



## SE3D User Study

Lon Barfield<sup>1</sup>, Guillaume Belrose, Rycharde Hawkes, Peter Toft, Erik Geelhoed  
Enterprise Storage and Systems Laboratory  
HP Laboratories Bristol  
HPL-2006-183  
December 22, 2006\*

user studies, user  
interface design,  
utility computing,  
market-based  
resource allocation,  
3D animation, CGI  
rendering, SE3D

This study made a large amount of remote, computer-processing power (the HP Labs Utility Rendering Service) available to a group of animators to support each of them in the creation of a computer-graphics short film for the SE3D showcase. This event was organized by HP Labs in collaboration with Autodesk and the Watershed Media Centre based in Bristol (UK).

Animators were required to continually bid for computing power using tokens in an online market place. At peak demand times the resources became more expensive. Different market mechanisms were used and data was recorded about the animators' interactions with both the markets and the rendering service in general. Towards the end of the experiment, interviews were carried out with the users and their responses are summarised in this document.

\* Internal Accession Date Only

<sup>1</sup>The Usability Design Partnership Ltd.

© Copyright 2006 Hewlett-Packard Development Company, L.P.

Approved for External Publication

# SE3D User Study

Lon Barfield<sup>1</sup>, Guillaume Belrose<sup>2</sup>, Rycharde Hawkes<sup>2</sup>, Peter Toft<sup>2</sup>, Erik Geelhoed<sup>2</sup>

*lonbarfield@gmail.com*

*{guillaume.belrose, rycharde.hawkes, peter.toft, erik.geelhoed}@hp.com*

<sup>1</sup> The Usability Design Partnership Ltd. <sup>2</sup> HP Laboratories, Bristol.

## Abstract

This study made a large amount of remote, computer-processing power (the HP Labs Utility Rendering Service) available to a group of animators to support each of them in the creation of a computer-graphics short film for the SE3D showcase. This event was organized by HP Labs in collaboration with Autodesk and the Watershed Media Centre based in Bristol (UK).

Animators were required to continually bid for computing power using tokens in an online market place. At peak demand times the resources became more expensive. Different market mechanisms were used and data was recorded about the animators' interactions with both the markets and the rendering service in general. Towards the end of the experiment, interviews were carried out with the users and their responses are summarised in this document.



# Table of Contents

|     |   |    |
|-----|---|----|
| 1   | Executive summary.....  | 1  |
| 1.1 | Outline of the experiment.....  | 1  |
| 1.2 | The applicability of market place economics to shared computer resources.....     | 1  |
| 1.3 | Utility Rendering Service usability.....  | 2  |
| 1.4 | Adoption of interaction design early on in the project life-cycle .....           | 2  |
| 2   | Introduction .....  | 3  |
| 2.1 | Terminology .....   | 3  |
| 2.2 | Computer animation.....   | 3  |
| 2.3 | Internet auctions.....  | 3  |
| 2.4 | System Overview .....   | 3  |
| 3   | The experiment .....  | 5  |
| 3.1 | Equal share .....   | 5  |
| 3.2 | Short term bidding (STB), two methods .....                                       | 5  |
| 3.3 | Long term bidding (LTB).....  | 5  |
| 4   | The Subjects .....  | 6  |
| 4.1 | Subjects work background .....  | 6  |
| 4.2 | Subjects location .....   | 6  |
| 4.3 | Subjects previous experience of Maya.....   | 6  |
| 4.4 | How fast is your internet? .....  | 6  |
| 4.5 | Work context for this project.....  | 6  |
| 4.6 | Subjects initial motivation for involvement .....                                 | 7  |
| 5   | Rendering service .....   | 8  |
| 5.1 | Stresses and strains .....  | 8  |
| 5.2 | Work flow used for this project .....   | 8  |
| 5.3 | What are animators like?.....   | 9  |
| 5.4 | Working with the URS.....   | 12 |
| 5.5 | Versioning .....  | 15 |
| 5.6 | Storage limits.....   | 19 |
| 5.7 | Comfort with the system .....   | 19 |
| 5.8 | Usability (of the URS) .....  | 22 |
| 5.9 | Features .....  | 22 |
| 6   | Bidding in general.....   | 25 |
| 6.1 | GV or share, which auction did subjects prefer? .....                             | 25 |
| 6.2 | Long term bidding and short term bidding, which auction did subjects prefer?..... | 25 |

|      |  |    |
|------|--|----|
| 6.3  | Pricing and predictability .....                     | 26 |
| 6.4  | Commercial use .....                                 | 27 |
| 7    | Short term bidding .....                             | 30 |
| 7.1  | Usability .....                                      | 30 |
| 7.2  | Bidding and managing finances .....                  | 30 |
| 7.3  | Features .....                                       | 31 |
| 8    | Long term bidding .....                              | 33 |
| 8.1  | Level of use .....                                   | 33 |
| 8.2  | Influence on work patterns .....                     | 33 |
| 8.3  | Usability and understanding of LTB.....              | 34 |
| 9    | Support.....   | 35 |
| 9.1  | Community.....                                       | 35 |
| 9.2  | Online forum .....                                   | 35 |
| 9.3  | Mentoring.....                                       | 36 |
| 9.4  | Training.....  | 36 |
| 9.5  | Downtime.....  | 36 |
| 10   | The organisations.....                               | 37 |
| 10.1 | Hewlett Packard.....                                 | 37 |
| 10.2 | Watershed and the project intermediary.....          | 37 |
| 10.3 | SE3D.....  | 37 |
| 11   | Conclusions .....                                    | 39 |
| 11.1 | Profiling the target user group .....                | 39 |
| 11.2 | Supporting the community spirit .....                | 40 |
| 11.3 | Making users follow chosen behaviours .....          | 40 |
| 11.4 | Experience with rendering.....                       | 40 |
| 11.5 | Experience with bidding.....                         | 41 |
| 11.6 | Interaction design and the development process ..... | 41 |
| 12   | References.....                                      | 43 |

# 1 Executive summary

## 1.1 Outline of the experiment

A large amount of remote, computer-processing power was made available to a group of animators to support each of them in the creation of a 3D rendered short film. Animators used the Utility Rendering Service, a tool developed at HP Labs Bristol for remote rendering of 3D computer graphics content over the Internet. With the Utility Rendering Service (URS) [1], animators were able to efficiently transfer their source content created with Autodesk Maya [2], render digital frames for their project and later download the completed images for further processing tasks such as compositing (see 2.4). The URS was made available to 12 groups of animators as part of the SE3D (“seed”) project [3], a showcase organised by HP Labs in collaboration with Autodesk and the Watershed Media Centre [4].

The supply of available computer processing power was often not enough to satisfy the combined demand of the animators unless they adapted their group behaviour to make more optimal use of the computing resources. This optimal use revolved around levelling the demand over time by means of planning their combined workload. This group behaviour was encouraged by providing the animators with tokens that they could use to bid for resources in an online market place. The price of the resources at any particular time was related to the demand thus encouraging the animators to schedule their use of resources at times when there was less demand and the price was cheaper.

Throughout the duration of the SE3D project different market mechanisms were used and data was recorded about the animators’ interaction with the system. Towards the end of the experiment interviews were carried out with the animators and the responses are summarised in this document. Of the initial group of 12 animators, 1 failed to achieve satisfactory completion of the project and so only 11 were included in the study

The remainder of this section outlines the key outcomes from the feedback.

## 1.2 The applicability of market place economics to shared computer resources

The applicability of the market place to the optimal use of computer resources can be considered from three different viewpoints:

### 1.2.1 Technically – was it possible?

Despite basing the project on a combination of novel technologies there were no serious, insurmountable technical problems with running a multi-user online market place for resources. This was due to the initial design of the infrastructure, the availability and skill of the staff involved during the project and the incorporation of a scheduled maintenance downtime for the service.

### 1.2.2 Subject acceptance – did the users like it?

The subjects found only minor problems in their use of the system; both the management of rendering jobs and the auction process to access resources for the rendering jobs. User acceptance problems were mitigated to some degree by the fact that the users were getting large amounts of processing power that they would not normally have access to.

The main area where users struggled with usability was when dealing with higher-level issues, in particular how the system matched the animators’ real-world work contexts.

Although the subjects could use the system and see the benefits of using a micro-market place to optimise resource distribution, they identified key problems with money and planning. On the money front they did not see how the service could become a service that operated in the open market. All the subjects saw themselves as small players whose funds available to buy computing resources would be much smaller than larger corporate players (in a real context, the URS will be sold to Small to Medium animation companies so this will be less of an issue). Another important consideration that

came up in the interviews was that the subjects' behaviour would be very different if the economy was based on or linked to real money instead of tokens.

On the planning front a key task in computer animation is minimising uncertainty in all areas. Although the system provided tools to help in this, the big underlying premise of the system was that by using it and planning efficiently you could get cheap resources but that prior to purchasing, there was no guarantee of the availability of those resources and no guarantee of the price of those resources. However, once purchased, their delivery was guaranteed.

### **1.2.3 Correct use – did it do what we wanted it to do?**

Having established that it could be done and that users on the whole were willing to use it with reservations about the applicability of the context, the final question was; 'were users using it in the way that the designers intended it to be used?' To a great extent they didn't and there were two observations here. Firstly, systems need to be designed to enforce desired behaviours not merely to suggest them. Secondly, interaction design should be applied at an early stage in the project to encourage desired user behaviours (see below).

## **1.3 Utility Rendering Service usability**

The URS was regarded as generally easy to set up and use with a few exceptions. The main one was the limited interface for specifying files for the rendering task. The URS exposed some command-line interaction through the GUI instead of direct manipulation and it made no use of default naming. This meant that subjects spent a lot of time specifying simple set-ups, knowing that it could be done more efficiently.

Other missing features fell into the categories of more control over the task, e.g. adjusting processor allocations and job priorities, and more insight into the progress of the task, e.g. estimated completion of jobs.

A final point is that the pre-use training was complex and led subjects to be concerned that the interface would be very complex.

## **1.4 Adoption of interaction design early on in the project life-cycle**

When embarking on a product that will be used by end users, either in a true commercial context or as part of an evaluation phase, interaction design needs to be involved to ensure adoption by the user group and to ensure that the system is used in the manner in which the developers wish it to be used.

This involvement needs to be early stage as decisions taken in the early stages of product realisation can have a great influence further down the development process. Indeed many aspects of the product design are the result of lack of design rather than bad design per se. Of particular interest in this system was the users' behaviour with the upload and versioning system. Their behaviour was shaped by their previous experience of uploading and versioning activities and the use of the terms 'upload' and 'version' in this new context. This led to user behaviour that was very different from that which was expected and desired, behaviour which could have been avoided had different terms and interaction been used.

Another important part of interaction design is the profiling of the users, in this case the user group was well defined and in the post-experiment user interviews it emerged that they had well-defined and identifiable traits that could have informed the design process were they known in the early stages of product development.

## 2 Introduction

### 2.1 Terminology

The *subjects* are the people/organisations participating in the experiment. They are referred to as: TJ, DL, AB, BL, SD, G, JH, UNI, AP, K and MB; FOCUS – comments from the focus group.

### 2.2 Computer animation

Many industries have undergone a paradigm shift when they have moved to the digital arena; this usually involves the idea of individuals or small teams doing something that was once the preserve of much larger companies. The earliest example was the shift to desktop publishing (DTP), previously large companies had in-house document and report publishing groups, when DTP came along anyone with a computer and printer could do the same. Since then similar things have taken place with photographic manipulation, video editing, sound and music creation and book publishing. Larger and older companies have taken part in this movement, both in providing digital tools to support it and in acquiring or teaming up with the smaller players to get their creative input. Computer animation is now undergoing a similar revolution, the tools and equipment have come down in price and the last expensive hurdle; rendering power, is starting to be sold as a service, leaving companies free to invest in people:

*Again [it's] the people who are the most expensive, and it's great in a way to have that redressed because we should spend more money on the people really, and talent. Um, and not have to worry about making our images, [BL]*

The introduction of affordable utility rendering services could ensure that there's a future for smaller animation houses. Today, animation is largely in the context of a production house, but with widespread utility computing, virtual communities could come together from all over the world to work on large projects.

### 2.3 Internet auctions

In parallel with the developments in digital media is a growth in commercial applications on the internet. As well as supporting conventional commerce models the multi-user, real-time nature of the internet means that new paradigms of commerce can be used or existing paradigms migrated into new areas. One such is the growth in the user of auctions as a transaction model, eBay being the classic example with a global annual figure of more than 40 billion dollars of goods traded in 2005.

There is also a growth in other, more complex types of internet auction such as Google's AdWords, which allows users to bid for targeted advertising space on the Google site where their advert is coupled to the search terms that the users are searching for and they pay for their placement through a bidding system.

### 2.4 System Overview

Figure 1 shows the relationship between an animator their URS. Having created an animated sequence in Maya, the animator used a small GUI client application to manage the transfer of the models, textures, animation data, etc. to the URS where it was stored in an asset store. Having done this, animators could then submit a job request to the service to render out the desired animation, the rendered output being transferred back to the animator's desktop via the same client application. Multiple jobs could be queued up using the client with the service processing them in the order they were submitted. The client was also used to monitor the progress of the submitted jobs and get detailed information about the related rendering process. The asset store and the upload of data to the service were implemented using Elephant Store (see 5.5.1).

Although the URS was always online, no rendering could be done unless the user had "bought" some machine time to do so. The GUI interface to the resource market was called Mercado and enabled



users to place bids, monitor the auction process and view historical information on bids they had placed, auction results (and therefore available resource levels), credit balance, and so on (see 3, 6, 7 and 8). The reservation system automatically changed the number of machines allocated to a service based on the auction results.

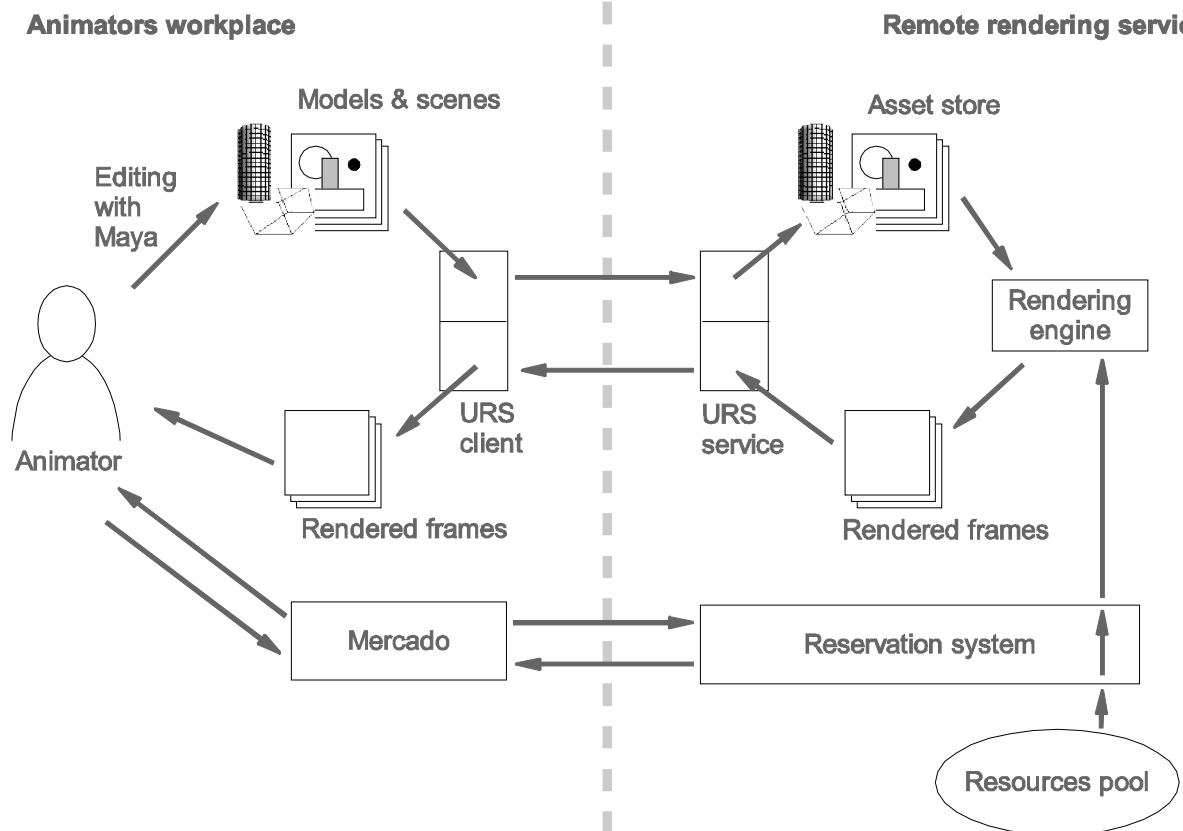


Figure 1: System overview focussing on the Animator-Service workflow.

## **3 The experiment**

The experiment comprised of three different phases of auctions:

### **3.1 Equal share**

In the first phase the subjects were allocated computing resources following an equal share system with all participating subjects receiving an equal share of what was available.

### **3.2 Short term bidding (STB), two methods**

The second phase saw the introduction of short term bidding, where subjects could bid for a share of the resources available in the next hour. The subjects were divided into two groups (A and B) and each group was allocated a subset of the resources to bid for and given a different type of short term bidding mechanism to bid for those resources. Group A used proportional share where the resources in their pool were allocated according to how much they had bid, thus if you bid a lot and everyone else in group A bids a little then you get the majority of the resources. The other group, group B, used a method of short term bidding called Generalised Vickrey (GV) where their bid consisted of how much they wanted to pay and how many resources they wanted to get for that price [6]. This method of bidding was more optimal than proportional share (as used by group A) but it was slightly more complex to use and understand.

### **3.3 Long term bidding (LTB)**

For the third phase both groups (A and B above) came together for the introduction of long term bidding. All the subjects used the same method of long term bidding; proportional share. Furthermore they all used the same method of short term bidding as well for this phase, this was also proportional share. This meant that group B would have used two different forms of short term bidding in the course of the experiment, first of all Generalised Vickrey and then, in phase three, proportional share.

## 4 The Subjects

### 4.1 Subjects work background

There was a good range of subject backgrounds, some of the subjects had experience of hands-on 3D animation, others came from the production side of the discipline being more concerned with overseeing the animation process. For a few this was their first taste of 3D animation although they had been involved in related fields such as web animation and digital music and performance.

### 4.2 Subjects location

The majority of the subjects were based or co-located in Bristol. Others were based in Wales or the North-West.

The community features (the online forum, events, festivals) seemed to intensify the existing feelings of community and subjects outside the Bristol animation community, who couldn't make it to all the events, had to rely on the more remote side of community support which they found less satisfactory than those based in and around Bristol.

One subject with a weekly work schedule between London and Bristol found that this fitted in well with the weekly cycle of scheduled downtime and the weekly window for long term bidding.

### 4.3 Subjects previous experience of Maya

Half the subjects had used Maya [2] before. Of those that had not, some were not actually doing the animation but simply supervising it. The majority of the problems experienced with Maya came from those with no previous experience of the system.

### 4.4 How fast is your internet?

Most subjects were on broadband or better.

### 4.5 Work context for this project

Many of the subjects were freelancers operating on their own or in groups. Some of them were in charge of a group. Groups were either in-house or geographically distributed.

If we consider the usage in an abstract manner we find that there are several different scenarios of use of the system. Two orthogonal classifications are one: are they single or group-users, and two: are they operating-alone or as part-of-a-larger-organisation. This yields four groups, but a further refinement of the group-users, again orthogonal, is whether the group of users were physically in the same space or distributed. This leads to two of the four groups identified above being split giving six groups in total:

- Single user operating alone as freelancer
- Single user operating within a larger organisation
- Group of users (two or more) operating as a single unit
- Group of users (two or more) operating within a larger organisation
- Group of users (two or more) operating as a single unit but distributed
- Group of users (two or more) operating within a larger organisation but distributed

To some extent the last two groups are very similar since a distributed group will be cut off from the larger organisation it is part of and will function in a similar way to a distributed group operating as a single unit.

## 4.6 Subjects initial motivation for involvement

When asked about their initial motivation for getting involved almost all the subjects said they wanted to make a film, only one gave a different reason and that was curiosity. Extra factors include the relaxed attitude to the ownership of the films produced.

*The thing that amazed me the most was that someone was prepared to help out on a film and not want to own it. For me that was an enormous pull because other places where I've worked, any idea they have, they want to have it, and own it, and keep it.  
[DL]*

## 5 Rendering service

### 5.1 Stresses and strains

Within the task of animation in general stresses and strains were many. Both in the conventional way of doing things (there is no single well defined work method, see 5.2) and in the methods adopted when using the rendering service. For freelancers working on their own, computer animation can be a very vertical activity, involving everything from client acquisition through to post-production which can lead to many frustrations.

Stepping outside this pipeline and being an animator in a larger company had its own frustrations. Subjects talked about having little contact with the client and little opportunity for creative input to the whole process.

Subjects working in such groups as producers rather than animators found problems breaking the tasks up into fragments to enable a team to work on aspects of it in parallel. This involved the isolation of aspects of the scene and models and the overall co-ordination of the process. There were also issues with managing remote groups.

On the technology side the animator's task is a complex one in terms of data management. They are dealing with many files that have complex interrelationships. These files often have many versions as they struggle to get a particular feature/lighting effect etc. just right. The batch nature of rendering means that while they are waiting for one version to be rendered they are often engaged in parallel tasks such as altering the files for another version. Specific, technology related problems included rendering incorrect scenes, and managing large numbers of files. A common worry of animators using rendering systems is that, due to the lack of feedback and the batch nature of the rendering task, you can waste a whole day rendering and then discover it's not what you were expecting:

*You can send it off to the rendering farm, even if it's kind of local or in the same building, and that's it and you wait and the next day you check your frames and there's nothing in between really. [SD]*

Although this was only cited by one subject when actually asked about stresses and strains, other subject's responses to other questions pointed to this being a generally important consideration for animators as a whole.

### 5.2 Work flow used for this project

For small animation companies, there appears to be a lack of any industry-wide standards, either designed or ad-hoc, for the workflow or for the management of digital resources in the workflow. What seems to be happening is piecemeal adoption of working methods from larger practices according to where the animators were working before becoming freelancers.

*I want to change how the schedule works. Animation production has got a c\*\*\* work schedule and needs to change. It's not viable. [FOCUS]*

*Some of them would come in and say 'I've worked in an animation studio, and what you do is you stack up all your work and then you do your renders at the end of the project', well that's not true. [FOCUS]*

There was little consistency in the work flows that subjects adopted. They seemed very open to adapting their work methods to the context and the technology available. Several of them spoke about evolving methods and using ad-hoc methods as they adapted to the new service.

*It kind of evolved, in fact almost the whole production process is still evolving I have to be honest, it's still evolving. [BL]*

One area where the URS impacted the subjects working practices was the speed of the rendering. Animators are used to working long hours and waiting long hours for renders to finish. Having the fast rendering of the URS meant that time normally spent waiting around for a render to finish could actually be put to productive use:

*We ended up putting in quite long days, and working weekends and that's traditionally your rendering times, so it meant we could just concentrate on getting the animation how we wanted it. [TJ]*

*You can access the frames, have a look at them, make sure they're rendering properly. And you just line your stuff up in a queue, just make sure that it's going, just sit there and let it do it's thing really. Go back and carry on with your work. [TJ]*

Another area of impact that yielded free time was not having local machines tied up with rendering. One subject cited the situation where animators used their own machines for rendering. The scene was split up between workers and they each rendered a small part on their own machine, thus tying each machine up:

*But one thing we found very much at Company X was that we were all rendering locally and so obviously when your machine's rendering there's nothing else you can do with it. And so to be able to have the files that you need, send them off somewhere, get that going, and then you can just get on and carry on with your work. [DL]*

A key conclusion is that we need to know more about animators work practices. Either to support them all or to design a better work flow and prescribe and support only that. Supporting many work flows can be achieved by including many ad-hoc systems for labelling and tracking things. (Think of the use of filenames for versioning and on the Mac the colour coding and the notes field for icons) one subject group were using a simple text file to log and control updates to their upload directory (see 5.5.3). Prescribing one work flow would involve designing it well and making the interaction so good that it becomes the way that people want to work.

## **5.3 What are animators like?**

During the subject interviews it became apparent that animators are quite a specialised group of users with their own traits. The main ones observed were:

### **5.3.1 Animators are very visual/tangible**

Animators are people who are used to thinking in very visual terms; they deal with objects, things, movement and space. Consider the following, very visual, comment on the abstract subject of the maths behind the bid allocation:

*Some blokes in lab coats somewhere calculate with their massive Tefal heads. You know. [TJ]*

When discussing working remotely two of the ten subjects made reference to animating on the moon – another highly visual and cartoon-like reference. The subjects are more accustomed to thinking and working with iconic and visual representations of data than with command line and textual interfaces. As such an interface based on textual commands (as we shall discuss in 5.4.1) is not the ideal approach and a design with more of a direct manipulation element would seem to be more appropriate.

An interesting follow up to this is the comments from the university group in Brazil. They were the only non-animation subjects to use the system. They came from a programming background and were only submitting a limited number of test jobs:

*The client interface (URS client) is efficient and simple; in some cases, too simple. It could have, for instance, a more practical way for collecting global processing logs (logs of all frames processed) and view job status. [UNI]*

The animators were generally aware that the interface was presented in a very 'programmer' type style.

*If you just sat down and thought about it in a very programmers sort of way that the way they would do things. [AB]*

*That's the whole thing about it, in the politest and most respectful way I can say it, it's been written by engineers. [BL]*

When all they really wanted was something simple and visual again, almost cartoon like:

*Really I just needed a big green button that I could press and just have the images on my hard drive some time the next morning. [AB]*

On a superficial level it appears that there could be parallels between these aspects of animator behaviour/preferences and those of children. It would be interesting to investigate if the body of research on user interfaces for children could be applied to user interfaces for animators.

### 5.3.2 Animators want to use it, not hear about it

During the project there was much explanation by technicians to animators. Animators are not technical people. Animators seem to be less interested in explanations and more interested in getting hands-on use of the bidding system to understand it.

*I was incredibly confused when they described it to me, when we went down to one of the events, ... and I just got very confused, it looked incredibly complex again. It was actually very simple once I actually had the software in my hands. [SD]*

Maybe there is a case for going straight to hands-on use of the system rather than telling them about it.

This is further supported by the subjects' comments on the use of the help system in the bidding marketplace; it was hardly used at all, only when the subjects hit problems.

### 5.3.3 Animators exhibit 'baby-sitting' behaviours

Animators are keen 'baby-sitters', they like to be in close contact with the processes and stay on top of what is going on. This behaviour is further influenced by the technology they use and it goes on to influence their work patterns. Baby-sitting (being present to exercise a very hands-on form of process management), seems to be an innate part of the way that animators deal with rendering. This baby-sitting applied to the animation process as a whole:

*Without stereotyping animators, without being detrimental to them, it's a controlling profession. It's based on having absolute control over very small details and I think the nature of them is that they're going to want to control things. [AP 10]*

*Our names going on this project, we want it to be the best it can be and the more in control we are of it, on top of it, managing it, then the closer we are to that goal. Rather than farming it out to someone else and then kicking yourself. [TJ 16]*

*I think the more you are on top of it the less anxiety you have because you know exactly where you are. [TJ]*

Baby sitting also applied to keeping an eye on things during an actual render:

*[subject describing an in-house render farm] you just kick everyone off their machines at six o'clock, copy all the database over to every machine, which, you know, took an age, set everything going and you'd just have to go round like a round robin to every machine and make sure everything hadn't crashed, and it was still going... it was an absolute nightmare. [DL]*

*The way we operate we like to sit on our work and literally see it as it happens rather than just sort of like set it off and come back to it the next morning. [MB 14]*

This working pattern seems to be difficult to drop even working in a context that doesn't require it. Furthermore, dropping it is not actively supported by the design of the user interface of the URS. Subjects found that even with the URS, they have a sense that they want to stay on top of what is going on and work with the render farm in a very micro-scaled, reactive manner. Often submitting small jobs and then resubmitting them again based on the results of the first submission. This work pattern led some subjects to avoid using the long-term bidding system altogether:

*It's a working practice on our part probably, because we spent six years at a company where you render something and then show it to a client. [TJ]*

*It was a much more continual, hands-on process with us. [TJ]*

There is a strong case for creating tools to give animators the things that baby-sitting gives them and thus help minimise it. We need tools to make it truly batch oriented but in a controlled and predictable manner.

**Confirmation** - Users need to really set up every last detail, confirm that the system understands exactly what they want and then let the system get on with it. E.g. automatic previewing of frames ('this is what the first frame will look like, this is the last, click if you want to cancel the render job').

**Control** – Users need control over what is happening on the URS, the ability to easily cancel render jobs, to pause jobs, to adjust the priorities and possibly to set up more complex control patterns. ('this job is vital, rush it through, then finish whichever of these three can be done the quickest, then get on with the other two'). Again this would enable them to ensure it was doing what they wanted and they could then leave it and do something else.

**Alerts** – Rather than going to have a look at how it is all going every five minutes, users need to be alerted when, and if, things need looking at or adjusting. They need support for Email and SMS updates on what is happening and possibly a web interface to the system to enable them to do things off-site.

### 5.3.4 Getting results quickly

When faced with any task a common reaction is to want to make a start rather than contemplate it/plan it etc. This feeling seems even more pronounced in animators, they want to get on with things quickly, often at the expense of good planning or learning about things that will aid them in that task. In this project the attitude sometimes interfered with attending the mentoring and training sessions and with the overall planning of the task.

*It's always a question, with every project- 'shall I stop and do this properly or shall I just keep going and hope it gets there?'. [AP]*

*But there's always this trade off of- 'do you want to go and talk to people, or do you just want to animate?' And the animators just want to animate, because all they can see is a job list, they don't really want to learn because they think they know animation already. [AP]*

*You've got your skeleton and you've got your skin, and the way that you rig the character [build it] can give you terrible problems later/ [K]*

*We were finding we'd been backed into corners, someone had built something a certain way and because that had been built in a certain way to do it quickly in two days that would add days and days to the project further down the line. [AP]*

*Normally you go into a production and you really do lock your story board before you start making anything, but we piled in and we were supposed to have a story-board but the story-board wasn't done but we piled into getting animating anyway. [AP]*

#### 5.3.4.1 Being overambitious

A related theme is having big ambitions. The subjects who had little previous experience of animation experienced some difficulties in adapting their previous work methods to the world of 3D animation, especially as they often had great expectations of the speed and flexibility of 3D animation.

*People hear 'render farm' and the first thing they think of is Pixar, and 'we can do anything'. [DL]*

The subjects with more experience handled things in a much simpler manner and other subjects came out of the project realising that they had to work within well defined limitations.

*People say 'ooh we can't do this and we can't find that' and maybe it was because I kept mine fairly simple, I'm not entirely sure. I just remember thinking 'I haven't had any of those problems'. [DL]*

*We wouldn't try and take on Pixar and DreamWorks at their own game, with a team*



*who don't quite know what they're doing. [AP]*

*My business model at the moment is built on creating animations that are less demanding. Back to the old way of doing it where you just you don't light so much and you don't do so much texturing. [AP]*

## **5.4 Working with the URS**

### **5.4.1 Filenames**

One common feature that came to light in animators work practices was the use of the filename for conveying extra data, particularly relating to versioning.

*With each scene file I'm saving, you know, thirty to fifty revisions for uploading on to the service, so with each change I make, even if it's just a little change or a little bit of movement or something I'll save that as a different file name. [FOCUS]*

This practice of using the file name to record scene and version information was also a possible cause of problems as the rendering service didn't react well to certain characters being used in the filenames. Some animators come from a Mac background where it is common practice to have filenames that are long, and include spaces and other punctuation; a practice which can cause problems for UNIX based systems such as the URS.

*Mostly the problems that we ever had with dealing with different pieces of software is just the naming of the file so I just took all the spaces out and made them underscores or deleted them, I can't remember what I did, and it was fine. [DL]*

*This is something left over from Company X as well, you never save anything with the same filename. If you change something, it's the next version- so it's shot 10, version 4 or whatever. [DL]*

The fact that the interface was command line and not browsable meant that subjects had to be very careful about typing in the correct name

*You have to cut and paste the file name in because I had quite complex file names so I can't afford any mistakes. [BL]*

*You had to type in the file name and obviously the more you have to type in, the greater the degree of error, and there's just more room for mistakes. [K]*

Animators insisted on using this file naming convention that they were used to even when they thought it might be compromising the advantages offered by Elephant Store (see 5.5.1):

*It wouldn't look for any differences because it was a different file name. So that much warranted memory saving thing never actually worked out really. [AB]*

In actual fact the Elephant Store works by comparing the content of files, not the names and thus the benefits it offered were still applicable. There is more on this problem in the discussion of browsable file systems (see 5.9.3).

### **5.4.2 Post production**

Some of the animators were using post-production techniques once the bits of rendering were finished. Sometimes the rendering would only be used to supply a small part of one layer in the end shot.

As well as looking at the rendering process we should also be aware of the context of the rendering process; what happens before the process and what happens after the process, and how can the design of the rendering engine assist the user at these points. One area identified is the issue of renaming multiple files output from the URS in order to feed them into a post-production tool.

Some of the subjects reported problems with the naming conventions adopted by the URS when returning frames. With any system it is useful to look at the workarounds; the way that people use the tools that are part of the interface design in new ways to suit their own goals and tasks rather than

those dictated by the interface. There was a workaround here where one of the subjects artificially alters his frame numbering to avoid file numbering problem:

*So, I had then problems when I downloaded it because it would start at 1,2,3,4 and actually I need double digits in my frames for my compositing. So I can tell Maya to do that, but I couldn't tell the URS to do that, but I got round it by moving everything to start at frame 100. [BL 6]*

### 5.4.3 Directory structures

Another workaround was having two directories of the operating files. A •messy• working version on the local computer and then a •clean• directory also on the local computer that was used to feed the upload. The files on the server were thus a mirror of a tidied subset of the real working files.

*Two folders, one is where I've got all the Maya files as I'm working, then I have another folder and its called my upload folder.*

*two folders, say for simplicity, one is where I've got all of my Maya files as I'm working, then I have another folder and it's called my upload folder and I put everything in that that I'm going to send up to the render service. [BL]*

More than one subject reported that they followed on with this approach even when having two directories on the client machine was a direct cause of user errors:

*I might have changed a source image and not updated it on the mirrored folder that was the only problem. [AB]*

### 5.4.4 A development tool

The conventional batch oriented model of rendering is •design – specify – design – specify – render•,

*The rendering has always just been the very last stage so never really assumed that much significance, only if you had problems. [AB]*

This batch oriented approach is based on older, slower rendering systems. With the speed of the URS some of the subjects, two in particular, were adapting their work practices to take advantage of the speed of rendering. They were doing things in less of a batch oriented way and more of an iterative way; •render – adjust – render – adjust•. As a result they tended to rely more on the short term bidding than the long term bidding since this fitted in better with this mindset and manner of working.

*Its kind of crazy no to just chuck it through [the rendering service] anyway, whether its actually finished or not [...] definitely a developmental tool, yeah. [TJ]*

*The little and often approach is just something that works for us. [TJ]*

*The rendering service allowed us to be more ambitious. [TJ]*

*If the rendering resources are tight and you get back a scene and you forgot to put the light on or the texturing isn't quite right here or there, its a really big drag. With this, you just put it through again. [TJ]*

*You don't have to compromise do you? You can almost get every shot composed perfectly how you want it. And you know, just keep rendering it until you're happy that any wonky pictures are sorted out. We wouldn't have that luxury normally. [TJ]*

*We could be more adventurous [when using the URS], you know, we could make the background more, put more into them, more detail, and think bigger. So, it took away the restraints of normally having to think 'well, it would be nice to do that but...'. [G]*

*We could render scenes out earlier when they're maybe not finished just to see what they look like and where we can do improvements. [G]*

They also said they were worried that rendering large amounts of stuff led to a render bottle-neck and having to sift through thousands of frames to check things. So it would appear that the little and often approach is not just a matter of speed but also a case of keeping the process more manageable.

#### 5.4.4.1 Using internal rendering services instead of the URS

Despite the power of the URS, there was still a trend to do very small amounts of rendering in-house rather than using the URS. This happened with:

*Small adjustments (pupil movement etc.)*

*Backgrounds*

*Small areas of particle based rendering such as smoke*

*Shots that wouldn't render on the URS for technical reasons*

Key factors that influenced this were the threshold of complexity necessary to submit a small job, and the complexity of getting the URS to do particle based rendering. Also the desire to render certain things with a renderer other than Maya:

*I actually did use Mental Ray to render the backgrounds, because of its global illumination and the way it bounces light around and I just found it much easier to light the set using that process. [DL]*

#### 5.4.5 Remote operation

One beneficial side-effect of the URS was that due to its net based nature it could be used wherever there was an internet connection. This fitted in well with the mobile nature of some of the subjects work practices:

*The fact that it's just there, you can just upload it and it'll start rendering, no matter where I am. I could do it from home or from the office. [SD]*

*Palo-Alto or on the moon, I don't care, if it comes down the wire and there's a support service then I'm happy. [AP]*

*We were up at Darwin at the facilities place, using their smoke machine, it became clear we needed to do a render, so we had our machine with us, so we just plugged it in up at Darwin, and then we could upload our file. So the guy at the facilities place was like 'that's brilliant, how do I get that?'. [K]*

*And if we can do that [work remotely in a rural location] with just a couple of machines across the smallest broadband connection available to any user with HD resolution which is huge, with fur which is the worst feature you can use on Maya- if we can do that now then I tell you, anything is possible. [MB]*

*You can just work anywhere. Set up an animation studio in Siberia or [unclear]. You know, it's irrelevant sort of suddenly. [TJ]*

*I was in the Hague for nearly six weeks and I carried on working, literally taking my laptop along to internet cafes and things, and that was surpassingly easy, actually, to do that. [JH]*

This coupled with the fact that you could bid when it wasn't busy meant that some people were monitoring and doing short term bidding from home:

*What I'd do is I'd just wait and bid at midnight, before bed. [TJ]*

Some even cited remote working as the top thing about SE3D:

*The fact that we could operate from the moon if we wanted to. [MB]*

#### 5.4.6 Group work

One subject found it a useful service for supporting group working [JH], and another subject found this was also the case as long as they were well organised and tidy in their use of files on the server side of the system.

*The render service was very good for us in particular as almost a sort of content management system in that we could both access it because the animators aren't based*

*in Bristol, they're based in London. [JH 2]*

*I thought it might be problematic, two sets of people using the service, I was using it from Bristol and they were using it from London. But actually in the end that ended up being a Godsend rather than a problem because we could actually all see what was going on at any one time. I think it might be quite a good use for this sort of system. [JH 4]*

*It actually meant going back and doing things the way that we'd done in Company X... lots of putting together and taking apart and putting back together again. [DL]*

However the added complication of this use meant that there was even more reason not to make use of the support for versioning:

*We would delete everything and just keep the version that was meant to be rendered. Just to make it... that was our own work flow, just to make it clear to each other. [TJ]*

A future URS system could even be a hub to support group work, either a group within a physical site or a distributed group. The remote rendering could be a 'behind the scenes' way of topping up local resources that the user is almost unaware of. The rendering service could be a piece of software that comes with free render time, just as mobiles come with a certain amount of free talk time per month.

## 5.5 Versioning

### 5.5.1 The Elephant Store

The Elephant Store [5] is a unique method of synchronising or copying data from one place (A) to another place (B). Each time a copy is made from A to B the system doesn't copy everything. It only copies what has changed since the last copy was done. Thus in the first copy everything is copied from A to B, but in subsequent copies only those things that have changed are copied. The Elephant Store works independently of file names and breaks open whole files to divide them into chunks so that it only has to copy the particular chunk of a file that has changed. This is a great advantage over systems that just look at the time stamp of files and update the whole file if anything has changed. With the Elephant Store, if a slight change is made before copying all the files from A to B then in reality only that little fragment that was changed is copied, the rest is the same as it was in the last copy. However, the optimal nature of the approach is balanced by the far more complex model of what is actually happening when compared to the brute force approach of copying everything from A to B every time that a copy is carried out.

### 5.5.2 Different concepts of 'version'

Here is a real world illustration of the 'typical' user model of a version. Imagine I give you a set of slides printed out on paper, then a few hours later I give you an updated set which I call 'version 2', Suddenly I realise that I have left one out, I get it and give it to you to add to version 2. The separate versions are clearly defined in our agreed model of what is going on despite the extra transaction to establish version two. With the SE3D project the Elephant Store would class the addition of that missing slide as the creation of another version, the incomplete set and then the complete set a few seconds later would be versions 2 and then 3.

This is exactly what happened when one of the subjects uploaded everything and missed a file out, they added that file, did another upload but this upload was automatically classified as a 'new version' by the elephant store:

*I haven't loaded up the image which should be in there for the file to be rendered. So I've got to stick the image in there and that creates a whole new version, just with one image just with in between that and [unclear – the other?] files and all the source images that have been uploaded, which is pretty bonkers. [AB]*

The URS treated every upload as being 'a new version'. Whereas for the subjects, the idea of 'a version' was a milestone in the development process chosen by them rather than the system. It could be that when it comes to versioning, the subject are less interested in a complete time based roll-back

•what was the version of this file on Thursday at 12:30?’ and more interested in a versioning system that works with snapshots at user defined points in time; •I’ve got the background sorted now, let’s make a snapshot now so that I can get back to it if anything starts to go wrong, I’ll call it •version with background OK’.

The danger is that you have here a situation where the versioning that is happening in the user’s mental model is different to that which is happening in the system and users will either become confused or they will have to spend time relearning their concept of a version to fit in with that of the system.

*That was something... I probably wasn't entirely sure for the reason for having that I must admit ... No, I mean, what was the thinking behind it? [DL]*

*I'm not sure why I would go back to a previous version. [SD]*

*That was a big, big thing for me- what is a version? Because it's a bit nannying as well, it's holding you by the hand. Whereas anybody who's worth their salt who's using Maya is going to be saving different [unclear – copies?] of scenes all the time, which is in essence your version. [AB]*

*I don't think that was very well explained, I don't think it was very well promoted. When we had one of the focus groups it turned out that I don't think anyone in the focus group had understood that's what was happening. And being paranoid animators they certainly didn't trust it. [AP]*

There also seemed to be some confusion about what files were involved in any particular version:

*I never knew exactly what we had uploaded and what we hadn't uploaded, I could never at any point sort of see exactly what was there and what the names of the files were. Because we had so many files all over the place that and I wouldn't know if someone had already rendered or not, if I was wasting time by rendering or not, so again in the theme of transparency it would be really nice just to be able to have a pull down list with all the files you had when you uploaded them, how big they are and all that sort of stuff. [FOCUS]*

*I don't remember if there was a way you could actually seeing what files are in the version, if you see what i mean? So you'd upload a version, and then think 'ok what's in that?'. [BL]*

It would appear that the use of the term ‘version’ in this context stems from the world of software development where concurrent project support tools allow users to upload ‘versions’ to a central repository, each upload is thus a version. This metaphor has then been migrated to the URS either explicitly or unintentionally.

### 5.5.3 Avoiding the ‘version’ issue

Another development is that subjects will modify their use of the system to side-step the whole ‘version’ issue, and that is exactly what the users did here. They added another step to the process and they adopted behaviours to negate the effects of the versioning, it wasn’t what they expected so they found ways around it. In the experiment nobody used the roll-back facilities offered with the versioning. All versioning was done on their local machine, with a separate directory to feed the upload (see 5.4.3) the contents of this ‘upload directory’ were then deleted on a regular basis.

*I sort of ignored that a bit actually, and in the end just stripped it down to, instead of saving that into a directory with the latest version, so it would upload the most recent, I just stripped everything down so that per scene file it contained only information for that particular scene. [MB]*

Probably the most telling comment comes from a distributed group who were using a simple text file to log and organise their activities. A huge research project produces a system that in the users mind is not as suited to the purpose as a simple ASCII text file.

*We just actually had a little text document that people used to put in, right put this up,*

*this is the reference, this is the name, these are the frames. [K]*

*It was more version, that was trying to control the versions as things went up again. We also had the shots stuck up on a wall and little red dots that go beside. [K]*

*We found that the best way to actually to manage the rendering system was to actually create a position for somebody who was responsible for uploading and downloading files, kind of acting as a gatekeeper so the animators didn't have to worry about it. They just put a copy on and somebody else could look after it and tell them how it was getting on [unclear] and if they signed it off. [FOCUS]*

The only advantage the subjects saw in the remote versions stored on the server was a form of back-up, though this was usually seen as a supplemental back-up to the back ups they already did locally.

*We were using it for something that it's not supposed to be, which is a back-up thing. [K]*

#### 5.5.4 Terms and technology

In analysis it would appear that there are two key terms underlying the operation of the Elephant Store, namely 'upload' and 'version'. For each of these terms what the user understands by the term is fundamentally different to what the system means by the term.

***Upload (users concept):** do a big transfer all my files to the remote file store. 'Are we ready to upload the scene with the tiger in it?'*

***Upload (systems concept):** do a small transfer of only what has changed since the last upload to the remote file store.*

***Version (users concept):** a snapshot of a set of files defining a project that have reached some key 'milestone' in that project. They embody some goal or concept in the mind of the person producing that collection. 'This is the version with the proper clouds in the background'.*

***Version (systems concept):** the new configuration of files on the server incorporating the last upload.*

In user terms a version is goal oriented, you have a definite version because you want to do something with it. What defines a version isn't the multiple uploads to get it there, it is the process that happens afterwards, the next step in the process. Consider software versioning; in the programmers model of the process different versions of a piece of software are indeed defined by every single commit that was done with the concurrent versioning system, but to the end user of that software a version is defined by the release number, the process of releasing that software for public use or evaluation.

#### 5.5.5 Upload

We have seen how the term 'version' has become confused due to the systems concept of it being different from the users' concept. Now what of 'upload'? Let us consider how the system was presented to the user. It was presented in two ways, firstly as an explanation of what was really going on, secondly it was presented in the actual interface with the complexity hidden. This means that subjects who didn't attend the talks or who didn't understand the extra complexity only had the interface to go on and the interface presented the system using the old model of 'upload'.

The end result of this situation was that because the versioning model offered by the system was not useful the subjects kept doing their own versions, and because the system used the term 'upload' when what it was doing was very different to a conventional upload, the subjects had the idea that when they had a directory full of different versions pointing an upload tool at that would result in doing a huge upload each time:

*But the change might only be a few kilobytes, a megabyte or something if it's done properly. I didn't understand that until that was explained to me, or anyone I was working with. [FOCUS]*

*When you're saving stuff as if revisions, you save it as a separate file in case you ever need to go back. And so what happens is you know, your scene file will be 'shot one version one', and where they end, you know, will be 'shot one version thirty-seven' and so if you're pointing it at that directory it's just going to want to upload all those new files each time. And so the way I work is, I have two databases [directories], my working one for me and the one for the URS. [FOCUS]*

The uncertain nature of the method of working meant that there was also an issue of trust:

*The animators I was working with did not trust it, they did not want to get involved with this, they wanted to manually upload everything. [FOCUS]*

*But they've got to trust it, if they don't trust it they'll try and get around it themselves and they'll f\*\*\* up. [FOCUS]*

### **5.5.6 Technology making terms obsolete**

When technological advances are made they usually involve the trivialisation of a previously non-trivial task. This means that the task can be hidden from the user.

Consider the term and concept of 'bootstrapping' a computer. This used to be a common task, with a name, there were bootstrap loaders. Today this all happens automatically and the only place where the term lives on is in terms like 'reboot'.

There is a parallel with the term 'upload' in this experiment. It has a definite meaning and concept because it is a non-trivial task, however the abilities of the Elephant Store turn what was a big task (uploading a huge set of files) into a trivial one (uploading only the minute differences).

### **5.5.7 Presenting the system to the user**

Often novel software will perform an action in an optimum way by doing something clever. This 'something clever' usually involves more complexity than if the action is done in a 'non-clever' manner. In such situations the usability of the action will depend upon the relationship between the user and the extra complexity.

In order to use the system efficiently the subject needed to know how it was working; they need to know that it is doing partial updates. They need to know about the clever complexities of the upload.

*I had to draw a very big diagram before they understood how it worked because there were three animators and they spent like two days arguing about how this would work and like we don't have two days to do that. I drew the diagram based on my instructions and then it worked but that was time wasted. [FOCUS]*

There are several ways of using this system and the correct way of using it has to be supported by making sure that the users really know what it going on behind the scenes, only when they have that understanding will they use the system in the most optimal way.

It would be better to have a system where the user didn't need to understand what was going on behind the scenes, a system that enforced only the optimal way of working and enforced it in such a way that the users were unaware of what was going on behind the scenes.

There is a lot of inertia in work practices, adopting a new work practice (usually embodied in a new system) involves the user understanding the new practice and being aware of the advantages of the new practice.

What is needed is a solution that is clever but that is presented in such a way that it is actually simpler to understand than the existing model. In this situation, what is a simpler model than an old fashioned, brute force upload? Well, consider hiding the upload completely. You work in a file space on your local computer, when you are ready to do a render you indicate the scene that you want to render by selecting a set of files and directories. This set is locked while it renders it and then a folder of frames appears in or next to it and the directory is unlocked.

Although uploading *does* happen the user is unaware of it and the model is a simpler one which hides

the complexity in a way that enforces the correct and optimum behaviour.

This is an idea for dealing with the upload part of the problem and not the version part, however that could be encompassed in some way as a back-up of the file system with some way that users could flag the current configuration as a 'version' that they may want to return to at a later date. This is purely an illustrative example to show that the area is a large area that needs exploring.

### **5.5.8 Involve user interface design right from the start of the project**

These observations underline the need for usability design being involved at all stages of the system development process. Not just designing the interface to the process but designing the underlying process itself and the terminology that the process is phrased in. The Elephant Store was a good example of this, none of the users had problems with the underlying model they simply ignored it and used it in an 'incorrect' manner. In order for it to be used in the optimum way the interface, presentation, terminology, and user model needs to be designed properly, if it just 'happens by accident' then any testing could be invalidated by people using it incorrectly.

## **5.6 Storage limits**

Only two of the subjects ever hit the storage limits. In one case this was due to the lack of control of ongoing multiple render processes. The physical limits for the data were beyond the limits at which the subjects had struggles with the complexity of the data (the confusion with 'versioning' described above.), so subjects tended to keep the data tidy and well organised by adopting a 'scorched earth' policy of server-side file management. When they went to upload a new set of files they would first delete everything on the server and then upload the new stuff.

*if [X] is setting up one of my shots, and she goes into the directory and there's a million versions, sometimes even though we'd take the latest version, sometimes it's not clear. So we would delete everything and just keep the version that was meant to be rendered. Just to make it. that was our own work flow, just to make it clear to each other. [TJ]*

*I didn't ever keep very much on the server. [DL]*

This approach appears to deactivate the efficiency of the Elephant Store which works by comparing the upload to existing versions in the store. However, behind the scenes the Elephant Store always maintains a copy of the last upload to be done and thus even if all the versions are deleted by the user there is still this 'hidden' version with which to compare the upload and the optimisations still apply of only transferring data that is different.

## **5.7 Comfort with the system**

### **5.7.1 Did it constrain them or liberate them**

Everybody saw the URS as a liberation, some (3 from 10) saw it as liberation with constraints and cited the textual interface and the fact that it used Maya as a rendering engine.

*We didn't have to worry about [the rendering] it means you can just concentrate on the juicier bits of the film making process. [TJ]*

*The whole issue of rendering ceased to become a problem, it became a non-issue. [T]*

*I just trust it completely to get my renders done. And that is a huge weight off my mind. [BL]*

However the liberation of having such incredible rendering power at their disposal sometimes served to accentuate any short comings in the interface:

*It was liberating to have 50 machines all purely for me rendering on a double processor. But at the same time it was quite constraining having to muck about with all this typing and uploading and stuff, which was a bit of a hassle [AB]*



Also there were many instances where subjects were avoiding using certain facilities because of simple problems some involving the interface between the render farm and the rendering software. As well as the active camera problems and rendering complex state dependent things like particle clouds and fur there were also problems with textures:

*We tried not to use textures for various reasons, one of them was that in the past we've known they can be a nightmare making sure that the renderer can find those textures, especially on a render farm. [SD]*

Although generalised and not URS specific it does illustrate the problem that the simple act of linking to a particular texture file has become a 'nightmare' in this new technical context.

### 5.7.2 Distance

No-one had any problems with the physical remoteness of the machines. There was a general idea that it was 'cool'. Having HP behind the system was once cited as a reassurance. Also the fact that things were remote was sometimes cited as a positive factor giving a sense of relief at having that aspect of it far away and dealt with by someone else.

*It felt like the machines were next door, or in the same room. [TJ]*

*They would fall over a lot of our machines. They would crash. [TJ talking about internal render farm]*

### 5.7.3 Security

Security was never really regarded as being an issue.

*If it's HP labs, it's going to be a pretty secure environment. As safe as going to Barclays (bank). [K]*

Hewlett Packard's association with DreamWorks was cited by several of the subjects as evidence of the dependability of their security.

*If they're doing stuff for DreamWorks they've got to get their security tight. [TJ]*

### 5.7.4 Expectations

For some it met expectations:

*Does what it says on the tin. [TJ]*

But for just over half, it was simpler and easier than expected.

*I'm a bit of a convert because I was quite sceptical going into it. [AB]*

*Very simple, only expectation was problems with remote element, but this was ok. [TJ]*

*I think it was a lot more, in terms of the user interface, it was infinitely more simple than I had anticipated, it was just really, not a problem. There was nothing to it. [K]*

One subject was worried about the whole 'remote rendering' aspect but found it to be no problem, but the main fear was complexity and only one subject found it more complex than expected and that was mainly due to the textual nature of the interface.

It is interesting to consider why more than half the subjects had such negative expectations. Could it be that the service was being introduced in a way that was more complex than it actually was to use. Once again this makes a case for not introducing the service by means of long explanations but by just letting the subjects get their hands-on to it in real life or with a simulation of some sort. A behaviour similar to the way you would get a child to use a system (see 5.3).

The speed exceeded expectation except where the subjects had no experience of rendering times. Although the first impressions were usually good, those involved in the planning realised that there were constraints, it was still necessary to exercise existing planning methods to make the best use of the resources.

### 5.7.5 Speed

Only three subjects had speed issues, one of which acknowledged that it was on their side of the connection. The others commented that there was slow downloads of frames when it was in the busy part of the project (close to the overall deadline).

It also enabled them to adopt more experimental work practices (see 5.4.4). However, the increase in rendering power was often being offset by the subjects trying to do new and more challenging operations. Such things as adding more detail/texture, rendering to higher-resolution image formats etc. As one of the subjects observed:

*Always takes 40 minutes to render a frame, you just pack more into it. [TJ]*

The subjects appreciated the power at their disposal:

*I don't know how we're going to go back to the real world now. [TJ]*

### 5.7.6 Bottle-necks

The majority of bottle-necks referred to were in the mechanics of uploading and downloading, in particular the need to specifying things in textual terms.

*Basically when I set off a render I just wanted it to go up there and bounce back to my hard disk and be there to use. I didn't want this whole explore images, giving it a name and all that basically. [FOCUS]*

Other bottle-necks were the downloading of frames, setting things up to render particles, deleting running rendering jobs and the slow 'delete version' function (see below).

All the bottle-necks were considered minor ones, although the textual interface one did keep coming up as a major frustration with the system (see 5.9.3).

Sometimes the distinction between bottle-necks at the server end and bottle-necks at the client end (on the local machine and the local internet connection) was not fully appreciated by the subjects.

#### 5.7.6.1 Deleting versions

Approaching storage quotas meant deleting versions of uploads that were still on the server and the fact that the delete couldn't be done in a batch manner caused bottle-necks, you had to baby-sit the whole deleting process. Here one of the subjects talks about the lack of a batch delete, although he uses the term 'jobs' he is talking about versions of files:

*That was a pain in the neck, deleting stuff because you had to wait to delete each individual job, you couldn't just assign the whole lot and shift click on them, I don't think you could shift click on one of the windows, you had to select each one individually anyway and it would actually take a matter of minutes and sometimes quite a few minutes to perhaps delete each job. So if you had to clear off a lot of space you had to click each one and wait five or six minutes, click the next one, five or six minutes... I had 400 jobs on there at times. [AB]*

#### 5.7.6.2 Uploading and downloading

When asked if there were problems or issues with the uploading of scenes or the downloading of finished frames two subjects cited problems with downloading – specifically one said that it sometimes missed frames out. Three cited problems with uploading including the lack of feedback while uploading which did lead one subject to ask for better feedback as they were never too sure if it had all crashed or not.

*At times when you were uploading a new job, it would take, it seemed to take a lifetime sometimes. And you weren't sure if the thing had crashed or not. After a couple of times we realised it's not crashing it's just doing it's job. [K]*

## 5.8 Usability (of the URS)

Subjects were accepting of the way that the URS worked. mainly because using it was a way to get the power they needed.

*It wasn't a pain to use. [SD]*

*Nobody's gone through with a user and sat down and said 'ok, it all works now how do we make it funky or cool, or work in a way that's easy? [BL]*

The subjects were quite definite about usability areas of which they were aware, and there were usability problems that they were not explicitly aware of but that were evidenced by their behaviour, for example the double directories for uploading described in 5.5.3.

### 5.8.1 Ease of the initial set up and installation

Subjects agreed unanimously that the URS was easy to set up and install. Sometimes to the surprise of the subjects, from their introduction to the service they were expecting something more complex. The comments for the question on setting up are borne out by the fact that subjects generally tried to get to grips with the service without initially using the manual – the commonest approach was to refer to the manual only when they were getting stuck.

## 5.9 Features

A key aspect of the features in design terms is making sure that they are tangible in the interface. During the interviews we had the situation where one subject when asked about features they would have liked to see included asked for a frame preview to check that renders were going okay, while another subject when asked about favourite features replied that the included frame preview was really useful.

Another example is that if the renderer hit problems with render jobs it would move on to the next one and return to have another go at the problem ones when it had got through all the jobs it could. Although this was an advantageous behaviour it was not as useful as it could be simply because the subjects didn't realise that it did this:

*I know people that said 'oh it's not rendering packets' I'd say 'Well did you just try leaving it?' And they all said 'No!'. I was like, 'Right, next time that happens, leave it and it'll try them again at the end'. So yeah, once I kind of figured out that that was how it worked, it was fine. [DL]*

### 5.9.1 What features are missing?

A breakdown of the responses to this question gives the following results some of which are dealt with in more detail below:

- Ability to quickly preview frames<sup>1</sup> being rendered 2 from 10 subjects.
- Rendering job status notification by email 2 subjects.
- Better naming of files (browsable interface see 5.9.3) 5 subjects.
- Ability to interact with rendering processes (setting priority and adjusting processor allocations) 2 subjects.
- Ability to overcome the shortfalls of Maya, and the version of Maya used 4 subject.

### 5.9.2 Previewing frames

One subject wanted more insight into how the rendering was going, when they were running it locally

---

<sup>1</sup> This capability was actually provided.

they could say 'show last render' but on the URS they had to download the frames first.

*It would just start up and I just want to go 'I just want to have a look at that frame, make sure it's....'. [DL]*

There is a need for a good preview operation that is readily apparent. Possible ideas would be, you set up a scene and then hit preview, it renders ten frames spread out along the length of the whole scene. Or you click on a simple slider and it renders just one frame from the approximate place in the sequence that you have chosen.

### 5.9.3 Browsable interface

One of the most common criticisms of the interface was that it should have been browsable, with the ability to select files for upload by point-and-click rather than by retyping filenames:

*Would be nice to have browsable interface. [TJ]*

*Because there's no browser you have to cut and paste the filename in. [BL]*

The form-like interface that required users to type in file names (and directories) meant that large amounts of rendering commands needed large amounts of repetitive typing. Many users resorted to cutting and pasting file names in rather than typing them, but even this workaround still left the user with a large amount of donkey work.

*It seems relatively ridiculous that for each job ... I've got to type 'scenes' each time. [AB]*

This was more than they were used to with in-house resources:

*Much more. 6 or 7 times more by the time you've typed in all your things to upload and then typed in all your things to download to. [AB]*

Subjects were acutely aware that the system was not using defaults or reusing names already input by the subject:

*And then it asks you what you want them to be called, and you think 'For God's sake, I've told you already what I need them to be called so why do i have to tell you again?'. [BL]*

*If you enter into other rendering utilities they'll just create the image with the default name of the file. And if you upload something called 'shock dock and bee' then it'll create the images called 'shock dock and bee'. Does that make sense? I think that's all it needs, some default settings you can perhaps just go in and alter if you do want to give a fancy grand name or something. [AB]*

Although a seemingly small issue the repetitive nature of the problem can lead to a huge build up of frustration and perceived time wasting.

*Then you have to put the name in, which takes two seconds but when you're downloading twenty-five separate scenes and each time you have to type in the name and remember which one you're downloading, it can get a bit of a painstaking course. I mean the best thing would be just to default to whatever name you put in. [SD]*

Once again there is the emphasis on visual and tangible interfaces identified earlier (see 5.3.1) as evidenced by this comment:

*Just visually again, if you'd been able to see it as sort of an extension of a windows interface. [FOCUS]*

### 5.9.4 Adjusting processor allocations

Two subjects commented that it would be useful to adjust the number of processors allocated to jobs as the jobs were running:

*That would let you then say 'well this job was supposed to take an hour to render but it's been on there 2 hours and it's only done 25%, let's bring more to bare on it'. [AP]*

### 5.9.5 Adjusting priorities of rendering jobs

There was no way that the user could manipulate the processing order of rendering jobs on the server. The system could be busy rendering and downloading frames for a number of jobs simultaneously, spreading itself between them all and thus making them all progress slowly. It would make no effort to complete one before the others meaning that the subject was left waiting for one of them, any one of them, to finish.

*I get this stack of ten of them partially downloaded none of which are any use to me ... you can't prioritise, no control whatsoever and that was the major, major... problem for me. [BL]*

Occasionally subjects would actually find ways around this in order to speed a job through the system, they would simply delete those jobs that were ahead of it in the queue and resubmit them later. This is a classic case of getting insight into the tasks that the user wants to perform by observing how they use the system.

### 5.9.6 Estimated completion

The estimated completion feature was inaccurate and inconsistent so some subjects ignored it completely, although they desperately needed some way of knowing when something was finished:

*I actually disregarded that completely. [TJ]*

*Maybe a little alarm clock that could go off when it's ready, to let you know your render is done, please come collect it.. [K]*

*What would have been interesting would have been to get an idea of how long, mind you that would be URS again, kind of like, anticipated delivery time, things like that. [K]*

This was a common complaint, about any URS process involving a degree of waiting:

*If you were in a rush the temptation is just to stare at this bar willing it on, and you could be there for forty minutes, so if there was a feature, that transparency issue, if it said 'OK, sorry your machine is c\*\*\*, this is going to take a long time, why don't you go and do something else?' If it gave you that information about what time it was going to take then you wouldn't be wasting. [FOCUS]*

### 5.9.7 Editing parameters

Another feature mentioned was the ability to alter render parameters without having to upload everything all over again. Often the subjects would realise that a render job was fine except for some parameter, it would be nice to reach in remotely and change this parameter without the overheads of deleting and reloading.

### 5.9.8 What features were good?

There were no features that stood out – the main good aspect was the fact that it did what it was meant to do, and quickly. The partial uploading of the Elephant store was cited by two subjects but it is difficult to judge whether all the subjects were in fact aware of the feature. Ease of use and looking at frames as they came out were also cited.

## 6 Bidding in general

### 6.1 GV or share, which auction did subjects prefer?

Occasionally those subjects that didn't have GV were critical of the unpredictability of the share market. This led one to actually ask for a GV type auction.

*One thing would have been really good, I don't know if you can do it, is rather than the amount you want to bid, is how many machines you want. If you see what I mean? [BL using share]*

However, there was a lot more feedback from those using GV about being mystified by what was going on:

*I was foxed as the actual process behind the scenes with the first bidding system [GV], I just didn't have a bloody clue what that was all about. [AB]*

*I was never completely shocked by what happened, but it wasn't always exactly what I predicted. [JH had GV]*

*eBay felt very familiar, the bidding environment, from go, from the minute I've used it but I can honesty say that still, to the very very end the URS always had surprises for us in outcomes of auctions so there was a feeling of 'oh why did that happen' to the very end I suppose. [MB had GV]*

Animators seem to be quite opinionated and have clear ideas about what they want, any commercial system can benefit from an analysis of the way that animators think that the economics should work, and as described in the next section, more insight is needed into the bidding and outcomes so that people 'get a feel' for the bidding behaviour without having to understand the math.

#### 6.1.1 Insight into bidding behaviour

The most interesting comment was that made by someone with extensive experience of eBay (a user since 2000). They said it was always useful to look at the outcomes of bidding sessions afterwards to analyse what happened; similar to the after match analysis of a game of football. They would have liked to have seen something similar in the interface to the rendering service. When the auction closed and you were told how much you had one it would have been useful to see who did win, how much they had bid, how many units they had asked for and how the system had shared out the pool of units.

*With eBay you can sort of monitor your own progress, quite simply by best studying the bid outcomes at the end and see what happened in an auction, and who came in last minute and, perhaps not how much they bid but what it took to secure the bidding. And there wasn't that sort of feedback on the URS where you were able to see at the end who got that auction and how much for and at what point they bid. [MB]*

Even the extra field in GV of number of units was seen to help with this:

*It was nice sort of having some feedback of how many you could go for with that bid amount, because it felt like an extra tool to lay your hands-on render boxes but I don't know how much difference it actually made. [MB]*

Another factor in support of this argument is that after the experiment several of the animators expressed an interest in seeing the statistical results of who bid how much for what in order to see how their behaviour fitted in with the big picture.

### 6.2 Long term bidding and short term bidding, which auction did subjects prefer?

Lack of long term planning and a sketchy understanding of long term bidding meant that it was not used as extensively as short term bidding, even though it offered advantages.

This points to understanding as a key tool in enabling the market mechanisms. The subjects need to know the key facts of the different ways of bidding. They need to know that there is always a proportion of resources available for long term bidding, they need to know that once they have won them they are 'safe', and as price was one of the key motivations for those that did use long term bidding, subjects need to know that long term bidding can often be cheaper than short term bidding and that if they overbuy they can sell back and still come out better off than if they had used short term bidding.

Uptake of the long term bidding could be increased using several methods including the following three suggestions:

*The mechanism could be better explained, rather than explaining how it works it would be better to explain what the effects of it are. Once again the fact that they are animators means that this can best be communicated to them as a graphic as part of the interface, a diagram which they see daily in their use of the short term bidding which informs them about that while at the same time making them realise the amount of resources that are available to them in the long term bidding.*

*More resources could be made available for long term bidding, the more are available in long term bidding, the more likely the subjects are to go to long term bidding for their resources. This will only work if they are made aware of this.*

*Finally, subjects will only turn to long term bidding if their needs are not being met with short term bidding alone. In particular they need to be made aware of the commercial advantages of using long term bidding; for example by cross selling, in short term bidding they need to be told: 'these resources cost you this much, you could have got them for this much if you had used long term bidding – click for more information'.*

### 6.2.1 Guaranteed Markets

Almost all the subjects wanted a guaranteed base line amount of resources and many suggested ways in which this could be organised. Few made the direct connection that LTB could provide this.

To take a parallel from real-world economics, the 2005 drought in Spain has had a large impact on the olive growers and the government are trying to introduce some form of predicative pricing mechanism so that the farmers know in advance what the returns for their olives will be and can plan accordingly. This same problem occurs in the purchase of resources in the SE3D environment. Subjects did not know how much they would end up paying and so there were concerns about budgeting commercial projects:

*I think it would be nice to have that guarantee that you will be able to have at least so many that you know you can always render. [TJ]*

*When you've got a set budget you don't want things to spiral out of control. [TJ]*

*I'm quite jittery about picking up someone's crumbs of rendering in order to finish something which is a massively important thing to me, you know commercially, but small fry on the big scale of the rendering system. [TJ]*

However, the final observation in this area is that several of the subjects offered to pay to have their access to the service continued after the experiment.

### 6.3 Pricing and predictability

A number of aspects of the system are geared up for dependability; there is good security, there is a pre-agreed limited maintenance downtime, there is consistent high power, there is frame previewing etc. This gives the service a high degree of predictability:

*I think it took a lot of the stress away in terms of, you know, rendering is sometimes an unknown entity. [TJ]*

The paradox is that the whole bidding process re-introduces unpredictability into the equation:

*I think it's just that unpredictability, and I think animation is, you know, the least of all unpredictable things, everything is tied down to a very finite thing, down to the frame of how long this shot's going to be, down to the frame, how the lip-sync goes, it's all down to the most minute level. Because so it's so time-consuming you're not wasting your time. And I think to suddenly throw in a random element, I think a lot of animators would freak out a bit. [DL]*

*to come in under budget, that would be fantastic but equally, the flip side of that is you could spend more than that. [TJ]*

*A couple of grand here or there could make or break us really. [TJ]*

*I wouldn't like it at all commercially. Because to me, you've got a budget, and it's not my money. You go to a broadcaster and the broadcaster says 'ok, this is how much we're going to pay for your service'. If you go over that you've got to spend it and it's like, well, I don't really want to necessarily be worrying about how much the rendering is going to cost. It's all about reducing risk in production, for the whole producers job ... Why would you want a system where you are opening yourself up to a massive amount of risk? [BL]*

*It needs to be something that's on our side and not a game, it's not a flipping lottery. [G]*

*We need definite things. Like if you're booking a dubbing suite or something, they don't say 'well maybe, if you give us enough money you might have it', they go 'ok, when do you want it, Tuesday? Fine ok, a pound an hour, thank you very much, deal done'. And you know where you stand. Whereas if they said, 'maybe, we'll see who else wants the time' then you'd go 'ok get stuffed, I'm going some where else'. [G]*

*One of my freelance animators is using a web server like that for one of his own projects, commercial projects, I can't remember what it's called but I'm sure there are lots of them out there, and he's really into it, he absolutely loves it. He says 'well I'm going to need this many hours to render so that's what I've got to buy, job done'. [FOCUS]*

Removing risk was seen as one of the factors in favour of the remote rendering farm over having an in-house render farm:

*When you're doing a production, you can't deal with maybes or things like that- if you've got a client and they want an advert at 9 am tomorrow morning and your render farm's a bit iffy. [G]*

Even long term bidding wasn't seen as a way around this problem of predictability as discussed elsewhere in this report.

## 6.4 Commercial use

When asked if they would be interested in it for commercial use almost all subjects answered yes. Some concerns were raised about the price of the service but this seemed to be the only factor that would effect its uptake. 7 from 10 subjects subject would make the choice by comparing costs to buying and running their own render farm. Many of the subjects seemed to have very definite ideas about the costs involved in setting up and running a render farm.

*If you average out trying to buy a new computer every couple of years, or ideally almost every year. Or renting in a rend box, a series of boxes. [TJ]*

*What's the cost to me of having to buy x number of dells, install them, support them provide office space cooling electricity for them and someone to look after them? [AP]*

When asked about whether the URS would be instead of or in addition to in-house rendering, most of them saw the URS as the major tool supplemented by little bits of rendering in-house (6 out of 7 subjects who made a response). One subject was interested in the service because it opened the door to rendering by part-time animators.



### 6.4.1 Real competition

The whole experiment was one in resource optimisation. But storage levels were high on the server and many subjects work patterns meant that these were never reached. The amount of credits that the subjects got was always ample for their workflow, any 'prudence' exhibited by the subjects was inherent and not based on the cash supply. Finally the power and number of processors was very high, far higher than anything the subjects were used to, this coupled with the relatively low numbers of participants taking part meant that there were rarely times when subjects were 'up against the wall' with resources.

*And because we had so much resources, we never felt any pressure of having... we knew it would get done at some stage, all you need is 7 or 10 free and it's gonna go through fine. [TJ]*

The division of resources into fixed pools for short-term and long-term bidding meant that there would always be resources available for subjects who were choosing to ignore the long-term bidding. It would be worth putting time into designing a more 'cut throat' experiment and finding ways of balancing this up against the fact that one of the attractions for participating is the huge amount of resources that the service makes available.

### 6.4.2 Pricing

The big challenge then is finding a pricing model that works commercially but that also works in terms of user comprehension and appreciation.

Animators currently have to factor in the cost of the rendering into their budgets. They don't factor in background/ongoing costs such as lighting, internet services. These are just part and parcel of the whole cost of keeping their operation going. Is it possible to run a rendering service in this way? Currently we have a 'half-way house'; in order for the market to work the users of the service have to be in control of how much of the resource they want. It is not a 'set-up and forget' service like electricity. With electricity you pay for a certain level of service and it is up to the service providers to ensure that the service can meet the peaks and troughs in demand.

The economics of the environment is an area that could benefit from further investigation. What service models are there that could function as metaphors for this new service? How could animators best be encouraged to view the new service? Is there some other sort of experiment that could be performed that is less structured and requires less overheads than an animation film festival?

The animators seemed more than willing to discuss possible ideas and there were references to different markets that could be employed to support the subjects buying the resources they needed. Examples that came up in the interviews included:

- EasyJet purchasing of plane tickets.
- Mobile phone free minutes.
- Rate capping on mortgages.
- Stock options.
- Hotel room reservations.
- Booking a dubbing studio.
- Electricity.
- A French render farm that gives resources away for free in the gaps between planning.

The underlying opinion was that although the unpredictable nature of the current system was not acceptable in a commercial context, there must be some formula that could work.

*But for it to work it's got to be something that will help us, not become a potential obstacle, it's got to be something friendly to us, it's got to be something we want to use, rather than go 'Nah'. [FOCUS]*

The issue of pricing was one of the main issues when the subjects were asked about their concerns with the system becoming widespread in the future; 6 out of 9 were worried about the economics of it; how to set the price and how to avoid the small operators being priced out of the market:

*More small time people like us would be priced out of the market. [TJ]*

*Because if its to be intended for the home animators and people working on their own how they would stop large companies coming in with a massive wad of cash and bidding these resources. [AB]*

*And so then a service that is set up for use by lots of people may in fact only ever get used by one group that have a lot of funds to spare. [DL]*

## 7 Short term bidding

### 7.1 Usability

Usability can be divided up into two level, firstly the surface usability, this deals with the layer of the system closest to the subjects; could users understand the controls and information being presented to them at the interface?

The next level was how well the subjects could understand and get to grips with what was going on behind the scenes, understanding the deeper parts of the system (or how well they could accept it and use it without understanding it).

#### 7.1.1 Surface usability

All the subjects found the bidding system easy to set up initially, and easy to use once it had been set up:

*It just becomes automatic so quickly, I'll just fool around and use the help if I get stuck. [TJ]*

*Very strangely familiar... it seemed to borrow familiar paradigms. [JH]*

However the interface was not without issues. A few users needed to be informed they had placed large recurring bids, probably in error as they had thought they were just making a single bid.

#### 7.1.2 Behind the scenes

No one really understood what was going on behind the scenes, but the consensus was that once you were using it you got a better grip on what was going on. One subject said that the lack of understanding would be frustrating if you were dealing with real money.

*My understanding of it was you just use it, just see how much you could get. [TJ]*

Although they didn't understand it, once they got to grips with it it was possible to get used to it.

*I just looked at my bidding advice thing, punched in the number and just saw how many units I'd roughly get ... generally I knew if I put down a wad of cash, pressed my little buttons to tell me how many units I'd get, I'd get roughly that amount of units. [AB]*

Only one subject felt that as the underlying logic was explained to them it was necessary that they did understand it and he was concerned that he didn't

*It wasn't that I didn't understand how to make a bid, but I never completely got my head round actually how the system made decisions beyond that point. It was explained to us at great length in one of the workshops, and I felt terribly stupid because I realised I didn't completely understand it. But whether that really mattered I don't know [...] maybe it's because they tried to explain it to us that I felt slightly inadequate at not knowing this. [JH]*

### 7.2 Bidding and managing finances

Everybody (8 from 8) adopted some form of strategy in the bidding. These ranged from establishing limits on the amount bid through to bidding patterns and general attitudes to bidding.

If it was expensive then subjects usually just tried again at less common times, e.g. last thing at night. The internet-based nature of the URS meant that subjects with home connections could bid at all hours of the evening and night.

At least two subjects reported that they carried out eBay style sniping tactics when bidding, waiting until the last moment to place a bid so that other bidders would have little chance to react.

There was no clear answer to the question of a feeling of the cost/benefit of the bidding system, this is possibly due to the lack of extreme conflict in the experiment, several subjects commented on this and they all found that they could easily get the resources when they needed them. Only one from nine subjects said they couldn't due to a lack of funds.

One aspect of the system that did have an impact on their bidding behaviour was the one hour time-slot, 5 from 7 subjects said that it was annoying missing the window, only 1 seemed to accept the situation saying that it was just tough.

*If you just missed the hour, you had to wait a whole hour even if there were machines available and I thought 'I don't mind spending the money'. Start rendering, why wait til the hour comes round, it should be active all the way. [BL]*

## 7.2.1 Real money

When talking about different aspects of their behaviour subjects often said that it would be different 'if it were real money'.

*I didn't really mind too much because I managed to get what I wanted but it was imaginary money, toy money, if it was real money I'd find that a bit scary. [JH]*

This came up in so many places that future experiments of this type ought to have some real-world value attached to the tokenisation of the money to encourage more 'real world' behaviour.

One area where the subjects did behave as though it were real money was the caution with which they did the bidding; they often wanted to make sure that they still had money left to bid should they need to do a lot of rendering later.

*Even though it was virtual money I always wanted to have some left to bank at the end, it's strange that. [SD]*

*There was this fear of being left without anything at a very crucial point. [JH]*

## 7.3 Features

### 7.3.1 Higher level tools

The two main features that have not been mentioned up to now were higher level tools and bid advice. Higher level tools is the idea of having some computer based agent mediating between the subject and the system so that subjects can phrase their wishes in higher level terms which the agent then translates into low level bidding patterns, something akin but more complex than the proxy bidding embodied in eBay. Users would be expected to specify things like; 'This job has to be ready by Thursday and I have 300 credits to play with'. When asked about this there was general approval although two subjects were worried that they would lose the hands-on, baby-sitting approach which seems to be such a common theme among animators.

Subjects did have fairly clear ideas about what sort of commands they wanted to be able to issue:

*Have a calendar on screen and just click select the next seven days and 'I want fifteen processors for that amount of time, how much is that gonna cost' sort of thing and maybe then the system would work it out and do it for you. [SD]*

### 7.3.2 Bid advice

The bid advice function was regarded as being very useful in the confusion about the mechanics of bidding.

*I'd say it was the most useful part of the whole system. [TJ]*

*Bid advice saved my life, I wouldn't have been able to do it without bid advice. [AB]*

Once again this ties in with the idea of predictability and certainty that the subjects have expressed. One subject did comment that it was sometimes unreliable though.

### **7.3.3 Desired features**

When asked for their thoughts on how they would like it work the subjects were on the whole happy with the system how it was. Others cited several things that could be changed, most of which have been dealt with elsewhere in this report:

- A simpler interface.
- Better integration of the URS and the bidding system.
- A different financial model.
- Estimated job completion times.
- More information on auction outcomes.

## 8 Long term bidding

### 8.1 Level of use

Having become users of the short term bidding there seemed to be a general reluctance to get to grips with the long term bidding. From 7 subjects 4 used it and 3 were confused by it but still used it. Subjects seemed adverse to long-term bidding for two main reasons. Firstly; it didn't fit in with their planning method (or lack of it) they preferred to work in an ad-hoc 'hand-to-mouth' manner.

*That became a real headache because you're having to use stuff en masse and suddenly you've got a render bottle-neck, and then you're having to sift through hundreds of thousands of frames to check, and it suddenly becomes difficult to manage very quickly. Whereas the little and often approach is just something that works for us. [TJ]*

*The thing is, I don't know what I'm going to be doing next week, what scenes are going to be ready. [TJ]*

Secondly; there was some confusion about the mechanics of the long term bidding.

*Interviewer: if you've got resources left over you can sell them back.*

*Subject: Oh! Really? [TJ]*

Some subjects even thought that once they had won the long term bid their allocation could still be used by other users. Elsewhere in the report we saw how subjects complained about the lack of guarantees on price and resources and how they did not seem to equate this with the returns offered by the long-term bidding system.

#### 8.1.1 Promoting long term bidding

The bidding delivers pecuniary punishments to those who do not plan ahead, but it should also offer support to encourage them to plan ahead, it should offer a carrot as well as a stick to encourage a particular behaviour.

We have seen that use of the long term bidding was hampered by the complexity and overheads of carrying out long term planning. Very often the subjects were aware that they could accrue benefits by doing long term bidding but they still had difficulty crossing the threshold to carrying out long term planning. The service should offer tools to support the subjects with long term planning. These could either be designed for animators or it may be possible to design 'vanilla' tools that could be used by any users making use of a market oriented system.

The system should also promote long term bidding when the subject is doing short term bidding, as discussed earlier.

### 8.2 Influence on work patterns

Subjects were asked about the markets influence on the way they worked. Some subjects reported no changes with one saying that the use of the markets was governed by their planning and short term outlook. Two subjects reported that it made them plan things more, with one of them acknowledging that this was probably a good thing.

The goal of the SE3D project is to make users shift tasks around to optimise the use of the resources. A deep part of this process is forcing them to plan things in the long term more efficiently. Once again they need to be made aware of the benefits of long term planning and ideally the service itself should offer them more tools to do long term planning of animation.

The underlying platform (Service Utility) is for the delivery of On Demand services and as such it is not just geared up for animators but for all users of on demand computing resources this raises the question of designing a universal planning tool that assists users in planning their needs of resources and in so doing helps optimise the use of those resources. An important factor to bear in mind here is that many of the subjects commented that their behaviour would be different 'if it were real money',

see the earlier treatment of this.

### 8.3 Usability and understanding of LTB

As well as the issues already mentioned there were some problems with all the different times being shown, these came to a head with long term bidding:

*I mean just all the information you've got could be stuffed into that list [the first tab], perhaps a certain one for short and long term market, that's all it needs. It just seems over cooked. [AB]*

Occasional issues were reported with the response time of the market mechanisms but nothing major.

When subjects buy resources they have the opportunity to sell them back into the market place if they do not expect to use them. They were asked if they made use of this facility and there was a range of responses. Some subjects didn't know it existed, others knew but didn't use it, or they knew and would have used it if it were real money, or they used it and thought it was great. There was no overall consensus except that when they did need it it was useful.

When subjects won the resources and used them there was some excess acquisition of resources with some subjects feeling that they were not using the resources efficiently. This can be put down to the lack of rigorous planning methods. One subject put the 'cavalier' attitude to acquiring resources down to the fact that it wasn't real money (see 7.2.1).

*What I found with the long term market more than the short term was I booked in stuff just in case I needed it. [AB]*

One problem with the selling back feature of the system was that often subjects didn't use them because they didn't have access to the system and as such they were not in a position to sell them back to the market. This is a problem of buying resources for future use without it being supported by good planning. It is using LTB to acquire resources while still using them in a short term manner.

*A lot of the time that I was buying them and wasting them I wasn't there to sell them back to the market. [AB]*

## 9 Support

The support was received positively but there was a feeling that although it was fast and enthusiastic it could have been better if it had been better structured. As has been pointed out earlier some subjects failed to attend the organised events such as workshops because they were too busy animating, this seems to be a common trait.

*We got the idea of sitting down and... when you really should be animating. [TJ]*

When asked about the response times for support calls, most people used the phone and they all reported that response times were very good.

### 9.1 Community

The Bristol animation community seems very tight knit and supportive:

*There's a good support network that we can get on the phone and ask someone's advice, get some pointers. People look after each other. [TJ]*

The SE3D community was able to base itself on this and intensify it with the facilities it offered.

*Workshops. you've got to show it to people to get that feedback or it's like it doesn't exist really. [TJ]*

The subject group valued the community aspect of animators in Bristol and beyond. Also animators have very specialised needs and ways of looking at things. This community could effectively be leveraged by setting up some form of peer-to-peer training if there is to be a SE3D-2 project, the best people to explain everything to the new animators are the animators who took part in this round.

### 9.2 Online forum

People thought the forum was a useful resource and they liked the idea, but they felt that in practice it wasn't as good as it could have been (slower than other channels), this set up a vicious circle with subjects using it less and less. The phone support (and technical home-visits) were very well received. The drawback to this is that if personal, high-level support is only a phone call away, no one is going to bother with less direct forms of support such as the on-line forum.

*In terms of online help, I didn't use that. [TJ]*

A forum is a useful resource for the community but it only really 'takes off' if the individual users perceive a benefit to using that channel of support above others.

*It was really nice to look online and see that you're part of a little community making things, we're all making things and having struggles when you were having them. Technically we didn't really use it but for a bit of solidarity it was good. [TJ]*

The system needs tools, designs and policies to bring all communication into the system and to ensure that there is more use made of the interactive aspects of the system. The forum based support should be as fast as possible and phone support should only be offered after an enquiry has gone through the forum this ensures that the forum takes off and becomes a community repository of information that other users can consult. It also becomes a useful survey of issues and problems that the users have encountered. By capturing email in this way you can also ensure that all emails get answered within a set time limit. Contrast this with phone support where there is no record of the problems or the solutions.

Using forum based support over phone based support was the initial premise of the experiment but it is easy for such agreements to become relaxed, especially where subjects need help quickly. Safeguards need to be designed into the process right from the start in a very rigorous manner to avoid inappropriate, casual systems developing.

One subject commented that there were probably a lot more people reading it than contributing to it ('lurkers') and it would be good to be aware of them as well, for example showing people that were



logged in but not participating in the chats. Another way to encourage use would be to integrate elements of the forum into commonly used parts of the system interface itself, so that subjects were always aware of what was happening in the forum (just as MS Messenger (TM) integrates email alerts into your desktop environment).

Other approaches mentioned include adopting more forceful use practices, for example making all users submit updates at regular intervals. Although this may seem overbearing client facing animators already do a lot of presentations of this sort, and animation itself relies upon evaluating intermediate versions of the film as it is being produced.

Update submission was initially included in the experiment, but one of the traits of animators is that they are single-minded about getting on with the job in hand and the regular submissions failed to take place. Some way of enforcing this behaviour is needed in future, for example presenting the supply of credits as a payment for 'desired behaviour' including regular submission of updates.

### 9.3 Mentoring

Although some didn't attend the mentoring sessions those that did found it invaluable, they were very well received.

*The script editing was invaluable. [TJ]*

*That was the best bit of it. The workshops were great. [AB]*

Even for those working within larger organisations, the SE3D mentoring was seen as superior to ordinary support within the organisation, partly because of its proactive nature:

*Within the creative people you're working in there is support, you advise each other, but there's not a dedicated, 'you need help with this'. You're working to deadlines, everyone's busy. [TJ]*

*We would have made a different film, I'm not sure if such a successful film if we hadn't. [TJ]*

The only criticism was that it could have been closer to the beginning of the project in order to integrate what they learnt into what they were doing. One Subject got his work done very quickly at the beginning of the project and as a result he found that the mentored sessions fitted in well with his agenda. Again it is a question of forcing planning and an agenda onto the participants.

### 9.4 Training

A large part of the training sessions concentrated on explaining the mechanism underlying the bidding models. Such an explanation was unnecessary. The subjects were more interested in the actual ramifications of the different models and these could best be appreciated by hands-on use of the systems.

Such complex explanations also led to feelings of failure at not being able to understand the explanations, a sense that they must be important as they were being offered and the subject was disadvantaged if they couldn't follow the explanation (see 7.1.2).

### 9.5 Downtime

No one really had any problems with the scheduled downtime, it was an annoyance but the transparency made it acceptable and possible to plan around it:

*It's necessary, you realise that... [TJ]*

In fact, the regular maintenance sometimes made them think about what was involved in running a render farm and appreciate the fact that someone else was doing it!

# 10 The organisations

## 10.1 Hewlett Packard

There was very positive feedback about HP as a partner and no problems with the 'big company/small company' communication.

*Excellent. We wish they could be based in Wales. [K]*

*We were doing this, traditional animation, CGI traditional animation, for film and television, which has a completely different feel to the industry than the games, and the games industry we spent ten years in, after our initial experience in the TV and film industry, was completely different to TV and film industry, and the big thing is, in TV and film, you definitely feel as though you're working with fellow creators I suppose, whereas in the games industry, the artists and programmer relationship does, it has a different feel to it. And this did bring a new feel which was like a hybrid games, TV and film industry feel. [MB]*

*I thought there might be some sort of corporate shenanigans but they've been hands off. [AB]*

*HP were techier than the techiest techies we have ever dealt with. [MB]*

## 10.2 Watershed and the project intermediary

The Watershed media centre was generally perceived as a local, accessible focus for the community feel and instrumental in setting things up.

*It was great to have somewhere local. [TJ]*

*If there wasn't the Watershed there things like this wouldn't happen. [AB]*

*Under their wing as it were, to feel part of the arts centre. [TJ]*

The distributed and diverse nature of Watershed's role occasionally led to ambiguity as to its role in the whole set up. Strangely enough the two who were confused were based in Bristol.

*They just seemed to provide space for us to meet in... it wasn't obvious what their specific involvement was, I don't know how much of it they were driving or pushing or just participating in. [DL]*

The role was amorphous, some people got it and others didn't, Watershed should look at branding/image issues; they do a good job but they need to make people aware of what a good job they are doing and more importantly that it is Watershed that is doing it.

Everybody was positive about the role of the Watershed's intermediary, in the whole venture and the film festivals also came in for much praise:

*Animated encounters was fabulously helpful. Um, to produce a film and to have it at it's own special screening, you know, in a festival, was brilliant and it was exposure the likes of which I don't think any of us could have mustered on our own. [DL]*

## 10.3 SE3D

SE3D too came in for praise:

*It's all been a case of 'let's all get on with it and make some films'. [DL]*

Although there was a perception that it could have been better, in particular the online community was lacking (see 9.2). It was seen as a good focus for the outside world in terms of publicity and branding.

*What was nice about SE3D was it put us in a position where we felt part of a much much bigger project but in an intimate way. [MB]*

The perception of the community aspects of SE3D was less for those located outside Bristol, non Bristol based subjects also found draw backs with the level of use of the forums, there was a feeling that they were used predominantly for calls for help and lacked a strong community focus.

SE3D was also seen as useful in terms of getting funding and particularly for PR:

*It's not just an animation company, it's a scheme, it's backed by HP, it's backed by DreamWorks, it's backed by Aardman, it's got - it's a story, it's a technology story. The press loves to write about technology, they don't like to write about companies who are just doing something. [AP]*

Against this positive experience there were very few and very minor comments on the negative aspects of the whole SE3D experience, the non-browsable problem with the interface was cited as were the need for a floating producer role to match the Watershed's intermediary. Finally there was a request for more training and less explaining; not; 'this is the maths that goes on behind the scenes', but 'here is how to use it'.

When asked what the top three things were that they liked about the SE3D service the subjects replies seemed to be centred around two key things; the community and the networking contacts and support that it brought with it and the sheer power of the service.

*Getting in touch with other people that you never knew were in Bristol, was great. The ability to set aside time to make a film and have that supported by a maintained render farm, plus a name like HP, plus the whole SE3D scheme. The prospect that it was for Animated encounters, you know, Watershed's involvement. To have names linked to... and the enormous amount of favours that were drawn in on the basis of that. [DL]*

Also SE3D (and the partner names) functioned as a lever when asking for favours etc.

*Films@59 did the sound and for me they did it for nothing which was so incredibly generous. [DL]*

*Craig was on the mentoring panel so he gave me a desk and computer so it was free of charge. [AB]*

*It's just opened up doors and you know, previously I was stuck as an animator behind my desk, not really talking to anyone or meeting anyone from the outside world. It's opened up opportunities to meet producers and mentors. [AB]*

*This has just been a fantastic opportunity because we're seeing ourselves that we're moving on so much. [TJ]*

*I couldn't really have done this project without SE3D actually. [JH]*

When asked if they would be interested in a follow-up SE3D2 project there was a unanimous yes all round. Finally, having had access to the facilities of SE3D and the URS some subjects were concerned about how they would feel having to go back to previous methods of CGI production:

*Look at all this power we've got! It's going to be a real shame to have to let it go. We're going to be gutted. [SD]*

*It's been a lovely little adventure as far as I am concerned. [K]*

*I don't know how we're going to go back to the real world now. [TJ]*

# 11 Conclusions

## 11.1 Profiling the target user group

During the interview process it became apparent that the majority of animators have a well defined sub-culture. Initially this was typified through casual remarks made by the subjects during the interview, but it was soon backed up with more rigorous answers to some of the questions. The key points observed were:

Animators are more visually oriented; they prefer direct manipulation style interfaces to text-command based interfaces.

They want richer feedback; they often expressed a desire for controls that were simple and clear, sometimes exaggerated in a very 'cartoon-like' style: *I just needed a big green button that I could press [AB]*

Although they are involved in a technical area they are not computer scientists, they are not interested in learning about the underlying techniques to the system that they use. However, although computer graphics does not require a deep understanding of the technologies it does necessitate the subjects keeping up to date and adapting to new technologies, as such they are easily able to start using unfamiliar technological systems and they learn to use them as they go along. This approach was typified by the overlooking of the manuals in using the system and the difference in perceived complexity from learning/hearing about the system to actually using the system. One observation of interest here was that they generally wanted more feedback about the auctions to enable them to build informal mental models about what was going on.

Their ability to try and adapt to new technologies means that they are often less critical of usability factors than other types of user. Problems with usability were often observed indirectly from interview responses and not directly from interview questions that asked the subject to itemise usability problems.

Animators like to be in control. They want to be on-hand monitoring the progress of rendering as much as possible. This desire for total control led to some frustrations at the lack of controls to manipulate certain aspects of the rendering and backup storage. Part of this sense of control is a dislike of uncertainties, however this is not manifested in a desire for long term planning, if anything animators did much micro-scaled planning and control but very little macro-scaled planning. This aversion to long-term planning was a problem since long-term planning is at the centre of the optimisation offered by the auction system, especially the long-term-bidding, and this was one of the areas where the subjects did not behave as the developers envisioned them behaving.

Animators are community aware. They often work in isolation but feel a strong sense of belonging to a sub-culture and enjoy meeting up to share experiences. Although they are sometimes in competition there is still camaraderie around helping one another out with technical difficulties.

### 11.1.1 Implications for design

A key part of requirements gathering for interaction design is the profiling of the target user group, given that the target user group for the SE3D project is very well defined there are plenty of areas where the observed traits of the user group could have been leveraged to increase the effectiveness of the design. This point also forms part of the later argument that interaction design should take place as early as possible in the development process.

Another, possibly useful observation is that many of these identified traits are also to be found in the way children work with digital technology. It could be that interaction design for children can be a productive ground from which to migrate design guidelines. This would be beneficial as there is already a large body of research for this area.

## 11.2 Supporting the community spirit

Following on from the above discussion, one of the traits of the animators was that they valued their sense of community. This sense of community was fostered by aspects of the SE3D project.

There was physical support in the form of events and the physical space of the Watershed. The subjects were positive about this physical support for the community, and were positive about the different organisational players in the project; in particular they found it useful to have consistent faces from each organisation.

This casual/personal approach did however result in a loss of more formal contact channels and this could have been responsible for a failure to grasp what the organisations were doing behind the friendly face (and sometimes who the organisations were, in particular there was some ambiguity about Watershed's role). In general the subjects spoke highly of the support for the community spirit afforded by the SE3D organisation.

There was also support for the community in an online forum and web site, once again the casual/personal approach undermined the structured communication channels set up in these online services leading to a feeling that the subjects enthusiasm and willingness to embrace new technologies could have resulted in something special but that it never really gathered enough momentum.

## 11.3 Making users follow chosen behaviours

The failure of the online community to be used in the manner in which it was expected to be used is a theme that occurs in other parts of the system. Other examples include:

- The use of the server end of the Elephant store; it could operate as a sophisticated version management system but no-one used it as such.
- The user of the client end of the Elephant Store; subjects routinely held two different images of data on their computers, one to work on and one to feed the upload.
- The use of a combination of long term bidding and short term bidding; in practice the subjects made far more use of STB, and those that used LTB were primarily motivated by it's cheapness due to the fact that so few other subjects were using it.

These observations show that achieving a desired user behaviours is not just a case of setting up the systems and instructing the users how they should be used.

The systems need to be designed to promote a manner of working that has perceived benefits for the users, to encourage them to use them in the prescribed way. The systems need to support this manner of working through good interaction design and the system needs to be designed to preclude other, less desirable, manners of working.

Once again, this points to the role of interaction design early on the system design. Not just the design of the online support but the design of policies for all other aspects of the users interaction with the system in its widest sense, even down to communication with the users through more formal channels such as post, email and telephone.

The design of the experimental context as a whole can also be seen as part of the interaction design. Some subjects commented on the lack of conflict when bidding, either because they were bidding when no one else was or because they were bidding on LTB when no-one else was. The experiment could have been formulated differently to increase the level of competition between the subjects and thus better investigate the optimisation effect of the auctions.

## 11.4 Experience with rendering

The absence of high-level interaction design in the development of the front-end of the rendering service led to some user problems with the usability. In particular there were issues of user control of the processes involved and lack of user awareness of some of the features offered by the system.

However, the usability problems were not perceived as major and often were not perceived as problems at all since much of the feedback about the usability of the system was negated by the high

desirability of the levels of processing power available. There was a sense that even if it was problematic to use it was still worth it to gain access to the incredibly fast rendering times. This is an important observation as it means that lack of high investment in usability can be offset simply by access to desirable resources. This situation leads to acceptance in the users mind but it can still lead to inappropriate or non-optimal use of the system from the system developers' point of view. Also this relationship can change; for example as internal systems approach similar capabilities or as competing external resources emerge, the margin of acceptability engendered by the access to resources will become less and users will once again become more aware of any usability drawbacks that are inherent in the system.

In general then the subjects understood and were very happy with the underlying functionality their only issues were with the features and interaction at the surface of this part of the system.

## 11.5 Experience with bidding

In contrast to the above, the users experience with the auction part of the system was the reverse of this. Interaction design was applied to the development of the auction part of the system with the result that the users had few problems with the surface features. However, they still had issues, but these were with the deeper aspects of the system; the complexity of the system and the underlying model of the different auction mechanisms.

One issue was that the subjects found the GV auction model more confusing than the share model. Having two parameters to manipulate meant they were more able to state what they wanted but they were less able to understand how the system was behaving when allocating them resources. One of the animators traits identified above plays an important role here. The trait is that of easily shifting over to make use of new technologies. Rather than trying to work out how to use new technology and then using it, animators tend to immediately start using it and work out the how as they are using it. This trait has implications for the introduction of new technologies to this user group. New technologies should be introduced in a hands-on manner, giving them access to the system or a 'sand box' version of the system that they can experiment with before using the real thing. Introducing them to the technology by first explaining it lead to many of the subjects being concerned about the level of complexity until they actually used the system.

A second example of struggles with the underlying model were the problems with the wider picture of how such an auction model would fit in with a real-world, commercial context. Subjects were concerned with three things:

- How the system would work if the subjects were using real money instead of token money.
- How the auctions would function if they were in competition with much larger players with access to far more funds.
- Finally the key problem with the whole approach was the uncertainty in planning and costing of a rendering project.

For that last point another of the identified traits of animators plays a role. Animators are keen on control. Total control is one of the attractions of computer animation, in a rendered world of computer graphics the animator is in charge and can do just about anything they want to. Animators dislike anything which takes this degree of control away from them, be it a client with strong views on the creative side or an auction system which introduces a degree of risk to the planning and budgeting.

*[Animation is] all about reducing risk in production, for the whole producers job ...  
Why would you want a system where you are opening yourself up to a massive amount  
of risk? [BL]*

## 11.6 Interaction design and the development process

Several times in the above discussion we have noted that interaction design must be applied and it must be applied early on in the development process. Such application does not necessarily have to be a high investment of time and effort. In some cases better design can be achieved by applying design to such simple factors as the terminology used within the system, as we saw with the 'upload' and

'version' issues.

As well as being applied early on, interaction design also needs to be applied across all aspects of the user interaction, paying attention to every aspect of the interface; the software applications, the support web site, the training and the communication channels to the organisations involved. Once again this does not have to be a high investment activity, adopting a few simple guidelines can have a big effect.

## 12 References

- [1] Utility Rendering Service. <http://www.hpl.hp.com/SE3D/whitepaper-urs.pdf>
- [2] Autodesk Maya. <http://www.autodesk.com/alias>
- [3] SE3D. <http://www.hpl.hp.com/se3d/>
- [4] Watershed Media Centre. <http://watershed.co.uk>
- [5] Eshghi, K., Lillibridge, M., Wilcock, L., Belrose, G. and Hawkes, R. (2007) Jumbo Store: Providing Efficient Incremental Upload and Versioning for a Utility Rendering Service. To appear in FAST '07 Conference proceedings.
- [6] Vickrey, W. 1961, "Counterspeculation, auctions, and competitive sealed tenders." *Journal of Finance* 16 (1961), pp8-37.