

## Process Documentation

### Project Meeting Minutes

This document contains the minutes from every project meeting that occurred throughout the lifetime of the project. It aims to provide the reader with an idea of the decisions made and work allocation that occurred on The i-scream Project.

#### Revision History

25/03/01	Initial creation	Committed by: ajm4	Verified by: pjw2
			Date: 26/03/01
		Committed by:	Verified by:
			Date:
		Committed by:	Verified by:
			Date:
		Committed by:	Verified by:
			Date:
		Committed by:	Verified by:
			Date:

Project Meeting Minutes

[Introduction](#) ..... 2  
[Meeting – 29/09/2000](#) ..... 3  
[Meeting – 05/10/2000](#) ..... 5  
[Meeting – 17/10/2000](#) ..... 6  
[Meeting – 18/10/2000](#) ..... 7  
[Meeting – 23/10/2000](#) ..... 8  
[Meeting – 30/10/2000](#) ..... 9  
[Meeting – 01/11/2000](#) ..... 10  
[Meeting – 06/11/2000](#) ..... 13  
[Meeting – 08/11/2000](#) ..... 14  
[Meeting – 13/11/2000](#) ..... 15  
[Meeting \(sub\) – 13/11/2000](#) ..... 17  
[Meeting – 15/11/2000](#) ..... 20  
[Meeting \(sub\)– 16/11/2000](#) ..... 22  
[Meeting – 20/11/2000](#) ..... 24  
[Meeting – 22/11/2000](#) ..... 26  
[Meeting – 27/11/2000](#) ..... 27  
[Meeting \(sub\) – 28/11/2000](#) ..... 28  
[Meeting – 29/11/2000](#) ..... 29  
[Meeting \(sub\) – 29/11/2000](#) ..... 30  
[Meeting – 04/12/2000](#) ..... 32  
[Meeting – 11/12/2000](#) ..... 33  
[Meeting – 08/01/2001](#) ..... 34  
[Meeting – 15/01/2001](#) ..... 35  
[Meeting – 22/01/2001](#) ..... 36  
[Meeting – 29/01/2001](#) ..... 37  
[Meeting – 05/02/2001](#) ..... 39  
[Meeting – 12/02/2001](#) ..... 40  
[Meeting – 19/02/2001](#) ..... 41  
[Meeting – 26/02/2001](#) ..... 42  
[Meeting – 05/03/2001](#) ..... 43  
[Meeting – 09/03/2001](#) ..... 44  
[Meeting – 12/03/2001](#) ..... 45  
[Meeting – 19/03/2001](#) ..... 46

## **Introduction**

This document contains the minutes that were produced after every meeting the team had concerning the i-scream project. It serves as a guide to how and when project decisions were made as well as a guide to the allocation of project tasks.

In all, 33 group meetings occurred. Of which, 29 were full meetings, and 4 were “sub” meetings where only certain members attended to discuss a particular aspect of the project.

## Meeting – 29/09/2000

**Time** – 2:30pm

**Location** – Computing Science Building

**Group Members Present** – Ash Beeson, Tim Bishop, Alex Moore, Paul Mutton

**Staff Members Present** – J.J.Cinnamond (initially), P.S.S.Camp (initially), I.A.Utting (latterly)

**Absent** – none

The initial stage of the meeting was with John Cinnamond (the setter of the project). During this part of the meeting general requirements of the proposed system were discussed. The details of these requirements follow.

Systems that will need to be monitored software:

- Solaris, NT, maybe Linux (jc)
- maybe FreeBSD (group)

Information that should be monitored & gathered:

- Backups – do they complete?
- Swap space – is there enough?
- Memory – is there enough?
- Load – is it too high?
- Disk usage – is there enough?  
Look at /var on certain machines
- Monitor individual users & processes,  
e.g. monitor for a day specifically
- tracking
- Log file collation
- easy viewing and notification of messages in logs

Alerting system:

- Priorities of alerts
- what escalation each alert should have
- Alerts – must be easily viewable and configurable
- Grouping contacts for notification
- Easy to cancel warnings & alerts

- Methods:

- Beeps
- E-mail
- Webpage
- Maybe pager or sms

Information reporting:

- Email
- Web
- Small desktop applications
- Graphs of data
- Roll back logs
- Public monitor pages for everyone to look at
- Overviews of machine status

Implementation notes:

John Cinnamond informed the group that he would be able to supply a test machine possibly more than one, should the need arise. Must be well written software, as the time when it will be needed most, will often be when a system is very tight on memory or has a high load. And they themselves must not crash the system. Must be able to throttle data sent (maybe UDP), so as not to flood bandwidth and to set how much information is sent and how frequently. System shouldn't give up after one failed test, it should retry tests with maybe levels of

warnings, before reporting a failure – again, this should be configurable. Note that some systems are spread across multiple machines e.g. raptor file store.

- Don't use kernel data structures, always use APIs as they will be much more efficient.
- Start small, aim for extendibility and scalability.
- Ensure ease of configuration through a text file or a GUI, specifically quick adding and removing of hosts.
- Solaris is the most important platform then NT & Linux.
- On NT systems you won't be able to monitor "load" in the unix sense, just CPU usage – which is expected to be low.

It was noted that the system will be very useful to them and that we should use John Cinnamond for any technical queries about implementation. It was left that we should produce a requirements document draft for John Connamond to review before proceeding any further. These minutes will be used as the basis for an initial draft of the requirements.

The latter stage of the meeting was with Ian Utting, the project supervisor for the group. It was noted that the first stage was to review technologies and strategies to determine what is needed to get the project underway. This includes determining what process the group will follow, as there are no "proper" methods useful to every project. The group should determine the right process for the project. Ian Utting suggested the use of an "Audit Trail", a method of recording the discards of ideas and why, the changes, the ideas and also the bad ideas that the group have chosen. The group was informed that a plan should be submitted early on i.e. who, what, when - of how the system will be developed. This plan can then be revised to include what we actually did, for when the project is submitted. The group should spend no more than a maximum of 1.25 days per week on the project in order to allow for other university work. Each member will need to present a talk for 15mins, for which the group needs to devise a presentation, by then should have a rough plan of timetable, ideas and requirements for the system. A regular meeting time was set with Ian Utting and the group. There will also be occasional meetings with John Cinnamond. This regular meeting will be:

Every Wednesday – 14:00 to 15:00

Technologies to consider were also discussed, these included Jini, SNMP, XML. Specifically the Jini Java technology that may allow for some quite functional and maintainable code for presenting the interface to the system. When discussing these subjects Ian Utting suggested making a case for why any particular design should be used and determining which is better through group discussion. It was also recommended that the group look at other products that perform similar functions, build a tick list of features and requirements and decide which should be considered for inclusion in the project system. The key thing to remember will be to prioritise requirements ensuring that the "extras" are left till last.

The following tasks were listed for the group to look into over the next week:

- Produce minutes of today's meeting
- Start drafting milestones and requirements
- Look into central location for documents and templates
  - i.e. a website
- Look into technologies, get URL's and other references to information
- Look into CVS for document and code storage and version control
- Brainstorm ideas
- Setup a mailing list for ease of communication within the group
- Think of a name and logo for the project
- Find out our filespace information for the project
  - e-mail cs-sysadmin

## Meeting – 05/10/2000

**Time** – 2:30pm

**Location** – UKC Computing Lab Meeting Room

**Group Members Present** – Ash Beeson, Tim Bishop, Alex Moore, Paul Mutton

**Absent** – none

The initial stages of the meeting were focused around looking at some of the requirements, as set out in the previous meetings minutes, and try to build a rough sketch of how the system will be put together. This was really an open debate, with each member walking up to the white board and sketching their ideas which were then discussed.

After many revisions a rough sketch of how the system will interact with its various components was devised (though it will be revised at a later date).

A copy of this diagram can be found here:

</documentation/minutes/system-20001005.jpg>

The above diagram took the bulk of the meeting. Latterly, it was decided that we should think of a name. After much debate "The i-Scream Project" was decided on, for reasons that are beyond the scope of this document, but can be found on the website.

The following requirements are still outstanding:

- Decide on a design model
- Decide on names for the various components
- Consult with customers (jc, pssc)
- Look at existing systems
- Setup webspace and investigate purchasing of domain?
- Setup CVS for file versioning and investigate usage on website
- Setup shared space
- Look at technologies that are available to us
- Learn how to use these technologies in the context of this project

## Meeting – 17/10/2000

**Time** – 7:00pm

**Location** – 2 Mead Way

**Group Members Present** – Ash Beeson, Tim Bishop, Paul Mutton

**Absent** – Alex Moore

Initial discussions surrounded the fact that iau wants a plan from us pretty soon. However, we soon digressed to the other things we needed to get working on. These are as follows;

- Produce a feature list - re: first minutes
- Produce an "implementation phases" document, based on the feature list
- Prepare the above two for review by jc (and iau?)
  
- Set ourselves some rough milestones (ie. phase 3 by xmas etc...)
- With the above in mind produce a plan for the overall project and assign initial roles.
  
- Define the Interfaces of the various server classes
- Define the Protocols for cross-network data

Discussions about all of our limited knowledge of C/C++ put an end to the plan that one of us could actually write all the host programs :)

Some questions and points were raised

- Differences between Solaris SPARC/Intel - will program run on both ?
- The "Client Interface" in the server - how much "thinking" will it do, and how much will be left to the client. (Thinking about bottlenecks).
- Database, need to get a machine sorted, and pick database.
- Would multicasting work for server to client Communications?

Next the document on the Implementation Phases was drawn up. It's only provisional at this stage, but gives us some initial ideas. It was noted that by using "phases" we could have milestones/breaks for testing, documentation and review. This would ensure that we keep things under control, and don't get swamped in code. It was also mentioned that using phases would help to keep the whole project in sync - ie. the server code would not run off ahead of the host code.

Paul Mutton leaves at 7:45pm, whilst Ash Beeson & Tim Bishop continue with implementation phases document.

The final Implementation Phases document is separately stored.

Meeting is concluded at 8:30pm to watch MIB.

## Meeting – 18/10/2000

**Time** – 2:00pm

**Location** – Computer Science Meeting Room

**Group Members Present** – Ash Beeson, Tim Bishop, Alex Moore, Paul Mutton

**Absent** – none

This meeting was spent producing a features list for review by jc. This list can be found elsewhere. The idea behind this list was to summarise what the end product was going to be like, from our current plans, and to ensure that it met with the initial requirements of jc.

Also produced was a revised diagram of the system, taking into account the following alterations from the original diagram of 05/10/2000.

- Web interface scrapped.

It was decided that this was no longer required, since any scripting language would connect directly to the database to retrieve information.

- Collector/Filter redefined.

The collector/filter part of the server was redefined such that a hierarchy of them could be arranged to spread the load of incoming data. Ultimately all data goes through the last (and main) one, but it does at least give an extra layer of organisation for the administrator. i.e. machines could be grouped (by site?) and have their own collector/filter which reports back to a central system - almost like a proxy.

- DBI (Database Interface) added.

This interface connects directly to the database, and is therefore the only part of the system that need know the exact workings of the database. It can also be given functionality to decide what information will be stored in the database, and how it will be done.

- Clients split to "real-time" and "historic"

The clients are split into two groups. Firstly the "real-time" clients (on the bottom left) connect directly to the system via a client interface. This allows the clients to receive up-to-date information directly, rather than through the database. Secondly, the "historic" clients connect to the database, either directly or via an interface, and allow information about the history of a machine to be viewed. Both of these types of clients could actually be implemented in one physical application, but the distinction has been made at this level. It should also be noted that the "real-time" clients have information pushed to them by the server, whilst the "historic" clients pull information themselves.

This system can be seen in the following diagram;

</documentation/minutes/system-20001018.gif>

The meeting was concluded at 5pm.

## **Meeting – 23/10/2000**

**Time** – 2:00pm

**Location** – Computer Science Meeting Room

**Group Members Present** – Ash Beeson, Tim Bishop, Alex Moore, Paul Mutton

**Absent** – none

Initial discussion was on the first deliverable for the project. This consists of an initial plan of the project, including rough outlines of the types of deliverables required and estimated time due to be spent on each section of the project.

It was estimated that a total of 1000 hours would be spend on the project.

Tim managed to untie the computer security cables without breaking the plastic cable-ties. It was decided that this was a \*real\* achievement and thus should be noted in these minutes.

Meeting is concluded @ 12:00

Meeting was re-convened @ 2.00pm. The plan was finished off with more detailed hours being placed on each section. It was printed and delivered to iau, giving him time to review it prior to the meeting on Wednesday.

Meeting was concluded at 3.15pm.

## Meeting – 30/10/2000

**Time** – 10:20am

**Location** – Computer Science Meeting Room

**Group Members Present** – Ash Beeson, Tim Bishop, Alex Moore

**Absent** – Paul Mutton

Absentee(s): Unable to attend due to adverse weather conditions.

The intention of the meeting was to complete the Time Plan Allocation, which is a required deliverable. Although almost finished tasks had not been allocated to group members.

Tim noted that a soft reboot of Compsoc1 may be required to enable the meeting to proceed as planned. Compsoc1 quieters down.

A discussion about the sizes of compiled C++ programs on Solaris concluded that the actual size of the binary executable is down to the set-up of the compiler on the target machine. It is decided that we should not worry about this hurdle until further into the projects development.

The technologies that the project would utilise were discussed and the following points rose.

- During the first few phases of the development our own protocol should be used for communication between Hosts and server, and server and clients. If a defined technology, such as SNMP, is decided upon then this can easily be substituted in due to the modular design of the applications.
- The protocol should be based on XML or a system not too dissimilar. The reason for this is it all text based and thus easier to process.
- The use of UDP and TCP/IP as communication layers between hosts and filer/collectors. To be discussed further with possible input from JC and PSSC.

The use of heart beats was also discussed. It was decided that further thought would have to be given to the implication of these before they would be finalised.

Meeting concluded @ 12:00pm

Meeting re-convened @ 3:00pm

Discussion on the implementation of the plugin technology concluded that data should flow through plugins in a sequential manner and should remain as plain text at all times. It was also discussed that plugins should have a 'order' coded into them, to allow the data to flow in a suitable manner for the plugins.

Meeting concluded @ 4:30pm

## Meeting – 01/11/2000

**Time** – 15:00pm

**Location** – Computer Science Meeting Room

**Group Members Present** – Ash Beeson, Tim Bishop, Alex Moore, Paul Mutton

**Absent** – none

It was decided that we should aim to start programming in the Project Week, which we believe is week 8. This gives us a week and a half to work out what we need to do before we can start the implementation, and do it.

Discussion on UDP packets, and whether we really want to use it for host to server communication. Need to find out more about how Java implements UDP, and whether jc wants to take this route.

It was decided that regardless of whether we used TCP or UDP, we would format the data sent between the host and the server in XML.

This is a rough list of the things we think we'll be needing to monitor, although some will not come until at least after the first main release.

### Solaris

- IP Address or Hostname
- MAC Address
- OS Name
- OS Version
  
- CPU(s) % Usage
- Load (3 values)
- Memory Total
- Memory Free
- Swap Total
- Swap Free
  
- Disk Space Total (per partition)
- Disk Space Free (per partition)
- Number of users
  
- Open sockets (how many etc)
- List "greedy processes" (ie. those that have consumed too many resources).

### Windows NT

- IP Address or Hostname
- MAC Address
- OS Name
- OS Version
  
- CPU(s) % Usage
- Memory Total
- Memory Free
  
- Disk Space Total (per partition)
- Disk Space Free (per partition)
  
- Shares in use
- User connected to resources

The discussion then side-tracked on to the host to server communication. It was discussed that having the server retain all configuration information would make management easier.

The host would know about the "well known port" that the Filter Manager is running on (maybe using /etc/services). Also, it could be possible to use a DNS CNAME (aka wpad!) to locate the machine that the server is running on. This would mean that the host has no configuration information stored locally, at least initially anyway. The configuration would be sent from the server to the host during the initial TCP communications between the host and the Filter Manager (or maybe initial communication with it's assigned host).

This is a very rough idea of the XML encoding we could use. It may not be 100% XML compliant, but that's not something we need to worry about right at the start. This first one was produced by Paul.

```
<?xml version="1.0" encoding="?????????" ?>
<data>
  <ip_address>###</ip_address>
  <mac_address>###</mac_address>
  <os_name>###</os_name>
  <os_ver>###</os_ver>
  <cpu>###</cpu>
  <load1>###</load1>
  <load5>###</load5>
  <load15>###</load15>
  <mem_total>###</mem_total>
  <mem_free>###</mem_free>
  <swap_total>###</swap_total>
  <swap_free>###</swap_free>
  <disk_total>###</disk_total>
  <disk_free>###</disk_free>
  <users>###</users>
  <open_sockets>###</open_sockets>
  <greedy>###</greedy>
</data>
```

Tim decided to put this forward, but doesn't have a clue about XML compliance... he just thinks it looks nicer. :)

```
<?xml version="1.0" encoding="?????????" ?>
<data>
  <identity>
    <ip_address>###</ip_address>
    <mac_address>###</mac_address>
    <os_name>###</os_name>
    <os_ver>###</os_ver>
  </identity>
  <system>
    <cpu>###</cpu>
    <load>
      <1>###</1>
      <5>###</5>
      <15>###</15>
    </load>
    <mem_total>###</mem_total>
    <mem_free>###</mem_free>
    <swap_total>###</swap_total>
    <swap_free>###</swap_free>
  </system>
  <disk>
    <part>
      <name>###</name>
      <total>###</total>
      <free>###</free>
```

```
</part>
</disk>
<procs>
  <users>###</users>
  <open_sockets>###</open_sockets>
  <greedy>###</greedy>
</procs>
</data>
```

Things we need to do before we can commence implementation:

- Investigate XML & Java
- Find out about UDP, and how it works in Java
- UML diagrams for the server and host

Meeting was concluded @ 5pm. Next meeting arranged for Monday, in the meeting room.  
Someone needs to book it soon.

## Meeting – 06/11/2000

**Time** – 10:00am

**Location** – Computer Science Meeting Room

**Group Members Present** – Tim Bishop, Alex Moore, Paul Mutton

**Absent** – Ash Beeson

Absentee(s): Unable to attend due to illness and bad meeting planning by the group as a whole.

The way in which CORBA will be sending information around the server system was discussed.

The filters will parse incoming XML sent via UDP and create an XMLPacket object. These objects are used to convey each host's information around the CORBA system. The object will provide accessors to an array of parameter name/value pairs.

Strategies for configuration distribution were discussed. The Logger class from last year's project is expected to make a reappearance in this current project. The main core of the server will have work put into it this week.

For the time being, we shall only be sending XML data that \*does not\* contain nested values.

Things that must be done before next Monday: - (oh yes, we're starting to implement bits...)

Ash -

In this person's absence, it was decided that they may write the test host application (in Java). This program shall connect to a specified server and send arbitrary non-nested XML data via UDP. (As per the 1st XML example at minutes/minutes-20001101.txt)

Tim & Alex -

Define a method of configuration distribution and produce the configurator. Adapt last year's Logger class as appropriate. Both the configurator and the logger should be registering themselves with the CORBA name service. It may also be necessary to write the CORE module of the server on which the rest resides. IDL definitions required.

Paul: -

Create the server's Filter class. This will accept incoming UDP connections and parse the XML data received. Create a separate XMLPacketMaker class to be used by Filter – this will be used to create the XMLPacket object for each incoming connection, providing a way of passing information about the CORBA system. Complete as much of this as possible before next Monday.

Meeting was concluded @ 13:00. Next meeting arranged for Wednesday 8th November 14:00-17:00, directly after our meeting with iau at 13:30 that day.

## **Meeting – 08/11/2000**

**Time** – 2:00pm

**Location** – Computer Science Meeting Room

**Group Members Present** – Ash Beeson, Tim Bishop, Alex Moore, Paul Mutton

**Absent** – none

It was decided that the meeting should be short due to the fact that we have deliverables for Monday.

Alex described his Configurator Idea. It is agreed that it is probably a good idea and probably should be researched further.

The logger class from last year has been added into CVS.

Discussion of UDP packets. Decided that UDP packets out of order should be dropped rather than the filters reorder.

Next meeting arranged and agreed to be 9am 13/11/00 Meeting room.

## Meeting – 13/11/2000

**Time** – 9:00am

**Location** – Computer Science Meeting Room

**Group Members Present** – Ash Beeson, Tim Bishop, Paul Mutton

**Absent** – Alex Moore

The communications between host and server were discussed yet again. Paul argued that the TCP communications were unnecessary, and UDP would suffice. Tim pointed out that the general unreliability of UDP would make it hard for the system to know whether a host had actually gone down, or whether it was a network problem. Tim also pointed out that the UDP packets may require sequencing, so the server knows whether packets were missed - for alerting purposes.

In the end it was decided that a combination would be used, with TCP used for heartbeats alongside the regular UDP communications. Different alerts could be raised depending on whether UDP packets weren't arriving, or the TCP heartbeat fails (the latter being considered worse).

Discussion continued with Paul outlining the basic situation with XML parsing. It was noted that various external java classes would be required for parsing (SAX, and javax...). This should not be a problem.

Finally the XMLPacket was discussed. It was noted that the XML data must somehow be laid out in the XMLPacket, retaining it's structure. The following was suggested, with the data being stored as tuples in the XMLPacket.

[XML]

```
<data>
  <value1>val1</value1>
  <sub1>
    <value2>val2</value2>
    <value3>val3</value3>
  </sub1>
</data>
```

[XMLPacket]

```
"data.value1", "val1"
"data.sub1.value2", "val2"
"data.sub1.value3", "val3"
```

This format makes it easy for the "end client" to extract information, only knowing the format of the XML data. This helps to ensure consistency across the system.

It was decided that some kind of Hash would be the quickest way of storing (and accessing) these tuples of data. The group agreed this idea would be best.

The lifetime and location of the XMLPacket is still an uncertain area. We must ensure they are left for the Garbage Collector when they are finished with.

Prior to the next meeting Paul will continue to investigate the XML parsing, and work on that side of the Filter. Ash will continue work on the host side (still in Java). Alex & Tim will collaborate on the FilterManager, ParentFilter, and possibly the CORBA side of Paul's filter system.

Finally, the group members (that were present) voted unanimously that Alex would be doing the hard parts of the C++ bits in the host. :)

Meeting was concluded @ 11:00. Next meeting booked for 11:00 on Wednesday, until 12:30.  
Meeting with iau @ 13:30 on that day.

## Meeting (sub) – 13/11/2000

**Time** – 2:00pm

**Location** – Eliot College, E3E room 8

**Group Members Present** – Tim Bishop, Alex Moore

**Absent** – none

This meeting was between ajm4 and tdb1 to discuss and possibly implement parts of the CORE and FilterManager elements of the system.

Initial discussion was on the configuration system of the CORE which has been partially completed to date. A key decision that was made was that it will NOT be possible for the elements of the system to update their own configuration, as this will cause difficulties if the user wants to use the file themselves (ie, add comments). This isn't seen to be a real problem though, as it was only going to be a "funky" extra ;-)

[Tim: this will also solve security issues with unauthorised third parties changing settings.]

A standard has almost be devised for filenames of configuration files. As it stands all hostname properties will be get in a file with the following name;

hostname.domain.properties

nb. This will always be \*lower case\*. This is important as some hosts (e.g. stuE...) have mixed case, so having all lower case saves confusion.

However, any other config files (such as the filterManager) can have any name in any case. The file itself will be of the name format;

name.properties

So in the case of the filterManager this would be "filterManager.properties".

A standard should be defined here for consistency.

Next point was whether or not we would support elements being able to be updated without restarting them, ie. a host would be able to detect its configuration has changed and adjust accordingly. The method chosen was to use the last modified date stamp of the configuration file as an indicator. When a Configuration object is passed to an element of the system it can determine when the loaded configuration was last edited. The element of the system can then periodically (the period can be set in the configuration ;-)) ask the Configurator if its configuration has been updated, and act accordingly. The methods:

long Configuration.getLastModified() and  
boolean Configurator.isModified(String conf, long curTime)

where thus implemented and the configurator test class modified to include a test of this system.

It should also be noted at this stage that a java 'long' is defined as a 'long long' in the IDL->Java mapping.

The next item discussed was the CORBA IDL and general system package structure. It was decided to use the i-scream domain as the root identifier for the project. Packages therefore follow the naming:

uk.org.iscream.<package>.<class>|<sub-package>

The IDL and currently implemented classes were updated to reflect this change.

The next item to be discussed was the initial thinking and possible implementation of the FilterManager. An initial FilterManager class was fleshed out with hooks to the logger and the configurator, it then creates a FilterListener class which will listen for new Hosts trying to connect to the system. It was NOT decided how Hosts should find the system however a method similar to the WPAD system used to locate web caches was suggested by Tim.

A Host contacts the FilterManager to initialise itself and obtain a host and port number of a Filter that it will talk to. When the FilterListener is contacted by a host it spawns a HostInit object to handle this intialisation. The first thing a Host does is obtain its configuration, this is done by the HostInit object obtaining the configuration of the host on its behalf. An initial protocol of how this system works is as follows (of course this is subject to later alteration, but seemed like a "good idea at the time"):

```
Host          FilterManager.HostInit
STARTCONFIG ->
              <- OK | ERROR
LASTMODIFIED ->
              <- LASTMODIFIED
DO {
  PROPERTY ->
              <- VALUE | ERROR
} UNTIL GOT CONFIG
ENDCONFIG ->
              <- OK
```

If there is an ERROR returned after STARTCONFIG, this indicates to the Host that there is no configuration available for it at this time. It may be possible to continue using default values, but this is up to the host configuration. Again, this is not a certain feature and should be discussed with other members of the group.

If the property section returns an ERROR then this indicates to the host that that property requested doesn't exist, the Host will then deal with this either by ending with an error on the local system or by ignoring it if it can.

The above features were implemented.

The next item needed to be developed is an architecture for FilterParent's and FilterChild's, as the next stage of Host initialisation is to be passed a 'reference' to a FilterChild. This 'reference' will be a server and a port and not a reference in the CORBA or Java way of thinking.

It was noted that on a "heart beat" with the system, a Host should check to see if its configuration has changed. If it has it should then re-initialise itself. This will allow configuration changes to be made to any part of the system knowing that on the next "heart beat" the configuration will take effect. A point to note about this is that if there is an error in the configuration the Host will simply error and die. What we may want (but still undecided) is a constant retry until the config is ok. So the system administrator will see in the logger destination that there has been an error in this new configuration and can make changes to rectify the problem, without having to interact with the Host program itself.

All code generated was placed in the "experimental" tree of CVS, awaiting approval from other group members. It should also be noted that although CVS records Tim as the 'checker inner', it was in fact Alex...as he was the king of code during this meeting ;-p

Many small bug fixes were made to the existing code.

It was also noted that there is a need for coding standards.

## Project Meeting Minutes

As we had reached an appropriate stage to end, and given the late hour, it was decided to conclude the meeting. It was, after all, nearly 7 hours of coding ;-)

Since this meeting took place tdb1 has produced Makefiles to allow easy compiling and configuration of the code.

Meeting was concluded @ 20:45. Next meeting booked for 11:00 on Wednesday, until 12:30. Meeting with iau @ 13:30 on that day.

## Meeting – 15/11/2000

**Time** – 11:00am

**Location** – UKC Computer Science Meeting Room

**Group Members Present** – Ash Beeson, Tim Bishop, Alex Moore, Paul Mutton

**Absent** – none

Meeting postponed until fire alarm finishes. It is noted that Ash and Paul would have been burnt alive if there was a real fire.

Meeting re-started at 11:20am.

Discussed the XML packet life problem. This has been identified as a problem because corba passes references to objects making it hard to determine when the object should be destroyed.

Paul begins implementation of a quotes page.

Paul suggests that packets should be stored in a queue structure, with 2 integers indicating how far through the queue each accessing function has got (from the start of the queue). This should be more efficient than storing flags inside each of the XML packet objects.

Someone needs to find out if you can 'clone' an object over CORBA. This would solve a lot of local copy problems. This thought was rejected by iau in the meeting.

Discussion of whether UDP packets should be numbered or time stamped proved controversial. In the end it was decided that each UDP packet should contain both a Sequence number and a timestamp (as defined by the host). It is therefore important that the host's time is set-up accurately by the sysadmin.

The whole issue of packet content is more of a host & client design issue than a server issue.

It was mentioned that the logging system should be able to deal with verbosity levels, in a similar way to JacORB. This would allow trivial messages to be hidden most of the time. The possibility of multiple loggers might also want to be considered (eg. file log with high verbosity, and screen log with low verbosity, running in parallel).

Meeting concluded @ 12:40

Meeting continued @ 12:45 by a tree

Present: Tim Bishop, Alex Moore, Paul Mutton

Discussion continued about the design of the filter system. The whole issue of how and where packets will be stored within the system needed clearing up before implementation could continue.

It was noted that the key function of the filter (given it's called a "filter") is to remove any packets of data it sees fit. With this in mind it was decided that the data could be passed on in text (or rather XML) format through the child filters.

This would work as follows in a child filter. Data would be received by one of two means, UDP or CORBA. The hosts would be sending UDP to the filter, and other "up-stream" child filters would send over CORBA. Regardless, it will always be the same content - a String of XML. In essence this means that the filter will be sending and receiving exactly the same string of XML - without any conversion required. Internally it may be verified through "plug-ins" to see if it should be dropped, but this would just be a series of independent tests. Finally the string will be passed on if the plug-ins allow.

This allows a chain of child filters going on and on in a tree-like fashion, which is what our original design permitted.

Finally, the parent filter will receive all the data from the child filters, and turn them into XMLPackets. These packets will be stored in some kind of data structure to be accessed by the various parts of the system.

This solves many of our key problems.

Meeting concluded @ 13:25

Meeting continued @ 13:40 with iau

iau briefly suggested that we alter the location of the database in our system. He suggested moving this into the parent filter, and then having the data passed straight on to the client interface.

Nothing firm was decided, but it should be analysed further.

Meeting concluded @ 13:55

## Meeting (sub)– 16/11/2000

**Time** – 2:00pm

**Location** – UKC Computer Science Meeting Room & Eliot College E3E Room 8

**Group Members Present** –Tim Bishop, Alex Moore

**Absent** – none

Another sub-meeting where a few design decisions were made regarding the core of the system, namely logging and configuration.

Firstly the idea of verbosity of logging messages was finalised, and later implemented. The following levels are currently supported by the write method of the logger, which can be accessed as follows.

```
Logger.write(String source, int verbosity, String message);
```

The verbosity can be one of the following values,

Logger.FATAL	- (0) Fatal or Critical Errors
Logger.ERROR	- (1) All Errors
Logger.WARNING	- (2) Warnings
Logger.SYSMSG	- (3) System Component messages
Logger.SYSINIT	- (4) System Component initialisation
Logger.DEBUG	- (5) All debugging messages

Although these are only integer values they should always be reference by their name on the left. This allows for easier reading of code, and greater expandability.

Next, still on the logging system, a need was found for a logger which allowed both logging to the screen and logging to a file, preferably with independent verbosity levels. It was decided that a multi-logger should be written that supported both features (using the existing classes).

The configuration system was also under review. There seems to be a need for default config files, grouping of configs, and hierarchical includes. These features will all be implemented allowing a greater degree of control over the system configuration, whilst retaining a simple and central point of management.

It was decided that the structure of the configuration would be as follows;

```
CONFROOT/  
  system.conf  
  hosts.conf  
  clients.conf  
  HOSTS/  
    host-hostname.domain.conf  
  CLIENTS/  
    client-somename.conf
```

system.conf is a system wide configuration file (for the logger and so on), whilst the hosts.conf and clients.conf are default configurations for hosts and clients respectively. The HOSTS and CLIENTS subdirectories contain individual configuration files, and possibly group files. The filenames are not yet final.

The intended use of the system is that the configuration system, when asked for a property, would first look in an individual host (or client) configuration file. If it failed to find it there it would then try the default file, and finally failing that it would return null.

Includes would also fit into this structure, although it isn't entirely clear how this whole structure fits together, with overriding and so on. Circular includes have been thought of, and will be dealt with.

Finally, it was noted that there were still problems with the initial configuration of the system - ie. how to configure the configuration system. Various ideas were suggested including passing parameters on the command line, or an initialisation configuration file.

The configuration plans above are to be considered and implemented by Alex, hopefully by Monday. Tim will work on the logging side of the system.

Next full meeting is arranged for Monday, and the meeting room is booked from 10am-12pm and 2pm-5pm.

## Meeting – 20/11/2000

**Time** – 2:00pm

**Location** – UKC Computer Science Meeting Room

**Group Members Present** – Ash Beeson, Tim Bishop, Alex Moore, Paul Mutton

**Absent** – none

Alex & Tim reported back about the latest modifications to the CORE of the system. The main point was the hierarchical layering of includes within the configuration files. The following format was decided:

```
include = file1.conf,file2.conf,file3.conf
```

This line would be in the root configuration file. Each include would be processed in turn, checking to see if they have any includes. All of these would then be brought together into one list.

Assuming the root configuration above is root.conf, and that each of the includes has two includes, we would get the following list of configuration files.

```
root.conf, file1.conf, file1a.conf, file1b.conf,  
file2.conf, file2a.conf, file2b.conf, file3.conf,  
file3a.conf, file3b.conf
```

When a property is requested each of these files will be checked, in order, until the requested property is found. It was noted that this would allow a very manageable and maintainable setup for the configuration system.

NB: The Configurator is now known as the ConfigurationManager in the CORE.

Paul reported back that the XML/UDP stuff was finished, and a few bugs had been fixed. Ash reported back that the Java host application was coming along.

Next to be discussed was the hierarchical filter system, and how this would all be tided together into the existing system. Firstly, the system was defined as such.

Each Filter object will listen for UDP packets (containing an XML string) on a given port, and they will also have a method that can be invoked (passing an xml string). Both of these ways of receiving an XML string result in the same process of events.

The Filter then passes the XML data through a chain of events that will ultimately dictate whether the data will be passed on. The Filter has a "parent" reference to another filter, to be contacting over CORBA. This parent could be the root Filter, or indeed just another filter. The data, if it hasn't been filtered out, will be delivered to the parent.

Internally a filter has a series of plugins. These plugins will be given the XML data (probably through an XMLPacket object) and will return a boolean. True indicates the packet is fine, False indicates it should be rejected. As soon as a plugin returns false the packet is dropped.

Filters will obtain their configuration over CORBA from the ConfigurationManager. This will include their parent filter reference, and possibly which port to listen on.

It was decided that Alex & Tim would sort out the few areas left in the CORE, and then prepare a base Filter (just the CORBA related stuff). Paul would then attach his UDP/XML code onto this to produce an almost complete Filter. This should hopefully be done by the next meeting, when implementation of plugins can be arranged.

Another issue that needed addressing was the effect CORBA referencing has on GarbageCollection. Ian has already said he'll look into it, but Paul has offered to question Richard Jones about it.

In terms of Goals it was decided that by next Monday the basics of the Server should be written (this includes the CORE and the Filter system). For the remaining three weeks it was put forward that we should attempt to have the Client Interface written at a basic level, the database set-up sorted, and a basic client running. A good Java host should also be completed. This target is important, as there is a lot more to do during the next term.

The next meeting is planned and booked for Wednesday 22nd, between 3pm and 5pm. Group members should see above for what they should aim to have completed by this date.

## Meeting – 22/11/2000

**Time** – 3:00pm

**Location** – UKC Computer Science Meeting Room

**Group Members Present** – Ash Beeson, Tim Bishop, Alex Moore, Paul Mutton

**Absent** – none

The meeting was short due to a prior engagement at 4pm.

The group reviewed the current progress of the server and produced a detailed diagram of the current state of the system. This allowed the group to clearly identify the parts that needed to be finished off, and familiarise everyone with how things are fitting together.

Alex & Tim agreed to finish off parts of the Filter Manager, and the Filter. This should all be completed for Monday.

Paul is to start some preparation work for the database part of the system. He will look at how things fit together with regard to Java, MySQL and PHP.

For information, here are the details of our MySQL database currently on raptor.

Database server: raptor.ukc.ac.uk  
Database name: co600\_10\_db

Username: co600\_10  
Password: ....as if we'd put the password in a document?

To get started with the database the following command would probably be quite useful;

```
mysql --database=co600_10_db -u co600_10 -p
```

A password will be prompted for.

Paul suggested that some time needs to be put into coming up with a design for the database. This should be discussed further at the next meeting.

Finally Ash said he should be able to have the Java host ready for Monday.

The group agreed that Monday would be a good point in the development process to have a code review session. This is so that all group members are aware of the processes involved in the whole system, and how it fits together. It will also bring out issues such as coding standards and design issues that should be consistent across all code.

The next meeting will be on Monday afternoon at 2pm. The group concluded that the morning meeting was pointless due to constant difficulties with people getting there, and other commitments in the later parts of the morning.

## Meeting – 27/11/2000

**Time** – 2:00pm

**Location** – UKC Computer Science Meeting Room

**Group Members Present** – Ash Beeson, Tim Bishop, Alex Moore, Paul Mutton

**Absent** – none

Alex reported back that the configuration system is finished, and has been fully documented in a separate document. Paul informed the group about the database bits he did, and the changes to the filter set-up. Ash said the Java host should be checked into CVS later on. Tim told the group that the TCP parts of the Filter were complete, and the whole system was in a working as it stood.

An issue was raised about the use of /experimental and /source in the CVS repository. There was some confusion between the group members over what should be put in each section.

It was agreed that code which would be part of the product would be put into /source, whilst /experimental would be used for separate bits of code - generally those testing an idea before it gets merged into the overall system.

Next the group reviewed the current state of the code. Everyone seemed happy with how things stood, and everyone got a good understanding of how the whole system went together.

It was decided that Tim should try and put together some coding standards based on ones used previously. These should be adapted to suit the current situation, and then be presented to the group. Once agreed, all code will be checked to ensure adherence to the standards.

The database was another issue raised. It would soon be necessary to ensure all data is logged into the database, and this requires some database design and planning. Paul agreed to put together a design for the database for review by the group. He also plans to look into, and implement, the java side of the database connectivity, and the PHP user interface.

Before the next meeting Ash will have checked in the Java host, and it should at least be able to send data to the server. Paul will have a database design, and maybe some documentation for the XML data. Alex will generate some more documentation, and a nicer version of the GUI-based logger. Tim will patch up a few of the missing bits around the server, and produce some coding standards and logging documentation.

The next meeting will be on Wednesday afternoon after the meeting with Ian. Meeting room is booked 3pm to 5pm.

## Meeting (sub) – 28/11/2000

**Time** – 1:00pm

**Location** – Eliot College, E3E room 8

**Group Members Present** – Tim Bishop, Paul Mutton

**Absent** – none

A small sub-meeting was held between Paul and Tim to discuss the plugin structure in the Filter, and some database design issues.

First to be discussed was the Filter plugin system. Paul has agreed to implement this setup. The following design was put forward, any objections should be noted at the next meeting.

A PluginManager will be in charge of handling the individual plugins, and will provide a single point of contact the the Filter system. The PluginManager will be constructed in the main method of the Filter, and a single reference passed to the three subcomponents - the FilterServant, UDPReader and TCPReader. These three in turn will pass the reference on to the FilterThread they create.

The FilterThread is currently the final point in the Filter that has the data. At present it just passes the original XML string on to it's parent. The PluginManager will be used just before this event occurs. It will provide a single method that can be called by the FilterThread. This method will take an XMLPacket object and return a boolean. The FilterThread will use this boolean value to decide whether to send the XML on to it's parent.

The idea behind having this single PluginManager object is that it will save recreating instances of the actual plugins each time a data item arrives. It might provide a method to reinitialise itself, but this has not been finalised.

The actual plugins themselves will implement a single Java interface. The PluginManager will hold an array of these objects which it will pass the XMLPacket through in order. These plugins will reside in a subdirectory, hopefully, and will either all be loaded, or be configured in the central ConfigurationManager.

The main issue with this is concurrency. Obviously only having one instance of the PluginManager could cause problems with three threads trying to call it. This should be discussed with the group, as threads can be a tricky thing to get absolutely right.

Other issues discussed include the database design. Paul has produced a design for the database tables, which is available in CVS for the group to examine. Some issues arose with the SQL insertion routine, and whether or not a lock would be required. This should be investigated further.

Paul also agreed to look into the Java side of the database system. This would be a single module that would link in to the root filter. The root filter would pass the raw XML strings into this module, which would in turn decode them and store them in the database.

This brought about another issue, namely the XML decoding classes. These are currently used in both the filter and root filter. It was decided that they would not be needed in the root filter as it would just pass the raw XML data. This only leaves one point they're needed, at present. However, this will change as the database interface will require them and no doubt the client side will too. It was suggested that these should be moved into a separate package that could be used by the whole system.

Packaging of classes is an issue that should be discussed at the next meeting, and then put in place.

Other group members should have read these minutes prior to the next meeting if possible.

## Meeting – 29/11/2000

**Time** – 3:00pm

**Location** – UKC Computer Science Meeting Room

**Group Members Present** – Ash Beeson, Tim Bishop, Alex Moore, Paul Mutton

**Absent** – none

First up on the agenda was the database design. The group spent quite some time trying to come up with a simple, yet efficient database design that proved to be quite tricky. Issues such as concurrency, locking of tables came up, as well as whether we should classify certain data items as "key" and store them separately.

It was eventually decided that a rather stateless design would be implemented, at least at this stage. This was a two table design with one table containing a single entry "per xml packet", and another table containing a single row per data item. This meant that each row in the first table could be associated with one or more rows in the second.

This design was not completely finalised, and Paul is to review this, and do some research into how this could be implemented. Ultimately the design needs to be tailored for our implementation, but not to the point that it restricts further growth and expansion of the system.

Paul and Tim presented the Plugin system ideas. The group agreed that it was fine and should go ahead. It was put forward that the PluginManager should be a singleton class to avoid passing too many references around. This seems a good idea.

Ash gave feedback on the current state of the java host. The group requested that the host be in a state that would permit all members to run it for testing the server with. Ash agreed to implement this.

The meeting finished with a discussion on what the next stage of development should be. Ash agreed to polish off the java host so it implements all the functionality that the final host will have. Alex suggested that he and Tim look at the existing code, especially the filter, with the aim of tidying things up and standardising all the classes. Tim put forward that the code should be placed in packages at some point soon, as some parts of the system were getting messy, and a few classes were needed in multiple places.

In the longer term development on the database front, and the client side should be started. Some design work will need to be done, but the group should be thinking about what could be done prior to the next meeting.

The next meeting will happen on Monday, although the room is not booked. Arrangements should be made prior to the weekend.

## Meeting (sub) – 29/11/2000

**Time** – 5:00pm

**Location** – UKC Computer Science multimedia lab & Eliot College computer room

**Group Members Present** – Tim Bishop, Alex Moore

**Absent** – none

Alex and Tim met to discuss and work on the code tidying issues. It was decided, afterwards, that the work should be documented in the form of minutes, so that a record can be made of what's happened.

Meeting start (MM Lab): 17:00

Break for dinner: 22:00

Resume (Eliot) : 22:30

Finish : 04:30

Firstly a new class was introduced to the system. This class, called the ReferenceManager, was originally designed to hold references to the various items that a system component (ie. the filter) would require. This would cut down on passing of references between classes.

However, as the was implemented it became clear that it could in fact handle the CORBA references as well, and then, further to that, all the CORBA initialisation. In the end it ended up handling all the CORBA code, and held references to all the things required by a system component.

The class was designed as a singleton, again to cut down on reference passing. Generally the class needs to be initialised before use, and this will be done in the main method of the component it's involved in. From then on a reference can be obtained by any class within the component using one simple method call.

This has significantly reduced the complexity of the main methods of the system components, and allowed a clear line to be drawn between the CORBA code and the rest of the code. This has also meant that all the CORBA error handling is focused in one place, and can therefore be more thoroughly dealt with.

The next item to be dealt with was the packaging. This was a major, but required, task and took some organising. The whole system now resides below the following package,

```
uk.org.iscream;
```

The currently list of subpackages are,

```
uk.org.iscream.core;  
uk.org.iscream.core.loggers;  
uk.org.iscream.filter;  
uk.org.iscream.filtermanager;  
uk.org.iscream.rootfilter;  
uk.org.iscream.util;
```

The first five containing the existing components, but the last has been created to house more general purpose classes. Currently it has the XML parsing classes, and the ReferenceManager. It will, in the future, hold a class to deal with naming and generation of toString()'s.

The inclusion of these packages has made the directory structure look a bit unwieldy, but as you'll see further down makefile's have been written to make compilation easier.

As a side note, the generated IDL classes reside in similar packages, although they are in a separate location to avoid confusion. Ultimately they will be merged.

Makefile's have been generated for the whole system. They allow easy compilation of any component, or the whole thing. They also allow for easy cleaning of compiled class files. It is expected that they will in future support packaging of the system into JAR files.

They work on a recursive structure, with a single root Makefile. Each directory contains its own Makefile for the files it contains, with links to any subdirectories that also contain Makefiles. This makes the system much more manageable when adding, changing, and removing class files. Ultimately only the root Makefile need be used, and it will recursively execute all the others.

As a result of the above packaging, and the ReferenceManager it became necessary to tidy large chunks of code. The main two done were the Filter and RootFilter. Both now make use of the ReferenceManager throughout, and have a nice common theme through them - similar variable names, and class structure. These classes are now much nearer the required standards.

Also modified were the loggers in the core. They now reside in a separate subdirectory of the core, although this could be moved. This layer of separation makes life much easier for managing and adding loggers.

That was about it for changes. A few bugs were fixed as things progressed, but the functionality of the system (from an external point of view) remained pretty much the same throughout.

A short list of "nice features" to be added was prepared. This list is in no certain order, and there is no guarantee they'll even be done in the near future.

- redesign the toString() methods. Modifying them, at present requires changing every class. This is bad. A new class in the util package could be created to centralise this behaviour. The toString() method is mainly used for logging purposes.

- implement "remote logging" sessions. This would require modifying the exist setup, but the majority of code is in place to support it already. The idea of remote logging is that a logging console could be fired up independently on the server, and allow the user to monitor the status of the system.

- improve error handling, and exception catching. Fatal errors should be informative to the end user.

- the SimpleSwingLogger is broken. It will, after quite a few thousand lines, run out of memory and crash. This bug was identified previously, but the cause only found during the course of this meet.

- allow hosts (and clients?) to log to the central logger as well. This would require a "logger proxy" of sorts, but it is suggested that this could be encapsulated in the existing UDP packages - ie. generate a packet of

  - type "logmsg" that the UDP reading class will direct straight into the system logger.

- sort of garbage collection issues, specifically with the Configuration objects, although there may be other areas that have yet to be found.

- the configuration system support dynamic updates, and the host makes good use of this feature. However, it would be nice if the server could implement this internally where possible.

The semi-meeting was concluded at a rather late (or early) hour after some very productive work.

## **Meeting – 04/12/2000**

**Time** – 2:00pm

**Location** – An Eliot College Seminar Room

**Group Members Present** – Ash Beeson, Tim Bishop, Alex Moore, Paul Mutton

**Absent** – none

All group members report back on current status. Paul had a few questions about the plugin system, although they are still to be resolved. Paul also informed the group of the current database situation, and the problems that still needed solved - apparently MySQL on raptor is a bit old ! Alex briefly mentions the work he and Tim did on the packaging last Wednesday (minuted elsewhere).

Next the group looked at where to go from here. It was decided that the client interface, database and web front-end should be worked on - this is the area the "other side" of the Root Filter. Alex & Tim will look into the Client Interface, whilst Paul will look into the Database Interface. Ash is to continue developing the host.

Alex raised a concern about whether the project is progressing fast enough. The group decided that the current milestone (to have a basic working system by Christmas) was a good target, and next term some more definite milestones should be set.

Next meeting on Wednesday as usual. Meeting room is booked, maybe we should try to book sooner ?

## **Meeting – 11/12/2000**

**Time** – 2:00pm

**Location** – UKC Computer Science Meeting Room

**Group Members Present** – Ash Beeson, Tim Bishop, Alex Moore, Paul Mutton

**Absent** – none

A brief review of our progress was carried out. We concluded that our goals for this term have been achieved.

Alex kept us up to date with the progress of the Component Manager.

The topic of the Christmas holiday was discussed. Ash will be using the period to learn C++ from his brother, Tim will be tidying bits up further and Paul shall be working on more database components and historical reports for the web.

Paul demonstrated the current work in progress on the historical reports. They are fully "i-scream blue" compliant.

We decided to wait until a future moment before deciding what our next 'big' milestone should be.

It was suggested that written documentation should be produced for using and maintaining each part of our system.

The next meeting will be on Wednesday afternoon after the meeting with Ian. Meeting room is booked 3pm to 5pm.

## Meeting – 08/01/2001

**Time** – 2:00pm

**Location** – UKC Computer Science Meeting Room

**Group Members Present** – Ash Beeson, Tim Bishop, Alex Moore, Paul Mutton

**Absent** – none

The meeting began in an enthusiastic manner, with all group members eager to get on - not what you'd expect for the first Monday of the Millennium!

First, a review of work done, if any, over the Christmas break. Paul was first to state that the "historical clients" he's been doing (web based) are a lot harder than they first seemed, but they are coming along very well. Tim reported that he had put a queue into the central point of the server, and added a few features (eg. javadoc generation) to the makefiles. Ash told the group about his work on the host, and plans to restructure it to aid porting to C++.

Alex was pleased to report back that it had been lovely and sunny in Tenerife, in contrast to the snow in the UK, and that he'd had some 'ideas'.... the group sighed :)

Initial discussions started with what needed to be done this term, and when by. It was agreed that there were probably 10 weeks at the most to get everything finished, and everyone was happy with the notion that it should be completed \*before\* the start of project week. A target of around 5

weeks was set to complete the majority of the coding, giving us the rest for documentation, testing and review. More precise goals may be set in weeks to come.

Alex mentioned that a "text based" client could be useful, and likened it to the "top" program found on raptor (/usr/local/sbin/top). This would display basic information in an easily digestible form, and would be quick to load. This is something that the customer has requested.

Alex also brought up an idea suggested by Tim a while back, the restructuring of logging in the server. The idea was that the log messages would be sent to a logging unit, as they do at present, but that this logging unit would pump these messages back into the filter system for delivery to clients. This would remove the single logging console (although it could still log to a file), and allow any client to view the current status of the server.

This brought on the discussion of how these would pass "un-filtered" through the system, which then led to the similar issue with heartbeats. This would be looked into by Tim, although no work would be done on the logging set-up at this stage. More emphasis was placed on making the server 'complete', which entails fixing the heartbeat problem.

Next it was decided what would be done in the immediate future. Paul requested doing the command line client after completion of the historical clients, which should be completed in around a week. The group agreed on this.

Alex said he had ideas for a GUI client, and would work on this. Ash said that a rewrite of the host was required to make porting to C++ easier. He also said work on the C++ host was to begin soon.

Tim is to complete the functionality in the client interface part of the server, and patch up other areas that needed fixing (eg. heartbeats). The aim is to "complete" the server at this stage.

Finally, it was pointed out that the server needs to be brought to a stage where it can be released and set running on raptor. This is required so the group members working on hosts and clients can test their setup is complete.

The next meeting will be on Wednesday afternoon after the meeting with Ian. Meeting room is booked 3pm to 5pm.

## Meeting – 15/01/2001

**Time** – 2:00pm

**Location** – UKC Computer Science Meeting Room

**Group Members Present** – Ash Beeson, Tim Bishop, Alex Moore, Paul Mutton

**Absent** – none

The meeting started with a review of the clients. Everyone was impressed with the status of them. Work is to continue over the next week or so to complete them.

The issue of the server locking up was to be addressed. However, in testing during the meeting it was decided that it was a problem in the ClientInterface - an issue that is still being resolved. Further investigation was put off until this has been completed.

Data to be grabbed was discussed. This is becoming an important issue, as the clients are requiring more precise details. It was decided the the "grabbing" part of the host should be separated, for now, from the main program. A Perl program will be written which will parse output from various system commands (which ports for different OS's) and output it in ASCII for use by the host application. This will also aid porting of the host into C++, which is continuing. It is hoped that this "grabbing" code will be moved to C++ eventually, but this is a good stopgap, and will also work well if moving to C++ cannot be completed.

The group expressed concern on the progress of the host application. With work moving fast on the client side a solid and reliable host was required. The current host is unpredictable and can crash unexpectedly. These issues are being addressed. The group also requested that the server be "completed" for a release soon, allowing it to be left running on Killigrew[1] for host and client testing.

Another issued raised was the client to server protocol. A protocol needs to be clearly defined to make server and client development easier. Alex & Tim discussed this and will produce a spec soon.

Tim raised the issue of Makefiles. The server benefits greatly from some extended Makefiles, and the other components could use some basic ones to allow quick & easier building. This will be looked into.

Finally, it became apparent that the util package of the server contains code required in the clients. It was agreed that this package should be available separately in a JAR file for use in the clients. Tim agreed to do this, arranging Makefiles and changing code to remove dependencies with the rest of the server.

The next meeting will be on Monday afternoon at the usual time. The meeting room is booked 2pm to 5pm.

[1] - Killigrew is a FreeBSD machine made available by Tim. Continued use is not guaranteed, but it does provide the group with a "shared" user for running the server. This makes stopping/starting/controlling the server possible by anyone in the group - something not available to us on Raptor.

## Meeting – 22/01/2001

**Time** – 2:00pm

**Location** – UKC Computer Science Meeting Room

**Group Members Present** – Ash Beeson, Tim Bishop, Alex Moore, Paul Mutton

**Absent** – none

The meeting began, as usual, with a review of the current status. Alex told the group about the rework of the swing client, and the work he and Tim had done on the server to client protocol. Tim reported that the client interface of the server now conformed to this protocol, but that more work would need to be done to meet the full requirements.

Tim also added that the bug found last week, of the server crashing, had been semi-solved by the use of Queue's. However, it was noted that the Queue's would still fill up and therefore cause the same problems. The group made a decision that when a Queue becomes full, data should be dropped. This has led to the obvious need for monitoring the state of the Queue's throughout the server, and this will provide valuable statistical information.

Ash said the C++ host was nearing completion, and the group clarified some of the issues with regard to sampling and averaging of data. Paul added that the Perl script for grabbing the system information was now in place, and ready to be used. Tim was quick to add that the Java host already used and supported it.

The group made a decision, last week, to use a Perl script as a stopgap/solution to the C++ problem of getting system information. The overhead is negligible, and the code is portable. It was considered acceptable for now, and in the long run, could be perfectly usable in a working system.

Paul made a request for the database to be populated with regard to demo'ing to Ian on Wednesday and data for graphing. It was decided to run the server and hosts for a period of around two days to provide a good supply of data. Tim added that permission had been granted by John Cinnamon to run the host on all the Rocks, providing that the load was not detrimental to other users.

The group discussed what would be shown to Ian, and made plans to have things done by then.

The next meeting will be on Wednesday afternoon after meeting with Ian - presumably he will have some ideas :) The meeting room is booked from 3pm to 5pm.

## Meeting – 29/01/2001

**Time** – 2:00pm

**Location** – UKC Computer Science Meeting Room

**Group Members Present** – Ash Beeson, Tim Bishop, Alex Moore, Paul Mutton

**Absent** – none

Paul informed the group about the host he had written in Perl during the week. It has most of the required features of a "basic" level host, and is more stable (due to regular expressions) than the Java host. Tim reported that the host (named ihost) was running fine on all the rocks, and had been for over 12 hours.

Tim reminded the group that another stab at a "24 hour" run was in progress, which had gone over the 10 hours of the first attempt. This was put down to the quota increase on raptor for MySQL.

Paul reminded the group of the changes to the database. Over the week he and Tim had discussed ways of improving database access, as 2 million rows for 10 hours was ridiculous. It was decided that a "flat" table with the XML string stored in a complete state would be best (with a few key fields extracted). Not only should this reduce the number of rows, but it should reduce the disk usage. Paul reported that this had already been implemented on the server side, and data was being added using this method.

Alex told the group about the progress with his client. Although there were only a few "cosmetic" changes, a lot of work had been done with the "behind the scenes" stuff to make it better implement the server protocol, and to make it more Swing thread safe.

Alex then informed the group of his plans to improve the configuration side of the client, to make it fully implement the server protocol, and to improve the visualisation of data.

Paul stated that the DBReporting tools would now need to be edited to use the new database format, and that he would work on this. He also said he'd get some more done on the command line client, although it was becoming tricky to make it work under both DOS and Unix.

Ash reported that the host development was going well, and all the central structure was in place. The "data grabbing" part still needed to be coupled with the Perl script, and the networking module had yet to be done. Tim added that he had some good stuff on networking, and would dig it out.

Ash also discussed some of the ideas behind alerting which both the host and the "local clients" would need to be based around. Tolerance levels will be stored in the central configuration, allowing them to be retrieved by both the host and the client. The host would use an "averaging" method to send data over longer periods of time, but based on these tolerances could send priority packets when something "bad" happened.

Tim was last to chip in and tell the group about progress with the server. The client interface was complete for TCP clients, such as those Alex and Paul are writing. The rest of the server is complete, and bugfixes have been carried out throughout the system as they are found. Tim suggested a "feature freeze" in the server for a while to allow work to continue on the rest of the system, and to provide a more stable testing platform for hosts and clients. The group agreed this was a good idea.

Tim said work was to continue on "patching and fixing" the server, but that he would now focus on setting up the architecture of the new "local clients". These would use a CORBA set-up to connect into the Client Interface part of the server. Further details to follow when design has been completed.

On a final note the group discussed the benefits of demonstrating the system to jc (this was later backed up by iau). The group agreed to get the client looking better, the reporting tools ready, and the local clients in a decent state before this could happen. It was noted that the feedback from a potential end user could be invaluable to the project, but that time must be given to implement any changes resulting from this feedback.

There will be a meeting as usual with Ian, but at the later time of 2.15pm (now a regular time). The next group meeting will be next Monday.

## Meeting – 05/02/2001

**Time** – 2:00pm

**Location** – UKC Computer Science Meeting Room

**Group Members Present** – Tim Bishop, Alex Moore, Paul Mutton

**Absent** – Ash Beeson

Absentee(s): Ash was absent from this meeting after oversleeping and not making it to campus on time. The group waited for around an hour before commencing the meeting anyway.

Initial discussions were, as usual, about the progress of the systems. A lot of work had been done since the last meeting, some due to the design change in the database. Paul put extensive work into completely rewriting the DBReporting tools, and they now run considerably quicker. Alex made the front-end of the client display nearly all the data in a graphical form, and changed more stuff "under the hood". Tim implemented the local client structure, on both the server, and a dummy client.

After feedback in the week from jc with regard to our database size, the group decided it was worth discussing his opinions. The group could understand the viewpoint, but stuck by the long standing decision to use XML formatting for the data. However, it was decided to look into compression techniques, and possibly removing some of the unneeded tags (using attributes?).

The group noted that time was slowly running out, and expressed concern on the development of the host. Ash was absent and unable to comment. There were 3 weeks remaining until project week, and a lot to do.

On the whole, the group were happy with the milestones at present, with some parts being further behind than planned, but others further ahead.

There will be no meeting with Ian this week. The next group meeting will be on Monday.

## Meeting – 12/02/2001

**Time** – 3:00pm

**Location** – UKC Computer Science Meeting Room

**Group Members Present** – Ash Beeson, Tim Bishop, Alex Moore, Paul Mutton

**Absent** – none

The meeting began with a review of the last two weeks, as all members were not present at the last meeting. However, a lot of these points were covered during the last minutes. Alex told the group about the new "service checks" that were implemented in the Filter system. These allowed checking of typical system services (WWW, FTP, SMTP, etc), the results being wrapped in XML in a heartbeat packet.

Tim had implemented the Queue monitoring system, and gave the group brief details of the packet layout. Ash requested some help with compiling a C++ socket library, which would hopefully allow the network parts to be hooked into the host. He also said development of the windows parts were progressing, but that the network stuff was also needed there.

Discussions were started about how the data from the service checks would be displayed. Tim and Paul discussed if any historical data could be generated, and Ash suggested a CGI script. Tim pointed out that there wasn't even any basic functionality in a local client yet, and this needed to be done first. He suggested the local client just pump data out to some text form (be it, csv, xml, html, etc) in the first instance, and then expand this afterwards.

Over the next week, or so, the group agreed that things should be pulled together. Alex said he would bring the client, Conient, to a complete point, and put some of the "funky" features on hold. Paul said he would finish of the database reports, and look at archiving/cleaning the database. Tim said he would look at the overall design of the local client, and implement it with a very modular design. Ash said, with the socket library compiled, that he would get the host working, and maybe look at CGI's for displaying service checks.

Alex brought up the issue of configuration grouping. It was shown before that it was needed, but it could be done without. Now that service checks have been implemented it was becoming even more apparent that it was required. Alex said he would look into doing this.

The group realised that not only was time running short, but everyone had a lot of other commitments too. The group agreed that completion of coding over the next week or two would be a good idea, as there are many other deliverables to be completed.

There will be a meeting with Ian at 2.15 on Wednesday, and a usual meeting on Monday at 2.15 (we have DEFINATELY booked it this time).

## Meeting – 19/02/2001

**Time** – 2:00pm

**Location** – UKC Computer Science Meeting Room

**Group Members Present** – Ash Beeson, Tim Bishop, Alex Moore, Paul Mutton

**Absent** – none

The meeting today focussed on looking at the C++ host.

The C++ host was looked at, and some problems fixed. Unfortunately this highlighted some more bugs in the server, which will need to be fixed. At the end of the meeting the C++ host was retrieving data and sending it to the server fine, albeit with a few UDP packet size problems.

This is good progress, as host stability is becoming more key for developing the rest of the system. The group agreed that the week ahead would produce more work, in the run up to project week.

There is now one week until project week, and planning was already being done. Meetings were arranged and booked for the week, and the group already have ideas of the goals that need to be reached during this time - namely the completion of coding. Documentation was another area that needing a lot of work, but the focus for now is to get the coding done.

The group needed to plan a time to demonstrate the meeting to jc, after a discussion with iau in the last meeting. It was clear that this would be beneficial, and should be done sooner rather than later - but there were a few things that needed to be finished first. It is hoped that this may happen towards the end of the week, but would most likely happen at the start of project week.

There is no meeting with Ian this week, as he is away. For next week we have booked two meetings - Monday @ 2.15pm, and Friday @ 2.15pm.

## **Meeting – 26/02/2001**

**Time** – 2:00pm

**Location** – UKC Computer Science Meeting Room

**Group Members Present** – Ash Beeson, Tim Bishop, Alex Moore, Paul Mutton

**Absent** – none

The meeting this week was merely a quick chat about the project week, before we went away to do some coding. The group decided Friday was the day to demo to John, although this should be checked with him. By this point, most coding should be complete anyway.

The general aim is to have most, if not all, coding done by the end of the project week - which is this week. That leaves us very little time, but we should be giving 100% to the project. The exception is Ash, who has business lectures still.

Friday's meeting will hopefully be just a review of the week, and checking if there is anything still to be done. Along with the demonstration of course.

We have a meeting with Ian this week, although Alex is unavailable due to a job interview. There is another meeting on Friday at 2pm, and then the usual meeting on Monday next week. Hopefully there will be a demonstration with John in Friday's meeting.

## Meeting – 05/03/2001

**Time** – 2:00pm

**Location** – UKC Computer Science Meeting Room

**Group Members Present** – Ash Beeson, Tim Bishop, Alex Moore, Paul Mutton

**Absent** – none

This meeting started with the long awaited demonstration to John... except he brought a few colleagues along :) With our captive audience, we demonstrated the past 20 odd week's work, and fortunately got lots of positive feedback. This certainly boosted the confidence of the group as it made all the effort seem worthwhile. One comment worth nothing was: -

"I wasn't expecting it to be quite so useful!"

After the demonstration, the group proceeded to discuss where to head next. Particular issues on the agenda were the webpage front end, and the alerting mechanisms.

Both Alex and Tim had spent a couple of very late nights writing the alerting mechanisms (the local clients) for the server. They noted that the logic behind it all was a lot more complex than had been anticipated, and that it really was a major part of the system. This work had been mostly done for some basic data (CPU levels) and just needed to be expanded to cover the rest. This was to be done over the following week.

Ash was pleased to report that the host was now working, although there still existed a problem with packet size. All the other issues with it crashing etc had been tidied up. He again noted that he was keen to get some work done on the webpage frontend.

Paul said he had completed most tasks, including the winhost and the DBReports. He volunteered to do some work on the webpage frontend.

The webpage front-end was something that needed doing. The group agreed it should be broken down into sections on latest host data, latest alerts, and historical information (db reports). These would all be tied together into one easily navigable interface, with a nice summary page. This would tie closely with the alerting mechanisms discussed above, so no plans for the actual implementation details could be made at this stage.

Another issue that evolved from this was the host sampling data and sending averages, and alerts. Ash said logic was in place to start this, and he had it planned. Alex pointed out that the initial design for the host was to be as lightweight as possible, and this would just complicate it, especially when the same work is carried out in the server. The topic wasn't completely resolved, although the group seemed to agree that it shouldn't be a priority with the short time left.

The final issue was documentation. This would be looked at over the week, and maybe another meeting arranged later on to plan it.

There will be a meeting with Ian at 2.15pm on Wednesday, and the usual meeting on Monday at 2.15pm. There may be an extra meeting in the week to discuss documentation.

## Meeting – 09/03/2001

**Time** – 2:00pm

**Location** – UKC Computer Science Meeting Room

**Group Members Present** – Ash Beeson, Tim Bishop, Alex Moore, Paul Mutton

**Absent** – none

This was a concise meeting to discuss the way forward over the next two weeks, specifically with regard to documentation.

Initially, Alex stated that both he and Tim had put a freeze on the Server and Client development, such that more time could be spent on documentation. There were still a few small issues that needed sorting, and of course bug fixes.

Tim discussed the problems with the server, although neither he nor anyone else really knew what was wrong with it. It was clear a memory leak existed, but where and how remained a mystery.

The main part of the meeting revolved around planning the documentation strategy. A list was put together of all the documents the group needed to produce. This list can be seen here;

[http://www.i-scream.org.uk/cgi-bin/docs.cgi?doc=plan/doc\\_strategy.txt](http://www.i-scream.org.uk/cgi-bin/docs.cgi?doc=plan/doc_strategy.txt)

Each group member should tackle any documents they feel they know best, and then the remaining documents will just be done by anyone. A template document, in word format, had been produced, and should be used for all documentation. A specification for documentation can be seen here:

[http://www.i-scream.org.uk/cgi-bin/docs.cgi?doc=specification/doc\\_spec.txt](http://www.i-scream.org.uk/cgi-bin/docs.cgi?doc=specification/doc_spec.txt)

It should be pointed out that Word format was chosen for because it allowed documents to be produced quickly. If time permitted more effort might have been taken to make documents more easily "transferable" to the web.

All documentation should be completed at the latest by the Friday before the deadline. This leaves almost a week to print and collate everything.

There will be a meeting on Monday at 2.15pm as usual, and one on Wednesday with Ian at 2.15pm.

## Meeting – 12/03/2001

**Time** – 2:00pm

**Location** – UKC Computer Science Meeting Room

**Group Members Present** – Ash Beeson, Tim Bishop, Alex Moore, Paul Mutton

**Absent** – none

Today's meeting again focussed on documentation. Paul reeled off a list of documents he had completed, and that were now available in cvs. These were mainly to do with the ihost and xml data layout.

Alex had made a start on the Conient docs, although there was still a way to go. Tim hadn't started on documentation, mainly because he was still trying to sort the server bugs. Ash was yet to start documenting due to assignments.

Documentation will be kept in cvs under documentation/printable. Time has ensured that .doc files will automatically be checked in as binary format.

It was again stressed that time was short, and that we needed to get moving on the documenting. Paul pointed out that the maintenance documentation took a lot longer for ihost than he imagined, and thought this would be the case elsewhere.

Aside from documentation, the webpages were discussed. Paul had, over the week, put together a php page that clearly laid out the data sent out by the server. This gives a virtually realtime view of the data coming through the system. This still needed tweaking, and linking with the historical side however. The alerting webpages were still to be done, although Tim pointed out that the server side was "ready" but exact formatting needed to be discussed.

The group left the meeting to quiz Ian about memory leak issues. He didn't, unfortunately, have a "magic wand" solution, but put forward a few suggestions. These will be looked in to.

There will be a meeting with Ian at 2.15pm on Wednesday, and the usual meeting on Monday at 2.15pm. There may be an extra meeting in the week if required.

## Meeting – 19/03/2001

**Time** – 2:00pm

**Location** – UKC Computer Science Meeting Room

**Group Members Present** – Ash Beeson, Tim Bishop, Alex Moore, Paul Mutton

**Absent** – none

The first part of the meeting was a usual "catch up" session. Each of the group members reported back on what they'd been doing, and where things stood.

Alex was proud to announce that Conient was now complete, or at least as complete as it was going to be for the project. There were of course many more features that could have been added, but time was restrictive. This has been the case with most things. The GUI had been reworking to provide a sorted list of hosts, which was much easier to navigate. A few bugs had been ironed out with the layout, and a general tidy up had been done.

Paul said the web reports were well in progress, maybe even finished. The pages for the "latest data" and "alerts" had been done, and linked together with the historical side of things. A summary alerts page had also been generated. The other group members were impressed by the webpages generally, but felt that a few more smaller things were required. Firstly a "one screen" summary of selected hosts, maybe for display on a "public monitor" near reception desks. Concern was also raised that the whole lot should be even more tightly integrated... so it appeared as one system, rather than a collection of three subpages.

Paul also added at this stage that winhost had been finished. It now reported uptime correctly, and the icon had been made blue (instead of pink). The few minor bugs that existed had been ironed out, and an installer had been made. Alex reported that it now worked on his system, and didn't give an OCX error as before.

Tim was pleased to inform the group that the server seemed to be fixed. He and Alex had spent a long night in the computer room debugging and fixing parts of the server. In the end it seems it was a deadlock (or livelock?) issue in the Queue. The system had been running for nearly 48 hours without any hassle and was looking good. Tim also added that the server was essentially complete, except for the LocalClient which still needed a few minor points ironing out with alerting.

It was also noted that statgrab and ihost were as good as finished, although a few minor issues still needed solving. These would be solved ASAP.

Ash informed the group that he had finally solved the 1024kb UDP packet size problem on the C++ host. It required changing the socket library, which was not satisfactory, but it at least worked. He also added that documentation on both the C++ host and the Java host had begun.

The group noted the urgency to complete documentation. With little over a week to go, and a lot to write, things were getting very close. As coding had virtually completed now, documentation could be done throughout the week. As luck would have it, everyone had a databases assignment due in midweek, which may take up valuable time.

There will be a meeting with Ian at 12pm on Thursday, and the usual meeting on Monday at 2.15pm.