

Appendix A - Project Management

Development Process

During the first few weeks of our project's lifetime, no system requirements had been clearly identified. Only some general characteristics that the system should have, had been discussed with our supervisor. Some of those after thorough examination had been changed too. Due to the dynamic nature of the requirements, the group decided to follow the XP development process as it lends itself to this type of project.

XP's approach of small releases could assist us, since we could start with the smallest possible features first, in our case the communication between peers, and add any new features as they were identified and agreed upon. This would allow the product to be delivered in phases, also allowing our client to monitor the progress and provide additional information on the direction of the product. Thus, further development could be guided by the client's evaluation of the current phase.

Since the product was developed in steps, any changes were agreed to be committed in the project's repository regularly, thus following the *continuous integration* practice of XP. During each integration, appropriate tests were carried out, to make sure that no existing functionality had been broken.

One of the main risks identified during risk management, was the issue of the absence of a member. It would be catastrophic, if in such a case, the rest of the group was unable to take over the absent member's work. XP's practice of shared responsibilities seemed to be an excellent approach to adopt, as it would not require all members to be present for development to continue undisturbed. Thus, all tasks/duties had been allocated to a pair of colleagues so that, in case of an absence, the others can still carry on with the work.

Team organization

The team made the decision to follow the 'classic' structure with a single project manager, which was decided to be Chris. Due to the fact that the group members were friends, adopting a democratic team organization might seem more suitable. It was felt though that it would be more appropriate if there was a single person that would act as the point of contact with the supervisor and would be responsible for solving any conflicts within the team. However, it was recognised that all members were equal to each other and of similar capabilities. Thus, during all project phases, all disputes relating to the features of the system were solved by carrying a majority vote.

The team tried to make as much use of the mix of talents present within the group, as possible. Some members were better at programming than others, who in turn were better at documentation. Given the above, it was decided that Burhan would be responsible for the programming aspects of the VPFS tool and would be coding along with Chris and Jason, whereas Paris would be the team's secretary and responsible for all documentation. It was also decided that at the end of the coding phase, Chris would become more actively involved in the documentation, whereas Burhan and Jason would carry on testing. Finally Paris was designated the client liaison. It was felt necessary that all communication with the client be made with one voice. This person would therefore be documenting the client's application and needs as far as the VPFS tool was concerned.

Appendix B – Technology Used

In developing the VPFS system, the following set of tools were used:

J2SE 1.4.1

The Java 2 Standard Edition version 1.4.1 software development kit is used in the implementation of the VPFS system.

CVS

CVS (Concurrent Versions System) is a version control system, which enables the collaboration between group members during the implementation of a system. With CVS, a group can keep track of the changes made in the project and allows rolling back to previous versions of code.

Eclipse

The Eclipse development platform was the main development tool used by the group. It provides an integrated programming environment that eases the editing of code and building the project. Eclipse runs on multiple platforms which was very useful because the members of our group preferred to work under different platforms. Eclipse also provides an excellent debugging tool to help us quickly find the bugs in any code, thus saving us a lot of time.

Apache Ant

Apache Ant is a Java-based build tool, similar to the Unix *make* tool. Apache ant was utilised by the group, as it provided an easy way to build the whole project especially in machines which did not have sufficient processing capability to run Eclipse. In addition, it was used to generate JavaDoc API documentation.

Together

Together version 6 was used for creating the UML diagrams.

JXTA

JXTA version 2.0 was used in building the VPFS network. Its usage is introduced in detail in chapter 5.

XML

Extensible Markup language (XML) is an open standard syntax for describing and structuring data by the use of custom tags. It provides the portability of data as it is simply text which can be moved between various platforms, hence providing platform and language independence. It enables data interchange and provides interoperability between systems by providing a method for modelling document which involves creating a specification that lays out the rules for how a document can look and hence focuses on the standard structure of the document. It provides extensibility as new markup tags can be created and added as required.

Appendix C – Test Results

System Initialisation Tests

| Test No | Test condition | Expected results | Result |
|---------|---|--|--------|
| 1 | Normal test, the system starts up with the peer profile | All the service modules should be started, and wait for user to type command | Pass |
| 2 | System starts up without peer profile | The system should create a default peer profile. | Pass |
| 3 | System starts up with incomplete peer profile | System should provide error message and exit. | Pass |
| 4 | System starts up with incorrect values in peer profile | System should give error message and exit. | Pass |

User Shell Tests:

| Commands | Test description | Expected results | Result |
|--------------|---|---|--------|
| <i>help</i> | Try to get the information about the command that can be used for file manipulation | All the available commands that can be used by the user should be listed | Pass |
| <i>cd</i> | Try to enter a directory which already exists in VPFS system | The user can enter the directory they specified. The new directory name should be displayed in the user shell prompt | Pass |
| <i>ls</i> | Try to list all the contents in a directory | The user should be able to see all the files and subdirectories under that directory | Pass |
| <i>chmod</i> | Try to change the access permissions of a file or directory | The access permissions of that file should be changed and updated in the corresponding file inode. When doing <i>ls</i> , the new access permissions should be shown. | Pass |
| <i>mkdir</i> | Try to make a new directory | The new directory inode should be created in VFAT. When doing <i>ls</i> , the new directory should be shown. | Pass |

File Manipulation Service Tests

CASE 1: Command Testing

Put File.

| Test No | Test condition | Expected Results | Result |
|---------|---|--|--------|
| 1 | Normal test. Try to put a local file which exists, user has read access on it and the target file doesn't exist in VPFS system. | The source file should be replicated and stored on the several machines (peers), the new file inode should created | Pass. |
| 2 | Try to put a file which doesn't exist locally onto the VPFS system | The system should give an error message that the source file does not exist | Pass |
| 3 | Try to put a local file which the user does not have read access to. | The system should give an error message that the user does not have read access on the source file | Pass |
| 4 | Try to put a local directory onto the VPFS system | The system should give an error message that a directory can not be put onto the system | Pass |
| 5 | Try to put a file onto VPFS system, where the target file already exists. | The system should give a message asking whether the user wants to overwrite the existing file. If yes, the existing file should be overwritten by the new file otherwise, the system should stop processing the command. | Pass |
| 6 | Try to put a file onto VPFS system where the destination directory does not exist. | The system should give an error message that the destination does not exist. | Pass |
| 7 | Try to put a file onto VPFS system where the destination is a directory | The system should put that file in the destination directory with the same file name as the source. | Pass |
| 8 | Try to put a file with the specified replication level, which exceeds the number of running peers. | The system should give an error message that the system can not find sufficient number of data stores. | Pass |

Get File.

| Test No | Test condition | Expected results | Result |
|---------|--|--|---|
| 1 | Normal test. Try to get a file from VPFS system, where the source file exists and the destination file does not exist. | The file should be obtained from one of the data stores and stored on the local file system. | Pass |
| 2 | Try to get a file which does not exist on the VPFS system. | The system should give an error message that the specified file does not exist. | Pass |
| 3 | Try to get a directory from the VPFS system. | The system should give an error message that the specified source path name is a directory. | Pass |
| 4 | Try to get a file to a destination that already exists on the local file system. | The system should give a message asking the user whether to overwrite. If yes, overwrite else the system should stop processing the command. | Pass |
| 5 | Try to get a file to the destination which is a directory on the local system | The file should be stored in the specified directory with the same name as the source file. | Pass |
| 6 | Try to get a file which has no replicas available. | The system should give an error message that the file cannot be retrieved at that moment. | Pass |
| 7 | Try to get a file where one of the replicas has changed. | The system should automatically get that file from another peer that holds a replica. This is transparent to the user. | Fail. The system gets the file randomly from one of the file stores. |
| 8 | Try to get a file to a destination directory which does not exist. | The system should give an error message that the destination does not exist. | Pass |

Delete File.

| Test No | Test condition | Expected results | Result |
|---------|---|--|--------|
| 1 | Normal test. Try to delete a file that exists in the VPFS system. | All the replicas of that file should be completely deleted; the inode should also be deleted. | Pass |
| 2 | Try to delete a file that does not exist in the VPFS system. | The system should give an error message that the specified file does not exist. | Pass |
| 3 | Try to delete a file which has no replicas in the VPFS system. | The system should delete the inode of that file. | Pass |
| 4 | Try to delete a directory recursively. | The system should delete that directory and all the files in it recursively. All the inodes should also be deleted | Pass |

CASE 2: Concurrency testing

| Test No | Test condition | Expected results | Result |
|---------|--|---|--------|
| 1 | Two users try to delete the same file from VPFS system at the same time. | One user should be given a success message and the other user should be given an error message. | Pass. |
| 2 | Two users try to put a file to the same destination in the VPFS system at the same time. | One user should be given a success message and the other user should be given an error message. | Pass |
| 3 | Two users try to get the same file from VPFS system at the same time | Both users should be able to get the file successfully. | Pass |

Platform Independence Tests

This test was carried out using testing environment B.

| Test No | Test condition | Expected Results | Result |
|---------|--|--|--------|
| 1 | Three machines with different architectures and operating systems run the VPFS system. | There should be no interoperability issues. All services should run correctly. | Pass |

Wide Area Network Tests

This test was carried out using testing environment C.

| Test No | Test condition | Expected Results | Result |
|---------|--|--|--------|
| 1 | Two machines running the VPFS system, one inside UCL, the other outside. | The external machine should be able to use all the VPFS services provided by the internal one. | Fail |

Appendix D – Risk Management

Technology

| <i>RISK</i> | <i>Probability of occurrence</i> | <i>Impact on project</i> | <i>Risk Management</i> |
|--|----------------------------------|--------------------------|--|
| Chosen framework technology not understandable by group | Medium | High | Adequate research and communication channels with supervisor and other member of staff |
| Platform chosen may not provide the required functionality | Medium | Medium / high | Adequate research and communication channels with supervisor and other member of staff |
| Integration problems of system components | Medium | High | Perform integration tests at the end of each iteration |
| Programming language may lack features available in other languages | Medium | High | Adequate research and communication channels with supervisor and other member of staff |
| Programming language chosen may slow progress(due to unfamiliarity) | Medium | Medium | Members of the group most familiar with the language take the responsibility of familiarising other members with that language |
| Unable to establish an adequate P2P infrastructure - | Low | High | Adequate research and communication channels with supervisor and other member of staff |
| System not being reliable (crashing) | Low | High | Unit testing, integration testing |
| Not being able to implement any policies | Low | Medium | Carefully design policies(not too ambitious) |
| Time wasted recreating existing functionality(“reinventing the wheel”) | Low | Medium | Adequate research and communication channels with supervisor and other member of staff |
| Misunderstanding of project requirements | Low / Medium | Low / Medium | Adequate meetings and documentation between group members as well as supervisor |
| Underestimating implementation duration of components | High | Medium | Communication channels with supervisor and other member of staff, learn from our mistakes |
| Too many requirements – unable to implement all of them | High | Medium | Prioritize requirements |
| Unintuitive user interface | Low / medium | Low / medium | Periodic evaluation of interface by external people |

Management

| <i>Risk</i> | <i>Probability of occurrence</i> | <i>Impact on project</i> | <i>Risk Management</i> |
|--|---|---------------------------------|---|
| Dead lines not met | Medium | High | Carefully planning milestones and revising the plan |
| Serious conflicts between team members | Low | High | Escalate problem to project manager/supervisor |
| Team members' long absence | Medium | Medium/ high | Shared responsibilities |
| Supervisor unable to continue with project | Low | Medium | Out of control |
| Lack of communication | Low | Medium | Adequate documentation |
| Inadequate documentation | Low | High | Reviews off all documentation |
| Shared responsibilities for tasks may become imbalanced | Low | Medium | Escalate the risk to project manager/supervisor |
| Allocation of Responsibilities may not be suitable for specific people | Low | Medium | Reviews of all responsibilities at regular intervals. Also rotation of responsibilities |
| Evaluation methods may not be adequate | Medium | Medium | Communication channels with supervisor plus reviewing the evaluation |

Appendix E – Sample XML Documents

Peer Profile XML Document

```
<?xml version="1.0" encoding="UTF-8"?>
<PeerProfile Default="False" Group="VPFSGroup" Peer="VPFSPeer">
  <FileManipulationService ProvidesService="True">
    <VPFSHome>./FileStore</VPFSHome>
    <TotalSpace>104857600</TotalSpace>
    <UsedSpace>28524</UsedSpace>
  </FileManipulationService>
  <UserProfileService ProvidesService="True">
    <MaxFragSize>10</MaxFragSize>
    <MinFragSize>5</MinFragSize>
    <XMLFilePath>UserProfiles.xml</XMLFilePath>
  </UserProfileService>
  <VFATService ProvidesService="True">
    <MaxFragSize>10</MaxFragSize>
    <MinFragSize>5</MinFragSize>
    <XMLFilePath>VFAT.xml</XMLFilePath>
  </VFATService>
  <PolicyService ProvidesService="True"/>
  <UserService ProvidesService="True"/>
</PeerProfile>
```

User Profile XML Document

```
<?xml version="1.0" encoding="UTF-8"?>
<UPM Peer="">
  <Group Name="VPFSGroup">
    <ParentPathName></ParentPathName>
    <TotalProfilesHeld>0</TotalProfilesHeld>
    <IsLocal>true</IsLocal>
    <User Name="admin">
      <Password>password</Password>
      <FullName>Root Administrator</FullName>
      <ParentPathName>/VPFSGroup</ParentPathName>
      <HomeDir>/VPFSGroup</HomeDir>
      <IsAdmin>true</IsAdmin>
    </User>
  </Group Name="Users">
    <ParentPathName>/VPFSGroup</ParentPathName>
    <TotalProfilesHeld>0</TotalProfilesHeld>
    <IsLocal>true</IsLocal>
    <User Name="john">
      <Password>abc123</Password>
      <FullName>John Doe</FullName>
      <ParentPathName>/VPFSGroup/Users</ParentPathName>
      <HomeDir>/VPFSGroup/john</HomeDir>
      <IsAdmin>false</IsAdmin>
    </User>
  </Group>
</UPM>
```

VFAT XML Document

```
<?xml version="1.0" encoding="UTF-8"?>
<VFAT Peer="">
  <Inode Name="VPFSGroup" Pathname="/" Type="Dir">
    <Owner>/VPFSGroup/admin</Owner>
    <Permissions>110000</Permissions>
    <Group>/VPFSGroup</Group>
    <Replication>1</Replication>
    <TotalNoOfFiles>3</TotalNoOfFiles>
    <NoOfFiles>0</NoOfFiles>
    <IsRoot>True</IsRoot>
    <IsLocal>True</IsLocal>
    <Inode Name="documentation" Pathname="/VPFSGroup/" Type="Dir">
      <Owner>/VPFSGroup/admin</Owner>
      <Permissions>110000</Permissions>
      <Group>/VPFSGroup</Group>
      <Replication>1</Replication>
      <TotalNoOfFiles>0</TotalNoOfFiles>
      <NoOfFiles>0</NoOfFiles>
      <IsRoot>False</IsRoot>
      <IsLocal>True</IsLocal>
    </Inode>
    <Inode Name="code" Pathname="/VPFSGroup/" Type="Dir">
      <Owner>/VPFSGroup/admin</Owner>
      <Permissions>110000</Permissions>
      <Group>/VPFSGroup</Group>
      <Replication>1</Replication>
      <TotalNoOfFiles>1</TotalNoOfFiles>
      <NoOfFiles>1</NoOfFiles>
      <IsRoot>False</IsRoot>
      <IsLocal>True</IsLocal>
      <Inode Name="Test.java" Pathname="/VPFSGroup/code/"
Type="File">
        <Owner>/VPFSGroup/admin</Owner>
        <Group>/VPFSGroup</Group>
        <Permissions>110000</Permissions>
        <Replication>1</Replication>
        <Replica>urn:jxta:uuid-
59616261646162614A787461503250334344AEF057B149CFBD81B50E976C9A0E03</Replica
>
        <FileSize>7131</FileSize>
        <Hash>7aae2c9028687db837afc9bc4e305a4b</Hash>
        <WriteLock>False</WriteLock>
        <Created>Thu Aug 28 17:19:53 GMT 2003</Created>
        <LastModified>Thu Aug 28 17:19:53 GMT 2003</LastModified>
      </Inode>
    </Inode>
  </Inode>
</VFAT>
```