

COMP4500 Final Report

AccessGrid Recording Services

Joseph Curtis(3952239)

Joe (Xianzheng Zhou)(3975416)

Quoc Le(3926021)

Rehan Walsh(3970757)

Chao Yan(3926755)

Viral Patel(4081509)

Alekh Aggarwal(3352818)

Colin Bernays(3232865)

October 21, 2005

Contents

1	Project Overview	3
1.1	Context	3
1.1.1	Original System	3
1.1.2	Problem With Original System	3
1.1.3	New System	3
1.2	Key Requirements	3
1.3	Key Deliverables	4
2	Performance	4
2.1	Key Requirements	4
2.1.1	Satisfied Requirements	4
2.1.2	Partially Satisfied Requirements	4
2.2	Key Deliverables	5
2.2.1	Deliverables	5
2.2.2	Not Delivered	5
2.3	Effort	5
3	Issues and Risks Faces	5
4	Lessons Learned	6
4.1	Individual Lessons Learned (Joseph)	6
4.1.1	What Went Right	6
4.1.2	What Went Wrong	6
4.1.3	Things That I Didn't Do	6
4.1.4	What I Would've Done Differently	6
4.2	Individual Lessons Learned (Joe)	6
4.2.1	Satisfaction	6
4.2.2	Regrets	6
4.2.3	Improvements Could Be Done	7
5	Project Record Discrepancies	7
5.1	Individual Record Discrepancies (Joseph)	7
5.2	Individual Record Discrepancies (Joe)	7
6	Conclusions	7
7	Recommended Next Steps	8

1 Project Overview

1.1 Context

1.1.1 Original System

- Prototype Remora server written in Java. Primitive telnet-based interface. Could only record and not play?
- File format which the original server recorded to, Specified in a document.
- Protocol which original server understood. Specified in same document.
- No client implemented for the original server.

1.1.2 Problem With Original System

- Was always intended to be a proof-of-concept prototype
- Written in Java, client much preferred for it to be written in C.
- Almost no security implemented in the original server.
- No client written to interface with the server. You have to manually enter commands through telnet interface.
- No post-production editing tools available.

1.1.3 New System

- Re-write Remora server in C. Porting would've been preferable however wasn't really possible.
- Improve on the original Remora design by adding in security. SSL for communication between the Remora Server and Client.
- New Remora server to record access grid sessions in the file-format that the original server used and is specified to us by the client.
- Implement a Client application to interface with the Remora Server for much simpler operation.
- Implement a post-production editing tool which can perform basic editing functions on pre-recorded access grid sessions stored in the same file format.
- Possibly look at conversion from the specified file format into standard video formats such as MPEG4. This was scoped out around half-way through the project due to technical difficulty and lack of time.

1.2 Key Requirements

- Remora server was to be re-written in C.
- A GUI remote control client would be implemented to control the server.
- The Remora server and client would communicate using a secure socket.
- Remora server would record and playback access grid sessions in the file format specified to us by the client.
- The Remora server would run on a POSIX compliant system connected to a multicast-enabled network.

- The Post Production Tool would also use the file format that is the same as the format recorded by Remora server.
- The Post Production Tool would help the users edit the file recorded by the remora server.

1.3 Key Deliverables

- Operational Concept Document
- Software Development Plan(Plan, Work Breakdown Structure)
- Software Requirements Specifications(Remora, Post-production Tool)
- Software Design Documents(Remora, Post-production Tool)
- Implementation(Remora-C Server and Client, Post-production Tool)
- Test Plan(Remora, Post-production Tool)
- Test Report(Remora, Post-production Tool)
- User Manual

2 Performance

2.1 Key Requirements

2.1.1 Satisfied Requirements

- The Remora has been re-written in C.
- The Remote control client has been implemented to control the server.
- The Remora server and remote control client can communicate using a secure socket.
- The Remora server would run on a POSIX compliant system connected to a multicast-enabled network.

2.1.2 Partially Satisfied Requirements

Remora Server Recorder The Remora server of new system can record and playback the AccessGrid session. However, since different computer hardware have different byte order configuration that the 16-bit and 32-bit integer data stored in the file have different representations so that the file recorded by the recorder vary depending on the hardware. As a result of this, the file recorded on the server that is a little-endian machine can not be playback on the server that is a big-endian machine. This problem is not solved. Because the Post-production Tools is implemented in JAVA while JAVA uses big-endian data representation, the Post-production Tools can only open the recorded file that is recorded by the remora server that is running on a big-endian machine.

Remote Control Client Playback The interfaces for controlling the remora server to schedule playback task is not properly implemented. This is due to the misunderstanding of requirements by the developer

2.2 Key Deliverables

2.2.1 Deliverables

Operation Concept Document One Operation Concept Document

Software Development Plan A Development Plan and a separately maintained Work Breakdown Structure.

Software Requirements Specifications A SRS document for Remora Server/Client and a SRS document for Post-production Tool.

Software Design Documents A SDD document for Remora including detailed design information extracted from the source codes by Javadoc. Same as the SDD document for Post-production Tool.

Implementation Remora server written in C; Remote Control client written in JAVA; Post-production Tool written in JAVA

Test Plan A Test Plan for testing how Remora server/client work and communicate. A Test Plan for testing the functionalities of Post-production Tool.

2.2.2 Not Delivered

Test Report Only testing has been done. No reports has been generated. Because we have been trying very hard to debug and make the recorder and player of remora server working. No time left for writing the test report.

User Manual Not delivered with the same reason as above

2.3 Effort

3 Issues and Risks Faces

Lack of Power We think managers do not have power to make final decisions. Sometimes, when some team members have different idea about a specific activity with team managers, they ignored the managers instructions if no agreement was made for that issue and made move on their own.

Lack of Communication The team lack of communication which results in ineffecient work initially. It was even very hard to make an agreement of a common meeting time with every team members. We managed to arrange work sessions, trying to have every team members to work together and help each other. This enhanced the productivity a lot.

Testing/Debugging for Too Long Time Technical difficulties engaged when implementing the recording and playback module of the remora server slows down the whole schedule of implementation because it is the core functionality of the remora application. Unfamiliarity with multicast network implementation and low-level file manipulation leads to low quality of initial outcome of implementation. As a result, testing and debugging that low quality peice of source code took much longer time than expected. Furthermore, this part can not be skipped because it is the core functionality of the remora application. This project won't make any sense without it.

Absent Team Member

4 Lessons Learned

4.1 Individual Lessons Learned (Joseph)

4.1.1 What Went Right

I feel that my contribution to the project as a whole was quite good. We have succeeded in providing value to our client and he is quite happy with the progress we have made.

4.1.2 What Went Wrong

We took quite a while to get going at the beginning of the project. During second term really not much happened. There was an extreme lack of communication at that time and that is reflected in the assessment of our progress by the Lecturers at that time. I also failed to create and follow a decent detailed plan towards the beginning of the year. There were high-level objectives such as create the SRS or code this prototype by this date however the details were often ad-hoc. I feel also that I really had no idea how to manage people at the beginning of the year. I would let people off through deadlines and was lax about checking results, This led to the prolonged development of Remora-C (which is only now complete). I underestimated the amount of effort it would require to develop this software. I also underestimated the coders I had assigned to develop. I had created a high-level plan and I left much of the detail to the individual developers. This was a mistake because a fundamental misunderstanding of how the system was to work (because of lack of communication) caused these people to fail completely at this. In the end this caused me to spend too much time coding myself and less time managing which in turn caused further problems.

4.1.3 Things That I Didn't Do

I didn't spend enough time managing people. I previously felt that I shouldn't have to check on people every five minutes to make sure that they do what they've been assigned. I know now that you have to assign tasks based on the capabilities of each individual. Carefully monitor them.

4.1.4 What I Would've Done Differently

If I could do it all again I would do many things differently. I feel that I have learnt a great deal in this project. I would start off with a much better plan. Monitor the plan regularly and chase after any discrepancies. I would also be more forceful with my third years. I felt that I let them off too easy and they began to perhaps take my recommendations of deadlines laxly.

4.2 Individual Lessons Learned (Joe)

4.2.1 Satisfaction

- Keep track the status of tasks and schedule new tasks.
- Make decisions to split the team into two sub-teams each of which works on a particular part of the project, and lead one of the team.
- Participating in most of the software development tasks.
- Learned a lot of multicasting network implementation techniques.

4.2.2 Regrets

- Managers shall act at a certain level of aggression, otherwise, the team members might not listen to you as they might think you are too nice.

- Since I am responsible for the recording/playback module of the remora server module, I have the experiences the worst case by now that how technical difficulties could affect a project.
- Though I made most of the plan of the project, I did not make precise measure of where the position of the project is. All I did was to roughly estimate where the position is so that the prediction or guess based on the estimation become more inaccurate. As a result of this, I experienced the worst case of management: “nothing goes as planned”
- Did not use gForge properly

4.2.3 Improvements Could Be Done

- I shall have addressed the technical issues of the project earlier to avoid the no-end debugging.
- I shall have arranged more work sessions to have all team members work together, which has been proven to be more productive so that more deliverables could be finished.
- I shall have kept making more accurate measure of the project position to help change the scope of plan and maintain the work breakdown structure.

5 Project Record Discrepancies

5.1 Individual Record Discrepancies (Joseph)

There were no significant discrepancies from GFORGE and reality because what I did manage to do was update it regularly after I had performed some work. Last year I got into no end of trouble with timesheets so I vowed this year to be much more vigilant.

There were some discrepancies with timing or tasks, starts and finishes however these didn't affect the work done.

5.2 Individual Record Discrepancies (Joe)

- Spent more hours than recorded in gForge. Because sometimes, after I closed the task, I realized that I should have put in the hours that I spent on studying the material and researching on the techniques for implementing the software. So the time I spent on reading and researching is not recorded in the gForge
- I am very sorry that I have a very bad habit when updating the records on gForge. I just realized it after receiving the email from Lynette. The problem is that I am used to update the hours recorded on the gForge once after a whole task has been finished rather than updating it for every time I did work. As a result of this, the descriptions for the big chunk of hours recorded is not clear. I am very sorry to have done this.

6 Conclusions

We performed perhaps less effectively than I had hoped. At the beginning of the project I did have high expectations. There were some major issues that we had to work through with progress in the middle of the year. I feel that we really pulled through in the end and made a pretty good job out of what could've been a disaster. It is a great pity that one of our team decided that they should no longer participate. That is one of the reasons I think that we were in such dire straits at one point. However our client is fairly happy with what we have produced and so are we.

7 Recommended Next Steps

- The server need to be improved and solve the byte order problem.
- The server need much better error tolerance capability because currently it is very easy to crash the server by sending it with unacceptable commands.
- The remote control client need more friendly user interface.