

# **New Practices in Flexible Learning**

## **Groove Educational Tool Suite for Peer-to-Peer Collaborative Learning Environments**

**Project report**

**David Said**

**Gautam B Ramamurthy**

©2003 Australian National Training Authority

This work has been produced with the assistance of funding provided by the Commonwealth Government through the Australian National Training Authority. Copyright for this document vests in ANTA. ANTA will allow free use of the material so long as ANTA's interest is acknowledged and the use is not for profit.

The views and opinions expressed in this document are those of the authors and do not necessarily reflect the views of ANTA. ANTA does not give any warranty or accept any liability in relation to the content of this document.

ISBN 1 920906 61 4 web edition

*An initiative within the Australian Flexible Learning Framework for the National Vocational  
Education and Training System 2000-2004*

*Managed by the Flexible Learning Advisory Group on behalf of the Commonwealth, all States and  
Territories in conjunction with ANTA*



## Table of Contents

Acknowledgements .....	iv
1. Executive summary .....	1
2. Introduction .....	2
3 Collaborative educational tools .....	3
4 Peer-to-Peer technology using Groove .....	4
5 Project scope and sequence .....	5
6. Design Architecture .....	7
7. Tool development phases .....	13
7.1 Definitions .....	13
7.2 Design .....	15
7.3 Development .....	18
7.4 Testing .....	26
7.5 Implementation.....	28
9. Future recommendations .....	31

## Appendices

Appendix A - Use case diagrams .....	32
Appendix B - Database schema .....	35
Appendix C - Functional test cases .....	36

## **Acknowledgements**

### **Mr Steve Fisher**

Project Manager  
Open Learning Institute  
TAFE, Brisbane  
Queensland

### **Mr Tony Sahama**

University Supervisor  
School of Info. Systems  
Queensland University of Technology

### **Mr Graham Willett**

Coordinator  
CEED Program  
Queensland University of Technology

### **Mr David Said**

Project Team Member  
Queensland University of Technology

### **Mr Gautam B Ramamurthy**

Project Team Member  
Queensland University of Technology

# 1 Executive summary

The project 'Educational tool suite for Peer-to-Peer Collaboration Learning Environments' was based on the idea of developing a collaborative-based e-learning tool for the Australian Flexible Learning Framework – a government funded program that supports Flexible Learning Opportunities (<http://www.flexiblelearning.net.au>). This tool suite is designed and developed to be hosted on the Australian Flexible Learning Framework website from where educators across Australia can download and use the suite in their e-learning programs.

There are currently very few tools available for effective collaborative e-learning processes. The *RolePlayControlTab* tool developed by this project team can be considered to be a first step towards an effective collaborative e-learning tool. This tool suite has been developed on the Groove Platform using Microsoft's .NET platform as the coding language. It has been designed specifically to facilitate the common activities that need to be performed within a collaborative learning environment such as dialogue, involvement, support and control.

This project was conducted efficiently in various phases by the project team, implemented by Gautam B Ramamurthy and David Said with the additional help of a team of dedicated educationists and domain specialists across Australia and around the world. The tool was developed using an alternative approach with the development and testing of tools using short timelines. The main phases involved in the development of the tool were: Defining, Designing, Development, Testing and Implementation. Sufficient time was spent on each of these phases during implementation. In so doing the quality of the development process was not compromised.

The *RolePlayControlTab* tool is accompanied by two manuals. The first manual is a Technical Specification Manual, which defines the technical details involved with the development and also the installation of the tool. The second manual is the much needed User Manual which describes in detail the various steps to be followed during installation including the steps required to use the various functionalities provided within the tool.

## 2 Introduction

Within the education and training sectors, the growing importance of collaborative learning (communities of practice) is becoming well recognised. Most existing online learning platforms and software systems consist of a centralised web-based model, with content systems and learning management systems residing on servers. Content is centrally created, stored and delivered in an e-version classroom (the 'virtual classroom'). Online teachers are restricted to html 'page' transmissions that contain a limited amount of embedded interactivity and that required some form of publishing process to create (from html/multimedia programming to using html editors). Communication is limited to bulletin boards and email. While this situation has served online learning 'early adopters' well, it is now proving to be too restrictive. Consequently, there is a need to change rather than perpetuate the current institutional paradigm.

Peer-to-Peer online environments using specific educational tool-sets changes this paradigm and allows the teacher to create content without the need for servers and without publishing instantly for learners. It allows content to be created by all users in real time (synchronously) as well as asynchronously. It allows interactivity based on human communication and socialisation rather than on software functionality. Through using simpler, more secure and lower cost new-age technologies, the collaborative e-learning tool developed by this project has enabled the education practitioner and learner, in ways not previously possible, to access the above-mentioned benefits of using educational tool sets that are specific to Peer-to-Peer online environments. The *RolePlayControlTab* e-learning tool developed by this project was developed using Groove Workspace (<http://www.groove.net>) and VB.NET development platform.

This document highlights the various steps involved in the design and development of the tool throughout the project. It includes details of the implementation phases, the use-case scenarios, the database schema used and test cases used during the development of the *RolePlayControlTab* tool. This document also includes administrative details such as reporting mechanisms, problems faced and future recommendations for improving the tool.

### 3 Collaborative educational tools

In today's world, rapid technology growth, coupled with the urge to learn more has made education through e-learning the most talked about concept in academia. But delivering all the advantages of a face-to-face interactive approach through an electronic format has always proved challenging. The concept of collaborative learning adds a complete new dimension to the meeting of this challenge by providing an instruction medium, that enables students of differing abilities to work together in small but balanced groups in the pursuit of a common goal.

Technologies used to-date by educators to assist learning include word processing, slide presentations on data projectors or overhead projectors, email, bespoke multimedia programs and enterprise Learning Management Systems (LMS). Currently, 'educational tools' are merely embedded applications used within larger applications. For example, LMS applications such as like Groove, WebCT and BlackBoard which have a set of default 'tools' like bulletin boards, glossaries, chat and webmail, are often used by third party software developers, in collaboration with LMS vendors, to integrate their products into LMSs.

Collaborative e-learning can be best defined as a process of learning in which Information and Communications Technology (ICT) is used to promote connections between one learner and other learners, between learners and tutors and between a learning community and its learning resources. Thus through granting of authority to make decisions, collaborative learning helps the user to make good judgements and to develop a bond of trust with other users.

Coomey & Stephenson have identified four common activities in online learning: Dialogue, Involvement, Support and Control. These activities formed the basis of this project using the *RolePlayControlTab* control tool developed by this project. Outlined below is a summary description of each of these four key online learning activities.

#### **Dialogue**

Dialogue is often expressed as synchronous and asynchronous communication. Common online facilities include email, Listsers, discussion forums (bulletin boards), instant messaging, chat, video and audio conferencing.

#### **Involvement**

Involvement is synonymous with interaction. Researchers talk about the concept of 'flow' being a state of total absorption by the student in online learning activities. Flow is associated with challenge, clear feedback, learner control and concentration.

#### **Support**

Support is one of the more frequently raised issues in online learning. Support can include online supervision, support from peers, feedback and advice from tutors and experts as well as more direct support services in administration and technical support.

## Control

Control refers to the degree to which learners have control over key learning activities and the extent to which the learner is encouraged to exercise that control.

## 4 Peer-to-Peer technology using Groove

### What is Peer-to-Peer technology?

Peer-to-Peer is a communications model in which all parties have the same capabilities including the ability to initiate a communication session. The use of P2P technology for work collaboration is in harmony with its use in teaching and learning. In teaching and learning there is a constant need for relevant, credible and up-to-date information. Thereby, the role of web-based systems, content managements systems with reusable learning objects and other resource repositories are valid. However, the interaction of teaching and learning is more than just content and information dissemination. It is a dialogue, a discourse. It is story-telling and swapping experiences. It is questioning and answering, exploring and investigating. If the interaction of teaching and learning is all these things, then the relevance of Peer-to-Peer systems that enable such activities to be undertaken by teachers and learners alike is clear.

### Groove Workspace

Groove (<http://www.groove.net>) is a collaborative software package that relatively effortlessly keeps people, information and PCs 'in sync'. It essentially provides a platform for people to work together. Groove Workspace accelerates business activity amongst small groups both within and across organisational boundaries. Teams interact and exchange information in shared spaces where

content and communication are in context, secure, and available online or offline. With its quick access to shared information, Groove workspace enables team members to shave time from project schedules and reduce coordination costs.

In contrast to document-centric systems, Groove focuses more on the process of collaboration rather than just on the provision of collaborative documents. It assumes a decentralised shared workspace context and delivers collaboration and powerful off-line/resynchronisation capabilities by asynchronously disseminating actions within the shared context instead of replicating documents. This is certainly not a right-wrong, either-or scenario. Groove was designed to address differing customer needs to those driving the development of document-centric systems and architectural differences as complementary rather than competitive.

## **5 Project scope and sequence**

The project involved the development of a collaborative e-learning tool (using Groove Workspace) that facilitates the delivery of online educational courses through the provision of suitable dialogue, involvement, support and control facilities.

In order to deliver these facilities, the metaphor of a role-play was selected as the basis for the collaborative e-learning tool.

Some of the advantages of online role-play include:

### **1. It is in line with adult learning principles**

Adults bring their life-experience to a learning environment. In a traditional classroom setting it is up to the teacher to take advantage of this breadth of experience. Unfortunately, many teachers miss this resource. Online role-play explicitly taps into these rich sources of experience. When an individual 'plays out' a role that differs from their everyday viewpoint, new insight is developed through stepping into another's shoes. Additionally, when an individual interacts with others who play the same role as that individual, yet from a differing base of life-experience, that individual is able to see alternative views of the same stakeholder role.

### **2. Cost effectiveness**

Online role-play is a cost effective strategy due to a minimum of graphics being needed to convey the context and interactivity. There is no need to create sophisticated avatars (as in virtual reality or VR). Instead, learners actively and collaboratively participate in constructing a simulation using text-based communication.

### **3. Ample time to learn and reflect**

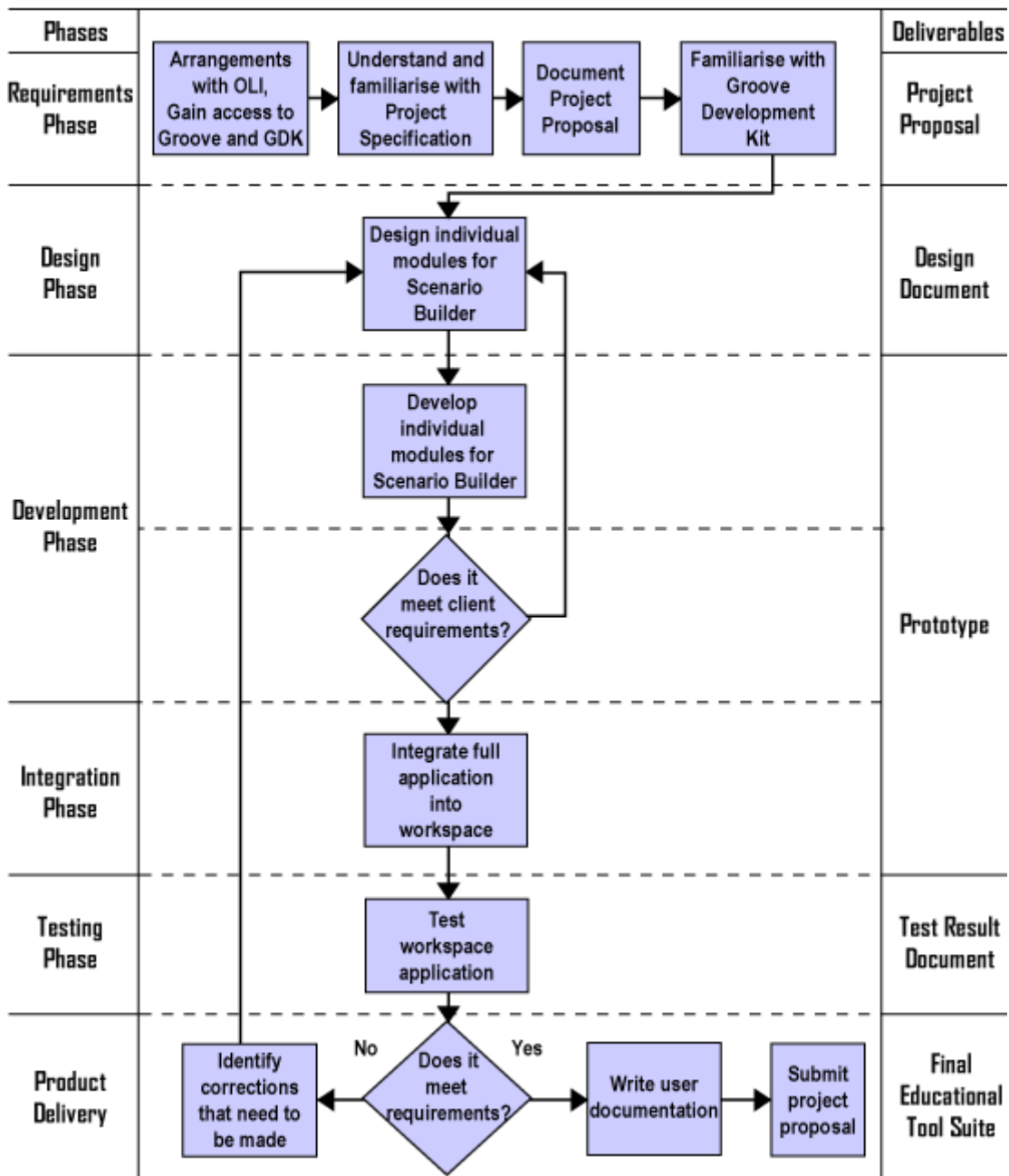
The role-play tool provides both synchronous and asynchronous modes of communication. The asynchronous mode provides ample time for players to research and consider alternatives using either 'in' or 'out-of-simulation' collaborative discussions before making a 'move'.

### 4. Anonymity

Online role-play overcomes the problem of playing a face-to-face role-play, where participants may feel shy about playing out a role. Anonymity allows for the free exchange of opinion without the risks associated with a face-to-face environment. While online role-play is not a ‘silver bullet’ for all types of learning; the engaging nature of online role-play certainly opens up new opportunities within adult learning.

The sequence for development of the *RolePlayControlTab* tool is illustrated by the following flow chart.

## Role Play Tool



## 6 Design Architecture

### Overview

The *RolePlayControlTab* tool design was based on the metaphor of an actual theatre play. The nomenclature included a Director, Characters and Analyser. Use of the tool is controlled by the space manager (teacher) who can build a cast of characters, assign a Director (or default to themselves as Director) and then develop the plot.

Space participants (students) are then assigned a character by the Director and given a public profile (seen by the Analyser and other characters) and a private profile known only to the Director and the character themselves. This private profile (released at the end of the play) makes it possible to covert instructions to pass from the Director to the character or for a character's hidden agenda to be created. Characters are able to view their profiles and to keep a private journal where they can record their feelings and reflections during the play.

The Director constructs the plot by first completing a text prologue to 'set the scene'. Either a single scene or a number of scenes are then created with the latter allowing for the plot to be progressively revealed in episodes that unfold dynamically as in real-life. The actual play consists of a chat panel contributed to by characters whose identity remains anonymous until the end. The director can release the scenes at their choice and the full play is shown as a cumulative chat with interspersed scene descriptors/dividers.

Finally, the Director closes the play and releases the 'Analyser'. The Analyser enables all play participants to review the play chat and to see private profiles and character identities. A small chat area allows discussion between the groups about the play.

### Functional scenarios

The *RolePlayControlTab* educational tool provides for the following functional scenarios:

- 'Character Editor'
- 'Scene Builder'
- 'My Character'
- 'The Play'
- 'Analyser'

The design for each scenario is as depicted in the screen captures following:

## 1 Character Editor

The 'Character Editor' tab allows the manager of the workspace to create characters for the role-play and assign the character to one of the members of the workspace.

### Options provided:

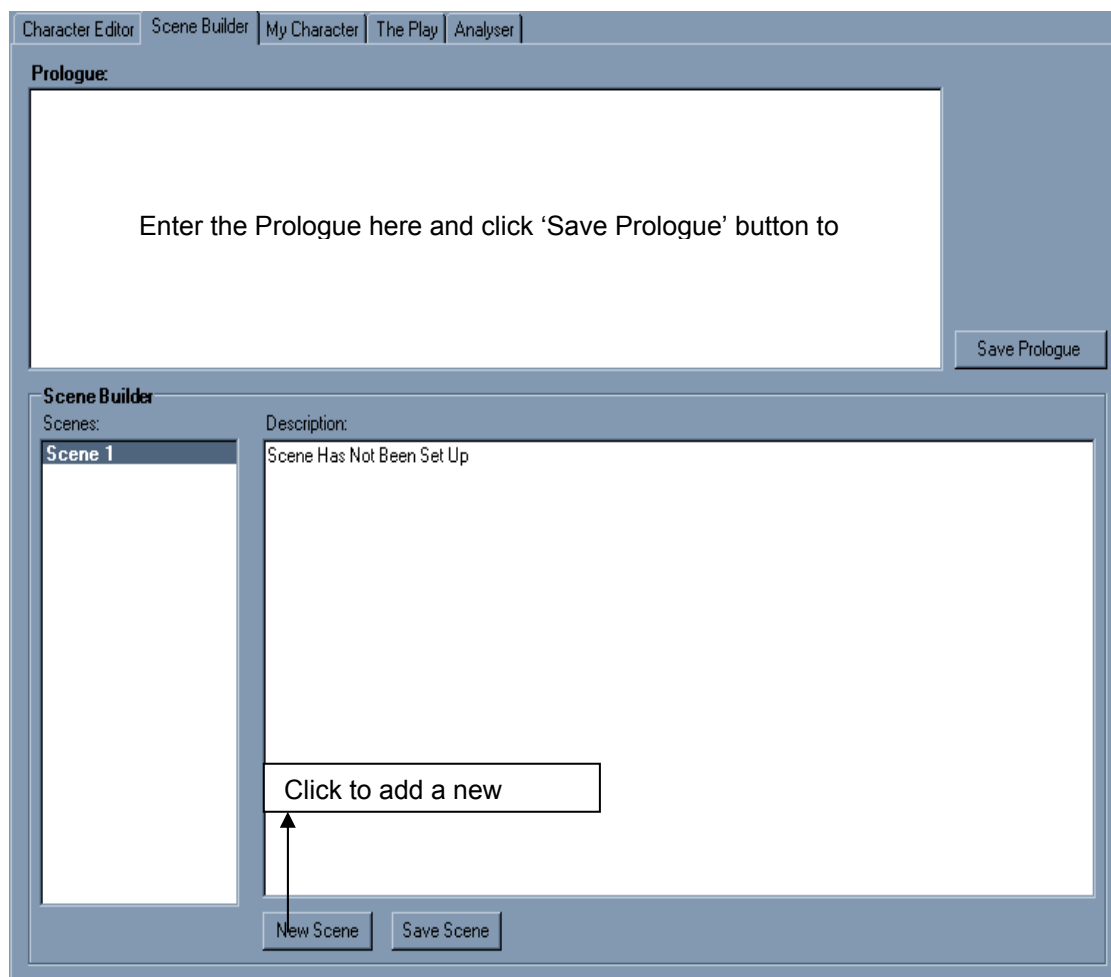
- Add Character – Adds the character to the database and displays it on the characters list
- Delete Character – Deletes the selected character
- New Character – Clears all the fields for the addition of a new character
- Update – Updates the database with the modifications to the character
- Clear – Clears all entries without saving

### Types of Character Roles:

- Director
- Player
- Analyser

## 2. Scene Builder

The 'Scene Builder' tab is used to help the manager plan out the overall role-play and the specific scenes that will be carried out in the play.



### Options available for Director of workspace:

- Save Prologue
- New Scene
- Save Scene

### 3 My Character

The 'My Character' tab is used by the user assigned to a particular character to view and update their journal entries.

Character Editor | Scene Builder | My Character | The Play | Analyser

**Edit Character**

Character Name:

Character Role:

Public Profile:

Private Profile:

My Journal:

Read-only fields

Click to update the modified character Journal.

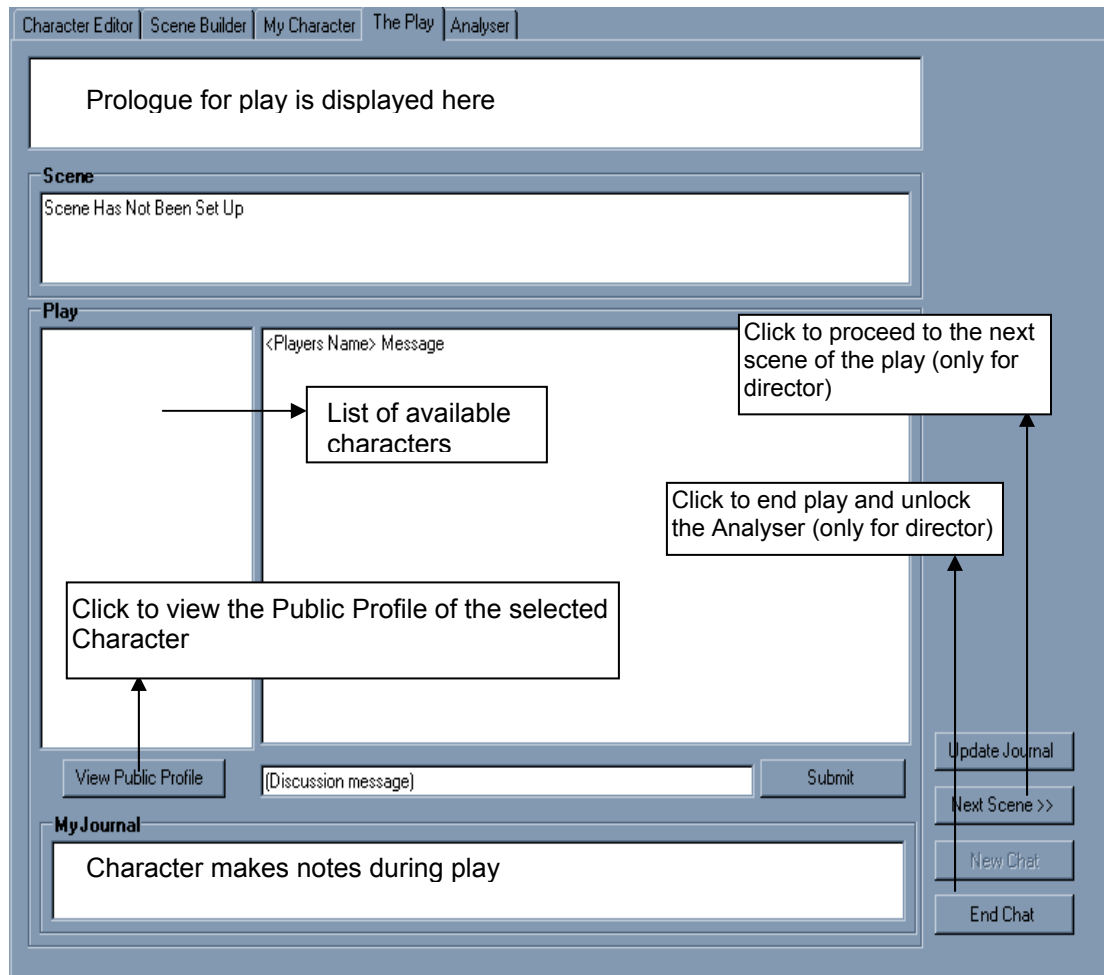
Update MyProfile

#### Options available for the User:

- Update Profile

## 4 The Play

The 'Play' tab is where the actual role-play is carried out. Participants/characters join together and act out the scenes.

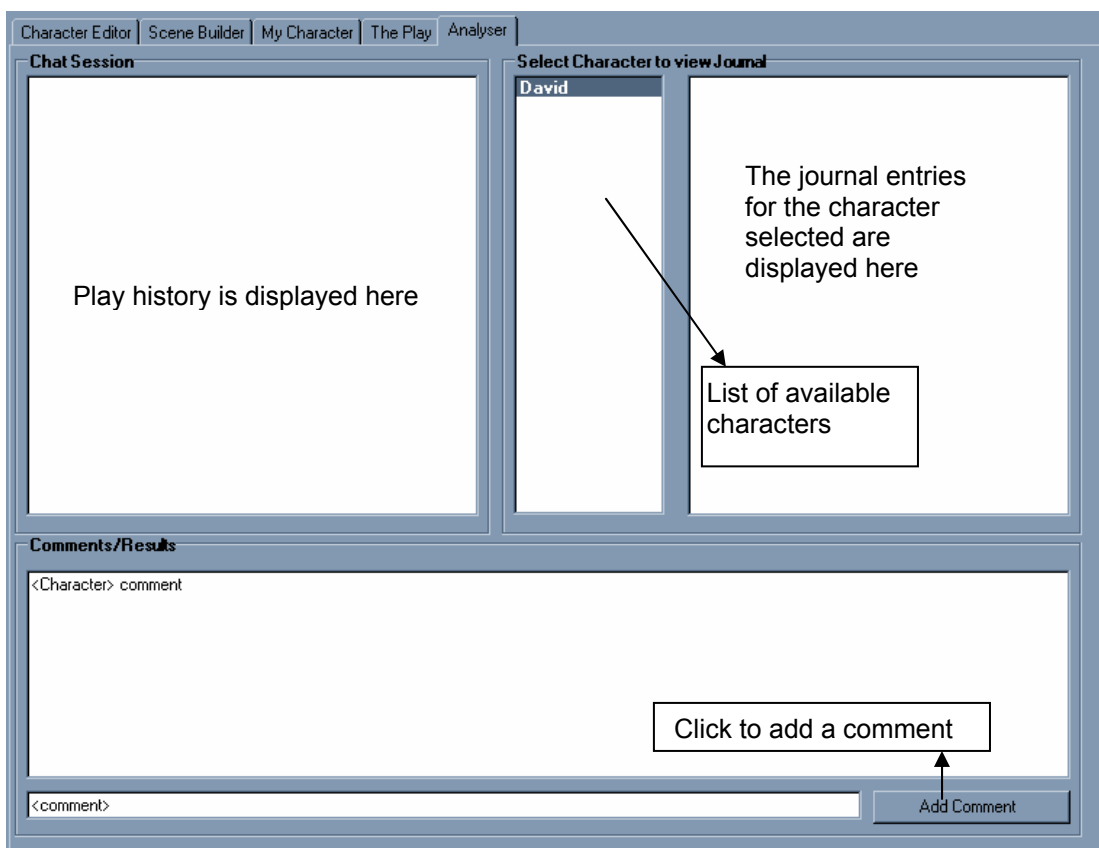


### Options available:

- View Public Profile (to all users)
- Update Journal (to all users)
- Next Scene (only to Director)
- End Chat (only to Director)

## 5 Analyser

The 'Analyser' tab is used when the role-play has been completed and all the users gather to analyse and review the role-play through providing comments.



### Options provided:

- Add comment

## 7 Tool development phases

### 7.1 Definitions

#### What are educational tools?

For some time now educators have used a range of technologies to educate students. These technologies have included word processors, email and Learning Management Systems (LMS). Educational tools are specific applications that are used within larger, more general applications that are dedicated to the domain of education. For example, applications such as WebCT and BlackBoard offer a large array of tools predefined within their existing systems. Yet they also allow other third party developers access to parts of the system so that these developers can integrate their own software within their particular Learning Management System.

#### Groove and educational tools?

Peer to Peer (P2P), although not a new technology, has recently been adopted by several companies as their core networking method. Companies such as Groove Networks have developed Groove Workspace, a platform that provides tools that can be used to carry out business activities online.

The use of Groove as a P2P platform offers the following benefits:

- Online collaboration  
Users work together online as they would anywhere else. The tools provided by Groove substitute real world items (eg the calendar and project planning tool).
- Peer to Peer technology  
The workspace is always online even if there are not any users presently online. Consequently, users are more free to come and go as they choose.
- Inexpensive means of communication  
A P2P network does not require an expensive server to function.
- Quick and simple user interface  
A lot of effort has gone into the Groove workspace interface to make sure that the application itself is intuitive and manageable.
- Instant messaging and Voice Chat  
By offering instant messaging services on top of the Groove application, several effective methods of communication that are familiar to most users of online applications has been provided.
- Data viewed online is always up to date  
When one user makes changes to any part of the Workspace the computers of all other users are also updated.

The default tools supplied by Groove are not limited to use within a business environment alone. Generally items such as the Forum Discussion, PinBoard and Calendar tool can all be used equally well within an educational environment. However, it is the ability to inject third party tools into Groove

Workspace that has also made Groove Workspace an ideal platform for development of online educational tools.

Based on the above-mentioned criteria for an effective educational tool, combined with the platform functionalities, offered by Groove Workspace, this project identified the following *RolePlayControlTab* tool requirements.

### **RolePlayControlTab tool requirements**

The following *RolePlayControlTab* tool requirements were based on the working design of a real life play. There is always a Director, Actors and an Audience.

1. When the tool starts it will allow the workspace manager to:
  - build a cast of characters and assign them to current members of the workspace
  - assign the role of Director to any of the space members (the manager) including the manager if desired
  - build the prologue to set the role-play, and
  - build the scenes required to support the prologue.
2. The members of the workspace who have been assigned characters in the role-play will have the following attributes:
  - be assigned a character by the Director or workspace manager
  - receive a public (viewable by everyone) and a private profile (only viewable by the player themselves or the director/workspace manager)
  - only have their private profile revealed to the other characters when the play has been finalised
  - have the ability to play out a hidden agenda, and
  - have the faculty for maintaining a private journal where their thoughts and feelings can be recorded as the role-play is carried out.
3. During the role-play the following conditions will be required:
  - the role-play is carried out in a chat panel and is contributed to by characters whose identity remains anonymous until the end of the role-play
  - there is the ability to break the plot into episodes that can be released when the director chooses so that elements can unfold dynamically as in real life
  - there is the option of showing the full play as a cumulative chat with interspersed scene descriptors/dividers
  - the Director is able to close the play and release the “Analyser” which enables all play participants to review the play chat, see private profiles and character identities, and
  - a small chat area that allows discussion between the members of the group about the play is provided.

NOTE: Refer to Appendix A for a complete representation of use-cases.

## 7.2 Design

Initially the *RolePlayControlTab* tool was to be composed of several tools with each tool performing a pre-defined task. The tools that were initially designed to setup and execute the role-play were:

- The Character Editor tool

The 'Character Editor' tool allows the manager of the role-play to create characters and to assign them to members within the Groove Workspace. As shown in the next section, the player can be assigned a character (with a public and private profile) and a role within the play that dictates what the player can do within the tool itself.

- The Scene Builder tool

The 'Scene Builder' tool allows the director or manager to build the role-play that is going to be carried out by the players. It allows them to add a prologue to the role-play scenario and add the required number of scenes.

- The My Character tool

The 'My Character' tool allows a player to view the details that have been assigned to their character. The player alone will be the only person able to update their journals in this section. The player will not however, be able to update the character details that have been assigned to them by a director or the manager of the workspace.

- The Play tool

The 'Play' tool is where the role-play will be carried out. It contains a list of the current registered players, the chat session and provide the ability for the player to update and view their journal as the role-play progresses.

- The Analyser tool

The 'Analyser' tool is unlocked when the role-play has been completed. In this tool all players, directors and viewers can come together to comment on how they think the role-play went and review the chat session and the other users' journals.

However this initial design was modified after the project team experienced great difficulty in getting the tools to communicate with one another.

The modified design incorporated all of the proposed tools into one single tool with the use of tabs to separate the different parts of the application. Although combining the proposed separate tools into one single tool simplified the installation process of the tool, it required a lot more work to make sure the interface was uncluttered and usable.

## User groups

Once the tools were defined, it was necessary to then define what the users would be able to access within the tool.

Below is a table of what the user can gain access to within the tool:

User Role	Access within the tool				
	Character Editor	Scene Builder	My Character	The Play	Analyser
Director	Yes	Yes	Yes	Yes	Yes
Player	No	No	Yes	Yes	Yes
Audience	No	No	No	Yes	Yes

Once user access within the tool was defined, it was necessary to then define what could and could not be done by the user within the role-play.

Below is a table displaying interaction with the tool and the user role within the tool showing what can be done by each user role within the particular section of the tool.

Tool Section	Interaction with the tool	User Role		
		Director	Player	Audience
Character Editor	Can update their journal from within the 'My Character' tab	Yes	Yes	N/A
Scene Builder	Can add a scene	Yes	No	No
	Can add a prologue to the role-play	Yes	No	No
My Character	Can update Journal	Yes	Yes	No
The Play	Can chat and interact with other players	Yes	Yes	Yes
	Can move to the next scene	Yes	No	No
	Can end the role-play	Yes	No	No
Analyser	Can comment on the role-play and how it progressed	Yes	Yes	Yes

Having a list of user roles and their access rights within the tool makes it now possible to allow and disallow access to particular features of the tool to users who have been assigned roles by the tool directors or workspace managers.

## **Database design**

In order to carry out the development of the *RolePlayControlTab* tool a well designed database needed to be created. This was important as all data that was to be used within the tool was not saved between sessions. This meant that, should a user leave the tool and then re-enter all the information saved and displayed to that point would be lost. The incorporation of a database would allow a role-play's settings and conversation to be stored and retrieved at any time.

The database structure was heavily influenced by the different tabs used in the application. By going through all the data that was to be entered into each of the different sections of the tool, the project developers were able to build appropriate database schemas.

NOTE: The Database Schema for all the databases used in the *RolePlayControlTab* tool can be found in Appendix B.

## 7.3 Development

### The development phase

The development phase is the process that is employed to reach the required solution. Initially an application is developed and thoroughly tested, before being released to the client for testing within the client's environment. Any arising issues identified by the client as a result of their testing are fed back to the development team who then work to address these identified issues.

Once these identified issues have been addressed by the development team, the application is returned to the client for further follow-up testing to ensure that the required changes have been provided.

When the client is satisfied with the application, the developers then prepare the application for distribution.

### The technology and development platform

The development platform was decided upon before the project was started. One of the main objectives of the project was to successfully develop and deploy an application using Groove technology. Another objective was to see how easy Groove application design and development would be when completely designed using Microsoft .Net technology.

With their recent attempts to enter the marketplace, Groove Networks have allowed parts of their application technology to become accessible to software developers. This has allowed developers to build applications as they would any other .Net application and then embed such applications within a Groove Workspace, making the applications available for use by all users of the Workspace.

Although the *RolePlayControlTab* tool was designed and developed using Microsoft's .NET technology, the final application was used within a Groove Workspace.

### Groove Workspace

Groove Workspace is an application that allows users to create and use Peer-to-Peer collaborative environments. This software application allows users to carry out business/education functions in a single online environment.

### Groove development kit

The Groove Development Kit (GDK) allows developers to build tools for use within Groove. Although the development language is in JavaScript and is extremely flexible, there is no application development platform such as Visual Studio .NET.

Groove .NET tools can now be deployed like any other Groove tool. When a new .NET tool is installed into a shared space, Groove automatically installs required .NET components if they are not already available on the system.

This feature adds to the attraction of developing applications using Visual Studio .NET and the Groove .NET Toolkit rather than just using the standalone Groove Development Kit.

## **Groove .NET toolkit**

The Groove .NET Toolkit is an add-in to Microsoft Visual Studio .NET. It allows the developer to create a complete Groove tool from within Visual Studio. The supported languages are C# or VB.NET.

The Groove .Net Toolkit now provides a native .NET interface to most of the Groove API. The API is now managed thereby freeing developers from worrying about Memory Management and .COM issues.

The .NET Toolkit also allows the toolbox to easily add Groove components to a desired application and to set their design-time properties and event handlers. The .NET Toolkit will also automatically build all required Groove XML Files.

A debug controller is also provided to allow tools to be run in debug mode within a temporary Groove Workspace. The debugger runs seamlessly from within Visual Studio .NET and works as though it was debugging any other .NET application.

## **Microsoft's .NET framework**

The .NET Framework provided by Microsoft is a new computing platform that simplifies application development in the highly distributed environment of the Internet. The .NET Framework is designed to fulfil the following objectives:

- To provide a consistent object-oriented programming environment whether object code is stored and executed locally, executed locally but Internet-distributed, or executed remotely.
- To provide a code-execution environment that minimizes software deployment and versioning conflicts.
- To provide a code-execution environment that guarantees safe execution of code, including code created by an unknown or semi-trusted third party.
- To provide a code-execution environment that eliminates the performance problems of scripted or interpreted environments.
- To make the developer experience consistent across widely varying types of applications, such as Microsoft Windows-based applications and web-based applications.

- To build all communication on industry standards to ensure that code based on the .NET Framework can integrate with any other code.

The .NET Framework has two main components: the common language runtime and the .NET Framework class library. The common language runtime manages memory, thread execution, code execution, code safety verification, compilation, and other system services. The .NET Framework class library is a collection of reusable types that tightly integrate with the common language runtime.

## **Visual Studio .Net**

Visual Studio .NET is the development platform used to develop applications that use .Net technology. There are a variety of languages supported (such as Visual Basic .NET and C#)

The chosen language to develop the application was Visual Basic.NET since it would allow for rapid application development.

## **The application development**

Once the platform and development applications were finalised attention was turned to the development of the application. With the tools that were provided the development was split into the GUI and the database backend that would make the role-play possible.

## **GUI development**

The GUI was designed as any other Visual Basic .NET application would have been. The Groove .NET Toolkit ties into Visual Studio .NET and provides most of the standard components that any other application would use (such as buttons, text boxes etc.) This allowed the GUI to be developed quickly and efficiently with necessary changes able to be made in as little time as possible. The toolbox allowed the easy addition of Groove components to the application and set their design-time properties and event handlers.

## **Database development**

The Groove .NET Toolkit is ADO-aware. This allows data to be stored and retrieved in the same manner as you would with any other application built in .NET

The Groove .NET Toolkit provides the following components:

- RecordSetEngine: Provides the underlying record-oriented storage and replication services.
- ToolCollections: Provides sorting, views and query facilities.

- RSEConnection: Provides connectivity to the RecordSetEngine and manages transactions.
- RSECommand: Defines data access commands (add, modify, delete, and query). The command statements themselves are expressed in SQL.
- RSEDataAdapter: Consists of a subclass of the ADO DataAdapter class and is used for reading and writing data. In general, it transfers data between the record set engine and an ADO data structure such as a DataSet, using an SQL command to define the transfer.

Once a RecordSetEngine has been added to a Groove .NET Solution, the RSE can be interacted with using RSECommand objects. An SQL statement can be passed through the RSECommand then executed via the ExecuteNonQuery function.

The database-related components used in the *RolePlayControlTab* tool are listed below:

<b>Character</b>	<b>Purpose</b>
Name	Stores the name of the character
Role	Stores the role of the player
PublicProfile	Stores the public profile of the character (viewable by all other players within the role-play)
PrivateProfile	Stores the private profile of the character (only viewable by the player himself and a director in the role-play or manager of the workspace)
Journal	Stores the user's personal journal that will contain their thoughts on the role-play as it progresses
UserName	The Groove username of the player who has been assigned to the character.

<b>Chat</b>	<b>Purpose</b>
Chatlog	The text message that was entered into the chat session by one of the players
_CreatedBy	The Groove username of player who entered the line in the chat session

<b>Settings</b>	<b>Purpose</b>
OptionName	The name of the setting we wish to store
OptionValue	The value of the setting we wish to store

<b>Comments</b>	<b>Purpose</b>
CommentEntry	The comment that was entered in the analyser tab when the role-play is completed
_Createdby	The Groove username of player who entered the comment into the comment section

<b>Scene</b>	<b>Purpose</b>
SceneName	The name of the scene. This is automatically assigned and is ordered from the first to the last scene
SceneDescription	Stores the description of the scene (this is provided by the user)

## The RecordSetEngine

The Groove RecordSetEngine is the basis for database storage within Groove. Once a RecordSetEngine has been created, all the other database-related components can be used to store and retrieve data within the Groove Workspace.

The list of RecordSetEngines created in the *RolePlayControl/Tab* tool are as follows:

Record Set Engine	Purpose
ThePlayRecordSetEngine	RecordSetEngine used to store all the chat logs
SceneBuilderRecordSetEngine	RecordSetEngine used to store all the scenes and their appropriate details
CommentsRecordSetEngine	RecordSetEngine used to store all the comments entered in the Analyser section of the application
SettingsRecordSetEngine	RecordSetEngine to store all the settings for the tool
CharacterEditorRecordSetEngine	RecordSetEngine to store the characters and all their required details

## The datasets

Datasets store data in a disconnected cache. The structure of a dataset is similar to that of a relational database: it exposes a hierarchical object model of tables, rows, and columns. In addition, the dataset contains constraints and relationships defined for the dataset.

Within the tool are defined several datasets that were used to store results from the RecordSetEngine that were gained from the execution of the RSECommands

The defined datasets within the tool are as follows:

Dataset	Explanation
ChatDataSet	This is the dataset to store any data retrieved from the execution of RSECommands on the ChatRecordSetEngine
ScenesDataSet	This is the dataset to store any data retrieved from the execution of RSECommands on the ScenesRecordSetEngine
CharacterEditorDataset	This is the dataset to store any data retrieved from the execution of RSECommands on the CharacterReocrdSetEngine
CommentsDataset	This is the dataset to store any data retrieved from the execution of RSECommands on the CommentsRecordSetEngine
ChatSettingsDataSet	This is the dataset to store any data retrieved from the execution of RSECommands on the ChatSettingsRecordSetEngine

## The RSE Data adapters and their appropriate commands

Groove Workspace operates in the same way as a standard .NET Data Adapter. Data Adapters are used to read and write data to a dataset or Groove RecordSetEngine. It allows SQL statements, that are assigned to RSECommands, to be executed.

Once assigned the RSECommands can be relayed to the Data Adapter and the required data is sent through to the RecordSetEngine. Once passed through, the data in the RecordSetEngine is either updated (Insert and Update statements) or the results are sent back and filed into a Data Set (Select statements)

Below is a list of Data Adapters and their appropriate RSE Commands:

Data Adapter	Command		
	Select	Insert	Update
The Play Data Adapter	The Play Select Command	The Play Insert Command	N/A
Character Editor Data Adapter	Character Editor Select Command	Character Editor Insert Command	Character Editor Update Command
Scene Builder Data Adapter	Scene Builder Select Command	Scene Builder Insert Command	Settings Update Command
Comments Data Adapter	Comments Select Command	Comments Insert Command	N/A
Settings Data Adapter	Settings Select Command	Settings Insert Command	N/A

## The RSEConnections

The RSEConnection object is similar to a .NET Connection. A Connection in the .NET Framework serves as a bridge between an application and a data source and is used to retrieve data from a data source and to reconcile changes to that data with the data source. The RSEConnection works with the RecordSetEngine to achieve the same objective.

The list of created RSEConnections within the *RolePlayControlTab* tool are as follows:

<b>Connections</b>	
CharacterEditorConnection	Connection to the CharacterRecordSetEngine
ThePlayConnection	Connection to the ThePlayRecordSetEngine
SceneBuilderConnection	Connection to the SceneBuilderRecordSetEngine
SettingsConnection	Connection to the SettingsRecordSetEngine
CommentsConnection	Connection to the CommentsRecordSetEngine

## The ToolCollection components

ToolCollection Components provide the ability to sort, view and query the appropriate RSEConnections. All connections require a ToolCollection Component to manipulate the data that is retrieved from the RSEData Adapter.

The ToolCollection components that were used in the *RolePlayControlTab* tool are:

<b>Collection Components</b>	
ThePlaytoolCollectionComponent	Provides sorting, views and query facilities for ThePlayConnection
SceneBuilderToolCollectionsComponent	Provides sorting, views and query facilities for SceneBuilderConnection
CharacterEditorToolCollectionComponent	Provides sorting, views and query facilities for CharacterEditorConnection
SettingsCollectionComponent	Provides sorting, views and query facilities for SettingsConnection
CommentsCollectionComponent	Provides sorting, views and query facilities for CommentsConnection

## 7.4 Testing

The testing of software during its development phase is extremely important. Because of this, the developers of this present project undertook a program of continuous testing to ensure that all features that were added during the development process worked as they should.

During testing, the developers ensured that:

- minor bug fixes did not create major bugs in the code
- the application was working as it should be
- the application was simple and intuitive
- no features that would simplify the application had been left out of the application, and
- the application had met all feature requirements and was therefore able to be released to the project supervisor.

### Unit testing

Unit tests were used to document use-cases that record what each added object of the application should do when given a controlled set of inputs. Such testing was required in order to confidently say that a given part of the application would perform as it should.

Prior to writing the code, a list of required tests was designed to confirm that any parameters that were passed, or any situation that occurred, was handled in the way it should. This allowed the developer to focus on smaller parts of the code and avoid continuously implementing new features that could not be thoroughly tested.

The developers of this project were also able to test any changes made to the code against the unit test. Through comparison analysis of the code against the unit test after a change had been made, the developers were able to identify whether or not that change had introduced a new bug to the application.

### Functional testing

While unit tests measure the reliability of the code, they do not take actual application usage into consideration. Functional testing is designed to test the user requirements of the application. Functional testing confirms that the designed system actually works as specified in the project requirements.

Through the use of functional testing, the developers were able to expose many problems that unit testing was not able to discover, and in so doing, confirm that the application was working in a useful and efficient manner.

## **Conclusion on testing with regard to the project**

The use of unit testing confirmed that the code was working as it should be while the use of functional testing confirmed the same with regard to the actual behavior of the application.

Through using both unit and functional testing methods, the developers were able to confirm that the application was working properly and that it had met the project specifications requirements.

Note: Refer to Appendix C for complete test cases.

## 7.5 Implementation

The implementation phase involves the incorporation of the newly developed application into the actual day-to-day operational context that the application was designed to be used in.

A description of the implementation phase of the *RolePlayControlTab* tool is provided below.

### ***RolePlayControlTab* installation procedures**

The *RolePlayControlTab* tools can be installed in different ways depending on how the educator decides to implement the tool in their workspace. Possible options are as follows.

1. Installing the *RolePlayControlTab* tool in own workspace:
  - a. required files are supplied to you on disc, or
  - b. required files are made available online and the educator is supplied with (or has downloaded) the file *RolePlayTool.grv*.
2. If joining a workspace which has the *RolePlayControlTab* tool already installed:
  - a. required files are made available online, or
  - b. required files are supplied to student on disc, or
  - c. required files are made available online and the educators have been supplied with (or have downloaded) the file *RolePlayTool.grv*.

These methods of implementation are explained in further detail below:

1. Installing the *RolePlayControlTab* tool in a workspace

To install the role-play tool in a workspace the educator will have to access the tool installer which will either be supplied via the Internet or provided on disc.

- a. Required files are supplied on disc

If the required installation files are supplied to the educator on disc the educator will simply have to enter the directory that the filename *RolePlayTool.grv* is located in and double click the file. The tool will then be injected into the Groove Workspace where it will be able to be added to the specific required workspace by selecting the "Add Tool" section of the workspace

- b. Required files are made available online and the educator is supplied with (or has downloaded) the file *RolePlayTool.grv*

This method requires an Internet connection to allow the tool to be downloaded to the educator's computer. If the *RolePlayTool.grv* file has been downloaded or supplied as a single file then the .grv file has the required location of the entire application and will attempt to download it

directly off the hosting server. The educator will simply have to save the *RolePlayTool.grv* file to their computer and double click it. Groove will automatically attempt to connect to the server storing the role-play tool and install it in their Groove Workspace.

The tool will be injected into the educator's Groove profile and the educator will then be able to add it to the workspace by selecting the 'Add Tool' section.

2. If joining a workspace which has the *RolePlayControlTab* tool already installed, these steps should be taken by the student user if the student user does not have any intention of installing the tool into their own Groove profile.

- a. Required files are made available online

If the student user is a member of a Groove Workspace that already has the tool installed and the installation files have been made available online, the installation will just be a matter of clicking on the applications tab in the workspace and the tool will automatically download and install itself in the student's workspace.

- b. Required files are supplied to the student user on disc

If the required installation files are supplied to the student user on disc the student will simply have to enter the directory that the *RolePlayTool.grv* is located in and double click the file. The tool will then be accessible to the student.

- c. Required files are made available online and the student user has been supplied (or has downloaded) the file *RolePlayTool.grv*

This method will require the student user to have an active Internet connection to allow the tool to be downloaded to their computer from a remote server. Simply allow the student user to save the *RolePlayTool.grv* file to their computer or supply it on disc and allow the student to double click the .grv file. Groove will automatically attempt to connect to the server storing the role-play tool and install it into the student's Groove profile.

## 8 Problems faced

The following table displays the problems that were encountered during the development of the *RolePlayControlTab* tool, the effect of those problems and the solution that was applied to resolve the problem.

Nos.	Description	Effect	Workaround
1	Tool Tips in .NET applications are not supported in Groove	No tool tips could be used in the application.	Had to use message boxes to inform the users of changes occurring within the application.
2	Separate tools installed in the same workspace can only communicate with each other with great difficulty	Tools installed can not share data with one another.	Build the role-play tool as a single application and split the different tool parts with tabs.
3	No external database system (such as Microsoft SQL server)	Had to find alternative means of data storage.	Used the Groove RecordSetEngine to store all the required data within the Groove Workspace.
4	Updates to the RecordSetEngine do not trigger events	When a user updates a scene or their details changes are not immediately viewable.	When button is pushed to update details check to see if an update on the database was made (if so update accordingly) Unfortunately this creates a rather large performance hit.
5	Database entry errors when data is entered too quickly	Error messages reporting that data could not be inserted.	Data must be entered and the user must wait for a short time while the delta files traverse across the workspace to all users.
6	Slow updates in chat session when a user is on a slower connection	When a user is on a slower connection there can be a delay (lag) in the chat session.	Code was implemented to avoid data being entered frequently (a pause of a few seconds). However, this was considered restrictive in the role-play, so it was removed.
7	Part of the API is still closed off or in Beta stage	Some features that may have improved the Role-play tool had to be ignored.	Avoid using the beta or testing parts of the API. There may not be support for that part in the future.

## 9 Future recommendations

The *RolePlayTabControl* tool (developed using Groove Workspace) is an experimental approach to identify the capability and adaptability of tools developed within Groove Workspace for collaborative e-learning initiatives. The tool developed by this project demonstrates that Groove Workspace is an ideal platform for developing customisable collaborative e-learning tools that have an educational focus and application through supporting the four key activities performed in an online environment: Dialogue, Involvement, Support and Control.

The *RolePlayControlTab* developed by this project is, however, still a prototype tool. Consequently, there is ample scope for further improvement and modification to improve its usability. Some of the areas that can be developed further are as follows:

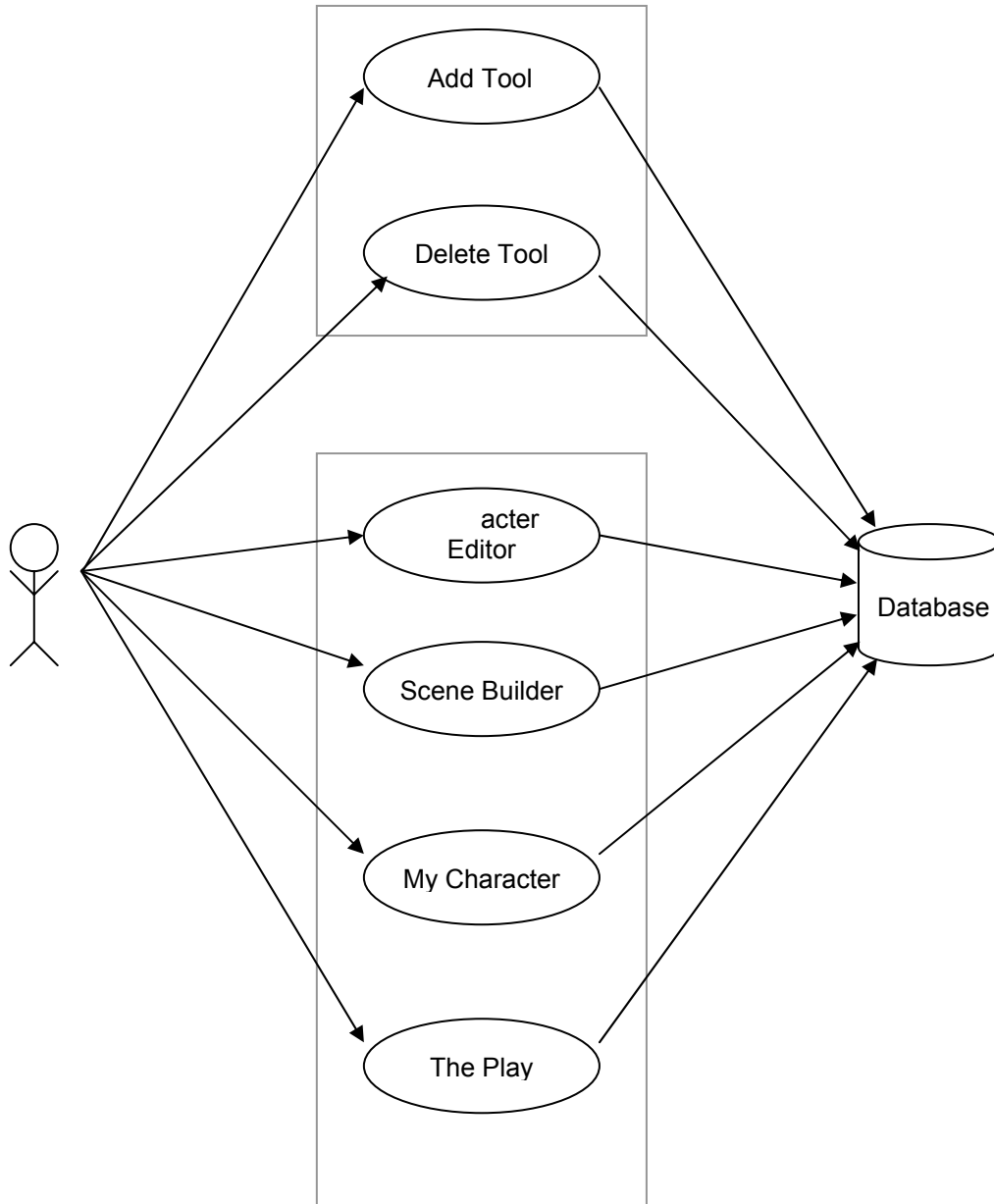
- incorporating a video tool within the *RolePlayControlTab* tool
- including graphics, animation, tables, critical-thinking tools etc to make the tool more interactive and interesting
- including a search functionality
- developing the *RolePlayControlTab* tool using 'Groove forms' for a better look and feel as per capture below

- developing a portfolio section for the student users, that can be used by educators for evaluation purposes, and/or
- making the role of the Analyser/Facilitator more interactive and versatile with the development of expert systems, instead of requiring the Director to monitor the play at all times.

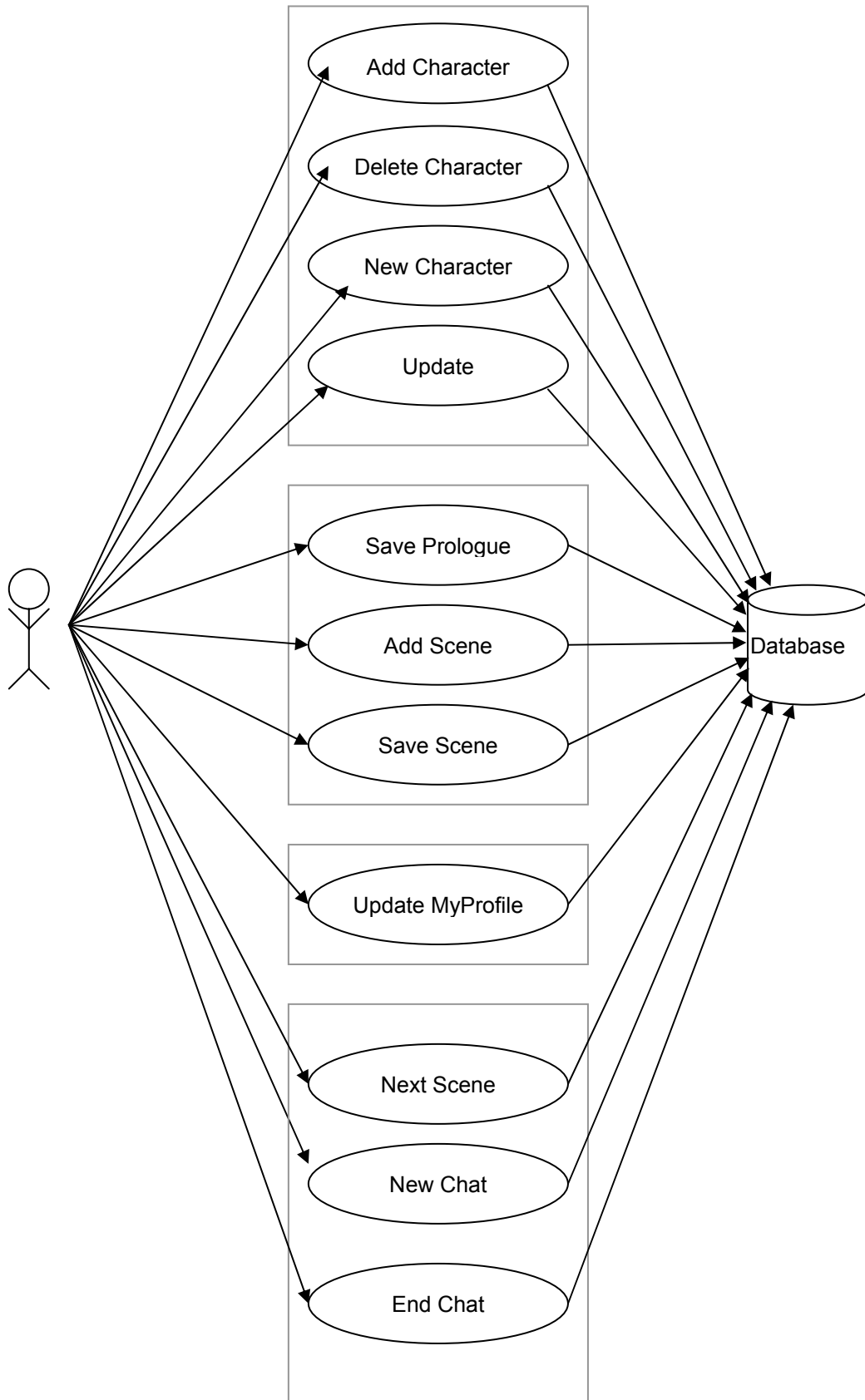
## Appendix A – use-case diagrams

These representations are used to show the use cases for each character role.

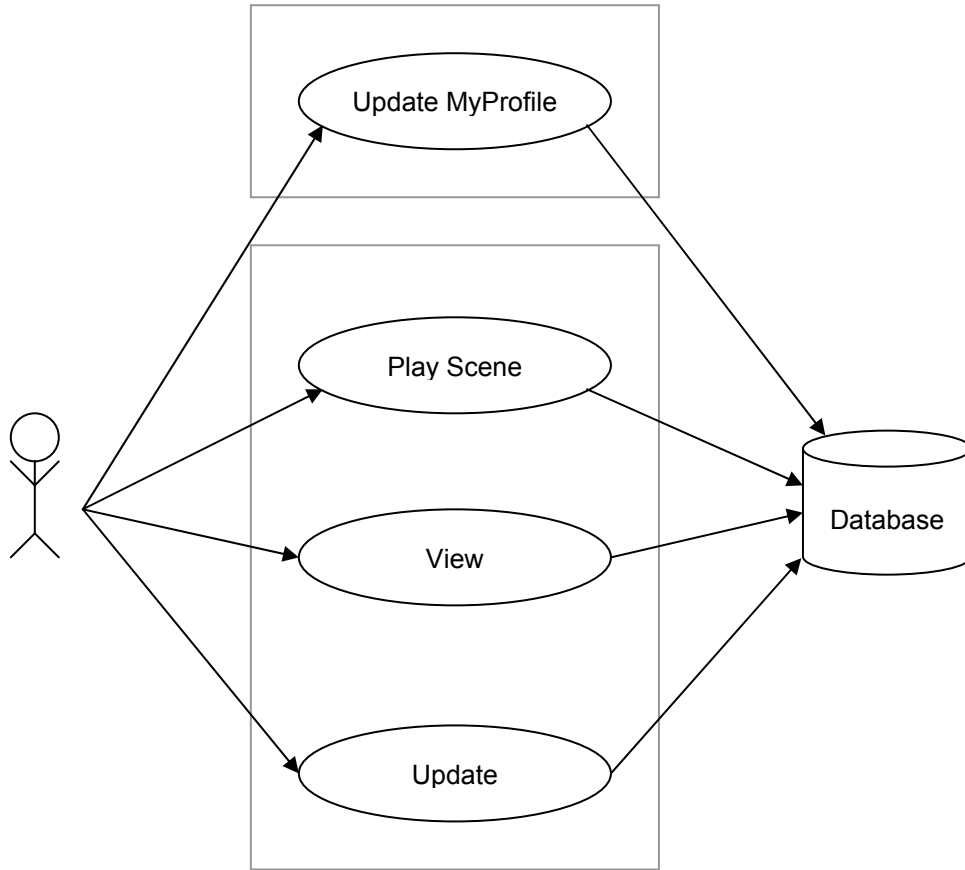
### Use case 1: User is a *'Manager'* of the workspace



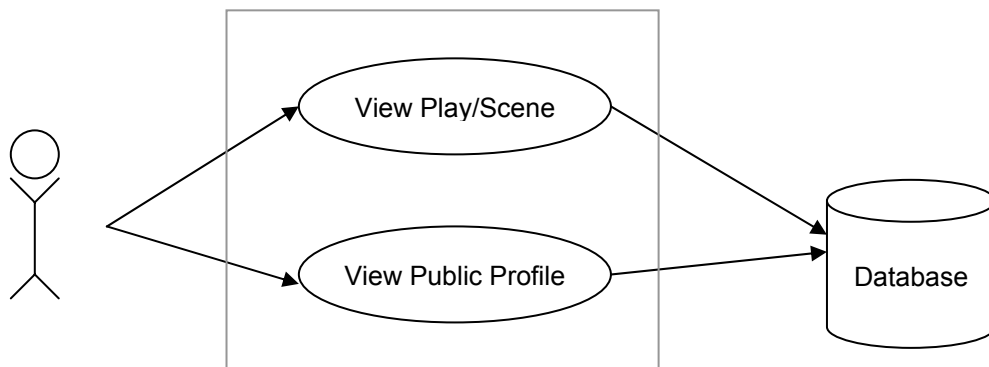
## Use case 2: User is a 'Director' of the workspace



### Use case 3: User is a 'Player' of the workspace



### Use case 4: User is 'Analyser' of workspace



## Appendix B - Database schema

### Database schema

<b>Character</b>
Name
Role
PublicProfile
PrivateProfile
Journal
UserName

<b>Chat</b>
ChatLog
_CreatedBy

<b>Scene</b>
SceneName
SceneDescription

<b>Settings</b>
OptionName
OptionValue

<b>Comments</b>
CommentEntry
_Createdby

## Appendix C - Functional test cases

### 1 Scenario: Add the *RolePlayTabControl* tool

This scenario tests that a user with suitable authorisation can add the *RolePlayTabControl* tool into the workspace.

Sl. No.	User Actions	Expected result	Pass/Fail
1.	<ol style="list-style-type: none"> <li>1. Login to the Groove Workspace with superuser/Manager authorisation.</li> <li>2. Navigate to your space and click on the 'Add Tool' option.</li> <li>3. Select the <i>RolePlayTabControl</i> tool from the list of available tools.</li> <li>4. Click on the 'Add selected' option.</li> </ol>	The user should be able to add the <i>RolePlayTabControl</i> tool and it should be placed before the 'Add tool' option on the space.	Pass
2.	<ol style="list-style-type: none"> <li>1. Login to the Groove Workspace without superuser/manager authorisation (to add tools).</li> <li>2. Navigate to your space and click on the 'Add Tool' option.</li> </ol>	The 'Add tool' option should not be visible and thus the user cannot add new tools.	Pass

## 2 Scenario: Delete the *RolePlayTabControl* tool

This scenario tests that a user with suitable authorisations can delete the *RolePlayTabControl* tool from the workspace.

Sl. No.	User Actions	Expected result	Pass/Fail
1.	<ol style="list-style-type: none"> <li>1. Login to the Groove Workspace with superuser/manager authorisation.</li> <li>2. Navigate to your space and right click on the tool tab.</li> <li>3. Click on the 'Delete tool' option.</li> <li>4. Click on 'Yes' when the system confirms the deletion.</li> </ol>	The 'Delete Tool' option should be enabled, thus allowing the user to delete the tool.	Pass
2.	<ol style="list-style-type: none"> <li>1. Check for the tool after successful deletion on the workspace.</li> </ol>	The deleted tool should be removed from the workspace after confirming the deletion.	Pass
3.	<ol style="list-style-type: none"> <li>1. Login to the Groove workspace without sufficient superuser/Manager authorisation.</li> <li>2. Navigate to your space and right click on the tool tab.</li> <li>3. Click on the 'Delete tool' option.</li> </ol>	The 'Delete Tool' option should not be enabled, thus preventing the user from deleting tools.	Pass

### 3 Scenario: Character Editor

This scenario tests that a user with suitable authorisations can create a character successfully using the *RolePlayControlTab* tool.

SI. No.	User Actions	Expected result	Pass/Fail
1.	<ol style="list-style-type: none"> <li>1. Login to the Groove Workspace with superuser/manager authorisation.</li> <li>2. Add the <i>RolePlayControlTab</i> tool.</li> <li>3. Click on the 'Character Editor' tab.</li> </ol>	The user should be able to access the 'Character Editor' tab, and all the fields should be enabled.	Pass
2.	<ol style="list-style-type: none"> <li>1. Login to the Groove workspace as a normal user without manager authorizations to create a character.</li> <li>2. Click on the Character Editor tab within the <i>RolePlayControlTab</i> tool.</li> </ol>	All the fields within the 'Character Editor' tab should be disabled, thus preventing the user from creating any new characters.	Pass
3.	<ol style="list-style-type: none"> <li>1. Login to the Groove Workspace with superuser/Manager authorisation.</li> <li>2. Add the <i>RolePlayControlTab</i> tool.</li> <li>3. Click on the Character Editor tab.</li> <li>4. Check for the status of buttons.</li> </ol>	Only the 'Add Character' and 'Clear' buttons should be enabled.	Pass
4.	<ol style="list-style-type: none"> <li>1. Login to the Groove Workspace with superuser/manager authorisation.</li> <li>2. Add the <i>RolePlayControlTab</i> tool.</li> <li>3. Click on the Character Editor tab.</li> <li>4. Create a character by entering relevant data into the following fields: <ul style="list-style-type: none"> <li>▪ Character name:</li> <li>▪ Character Role:</li> <li>▪ Select Player:</li> <li>▪ Public Profile:</li> <li>▪ Private Profile:</li> <li>▪ Journal:</li> </ul> </li> </ol>	The user should be able to enter relevant data into all the fields.	Pass
5.	Click on the 'Add character' button.	<p>The character should be added into the dataset.</p> <p>The character name should be displayed on the Characters list on the left hand side of the Character Editor Tab.</p>	Pass

Sl. No.	User Actions	Expected result	Pass/Fail
6.	Click on the 'Add character' button without entering data into all the fields.	The system should display a message box prompting the user to enter valid data into all the required fields.	Pass
7.	Add multiple characters by following steps 4 and 5.	All the characters added must be displayed within the characters list.	Pass
8.	<ol style="list-style-type: none"> <li>1. Click on any character name from the list of characters created.</li> <li>2. Check the status of the buttons.</li> </ol>	<p>Only, the following buttons should be enabled:</p> <p>Delete Character</p> <p>New Character</p> <p>Update</p>	Pass
9.	Click on any character name from the list of characters created.	The details of the character should be updated on to their respective fields on the 'Character Editor' tab window.	Pass
10.	<ol style="list-style-type: none"> <li>1. Select and character from the list of characters created.</li> <li>2. Click on the 'Delete Character' button.</li> </ol>	The selected character should be deleted from the list.	Pass
11.	<ol style="list-style-type: none"> <li>1. Select and character from the list of characters created.</li> <li>2. Update character details.</li> <li>3. Click on 'Update' button.</li> </ol>	The character data should be updated with the changes made.	Pass
12.	<ol style="list-style-type: none"> <li>1. Select and character from the list of characters created.</li> <li>2. Click on the 'New Character' button.</li> </ol>	All the data entry fields within the 'Character Editor' tab window should be cleared, thus allowing the user to create a new character.	Pass
13.	Without completing the creation of a new character, click on the 'Clear' button.	All the data entry fields within the 'Character Editor' tab window should be cleared without saving any data.	Pass
14.	While creating a character, select a player who has already been allocated a character and click on the 'Add Character' button.	A message box pops up asking the user to choose a different player for the character.	Pass

## 4 Scenario: Scene Builder

This scenario tests that a user with suitable authorisations can create/add scenes to be played using the *RolePlayControlTab* tool.

SI. No.	User Actions	Expected result	Pass/Fail
1.	<ol style="list-style-type: none"> <li>1. Login to the Groove Workspace with superuser/manager authorisation.</li> <li>2. Add the <i>RolePlayControlTab</i> tool.</li> <li>3. Click on the 'Scene Builder' tab.</li> </ol>	The user should be able to access the 'Scene Builder' tab with all the fields being enabled.	Pass
2.	<ol style="list-style-type: none"> <li>1. Login to the Groove Workspace as a normal user without manager authorisations to create a scene.</li> <li>2. Click on the 'Scene Builder' tab within the <i>RolePlayControlTab</i> tool.</li> </ol>	All the fields within the 'Scene Builder' tab should be disabled, thus preventing the user from adding/creating any new scenes.	Pass
3.	<ol style="list-style-type: none"> <li>1. Login to the Groove Workspace as a normal user without manager authorisations to create a scene.</li> <li>2. Click on the Scene Builder tab within the <i>RolePlayControlTab</i> tool.</li> <li>3. Check for the status of the buttons on the screen.</li> </ol>	The following buttons should be enabled: Save Prologue New Scene Save Scene	Pass
4.	Within the 'Scene Builder' tab window, enter the prologue(data) within the prologue text box and click on the 'Save Prologue' button.	The prologue should be saved into the database and should be displayed in the prologue display field on 'The Play' tab screen.	Pass
5.	Click on the 'New Scene' button.	Scene 1 should be added with the description field empty.	Pass
6.	Enter scene data into the description field and click on 'Save Scene' button.	The scene data for scene 1 should be saved and the description field should be cleared.	Pass
7.	Click on the 'New Scene' button.	Scene 2 should be added with the description field empty.	Pass
8.	Enter scene data into the description field and click on 'Save Scene' button.	The scene data for scene 2 should be saved and the description field should be cleared.	Pass

## 5 Scenario: My Character

This scenario tests that a user with an assigned character can view and update his profile, using the *RolePlayControlTab* tool.

Sl. No.	User Action	Expe	Pass/Fail
1.	1. Login to the Groove Workspace as a normal user. 2. Check for authorised character.	If the user has been assigned any character by the Director of the space, then the system will pop-up a message box prompting that user has been assigned the role of 'xyz'.  The character profile assigned to the user will be displayed in the 'My Character' tab screen.  Only the 'My Character' and 'The Play' tab should be enabled.	Pass
2.	Login to the Groove Workspace without any characters authorised.	The user will be displayed with a message box prompting that that user has not been allocated any character.  All the tabs will be disabled and all the fields will be empty.	Pass
3.	Check the details displayed within the fields for the character.	The details filled in by the Director of the space for the character should be displayed within the respective fields.	Pass
4.	Check for the editable fields.	Except for the journal text box, all the other fields should be made read-only. This allows the character to edit his personal journal as the play progresses.	Pass
5.	Update the journal and click on the 'Update My Profile' button.	The details of the character must be updated suitably.	Pass

## 6 Scenario: The Play

This scenario tests that a user with an assigned character can collaborate with other characters in the space to play the scenes, using the *RolePlayControlTab* tool.

Sl. No.	User Action	Expected result	Pass/Fail
1.	1. Login to the Groove Workspace as a normal user. 2. Check for authorised character.	If the user has been assigned any character (role : player) by the Director of the space, then the system will pop-up a message box prompting that user has been assigned the role of 'xyz'. The character profile assigned to the user will be displayed in the 'My Character' tab screen. Only the 'My Character' and 'The Play' tab should be enabled.	Pass
2.	Login to the Groove Workspace without any characters authorised.	The user should be displayed with a message box prompting that that user has not been allocated any character. All the tabs should be disabled and all the fields will be empty.	Pass
3.	Login to the Groove Workspace as a Director and check for the permissions within 'The Play' tab.	The following buttons should be enabled for a director: Update Journal Next Scene New Chat End Chat Note: All these buttons except the 'Update Journal' should be disabled for a normal player.	Pass
4.	Check for the display of the 'Prologue' and 'My journal' fields.	The prologue saved by the director during the creation of the scenes should be displayed in the prologue display field. The 'My Journal' field should display any journal details of the character.	Pass
5.	Check for the list of players within the 'Play' group box.	The list of players/characters active for the play should be displayed within the list within the 'Play' group box.	Pass
6.	Select any character and click on the 'View Public Profile' button.	The public profile of the selected character should be displayed.	Pass
7.	Check for the scene display field.	The first scene should be displayed by default within the scene display field.	Pass

Sl. No.	User Action	Expected result	Pass/Fail
8.	Click on the 'Next scene' button.	The system should display the next scene within the scene display field. Also, the chat display field should display a message stating that a new scene has started.	Pass
9.	Click on the 'Next Scene' button after the last scene has been displayed (only for Directors).	The user should be prompted by a message box stating that there are no more scenes available.	Pass
10.	Enter text into the chat entry field and hit enter key during the play.	The entered data should be displayed within the chat display screen against the name of the character(eg <character name> 'message').	Pass
11.	Enter text into the chat entry field and click on the 'Submit' button.	The entered data should be displayed within the chat display screen against the name of the character (eg <character name> 'message').	Pass
12.	Make journal entries into the 'My Journal' field and click on the 'Update Journal' button.	The entered journal data must be update into the database.	Pass
13.	Click on the 'New Chat' button (only for Directors).	The chat fields should be cleared.	Not implemented
14.	At the end of the play, click on the 'End chat' button (only for Directors).	The play is ended and the Analyser tab should be unlocked. The system prompts the users that the play has ended.	Pass

## 7 Scenario: Analyser

This scenario tests that a user with an assigned character can take part in the analysis of the play with other characters in the space at the end of play, using the *RolePlayControlTab* tool.

Sl. No.	User Action	Expected result	Pass/Fail
1.	<ol style="list-style-type: none"> <li>1. Login to the Groove Workspace as a normal user.</li> <li>2. Check for authorised character.</li> </ol>	<p>If the user has been assigned any character (role: Analyser) by the Director of the space, then the system will popup a message box prompting that user has been assigned the role of 'xyz'.</p> <p>Only 'The Play' tab should be enabled.</p> <p>The analyser tab should be disabled/locked by default to all users.</p>	Pass
2.	Click the 'End chat' button within 'The Play' tab.	<p>The analyser tab should be unlocked and the Comments/Results entry field should be enabled.</p> <p>The Add comment button is enabled.</p>	Pass
3.	Check for data display within the Analyser tab.	<p>The chat session field should display the whole chat record of the play.</p> <p>The characters list box within the 'Select Character to view Journal' group box should be filled with the list of the characters allocated for the role-play.</p>	Pass
4.	Select any character from the list.	The selected character's journals should be displayed within the journal display field.	Pass
5.	Enter comments into the comment entry field and hit the enter key.	The entered comments should be displayed within the comments display field against the name of the character (eg <character name> 'comment').	Pass
6.	Enter comments into the comment entry field and click on the 'Add Comment' button.	The entered comments should be displayed within the comments display field against the name of the character (eg <character name> 'comment').	Pass