by May 2003].

Roberts, J. C. (2000): 'Display models- ways to classify visual representations', *International Journal of Computer Integrated Design and Construction*, 2(4), pp. 242-250.

SAP (2003): available from: www.sap.com, [accessed May 2003].

Shi, J. J., Cheung, S. O. and Arditi D. (2001): 'Construction delay computation method' in *Journal of Construction Engineering and Management*, 127, 1, USA: ASCE, p.60.

Shroeder, W., Martin, K., Lorensen, B. (1998) *The Visualization Toolkit*, USA: Prentice Hall.

Shneiderman, B. (1996): 'The eyes have it: a task by data type taxonomy for information visualizations' in *Proceedings of IEEE Symposium on Visual Languages '96*, Languages '96, USA, pp.336-343.

Spence, R. (2001): *Information Visalization*, Addison-Wesley, UK.

Strothotte, T. (1998): *Computational Visualization: Graphics, Abstraction and Interactivity*, USA: Springer-Verleg.

Tufte, Edward (1983): T*he Visual Display of Quantitative Information*, USA: Cheshire Press.

Tufte, Edward (1990): *Envisioning Information*, USA: Cheshire Press.

Tufte, Edward (1997): *Visual Explanations*, USA: Cheshire Press.

Turing, Alan (1950): 'Computing machinery and intelligence', *Mind*, October 1950.

Turner, Rodney J. (1993): *The handbook of project-based management: improving the processes for achieving strategic objectives*, UK: McGraw-Hill.

US Department of Defense and NASA (1962): *DoD and NASA Guide to PERT/Cost*, USA: Government Printing Office.

US Department of Defense (1967): *Cost/Schedule Control Systems Criteria (C/SCSC)*, USA: Government Printing Office.

VR_Systems U.K. (1998): press release posted to the *UKVRSIG Internet Mailing list*, 15th April.

Turner, Ian (1997): 'Intelligent systems', *The Armathwaite Initiative: Global Construction IT Futures International Meeting*, UK: Construct IT Centre of Excellence, pp.85-88.

Welcome (2002): Open Plan and Spider product information, available from: http://www.welcome.com, [accessed August 2002].

Wideman, R. Max (1999): *Wideman Comparative Glossary of Common Project Management Terms v3.0*, available from: http://www.maxwideman.com/pmglossary/ [accessed: August 2002].

Wieringa, R. J. (1996): *Requirements Engineering: frameworks for Understanding*, UK: John Wiley & Sons.

Winch, G., Usami, A., Edkins, A. (1998): 'Towards Total Project Quality: a Gap Analysis approach', *Construction Management and Economics*, 16.

Winch, Graham M. (2002): *Managing Construction Projects: an Information Processing Approach,* Oxford: Blackwell Science, ISBN 0-632-05888-9.

Wiss, U., Carr, D. & Jonsson, H. (1998): 'Evaluating 3-Dimensional Information Visualization Designs: a Case Study' in *Proceedings of the IEEE Conference on Information Visualization*, London, UK., pp.137-144.

World Wide Web Consortium- W3C (2002): *Extensible Markup Language (XML)*, available from: http://www.w3.org/XML/, [accessed August 2002].

Yeh, I-Cheng (1998):'Quantity estimating of building with logarithm-neuron networks', *Journal of Construction Engineering and Management*, 124/5, USA: American Society of Civil Engineers, pp.374-380.

Zowghi, D. (1996): available from: http://www-comp.mpce.mq.edu.au/~didar/seweb/keyissues.html, [accessed May 2003].

Zhang, Ping (1996): 'Visualizing production planning data', *IEEE Computer Graphics & Applications*, September, pp.7-10.

Zhang, Ping (2000): 'Business Information Visualization: Guidances for Research and Practice', *Encyclopedia of Library and Information Science*, Volume 69.

Zhang, Ping, & Dan Zhu (1997): 'Information Visualization in Project Management and Scheduling' in *Proceedings of The 4th Conference of the International Society for Decision Support Systems* (ISDSS'97), Switzerland, July 21-22 1997.

Appendix 1: Software development incremental journal notes

Procession v1.0 Visualization ToolKit version changes and bugs list:

d1 Feb 1999

bugs:

1.'Save as...' not working

2.'Animation' menu tick not up-dating, can not turn off

d2

Fixed:

1. Re-Added centring of child-frame,
'cause multiple windows were cascading.
2. Changed camera position.
3. Changed names of 'deviation' to relevant variance name.
4. Modified animation bit in processionView.cpp,
so produces more interesting cycle.

Added:

1.Added axes to processionView.cpp in
DoMyShowVisualisation().

Still To Do:

1. Change .csv file format and parser to get % of task complete
and use for agent to work out if project completed.
2. Write agent.
3. Get axes auto-adjusting to size of visualisation.
4. 'On-mouse over' to show task name on status bar.
5. Get axes to animate.

Bugs:

1.'Save as...' not working

2.'Animation' menu tick not up-dating, can not turn off.

d3 19/3/99

Fixed:

1. "save as" now works- added modeWrite in processionView.cpp.
2. Found and fixed bug in MyFileParser.cpp- When creating task array, y operator incrementing and creating empty element between valid elements. Added z operator.
3. Animation tick works most of the time.

Added:

1. Added first attempt at agent coding.
2. Added % complete to CSV file (not actually changed Project98 map).
3. Added legacy archive and all in/out routines.
4. Agent weighting values are currently generated randomly.
5. Visualisation is now accurately mapping deviations to cells on data surface. Colour of surface currently applied to all three parameters in each task instead of individually.
6. Changed data surface grid so that it had 3 parameters in 'x'.
7. Added compiled Help.

Still To do:

1. Get axes auto-adjusting to size of visualisation.
2. 'On-mouse over' to show task name + value on status bar.
3. Get axes to animate.
4. Filter total number of tasks visualised (i.e. report by exception).
5. Add 3d text labels on axes?
6. Calculate weightings 'intelligently'

7. Map weightings to individual parameters in visualisation.
8. General status info to be shown on status bar.
9. Error trapping needs checking and adding both generally and particularly in HTTP client.
10. Figure out problems with (and then write) compiled help.

Bugs:
1. Still MDI problems with window freeze in some situations.

d4 19/3/99

Fixed:
Added:
Still to do:
1. Up-date virtual key map
2. Move 'save as' back to 'File' menu (requires making file contents global.
3. Check for memory leaks.
4. Add 'tickable' menu items for VTK stuff:
stereo, wire and solid.
5. Use dates as default file name for csv,bmp+wrl.
6. Add 'reset archive' (i.e. delete) as menu option.
7. Filtering+add ability to specify max number of tasks
(filtering level).
8.

Bugs:
1.Maximize not working on parent window.
2.Archive file path needs to be fixed to install directory...so that when a csv file is opened in another dir location, it doesn't create a new archive there.

3. Still MDI problems with window freeze in
some situations.
4. Black windows in some situations.

Nice-to-haves:
1. Multi-user environment

d8 26/4/99

Fixed:
1. Removed 'tile' etc. from Window menu 'cause not doing anything with child window permanently maxed.

Added:

1. Weightings now calculated 'intelligently' (agent.cpp).
2. Weightings now mapped to individual parameters in visualisation (agent.cpp).
3. Verbose tickable menu item allows alert messages to be turned on or off.
4. Alert boxes display the calculations used to determine relevance weightings.

Still to do:
1. Up-date virtual key map
2. Move 'save as' back to 'File' menu
(requires making file contents global).
3. Check for memory leaks.
4. Add 'tickable' menu items for VTK stuff:
stereo, wire, solid, reset camera position and stereo on/off.

5. Use dates as default file name for csv,bmp+wrl.
6. Add 'reset archive' (i.e. delete) as menu option.
7. Filtering: add ability to specify max number of tasks (filtering level).
8. Get axes auto-adjusting to size of visualisation.
9. 'On-mouse over' to show task name + value on status bar.
10. Get axes to animate.
11. Add 3d text labels on axes?
10. Error trapping needs checking and adding both generally and particularly in HTTP client.

Bugs:
1.Maximize not working on parent window.
2.Archive file path needs to be fixed to install directory...so that when a csv file is opened in another dir location, it doesn't create a new archive there.
3. Still MDI problems with window freeze in some situations.
4. Black windows in some situations.
5. Colour scale probs-different colours on first run.
6. Compiled help not working properly.

d10 8/2/00

Fixed:
1. Increased const int "BUFF_SIZE" in myFileParser.cpp from 1024 to 32768. This variable is used to size arrays holding loaded data. Made change because crashed on first test with a decent sized data-file.

2. Did same as 1. in myVisualisation.cpp.
3. Changed warning level from 3 to 2 (settings).
4. Added new method for calculating surface scale, in myVisualisation.cpp.
5. Fixed compiled help and wrote skeleton of file.

Bugs:
1.Maximize not working on parent window.
2.Archive file path needs to be fixed to install directory...so that when a csv file is opened in another dir location, it doesn't create a new archive there.
3. Still MDI problems with window freeze in some situations.
4. Black windows in some situations.
5. Colour scale probs-different colours on first run.

Appendix 2: C++ code for operational prototype of Procession

```
// Steve North 2000. All Rights Reserved

// Procession

#include "stdafx.h"
#include "procession.h"
#include "processionDlg.h"
#include "series.h"
#include "panel.h"
#include "marks.h"
#include "customseries.h"
#include "legend.h"
#include "Canvas.h"
#include "Axes.h"
#include "Axis.h"
#include "AxisTitle.h"
#include "ChartFont.h"
#include "Pen.h"
#include "Aspect.h"
#include "TeeOpenGL.h"
#include "TeeChartDefines.h"
#include "valuelist.h"
#include "walls.h"
#include "axislabels.h"

#include "zoom.h"
#include "titles.h"
#include "strings.h"
#include "scroll.h"

#include "printer.h"
#include "export.h"

// next required for FTP client
#include <afxinet.h>
#include <iostream.h>

#ifdef _DEBUG
#define new DEBUG_NEW
#undef THIS_FILE
static char THIS_FILE[] = __FILE__;
#endif

// next value selects type of graph from TeeChart types
const unsigned int mySeries=3;

// default settings for FTP client
CString defaultURL="ftp.tripod.com";
CString defaultUserName="p_procession";
```

```
CString defaultPassWord="myPass";
CString defaultRemotePath="/procession/example/";
CString defaultRemoteFileName="example1.dbf";
CString defaultLocalFileName="test1.dbf";

// the directory set up as an ODBC source
CString defaultLocalODBCDataDirectory="c:\\procession\\data\\";

// header for view source option from client
CString display="negative value (-) indicates days ahead\n";

// default message at bottom of screen
CString statusBarDefaultMessage="Task info...";

// font RGB colour
const unsigned long textRedValue=0;
const unsigned long textGreenValue=0;
const unsigned long textBlueValue=0;

// background RGB colour
const unsigned long panelRedValue=255;
const unsigned long panelGreenValue=255;
const unsigned long panelBlueValue=255;

// 3d navigation panel RGB colour
const unsigned long NavPanelRedValue=0;
const unsigned long NavPanelGreenValue=0;
const unsigned long NavPanelBlueValue=0;

// axis labels
CString YAxisTitle="Days At Completion";
CString XAxisTitle="Project Tasks";
const unsigned long XaxisTitleFontSize=20;
const unsigned long YaxisTitleFontSize=10;

// more FTP stuff
CString m_strFTPLocalDirAndFile;
CString m_strFTPRemoteDirAndFile;

// used to generate default file names using time
char theTime[100];
time_t ltime;

// animation and fly-thru stuff
// controls speed...bigger is slower
int defaultTimerSpeed=500;

bool flyThru=false;
bool animateSurface=false;
int frameNumber;

// these are the animation frame
// values where each array set
// is in the format {Rotation,Elevation,Tilt,HorizontalOffset,VerticalOffset,Zoom}
const unsigned int maxFrame=28;
int animationKeys[maxFrame][6]=
```

```
{
{19,346,0,29,-24,146},{20,349,0,9,16,294},{341,347,0,9,16,294},{260,346,0,9,16,294},
{260,356,0,9,16,294}, {260,356,0,-2,11,579},{272,12,0,-2,11,579},{272,12,0,-2,11,784},
{272,10,0,-2,11,7305},{272,9,0,-6,41,634},{293,9,0,-6,41,634},{321,9,0,-6,41,634},
{326,329,0,-6,41,500},{326,329,0,-6,41,287},{349,293,0,-6,41,287},{349,317,360,-6,41,166},
{300,350,0,-9,9,145},{309,301,360,-9,9,145},{266,307,9,-9,9,145},{268,351,360,-9,9,145},
{268,357,360,-9,9,464}, {264,348,360,-2,26,995},{252,298,360,-2,26,995},{324,298,360,-1,-
44,98},
{260,346,0,9,16,294}, {341,347,0,9,16,294},{40,349,0,9,16,294},{20,346,0,29,-24,294},
};

// note setting this next value too high (i.e. 1000) causes floating point errors in Win95
// Problem seems to be non-OpenGL related.
int numberOfTasksInSimulatedAnimation=100;
int numberDigitsInSecsSince1970=15;

CString revealValue;

const int BUFF_SIZE=1024;
char buffer[BUFF_SIZE]={NULL};

/////////////////////////////////////////////////////////////////////////////
// CProcessionDlg dialog

CProcessionDlg::CProcessionDlg(CWnd* pParent /*=NULL*/)
	: CDialog(CProcessionDlg::IDD, pParent)
{
	//{{AFX_DATA_INIT(CProcessionDlg)
	m_strInfoWindow = _T("");
	m_elevation = _T("");
	m_horizontal = _T("");
	m_vertical = _T("");
	m_zoom = _T("");
	m_tilt = _T("");
	m_rotation = _T("");
	m_srtURL = _T("");
	m_strUserName = _T("");
	m_strLocalODBCDataDirectory = _T("");
	m_strLocalFileName = _T("");
	m_strPassWord = _T("");
	m_strRemoteFileName = _T("");
	m_strRemotePath = _T("");
	//}}AFX_DATA_INIT
	// Note that LoadIcon does not require a subsequent DestroyIcon in Win32
	m_hIcon = AfxGetApp()->LoadIcon(IDR_MAINFRAME);
}

void CProcessionDlg::DoDataExchange(CDataExchange* pDX)
{
	CDialog::DoDataExchange(pDX);
	//{{AFX_DATA_MAP(CProcessionDlg)
	DDX_Control(pDX, IDC_TCHART1, m_Chart1);
	DDX_Control(pDX, IDC_TEECOMMANDER1, m_TeeCommander1);
	DDX_Text(pDX, IDC_INFO_WINDOW, m_strInfoWindow);
	DDX_Text(pDX, IDC_ELEVATION, m_elevation);
	DDX_Text(pDX, IDC_HORIZONTAL, m_horizontal);
```

```
		DDX_Text(pDX, IDC_VERTICAL, m_vertical);
		DDX_Text(pDX, IDC_ZOOM, m_zoom);
		DDX_Text(pDX, IDC_TILTDISPLAY, m_tilt);
		DDX_Text(pDX, IDC_ROTATION, m_rotation);
		DDX_Text(pDX, IDC_URL, m_srtURL);
		DDX_Text(pDX, IDC_USERNAME, m_strUserName);
		DDX_Text(pDX, IDC_LOCALDIRECTORY, m_strLocalODBCDataDirectory);
		DDX_Text(pDX, IDC_LOCALFILE, m_strLocalFileName);
		DDX_Text(pDX, IDC_PASSWORD, m_strPassWord);
		DDX_Text(pDX, IDC_REMOTEFILENAME, m_strRemoteFileName);
		DDX_Text(pDX, IDC_REMOTEPATH, m_strRemotePath);
		//}}AFX_DATA_MAP
}

BEGIN_MESSAGE_MAP(CProcessionDlg, CDialog)
		//{{AFX_MSG_MAP(CProcessionDlg)
		ON_WM_PAINT()
		ON_WM_QUERYDRAGICON()
		ON_COMMAND(ID_FILE_OPEN, OnFileOpen)
		ON_BN_CLICKED(IDC_BUTTON1, OnButton1)
		ON_BN_CLICKED(IDC_HOME, OnHome)
		ON_BN_CLICKED(IDC_FLIP, OnFlip)
		ON_BN_CLICKED(IDC_TURN, OnTurn)
		ON_BN_CLICKED(IDC_TILT, OnTilt)
		ON_BN_CLICKED(IDC_STRAIGHTEN, OnStraighten)
		ON_BN_CLICKED(IDC_URLSUBMIT, OnUrlsubmit)
		ON_BN_CLICKED(IDC_FLYTHRU, OnFlythru)
		ON_COMMAND(IDPRINT, OnPrint)
		ON_COMMAND(IDSCREENSHOT, OnScreenshot)
		ON_COMMAND(ID_VIEWDATA, OnViewdata)
		//}}AFX_MSG_MAP
END_MESSAGE_MAP()

/////////////////////////////////////////////////////////////////////////////
// CProcessionDlg message handlers

BOOL CProcessionDlg::OnInitDialog()
{
		CDialog::OnInitDialog();

// set all FTP edit box values to defaults
m_srtURL=defaultURL;
m_strUserName=defaultUserName;
m_strPassWord=defaultPassWord;
m_strRemotePath=defaultRemotePath;
m_strRemoteFileName=defaultRemoteFileName;
m_strLocalFileName=defaultLocalFileName;
m_strLocalODBCDataDirectory=defaultLocalODBCDataDirectory;
m_strFTPLocalDirAndFile=m_strLocalODBCDataDirectory+m_strLocalFileName;
m_strFTPRemoteDirAndFile=m_strRemotePath+m_strRemoteFileName;

// do Icon stuff
		SetIcon(m_hIcon, TRUE);			// Set big icon
		SetIcon(m_hIcon, FALSE);		// Set small icon

		// link chart to 3d navigation bar
```

```
	m_TeeCommander1.SetChartLink(m_Chart1.GetChartLink());

// set 3d depth to 100%
m_Chart1.GetAspect().SetChart3DPercent(100);

// turn on OpenGL functionality...leave false for native 3d format
// native is faster but rendering is inferior, no lights and
// no 360 degree rotation
m_Chart1.GetAspect().GetOpenGL().SetActive(true);

// set axis sizes and titles
m_Chart1.GetAxis().GetLeft().GetTitle().GetFont().SetSize(YaxisTitleFontSize);
m_Chart1.GetAxis().GetBottom().GetTitle().GetFont().SetSize(XaxisTitleFontSize);
m_Chart1.GetAxis().GetLeft().GetTitle().SetCaption(YAxisTitle);
m_Chart1.GetAxis().GetBottom().GetTitle().SetCaption(XAxisTitle);

// set colour of 3d nav control panel
m_TeeCommander1.SetColor(RGB(NavPanelRedValue,NavPanelGreenValue,NavPanelBlu
eValue));

// set all font and backgrounds to default colours
m_Chart1.GetPanel().SetColor(RGB(panelRedValue,panelGreenValue,panelBlueValue));
m_Chart1.GetAxis().GetLeft().GetTitle().GetFont().SetColor(RGB(textRedValue,textGreenVa
lue,textBlueValue));
m_Chart1.GetAxis().GetBottom().GetTitle().GetFont().SetColor(RGB(textRedValue,textGree
nValue,textBlueValue));
m_Chart1.GetAxis().GetLeft().GetTicks().SetColor(RGB(textRedValue,textGreenValue,textBl
ueValue));
m_Chart1.GetAxis().GetBottom().GetTicks().SetColor(RGB(textRedValue,textGreenValue,te
xtBlueValue));
m_Chart1.GetAxis().GetLeft().GetLabels().GetFont().SetColor(RGB(textRedValue,textGreen
Value,textBlueValue));
m_Chart1.GetAxis().GetBottom().GetLabels().GetFont().SetColor(RGB(textRedValue,textGr
eenValue,textBlueValue));

// set left axis values to calculate automatically
m_Chart1.GetAxis().GetLeft().SetAutomatic(true);

// turn off default zoom and scroll on mouse click- very annoying!
m_Chart1.GetZoom().SetEnable(false);
m_Chart1.GetScroll().SetEnable(pmNone);

// get 3d position to "home"
OnHome();

// set animation timer speed
m_Chart1.SetTimerInterval(defaultTimerSpeed);

// turn off solid walls
m_Chart1.GetWalls().SetVisible(false);

// turn off chart key
m_Chart1.GetLegend().SetVisible(false);

// check that this really is a blank canvas...
if (m_Chart1.GetSeriesCount() == 0)
```

```
{
// set message at bottom of screen to default
revealValue=statusBarDefaultMessage;
m_strInfoWindow=revealValue;
// force screen up-date
UpdateData(FALSE);

// create two charts...one of them will never be visible...read on...
m_Chart1.AddSeries(mySeries);
m_Chart1.AddSeries(mySeries);

// add a flat surface...all set to zero

 m_Chart1.GetSeries(0).Add(0,"",clTeeColor);
 m_Chart1.GetSeries(0).Add(0,"",clTeeColor);
 m_Chart1.GetSeries(0).Add(0,"",clTeeColor);
 m_Chart1.GetSeries(0).Add(0,"",clTeeColor);
 m_Chart1.GetSeries(0).Add(0,"",clTeeColor);
 m_Chart1.GetSeries(0).Add(0,"",clTeeColor);
 m_Chart1.GetSeries(0).Add(0,"",clTeeColor);
 m_Chart1.GetSeries(0).Add(0,"",clTeeColor);
 m_Chart1.GetSeries(0).Add(0,"",clTeeColor);
 m_Chart1.GetSeries(0).Add(0,"",clTeeColor);
 m_Chart1.GetSeries(0).Add(0,"",clTeeColor);
 m_Chart1.GetSeries(0).Add(0,"",clTeeColor);

// need next line to get rid of default TeeChart branding...
m_Chart1.GetHeader().GetText().Clear();

// colour each task
m_Chart1.GetSeries(0).SetColorEachPoint(true);

} // close loop checking really empty world
        return TRUE;
}

void CProcessionDlg::OnPaint()
{

// do Windows icon stuff...not crucial

        if (IsIconic())
        {
                CPaintDC dc(this); // device context for painting

                SendMessage(WM_ICONERASEBKGND, (WPARAM) dc.GetSafeHdc(), 0);

                // Center icon in client rectangle
                int cxIcon = GetSystemMetrics(SM_CXICON);
                int cyIcon = GetSystemMetrics(SM_CYICON);
                CRect rect;
                GetClientRect(&rect);
                int x = (rect.Width() - cxIcon + 1) / 2;
                int y = (rect.Height() - cyIcon + 1) / 2;

                // Draw the icon
```

```
		dc.DrawIcon(x, y, m_hIcon);
	}
	else
	{
		CDialog::OnPaint();
	}
}


// more Windows stuff...
HCURSOR CProcessionDlg::OnQueryDragIcon()
{
	return (HCURSOR) m_hIcon;
}


BEGIN_EVENTSINK_MAP(CProcessionDlg, CDialog)
   //{{AFX_EVENTSINK_MAP(CProcessionDlg)
	ON_EVENT(CProcessionDlg, IDC_TCHART1, 1 /* OnAfterDraw */,
OnOnAfterDrawTchart1, VTS_NONE)
	ON_EVENT(CProcessionDlg, IDC_TCHART1, 7 /* OnClickSeries */,
OnOnClickSeriesTchart1, VTS_I4 VTS_I4 VTS_I4 VTS_I4 VTS_I4 VTS_I4)
	ON_EVENT(CProcessionDlg, IDC_TCHART1, 31 /* OnTimer */,
OnOnTimerTchart1, VTS_NONE)
	//}}AFX_EVENTSINK_MAP
END_EVENTSINK_MAP()


// this is called after paint...
void CProcessionDlg::OnOnAfterDrawTchart1()
{

CString HUDrotation;
CString HUDelevation;
CString HUDtilt;
CString HUDhorizOffset;
CString HUDvertOffset;
CString HUDzoom;

// extra labels on axis...
// need to do them here because otherwise
// they'll get overpainted...
m_Chart1.GetCanvas().GetFont().SetColor(RGB(textRedValue,textGreenValue,textBlueValu
e));
m_Chart1.GetCanvas().GetFont().SetHeight(YaxisTitleFontSize+5);
m_Chart1.GetCanvas().TextOut(-11,195,"ahead");
m_Chart1.GetCanvas().TextOut(-11,-10,"behind");

// get 3d position and orientation and display on frame

_gcvt(int(m_Chart1.GetAspect().GetRotation()),10,buffer);
HUDrotation="rotation= ";
HUDrotation+=buffer;
m_rotation=HUDrotation;
```

```
_gcvt(int(m_Chart1.GetAspect().GetElevation()),10,buffer);
HUDelevation="elevation= ";
HUDelevation+=buffer;
m_elevation=HUDelevation;

_gcvt(int(m_Chart1.GetAspect().GetTilt()),10,buffer);
HUDtilt="tilt= ";
HUDtilt+=buffer;
m_tilt=HUDtilt;

_gcvt(int(m_Chart1.GetAspect().GetHorizOffset()),10,buffer);
HUDhorizOffset="horizontal offset= ";
HUDhorizOffset+=buffer;
m_horizontal=HUDhorizOffset;


_gcvt(int(m_Chart1.GetAspect().GetVertOffset()),10,buffer);
HUDvertOffset="vertical offset= ";
HUDvertOffset+=buffer;
m_vertical=HUDvertOffset;

_gcvt(int(m_Chart1.GetAspect().GetZoom()),10,buffer);
HUDzoom="zoom= ";
HUDzoom+=buffer;
m_zoom=HUDzoom;

// do this or never appears
UpdateData(FALSE);

}

// open a file from disk
void CProcessionDlg::OnFileOpen()
{

CFileDialog dlgFile(
TRUE,
_T(".dbf"),
NULL,
OFN_HIDEREADONLY | OFN_OVERWRITEPROMPT,
_T("DBF 4(dBase IV) (*.dbf)|*.dbf||"));

// if file not selected...back to normal
if (dlgFile.DoModal() != IDOK)
{
return;
}

// file is now open

// get the name and path of file
CString myDataBaseFile=dlgFile.GetFileName();
CString myDataBaseFilePath=dlgFile.GetPathName();

// go to default 3d viewpoint
OnHome();
```

```
// reset message at screen bottom
revealValue=statusBarDefaultMessage;

// clear both charts (second is a dummy)
m_Chart1.GetSeries(0).Clear();
m_Chart1.GetSeries(1).Clear();

// turn off task names because too intrusive
m_Chart1.GetAxis().GetBottom().GetLabels().SetVisible(false);

// get tbe task names from the relevant database field
m_Chart1.GetSeries(1).SetLabelsSource(CString("Task_Name"));

// set the Y axis values to the duration variance values from the database
m_Chart1.GetSeries(0).GetYValues().SetValueSource("DV");

// set the Y axis values to the cost variance values from the database
m_Chart1.GetSeries(1).GetYValues().SetValueSource("CV");

//NOTE: using Series(1) is a hack to avoid TeeChart bug.
// Can't get SetLabelsSource to find task names from Series(0).
// This is the only reason that there are two chart series.

// Allocate the ODBC DSN to each chart...procession is derived from the standard
// dBase 5 driver.
m_Chart1.GetSeries(0).SetDataSource("DSN=procession; TABLE="+myDataBaseFile);
m_Chart1.GetSeries(1).SetDataSource("DSN=procession; TABLE="+myDataBaseFile);

// make Series(1) invisible
m_Chart1.GetSeries(1).SetActive(false);

// update text box to show name of open file...instead of default
m_strLocalFileName=myDataBaseFile;
}


// this is called when a part of the 3d chart is left-clicked with the mouse
void CProcessionDlg::OnOnClickSeriesTchart1(long SeriesIndex, long ValueIndex, long
Button, long Shift, long X, long Y)
{
// get duration variance value at point clicked
CString
str_currentValue=m_Chart1.GetSeries(SeriesIndex).GetValueMarkText(ValueIndex);

// convert from a string to an integer
int currentValue=atoi(str_currentValue);

CString status="";

if(currentValue<0)
{
status="days AHEAD of schedule";
// value is minus but would look better
// as positive...i.e. 5 days ahead...rather than -5 days ahead
// so...wipe out the minus sign in string
```

```
LPTSTR withOutMinusSign=new TCHAR[str_currentValue.GetLength()+1];
_tcscpy(withOutMinusSign,str_currentValue);
withOutMinusSign[0]=' ';
str_currentValue=withOutMinusSign;
delete withOutMinusSign;
}
if(currentValue>0)
{
status="days BEHIND schedule";
}
if(currentValue==0)
{
status="days and therefore ON schedule";
}

// Display the task name and value in the frame
revealValue=
"This task is "+str_currentValue+" "+status+"\n"+
// use ValueIndex+1 because need to adjust array index 0 to task 1 etc..
"Task: "+m_Chart1.GetSeries(SeriesIndex).XValueToText(ValueIndex+1)+" "+

m_Chart1.GetSeries(1).GetPointLabel(ValueIndex);
m_strInfoWindow=revealValue;

// needs this or won't appear
UpdateData(FALSE);

}


// called each time animation timer cycles
void CProcessionDlg::OnOnTimerTchart1()
{
// if flythru has been requested...
if(flyThru)
{
// play next frame in animation...
m_Chart1.GetAspect().SetRotation(animationKeys[frameNumber][0]);
m_Chart1.GetAspect().SetElevation(animationKeys[frameNumber][1]);
m_Chart1.GetAspect().SetTilt(animationKeys[frameNumber][2]);
m_Chart1.GetAspect().SetHorizOffset(animationKeys[frameNumber][3]);
m_Chart1.GetAspect().SetVertOffset(animationKeys[frameNumber][4]);
m_Chart1.GetAspect().SetZoom(animationKeys[frameNumber][5]);

// if last frame
if(frameNumber==(maxFrame-1))
{
// start again...
frameNumber=0;
}
else
{
// go to next frame
frameNumber++;
}
}
```

```
// if surface animation required
if(animateSurface)
{
// clear existing stuff...
m_Chart1.GetSeries(0).Clear();
m_Chart1.GetSeries(1).Clear();

// generate random values...
m_Chart1.GetSeries(0).FillSampleValues(numberOfTasksInSimulatedAnimation);
// Putting in either of the following provides animation
// of entire surface in addition to change in surface values
//m_Chart1.GetAspect().SetRotation(m_Chart1.GetAspect().GetRotation()+10);
//m_Chart1.GetAspect().SetTilt(m_Chart1.GetAspect().GetTilt()+3);
}
}

// if animation button pushed
void CProcessionDlg::OnButton1()
{

if(m_Chart1.GetTimerEnabled()&&!flyThru&&animateSurface)
// this animation and the timer are running and the other isn't
{
// therefore user wants to stop this animation and the timer
m_Chart1.SetTimerEnabled(false);
animateSurface=false;
return; // break out to prevent retrigger
}

if (!m_Chart1.GetTimerEnabled()&&!flyThru&&!animateSurface)
// this animation is off and so is the other animation
{
// so user wants to turn on the timer and this animation
m_Chart1.SetTimerEnabled(true);
animateSurface=true;
return;
}

if(m_Chart1.GetTimerEnabled()&&flyThru&&animateSurface)
// everything is on!
{
// turn off this animation but leave timer for other one
animateSurface=false;
}
else
// the timer and other animation are on... this one isn't

{
// turn this one on...
animateSurface=true;
}

}

// if fly-thru button pushed
```

```
void CProcessionDlg::OnFlythru()
{
if(m_Chart1.GetTimerEnabled()&&flyThru&&!animateSurface)
// this animation and the timer are running and the other isn't
{
// therefore user wants to stop this animation and the timer
m_Chart1.SetTimerEnabled(false);
flyThru=false;
return; // break out to prevent retrigger
}

if (!m_Chart1.GetTimerEnabled()&&!flyThru&&!animateSurface)
// this animation is off and so is the other animation
{
// so user wants to turn on the timer and this animation
m_Chart1.SetTimerEnabled(true);
flyThru=true;
return;
}

if(m_Chart1.GetTimerEnabled()&&flyThru&&animateSurface)
// everything is on!
{
// turn off this animation but leave timer for other one
flyThru=false;
}
else
// the timer and other animation are on... this one isn't
{
// turn this one on...
flyThru=true;
}
}


// set viewpoint to default 'home' position
void CProcessionDlg::OnHome()
{
m_Chart1.GetAspect().SetRotation(19);
m_Chart1.GetAspect().SetElevation(346);
m_Chart1.GetAspect().SetTilt(0);
m_Chart1.GetAspect().SetHorizOffset(34);
m_Chart1.GetAspect().SetVertOffset(-49);
m_Chart1.GetAspect().SetZoom(112);
}

//rotate model around x axis
void CProcessionDlg::OnFlip()
{
m_Chart1.GetAspect().SetElevation(m_Chart1.GetAspect().GetElevation()-90);
}

//rotate model around y axis
void CProcessionDlg::OnTurn()
{
m_Chart1.GetAspect().SetRotation(m_Chart1.GetAspect().GetRotation()+90);
```

```
}

//rotate model around z axis
void CProcessionDlg::OnTilt()
{
m_Chart1.GetAspect().SetTilt(m_Chart1.GetAspect().GetTilt()+45);
}

// similar viewpoint to home but with no 3d perspective
void CProcessionDlg::OnStraighten()
{
m_Chart1.GetAspect().SetRotation(0);
m_Chart1.GetAspect().SetElevation(0);
m_Chart1.GetAspect().SetTilt(0);
m_Chart1.GetAspect().SetHorizOffset(18);
m_Chart1.GetAspect().SetVertOffset(-23);
m_Chart1.GetAspect().SetZoom(111);
}

// called when "get remote file" button is down
void CProcessionDlg::OnUrlsubmit()
{
// get contents of editable text boxes...for server name etc..
UpdateData();
// if there is a server name in the text box
if(m_srtURL.IsEmpty()==false)
{
UpdateData(false);
// get the local file directory and name into one string
m_strFTPLocalDirAndFile=m_strLocalODBCDataDirectory+m_strLocalFileName;
// get the remote file directory and name into one string
m_strFTPRemoteDirAndFile=m_strRemotePath+m_strRemoteFileName;

// instance net session and FTP session objects
CInternetSession session("Procession File Retrieval");
CFtpConnection* pConn=NULL;

m_strInfoWindow="Getting file from remote server...";
// last line needs next or won't show
UpdateData(false);

// initiate FTP object with the remote server, user and pass
pConn = session.GetFtpConnection(m_srtURL,m_strUserName,m_strPassWord);

// try FTPing the remote file into the local location
if (!pConn->GetFile(m_strFTPRemoteDirAndFile, m_strFTPLocalDirAndFile,false))
{
// if FTP fails...tell user
AfxMessageBox("Procession is unable to retrieve file");
// reset default status bar message
m_strInfoWindow=statusBarDefaultMessage;
// make status bar change visible
UpdateData(false);
return; // break out
} // end of loop if FTP fails
```

```
// only gets here if FTP ok
m_strInfoWindow=statusBarDefaultMessage;
UpdateData(false);

// tidy up
delete pConn;
session.Close();

// reset 3d to home viewpoint
OnHome();

// clear charts
m_Chart1.GetSeries(0).Clear();
m_Chart1.GetSeries(1).Clear();

// turn off task names
m_Chart1.GetAxis().GetBottom().GetLabels().SetVisible(false);

// set ODBC data source to file just FTP'd to local directory
m_Chart1.GetSeries(1).SetLabelsSource(CString("Task_Name"));
m_Chart1.GetSeries(1).GetYValues().SetValueSource("CV");
m_Chart1.GetSeries(0).GetYValues().SetValueSource("DV");
m_Chart1.GetSeries(0).SetDataSource("DSN=procession; TABLE="+m_strLocalFileName);
m_Chart1.GetSeries(1).SetDataSource("DSN=procession; TABLE="+m_strLocalFileName);

m_Chart1.GetSeries(1).SetActive(false);

}// close if loop for checking server name present
}

void CProcessionDlg::OnPrint()
{
// turn of OpenGL rendering because printing doesn't work unless on native
m_Chart1.GetAspect().GetOpenGL().SetActive(false);
m_Chart1.GetPrinter().PrintChart();
// instead of PrintChart...use ShowPreview allowing changes to print settings
//m_Chart1.GetPrinter().ShowPreview();
// turn OpenGL back on
m_Chart1.GetAspect().GetOpenGL().SetActive(true);
}

void CProcessionDlg::OnScreenshot()
{
// turn of OpenGL rendering because SaveToBitmapFile doesn't work unless on native
m_Chart1.GetAspect().GetOpenGL().SetActive(false);
// get latest file and dir from edit boxes
UpdateData(false);
// get current number of secs since 1970
time(&ltime);
// convert number to string
_gcvt(ltime,numberDigitsInSecsSince1970,theTime);
// use number of secs as a BMP file name
m_Chart1.GetExport().SaveToBitmapFile(m_strLocalODBCDataDirectory+theTime+"bmp");
CString imageName="Screenshot saved as ";
imageName+=m_strLocalODBCDataDirectory;
imageName+=theTime;
```

```
imageName+="bmp";
// tell the user the name of the file saved
AfxMessageBox(imageName);
// turn OpenGL back on
m_Chart1.GetAspect().GetOpenGL().SetActive(true);
}

void CProcessionDlg::OnViewdata()
{
// count the number of values in current data
// for each point create a string
for(int x=0;x<m_Chart1.GetSeries(0).GetCount();x++)
{
// use x+1 because need to adjust array index 0 to task 1 etc..
display+="Task "+m_Chart1.GetSeries(0).XValueToText(x+1)+": ";
display+=m_Chart1.GetSeries(1).GetPointLabel(x)+": ";
display+=m_Chart1.GetSeries(0).GetValueMarkText(x)+" days\n";
}
// launch a message box with a list of all data values
::MessageBox(::GetTopWindow(::GetDesktopWindow()),display,"View Data Source",
MB_OK);
// reset string for next time
display="";
}
```

Appendix 3: Microsoft Project export macro code in Visual Basic for Applications

```
Sub ExportProcession()
' Export to Procession 3d
Dim objExcel As Object

' Save project file in CSV format using Procession export map
FileSaveAs Name:="C:\windows\temp\tempProject.csv",
FormatID:="MSProject.CSV.8", map:="procession"

' open up a copy of Excel
Set objExcel = CreateObject("Excel.Application")

' open the CSV file in Excel
objExcel.Workbooks.OpenText
FileName:="C:\windows\temp\tempProject.csv", StartRow:=1, DataType:=1
' make Excel visible
objExcel.Visible = True
' make task names column very wide so that names
' don't get cut off
objExcel.Columns("A:A").ColumnWidth = 150
' make other columns wide enough for big numbers
objExcel.Columns("B:F").ColumnWidth = 30

' use the current time and date for dBase file name
Dim MyDate
MyTime = Time
MyDate = Date
theTime = Format(MyTime, "h_m_")
theDate = Format(MyDate, "dd_mmm_yyyy")
myFileName = theTime + theDate + ".dbf"
myPath = "C:\procession\data\" + myFileName
'objExcel.ActiveWorkbook.SaveAs
FileName:="C:\procession\data\Project1.dbf", FileFormat:=xlDBF4,
CreateBackup:=False
'objExcel.ActiveWorkbook.SaveAs FileName:=myPath, FileFormat:=xlDBF4,
CreateBackup:=False

MyAppID = Shell("C:\PROCESSION\PROCESSION.EXE", 1)
AppActivate MyAppID

End Sub
```

Appendix 4: Information provided to RSL Informants before experiment

Notes on interview process:

Two fictional 'scenarios' have been produced, based on the real project described in 'Notes on project for RSL Development Officers'.

One of these scenarios will be used at each visit.

1. At the first of the two visits only, you will be asked questions about how you currently get project information.

For both visits the remainder will apply:

2. You will be shown an s-curve chart of a project scenario.
Earned Value Analysis provides the budgeted cost of completing the tasks actually finished at an observation point. This sum is the 'earned value'. Comparing this with the amount actually spent, is a good indicator of project performance. There is an acronym for each of the three s-curves used:

- ACWP (Actual Cost of Work Performed. Actual cost of performing the completed tasks)
- BCWS (Budgeted Cost of Work Scheduled. The planned budget for the scheduled tasks)
- BCWP (Budgeted Cost of Work Performed. The planned budget for the tasks that were actually completed- the earned value).

Hints:

All three curves should correspond if all is well with project. This indicates that the budgeted amount of money has been spent to achieve the budgeted amount of work.

If ACWP is greater than BCWP at point of observation, the difference is the Cost Variance. This means that the work completed is over-budget.

If BCWP is less than BCWS, the difference is the Schedule Variance. This indicates schedule slippage, because it uses the budgeted costs as a progress indicator i.e. if the budget for painting a wall is 2 days and £10, at the end of day 1, BCWP and BCWS should both be £5. If BCWP is less than BCWS (£5), then the schedule is behind.

In summary:

BCWP can be seen as the baseline against which performance is measured
ACWP greater than BCWP = over budget
ACWP less than BCWP = under budget
BCWS greater than BCWP = behind schedule
BCWS less than BCWP = ahead of schedule

3. You will be asked to choose a description that best fits the condition of the project, at a specific milestone.

4. You will run through several of the milestones using the software prototype.

5. You will be asked AGAIN to choose a description that NOW best fits the condition of the project, at a specific milestone.

6. You will run through the remaining milestones using the software prototype.

7. You will be asked questions about the information provided by the software prototype.

8. You will be asked questions about the usability of the software.

Notes on project:

1. Introduction
The fictional project scenarios are based on a real 19 month project to refurbish flats and houses on an ex-local authority estate in London.
The scheme objectives were:
to provide new and clearly defined street patterns,
to enlarge the gardens
and
to remove crime-ridden pedestrian routes.

2. History
Following a vote by tenants in the late 1990s, management of the featured estate was transferred from an inner London Borough to an RSL, under the ERCF (Estate Renewal Challenge Fund). This project represents one phase of the continuing ERCF funded improvements.

3. Unit Profile
Total estate has 517 units. This project focuses on 262 of these units (123 houses and 129 flats).
Unit types include:
Groups of 10,8,7,6,5 and 4 houses
Blocks of 15,14,10 and 9 flats

4. Tenant Issues
Work on the houses will be conducted without decanting most of the tenants.

5. House Works
A unique aspect of the project is the 'porch turnarounds' on the houses. This entails making the back of the house into a new entrance porch (brick and glass, zinc roof). The houses are given enlarged back gardens by gains made from old pedestrian ways. The new front garden has a parking space.

6. Flat Works
Flats receive basic internal refurbishment, including all kitchen fittings, vinyl kitchen flooring, decorations (emulsion- 2 coats and gloss primer-2 coats).

7. Estate Works
Extensive landscaping (turf, trees, re-route of services (gas and drainage) and new street patterns (pavements, 2 lane 7.3m wide urban roads and improved lighting).

8. Costs
The budgeted total cost before the project starts is £6,247,849.

9. Milestone Progress Reports

When running through the research scenarios, RSL development workers will receive project progress reports at the following milestone intervals:

Milestone 1 end of 1st quarter year 1 (actually 2 months after start on-site)
Milestone 2 end of 2nd quarter year 1 (31st March 2000
Milestone 3 end of 3rd quarter year 1 (30th June 2000)
Milestone 4 end of 4th quarter year 1 (30th September 2000)
Milestone 5 end of 1st quarter year 2 (31st December 2000)
Milestone 6 end of 2nd quarter year 2 (31st March 2001)
Milestone 7 end of 3rd quarter year 2 (30th June 2001)

Appendix 5: Client Information Requirement Protocol

Informant's Designated Reference Number:

Informant's Name:
Date:

RSL/Housing Association:

Section A. Perceived quality of information provided by CURRENT METHOD of progress reporting.

NOTE: Use this protocol BEFORE the RSL Informant has seen the Initial Prototype.
NOTE: The first four questions only apply to the current method. These have been numbered below 1.0, in order to maintain question number consistency across all three protocols.

A0.20. In general, how do you rate your experience and confidence in using computers? (1-5)

LOW — HIGH

1 2 3 4 5

Inexperienced and not confident — Experienced and confident

A0.40. Please indicate the TECHNICAL COMPLEXITY of the MAIN method currently providing you with project progress reports? (1-5)

LOW — HIGH

1 2 3 4 5

Verbal (chat, meeting, phone) — Paper (reports, letters) — Computer (email, file sharing)

TOTAL SCORE THIS PAGE: ....

A0.60. Please indicate the MAIN TYPE of information provided in your current project progress reports? (1-5)

LOW HIGH

1 2 3 4 5

Descriptive (i.e. site report- verbal or written)	Numerical (i.e. table)	Graphical (graphs, s-curves)

A0.80. Please briefly describe ALL of the method(s) by which you currently receive project progress reports (i.e. paper, verbal, phone, fax, email etc..):

....

....

Please score the following aspects of your CURRENT method of progress reporting, in terms of your level of satisfaction (1-5, where 1 is completely dissatisfied and 5 is extremely satisfied):

A1. The length of TIME taken to receive a report? (1-5)

LOW HIGH

1 2 3 4 5

Dissatisfied with time taken ... Very satisfied with time taken

A2. The QUALITY of information received? (1-5)

LOW HIGH

1 2 3 4 5

Dissatisfied with quality of info ... Very satisfied with quality of info

TOTAL SCORE THIS PAGE: ....

A3. Your confidence in the ACCURACY of the information? (1-5)

LOW HIGH
1 2 3 4 5

Not confident in accuracy   Very confident in accuracy

A4. The RELEVANCE of the information provided? (1-5)

LOW HIGH
1 2 3 4 5

Dissatisfied with relevance   Very satisfied with relevance

A5. The INTELLIGIBILITY/ACCESSIBILITY of the information- i.e. do you understand it? (1-5)

LOW HIGH
1 2 3 4 5

Don't understand information   Good understanding of info

A6. The LEVEL OF DETAIL provided? (1-5)

LOW HIGH
1 2 3 4 5

Dissatisfied with detail provided   Very satisfied with detail

A7. The FORMAT of the report? (1-5)

LOW HIGH
1 2 3 4 5

Dissatisfied with format   Very satisfied with format

TOTAL SCORE THIS PAGE: ....

A8. Do you have any other comments about the reports provided by your CURRENT method?

....

....

....

A9. Is there information not currently provided by these progress reports that you would find useful (please detail)?

....

....

....

A10. Please describe your overall impressions of attempting to obtain information by your CURRENT progress reporting method.

....

....

TOTAL SCORE THIS SECTION: ....

TOTAL SCORE THIS INTERVIEW: ....

Appendix 6: Initial Prototype Usability and Information Evaluation Protocol

Date:..

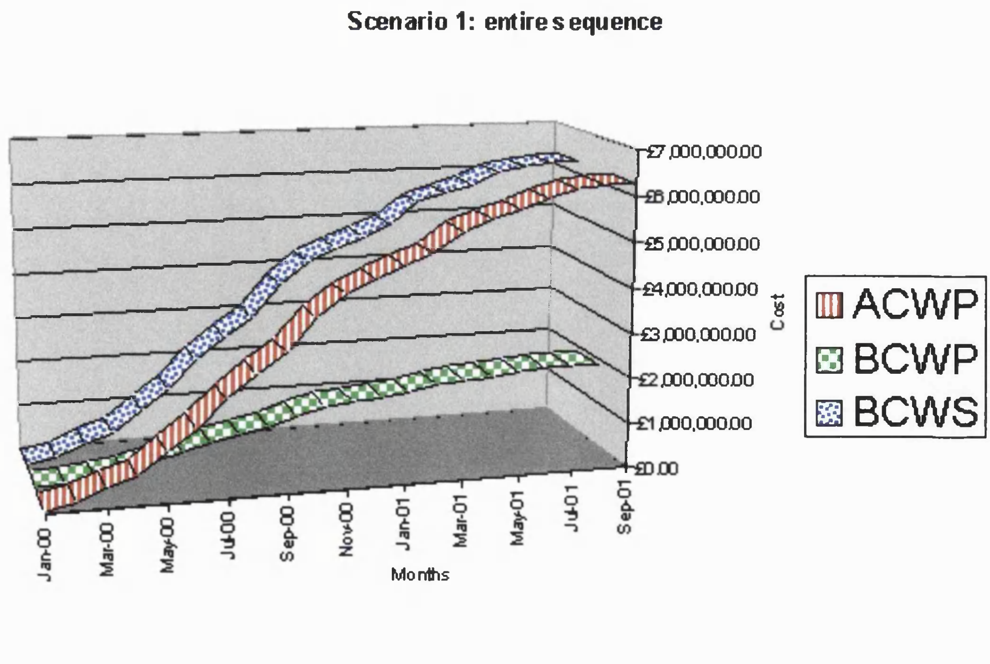
Informant's Designated Reference Number:...

Informant's Name:..

RSL/Housing Association:..

Section A. Perceived quality of information provided by the PROTOTYPE.

Note: First question to be completed BEFORE the simulated project but AFTER showing the earned value s-curve for scenario 1 (below) to the RSL Informant.



A0.10 Considering ONLY the information provided in the s-curve (above), please select ONE status code from the list located at the end of this interview. Your choice should describe your impression of scenario 1's status, as at milestone 4 (December 2000).

(For example, if you think that Scenario 1 is "Under budget and ahead of schedule, with schedule performance improving and cost performance stabilised", then your selected status code should be: 3.3.)

Status code: ....

Note: Researcher to score accuracy post-interview (1-5)

LOW — HIGH

1 2 3 4 5

Impression of status inaccurate — Impression of status accurate

With reference to the s-curve, please describe your impressions of scenario 1's status. Did the s-curve reveal any specific issues in the project that you would consider significant, or worthy of concern?

....

....

....

....

TOTAL SCORE THIS PAGE: ....

NOTE: Run through scenario 1 using initial prototype at all seven milestones.

Stop at milestone 4 and ask question A0.30.

Researcher's observational notes, taken while RSL Informant is using the initial prototype:

....

....

....

....

....

....

....

....

....

....

....

NOTE: To be completed immediately AFTER showing milestone 4 of the simulated project.

A0.30 Considering ONLY the information provided by the software tool, for the second time please select ONE status from the list located at the end of this interview. Your choice should describe your impression of scenario 1's status, as at milestone 4 (December 2000).
(Please base your response on what you've just seen, rather than your previous response for the s-curve.)

Status code: ....

Note: Researcher to score accuracy post-interview (1-5)

LOW HIGH
1 2 3 4 5

Impression of status inaccurate Impression of status accurate

With reference to the software prototype, please describe your impressions of scenario 1's status. Did the prototype reveal any specific issues in the project that you would consider significant, or worthy of concern?

....

....

....

....

TOTAL SCORE THIS PAGE: ....

Please score the following aspects of the progress report, in terms of your level of satisfaction (1-5, where 1 is completely dissatisfied and 5 is extremely satisfied):

A1. The length of TIME taken to receive the report? (1-5)

LOW HIGH

1 2 3 4 5

Dissatisfied with time taken | Very satisfied with time taken

A2. The QUALITY of information received? (1-5)

LOW HIGH

1 2 3 4 5

Dissatisfied with quality of info | Very satisfied with quality of info

A3. Your confidence in the ACCURACY of the information? (1-5)

LOW HIGH

1 2 3 4 5

Not confident in accuracy | Very confident in accuracy

A4. The RELEVANCE of the information provided? (1-5)

LOW HIGH

1 2 3 4 5

Dissatisfied with relevance | Very satisfied with relevance

TOTAL SCORE THIS PAGE: ....

A5. The INTELLIGIBILITY/ACCESSIBILITY of the information- i.e. do you understand it? (1-5)

LOW — HIGH

1 2 3 4 5

Don't understand information — Good understanding of info

A6. The LEVEL OF DETAIL provided? (1-5)

LOW — HIGH

1 2 3 4 5

Dissatisfied with detail provided — Very satisfied with detail

A7. The FORMAT of the report? (1-5)

LOW — HIGH

1 2 3 4 5

Dissatisfied with format — Very satisfied with format

A8. Do you have any other comments about the progress reports provided by the Prototype?

....

....

....

TOTAL SCORE THIS PAGE: ....

A9. Is there information not currently provided by these progress reports that you would find useful (please detail)?

....

....

....

A10. Please describe your overall impressions of this attempt to obtain information.

....

....

....

TOTAL SCORE THIS SECTION: ....

Section B. Usability of software

NOTE: To be completed AFTER the project scenario.

Please score the following for the software's ability (1-5, where 1 is low and 5 is high):

B1. To provide a consistency of look and feel i.e. does the software have a consistent use of messages, menus, windows etc., throughout its function (1-5).

LOW HIGH

1 2 3 4 5

Not consistent Very consistent

Comments: ....

....

....

TOTAL SCORE THIS PAGE: ....

B2. To display information in a manner which matches your own way of thinking (1-5).

LOW HIGH

1 2 3 4 5

Info matches your thinking Info doesn't match your thinking

Comments: ....

....

....

B3. To fit in with your daily working when requiring project information (1-5).

LOW HIGH

1 2 3 4 5

Doesn't fit daily working Exactly fits daily working

Comments: ....

....

....

TOTAL SCORE THIS PAGE: ....

B4. To use language and terms that you are familiar with (1-5).

LOW HIGH

1 2 3 4 5

Unfamiliar language Familiar language

Comments: ....

....

....

B5. To do everything that you expected it to do (1-5).

LOW HIGH

1 2 3 4 5

Does nothing expected Does everything expected

Comments: ....

....

....

TOTAL SCORE THIS PAGE: ....

B6. To work in all situations where project information is required (1-5).

LOW HIGH

1 2 3 4 5

Works in no situations Works in all situations

Comments: ....

....

....

B7. To make it easier and quicker to obtain project information (1-5).

LOW HIGH

1 2 3 4 5

Not easy and quick Extremely easy and quick

Comments: ....

....

....

TOTAL SCORE THIS PAGE: ....

B8. To seem like a logical step from your previous way of obtaining information (1-5).

LOW HIGH
1 2 3 4 5

Not logical step from previous | Very logical step from previous

Comments: ....

....

....

B9. To provide the necessary help, messages and warnings to allow efficient working (1-5).

LOW HIGH
1 2 3 4 5

Doesn't provide necessary help | Provides necessary help

Comments: ....

....

....

TOTAL SCORE THIS PAGE: ....

B10. To provide a satisfactory Installation Process (1-5).

LOW				HIGH
1	2	3	4	5

Unsatisfactory installation — Very satisfactory installation

Comments: ....

....

....

B11. To provide Menu Bar categories which are logical and intelligible (1-5).

LOW				HIGH
1	2	3	4	5

Bad menu bars categories — Good menu bars categories

Comments: ....

....

....

TOTAL SCORE THIS PAGE: ....

B12. To provide Menus Items and Sub-Items which are logical and intelligible (1-5).

LOW HIGH

1 2 3 4 5

Bad menu items Good menu items

Comments: ....

....

....

B13. To utilise Text and Fonts which are legible and intelligible (1-5).

LOW HIGH

1 2 3 4 5

Bad use of text messages Good use of text messages

Comments: ....

....

....

TOTAL SCORE THIS PAGE: ....

B14. To provide an Application Window and Multiple Sub-Windows that behave as you would expect them to (1-5).

LOW				HIGH
1	2	3	4	5

Bad window functionality — Good window functionality

Comments: ....

....

....

B15. To provide a system of 3D navigation that you find suitable (1-5).

LOW				HIGH
1	2	3	4	5

Unsuitable 3D navigation — Excellent 3D Navigation

Comments: ....

....

....

TOTAL SCORE THIS PAGE: ....

B16. To behave in an expected and logical manner when presenting Dialog Box choices (1-5).

LOW HIGH

1 2 3 4 5

Bad dialog boxes Good dialog boxes

Comments: ....

....

....

B17. To make use of interface Colours in a manner which enables and supports. (1-5).

LOW HIGH

1 2 3 4 5

Bad colour use Good colour use

Comments: ....

....

....

TOTAL SCORE THIS PAGE: ....

B18. To function in a way which is consistent with other software familiar to you (1-5).

LOW ... HIGH

1 2 3 4 5

Not like other software — Very like other software

Comments: ....

....

....

TOTAL SCORE THIS PAGE: ....

TOTAL SCORE THIS SECTION: ....

TOTAL SCORE THIS INTERVIEW: ....

1.0 Under budget behind schedule

1.1 Under budget behind schedule- schedule performance improving and cost performance improving
1.2 Under budget behind schedule-schedule performance improving and cost performance deteriorating
1.3 Under budget behind schedule-schedule performance improving and cost performance stabilised
1.4 Under budget behind schedule-schedule performance deteriorating and cost performance improving
1.5 Under budget behind schedule-schedule performance deteriorating and cost performance deteriorating
1.6 Under budget behind schedule-schedule performance deteriorating and cost performance stabilised
1.7 Under budget behind schedule-schedule performance stabilised and cost performance improving
1.8 Under budget behind schedule-schedule performance stabilised and cost performance deteriorating
1.9 Under budget behind schedule-schedule performance stabilised and cost performance stabilised

2.0 Under budget on schedule

2.1 Under budget on schedule- schedule performance improving and cost performance improving
2.2 Under budget on schedule- schedule performance improving and cost performance deteriorating
2.3 Under budget on schedule- schedule performance improving and cost performance stabilised
2.4 Under budget on schedule- schedule performance deteriorating and cost performance improving
2.5 Under budget on schedule- schedule performance deteriorating and cost performance deteriorating
2.6 Under budget on schedule- schedule performance deteriorating and cost performance stabilised
2.7 Under budget on schedule- schedule performance stabilised and cost performance improving
2.8 Under budget on schedule- schedule performance stabilised and cost performance deteriorating
2.9 Under budget on schedule- schedule performance stabilised and cost performance stabilised

3.0 **Under budget ahead of schedule**

3.1 Under budget ahead of schedule- schedule performance improving and cost performance improving
3.2 Under budget ahead of schedule- schedule performance improving and cost performance deteriorating
3.3 Under budget ahead of schedule- schedule performance improving and cost performance stabilised
3.4 Under budget ahead of schedule- schedule performance deteriorating and cost performance improving
3.5 Under budget ahead of schedule- schedule performance deteriorating and cost performance deteriorating
3.6 Under budget ahead of schedule- schedule performance deteriorating and cost performance stabilised
3.7 Under budget ahead of schedule- schedule performance stabilised and cost performance improving
3.8 Under budget ahead of schedule- schedule performance stabilised and cost performance deteriorating
3.9 Under budget ahead of schedule- schedule performance stabilised and cost performance stabilised

4.0 On budget behind schedule

- 4.1 On budget behind schedule- schedule performance improving and cost performance improving
- 4.2 On budget behind schedule- schedule performance improving and cost performance deteriorating
- 4.3 On budget behind schedule- schedule performance improving and cost performance stabilised
- 4.4 On budget behind schedule- schedule performance deteriorating and cost performance improving
- 4.5 On budget behind schedule- schedule performance deteriorating and cost performance deteriorating
- 4.6 On budget behind schedule- schedule performance deteriorating and cost performance stabilised
- 4.7 On budget behind schedule- schedule performance stabilised and cost performance improving
- 4.8 On budget behind schedule- schedule performance stabilised and cost performance deteriorating
- 4.9 On budget behind schedule- schedule performance stabilised and cost performance stabilised

5.0 On budget on schedule

- 5.1 On budget on schedule- schedule performance improving and cost performance improving
- 5.2 On budget on schedule- schedule performance improving and cost performance deteriorating
- 5.3 On budget on schedule- schedule performance improving and cost performance stabilised
- 5.4 On budget on schedule- schedule performance deteriorating and cost performance improving
- 5.5 On budget on schedule- schedule performance deteriorating and cost performance deteriorating
- 5.6 On budget on schedule- schedule performance deteriorating and cost performance stabilised
- 5.7 On budget on schedule- schedule performance stabilised and cost performance improving
- 5.8 On budget on schedule- schedule performance stabilised and cost performance deteriorating
- 5.9 On budget on schedule- schedule performance stabilised and cost performance stabilised

6.0 On budget ahead of schedule

- 6.1 On budget ahead of schedule- schedule performance improving and cost performance improving
- 6.2 On budget ahead of schedule- schedule performance improving and cost performance deteriorating
- 6.3 On budget ahead of schedule- schedule performance improving and cost performance stabilised
- 6.4 On budget ahead of schedule- schedule performance deteriorating and cost performance improving
- 6.5 On budget ahead of schedule- schedule performance deteriorating and cost performance deteriorating
- 6.6 On budget ahead of schedule- schedule performance deteriorating and cost performance stabilised
- 6.7 On budget ahead of schedule- schedule performance stabilised and cost performance improving
- 6.8 On budget ahead of schedule- schedule performance stabilised and cost performance deteriorating
- 6.9 On budget ahead of schedule- schedule performance stabilised and cost performance stabilised

7.0 Over budget behind schedule

7.1 Over budget behind schedule- schedule performance improving and cost performance improving
7.2 Over budget behind schedule- schedule performance improving and cost performance deteriorating
7.3 Over budget behind schedule- schedule performance improving and cost performance stabilised
7.4 Over budget behind schedule- schedule performance deteriorating and cost performance improving
7.5 Over budget behind schedule- schedule performance deteriorating and cost performance deteriorating
7.6 Over budget behind schedule- schedule performance deteriorating and cost performance stabilised
7.7 Over budget behind schedule- schedule performance stabilised and cost performance improving
7.8 Over budget behind schedule- schedule performance stabilised and cost performance deteriorating
7.9 Over budget behind schedule- schedule performance stabilised and cost performance stabilised

8.0 Over budget on schedule

8.1 Over budget on schedule- schedule performance improving and cost performance improving
8.2 Over budget on schedule- schedule performance improving and cost performance deteriorating
8.3 Over budget on schedule- schedule performance improving and cost performance stabilised
8.4 Over budget on schedule- schedule performance deteriorating and cost performance improving
8.5 Over budget on schedule- schedule performance deteriorating and cost performance deteriorating
8.6 Over budget on schedule- schedule performance deteriorating and cost performance stabilised
8.7 Over budget on schedule- schedule performance stabilised and cost performance improving
8.8 Over budget on schedule- schedule performance stabilised and cost performance deteriorating

9.0 Over budget ahead of schedule

9.1 Over budget ahead of schedule- schedule performance improving and cost performance improving
9.2 Over budget ahead of schedule- schedule performance improving and cost performance deteriorating
9.3 Over budget ahead of schedule- schedule performance improving and cost performance stabilised
9.4 Over budget ahead of schedule- schedule performance deteriorating and cost performance improving
9.5 Over budget ahead of schedule- schedule performance deteriorating and cost performance deteriorating
9.6 Over budget ahead of schedule- schedule performance deteriorating and cost performance stabilised
9.7 Over budget ahead of schedule- schedule performance stabilised and cost performance improving
9.8 Over budget ahead of schedule- schedule performance stabilised and cost performance deteriorating
9.9 Over budget ahead of schedule- schedule performance stabilised and cost performance stabilised

Appendix 7: Operational Prototype Usability and Information Evaluation Protocol

Date:..

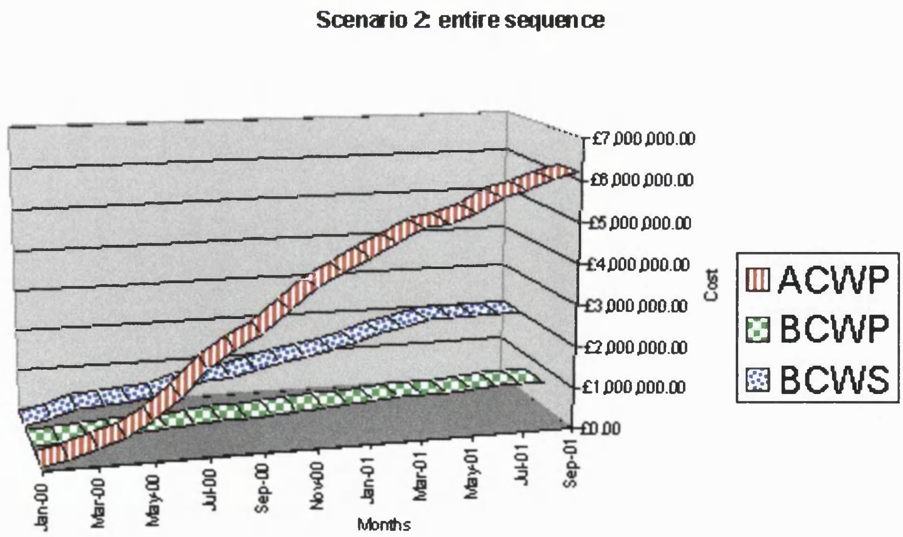
Informant's Designated Reference Number:..

Informant's Name:...

RSL/Housing Association:...

Section A. Perceived quality of information provided by the PROTOTYPE.

Note: First question to be completed BEFORE the simulated project but AFTER showing the earned value s-curve for scenario 2 (below) to the RSL Informant.



A0.10 Considering ONLY the information provided in the s-curve (above), please select ONE status code from the list located at the end of this interview. Your choice should describe your impression of scenario 2's status, as at milestone 4 (December 2000).

(For example, if you think that Scenario 2 is "Under budget and ahead of schedule, with schedule performance improving and cost performance stabilised", then your selected status code should be: 3.3.)

Status code: ....
Note: Researcher to score accuracy post-interview (1-5)

LOW HIGH
1 2 3 4 5

Impression of status inaccurate Impression of status accurate

With reference to the s-curve, please describe your impressions of scenario 2's status. Did the s-curve reveal any specific issues in the project that you would consider significant, or worthy of concern?

....

....

....

....

TOTAL SCORE THIS PAGE: ....

NOTE: Run through scenario 2 using initial prototype at all seven milestones.

Stop at milestone 4 and ask question A0.30.

Researcher's observational notes, taken while RSL Informant is using the initial prototype:

....

....

....

....

....

....

....

....

....

....

....

NOTE: To be completed immediately AFTER showing milestone 4 of the simulated project.

A0.30 Considering ONLY the information provided by the software tool, for the second time please select ONE status from the list located at the end of this interview. Your choice should describe your impression of scenario 2's status, as at milestone 4 (December 2000).
(Please base your response on what you've just seen, rather than your previous response for the s-curve.)

Status code: ....

Note: Researcher to score accuracy post-interview (1-5)

LOW HIGH

1 2 3 4 5

Impression of status inaccurate Impression of status accurate

With reference to the software prototype, please describe your impressions of scenario 2's status. Did the prototype reveal any specific issues in the project that you would consider significant, or worthy of concern?

....

....

....

....

TOTAL SCORE THIS PAGE: ....

Please score the following aspects of the progress report, in terms of your level of satisfaction (1-5, where 1 is completely dissatisfied and 5 is extremely satisfied):

A1. The length of TIME taken to receive the report? (1-5)

LOW HIGH

1 2 3 4 5

Dissatisfied with time taken Very satisfied with time taken

A2. The QUALITY of information received? (1-5)

LOW HIGH

1 2 3 4 5

Dissatisfied with quality of info Very satisfied with quality of info

A3. Your confidence in the ACCURACY of the information? (1-5)

LOW HIGH

1 2 3 4 5

Not confident in accuracy Very confident in accuracy

A4. The RELEVANCE of the information provided? (1-5)

LOW HIGH

1 2 3 4 5

Dissatisfied with relevance Very satisfied with relevance

TOTAL SCORE THIS PAGE: ....

A5. The INTELLIGIBILITY/ACCESSIBILITY of the information- i.e. do you understand it? (1-5)

LOW HIGH

1 2 3 4 5

Don't understand information Good understanding of info

A6. The LEVEL OF DETAIL provided? (1-5)

LOW HIGH

1 2 3 4 5

Dissatisfied with detail provided Very satisfied with detail

A7. The FORMAT of the report? (1-5)

LOW HIGH

1 2 3 4 5

Dissatisfied with format Very satisfied with format

A8. Do you have any other comments about the progress reports provided by the Prototype?

....

....

....

TOTAL SCORE THIS PAGE: ....

A9. Is there information not currently provided by these progress reports that you would find useful (please detail)?

....

....

....

A10. Please describe your overall impressions of this attempt to obtain information.

....

....

....

TOTAL SCORE THIS SECTION: ....

Section B. Usability of software

NOTE: To be completed AFTER the project scenario.

Please score the following for the software's ability (1-5, where 1 is low and 5 is high):

B1. To provide a consistency of look and feel i.e. does the software have a consistent use of messages, menus, windows etc., throughout its function (1-5).

LOW HIGH

1 2 3 4 5

Not consistent Very consistent

Comments: ....

....

....

TOTAL SCORE THIS PAGE: ....

B2. To display information in a manner which matches your own way of thinking (1-5).

LOW HIGH

1 2 3 4 5

Info matches your thinking Info doesn't match your thinking

Comments: ....

....

....

B3. To fit in with your daily working when requiring project information (1-5).

LOW HIGH

1 2 3 4 5

Doesn't fit daily working Exactly fits daily working

Comments: ....

....

....

TOTAL SCORE THIS PAGE: ....

B4. To use language and terms that you are familiar with (1-5).

LOW				HIGH
1	2	3	4	5

Unfamiliar language   Familiar language

Comments:

....

....

....

B5. To do everything that you expected it to do (1-5).

LOW				HIGH
1	2	3	4	5

Does nothing expected   Does everything expected

Comments: ....

....

....

TOTAL SCORE THIS PAGE: ....

B6. To work in all situations where project information is required (1-5).

LOW HIGH

1 2 3 4 5

Works in no situations Works in all situations

Comments: ....

....

....

B7. To make it easier and quicker to obtain project information (1-5).

LOW HIGH

1 2 3 4 5

Not easy and quick Extremely easy and quick

Comments: ....

....

....

TOTAL SCORE THIS PAGE: ....

B8. To seem like a logical step from your previous way of obtaining information (1-5).

LOW — HIGH
1 2 3 4 5

Not logical step from previous — Very logical step from previous

Comments: ....

....

....

B9. To provide the necessary help, messages and warnings to allow efficient working (1-5).

LOW — HIGH
1 2 3 4 5

Doesn't provide necessary help — Provides necessary help

Comments: ....

....

....

TOTAL SCORE THIS PAGE: ....

B10. To provide a satisfactory Installation Process (1-5).

LOW HIGH

1 2 3 4 5

Unsatisfactory installation | Very satisfactory installation

Comments: ....

....

....

B11. To provide Menu Bar categories which are logical and intelligible (1-5).

LOW HIGH

1 2 3 4 5

Bad menu bars categories | Good menu bars categories

Comments: ....

....

....

TOTAL SCORE THIS PAGE: ....

B12. To provide Menus Items and Sub-Items which are logical and intelligible (1-5).

LOW HIGH

1 2 3 4 5

Bad menu items Good menu items

Comments: ....

....

....

B13. To utilise Text and Fonts which are legible and intelligible (1-5).

LOW HIGH

1 2 3 4 5

Bad use of text messages Good use of text messages

Comments: ....

....

....

TOTAL SCORE THIS PAGE: ....

B14. To provide an Application Window and Multiple Sub-Windows that behave as you would expect them to (1-5).

LOW HIGH

1 2 3 4 5

Bad window functionality Good window functionality

Comments: ....

....

....

B15. To provide a system of 3D navigation that you find suitable (1-5).

LOW HIGH

1 2 3 4 5

Unsuitable 3D navigation Excellent 3D Navigation

Comments: ....

....

....

TOTAL SCORE THIS PAGE: ....

B16. To behave in an expected and logical manner when presenting Dialog Box choices (1-5).

LOW HIGH

1 2 3 4 5

Bad dialog boxes Good dialog boxes

Comments: ....

....

....

B17. To make use of interface Colours in a manner which enables and supports. (1-5).

LOW HIGH

1 2 3 4 5

Bad colour use Good colour use

Comments: ....

....

....

TOTAL SCORE THIS PAGE: ....

B18. To function in a way which is consistent with other software familiar to you (1-5).

LOW — HIGH

1 2 3 4 5

Not like other software — Very like other software

Comments: ....

....

....

TOTAL SCORE THIS PAGE: ....

TOTAL SCORE THIS SECTION: ....

TOTAL SCORE THIS INTERVIEW: ....

Appendix 8: Results data set

Client Information Requirement Protocol

RSL Informants

Questions	Section	Subsection	Question	1	2	3	4	5	Q Totals	Q Means	Sec Totals	Sec Means	Sec Mean Tots
Questions	A	computer familia-arity	A0.20	3.00	5.00	3.00	2.00	5.00	18.00	3.60			
			A0.40	3.00	5.00	5.00	5.00	4.00	22.00	4.40			
			A0.60	2.00	3.00	5.00	1.00	2.00	13.00	2.60			
											53.00	3.53	10.60
		info quality	A1	4.00	2.00	4.00	5.00	3.00	18.00	3.60			
			A2	4.00	3.00	3.00	4.00	2.00	16.00	2.00			
			A3	4.00	4.00	3.00	4.00	3.00	18.00	3.60			
			A4	5.00	4.00	4.00	4.00	2.00	19.00	3.80			
			A5	5.00	5.00	3.00	4.00	4.00	21.00	4.20			
			A6	4.00	4.00	3.00	4.00	3.00	18.00	3.60			
			A7	4.00	4.00	4.00	4.00	2.00	18.00	3.60			
											128.00	3.49	24.40
			Score Totals	38.00	39.00	37.00	37.00	30.00	181.00		181.00		
			RSL Means	3.80	3.90	3.70	3.70	3.00					
			Mean Totals							35.00			35.00
			Mean Means				3.62		3.62	3.50	3.62		3.50

Number of RSL informants:	5
Number of questions:	10
Number of sections:	2
Number of scores:	50

Initial Prototype Usability and Information Evaluation Protocol

RSL Informants

			1	2	3	4	5	Q Totals	Q Means	Sec Totals	Sec Means	Sec Mean Tots
A	project status assess.	A0.10	2.00	5.00	3.00	1.00	1.00	12.00	2.40			
		A0.30	2.00	5.00	4.00	1.00	4.00	16.00	3.20			
										28.00	2.80	5.60
	info quality	A1	1.00	5.00	4.00	4.00	4.00	18.00	3.60			
		A2	1.00	3.00	4.00	4.00	3.00	15.00	1.88			
		A3	4.00	3.00	3.00	4.00	5.00	19.00	3.80			
		A4	3.00	3.00	4.00	4.00	2.00	16.00	3.20			
		A5	1.00	2.00	4.00	4.00	2.00	13.00	2.60			
		A6	2.00	4.00	4.00	4.00	4.00	18.00	3.60			
		A7	1.00	2.00	4.00	4.00	3.00	14.00	2.80			
										113.00	3.07	21.48
B	USABILITY	B1	3.00	4.00	4.00	4.00	3.00	18.00	3.60			
		B2	1.00	3.00	4.00	3.00	3.00	14.00	2.80			
		B3	1.00	4.00	4.00	3.00	3.00	15.00	3.00			
		B4	1.00	4.00	3.00	2.00	3.00	13.00	2.60			
		B5	2.00	3.00	3.00	3.00	3.00	14.00	2.80			
		B6	2.00	4.00	3.00	4.00	3.00	16.00	3.20			
		B7	1.00	4.00	4.00	4.00	3.00	16.00	3.20			
		B8	1.00	4.00	3.00	4.00	3.00	15.00	3.00			
		B9	2.00	3.00	3.00	3.00	3.00	14.00	2.80			
		B10	4.00	4.00	4.00	4.00	3.00	19.00	3.80			
		B11	4.00	4.00	3.00	4.00	3.00	18.00	3.60			
		B12	4.00	4.00	3.00	3.00	3.00	17.00	3.40			
		B13	5.00	3.00	3.00	3.00	3.00	17.00	3.40			
		B14	4.00	4.00	3.00	3.00	3.00	17.00	3.40			
		B15	3.00	4.00	4.00	4.00	3.00	18.00	3.60			
		B16	5.00	4.00	3.00	3.00	3.00	18.00	3.60			
		B17	5.00	2.00	4.00	4.00	3.00	18.00	3.60			
		B18	4.00	3.00	4.00	4.00	3.00	18.00	3.60			
										295.00	3.28	59.00
		Score Totals	69.00	97.00	96.00	92.00	82.00	436.00		436.00		
		RSL Means	2.56	3.59	3.56	3.41	3.04					
		Mean Totals							86.08			86.08
		Mean Means				3.23		3.23	3.19	3.23		3.19

No. of RSL informants: 5
No. of questions: 27
No. of sections: 3
No. of scores: 135

Questions

Operational Prototype Usability and Information Evaluation Protocol

RSL Informants

			1	2	3	4	5	Q Totals	Q Means	Sec Totals	Sec Means	Sec Mean Tots
	project status assess.	A0.10	3.00	5.00	5.00	1.00	1.00	15.00	3.00			
		A0.30	5.00	5.00	5.00	1.00	3.00	19.00	3.80			
										34.00	3.40	6.80
A	info quality	A1	4.00	4.00	4.00	4.00	4.00	20.00	4.00			
		A2	3.00	4.00	4.00	5.00	4.00	20.00	2.50			
		A3	5.00	3.00	3.00	5.00	4.00	20.00	4.00			
		A4	4.00	3.00	4.00	5.00	3.00	19.00	3.80			
		A5	4.00	4.00	4.00	4.00	3.00	19.00	3.80			
		A6	3.00	3.00	4.00	5.00	4.00	19.00	3.80			
		A7	3.00	4.00	4.00	5.00	4.00	20.00	4.00			
										137.00	3.70	25.90
B	USABILITY	B1	5.00	4.00	4.00	4.00	4.00	21.00	4.20			
		B2	3.00	2.00	4.00	4.00	4.00	17.00	3.40			
		B3	4.00	4.00	5.00	4.00	4.00	21.00	4.20			
		B4	4.00	4.00	3.00	3.00	5.00	19.00	3.80			
		B5	3.00	3.00	3.00	4.00	3.00	16.00	3.20			
		B6	2.00	4.00	4.00	5.00	4.00	19.00	3.80			
		B7	4.00	4.00	3.00	5.00	5.00	21.00	4.20			
		B8	4.00	2.00	3.00	4.00	3.00	16.00	3.20			
		B9	3.00	4.00	3.00	4.00	3.00	17.00	3.40			
		B10	4.00	4.00	4.00	4.00	4.00	20.00	4.00			
		B11	4.00	4.00	4.00	4.00	4.00	20.00	4.00			
		B12	5.00	4.00	4.00	5.00	4.00	22.00	4.40			
		B13	4.00	3.00	3.00	3.00	4.00	17.00	3.40			
		B14	4.00	4.00	4.00	4.00	4.00	20.00	4.00			
		B15	4.00	4.00	4.00	4.00	4.00	20.00	4.00			
		B16	4.00	4.00	4.00	4.00	4.00	20.00	4.00			
		B17	3.00	4.00	3.00	4.00	4.00	18.00	3.60			
		B18	3.00	4.00	4.00	4.00	4.00	19.00	3.80			
										343.00	3.81	68.60
		Score Totals	101.00	101.00	103.00	108.00	101.00	514.00		514.00		
		RSL Means	3.74	3.74	3.81	4.00	3.74					
		Mean Totals							101.30			101.30
		Mean Means				3.81		3.81	3.75	3.81		3.75

No. of RSL informants:	5
No. of questions:	27
No. of sections:	3
No. of scores:	135

Questions

Summary

			Client Protocol (all RSLs)		Initial Prototype (all RSLs)		Operational Prototype (all RSLs)	
			Question Means	Section Means	Question Means	Section Means	Question Means	Section Means
A	computer familia-arity	A0.20	3.60 V1					
		A0.40	4.40 V2					
		A0.60	2.60 V3					
				3.53 V39				
	project status assess.	A0.10			2.40 V4		3.00 V5	
		A0.30			3.20 V7		3.80 V8	
						2.80 V10		3.40 V11
	info quality	A1	3.60 V12		3.60 V13		4.00 V14	
		A2	2.00 V15		1.88 V16		2.50 V17	
		A3	3.60 V18		3.80 V19		4.00 V20	
		A4	3.80 V21		3.20 V22		3.80 V23	
		A5	4.20 V24		2.60 V25		3.80 V26	
		A6	3.60 V27		3.60 V28		3.80 V29	
		A7	3.60 V30		2.80 V31		4.00 V32	
				3.49 V33		3.07 V34		3.70 V35
B						3.28 V37		3.81 V38

Questions

Info Quality mean of both prototypes	3.38	V36

Mean of A0.10 s-curve assessment	2.70	V6
Mean of A0.30 software assessment	3.50	V9

Appendix 9: Results value code index

Explanation of how values in results table are obtained:

V1
The mean of all RSL Informant scores for question A0.20, Client Information Requirement Protocol.

V2
The mean of all RSL Informant scores for question A0.40, Client Information Requirement Protocol.

V3
The mean of all RSL Informant scores for question A0.60, Client Information Requirement Protocol.

V4
The mean of all RSL Informant scores for question A0.10, Initial Prototype Usability and Information Evaluation Protocol.

V5
The mean of all RSL Informant scores for question A0.10, Operational Prototype Usability and Information Evaluation Protocol.

V6
The mean of the means for all RSL Informant scores for question A0.10, Initial Prototype Usability and Information Evaluation Protocol (V4) AND the mean for all RSL Informant scores for question A0.10, Operational Prototype Usability and Information Evaluation Protocol (V5).

V7
The mean of all RSL Informant scores for question A0.30, Initial Prototype Usability and Information Evaluation Protocol.

V8
The mean of all RSL Informant scores for question A0.30, Operational Prototype Usability and Information Evaluation Protocol.

V9
The mean of the means for all RSL Informant scores for question A0.30, Initial Prototype Usability and Information Evaluation Protocol (V7) AND the mean for all RSL Informant scores for question A0.30, Operational Prototype Usability and Information Evaluation Protocol (V8).

V10
The sectional mean of the question score means for all RSL Informants for section A (Project Status Assessment): questions A0.010 and A0.30, Initial Prototype Usability and Information Evaluation Protocol.

V11
The sectional mean of the question score means for all RSL Informants for section A (Project Status Assessment): questions A0.010 and A0.30, Operational Prototype Usability and Information Evaluation Protocol.

V12
The mean of all RSL Informant scores for question A1, Client Information Requirement Protocol.

V13
The mean of all RSL Informant scores for question A1, Initial Prototype Usability and Information Evaluation Protocol.

V14
The mean of all RSL Informant scores for question A1, Operational Prototype Usability and Information Evaluation Protocol.

V15
The mean of all RSL Informant scores for question A2, Client Information Requirement Protocol.

V16
The mean of all RSL Informant scores for question A2, Initial Prototype Usability and Information Evaluation Protocol.

V17
The mean of all RSL Informant scores for question A2, Operational Prototype Usability and Information Evaluation Protocol.

V18
The mean of all RSL Informant scores for question A3, Client Information Requirement Protocol.

V19
The mean of all RSL Informant scores for question A3, Initial Prototype Usability and Information Evaluation Protocol.

V20
The mean of all RSL Informant scores for question A3, Operational Prototype Usability and Information Evaluation Protocol.

V21
The mean of all RSL Informant scores for question A4, Client Information Requirement Protocol.

V22
The mean of all RSL Informant scores for question A4, Initial Prototype Usability and Information Evaluation Protocol.

V23
The mean of all RSL Informant scores for question A4, Operational Prototype Usability and Information Evaluation Protocol.

V24
The mean of all RSL Informant scores for question A5, Client Information Requirement Protocol.

V25
The mean of all RSL Informant scores for question A5, Initial Prototype Usability and Information Evaluation Protocol.

V26
The mean of all RSL Informant scores for question A5, Operational Prototype Usability and Information Evaluation Protocol.

V27
The mean of all RSL Informant scores for question A6, Client Information Requirement Protocol.

V28
The mean of all RSL Informant scores for question A6, Initial Prototype Usability and Information Evaluation Protocol.

V29
The mean of all RSL Informant scores for question A6, Operational Prototype Usability and Information Evaluation Protocol.

V30
The mean of all RSL Informant scores for question A7, Client Information Requirement Protocol.

V31
The mean of all RSL Informant scores for question A7, Initial Prototype Usability and Information Evaluation Protocol.

V32
The mean of all RSL Informant scores for question A7, Operational Prototype Usability and Information Evaluation Protocol.

V33
The sectional mean of the question score means for all RSL Informants for section A (Information Quality): questions A1-A7, Client Information Requirement Protocol.

V34
The sectional mean of the question score means for all RSL Informants for section A (Information Quality): questions A1-A7, Initial Prototype Usability and Information Evaluation Protocol.

V35
The sectional mean of the question score means for all RSL Informants for section A (Information Quality): questions A1-A7, Operational Prototype Usability and Information Evaluation Protocol.

V36
Mean of means
The mean of the means for all RSL Informant scores for questions A1-A7, Initial Prototype Usability and Information Evaluation Protocol (V34) AND the mean for all RSL Informant scores for questions A1-A7, Operational Prototype Usability and Information Evaluation Protocol (V35).

V37
The sectional mean of the question score means for all RSL Informants for section B (Software Usability): questions B1-B18, Initial Prototype Usability and Information Evaluation Protocol.

V38
The sectional mean of the question score means for all RSL Informants for section B (Software Usability): questions B1-B18, Operational Prototype Usability and Information Evaluation Protocol.

V39
The sectional mean of the question score means for all RSL Informants for section A (Information Quality): questions A0.20, A0.40 and A0.60, Client Information Requirement Protocol.

Appendix 10: Initial results and observations after visit 1

The question scores referenced in this section can be found in the tabulated results (Appendix 8).

Conclusions from assessments of Client Information Requirement Protocol scores

- Informants' current reporting methods generally have a high level of technical complexity (question A0.40, mean 4.4).

- The content of Informants' current reports tends to be descriptive and/or numerical, rather than graphical (question A0.60, mean 2.6).

- Informants perceive the QUALITY of information in their current reports to be low (question A2, mean 2.0).

- Informants perceive the ACCURACY (A3, mean 3.6), RELEVANCE (A4, mean 3.8), LEVEL OF DETAIL (A6, mean 3.6), FORMAT (A7, mean 3.6) and delivery TIME (question A1, mean 3.6) of their current report to be satisfactory.

- Informants perceive the INTELLIGIBILITY of information provided by their current reports to be high (question A5, mean 4.2).

Conclusions from assessments of Client Information Requirement Protocol comments

The following are edited responses from Client Information Requirement Protocol, questions A8 and A9:

- Would like report dates standardised- information not always available when required.

- Figure based information from employer's agents is not readily transferable to internal committee reports

- Not currently enough information on associated works (i.e. project elements not directly under an Informant's control but with a significant impact on outcome)

- Time, cost and quality implications are not always easily extracted from an over-detailed report, would prefer something more simple and indicative (described a dashboard display with gauges... if all needles on zero... all is well with project).

Conclusions from assessments of Initial Prototype Usability and Information Evaluation Protocol scores

Informational:

- Informants perceive the QUALITY of information provided by the Initial Prototype to be low (question A2, mean 1.88).

- Informants perceive the ACCURACY of the information provided by the Initial Prototype to be high (question A3, mean 3.8).

Usability:

- The lowest mean score given was for the use of familiar language (question B4, mean 2.6).

- All mean scores were over 2.5.

- The highest mean score was for the installation process (question B10, mean 3.8).

Conclusions from assessments of Initial Prototype Usability and Information Evaluation Protocol comments

General and Informational:

- Informants would like more time to try out prototype. Ideally, several weeks. Several of them felt that they could not accurately answer the Prototype Usability and Information Evaluation questions, without a longer trial. Therefore, their evaluation scores for some questions were based on how they 'imagined' the tool might change their working experience. Certain Informants volunteered the opinion that they were scoring the prototype fairly low in some areas, because of lack of long-term evaluation.

- Software provides more useful information than current reporting method but presentation needs improvement.

- RSL Informants are more interested in predictions of project status 'at completion' than performance 'to date'. Performance to date may be of more value to the Employer's Agent (?)

- Inclusion of time dimension

Usability Issues-

- TeeChart problems in Win95 (process not shutting down correctly)

- Colour scheme- white background and black text.

- Take out task names on data surface axis but improve system for displaying names in the application border (status bar?).

- Task names only show if large deviations present in task

- Uninstaller crashed one time in five (it is possible that this related to the problems with Windows 95, rather than the installer).

- Font sizes of axes labels and scales need attention.

- Print screen function

- Reduce "screen clutter"

Overall observations on the application of methodology

- At this stage, the mean satisfaction with information provided by current reporting was found to be higher than that provided by the Initial Prototype (see Appendix 8). However, Informants seem to assess project status more accurately with the first prototype than with the s-curve. At first glance, this would seem to be contradictory. One initial thought is that comparing the quality of information provided by an established reporting method and a prototype, may be misleading. The initial prototype does not benefit from requirements capture.

- The majority of Informants were not familiar with Earned Value Analysis or other performance measurements. This must call into question the validity of comparing Informants' analysis of the EVA s-curve, with their analysis of information provided by the prototype.

- There were fewer problems introducing the concept of 3D information visualisation than expected.

- Informants seemed generally inexperienced when dealing with cost and schedule as separate parameters. Their current reporting methodologies often focused on schedule as the key factor. Where cost was used, it was purely as an indicator of schedule i.e. an employer's agent would plot (or provide figures for) actual cost against budget to reveal progress. Therefore, it is probable that the widespread use of fixed-price contracts influences the reporting methodologies utilised.

Software changes made for Operational Prototype

Further development required (issues not identified during interviews)-

- Internet functionality (transfer code from VTK prototype)
- Jewel case inlay design
- CD label design
- Write help file
- Test prototype across 95/98/NT/2000
- Project98 templates and macros (including possible macro to convert CSV to dBase V via Excel)
- Refine installer...make a note of registry entry command to register TeeChart...may need to add command to amend ODBC set-up (?)
- May also need OpenGL installed...check this...
- Check polarity of all deviation parameters is the same i.e. positive values are all bad news...
- Logarithmic scale issue...bringing things into range...

Identified during interviews (initial prototype):

- Text seems to work better on application frame (status bar), rather than in 3D space. Informants want more textual information than currently provided by prototype but additionally want the visual simplicity of the 3D data surface. Reduce text on data surface as much as possible BUT replace this with a very efficient system of converting 'mouse-over surface' messages to frame text messages.

- Remove deviation parameters concerning 'to date' values, only keep 'at completion' information.

- Make sure that language used is familiar to Informants.

- Generally reduce 'screen clutter'.

- Quality of information is perceived by Informants to be low in both current methods and Initial Prototype.

- Colour scheme- white background?

- Add print screen option?

Appendix 11: Article 'Enter the Virtual Building Site' in *Housing* magazine

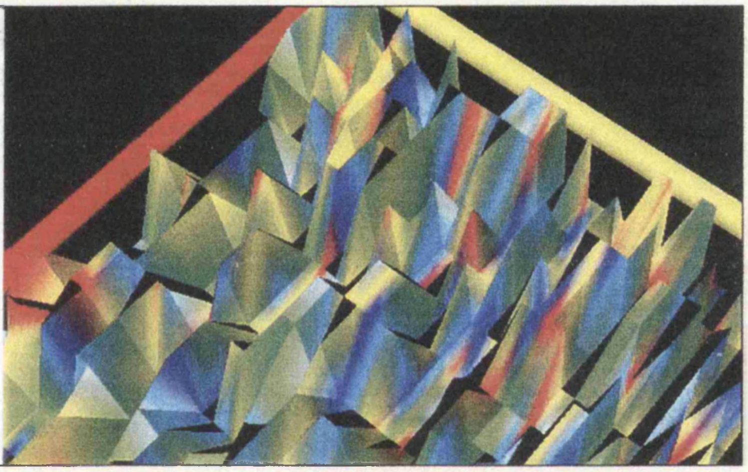
housing

April 1999

the magazine of the Chartered Institute of Housing

Enter the virtual building site

Registered social landlords with projects about to go on site are needed to pilot intelligent virtual reality software that can instantly deliver an update of progress on a development and pinpoint problems before they become major difficulties. Steve North, doctoral researcher and lecturer in virtual environments at University College London, explains



Imagine the scene ...the year is 2010 and your latest development project has just gone 'on-site'. You're a busy senior development officer for a large RSL and you're late for a 9am meeting with housing management colleagues. They're going to want to know the latest estimated 'hand-over' date and if everything else in the scheme is on track.

You need an accurate, up-to-date summary of your project and you need it now. Simple: you speak kindly to your computer (speech recognition is by now standard) and it instantly delivers a three-dimensional representation of your project. Not a model of the built product but an abstraction of the information that makes up the project process, retrieved from a computer located elsewhere on the internet.

Such a scenario may sound unlikely, given the state of technology in use by RSLs. However, this is exactly the subject of my current research project, the development of a software tool called procession.

This is one of many innovative projects under development at the Virtual Reality Centre for the Built Environment, University College London (UCL). Winner of an Office of Science & Technology Foresight Challenge Award, the centre is an interdisciplinary initiative between UCL, Imperial College and a consortium of 16 construction industry partners.

Still in the early stages of feasibility, it is hoped that procession will provide an 'intelligent' visualisation software tool for RSL construction clients. By applying artificial intelligence and the latest developments in Virtual Reality (VR) technology, procession hints at a new generation of 3D graphing and presentation tools.

In practical terms, this may mean an evolutionary move from the traditional 2D S-curve to the use of 3D data surfaces (above). In its simplest form, procession's data surface will be able to show the user at a glance the current 'health' of a construction project. A flat terrain will indicate that the project is proceeding according to its intended program. Spikes and troughs warn the viewer that deviations have occurred in terms of time, cost and quality. Colour can then be used to indicate the seriousness of a deviation from programme.

By 'learning' from project to project, procession, will be able to use colour to warn about a deviation (however small) that has had severe outcome implications for previous projects. The software should also be able to make decisions about the relevance of data and what should be filtered out, in order to provide only a summary.

Underlying this research, is the ethos that procession should be simple, affordable and usable by an RSL on the average PC found in the office. The information used will be obtained over the internet from either a standard web server or from a project database. It is likely to utilise the format of a common project management package, such as Microsoft Project 98.

While the initial focus of this research is on construction projects, there may be future applications for such techniques on other projects types and in other areas of a field known as 'information visualisation'.

As the volume of available information continues to increase, 3D information visualisation provides one route to aiding intelligibility. It is becoming a popular approach to exploring and understanding complex data sets, such as the financial markets and projects. It is even arguable that the real virtual worlds of tomorrow may be functional representations of society's information, rather than the purely social spaces envisioned by early VR commentators.

If all of this seems far removed from your daily experience of using computers in social housing, just remember how irrelevant e-mail and the web seemed four years ago.

I would very much like to hear from RSLs interested in helping to develop and evaluate procession. The time commitment is expected to be minimal.

Please contact me by e-mail:
newpeople@newhomes.demon.co.uk
or write to: Steve North, The Virtual Reality Centre for the Built Environment, The Bartlett School of Graduate Studies, University College London, 1-19 Torrington Place, London WC1E 6BT

Further information on The VR Centre For The Built Environment can be found at www.vr.ucl.ac.uk.

Appendix 12: Article 'Reclaiming the backlands' in *Building Homes* magazine

Reclaiming the backlands

Existing low density urban estates may yield small but significant sites for development. Alleyways, gardens, even street corners can offer development potential, reports Josephine Smit.

Front to back

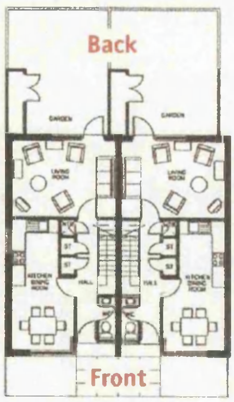
Turning existing houses back to front is by no means a new idea. The process basically involves retaining the internal room layout of a house, but re-elevating its exterior to give the back of the house a suitably impressive front.

By doing this HTA Architects is helping to rectify the many faults of the Radburn layout of the Moorlands Estate in Brixton, south London. Around half the estate's 517 homes are being reversed by the addition of new brick and glass entrance porches with zinc pitched roofs (right), and in some cases bay windows, under an ongoing ERCF project with London Borough of Lambeth and Metropolitan Housing Trust.

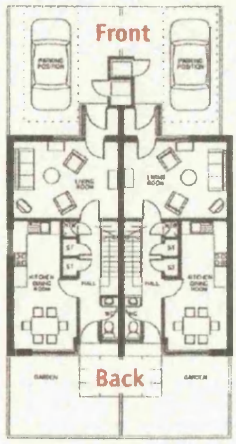
The switch is part of a strategy to give the estate a new and clearly defined street pattern, with boundaries redrawn and threatening pedestrian routes surrounding the homes removed. As a result, tenants not only get re-styled homes, but bigger and more secure gardens (below right) within a better quality environment.

HTA applied the solution successfully to a Radburn layout estate in Birmingham nearly a decade ago and before that was using it to transform older housing. "At Moorlands we are doing it while people are living in the houses, which cuts out the cost of decanting tenants," says Mike De'Ath, project director with the architect. Contractor on the project is Mansell. As ERCF only funds improvement works, no new build has yet been incorporated at Moorlands, but separate funding is being looked at for provision of new homes.

Before



After



Appendix 13: Design layout for Procession's CD label and jewel case

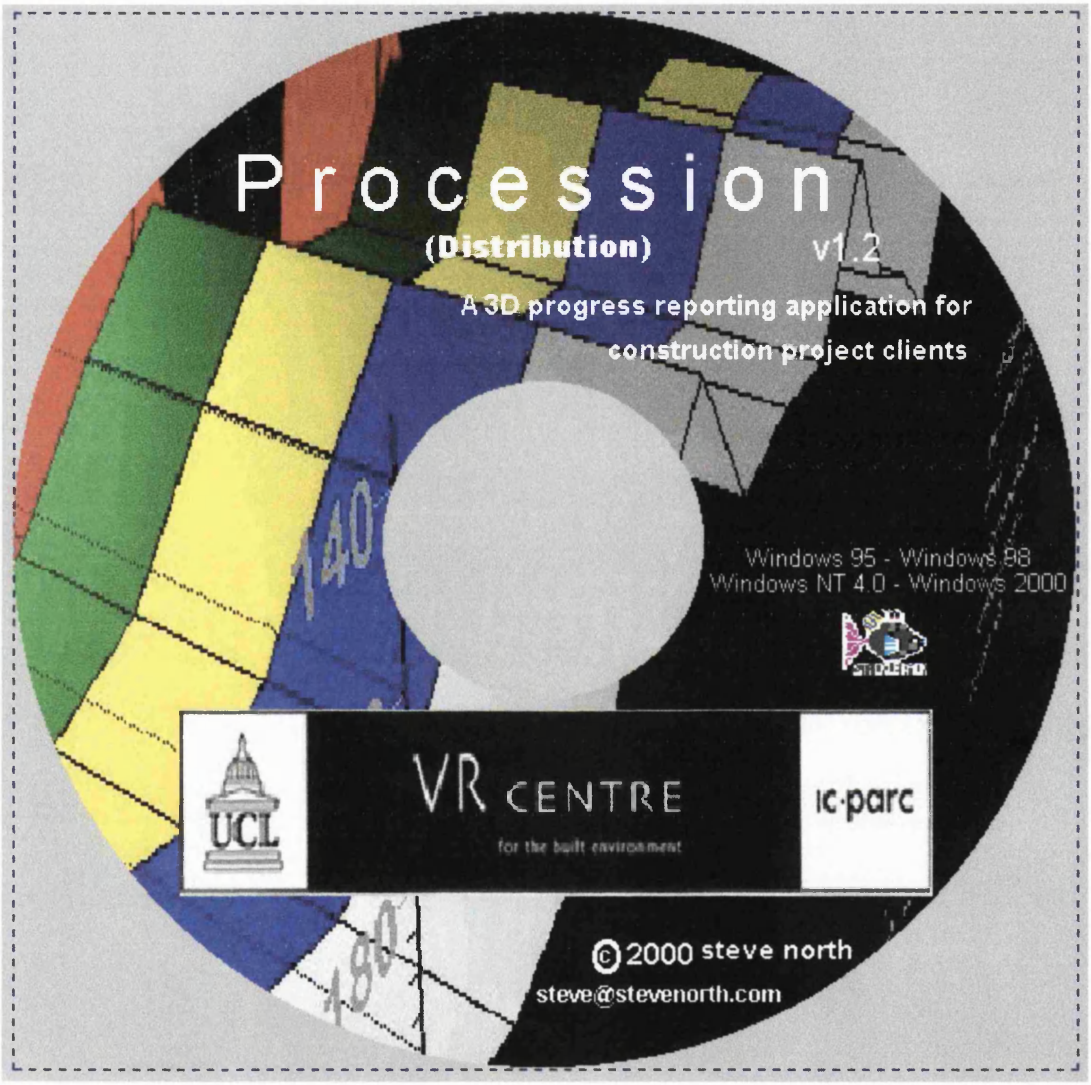


Figure 0.1 Procession CD label design, source: author.

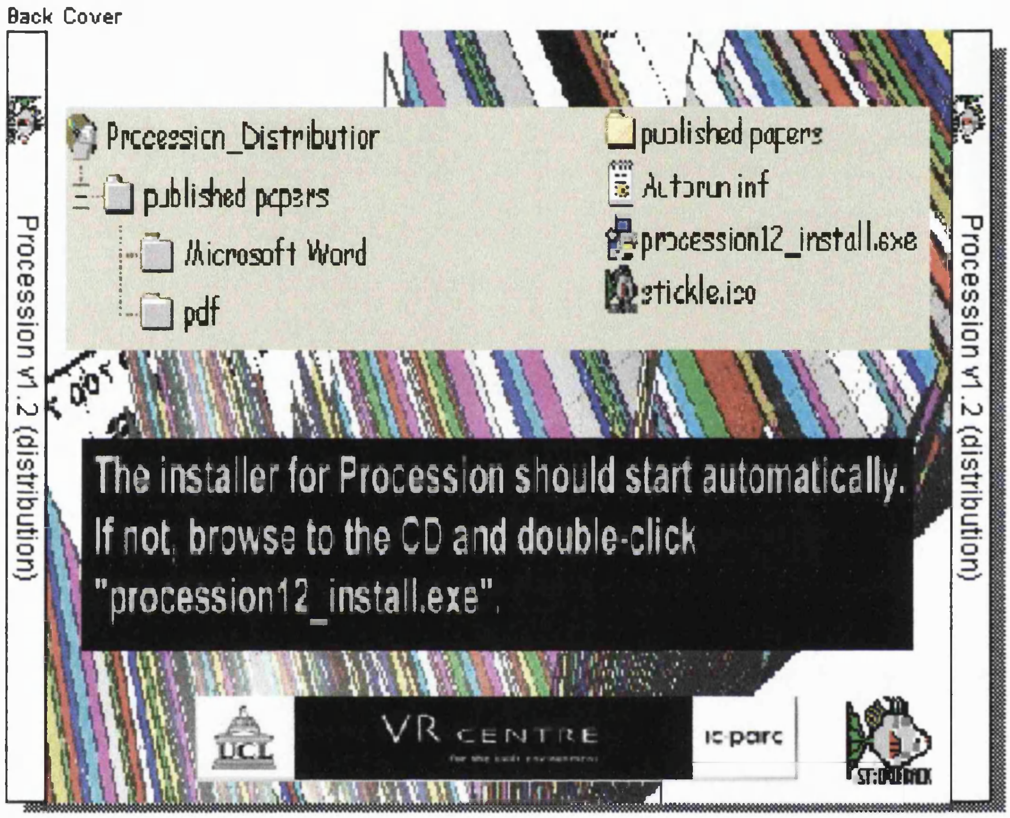


Figure 0.2 Procession CD jewel box back cover design, source: author.

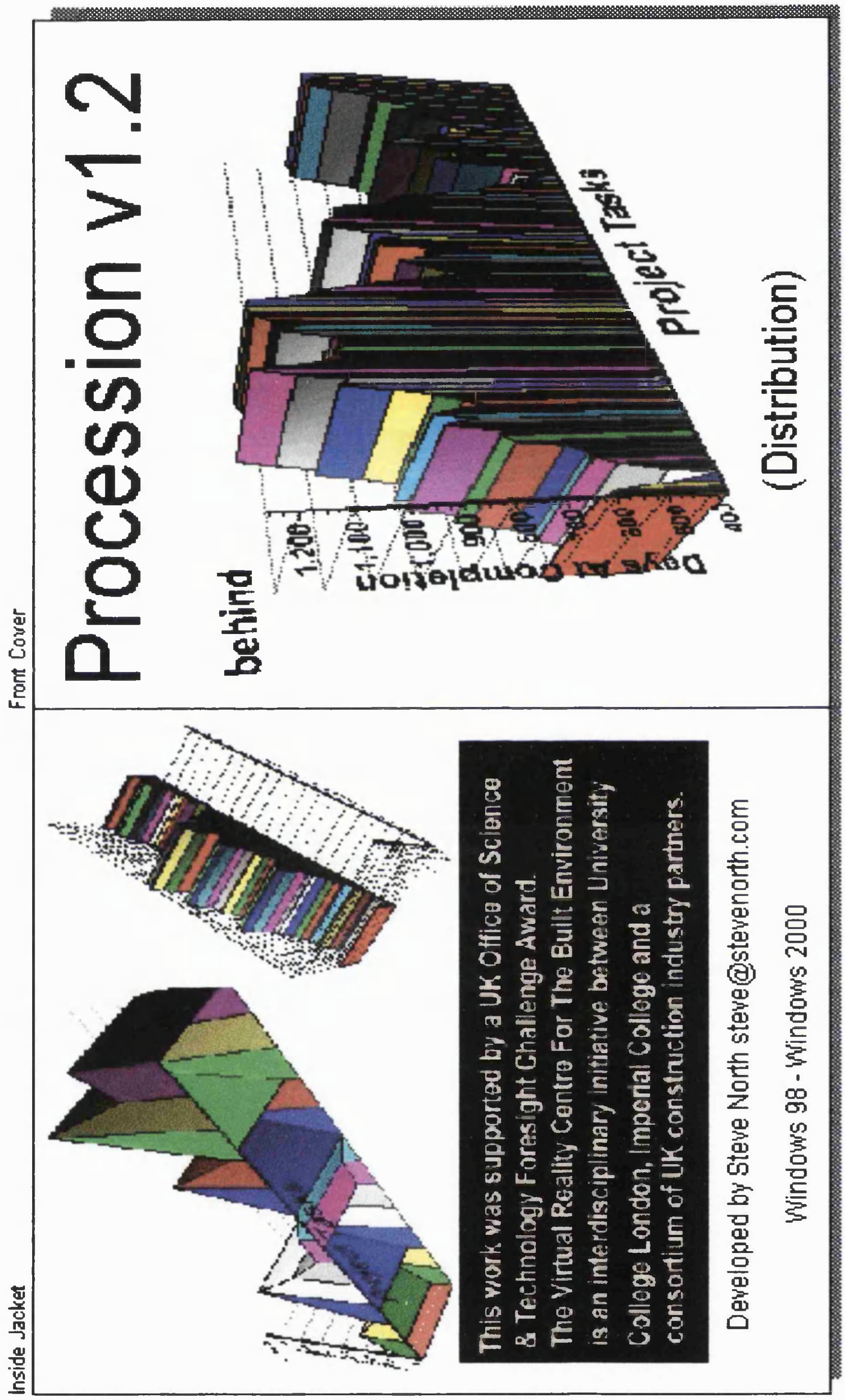


Figure 0.3 Procession CD jewel box front and inside cover design, source: author.

Appendix 14: Draft help file for Procession v1.2

Integration with Microsoft Project

In an ideal world, the exchange of data between software tools should be seamless. In practice, it often does not work like that. Although Project is well equipped with export formats and data base connectivity (ODBC), problems were identified during the integration stage. The original intention was that Microsoft Access (.mdb) files or CSV files would be used instead of dBase. Unfortunately, TeeChart cannot find the data-type "LONG-TEXT", which is used by Microsoft Project for its .mdb task name field. Microsoft Project 98 does not allow editing of the export map data-types, so it cannot be changed to "TEXT". TXT/CSV ODBC format works fine when accessed from TeeChart's manual editor. When accessed from code, TeeChart cannot find the source (throws: ODBC error 100 SQLSetConnectAttr failed).

As a result, the process of data exchange is currently only semi-automatic and less than elegant. It is hoped that this will be resolved in any future releases.

Automatically exporting files from Microsoft Project to Procession:

Procession requires its data in the dBase (.dbf) format. dBase is a standard that has been adopted by many other database applications, including Fox Pro. See www.dbase2000.com for more information.

As Project does not directly export to this format, an additional process is required.

1. Open the required file in Project.
2. Click the "Export Procession" toolbar button.
3. Microsoft Excel will open showing the exported values.
4. Procession will launch (just so that it is handy for later)

5. Find the column headed "DV".

6. Select all cells in the "DV" column, including the "DV".

7. Right-click and select the item "Format Cells"

8. Select "category" "number" (if not already selected) and "OK"

9. For each cell in the "DV" column, edit the value to get rid of the word "days". Not doing this will result in Procession interpreting the values as strings and showing duration variances of zero.

10. After the last value in the column "DV", enter two blank lines with "DV" and "CV" values of zero. These are required to overcome a bug, whereby the last two tasks are ignored.

11.From the "file" menu, select "save as...".

12. Set the "Save as type" to DBF 4 (dBase IV).

13. Browse to c:\procession\data\

14. Choose a filename...for example "myProject.dbf" and click "save".

15. Close Excel.

16. In Procession, browse to c:\procession\data\ and open your new file.

Manually exporting files from Microsoft Project to Procession

1. Open the required file in Project.

2. From the "file" menu, select "save as...".

3. Set the "Save as type" to CSV comma delimited.

4. Browse to a suitable location for the temp file.

5. Choose a filename...for example "myProject.csv" and click "save".

6. In the "Export Format" window, click "new map".

7. Type something suitable i.e. "procession"." in the "Import/Export map name" box.

8. Under "Options" tab, select "Tasks".

9. Under the "Task Mapping" tab, set "Export filter" to "summary tasks".

10. Set "From: Microsoft Project field" to "name" and change the "To: text file field" to "Task_Name "

11. Set "From: Microsoft Project field" to "CV" and leave "To: text file field" as "CV"

12. Set "From: Microsoft Project field" to "Duration Variance" and change the "To: text file field" to "DV".

13. Click "ok".

14. In the "Export Format" window, click "save".

15. Open Microsoft Excel.

16. Open CSV file.

17. Drag border of column A until all task names are revealed, or DBF file may have truncated names.

18. Repeat 17. for any values too big for columns.

19. Follow steps 5 to 16 in last section.

Note: on subsequent manual exports, steps 7 to 14 can be excluded because, from the "Export Format" window, you should now be able to select the export map that you created the first time.

Manually exporting files from other applications to Procession

The main requirements are that the file includes the three fields "Task_Name", "CV" and "DV" and that format is *.dbf.

Microsoft Project Macros

Project makes use of a programming language called Visual Basic for Applications (VBA) to record and automate tasks. These small programmes are known as macros. Once you can see the Procession toolbar buttons in Project, you also have access to specially written macros for formatting and exporting data to Procession. Appendix 3 shows the VBA code for the macro included with Procession v1.2.

ODBC, ActiveX and OpenGL set-up issues

Procession gets its data by opening and querying a database through the Open Database Connectivity (ODBC) interface. To work correctly Procession requires the ODBC 3.0 core components to be installed on your machine.

If you cannot get Procession to open files...it is because ODBC is not properly configured. Microsoft's ODBC Desktop Database Drivers include 32-bit drivers for Microsoft Access, dBASE, Microsoft Excel, Microsoft FoxPro, Paradox, and Text. If you've got Microsoft Office, Visual Basic Professional, Visual C++ or a similar product installed, the chances are that you've got ODBC 3.0. The main information web site for ODBC is: www.microsoft.com/odbc.

Procession uses the dBase ODBC driver (for the file format .dbf). If dBase driver not present, use Fox Pro. Procession requires a data source to be defined for the directory c:\procession\data\

To add a data source:

1 Open Control Panel. In the Control Panel window, double-click the ODBC icon.

2 Choose the User DSN tab.

3 Choose the Add button.

4 In the Create New Data Source dialog box, select Microsoft dBase Driver (*.dbf) and follow the instructions of the wizard.

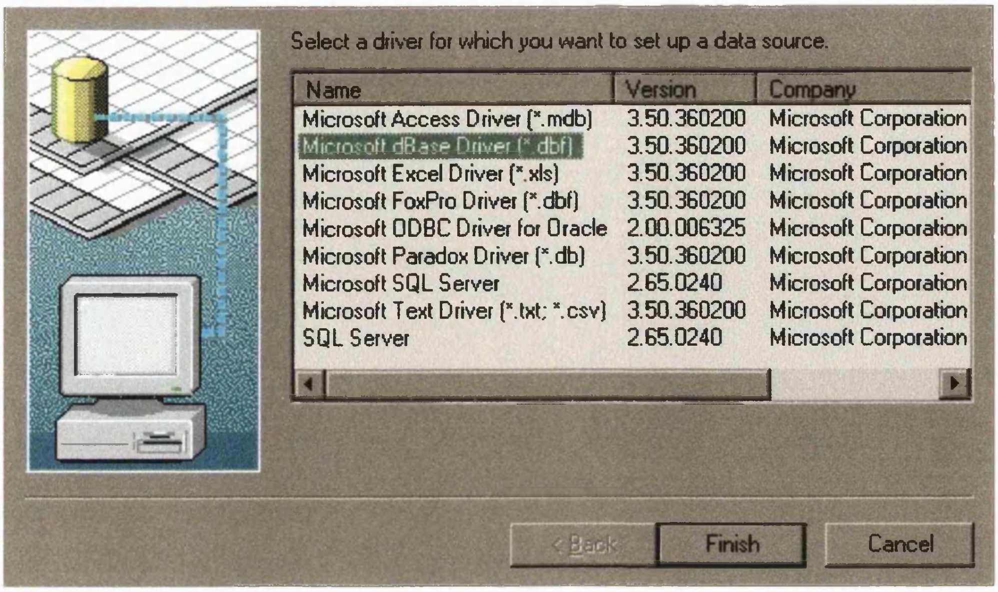


Figure 0.4 ODBC data source administrator in Microsoft Windows, source: author.

5. If the dBase driver is not shown in the list, the FoxPro driver can be used as an alternative.
6. If neither driver is shown, you will need to install ODBC 3.0: In the directory c:\procession\, double click the file "ODBC3_installer.exe" and follow the instructions. It may require you to restart your computer.
7. Back in the Create New Data Source dialog box, select the dBAse or FoxPRo driver.
8. In the ODBC dBase Setup dialog box., set the "Data Source Name" box to "dBase" ...Version dBase 5.0..."Use current directory" un-ticked and directory set to "c:\procession\data". The data source name should be set to "procession". Note: use a small p, this is case sensitive.
9. You should now have a User Data Source called "procession".
10. Click OK
11.Procession should now be able to open files.

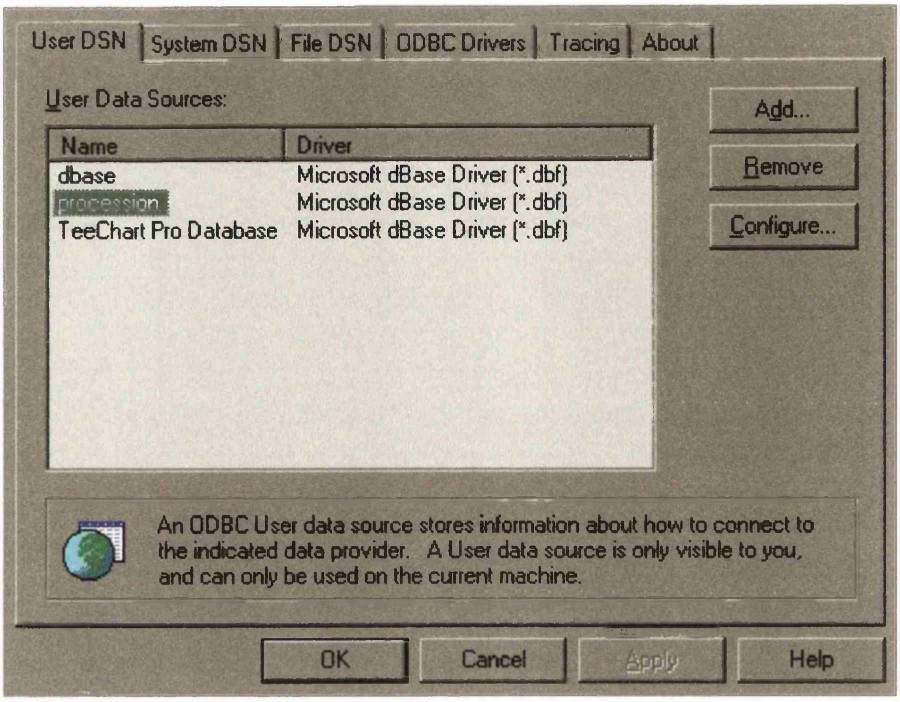


Figure 0.5 ODBC data source administrator in Microsoft Windows: User DSN tab, source: author

TeeChart Active X control

Procession uses an Active X control for its graphics functionality. Its installation should be handled automatically. In the very unlikely event that the installer fails to register the control correctly, proceed as follows:

1. Check that the file "teechart.ocx" is in your directory c:\windows\system\ or c:\winnt\system32\.
2. To make sure the TeeChart ActiveX control is accessible, you can execute this command to register TeeChart Pro Activex under Windows : regsvr32.exe c:\windows\system\teechart.ocx (regsvr32.exe is a Microsoft application in \windows\system)

Procession uses the OpenGL library for its 3D functionality. For more info see: www.opengl.org/. If Procession will not start or its 3D functionality is impaired, it could be an OpenGL problem. Windows NT v4.0, Windows 2000, Windows 98, and Windows 95 (OSR2) all ship with OpenGL v1.1, so if you are using one of these operating systems, you're all set. However, earlier versions of Windows 95 may not have OpenGL v1.1 installed. If you have an earlier version of Windows 95 and you do not have OpenGL installed, then you should re-name the file c:\procession\opengl32.dl_ to opengl32.dll. Placing opengl32.dll in either c:\procession\ or c:\windows\system\ should be the correct course of action. Alternatively, Microsoft's OpenGL for Windows 95 can be downloaded from the Microsoft site at: http://download.microsoft.com/download/win95upg/info/1/W95/EN-US/Opengl95.exe or ftp://ftp.microsoft.com/softlib/mslfiles/opengl95.exe. This provides a full installation routine and READ ME instructions.

Using Procession

Procession comes with example data files. These files are also available on the Internet, to demonstrate how Procession might be used during a physically distributed project. When Procession first runs, it displays a flat surface. All duration variance values are set to "zero days".
To open a file:
Select "open" from the "file" menu.

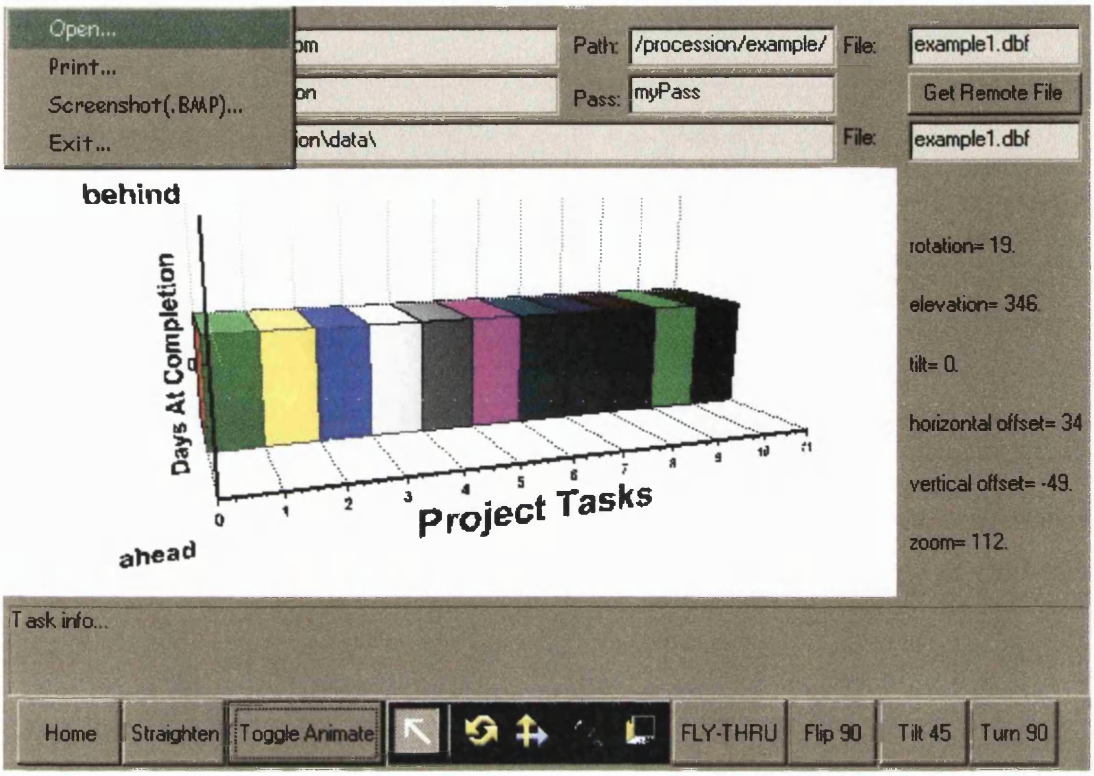


Figure 0.6 Opening a local file with Procession (stage 1), source: author.

Choose an example data file from the directory c:\procession\data\ and click "open".

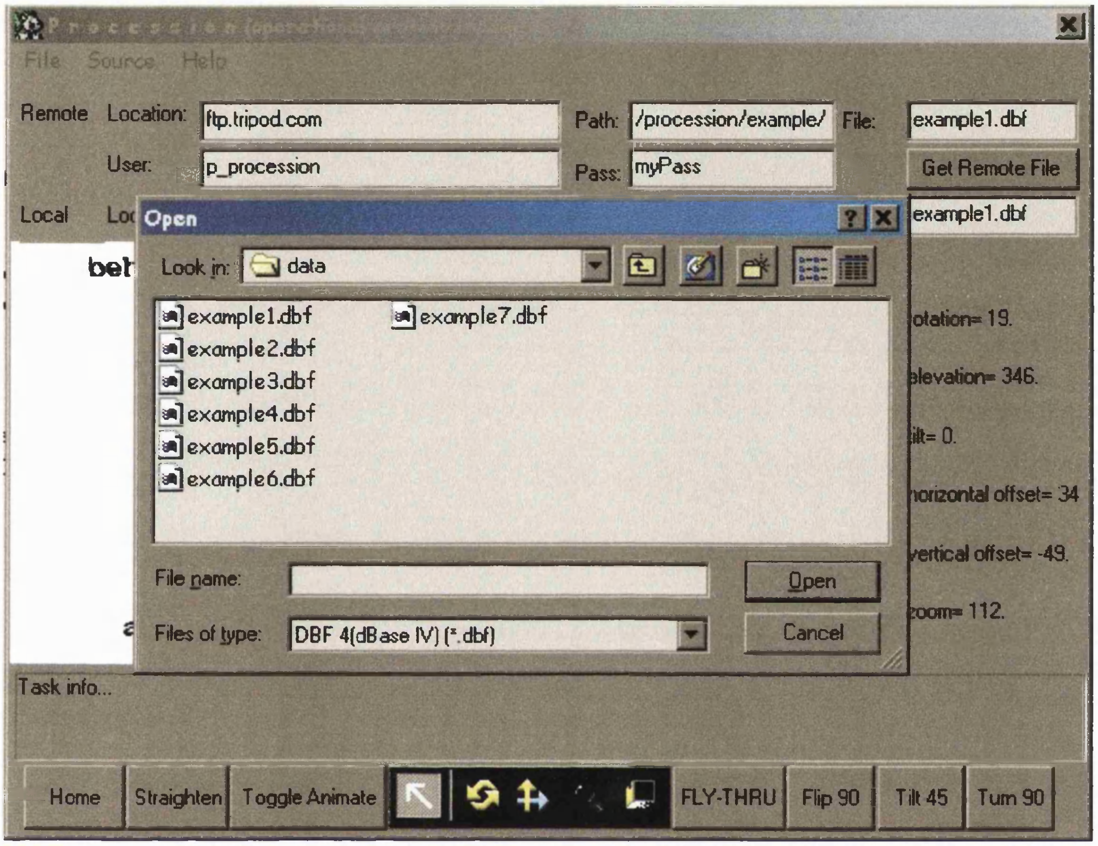


Figure 0.7 Opening a local file with Procession (stage 2), source: author.

If all is well, the flat 3D surface will be replaced with one with varying heights and the text box "Local", "File" will contain the name of the file opened. If the file does not open and an error message comes up, please see the ODBC section.

Left clicking on different parts of the surface will reveal the selected task's status and value. The information comes up in the box directly above the navigation bar (default message "Task info..."). Important: this will only work when the navigation button with the arrow pointing to top left is selected. Click the menu item "Source" for a pop-up page, giving the values of each task in the surface.

To try out Procession's Internet abilities, make sure that your computer is connected to the Internet (directly or through a dial-up connection) and then click on the "Get Remote File" button. Procession will FTP the file "example1.dbf" from the server ftp.tripod.com and open it. If this is successful, try changing the name in the text box "Local" , "File" to "example2.dbf" etc.. If you have your own web server space (which must be accessible with FTP), you can up-load files and then access them with Procession. Procession's FTP settings are all in the text boxes at the top of the application. The top line boxes are the server name, directory location and file name of the file that you want to visualise. The second line boxes are the username and password for the target server (they may not be required...anonymous servers require an email address for a username and the pass left blank).

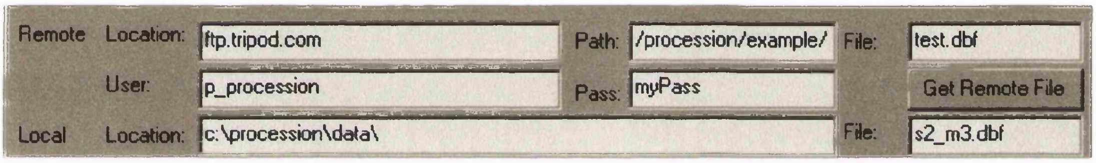


Figure 0.8 Procession's FTP control panel, source: author.

Navigation and interface

Procession provides a panel of navigation controls for exploring and examining the 3D surface. Starting at the left of the five central buttons on the black background. The default navigation mode is the selection arrow. This does not allow you to move in the 3D space and is mainly for clicking on the surface to reveal values. Next left, is the rotate arrow. This allows full orientation within the 3D space. Having selected this button, left click and hold the left mouse button anywhere in the 3D window. With the left button still down, drag left, right, up or down and observe the effect on the data surface. This is the principle method for examining the 3D surface. Still in this mode, right clicking and dragging will allow you to move the surface left, right, up and down.

Next left, is the move arrow. This performs the same function on left mouse drag as described for right mouse, under the last mode i.e. a sliding of the surface relative to the user.

The small magnifying glass zooms in or out of the 3D space. In the context of Virtual Reality applications, this can be thought of as "walking" into the scene. Having selected this button, left click and hold the left mouse button anywhere in the 3D window. With the left button still down, moving up the screen zooms in, down zooms out.

The final black background button alters the 3D "depth" of the scene. Having selected this button, left click and hold the left mouse button anywhere in the 3D window. With the left button still down, moving up and down the screen squeezes and expands the 3D effect.

Figure 0.9 Procession's 3D navigation panel, source: author.

Other buttons work as follows:

"Home" Return 3D viewpoint to a default position and orientation.

"Straighten" Return 3D viewpoint to a default position..straight on to user.

"Toggle Animate" This turns on or off a simulation of changing surface values which may in future releases be used to represent "time". Current values are random.

"Fly-Thru" Turns on or off a flight through the data surface for presentation purposes.

"Flip 90" Rotate surface 90 degrees around the X axis.

"Tilt 45" Rotate surface 45 degrees around the Z axis.

"Turn 90" Rotate surface 90 degrees around the Y axis.

Right hand display panel shows current 3D position and orientation.

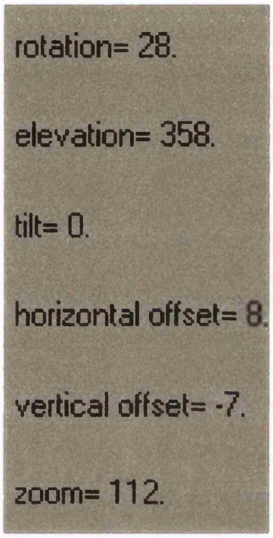


Figure 0.10 Procession's 3D position panel, source: author.

Exporting suitable data from other applications

The main requirements are that the file includes the three fields "Task_Name", "CV" and "DV" and that format is *.dbf.

Creating data files by hand

In order to try out Procession without a real project file, proceed as follows:

	A	B	C
1	TASK_NAME	CV	DV
2	some task	0.00	3.00
3	another task	0.00	15.65
4	thingy task	0.00	-345.00
5	this old task	0.00	1.00
6	blank	0.00	0.00
7	blank	0.00	0.00

Figure 0.11 Example data in Microsoft Excel, source: author.

1. Open a new Microsoft Excel worksheet.
2. Create three column headings:

Call one of the cells in row 1 (i.e. 1A) "TASK_NAME"

Call one of the cells in row 1 (i.e. 1B) "DV"

Call one of the cells in row 1 (i.e. 1C) "CV"

The order of these three headings in row 1 is not important, but there should not be any empty columns between them. It is ok to have other named columns in the same file, even columns in between the three required headings. Additional named columns do not require contents, just an entry in row 1.

Row cells below the heading "TASK_NAME" contain the task names i.e. in cell 2A type "task 1", 3A "task2" etc..

Row cells below the heading "DV" contain the duration variance values (in hours) for the corresponding task name i.e. in cell 2B type "34", 3B "-24" etc.. A positive value indicates hours behind schedule, a negative value is ahead of schedule.

The column headed "CV" is a dummy row required for technical reasons and it should be filled with 0.0 values, matching the number of values in the other two columns.

Important: Procession ignores the last two rows in each file. To make sure that the last two of your tasks are visible, enter two dummy tasks with DV and CV values of 0.0.

3. Format all active cells as follows:

Select all relevant cells

Go to Format menu->Cells option->Number tab

Set Category to "number" and decimal places to "2"

4. Save example file in dBase format as follows:

Go to Edit menu->Save As option

From the "type" dropdown. Select "DBF 4 (dBase IV)" and save your file with the extension *.dbf i.e. example.dbf

It is a good idea to save your file in Procession's default ODBC user DSN directory: C:\procession\data\

Procession should now be able to open this file.

Appendix 15: A contextual discussion of Artificial Intelligence

Many software tools applied to the construction industry make claims of 'intelligence'. As such it is useful to review exactly what computer science understands by this term. It could be argued that Artificial Intelligence (AI) can be traced back to Alan Turing's landmark paper, 'Computing machinery and intelligence' (1950). In this he proposes a 'test' to determine a machine's ability to convince an interrogator of its humanity. AI is defined by one of its figureheads, Marvin Minksky, as "the field of research concerned with making machines do things that people consider to require intelligence" (1986, p.326). From an alternative viewpoint, Watson states that, "AI may be defined as the branch of computer science that is concerned with the automation of intelligent behaviour" (1997, p.85). At first glance, there may seem little difference between these descriptions. However, closer examination reveals a paradigmatic split that reflects divisions within the field itself. Watson's view seems to assume that 'intelligence' has one universal model, as exemplified by humans. Minsky's more cautious response, may imply a criticism of any assumption that machine intelligence needs to be comparable with the human variety.

Watson's position appears to be that focusing on the nature of intelligence is a waste of time. Rather than attempting to create genuine machine intelligence, Watson appears to suggest that AI is about 'automating' processes that humans consider to require intelligence. This is a very practical viewpoint that positions AI as the provider of 'smartness', 'intelligence' and 'automation' for software products. However, it is important to remember that this is not the only paradigm operating within the AI community. Minsky hints at an entirely different research strand that regards itself as heading towards the 'holy grail' of thinking, feeling machines. It is not necessary that the intelligence demonstrated by such machines would even be comparable with the human variety.

This divergence of opinion is one of two main splits between AI researchers. The Minsky/Watson division can be categorised as 'Hard' versus 'Soft' AI. This is essentially a paradigmatic division, which might be expressed as: "What is the goal of AI?" Another split is between 'Deliberative' and 'Reactive'. This is more a methodological distinction, which can be summarised as: "How do we reach the goal?".

The hard/soft AI division will now be summarised. The Hard AI paradigm suggests that one (or both) of the branches of AI (Deliberative or Reactive) will ultimately produce universally acknowledged machine intelligence. By way of contrast, Soft AI argues that genuine machine intelligence will never be achieved. That given, Soft AI claims that Deliberative and/or Reactive AI can still provide 'smart' tools that are extremely useful (possibly more useful than an autonomous intelligence that might have its own ideas about what it wants to do).

The Deliberative/Reactive split will now be further considered. From the 1950s until the early 1970s, the Deliberative AI paradigm was the only recognised approach. In keeping with scientific thought of the time, the approach was very 'top-down'. It was thought that machine intelligence could be achieved by providing the computer with an increasingly detailed picture of the world, largely through language. Linguistics was seen as a mathematical process that could be deconstructed and programmed into the computer. This deconstruction is known as Natural Language Parsing. However, it rapidly became apparent that the understanding of language was about more than grammatical structure. A human's ability to interpret language is now widely regarded as being dependent on contextual experience. Words can have a variety of meanings and the selection of a correct definition relies on our understanding of many experiential signifiers.

While it seems increasingly unlikely that Deliberative AI will satisfy the goals of Hard AI enthusiasts, there are some researchers who believe that this is still possible. Their argument is principally that: just because computer science has so far failed to find a top-down system for encoding language and meaning, does not mean there is not one to be found. From a more objective perspective, it is undeniable that the majority of software tools that claim to use AI, are based on Deliberative techniques. For example: machine learning through Case-Based Reasoning (CBR). This is a method whereby the computer system creates heuristics ('rules of thumb') from previous experiences. Many Expert Systems (for example automated customer support) work on this principle.

The 1970s saw the emergence of the Reactive paradigm as a new 'bottom-up' way of thinking about AI. This belief can be seen as part of a holistic change of perspective linked to other emergent ideas, such as Chaos Theory and Fractal Mathematics. Reactive AI researchers "believe that intelligent behaviour can be generated without the sort of explicit-level representations (and hence, reasoning) prevalent in AI" (Nwana 1996, section 5).

There are a variety of techniques that are used to actualise Reactive AI. Neural networks are often described as being analogous to a human brain. When implemented in software, 'neurones' are arranged in networks determined by their connectivity. Inputs to the software are processed by the network and an output produced. Like a brain, the network can 'learn' or be 'trained' to reconfigure itself and produce a desired result. Software using this method seems to be very good at interpreting complex data-sets, such as the stock-market. Examples of recent neural network work related to construction include: cost estimation (Hegazy & Ayed 1998), quantity estimation (Yeh 1998) and information content recognition (Modin 1995).

Kelly describes an "out of control" Reactive AI called "Swarmware" (1994, p.31). Advocates of Swarmware seem to believe that a complex system (even an intelligent one) can be built from small, usually independent, units following simple pre-programmed rules. The desired system output is then emergent from the behaviour of the swarm.

Other Reactive approaches include: Fuzzy logic (for example: Miroslaw 1998) and Genetic Algorithms. To summarise, Reactive Systems are particularly suited to tasks such as: Handwriting Recognition, Machine Vision and general pattern analysis.

In concluding this section, it is useful to consider the relative merits of the discussed methodologies:

- Where the goal is supreme *control* and the need to monitor, or modify, are paramount, Deliberative AI seems the most sensible choice.

- Where the goal is supreme *adaptability* and other considerations are secondary, Reactive AI provides an answer. (Kelly 1994, p.31).

Reactive systems may well prove productive for future construction software tools. Deliberative techniques seem currently to be the safest choice for developing a simple, usable application.

Appendix 16: Details of AI approaches implemented in Procession v1.0

The Significance of Deviance Algorithm (as used in Procession VTK prototype v1.0) calculates the significance of an individual project task to the current total value for a specific deviation parameter in a specific project. The mathematical approach utilised is based on statistical methodologies, such as ANalysis Of VAriance (ANOVA), where the term 'deviate' is the actual value for an individual score X, minus the average score for this group of scores, i.e. Xi – Mx (Lowry 2000, chapter 13). In Procession v1.0, this value has been termed 'the significance of deviance', which is a current snapshot of significance. By using the current value to increase or decrease an on going record of previous significance (the legacy archive), Procession v1.0 can 'learn' from its runtime experiences. Figure 0.12, shows the flow of data from CSV file to 3D visual structures, via significance calculation and up dating of the legacy archive.

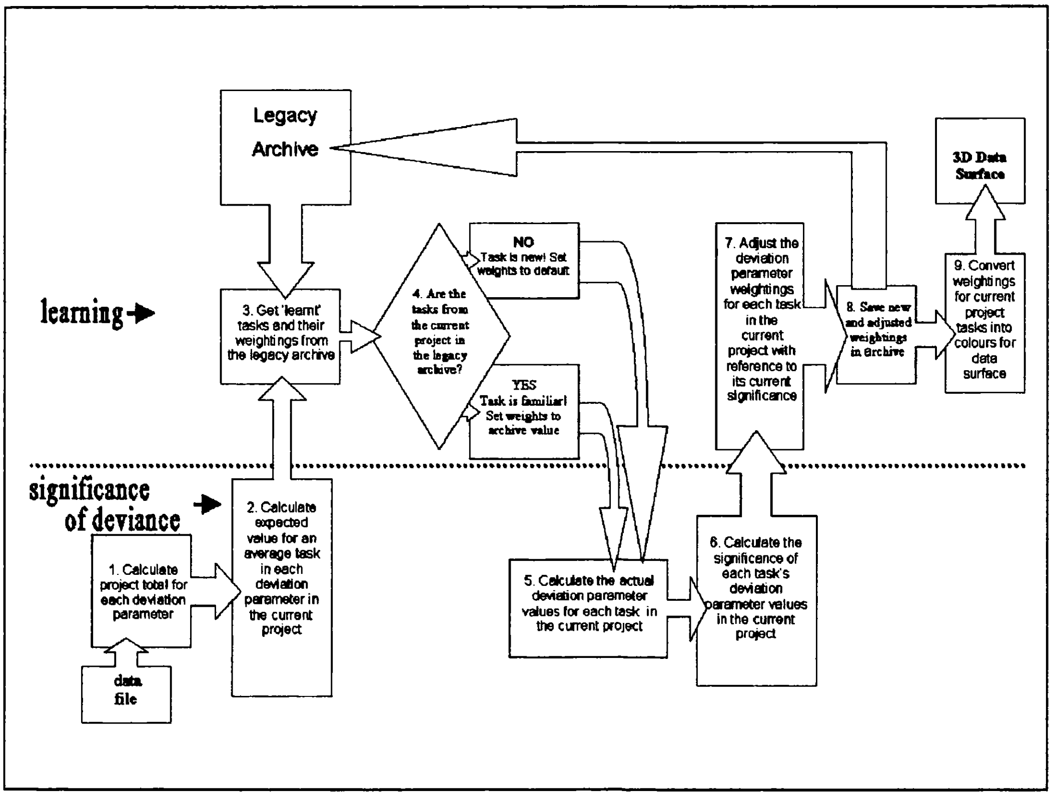


Figure 0.12 Procession v1.0's symbolic internal data-flow for calculating the significance of deviance and 'learning', source: author.

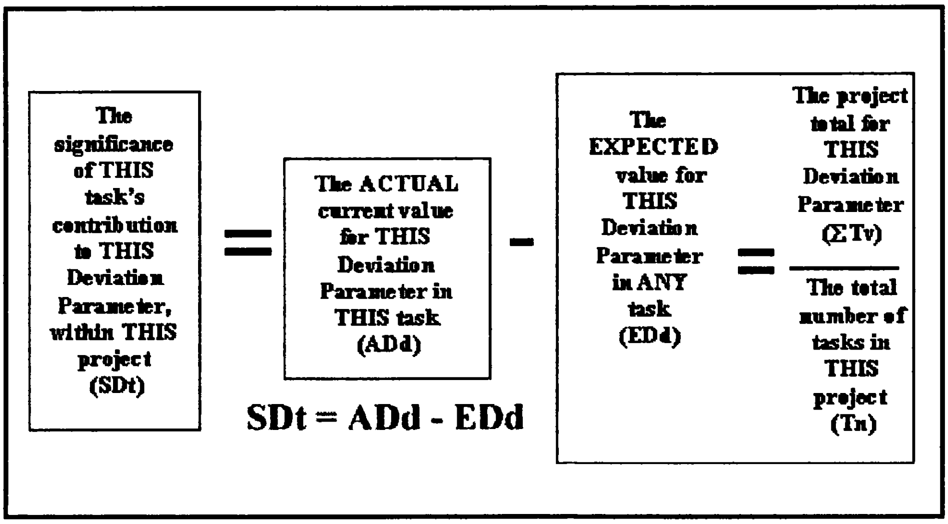


Figure 0.13 Procession v1.0's Significance of Deviance Algorithm, source: author.

A more in depth explanation of the significance calculation now follows. Figure 0.13 provides the formula for Procession's Significance of Deviance Algorithm. The following definitions have been assumed:

SDt = The significance of a specified task's (t) contribution to a specified deviation parameter, within a specified project

d = a specified deviation parameter

p = a specified project

t = a specified task

Tn = the number of tasks in a specified project

Tv = the current value of a specified deviation parameter value (d) for a specified task (t) in a specified project (p)

ADd = The actual current value for a specified deviation parameter in a specified task

EDd = The expected value for a specified deviation parameter (d) for any task (t) in a specified project (p)

$EDd = \sum(Tv) / Tn$

The resulting significance value can be interpreted as follows:

- A positive result value indicates that this task is making a greater than average contribution to the total for this deviation parameter. Therefore, the weighting should be increased.

- A negative result value indicates that this task is making a smaller than average contribution to the total for this deviation parameter. Therefore, the weighting should be decreased.

	DEVIATION PARAMETERS	
TASK NAME	COST VARIANCE	DURATION VARIANCE
Plumbing	£4.00	1 day
Electrical	£2.00	3 days

Figure 0.14 Example values for two deviation parameters, source: author.

Figure 0.14, shows an example of a progress report from a simple two-task project. The only deviation parameters under consideration here are 'cost variance' and 'duration variance'. For both of the deviation parameters, the total value is calculated across all tasks ($\Sigma(Tv)$). In this case, the total cost variance is six and the total duration variance is four. Next, the total number of tasks contributing to each deviation parameter is determined (Tn): cost variance has two tasks and duration variance also has four. If all tasks were contributing equally to the total deviation values, you would expect each task to have an individual cost variance of three (six divided by two tasks) and a duration variance of two (four divided by two tasks). These are the expected deviations for cost variance and duration variance in this project, at this time (EDd).

The C++ code exert in Figure 0.15 demonstrates Procession v1.0's calculation for expected deviation.

```
// first calculate project deviation parameter totals
for(v=0;v<theNumberOfDeviationParameters;v++)
{
        for (i=0; i<totalSubTasks; i++)
        {
totalProjectDeviations[v]=totalProjectDeviations[v]+subTaskParameters[i][v];
        }
        // now calculate expected difference factors for each deviation
parameter
        expectedProjectDifferenceFactors[v]=
        totalProjectDeviations[v]/(float)totalSubTasks;
}
```

Figure 0.15 Expected Deviance calculation, source: author.

However, it is unlikely that all tasks will contribute equally to the deviation parameters. They will probably have quite different values. The real cost variance and duration variance values for each task are called the actual deviations (ADd). For each deviation parameter, the difference between each task's expected deviation value and its actual deviation value, is the significance of deviance (SDt). The value of a significance of deviance indicates the significance of this task to the current total value for this deviation parameter in this project. The difference between the expected and actual deviations (the significance of deviance) is calculated by subtracting the expected from the actual.

The C++ code exert in Figure 0.16 demonstrates Procession v1.0's calculation for significance of deviance and its application to the weightings.

```
for(i=0; i<totalSubTasks; i++)
	{
		for(v=0; v<theNumberOfDeviationParameters; v++)
		{
			// adjust deviation parameter weightings by current
			significance
			subTaskWeightings[i][v]=subTaskWeightings[i][v]
			+ subTaskParameters[i][v]
			- expectedProjectDifferenceFactors[v]);
		}//close deviations loop
	}//close subtasks loop
```

Figure 0.16 Significance of Deviance calculation, source: author.

Thus, the calculations for the example in table 1 would be as follows:

Plumbing:

Cost 4 (ADt) – (6/2= 3 (EDd)) = a significance (SDt) of 1 (therefore increase weighting)

Duration 1 (ADt) – (4/2= 2 (EDd)) = a significance (SDt) of -1 (therefore decrease weighting)

Electrical:

Cost 2 (ADt) – (6/2= 3 (EDd)) = a significance (SDt) of –1 (therefore decrease weighting)

Duration 3 (ADt) – (4/2= 2 (EDd)) = a significance (SDt) of 1 (therefore increase weighting)

Procession v1.0 would draw the following conclusions from these calculations and colour the data surface accordingly:

- Plumbing is currently making a greater than average contribution to total cost variances and its weighting value for this deviation parameter will be increased by the significance value.
- Plumbing is currently making a lower than average contribution to total duration variances and its weighting value for this deviation parameter will be decreased by the significance value.
- Electrical is currently making a greater than average contribution to total duration variances and its weighting value for this deviation parameter will be increased by the significance value.
- Electrical is currently making a lower than average contribution to total duration variances and its weighting value for this deviation parameter will be decreased by the significance value.

The Legacy Archive

The Legacy Archive file contains a set of deviation parameter weightings for each task. On running Procession v1.0 for the first time, all of the weightings are set to a default value. The total number of significance weighting values corresponds to the total number of tasks multiplied by the total number of deviation parameters. Weighting values are increased or decreased by Procession, according to the relevant significance values. A positive significance value indicates that a task is making a greater than average contribution to the total for a given deviation parameter. Therefore, the corresponding weighting is increased. A negative significance value indicates that a task is making a smaller than average contribution to the total for a deviation parameter. The weighting is then decreased.

When developing Procession v1.0, two different strategies were tried for calculating the legacy archive weightings. In the first version, there was not a graduated correlation between the significance value and its corresponding significance weighting. Changes to the weighting were only of one unit, in response to the sign (+ or -) of the significance value. The weightings did not change in proportion to the significance value. This meant that two deviation values, both above the expected deviation but to widely varying degrees (i.e. a small and large peak in the data-surface), were being assigned the same weightings. Figure 0.17 shows how the calculation for this was implemented.

```
if (deviance significance (SDt) is a positive value)
{
weighting++
}
else
weighting--
```

Figure 0.17 Legacy Archive weighting calculation version 1, source: author.

Figure 0.18 shows that the prototype of Procession used for evaluation featured an improved calculation, where the weighting changed in response to the size of the significance value.

weighting = weighting + deviance significance (SDt)

Figure 0.18 Legacy Archive weighting calculation version 2,

source: author.

The legacy archive is an ASCII text file, stored in the root of the primary Windows hard drive (usually C:\ ). Each line in the legacy archive file is space-delimited and has the format: task name, weighting one (cost variance), weighting two (schedule variance), weighting three (variance at completion), weighting four (duration variance) and weighting five (work variance).

Figure 0.19 shows how the legacy archive might look for the project example in Figure 0.14.

```
#Legacy Archive file created by
#P r o c e s s i o n application
#1999/2000
#Do Not Edit!
#file format:
#<TASK NAME> <PARAMETER 1 WEIGHTING>
# <PARAM 2 WEIGHTING>...etc.
#Last Updated: Fri Mar 20 15:00:42 2000
#-----START DATA-----
Plumbing 46 57 32 48 51
Electrical 50 56 35 67 77
```

Figure 0.19 Example values for two deviation parameters, source: author.

Appendix 17: Remaining technical issues with Procession operational prototype v1.2

After development, Procession v1.2 was found to have the following bugs:

- User pressing the <return> key while the focus is on any of the editable text boxes, causes Procession to exit.

- The view data source box is not scrollable and therefore large data sets disappear off screen.

- Occasionally, after multiple files have been opened, the surface inappropriately displays a flat surface for all files. If this happens, left mouse clicking on the surface will still reveal the correct values. The only solution is to exit Procession and restart it again.

- The orientation for screenshots and print screen are not very accurate. This is because the TeeChart 3D functionality currently requires OpenGL to be turned off before the screen captures and this slightly alters the orientation settings.

- Certain graphics cards (for example Elsa Winner II) seem to have problems with Procession's OpenGL rendering. Symptoms include: 3D text not visible (or disappears at close distance) and lighting is too bright i.e. not using the spotlight specified.

- The values displayed on "mouse down" on surface are not 100% accurate. This seems to be a problem with TeeChart which may be resolved in their next release. The preferred solution would be for values to change dynamically, on "mouse over" surface.

- Last two tasks in project are not displayed. Solution: add two dummy tasks.

- Certain problems have been observed under Windows 95. It is advised that Windows 98 is the minimum recommended platform. However, the observations were:

  - Application not exiting correctly (process still active when <control><alt><delete> is run)- could be an Active X problem? Solution: kill process manually.
  - Occasional floating point error under heavy 3D load- OpenGL problem?


Procession
(Submission Version) v1.2
A 3D progress reporting application for
construction project clients
Windows 95 - Windows 98
Windows NT 4.0 - Windows 200
UCL
VR CENTRE
for the built environment
ic·parc
© 2000 steve north
steve@stevenorth.com