國立中山大學電機工程學系

碩士論文

Department of Electrical Engineering

National Sun Yat-sen University

Master Thesis

機器人人偶表演劇本之圖形化編輯平台

The Graphic Authoring Platform of Screenplays for Robotic Puppet Shows

研究生：蕭智中

Jhih-Jhong Siao

指導教授：黃國勝 博士

Dr. Kao-Shing Hwang

中華民國 101 年 8 月

August 2012

國立中山大學研究生學位論文審定書

本校電機工程學系碩士班

研究生 蕭智中 (學號：M983010099) 所提論文

機器人人偶表演劇本之圖形化編輯平台

The Graphic Authoring Platform of Screenplays for Robotic Puppet Shows

於中華民國 101 年 8 月 28 日經本委員會審查並舉行口試，符合碩士學位論文標準。

學位考試委員簽章：

召集人＿＿＿＿＿＿ 委 員＿＿＿＿＿＿

委 員＿＿＿＿＿＿ 委 員＿＿＿＿＿＿

指導教授＿＿＿＿＿＿

eThesys 學位論文

博碩士論文授權書

（國科會科學技術資料中心版本 93.2.6）

本授權書所授權之論文為本人在 國 立 中 山 大學(學院) 電機工程 系所 網路多媒體 組 101 學年度第 1 學期取得 碩 士學位之論文。

論文名稱： 機器人人偶表演劇本之圖形化編輯平台

☑同意 ☐不同意

本人具有著作財產權之論文全文資料，授予行政院國家科學委員會科學技術資料中心（或其改制後之機構）、國家圖書館及本人畢業學校圖書館，得不限地域、時間與次數以微縮、光碟或數位化等各種方式重製後散布發行或上載網路。

本論文為本人向經濟部智慧財產局申請專利(未申請者本條款請不予理會)的附件之一，申請文號為：__________，註明文號者請將全文資料延後半年再公開。

☑同意 ☐不同意

本人具有著作財產權之論文全文資料，授予教育部指定送繳之圖書館及本人畢業學校圖書館，為學術研究之目的以各種方法重製，或為上述目的再授權他人以各種方法重製，不限地域與時間，惟每人以一份為限。

上述授權內容均無須訂立讓與及授權契約書。依本授權之發行權為非專屬性發行權利。依本授權所為之收錄、重製、發行及學術研發利用均為無償。上述同意與不同意之欄位若未鉤選，本人同意視同授權。

指導教授姓名：

研究生簽名： 學號:M983010099

（親筆正楷） （務必填寫）

日期:民國 101 年 9 月 12 日

1. 本授權書（得自 http://sticnet.stic.gov.tw/sticweb/html/theses/authorize.html 下載，或至 http://www.stic.gov.tw 首頁右下方下載）請以黑筆撰寫並影印裝訂於書名頁之次頁。
2. 授權第一項者，請確認學校是否代收，若無者，請個別再寄論文一本至台北市(106-36)和平東路二段 106 號 1702 室 國科會科學技術資料中心 黃善平小姐。(本授權書諮詢電話:02-27377606 傳真：02-27377689)

學年度： 101
學期： 1
校院： 國立中山大學
系所： 電機工程學系
論文名稱(中)： 機器人人偶表演劇本之圖形化編輯平台
論文名稱(英)： The Graphic Authoring Platform of Screenplays for Robotic Puppet Shows
學位類別： 碩士
語文別： Eng
學號： M983010099
提要開放使用： 是
頁數： 76
研究生(中)姓： 蕭
研究生(中)名： 智中
研究生(英)姓： Siao
研究生(英)名： Chih-Chong
指導教授(中)姓名： 黃國勝
指導教授(英)姓名： Kao-Shing Hwang
關鍵字(中)： 機器人控制器
關鍵字(英)： User interface
關鍵字(英)： XML
關鍵字(英)： DARwIn-OP
關鍵字(英)： Puppet show

中文提要:

隨著網路的發展越來越興盛，人們亦越來越習慣在網路上進行資訊交換。因此，在機器人控制器的選擇，也不再侷限於控制單一的機器人。本論文的目的在於提供一個大眾化的機器人劇場表演製作平台；可控制多隻機器人；每隻機器人根據劇本編輯平台所設定的動作做表演。無論何時何地，任何人想要使用機器人劇場表演製作平台來設計機器人劇本，只要手邊有一個可以上網的裝置，即可進行。機器人劇場表演製作平台分為三個部分：人偶劇場之圖形化劇本編輯平台(GAP)、劇場機器人之轉譯器開發(SI)、表演機器人。本論文的重點在於圖形化劇本編輯平台以及表演機器人(DARwIn-OP)的實作。人偶劇場之圖形化劇本編輯平台提供多種機器人給使用者選擇;使用者透過拖曳圖片的操作方法可輕鬆的設計自己的機器人劇本；以圖片來表示機器人的動作、行為以及劇本；在每一次的劇本更新時，系統會自動將劇本存成XML檔案格式。此圖形化劇本編輯平台提供多組姿勢或者是行為給使用者；使用者可自行輸入機器人台詞，機器人會說出

使用者所輸入的台詞；在情緒表達方面，使用者可自行設定動作速度，藉以傳達該事件的情緒。系統最後以劇本”do-as-I-do”來呈現，並將此劇本錄製成影片放在YouTube: “http://www.youtube.com/watch?v=v8ErTOgAQSo”.

英文提要:

With the development of the network, people are increasingly used to exchanging information on the Internet. Therefore, the capability of robot controller should not be limited to control robots locally. The objective of this thesis is to provide a system, the screenplay based performance platform of Robotic puppet shows (SBPP), commanding multiple robots; each robot performing its own role based on a script composed by the developed authoring tool. Wherever and whenever a user wants to use SBPP, he/she just needs connect to the network and begins to design a script. SBPP consists of three parts: the graphic authoring platform (GAP) of screenplays for robotic puppet shows, the screenplay interpreter (SI) for multi-morphic robots, and robots themself. The work of this thesis is concentrated on the implementation of the GAP and robots (model: DARwIn-OP) control. The GAP provides options for a variety of robots to users. The users can easily design their own robot scripts merely by drag-and-drop operating on icons representing the actions, behavior, and short scripts, respectively. Whenever a script is created or updated, GAP will automatically save the script as an XML file format internally. In addition, robots can be conducted to express their emotions orally by utter the lines composed. The system is demonstrated by a play of “do-as-I-do” and recoded in a video at YouTube:” http://www.youtube.com/watch?v=v8ErTOgAQSo”.

致謝

在研究所的生涯中，首先要感謝指導教授 黃國勝老師的指導與教誨。感謝老師在每次的討論中都清楚地指出我的想法中的不足，給我明確的努力方向。真的很高興在研究所生涯遇到這麼親切的老師，老師的指導讓我在研究、學習以及思考方面都獲益良多。

感謝所有實驗室學長、學弟、同學。每當我累的時候總是可以找到人說說話，緩和心情。感謝蔣惟丞學長以及劉佳諺學長，在我的實作過程中給予許多寶貴的意見，也在我撰寫論文的過程中幫助我許多。感謝豐全、哲勛、又齊、晁瑋、凡瑋、力維，因為有你們實驗室變得熱鬧許多。感謝宇安、旻璋，在我遇到瓶頸的時候和我一起想辦法解決，在我的研究過程中你們幫助我許多，讓我得以完成我的研究。謝謝實驗室的各位，研究所生涯的點點滴滴都是我最珍貴的回憶。

最後要感謝我的家人。感謝爸媽在我求學期間的支持與耐心，讓我能夠衣食無憂地完成我的求學旅程；感謝你們多年的辛苦與栽培，在此對我的家人獻上我最大的敬意與謝意。

蕭智中 謹致於國立中山大學電機工程研究所

智慧型機器人及訊息系統實驗室

中華民國一百零一年九月

摘要

隨著網路的發展越來越興盛，人們亦越來越習慣在網路上進行資訊交換。因此，在機器人控制器的選擇，也不再侷限於控制單一的機器人。本論文的目的在於提供一個大眾化的機器人劇場表演製作平台；可控制多隻機器人；每隻機器人根據劇本編輯平台所設定的動作做表演。無論何時何地，任何人想要使用機器人劇場表演製作平台來設計機器人劇本，只要手邊有一個可以上網的裝置，即可進行。機器人劇場表演製作平台分為三個部分：人偶劇場之圖形化劇本編輯平台(GAP)、劇場機器人之轉譯器開發(SI)、表演機器人。本論文的重點在於圖形化劇本編輯平台以及表演機器人(DARwIn-OP)的實作。人偶劇場之圖形化劇本編輯平台提供多種機器人給使用者選擇;使用者透過拖曳圖片的操作方法可輕鬆的設計自己的機器人劇本；以圖片來表示機器人的動作、行為以及劇本；在每一次的劇本更新時，系統會自動將劇本存成 XML 檔案格式。此圖形化劇本編輯平台提供多組姿勢或者是行為給使用者；使用者可自行輸入機器人台詞，機器人會說出使用者所輸入的台詞；在情緒表達方面，使用者可自行設定動作速度，藉以傳達該事件的情緒。系統最後以劇本”do-as-I-do”來呈現，並將此劇本錄製成影片放在 YouTube: “http://www.youtube.com/watch?v=v8ErTOgAQSo”.

關鍵詞：人機介面、XML、DARwIn-OP、人偶劇場、機器人控制器

Abstract

With the development of the network, people are increasingly used to exchanging information on the Internet. Therefore, the capability of robot controller should not be limited to control robots locally. The objective of this thesis is to provide a system, the screenplay based performance platform of Robotic puppet shows (SBPP), commanding multiple robots; each robot performing its own role based on a script composed by the developed authoring tool. Wherever and whenever a user wants to use SBPP, he/she just needs connect to the network and begins to design a script. SBPP consists of three parts: the graphic authoring platform (GAP) of screenplays for robotic puppet shows, the screenplay interpreter (SI) for multi-morphic robots, and robots themself. The work of this thesis is concentrated on the implementation of the GAP and robots (model: DARwIn-OP) control. The GAP provides options for a variety of robots to users. The users can easily design their own robot scripts merely by drag-and-drop operating on icons representing the actions, behavior, and short scripts, respectively. Whenever a script is created or updated, GAP will automatically save the script as an XML file format internally. In addition, robots can be conducted to express their emotions orally by utter the lines composed. The system is demonstrated by a play of "do-as-I-do" and recoded in a video at YouTube:" http://www.youtube.com/watch?v=v8ErTOgAQSo".

Keyword: user interface, XML, DARwIn-OP, puppet show, robot controller

TABLE OF CONTENTS

PAGE

LIST OF FIGURES

Figure Page

LIST OF TABLES

I. Introduction

1.1 Motivation

With the advancement of Technology, the scene that robots and humans live together we seen at movies, it will be realized. Robots into the human life will dramatically change the current human lifestyle. For example, robot will do the repeated trivial work instead of human; humans cannot load the high-precision work will also use robots; even more, the robot will walk into the entertainment of human daily life.

Although the robots become more and more diverse, but the entertainment robot is still relatively unpopular. However, entertainment is an integral part of human beings. In recent years with population aging, Taiwan gradually become an aging society, accompanied by robots and entertainment robots by the people to pay attention; SONY Corporation of Japan has published a robotic pet dog which is named AIBO. AIBO is an imitation of a real dog, almost all dogs can do things AIBO also can be done. There are more and more entertainment robots; however, most of the functions of the robot are too monotonous, and users cannot design the robot action.

Operation of the robot, there is a great gap between developers and users. When people think of robots, the first thought may be very complex and difficult circuit;

second may be a large number of motor parameters; third possible is its user interface is overly complex, because there is too much function on its interface. Based on the above three points, most users may fear the field of the robot.

In this thesis, we aim for the drawback that the interface is too complex and the functions are overly monotonous, we provide a robot controller, the graphic authoring platform (GAP) of screenplays for robotic puppet shows which can control multi-vendor and multi-model robots. Users can design their own robot action, the interaction between the robots and the audiences, but also the robots can speak; more complete, the platform allows users to design their own scripts, after completing the script, and the robot will perform the script.

1.2 Objective

Although there are many types of robots, service, entertainment, commercial, intelligent robots and so on. However, most robot controller interface at the development phase is not taking into consideration the operational difficulties caused by the general public due to lack of professional knowledge.

The object of this thesis is to provide a robot control platform, which can be accessed by network anytime and anywhere. Also it can operate and set the robot moves the robot. Regardless of children, adolescents, adults or the elderly can easily operate the robot. Users can edit the user's exclusive script through the platform, the script can be exchanged with other users; the design of this platform will allow robots more active in this field of home entertainment, but also to allow different types of robots have an ability to accompany.

1.3 Organization of thesis

The architecture of the thesis is divided into five chapters. The first chapter explains the motivation and research objectives of this thesis, where we pick out the different between our platform and other robot controller. The second chapter describes the platform environment, as well as the types of the robot, and proposes what kind of robot suitable for application in the platform; also made reference to other robot interface, think about how the human-computer interface is applied in the platform we proposed.

The third chapter is system analysis and design, where we analyze the system, the graphic authoring platform (GAP) of screenplays for robotic puppet shows, through 5W1H analysis approximately. Then we use the Use Case and analysis model to do further analysis. In the analysis model, we design the class which used in the system, and then use flow chart to represent the entire operational process of the system. Finally, we illustrate the development tool and the overall concept of the platform. Chapter IV descripts the implementation detail and demonstrates the result.

Chapter five is conclusion and future works.

II. Background

In this chapter, we will introduce the environment of our system first, and the whole tools we used in our implementation. Second, we discuss about the robots which used at our system. Third, in the viewpoint of human machine interface, we discuss some interfaces of robotic operation in the market.

2.1 The Introduction of System Environment and Development Tool

In this chapter, there are two sections about the system environment and the development tool we used to implement the system.

2.1.1 System Environment (LAMP)

LAMP is composed of four words: Linux, Apache, MySQL, and PHP/Perl/Python. LAMP is the most popular and powerful website erection tool. As it's free and open source features, making LAMP quickly welcomed by a large number of internet users. At the same time, because the LAMP is so widespread, when users set up a website for the first time, they always can obtain a number of supports from the internet. Following, we will make a brief introduction of these four elements separately.

- **Ubuntu**

Ubuntu is an operating system based on Linux, the goal of Ubuntu is that provide general users with a stable operating system which is mainly constructed by free software. The following is the main features of Ubuntu:

- Hardware support pretty good
- High degree of freedom
- Plenty operating suites

The above all is the main reason that we choose Ubuntu as the executive environment of GAP. Most important thing is the high degree of freedom of Ubuntu offer users a probability to customize their own system. Simultaneously, the regular update of Ubuntu is also in our considerations.

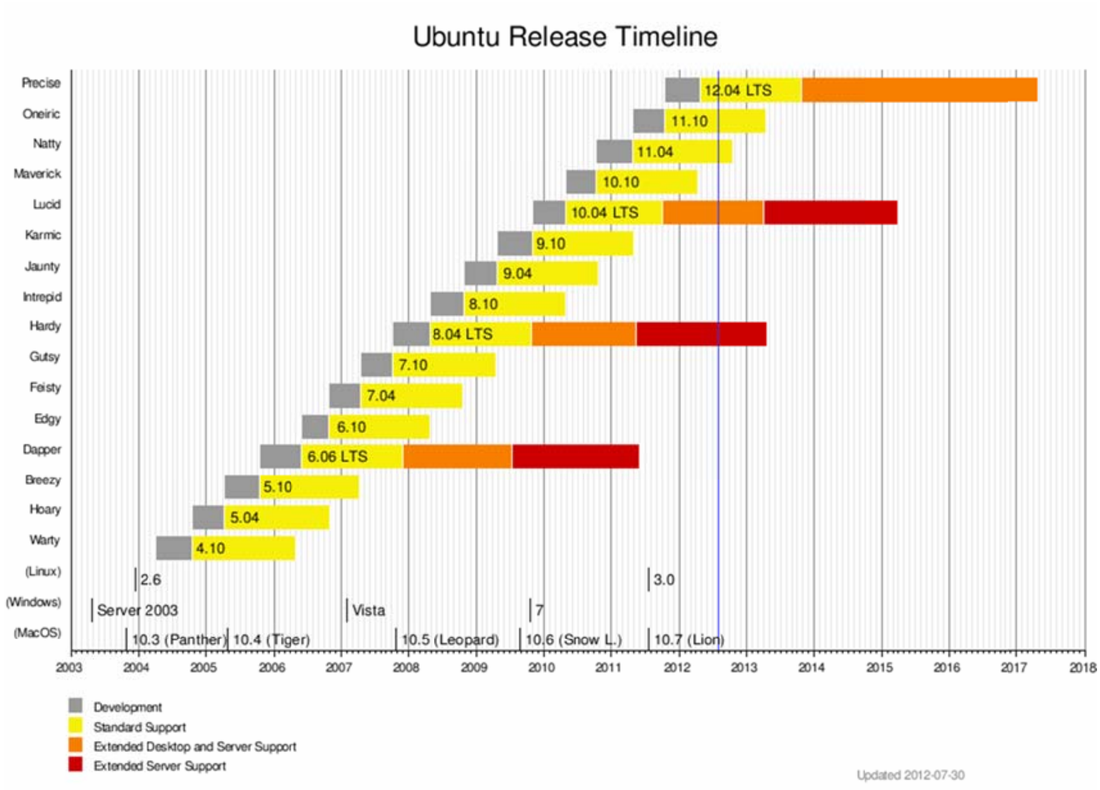


Figure 2-1 Ubuntu release time-line

From Figure 2-1, we could know that Ubuntu update frequently, so almost the system state is latest. Even though the version of the system no more update, users can upgrade their system, it can significantly reduce the problem of the incompatible version of software. Above all, the Ubuntu operating system as the environment of GAP is quite correct.

- **Apache**

GAP is erected on the Internet, taking into account the convenience of development and the stability of connection, the server of the website should be consider carefully. We compare the high utilization rate of several Web servers in the market.

Table 2-1 Comparison with web servers

	Usage rate	**Open Source**	**OS support**	**Latest release time**	**Developed by**	**Free**
Apache	69.39%	Yes	High	2012-04-17	Apache software Foundation	Yes
Microsoft IIS	15.6%	No	Windows only	2010-02-01	Microsoft	Yes
Nginx	6.67%	yes	Medium	2012-04-23	Igor Sysoev	Yes
LiteSpeed	0.94%	No	Medium	2011-05-20	LiteSpeed Techonology	no

From Table 2-1, we know Apache server is the most popular web server. In addition to the consideration of the utilization rate, when we choose our web server, we also should take account into the degree of support of operating system. More support of operating system will be more convenience if we want to transplant the system.

By the way, weather the vision of the web server updates regularly and the source code is free; these are all we should concern. Once the source code is free, even if the original function of the web server is not enough, developers can expand the function of the server by themselves. Above all, Apache is the server of GAP of the thesis.

- **MySQL**

Database offers web site a space to save any data, including the accounts and passwords of all users and the script user designed. After user logging in, the system obtains the script which the user created and other details from the database then shows this information on the web site. With the increase of users, as well as the accumulation of scripts, the data of whole system will become large and the management of the system will be too difficult. Therefore, a database is responsible for all data categories, so when the web page loads may also reduce the loading time. This is why the system needs a database.

Table 2-2 Comparison with databases

	Software license	**OS support**	**Max DB Size**	**Latest Release Date**	**Maintainer**
MS Access	Proprietary	Low	2GB	2010	Microsoft
Oracle	Proprietary	Medium	Unlimited	2009-09	Oracle Corporation
MySQL	GPL or Proprietary	high	Unlimited	2011-10-21	Oracle Corporation
SQLite	Public domain	High	32TB	2011-05-19	D.Richard Hipp
DB2	Proprietary	Medium	512TB	2012-04-30	IBM
MS SQL-Server	Proprietary	Low	524,258TB	2008	Microsoft

When we decide the database, there are some points we should concern. First, whether the database is open source while also free software. The characteristic of the open source allows developers to develop their own functions. By the way, the software license should be GPL or public. Higher degree of support of operating system, the difficulties encountered in the database transplant in the future will be less. The most important is the capacity of the database. On the other hand, the release time of the latest vision is the judgment to determine whether the official maintained continually. Above all, these are the conditions of database of GAP. From table 2-2 we could know MySQL is the best choice

- PHP

In the architecture of LAMP, the letter P usually stands for three types of programming language, PHP, Perl and Python. One of PHP library offers a number of functions to communicate with MySQL. Because of substantial support in PHP for MySQL, we choose PHP as a tool to communicate with the database in LAMP.

The entire name of PHP is PHP: Hypertext Preprocessor, it is a scripting language running on the computer. The main purpose of PHP is to deal with dynamic web pages which contain the command-line interface, or generate a graphical user interface (GUI). In general, PHP is usually running on a web server, it generates the pages which users browse by the implementation of PHP. With PHP and database, each user can enjoy their personal web pages. PHP is completely free, while the development team keeps updating the version of PHP.

Table 2-3 PHP release time

Version	**Release time**	**explanation**
5.2.10	2009/6/18	Fix bugs and security vulnerabilities
5.3.0	2009/6/30	Support namespace
5.3.3	2010/7/22	Add the class of namespace. member function is not a constructor if its name is the same with class
6.0.0	unknown	Support Unicode

From table 2-3 we could know the situation of the version update of PHP recently. Thus, LAMP has been generally introduced.

2.1.2 Development Tool of User Interface

In the web development, programming language used by the developers is usually JavaScript. But if we use JavaScript to implement an effect of web pages, it usually cause a huge code and difficult to maintain the program. The biggest drawback of JavaScript is different browsers have different support degree of JavaScript. As a result, if we want to show an effect, we should consider the browser user may use and aim at different browsers to development different programs. This is a great hardship of the developer.

jQuery is a cross-browser JavaScript library, it strengthen the operation between the HTML and JavaScript. jQuery for free and Open Source, its syntax design makes many operations will be easy. Such as the operation file (document), select the DOM element, as well as the development of AJAX.

In addition, jQuery provides an API to allow developers to own the development of functionality to jQuery. The following is the jQuery features:

- DOM elements cross-browser
- Convenient operation of the CSS
- AJAX(Asynchronous JavaScript and XML)

- High extensibility

These are main features of jQuery. Through jQuery, the number of code can be substantially reduced, users can easily reach the original effect; cross-browser features is also the perfect solution that developers need developed for different browsers. In addition, jQuery also has many free resources. Through the exchange of many developers, many of the functions do not require developers to write their own; jQuery version is updated frequently.

Table 2-4 jQuery release time

Version	**Release time**
jQuery-1.6.4	2011/09/12
jQuery-1.7	2011/11/03
jQuery-1.7.1	2011/11/21
jQuery-1.7.2	2012/03/21

2.1.3 The Communication between the Platform and Robots

There are many kinds of means of communication between the platform and the robot, whether it is a wired network or wireless network. Taking into consideration the appearance of the robot during the show, as well as mobile convenience, we choose a wireless network as a way to communicate with the robot in the thesis.

There are many different technologies of wireless network. The following will describe several wireless network technology can be used in theater performances, and why we choose the technology in the thesis.

Table 2-5 Comparison of three different wireless technologies

	Wi-Fi	**Bluetooth**	**ZigBee**
Range	50-100 meters	10-100 meters	10-100 meters
Networking Topology	Point to hub	Ad-hoc, very small networks	ad-hoc, peer to peer
Operating Frequency	2.4 – 5 GHz	2.4 GHz	2.4GHz (worldwide)
Complexity (Device and application impact)	High	High	low
Power Consumption (Battery option and life)	High	Medium	Very low(low power is a design goal)
Typical Applications	Wireless LAN connectivity	Wireless connectivity between devices	Industrial control and monitoring sensors networks

From table 2-5 we could know the transmission range of Wi-Fi is the furthest in terms of power consumption is the largest. In the viewpoint of spectrum, Wi-Fi, Bluetooth, and ZigBee are operating in the same spectral range. Wi-Fi is a kind of wireless technology. Although compared with other transmission technology, its power consumption is the biggest, bur the transmission speed is the fastest. More importantly, Wi-Fi generally use in public environments. Given the wide application of Wi-Fi around the world, we therefore selected transmission technology between the Wi-Fi as robots and the platform in the thesis.

2.2 The Robot of Theater Performances

In this section, there are two parts. First we will explant the principle when we choose the robots which apply in the robotic puppet show, and then we introduce the robot which we applied.

2.2.1 The Principle of The Robot

GAP provides a variety of different brands and different models of the robot to the user selection; this does not mean that all the robots are applicable in our system, therefore when we select the robots; the robots need to meet at least the following requirements:

- Any kind of wireless network module

The robot cannot be a single machine. Because it needs to accept instructions from the platform at any time, and it requires at least a wireless transmission technology to receive the instructions.

- Developers can develop their own functions

In accordance with the needs of the theater, developers can develop the special function for the theater performances. If the robot can only use the default number of functions, then the script content will be too fixed.

➢ Developers could design the robot moves

There are many different settings for each user. Thus if the developers can design their own actions, it will be a major focus of the script whether rich.

Table 2-6 Comparison with robots

	Wireless connection	**Open Source**	**Setting Actions by Developers**
Darwin-op	Wi-Fi	Yes	Yes
NAO	Wi-Fi	Yes	Yes
AIBO	Wi-Fi	No	Yes
Bioloid Premium kit	ZigBee	Yes	Yes
Robobuilder	No	yes	Yes

Table 2-6 is a comparison with multiple robots. From table 2-6 we could know that DARwIn-OP, NAO, and Bioloid premium kit all match our requirements. However, due to the communication technology of the Bioloid Premium kit is ZigBee. After considering the convenience of the developer’s development, we have decided to temporarily exclude the Bioloid Premium kit. Therefore, the robot used in the thesis is DARwIn-OP and NAO, as shown in Figure 2-2.

Figure 2-2 The robots of the system: DARwIn-OP (left), NAO (right)

2.2.2 Introduction of DARwIn-OP

DARwIn-OP as a performing robot, it matches the three principles we mentioned in section 2.2.1. DARwIn-OP built a personal computer and install Ubuntu 9.04 operating system. So whether it is in use or development of the DARwIn-OP, developers can quickly get started. DARwIn-OP used in theater needs to be able to walk, squat, or other more specific action. We can design action by the original software – Action Editor.

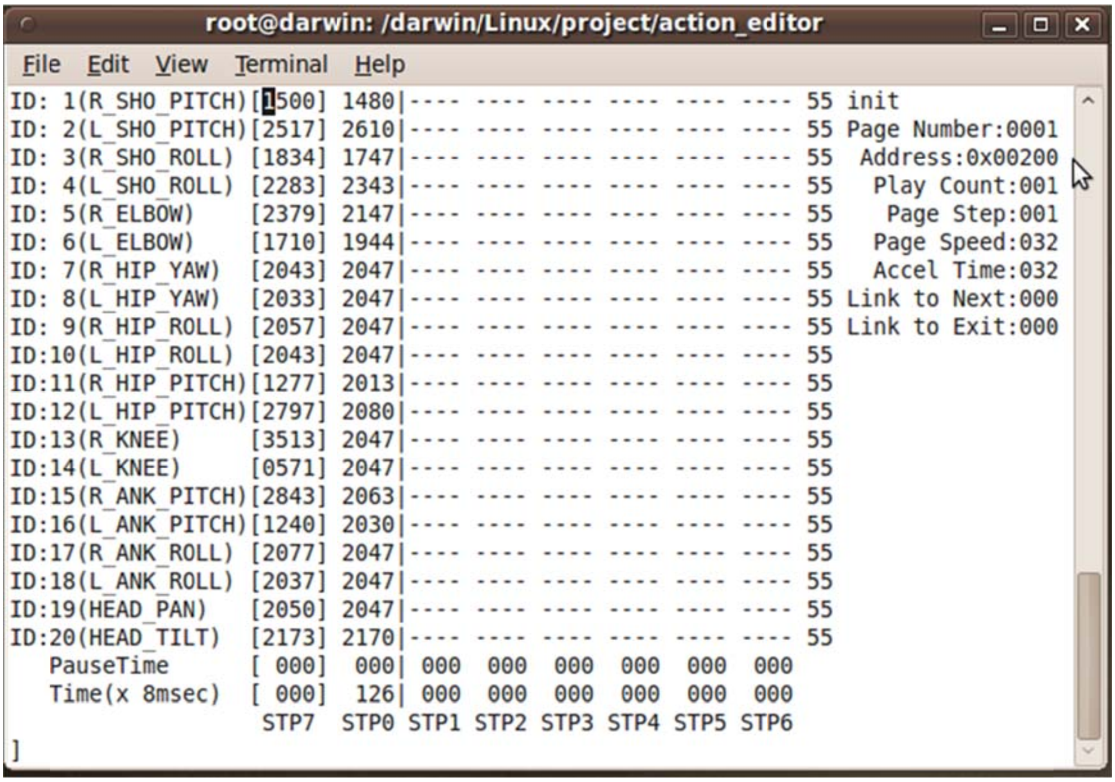


Figure 2-3 Original action editing software – Action Editor

In order to achieve the effect of changing the DARwIn-OP action, Action Editor allows developers to freely enter each joint motor parameter. Developers can freely set the action of the DARwIn-OP; let Action Editor reads into the motor parameters.

On the other hand, Because of the wireless capabilities of DARwIn-OP, developers only need to establish a socket, and then users can easily command to the DARwIn-OP through the socket from GAP.

2.3 Human Machine Interface

Human machine interface is a bridge of communication and interaction between the system and users. User-friendly buttons remind users to complete the objectives.

Through a friendly user interface, users can reduce the large number of explore time to complete tasks more quickly. The following will describe a few principles of human-computer interface design, and compare a variety of different robot operation interface.

2.3.1 The Principles of Human Machine Interface

In the implementation, because the purpose of GAP is available to users of all ages, in the design of user interface and viewpoint of users, we should strive to give users the most intuitive and cordial operating experience. The following lists a few principles of human computer interface.

- Visibility of system status

  Through the appropriate feedback of the system, users know the current position in the system any time.

- Match between system and real world

  Notice or syntax of the system should be consistent with the real world. For example, when users make mistakes in using the system, the system notice should clearly indicate the cause of the error.

- Error prevention

In the design of the system, we should try to avoid the possibility of the user operating error occurred. Such as the users' instructions are dangerous to the system, we should reconfirm in order to avoid irreparable error.

- Recognition rather than recall

  Allow users to have a clear understanding of all the buttons and functionality of the system is better than recall the appropriate mode of operation.

- Consistency and Standards

  The standards of the system should be consistent. The same meaning if the text should not appear a second time. Do not allow the users in the process of the system also need to think about the text may result in different fuzzy linguistic.

In fact, in the field of human computer interface of computer science, user feel should be the main consideration. For example, how to allow users to notice the system information, as well as the fluidity of the system, is human computer interface should be concern about.

2.3.2 Robot Control Interface

In this section, we will introduce the other robot control interface, as references during platform development. By the way, we also introduce some concepts about robot control interface.

Figure 2-4 Humanoid robot control interface

Figure 2-4 is a humanoid robot control interface. Developers operate the humanoid robot through the interface to complete the goal of the independent handling objects or carry objects, and so on. The interface of figure 2-4 provides many details of settings, including motor speed and the implementation of the mirror [1-3]. Although many options available for setting, but too much detail causes the inconvenience of the lack of professional background of users, this is a small drawback of this interface.

Figure 2-5 Virtual puppet control panel

In Figure 2-5, the developer uses this interface to design the action of the virtual shadow puppets [4]. In Malaysia, the shadow play is a traditional culture. The developers hope that more users can design their own virtual shadow puppet action through this interface, so the traditional culture will become increasingly popular. However, this interface provides only the joint angles of dolls set; users cannot see the puppet action, slightly less the convenience during operation.

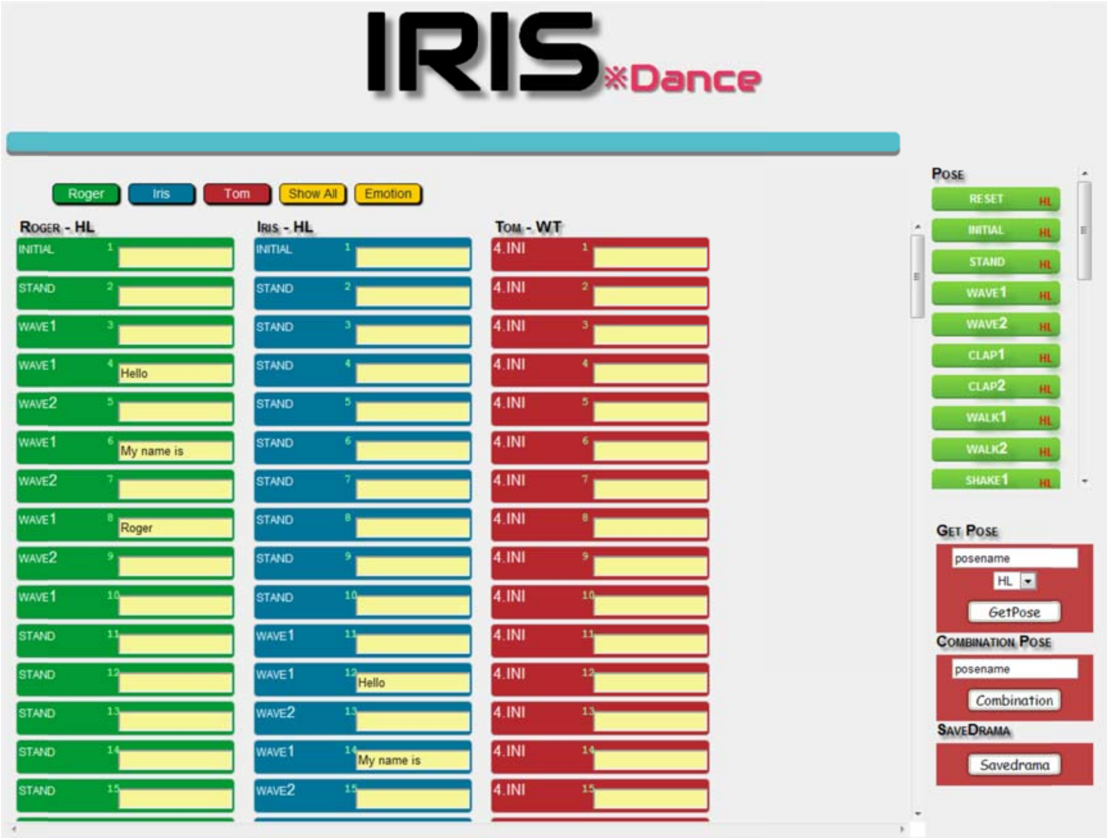


Figure 2-6 Screenplay platform interface

Figure 2-6 is another robot action editing interface of robot platform. This interface provides users with a custom action, and the script can be downloaded to the user's mobile device [5-7]. In [6], this thesis provides a platform which could control robots. Users can design robot actions through this platform; each robot has a smartphone as his brain; when users designed a script, the platform will sent the script to the smart phone, and the smart phone will be responsible for controlling the robot performance. [5] extends the research of [6]; [5] improved the platform of [6], and make the user interface more friendly; while [6] improved the synchronization strategy between the multi-robot, therefore the robot performance is closer to reality.

But the drawback is that when the first use, users need to spend more time to become familiar; that is to say, the lack of friendliness of the system. This interface is improvement from reference, the continuation of its framework for a deeper discussion. The differences between [5, 6] and the thesis are that the platform of the thesis could control several multi-morphic robots, developers could extend other robots, and the biggest difference is that the platform control the robots directly.

According to the above-mentioned several robot control interfaces, we make a conclusion about the user interface we will develop:

- Intuitive user interface
- Appropriate robot detail settings
- Excellent user prompts

Above all are the points we should consider when we implement GAP. In fact, the robot control interface is also a visual programming language [8, 9]. The underlying codes packaged as an icon or a button, the users can quickly assemble pictures or buttons to complete a series of robot set through an easy operation; this design is not uncommon, for example, such as the Lego robot programming interface [10].

2.4 Related Work

In this section, we will introduce a number of topics related to the graphical script editor, for example, drama management, human-robot interaction, and robotics research.

In [11], Nelson M.J. et al. propose a mechanism to coordinate modifications, which is called declarative optimization-based drama management; it could easily scale the script. By the way, drama could also be a discussion and educational tool [12, 13]. Through the drama, audiences could be more into the situation; the same, the learning effect of children is better than traditional education.

In the robotic research, there are a lot of challenges about developing robots. Some people try to implement an intuitive and easy way to design the pose of the robot, such as [14] it captures the motion of human beings, and robot can represent the motion; some people try to use different device to driven robots [15], they implement a mobile device to control the robot.

The most important is the system of robots should be an open system, if the system is open users could design their own functions. In [16], developers make a robot called DARwIn-OP which builds in a personal computer.

III. System Analysis and Design

In this chapter, we will introduce the architecture of GAP. First of all, through the 5W1H analysis method to do the preliminary system analysis, then use the Use Case and Analysis Model to do further analysis of the system. Then introduce the class of GAP implementation, and the last will be the system flow chart to represent the overall operation of the system process.

3.1 System Architecture

The system, the screenplay based performance platform of Robotic puppet shows (SBPP), is divided into three parts: the graphic authoring platform (GAP) of screenplays for robotic puppet shows, The development of screen play interpreter (SI) for multi-morphic robots, performing robot; the robot can be divided into DARwIn-OP and NAO. This paper focuses on the web platform and DARwIn-OP.

3.1.1 System Architecture Analysis

Typical robot and autonomy architectures are comprised of three-levels [18-21]: Planner, executive, functional.

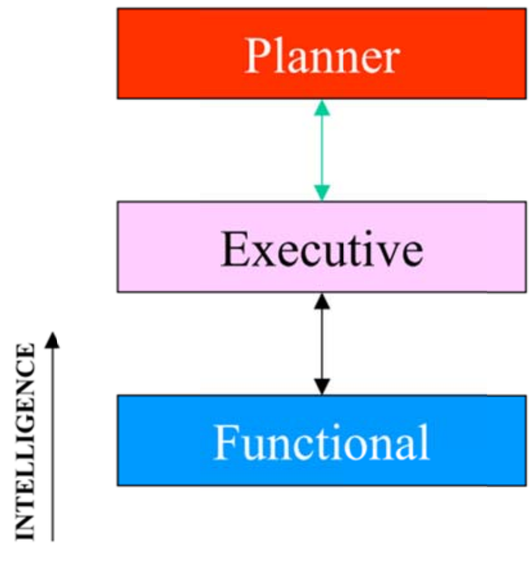


Figure 3-1 Typical three-level architecture

- Planner: Users make a decision at this stage.
- Executive: Responsible for the implementation of the command generated by the planner stage.
- Functional: According to the Executive stage, the functional layer provides a variety of functions to complete the command.

According to typical robot and autonomous architecture, and it corresponds to SBPP we provide, the diagram is as follows:

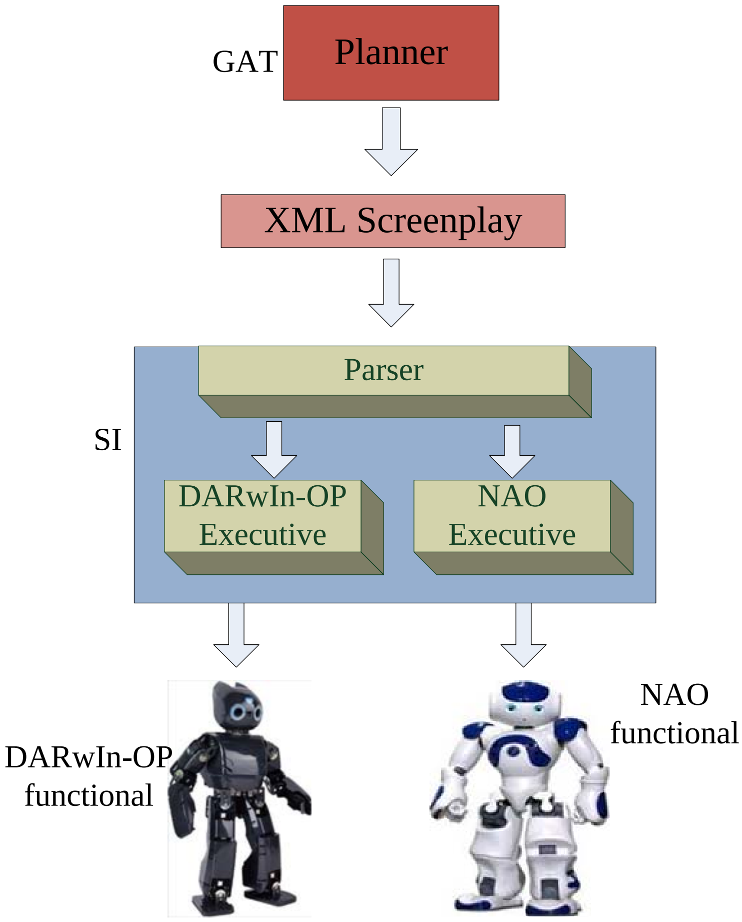


Figure 3-2 CLARAty adapts in the system

Figure 3-2 means that CLARAty architecture corresponds to SBPP. The Planner corresponds to user interface of SBPP. The user designs his script on the user interface, and then the script is stored as XML files. SI parses XML script, and then it transmit command to the executive layer; finally select the function which corresponds to the executive layer according to the command.

3.1.2 Introduction of System Architecture

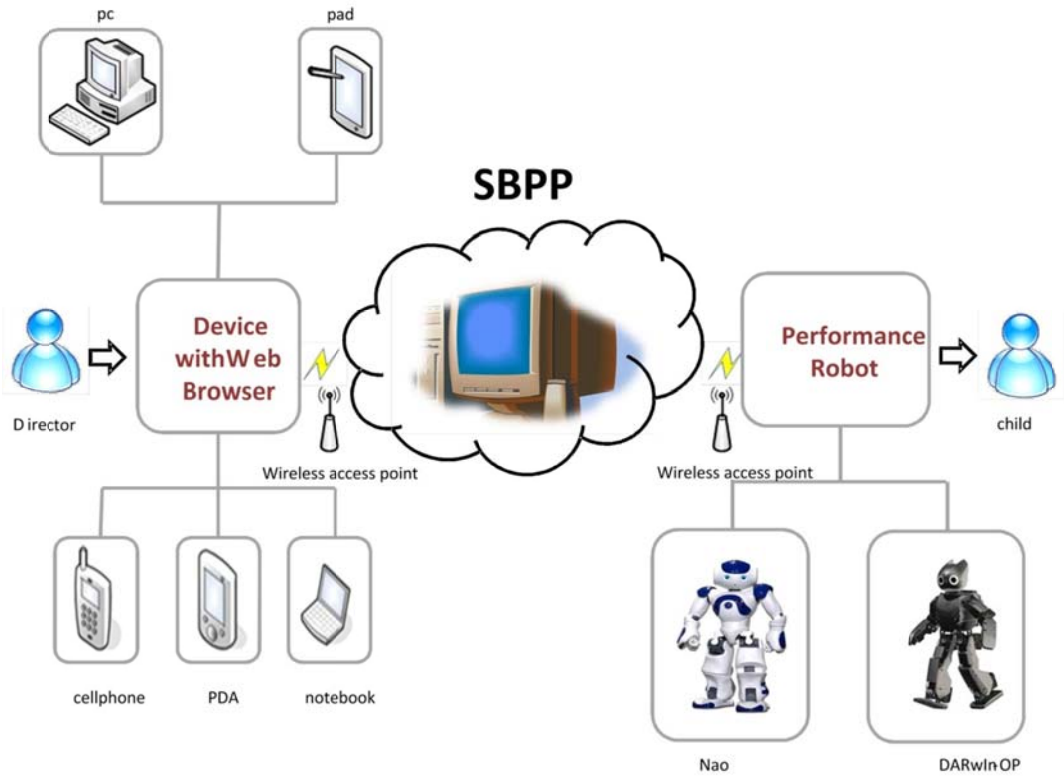


Figure 3-3 SBPP architecture

Figure 3-3 is the architecture diagram of SBPP. Users through any mobile device, such as smart phones, notebook computers, PC, connect to the Internet to enter GAP. GAP erected on the network may be exempted from the user to install software, and can be used at any time. User designed a script and sent the script, and then they can enjoy the show brought by robots. GAP provides users with three functions: script editing, script mergence, and script performance. These three functions will be descripted later.

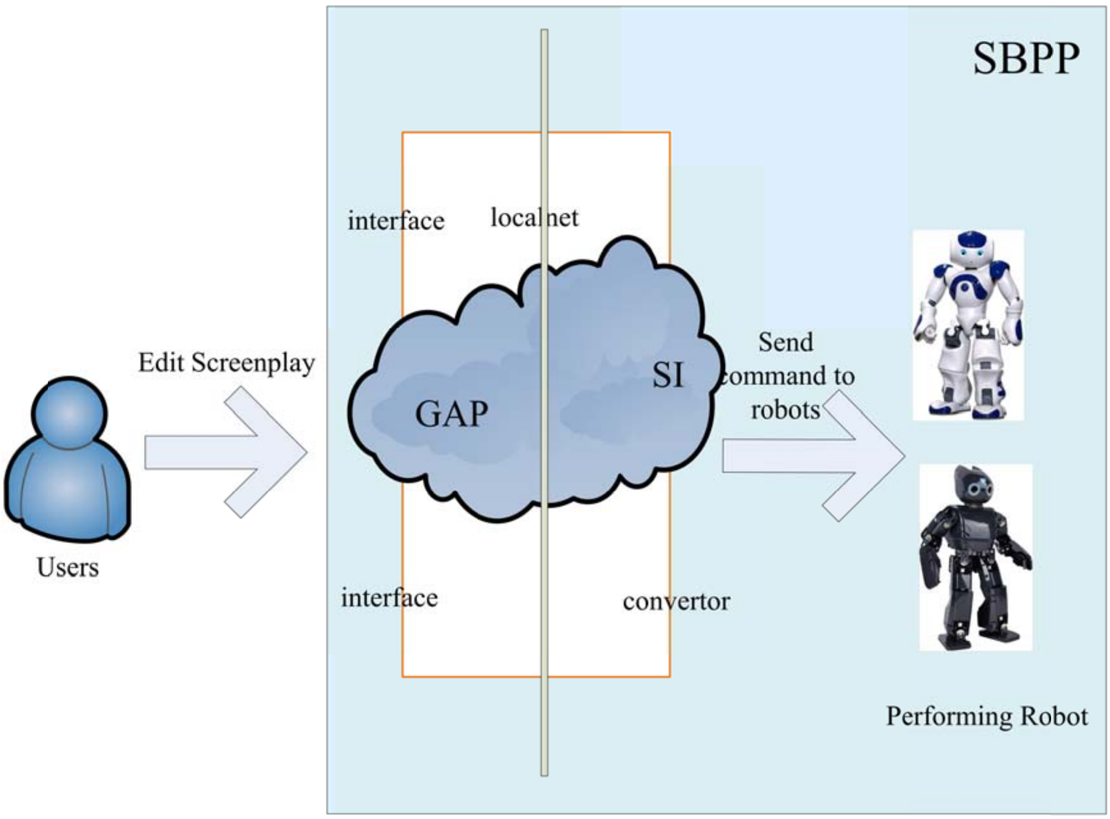


Figure 3-4 SBPP components

From figure 3-4, we could know the components of SBPP. Users edit scripts on GAP, and the GAP send the XML scripts to SI. Last SI sends commands to performing robots.

3.1.3 Noun of GAP

This section we will define the proper nouns of GAP, in order to avoid confusion when users operate GAP. GAP provides the option of motion and behavior; therefore we define pose, motion, and behavior here, figure 3-5.

- Pose: a constant posture.
- Motion: the transition between two poses.
- Behavior: composed of two or more motions.

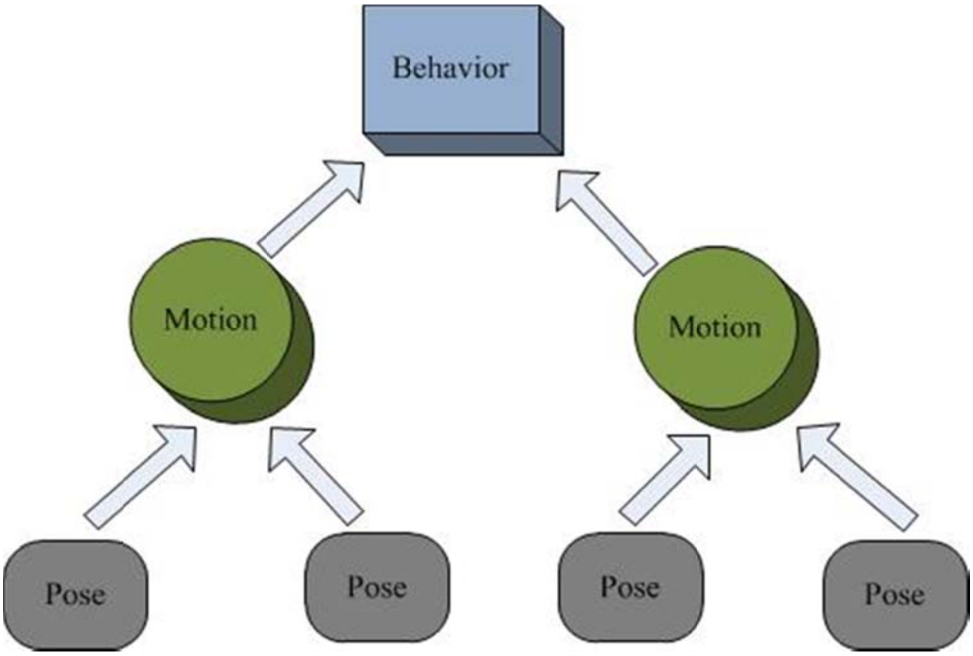


Figure 3-5 Relationship between pose, motion, and behavior

3.2 System Analysis

This section we will use 5W1H analysis to analyze GAP, and in accordance with the analysis of software engineering process, step by step analyze functional requirements, nonfunctional requirements, use cases and analysis models of GAP.

3.2.1 5W1H Analysis

We will use 5W1H analysis to analyze GAP, and make a conclusion in table 3-1. The first W stands for why. Corresponding to the thesis, it means why we want to realize GAP for users who are interesting with robots. The foundational reason is that robot controllers in the market are too complex to use. GAP hides the details which users do not need to know, and then users have a good experience for the robot.

The second W means what; GAP provides what kinds of functions to users. As a screenplay editing platform, first of all we need to provide the basic set of scripts, such as actors, actor’s lines and actor’s motion. Furthermore, a good script may be reused; scripts mergence let the script reusable and reduce the design time of users. If a script as an element, scripts mergence could be regarded as a modular process. Finally, after setting the script, user send the script to perform, and the most intuitive interface will reduce the time users to explore.

The third W means where users can use GAP. As long as the users have a device to connect to network, regardless of users’ location users can operate GAP anywhere. The fourth W said that when users can use GAP. When the user wants to enjoy a robotic show, or when the user wants to share the script with friends, they can always connect to GAP. Last W stands for whom. H represents that how to use this platform, GAP requires only a mouse and keyboard for operation.

Table 3-1 5W1H analysis

5W1H	Conclusion
Why – what is the purpose of GAP	A new entertainment To attract more potential users interested in the robot More convenience for operating robots
What – what kind of function of GAP	Script Editing Scripts Mergence Scripts Performance
Where – where user can use GAP	Anywhere, as long as network
When – when user can use GAP	Anywhere, as long as network
Who – who is GAP user	Anyone interested in robots
How – how to operate GAP	Keyboard and mouse

3.2.2 Functional and Non-functional Requirement

- **Functional Requirement**

GAP provides three major functions: script editing, scripts mergence, scripts performance. The first function provides the users create their own script; the script contains the sequence of robotic events, robotic types, and the lines of the robot, the action or behavior of the robot. The second function is scripts mergence, the user can freely select more than one script, and they will be combined into a new script. The third function provides the robot to quickly show the script, the user can select three scripts at most, and drag them to the script performance area, and therefore users just need click one button to enjoy the show.

Table 3-2 Functional requirement

Functional Requirement
Script Editing.
Scripts Mergence, a modular process.
Scripts Performance.

- **Non-functional Requirement**

The user group of GAP is the one who is interested in robots, and the interface of GAP should be popular. The three major functions of GAP are three-pages separately. Even if the order is fixed, users can free access them. On the other hand, there is only three scripts on the screen once, therefore the screen will be clearer.

When user designs the script, each process will appear step by step, so user will not be confused. We use pictures to represent the movements of the robot, and then user could understand the means of the movements of the action. We make a conclusion in table 3-3.

Table 3-3 Non-functional Requirement

Non-functional Requirement
The background image should be able to represent GAP
All information put in the same forum
Operation order is not fixed
All scripts do not appear together to avoid the confusions of users
The processes of script editing appears in order
Pictures represent the movements of the robot
Static picture represents motion; Dynamic pictures stands for behavior

3.2.3 Use Case Analysis

The major user of GAP – anyone interested in robots, as shown in figure 3-6. GAP provides three major functions. The first feature: script setting; users follow the step of GAP, and they can create or delete the script easily. The second feature: scripts mergence; each predetermined script as small modules through this feature the scripts which users selected into a great script. The third function: scripts performance; the robots can play three scripts at most, but the number of actually performing scripts is no limit if user uses second function – scripts mergence.

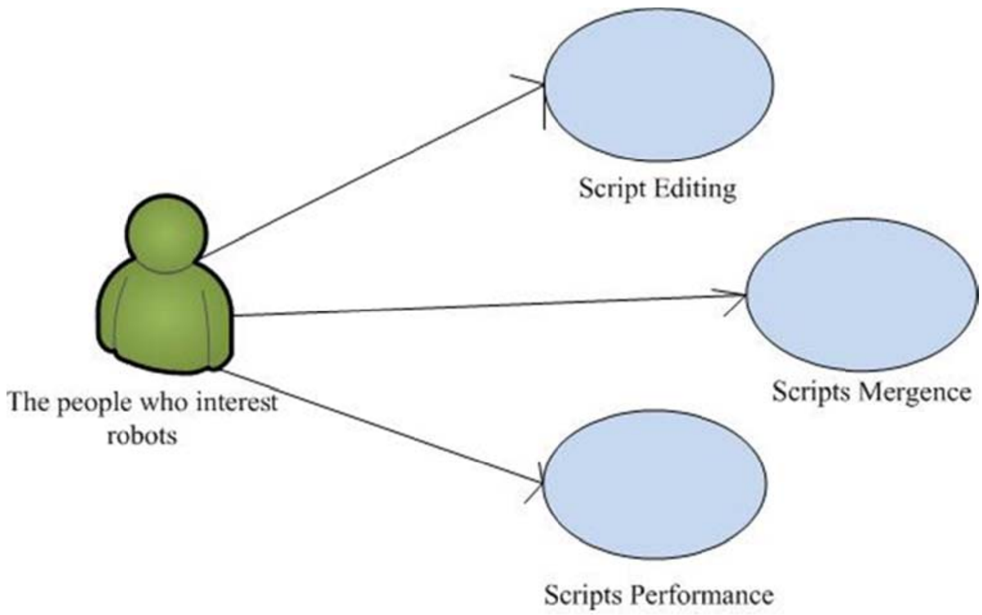


Figure 3-6 Use Case of GAP

There are three functions of GAP in the figure 3-6. In the first function – Script editing, the user sets the details of the script, such as performing the sequence of robotic events, robotic types, the robot lines, the robot's actions or behavior. Table 3-4 lists the feedbacks of GAP which correspond to above steps.

Table 3-4 Use Case – Script Editing

<table>
<tr><td>Use Case Name</td><td colspan="2">Script Editing</td></tr>
<tr><td>Summary</td><td colspan="2">Users set the details of the scripts</td></tr>
<tr><td>Actor</td><td colspan="2">Anyone who interested in robots</td></tr>
<tr><td>Precondition</td><td colspan="2">Make sure available to network
Register Account</td></tr>
<tr><td rowspan="5">Description</td><td>Actor Actions</td><td>System responses</td></tr>
<tr><td>Set the sequence of robotic events</td><td>Marked the sequence with special color</td></tr>
<tr><td>Choose robotic types</td><td>Enlarge the picture of robot and the frame is red</td></tr>
<tr><td>Type robotic lines</td><td>Show the lines on the screen</td></tr>
<tr><td>Select motion or behavior of the robot</td><td>Enlarge the picture of robot and the frame is red</td></tr>
<tr><td>Post-condition</td><td colspan="2">After editing, GAP increases the set to database；this is a complete set of robotic event</td></tr>
</table>

The second function, Scripts Mergence, provides the user a function to merge the scripts. Each script can be regarded as a module, each module are able to freely combination with other modules to form a bigger script; the scripts are reusable, but also can extend infinitely. Table 3-5 describes the use cases for Scripts Mergence.

Table 3-5 Use Case – Scripts Mergence

<table>
<tr><th>Use Case Name</th><th>Scripts Mergence</th></tr>
<tr><td>Summary</td><td>User could select several scripts and merge them to a bigger script</td></tr>
<tr><td>Actor</td><td>Anyone who interested in robots</td></tr>
<tr><td>Precondition</td><td>Make sure available to network
Register Account</td></tr>
<tr><td>Description</td><td><table>
<tr><th>Actor Actions</th><th>System responses</th></tr>
<tr><td>View the content of the script</td><td>When mouse over the picture of the script, the content of the script will show at right corner of screen</td></tr>
<tr><td>Select the scripts you want to merge</td><td>User drag a script and drop it in the mergence area</td></tr>
</table></td></tr>
<tr><td>Post-condition</td><td>After selecting, user types a script name, and decides how many robots in the script, last make a decision about privacy</td></tr>
</table>

The third function is Scripts Performance. This feature provides a simple ability to performance the script, when the user chooses the script you want to play and drag it then drop it to the performance area; user clicks the "camera" button to send the script to the robot for performance. Table 3-6 describes the use case for Scripts Performance.

Table 3-6 Use Case - Scripts Performance

<table>
<tr><td>Use Case Name</td><td>Scripts Performance</td></tr>
<tr><td>Summary</td><td>Robots once perform three scripts at most</td></tr>
<tr><td>Actor</td><td>Anyone who interested in robots</td></tr>
<tr><td>Precondition</td><td>Make sure available to network
Register Account
Make sure there is a script at least</td></tr>
<tr><td>Description</td><td><table>
<tr><td>Actor Actions</td><td>System responses</td></tr>
<tr><td>View the content of the script</td><td>When mouse over the picture of the script, the content of the script will show at right corner of screen</td></tr>
<tr><td>User select the scripts you want to enjoy</td><td>User drag a script and drop it in the performance area</td></tr>
</table></td></tr>
<tr><td>Post-condition</td><td>After checking the scripts which user selected, click “camera” button, and then the robots perform the scripts</td></tr>
</table>

3.2.4 Analysis Model

Class diagram of GAP is shown at figure 3-7. First, the users connect to GAP, there are five pages in the GAP, index, screenplay list, script editing, scripts mergence, and scripts performance. In GAP, the users could create, delete, or edit the script, and they also could merge several scripts to a new screenplay. The most important is they could send the scripts to robots for performance through GAP.

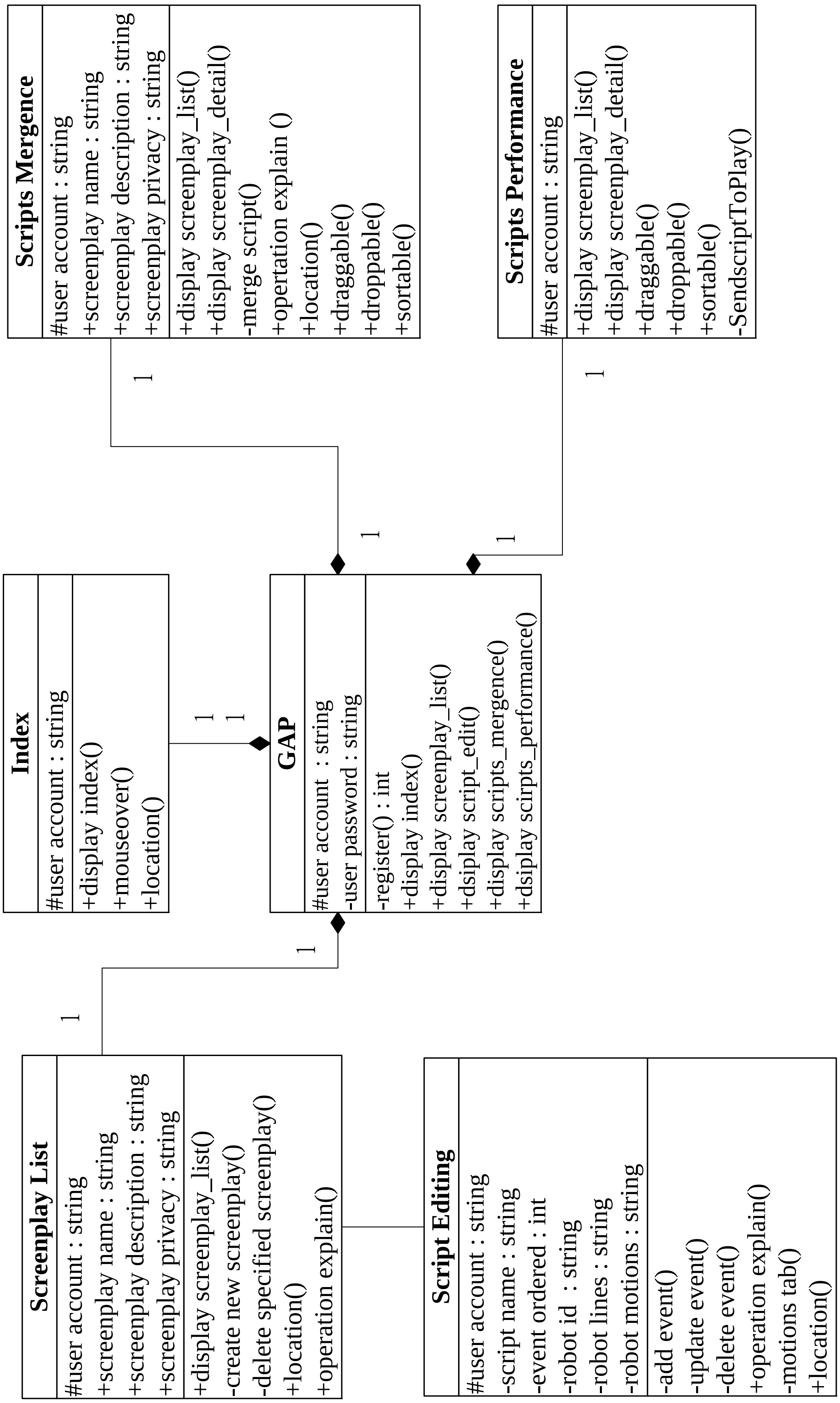


Figure 3-7 Class diagram of GAP

3.3 System Procedure

In this section, there are two sections. We will use flow chart to explant the procedure of GAP and DARwIn-OP briefly.

3.3.1 GAP procedure design

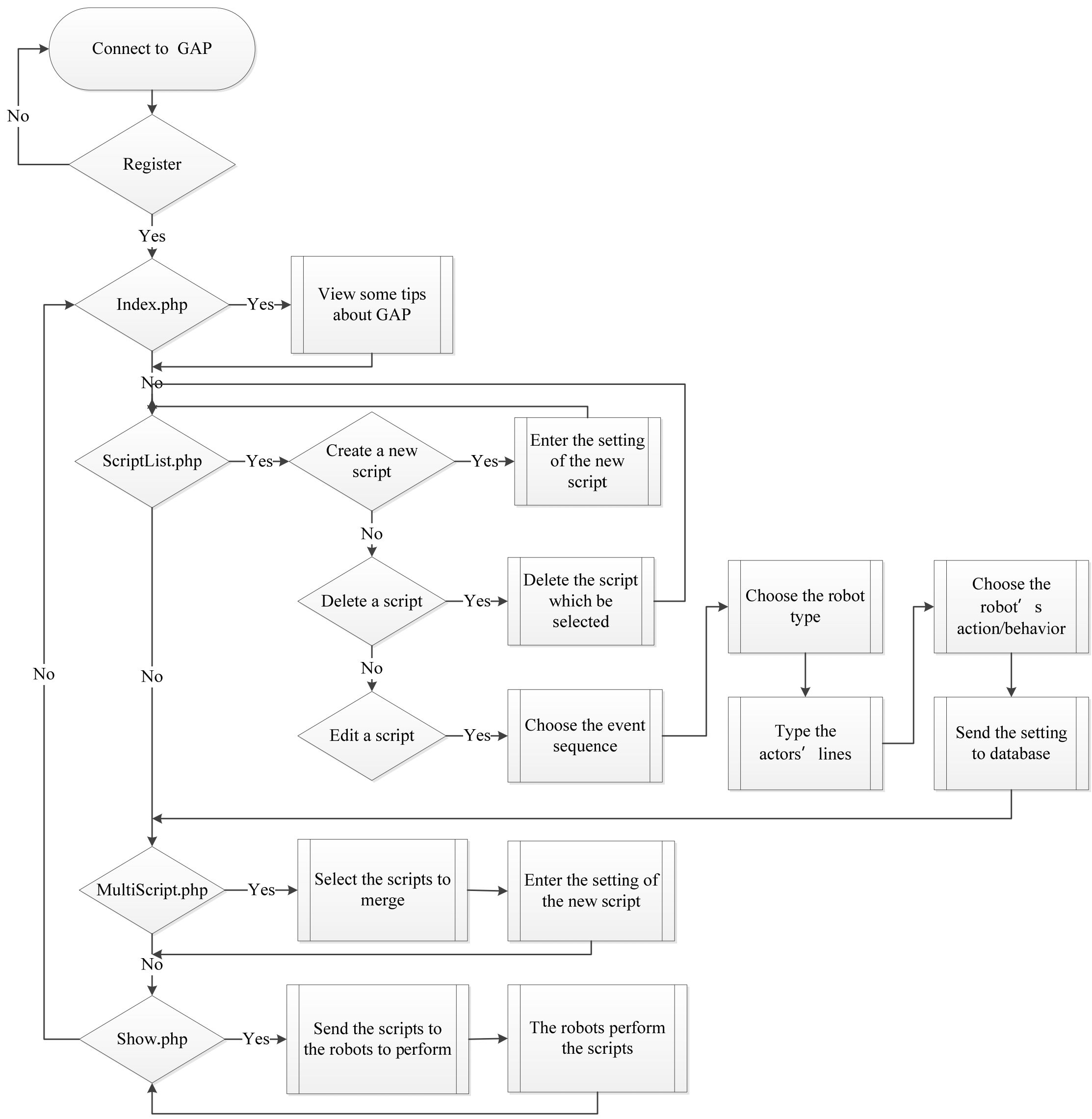


Figure 3-8 GAP flow chart

Figure 3-8 is the flow chart of GAP. In the beginning, the user connects to the login page of GAP, and then if the user already has an account to log in, the user does not need to register again. After log in page, the user will go into the index page of GAP. After reading the tips, user goes to scripts list page, and this page provides user to create, delete, or edit the script. After script editing, if user wants to use the function, scripts mergence, he can experience it immediately, or user goes to scripts performance. Finally, user selects the scripts to send to the robots, the robots perform the scripts.

3.3.2 DARwIn-OP procedure design

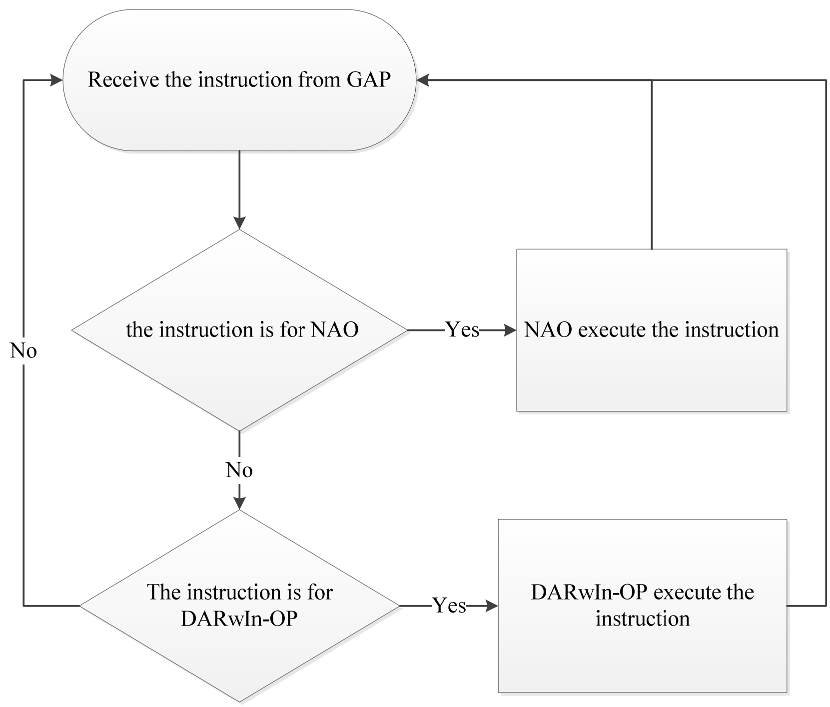


Figure 3-9 Procedure of robot performance

Figure 3-9 is the procedure of robot performance. First SI judges the instruction belong to which robot, if the format of instruction is correct then the robot start to execute the instruction. After completing the instruction, the state of the robot transit to wait for next instruction.

3.4 The Architecture of GAP and Robot

The overall system architecture of the thesis is divided into three parts, the graphic authoring platform (GAP) of screenplays for robotic puppet shows, the control end of the robot and the performance end of the robot. GAP provides some functions about script, such as script editing, scripts mergence, scripts performance, account registration, and the privacy of script. The control end of robot is responsible for receiving the instruction from SI, and send commands to the robot; when the performance side of the robot receives the instruction from the control side, robot perform the instruction. After completing performance, robot waits for next instruction.

3.4.1 Design of GAP

In GAP, the user group is who interested in the robot; therefore, the interface of GAP should reduce robotic knowledge as much as possible. First, taking into consideration the possibility of personal script, user should register an account before they operate GAP. Then user goes to index page, each function is listed in upper right corner, and followed by script editing, scripts mergence, scripts performance. When user wants to edit a script, user sets simple settings about the script first. For example, the script name, the script description, and types of robots. In the process of editing script, whenever a script is created or updated, GAP will automatically save the script as an XML file format internally and save the script to database.

Above all, we make a conclusion about the functions of GAP:

- Account registration
- The privacy of script
- Script editing
- Scripts mergence
- Scripts performance
- XML script

3.4.2 XML Script

The full name of the XML is the Extensible Markup Language, and the language used by a markup data. Its purpose is used to transfer and carry data, rather than display information. The main function of XML is as follows:

- Rich Documents：customize the content of document
- Metadata：descript documents or information on Internet
- Configuration files：descript the parameters of software

The great benefit of XML is label customization; through a custom label, we could structure the content of XML, and we know the content meaning from labels. In the standard architecture of XML, there is only one root element, and the next level elements are root sub-elements. Figure 3-10 is a simple example of XML.

```
<?xml version="1.0" encoding="UTF-8"?>
<message>
        <receive>John</receive>
        <send>Emily</send>
        <title>hello</title>
        <content>Long time no see</content>
</message>
```

Figure 3-10 A simple example of XML

In the first column of Figure 3-10, "xml version = '1 .0 '" stands for this file is a XML file, it would tell the browser or parser that the file should be resolved in accordance with the XML rules; the file is encoded in UTF-8. “message” is the root element of the example, and other labels are “message” sub-elements; they are siblings.

In the implementation of the thesis, we have an XML script file format, and customize the labels, in order to achieve the object of the transfer script.

3.4.3 Communication between GAP and Robots

As reference in section 2.1.3, the communication between GAP and robots is based on wireless technology, and apply Wi-Fi wireless transmission technology; packet transmission can be divided into TCP and UDP. In short, TCP provides reliable data transmission service, while UDP provides a non-reliable data transfer service. Table 3-7 for a comparison table of TCP and UDP.

Table 3-7 Comparison with TCP and UDP

Protocol	Advantage	Drawback
TCP	Reliable transmission, reliable mechanism can be omitted.	Lower transmission speed
UDP	Faster transmission speed, and larger amount of transmission than TCP	Non-reliable transmission, and developer needs programming reliable mechanism

The packets of robots are small, but we should avoid packet loss, and therefore we choose TCP as packets transmission protocol. The situation of packet transmission is shown in figure 3-11.

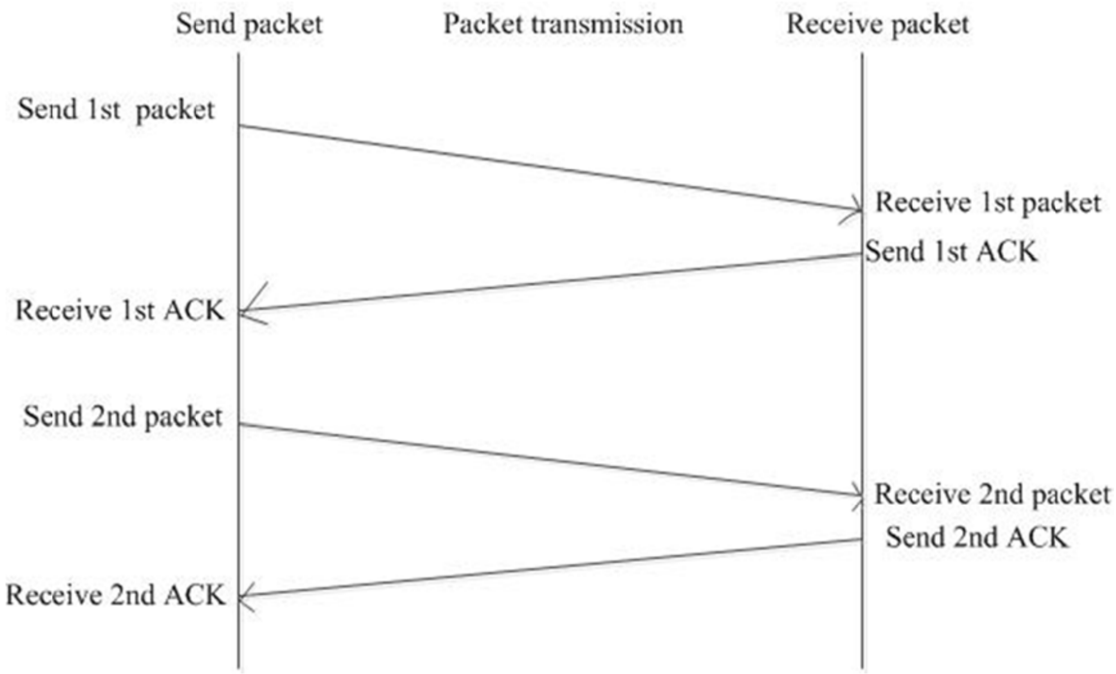


Figure 3-11 Packet transmission in TCP

From figure 3-13, we could know that every time sender sends a packet to receiver, receiver will send an ACK to the sender when he receives the packet; if sender get the ACK, he will send the next packet. TCP also supports packet retransmission, but TCP does not guarantee the order of packets.

IV. Implementation

Before the system implementation, first we must determine the system environment, and the environment has been discussed in section 2.1.1. The graphic authoring platform (GAP) of screenplays for the puppet shows is erected on Ubuntu 12.04, and it contains a database which is responsible for storing data; the robot control side is also built on Ubuntu 12.04, and it is responsible for sending commands to the robot. We choose DARwIn-OP as perform robot, and the development environment of DARwIn-OP is also Ubuntu.

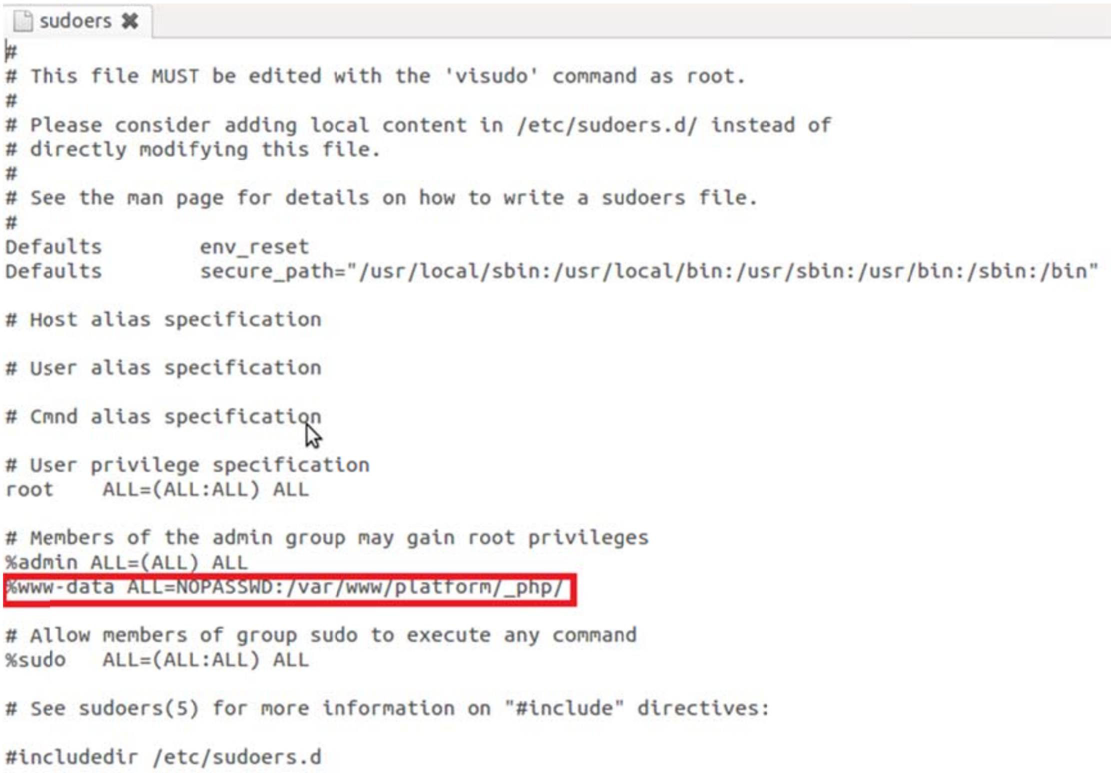
4.1 Setting of GAP Environment

- **Code::Blocks**

Code::Blocks is an open source IDE (Integrated Development Environment), but also has the ability to cross-platform (Mac OS X, Linux, and Windows). Code::Blocks supports multiple compiler such as GCC/G++; Code::Blocks also supports programs written by other IDE; for example, the program written by visual studio 2008 could be import to Code::Blocks; this feature can reduce the limitations of the development environment. We use Code::Blocks to implement the control end of the robot in the thesis.

- **User rights**

The architecture of the thesis is that user command to robot for execution from GAP, so it is necessary to release the user's privileges. In the original setting of web pages, OS do not allow user to execute shell script, and the reason is to avoid users inadvertently or maliciously damage the operating system. In the implementation, we drive NAO by executing shell script, so we must modify user rights.



```
sudoers
#
# This file MUST be edited with the 'visudo' command as root.
#
# Please consider adding local content in /etc/sudoers.d/ instead of
# directly modifying this file.
#
# See the man page for details on how to write a sudoers file.
#
Defaults        env_reset
Defaults        secure_path="/usr/local/sbin:/usr/local/bin:/usr/sbin:/usr/bin:/sbin:/bin"

# Host alias specification

# User alias specification

# Cmnd alias specification

# User privilege specification
root    ALL=(ALL:ALL) ALL

# Members of the admin group may gain root privileges
%admin ALL=(ALL) ALL
%www-data ALL=NOPASSWD:/var/www/platform/_php/

# Allow members of group sudo to execute any command
%sudo   ALL=(ALL:ALL) ALL

# See sudoers(5) for more information on "#include" directives:

#includedir /etc/sudoers.d
```

Figure 4-1 User rights modification

In figure 4-1, there is an instruction draw focus. In the Linux, each web user account is called “www-data”, and we should release the right of web directory to ”www-data” through the instruction focused in figure 4-1. Therefore all shell scripts we need should put in the web directory, and we could drive NAO.

4.2 Implementation of GAP

4.2.1 Database Setting

Before we construct the web pages, first we need to consider what information should be stored in the database, and what the type of information is.

Figure 4-2 Content of database

The content of database is shown in figure 4-2. First we create a database which called “iris”, and it contains four tables, behavior, robot, screenplay, and user. The behavior table stores the name and behavior which create by users and create time; the robot table contains models of the robots, robotic movement, the motion speed, robotic lines, the motion belongs to the script, performing order; screenplay table stores details of the scripts, the script name, description of the scripts, models of the robot, script privacy, scripts establish time; the user table stores all accounts and passwords.

4.2.2 GAP Function Implementation

- **User Registration**

When the user logs in to GAP first time, you must register for an account.

Figure 4-3 Log in page of GAP

Be repeated password twice after user types account to ensure the correctness of the password. After sending account and password, the server will check whether the account is existed, if account has existed, the system will require users to re-enter the account.

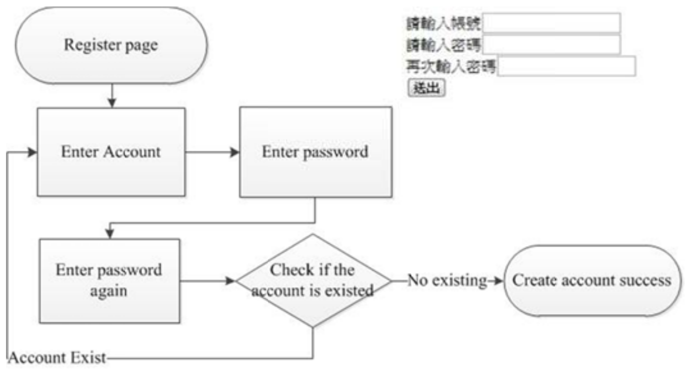


Figure 4-4 Register flow chart

- **User tips**

After user registration, user goes to index page. In this page, GAP will introduce user all function about GAP, if he/she moves the mouse over the icon. The index page is shown at figure 4-5.

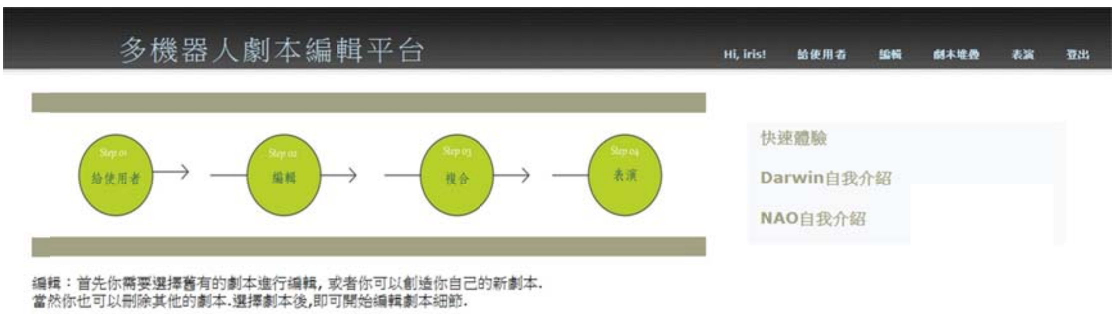


Figure 4-5 Index page of GAP

- **Script privacy**

Figure 4-6 Script privacy setting

When the user creates a script, there will be an option whether offer the script to other users, as shown as Figure 4-6. If the user chooses public, other users can see this script and edit the screenplay; otherwise, there is no anyone can see the script except the author.

- **Script Editing**

Figure 4-7 Script list of GAP

When user clicks the “Edit” button, you will see the scripts can be edited in the database, as Figure 4-7. There is total number of scripts on the script list, and show the scripts at the bottom. Each box is a script which includes picture, script name, and script description. Click left or right arrow could view other scripts. In this page, user can delete, create, or select a script to edit.

To delete a script, user needs to choose the script first, and click the “delete” button. If user wants to edit a script, user chooses the script as usual, and click “select” button to edit.

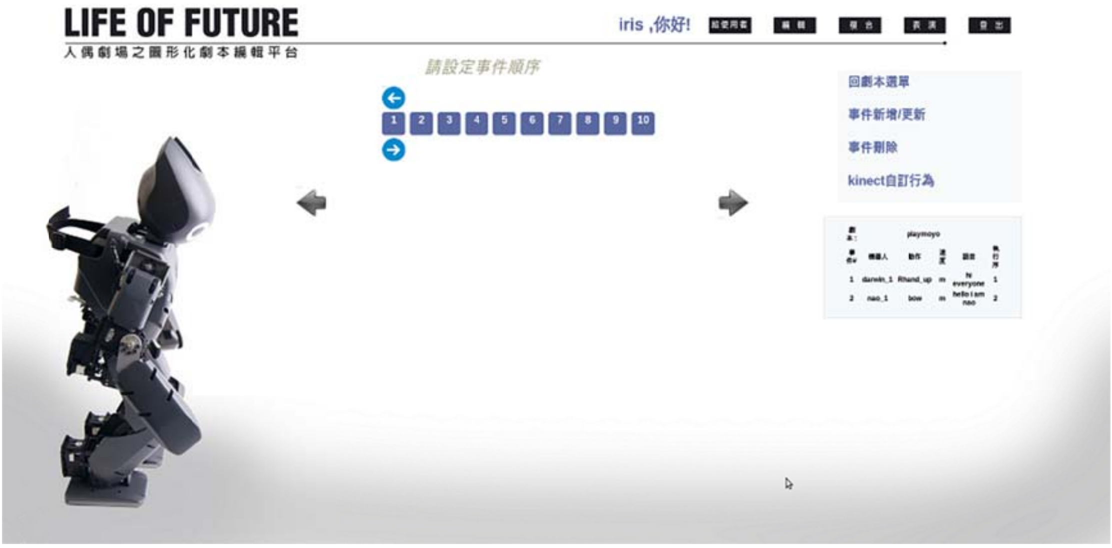


Figure 4-8 Set performance order

When user goes into the edit page, they see the events order setting bar first, as Figure 4-8. User should choose an event to edit at this step; each box is an event. Script performances order in accordance with the sequence of events set by the user. When hovering over an event, the background color of event will change to remind user. If the user clicks on the event, the field background color will be different; it is an evident to check that user selected the events. To delete unnecessary event, you just click the event you want to remove, and then click the "delete" button, you can easily delete the event. Complete the sequence of events, and then user need to select the performance robot of the event.

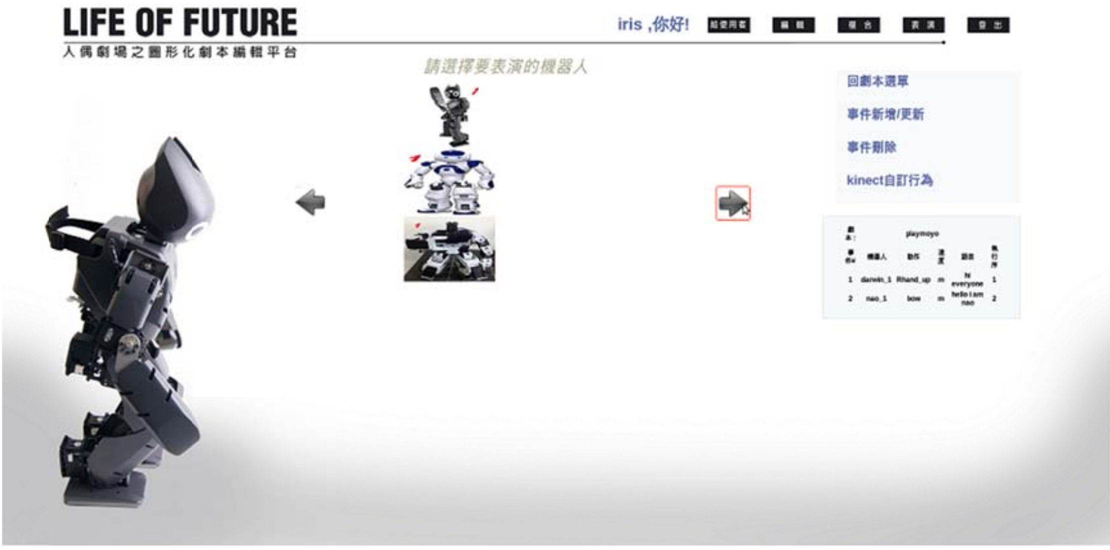


Figure 4-9 Perform robot selection

In this step, it will display the robot in this script for users to choose; Generally speaking, there will be three different robots are DARwIn-OP, NAO, and Bioloid in the script, as shown in Figure 4-9; user sets the number of robots in the establishment of script. When user clicks on desired robot, the robot picture will be zoom display to highlight the user's choice.

Figure 4-10 Robot lines field

After choosing robot, user can enter lines let robot said, as shown in Figure 4-10. If user needs to modify lines, he/she can also click a broom icon; you can easily clear the lines. Finally, the last step is choosing robotic activity.

Figure 4-11 Robot activity selection

Figure 4-11 is the selection page for robot motion or behavior. In this page, we define motion and behavior to clarify the concept of these two key words. In the event, action and behavior can choose one.

Figure 4-12 Robot motion selection page

Figure 4-12 is robot motion selection page. This page provides the robot motions for the user to choose. The motion name display at the top of the page and the picture frame will be red when mouse over the motion picture. Each motion is available in three speeds in order to let the robots express emotions. Figure 4-13 is robotic behavior selection page, and its operation is the same the robotic motion selection page.

Figure 4-13 Robot behavior selection page

The entire setting of an event has been completely wound, the last step user needs to click "add event" button to add a new event, as shown in Figure 4-14, you can see the details of the script.

劇本: playmoyo

事件#	機器人	動作	速度	語音	執行序
1	darwin_1	Rhand_up	m	hi everyone	1
2	nao_1	bow	m	hello i am nao	2
3	nao_1	wave2hand	m	none	3
4	darwin_1	head_yes	m	question	4
5	darwin_1	head_right	m	none	5
6	nao_1	walk_04	m	none	6

Figure 4-14 Details of a script

- **Scripts Mergence**

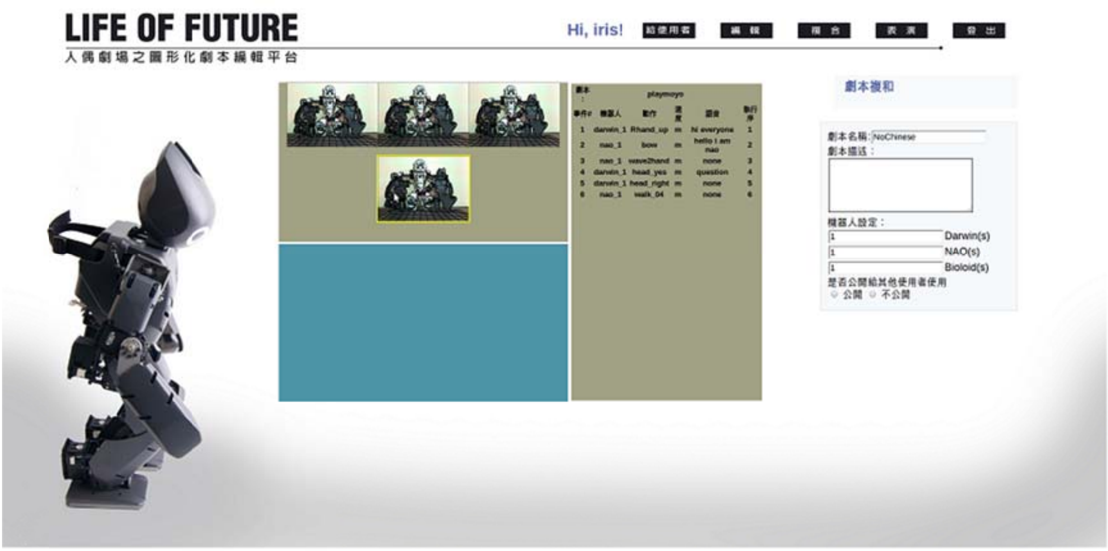


Figure 4-15 Scripts Mergence page

In Scripts Mergence, as shown in figure 4-15, there are three major blocks. The currently available scripts for uses are put at the left upper corner; when mouse over the icon of scripts, the details of the script will show at the right block. The bottom left block is mergence area, and users can drag the scripts which they want to merge to mergence area; then users set up the setting of the script as create a new script, and click “script mergence” to new a script.

- **Scripts Performance**

As the last feature of GAP, scripts performance, the operations of this function as the scripts mergence. The difference between scripts mergence and scripts performance is that script performance provides three scripts performance area, and therefore users can send three scripts to perform at most in the same time. Even if the scripts have been placed, the users can modify the performance order. Although there are only three script performance areas, actually the number of scripts users send is unlimited through scripts mergence. The scripts performance interface is shown in figure 4-16.

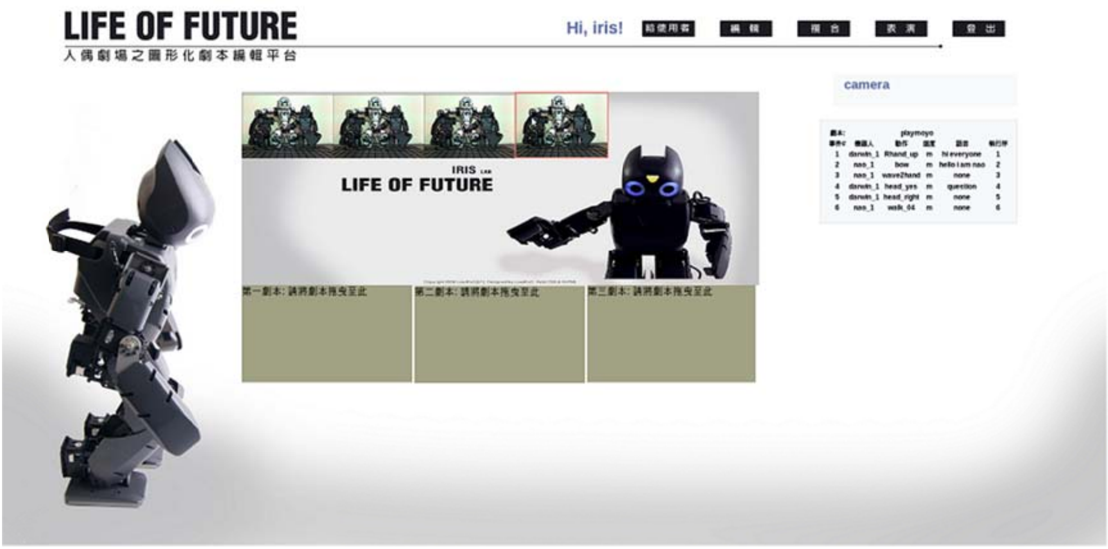


Figure 4-16 Scripts performance interface

4.2.3 XML Script Implementation

XML is the file format of the robotic script in this thesis, and the reason we choose XML is because XML suits for information exchange and the advantage of custom labels. In the script content, we also use our custom labels to describe the robotic script, as shown in Figure 4-17.

```
−<drama name="darwin">
  −<events>
    −<event id="1">
        <robot_type>darwin_1</robot_type>
        <speed>s</speed>
        <activity>Lelbow_90_go</activity>
        <speak>none</speak>
      </event>
    −<event id="2">
        <robot_type>nao_1</robot_type>
        <speed>m</speed>
        <activity>head_down</activity>
        <speak>none</speak>
      </event>
    −<event id="3">
        <robot_type>darwin_1</robot_type>
        <speed>s</speed>
        <activity>wavehand</activity>
        <speak>none</speak>
      </event>
    </events>
  </drama>
```

Figure 4-17 Content of XML robotic script

Figure 4-17 shows the format of XML robotic script. First, “drama” is root node, and “name” represents the name of the script. “drama” is composed of several “event” labels, and every “event” label consists of a “robot_type” label, a “speed” label, a ”activity” label, and a “speak” label; “id” stands for the performance order of the script. Whenever a script is created or modified, GAP will rewrite the XML script; at the same time, GAP will obtain the data from the database.

4.3 Implementation of DARwIn-OP

The implementation of DARwIn-OP can be divided into two sections: client and server. The Client is erected on the computer which GAP is located, and the server is

DARwIn-OP himself. TCP socket established for the communication between the client and server.

4.3.1 Client of DARwIn-OP

The client is erected where GAP located, and the purpose of the client is sent the content of a script to DARwIn-OP. The socket is a two-way data transmission channel, the program through this channel with the local or remote programs to communicate with. General client socket establishment has the following steps:

1. Create a socket: use socket() function to establish. Successfully created, return 0, otherwise it returns -1.
2. Requirements to connect to server: connect() function. If successfully connected, return 0, and the client and server can start interoperability information; otherwise returns -1.
3. Socket successfully created: In this thesis, using the send() function to send data, if successful then the server to receive the sum of the information, if it fails then return -1; recv() function to receive information from the server, if success that the client receives the server information coming from the server, if failure returns -1.
4. Close the socket: Apply close() function to terminate the connection of the client and server. Success returns 0 on failure returns -1.

Above we mentioned is a simple procedure to create a client socket. In the socket of DARwIn-OP, each packet length is 512 bytes which includes a 3 bytes motion page number and 509 bytes robotic lines, as shown in figure 4-18.

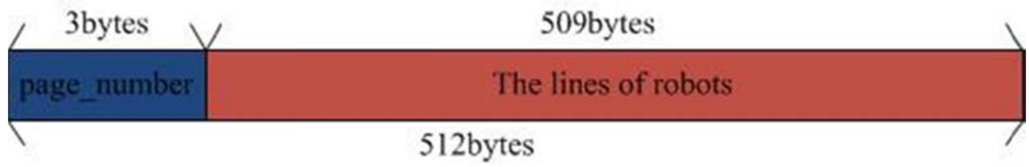


Figure 4-18 Packet format

The contents of each packet are translated by SI. SI translates every element of a script, and capture necessary elements to compose a packet. The translation process is as follows:

1. SI reads the XML script, and translates the script in accordance with events order.
2. First determine the robotic model: if the robotic model is DARwIn-OP, and then query DARwIn-OP table.
3. To judge the event: if the content of “activity” label is "none", SI returns the motion page number: 000. Otherwise, look-up table according to the “activity” and “speed” in the event.
4. Determine the robotic lines: if the lines are "none", it means this event robot with no lines. Otherwise, SI parses the lines.

The implementation of server of DARwIn-OP can be divided into three sections, motion design, voice-enabled, robot walking, and server socket establishment.

4.3.2 Server of DARwIn-OP

- **Motion Design**

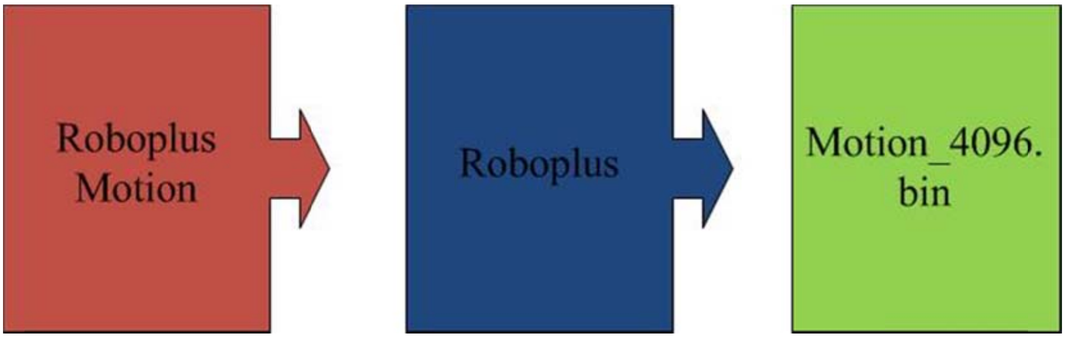


Figure 4-19 DARwIn-OP motion design procedure

Figure 4-19 is overall DARwIn-OP motion design procedure. First, we use Roboplus Motion to design postures, as shown in Figure 4-20.

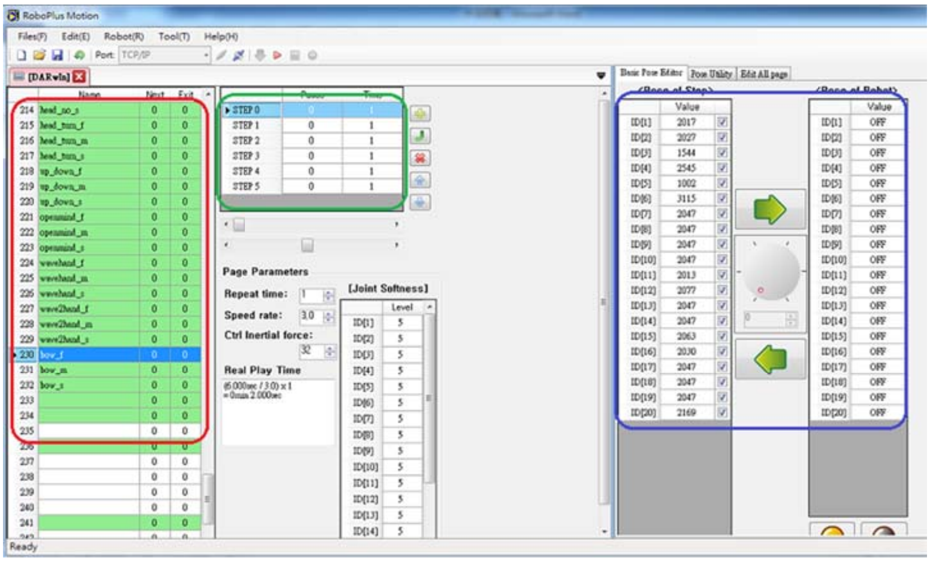


Figure 4-20 Introduction of Roboplus Motion

We can see the Roboplus Motion actually usage in figure 4-20. First in the red box on the left, we can see all motion pages in the Motion_4096.bin, and green background means the motion page has been used. Each motion page provides seven posture fields to design at most, and we can decide motion speed and posture maintenance time. The current motor parameters of DARwIn-OP are shown in the left side of the blue box; the designer can enter a motor parameter to modify the DARwIn-OP posture, or modify DARwIn-OP posture then Roboplus Motion reads the motor parameters. After setting the motion page we save the file, and Roboplus Motion will send this motion page to DARwIn-OP through roboplus.

Roboplus is a program built in DARwIn-OP, and its capability is similar to socket; the roboplus writes the data of motion page which designed by users into motion_4096.bin. The motion_4096.bin is a binary file, and it contains all the relevant parameters of the DARwIn-OP motion, such as the motion page name, the movement speed of the page.

- **Voice-enabled**

The operating system of DARwIn-OP is Ubuntu; we use the function Ubuntu provided to implement voice-enabled. Ubuntu provides a command called “espeak”, and we could use this command to implement text-to-speech easily. The voice-enabled implemented in this thesis is that we write the robotic lines into a text file, and voice capability could be done through the following command:

espeak –f “speech.txt”

We write the robotic lines into speech.txt, and then DARwIn-OP reads the file and speaks it.

- **Robot walking**

Because DARwIn-OP provides a walk function, and therefore we can call walk function to make DARwIn-OP walking. In the planning of GAP, GAP offers users five options about walk, 10cm, 20cm, 30cm, 40cm, and 50cm respectively. Each option corresponds to a motion page number. When DARwIn-OP receives the motion page number, he calls walking function:

Walking::GetInstance()->Start()

To meet the requirement of different walking distance, DARwIn-OP decides walking time in accordance with motion page number. In the thesis, we apply I/O output to maintain DARwIn-OP walking time. Once DARwIn-OP meets the walking distance, DARwIn-OP stops walking by call stop function:

Walking::GetInstance()->Stop()

- **Server socket establishment**

As a server, DARwIn-OP needs to create a server socket. Server socket establishment of the following steps:

1. Create a socket: socket().
2. Connect to the socket: bind(). If success returns 0, or returns-1.
3. Listen the socket: use listen() function to monitor the established socket, and wait for the connection requirement from the client. Successfully returns 0, or returns -1.
4. Wait for client request: accept(). When server receives the request from client, it will store the request in the queue, and then server calls accept() to handle and accept the request. If success returns the socket ID, or returns -1.
5. Receive/transmit packets: when the connection is successfully established, we use the send() function to transfer data and recv() function to receive data.
6. Close the socket: close().

The above is the detail procedure of server socket establishment. In section 4.3.1, we mentioned that the composition of the packet format. On the server side, we need to dismantle packets. In the thesis, the length of a packet is 512 bytes and first three bytes is motion page number, and therefore we need to obtain first three bytes, and then assign it to a string variable; we also assign last 509 bytes to a string variable, as shown in figure 4-21.

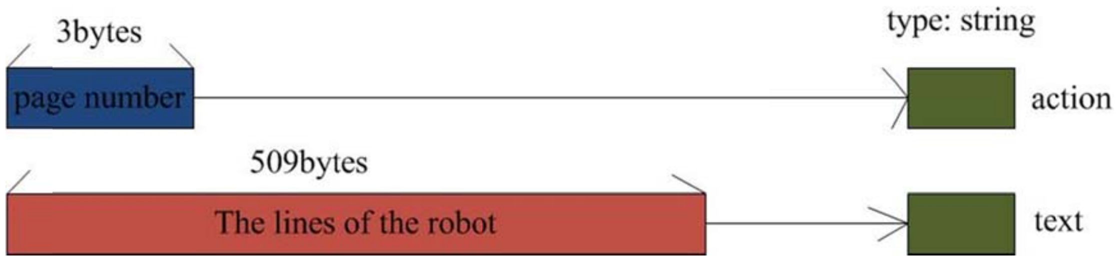


Figure 4-21 Packet dismantling

Now, GAP and the client and server of DARwIn-OP have been implemented, and users connect to GAP they can easily control the robots.

4-4 GAP Experiment

In this section, we designed a questionnaire to verify the practicality of GAP. The complete questionnaire is present in appendix. Users are required to complete two tasks:

1. Create a new screenplay, and it contains five events
2. Merge two screenplays to a new script

There are fourteen samples, and the results of questionnaires are as follows:

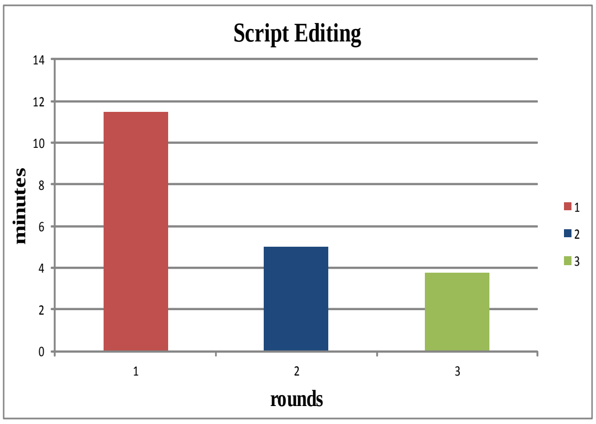


Figure 4-22 (a)

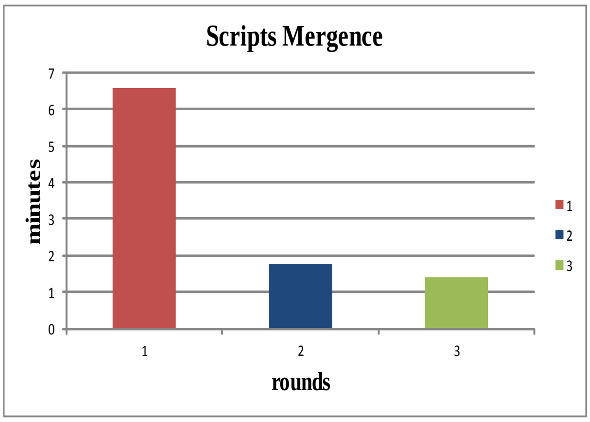
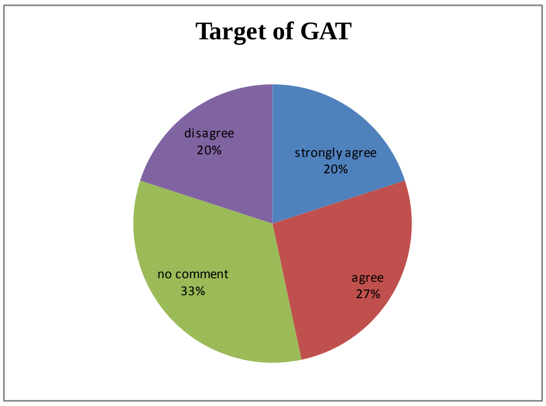


figure 4-22 (b)

Figure 4-22 Task complete time

Figure 4-22 (a) stands for the execution time for Script Editing. This task should be performed three times, and take the average of all samples. From figure 4-22 (a) we could know each execution time reduced. When users enter GAP for first time, they may need time to understand the operation of various functions, so it would be a long time for the first time to perform tasks. When the users learn how to operate the functions, the execution time will be substantially reduced. Figure 4-22 (b) shows the execution time for Scripts Mergence. The users need time to familiar the function at first using as the same as Script Editing, and then each combination of the script can be done quickly.



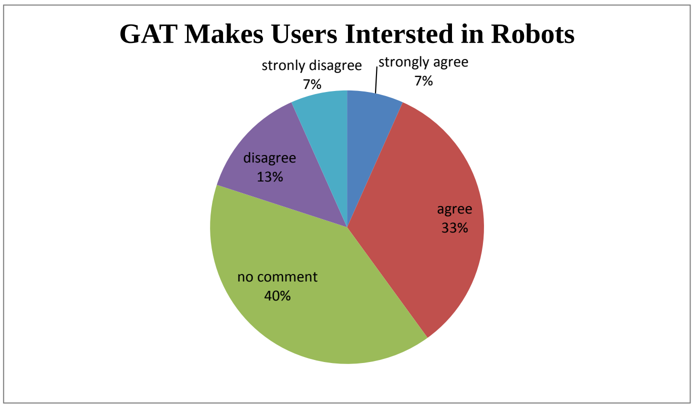


Figure 4-23 (a) Figure 4-23(b)

Figure 4-23 Positive questionnaire result

Figure 4-23 shows some results of the questionnaire. From figure 4-23 (a) we could know that most users can understand what the purpose of GAP is, and the users are more interested in robots through GAP.

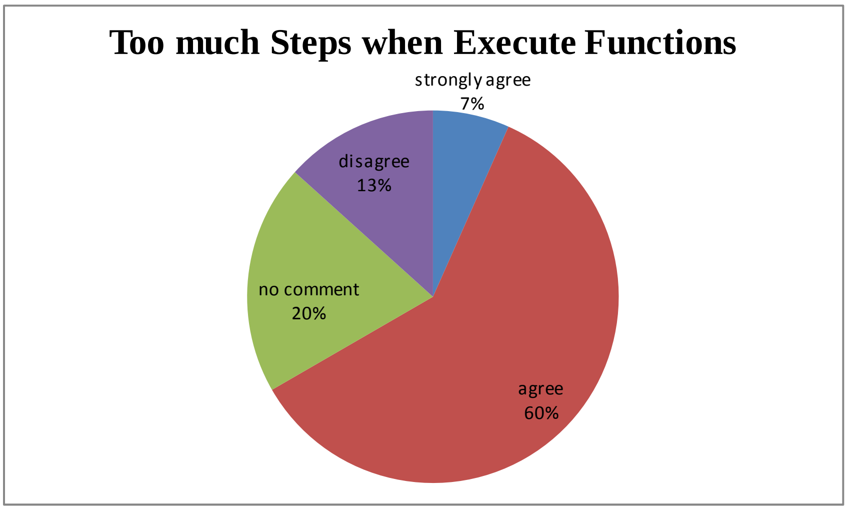


Figure 4-24 Negative questionnaire result

From figure 4-24 we could know that most users think there are too much steps when they execute functions. This is a direction of improvement; we should reduce the steps of using GAP, and the give user a good experiment about GAP.

V. Conclusion and Future Work

5-1 Conclusion

In this thesis, we implement a graphic authoring platform (GAP) of screenplays for robotic puppet shows and a performance robot. GAP is an interesting and friendly web platform. In the viewpoint of operation, people could operate GAP by drag-and-drop, this is an institute operation that human being can easily familiar with.

Base on modularization concept, all motions and behaviors in GAP could be re-used; each script also could be merged as a new script, so all scripts can be re-used as the same as motions and behaviors. People save their time by this feature. All scripts would be saved as XML file, so people could exchange their scripts easily.

Finally, we create a script which called “do-as-I-do” to demonstrate our system. In this script, the robots will say something in English, and they will challenge each other.

5.2 Future Work

GAP provides two different robots, and the future will be incorporated into more different types of robots in order to provide more options to users. In motion editing, we will integrate GAP with behavior emulation and representation of humanoid robot, and then users can design their own behaviors through a Kinect; it will make users more sense of participation.

We also could let robots have an ability to interact with audiences; when users meet specific conditions, the robot will do a specific action to interact with the audience; when users from the audiences into participants, they will be more interested in GAP.

REFERENCE

[1] J. C. Huang, "The Design and Realization of Interactive Biped Robots (I) Gesture Recognition," National Central University, 2008.

[2] J. Y. WU, "The Design and Realization of Interactive Biped Robots (II) Basic Motions Control of Biped Robots," National Central University, 2008.

[3] C. L. Yu, "The Design and Realization of Interactive Biped Robots (III) Execution of Interactive Algorithm for Two Robots," National Central University, 2008.

[4] S. Pimen and A. Z. Talib, "Puppet Modeling for Real-time and Interactive Virtual Shadow Puppet Play," *Second International Conference on Digital Information and Communication Technology and it's Applications (DICTAP)*, pp. 110-114, May 2012.

[5] J. S. Li, "An Authoring Platform for Screenplay of Robotic Puppet Shows," National Chung Cheng University, 2012.

[6] C. A. Chen, "The Authoring Tool and Performance Platform for Robotic Puppet Show," National Chung Cheng University, 2011.

[7] T. L. Kuo, "Intelligent Robotic Puppet Developments," National Chung Cheng University, 2010.

[8] J. Kun and C. Y. Zhao, “Virtual Language Techniques for Software Development,” *Journal of Software*, Vol.19, No.8, pp. 1902-1909, August 2008.

[9] Y. W. Chen, “Design and Implementation of Visual Programming Interface for Low-Cost Embedded Systems,” National Chung Cheng University, 2009.

[10] S. H. Kim and J. W. Jeon, “Programming LEGO Mindstorms NXT with Visual Programming,” *International Conference on Control, Automation and Systems, ICCA.* pp. 2468-2472, Oct. 2007.

[11] M. J. Nelson, M. Mateas, D.L. Roberts and C. L. Isbell, “Declarative Optimization-based Drama Management in Interactive Fiction,” *IEEE Computer Graphics and Applications*, Vol. 26 , Issue 3, pp. 32-41, 2006.

[12] A. R. Chatley, K. Dautenhahn, M. L. Walters, D. S. Syrdal, and B. Christianson, “Theatre as a Discussion Tool in Human-Robot Interaction Experiments - A Pilot Study,” *Third International Conference on Advances in Computer-Human Interactions, ACHI,* pp. 73-78, Feb. 2010.

[13] W. Kumiko, E. Ryohei, N. Miki, K. Fusako, M. Hiroshi and I. Shigenori, “Evaluation of the Universal Puppet Theater based on Inclusive Design Method,” *IEEE Fourth International Conference on Digital Game and Intelligent Toy Enhanced Learning (DIGITEL),* pp. 135 - 137 , March 2012.

[14] J. S. Hu, J. J. Wang and G. Q. Sun, “The Glove Puppet Robot: X-puppet,” *International Conference on Intelligent Robots and Systems, IROS. IEEE/RSJ*, pp. 4145 - 4146, Sept. 2008.

[15] A. Bertolino, G. De Angelis, F. Lonetti and A. Sabetta, “Let the Puppets Move! Automated Testbed Generation for Service-oriented Mobile Applications,” *34th Euromicro Conference Software Engineering and Advanced Applications, SEAA.* , pp. 321-328, Sept. 2008.

[16] I. Ha, Y. Tamura, H. Asama, J. Han and D.W. Hong, “Development of Open Humanoid Platform DARwIn-OP,” *Proceedings of SICE Annual Conference (SICE)*, pp. 2178 – 2181, Sept. 2011.

[17] R. Volpe, I.A.D. Nesnas, T. Estlin, D. Mutz, R. Petras, H. Das, "The CLARAty Architecture for Robotic Autonomy." *Proceedings of the 2001 IEEE Aerospace Conference*, March 2001.

[18] I.A.D. Nesnas, R. Volpe, T. Estlin, H. Das, R. Petras D. Mutz, "Toward Developing Reusable Software Components for Robotic Applications" *Proceedings of the International Conference on Intelligent Robots and Systems (IROS)*, Oct - Nov 2001.

[19] R. Volpe, I.A.D. Nesnas, T. Estlin, D. Mutz, R. Petras, H. Das, "CLARAty: Coupled Layer Architecture for Robotic Autonomy." *JPL Technical Report D-19975*, Dec 2000.

[20] T. Estlin, R. Volpe, I.A.D. Nesnas, D. Mutz, F. Fisher, B. Engelhardt, S. Chien, "Decision-Making in a Robotic Architecture for Autonomy." *Proceedings of 6th International Symposium on Artificial Intelligence, Robotics, and Automation in Space (i-SAIRAS)*, June 2001.

Appendix

A questionnaire of the graphic authoring platform of screenplays for robotic puppet shows

- 機器人台詞請勿使用 ‘ 此標點符號
- 台詞僅支援英文
- 劇本名稱請勿輸入中文

功能親切度

	第一次（單位：minute）	第二次（單位：minute）	第三次（單位：minute）
建立一個劇本，包含五個事件			
將兩個劇本組合成一個新劇本			

網頁設計

	非常同意	同意	沒意見	不同意	非常不同意
網站的排版清楚易懂					
我能理解網站的輔助說明					
我願意花時間使用此網站					
網站的定位清楚明確					
執行某些功能時，須點選許多畫面					
我常有不知身處何方的感覺					
執行網站功能的過程相當順利					
網頁的美觀令我滿意					
網頁的操作方式很直覺					
此網頁令我對機器人產生興趣					