

# 國立交通大學

資訊學院 資訊學程

碩士論文

視覺化劇情呈現需求編輯器

Authoring tool for visual scenario representation for software  
requirement

研究生：黃維明

指導教授：陳登吉 教授

中華民國一百年八月

視覺化劇情呈現需求編輯器  
Authoring tool for visual scenario representation for software  
requirement

研究生：黃維明

Student : Wei-Ming Huang

指導教授：陳登吉

Advisor : Deng-Jyi Chen



A Thesis  
Submitted to College of Computer Science  
National Chiao Tung University  
in Partial Fulfillment of the Requirements  
for the Degree of  
Master of Science  
in  
Computer Science  
August 2011

Hsinchu, Taiwan, Republic of China

中華民國一百年八月

# 視覺化劇情呈現需求編輯器

學生：黃維明

指導教授：陳登吉 博士

國立交通大學 資訊學院 資訊學程碩士班

## 摘要

隨著社會經濟和現代化生活對軟體的依賴程度劇增，軟體需求日益複雜，一個軟體系統的成功與否在於它是否能達到所預期的功能要求，是需求工程一直在討論的問題。多數專案失敗的原因在於需求無法掌握，因此在軟體工程知識領域中相關的需求工程科學顯得更為重要。因為若在初步階段沒有掌握需求內容，後續程式開發會遇到的困境將接踵而來，致使延宕開發時程；如何掌握需求便成為軟體開發的首要任務。

一般常見的需求規格呈現方式大都以文字模式，例如會議紀錄、組織現有的表單、問卷調查結果等大量的文字來呈現。需求除了以文字模式呈現，也可以以視覺化模式呈現。例如，利用影片、UML Modeling 或 authoring tools 編輯出的互動式多媒體程式。兩者的呈現方式雖然不同，但是在觀念是卻是相同的，即兩者所表示的都是”軟體需求”的內容。

根據研究，以視覺化劇情呈現需求有下列優點(1) 更接近自然的需求表達方式、(2)比較容易了解需求、(3)對於使用者與系統開發人員之間的溝通更加完善。因此本研究探討以”視覺化劇情呈現需求”，並且設計實作一套編輯工具，利用此工具根據文字需求編輯出以”視覺化劇情呈現需求”的範例。

關鍵字：需求工程、多媒體編輯器、視覺化程式語言

# Authoring tool for visual scenario representation for software requirement

Student : Wei-Ming Huang

Advisor : Dr. Deng-Jyi Chen

Degree Program of Computer Science  
National Chiao Tung University

## ABSTRACT

With the socio-economic and modern life of the dramatic increase in dependence on software, software requirements become more complex. A software system's success is its ability to achieve the desired functional requirements which has been discussed in requirement engineering. Most projects fail because the demand can not be mastered, so the requirement engineering in software engineering knowledge is more important. Because if not master the requirements in the initial stages of the content, follow-up program development will come one after another to encounter difficulties. How to master the demand has become the primary task of software development.

General presentation of the common requirements specification mostly in text mode, such as minutes of meetings, the existing forms, the survey results. In addition to text mode rendering, visual model can also be present for requirement representation. For example, video, UML Modeling or multimedia rendered by authoring tools. Although the presentation of two different, but it is in the concept is the same in that they are expressed "Software Requirements" of the content.

According to the study, visual scenario representation for software requirement has following advantages:(1)The representation of requirement is more naturally,(2)It's easy to understand requirement,(3)The communication between user and system developer is more complete. In this paper, we designed and developed a authoring tool for visual requirement representation and demonstrate some examples created by this tool.

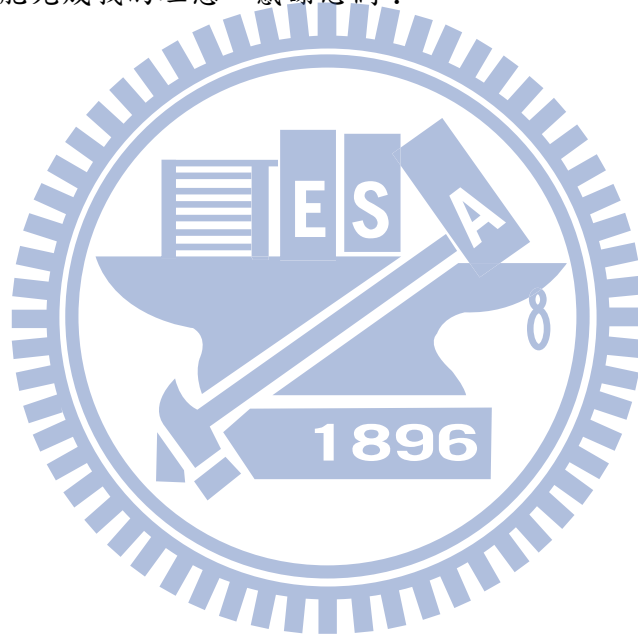
Keywords: Requirement Engineering, Multimedia Tools, Visual Programming Language

## 誌謝

衷心感謝陳登吉教授耐心的指導，以及師母的鼓勵，除了在研究的過程中，時時給予建議與方向，對於一般日常生活的問題也時時給予關心。因此才能順利完成本論文。

另外在研究的過程中，則要感謝學長毓維的指導幫助，以及同窗興郎、志帆、豪岱的互相鼓勵與一同進行研究，透過彼此的分享與討論，使我可以更清楚相關的研究方向。

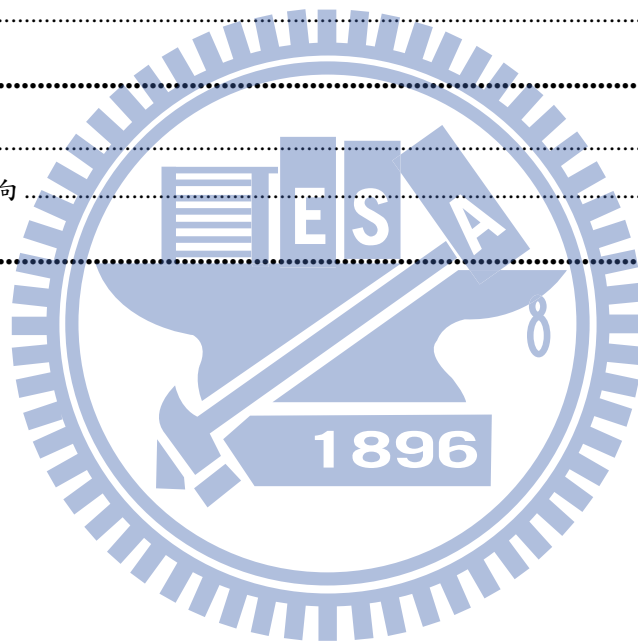
最後，要感謝我的家人與同事在求學的這一段期間內給予的支持與鼓勵，因為您們的支持才能完成我的理想，感謝您們！



# 目錄

摘要.....	I
ABSTRACT.....	II
誌謝.....	III
表目錄.....	VI
圖目錄.....	VII
一、 緒論.....	1
1.1 軟體開發與需求工程.....	1
1.2 動機與目的.....	3
1.3 研究方法與步驟.....	5
1.4 章節概要.....	5
二、 相關研究.....	6
2.1 視覺化呈現需求相關研究.....	6
2.1.1 Video 方式呈現需求.....	6
2.1.2 UML Modeling 方式呈現需求.....	7
2.1.3 Multimedia 方式呈現需求.....	8
2.2 多媒體編輯工具的比較.....	8
2.2.1 使用 Microsoft PowerPoint 編輯.....	9
2.2.2 使用 Adobe Flash 編輯.....	11
2.2.3 使用 Authoring Tools 編輯.....	12
2.1.3 綜合比較.....	13
2.3 理想的多媒體編輯工具應用於視覺化劇情呈現需求.....	14
三、 多媒體編輯工具系統需求分析.....	15
3.1 視覺化的編輯方法.....	15
3.1.1 視覺化語言.....	15
3.1.2 視覺化語言的實現.....	16
3.1.3 視覺化的編輯模式.....	16
3.2 多媒體的組成與軟體需求.....	17
3.2.1 互動式多媒體的組成.....	17
3.2.2 Authoring Tool For Visual Software Requirement Representation.....	18
3.2.2 文字需求的轉換過程.....	19

3.3 多媒體編輯工具系統需求 .....	22
3.3.1 編輯機制.....	23
3.3.2 系統功能.....	23
<b>四、 多媒體編輯工具系統設計與實作 .....</b>	<b>24</b>
4.1 編輯畫面設計 .....	24
4.2 系統架構設計 .....	29
4.2.1 系統語法定義.....	30
4.2.2 系統設計樣式.....	31
4.3 系統 CLASS DIAGRAM .....	33
4.4 系統 SEQUENCE DIAGRAM .....	35
<b>五、 實作範例.....</b>	<b>36</b>
5.1 範例.....	36
<b>六、 結論.....</b>	<b>45</b>
6.1 總結 .....	45
6.2 未來發展方向 .....	46
<b>參考文獻或資料.....</b>	<b>47</b>



# 表目錄

表 1- SOFTWARE PROJECT COSTS BY EACH PHASE.....	2
表 2- COST OF CORRECTING SOFTWARE ERRORS.....	3
表 3- 文字模式需求範例.....	4
表 4- 編輯工具比較.....	13





# 圖目錄

圖 1- SYSTEMS DEVELOPMENT LIFE CYCLE (SDLC).....	2
圖 2-TEXTUAL FORM AND VISUAL FORM OF USER REQUIREMENT.....	4
圖 3- VIDEO WALL ACTIVITY: VIDEO CLIPS ARE LABELED AND ARRANGED AS REQUIREMENTS ACCORDING TO THEIR IMPORTANCE.....	6
圖 4- UML MODELING.....	7
圖 5- ICONS DEFINITION.....	8
圖 6- VISUAL SRS.....	8
圖 7- MICROSOFT POWERPOINT 圖示.....	9
圖 8- MICROSOFT POWERPOINT 自訂動畫.....	10
圖 9- MICROSOFT POWERPOINT VBA MACRO CODE EDITOR.....	10
圖 10- ADOBE FLASH 編輯畫面[13].....	11
圖 11- ADOBE FLASH ACTION SCRIPT 編輯畫面.....	11
圖 12- VRDL 編輯畫面.....	12
圖 13- ICONS DEFINITION.....	12
圖 14- 互動式多媒體的組成.....	17
圖 15- VISUAL REQUIREMENT REPRESENTATION.....	19
圖 16- 文字需求分析流程.....	20
圖 17- 視覺化呈現的需求的四個基本元素對應.....	21
圖 18- 視覺化劇情呈現需求編輯流程.....	22
圖 19- 多媒體編輯工具系統主畫面.....	24
圖 20- 多媒體編輯工具系統選取 ACTOR.....	25
圖 21- 多媒體編輯工具系統定義流程(ACTION=RIGHTCLICK).....	26
圖 22- 多媒體編輯工具系統定義流程(ACTION=DOUBLECLICK).....	26
圖 23- 多媒體編輯工具系統定義流程(ACTION=DRAGDROP).....	27
圖 24- 多媒體編輯工具系統播放互動流程.....	27
圖 25- SCRIPT 預覽.....	28
圖 26- 多媒體編輯工具系統架構.....	29
圖 27- INTERPRETER CLASS DIAGRAM.....	32
圖 28- MEDIATOR CLASS DIAGRAM.....	33
圖 29- SYSTEM CLASS DIAGRAM.....	34
圖 30- SYSTEM SEQUENCE DIAGRAM.....	35
圖 31- 新增 ICON.....	36
圖 32- 選取並拖曳 ICON.....	37
圖 33- 定義流程:提款卡插入提款機, SHOW 出輸入密碼畫面.....	38
圖 34- 定義流程:提款卡插入提款機, SHOW 出輸入密碼畫面.....	38

圖 35-定義流程:輸入密碼錯誤時，SHOW 出錯誤訊息 .....	39
圖 36-定義流程:輸入密碼正確時，且提款成功時，SHOW 出鈔票圖示.....	39
圖 37-檢視 SCRIPT.....	40
圖 38-播放流程:DOUBLE CLICK 使用者 ICON，系統 SHOW 出提款卡 .....	41
圖 39-播放流程:拖曳提款卡插入提款機，SHOW 出輸入密碼畫面 .....	41
圖 40-播放流程:拖曳提款卡插入提款機，SHOW 出輸入密碼畫面 .....	42
圖 41-播放流程:右鍵輸入密碼錯誤時，SHOW 出錯誤訊息 .....	42
圖 42-播放流程:右鍵輸入密碼錯誤時，SHOW 出錯誤訊息 .....	43
圖 43-播放流程:右鍵輸入餘額不足時，SHOW 出錯誤訊息 .....	43
圖 44-播放流程:右鍵輸入時提款成功時，SHOW 出金錢圖示 .....	44



# 一、緒論

## 1.1 軟體開發與需求工程

目前全球所有國家幾乎都必須依賴複雜的電腦化系統，而且越來越多產品也都以某種形式整合了電腦與控制的軟體。在這些系統中，軟體占系統總成本的比例非常高[1]，而且正逐漸增加中。因此，以合乎成本效益的方式生產軟體是非常重要的。

簡單的說，軟體系統開發就是開發人員根據使用者的需求，將需求轉換成系統的一個過程。以軟體開發的程序模型(Software Process Models)中的”瀑布式模型”(Waterfall)為例，分成了五個階段[圖 1]:

1. 需求定義(Requirement)
  - 在這個階段主要是分析出系統所要解決的問題，包含了分析系統的目標、解構出不同的子問題並與使用者共同定義出需求。
2. 系統設計(Design)
  - 詳細描述功能和操作方式，也包含介面的呈現設計，商業邏輯流程，規畫出不同的子系統和模組。
3. 系統實作(Implementation)
  - 根據系統規格進行系統實作。
4. 系統測試(Testing)
  - 包含了單元測試(Unit Test)、系統測試(System Test)、系統整合測試(System Integration Test)...等功能面測試;另外也包含非功能面需求測試，例如效能測試(Performance Test)。
5. 系統維護(Maintenance)
  - 在系統淘汰(sunset)前的任何變更，包含需求變更(Requirement change)或是需求增強(Requirement enhancement)都算是系統維護。

這 5 個步驟的循環，組成了”軟體開發生命週期”(Systems Development Life Cycle)，簡稱為 SDLC。

其中 ”Requirement” 階段，是系統開發人員與使用者進行諮商後，定義出系統提供的服務、限制以及目標，成為詳細的系統規格。

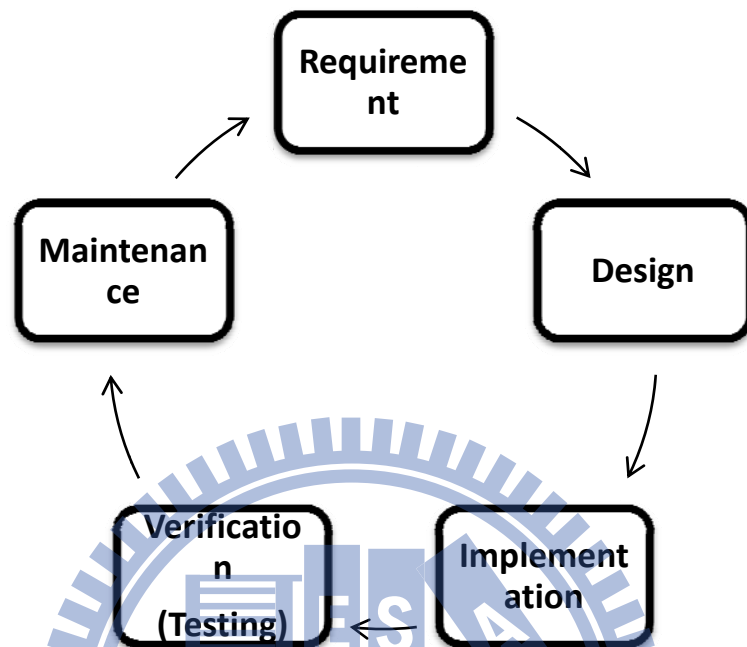


圖 1- Systems Development Life Cycle (SDLC)

根據研究[2]，在”軟體開發生命週期”的 5 個階段中所需要的成本，以”Requirement”階段最低[表 1]。但是因為需求階段主要是蒐集使用者需求為主，若是收集錯誤或是遺漏，將對後續階段的工作造成影響。因此如果錯誤在最”後維護階段”時才被發現，將相對於”需求階段”發現錯誤所需付出的 10-100 倍的代價。[表 2]

表 1- Software Project Costs by Each Phase

Software Project Phase	Percent of Project
Requirements	3
Design	8
Programming	7
Testing	15
Maintenance	67

表 2- Cost of Correcting Software Errors

Software Development Phase	Requirements Analysis	Design	Code and Unit Test	Integration Test	Validation and Documentation	Operational Maintenance
Development Funds	5%	25%	10%	50%	10%	
Errors Introduced	55%	30%		10%		5%
Errors Found	18%	10%		50%		22%
Relative Cost to Correct	1x	1-1.5x		1-5x		10-100x

## 1.2 動機與目的

由上節我們了解到了”需求階段”在整個軟體開發生命週期中的重要性，因此我們以需求的呈現方式來探討對於”需求階段”的幫助。

文字跟其他媒體都是在協助人類表達認知的一種型態，例如，用文字樂譜來表達一段樂曲，和直接用聲音來表達一段樂曲兩者的內容是相同的。而在軟體工程中，一般最常見的需求呈現方式就是透過文字模式(Textual Form)，以大量的文字檔案，如組織現有的文件、會議紀錄、計畫書、問卷調查，都可以成為需求規格的一部分。另一種方式，就是視覺化模式(Visual Form)的表達方式，以影片、動畫、或互動多媒體方式等，來表達需求。[3][圖 2]

根據研究[3]，比較上述的兩種表現需求的模式，以視覺化(Visual Form)的需求呈現方式是一種比文字模式(Textual Form)較直覺的方式，因為文字模式的軟體需求如大量的文字檔案比較難以閱讀，需要多一點想像力，才能比較接近使用者的想法。對於使用者方面，也可以讓使用者對於自己想像的需求更具體化，以確認需求的內容。

因此，以視覺化呈現需求有下列優點：

- (1) 是一種更接近自然的需求表達方式。
- (2) 比較容易了解需求內容。
- (3) 對於使用者或是系統開發人員之間的溝通更加完善。

雖然兩者的呈現方式不同，但是不論是文字模式，或是視覺化模式的需求表達方式，他們在意義上是相同的(Conceptual Equivalent)，都是表達使用者需求。

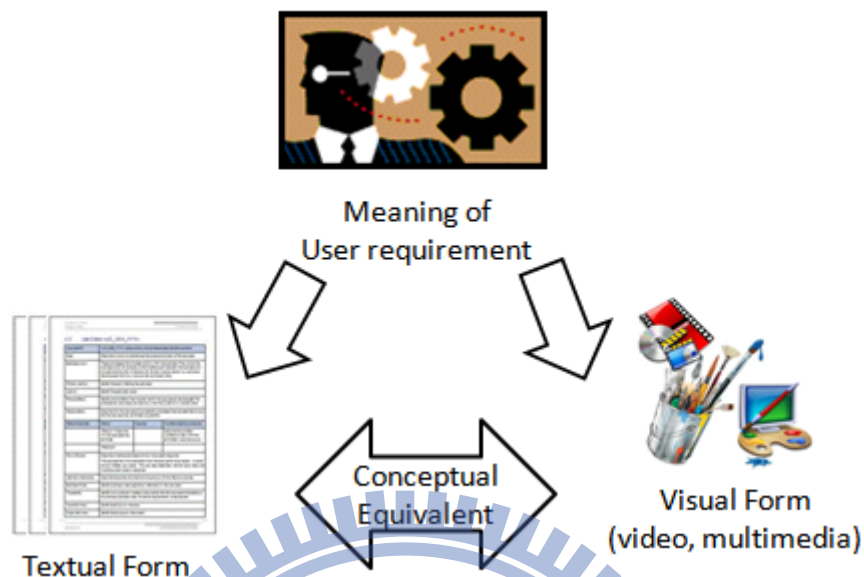


圖 2-Textual Form and Visual Form of User Requirement

我們以一個常見的文字需求-以銀行 ATM 提款機需求為例子[表 3]，通常描述條列出需求，並描述需求細部內容。編輯出文字模式的需求所需的工具，最常使用的就是一般的文字編輯器(Text Editor)，例如:Microsoft Word、PDF...等工具來完成。但是這樣的文字模式需求，要如何以視覺化方式來呈現呢?綜合上述對於文字模式與視覺化模式需求呈現的比較與此動機，本論文的目標就是設計一個工具，以多媒體編輯的方式，並以 ATM 提款系統需求為例，編輯出以”視覺化劇情”呈現需求。

表 3- 文字模式需求範例

<p><b>銀行 ATM 系統需求(範例):</b></p> <ul style="list-style-type: none"> <li>● <b>提款</b>            客戶可以在 ATM 上提款，提款時先插入提款卡，輸入密碼。若輸入密碼錯誤，系統會秀出密碼錯誤的訊息，若密碼正確，系統秀出要使用者輸入提款金額。客戶輸入金額，提款成功。若提款金額大於帳戶金額，則系統秀出金額不足的錯誤訊息。</li> <li>● <b>存款</b>            客戶可以在 ATM 上存款。存款時必須輸入帳號與密碼。            .....</li> </ul>
---

### 1.3 研究方法與步驟

首先我們先進行相關性研究，探討相關以視覺化模式呈現需求的研究，以及使用的工具及特性。接著探討現有的工具特性，並針對其不足與限制提出解決的方法。之後針對我們提出的方法做需求分析以及實作，詳細的步驟條列如下：

- (1) 多媒體編輯工具的探討與相關研究。
- (2) 探討現有多媒體編輯工具應用於”視覺化劇情呈現需求”的不足。
- (3) 分析所需額外提供的功能。
- (4) 設計系統，並加入設計模式，實作出視覺化編輯工具並應用於編輯”視覺化劇情呈現需求”。

### 1.4 章節概要

第一章，我們提出撰寫本篇論文的動機與目的，以及研究方法與步驟。

第二章，分析目前相關視覺化呈現需求的研究，探討其限制及不足。

第三章，針對提出的解決方法進行功能需求分析。

第四章，設計實作出”視覺化劇情呈現需求編輯器”系統。

第五章，以實際範例示範，將文字模式的需求利用”視覺化劇情呈現需求編輯器”編輯出視覺化模式呈現需求。

第六章，本論文的總結，並對未來的發展提出一些建議方向。

## 二、 相關研究

在本章節中主要內容分為兩個部分:(1)蒐集視覺化呈現需求的相關研究，並介紹各篇研究的不同呈現方式、以及其編輯工具。(2)以多媒體編輯工具為主，探討三種不同的多媒體編輯工具應用在編輯以視覺化劇情呈現需求上的方式及特性。

### 2.1 視覺化呈現需求相關研究

視覺化呈現需求的相關研究，我們蒐集了 5 篇相關的 papers，分成三類：

- (1) 以 Video 方式呈現。[4][5]
- (2) 以 UML Models 方式呈現。[6]
- (3) 以 Multimedia 方式呈現。[7][8]

以下分別是三種不同呈現方式的介紹，並探討所需工具：

#### 2.1.1 Video 方式呈現需求

以此篇論文”Ethnographic Video as Design Specs / Jacob Buur”為例子，研究如何以 Video Spec 來幫助開發過程，建立明確的限制與解決方案。將使用者需求拍成影片，並加以編排。影片就像電影一樣有劇情，用劇情來描述需求的內容。[圖 3]是一個 video 呈現需求的範例，將需求以 video 方式呈現並且分類成 3 種不同需求層級，讓開發者可以明確的了解需求的內容與優先程度。



圖 3- Video Wall activity: Video clips are labeled and arranged as requirements according to their importance



將使用者需求以 video 方式呈現，所需的工具在硬體方面需要 V8 或其他可錄影之器材；軟體方面，則需要將拍攝後的影像利用影像編輯軟體後製編排。

## 2.1.2 UML Modeling 方式呈現需求

在業界利用”統一塑模語言”(UML),”模型驅動開發”(MDD)與”模型驅動架構”(MDA)開發方式漸漸興起[6]。UML 中的”循序圖”(Sequence Diagram)或是”活動圖”(Activity Diagram)用來描述流程、”狀態圖”(State Diagram)用來描述流程中特定物件狀態的轉移與變化過程、”類別圖”(Class Diagram)用來描述物件與物件之間的關係。以下是使用 UML Modeling Tool 的一個範例。

在此篇論文”Visualizing Requirements in UML Models / Sascha Konrad”利用 UML 相關圖形的特性，建立一個流程(ReVU)，將”功能需求”(Functional Requirements)用視覺化方式 UML modeling 方式來呈現。

將使用者需求以 UML Modeling 方式呈現，所需工具需要 UML Modeling Tools 來方便編輯[圖 4]。現行已有許多 freeware 的 UML Modeling Tools 如”StarUML”、”JUDE”...等。

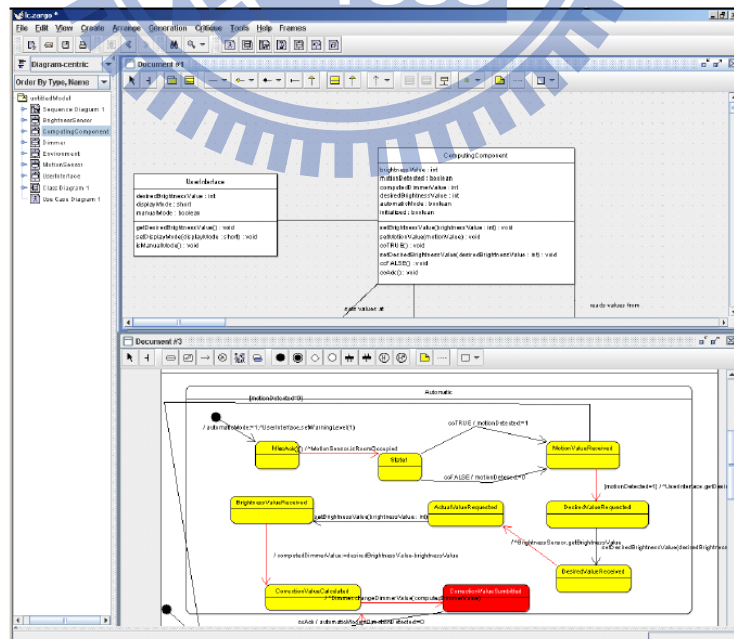





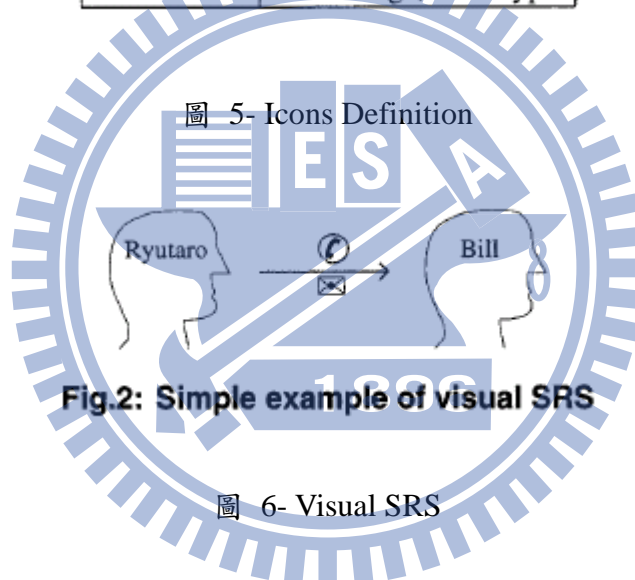
圖 4- UML Modeling

## 2.1.3 Multimedia 方式呈現需求

以多媒體方式呈現需求而言- “Visual Software Requirements Definition Environment / Atsushi Ohnishi” 這篇研究，以定義有意義的 icon 圖示 (semantics of icons)來代表物件[圖 5]，用線條來代表物件之間的關連。使用者利用 icons 與建立關聯來完成需求的流程編輯，最後再以動畫方式呈現編輯結果 [圖 6]。

Table 3: Examples of icon definition

Icon shape	semantics
	human type
	"Fax," device type
	"a message," data type



## 2.2 多媒體編輯工具的比較

目前市面上多媒體編輯工具非常多，例如 Microsoft PowerPoint、Adobe Flash 或其他 authoring tools。我們嘗試以此多媒體編輯工具來編輯出以視覺化劇情呈現需求，以下針對不同的工具做簡單的特性描述，並比較其結果。首先我們的文字需求範例以銀行 ATM 提款需求為例。根據需求的描述我們發現有許多互動行為與流程控制，例如，“插入提款卡”可以視為一個互動式的行為；而“輸入密碼”之後會需要不同的條件控制，“輸入密碼成功”則可以進行提款，“輸入密碼錯誤”則系統秀出錯誤訊息。



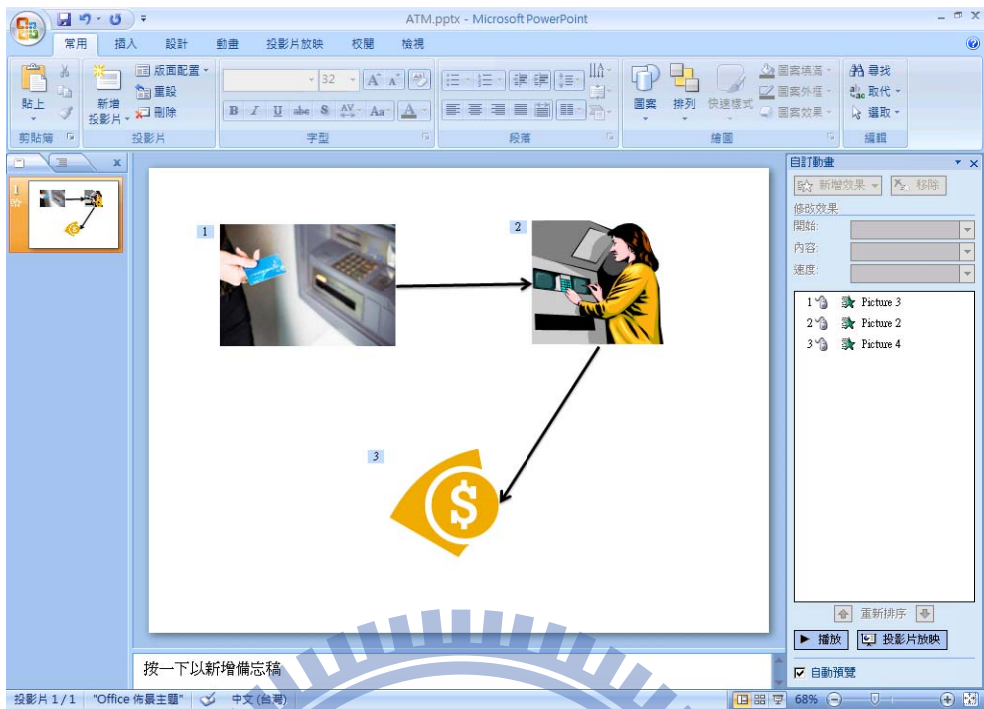


圖 8- Microsoft PowerPoint 自訂動畫

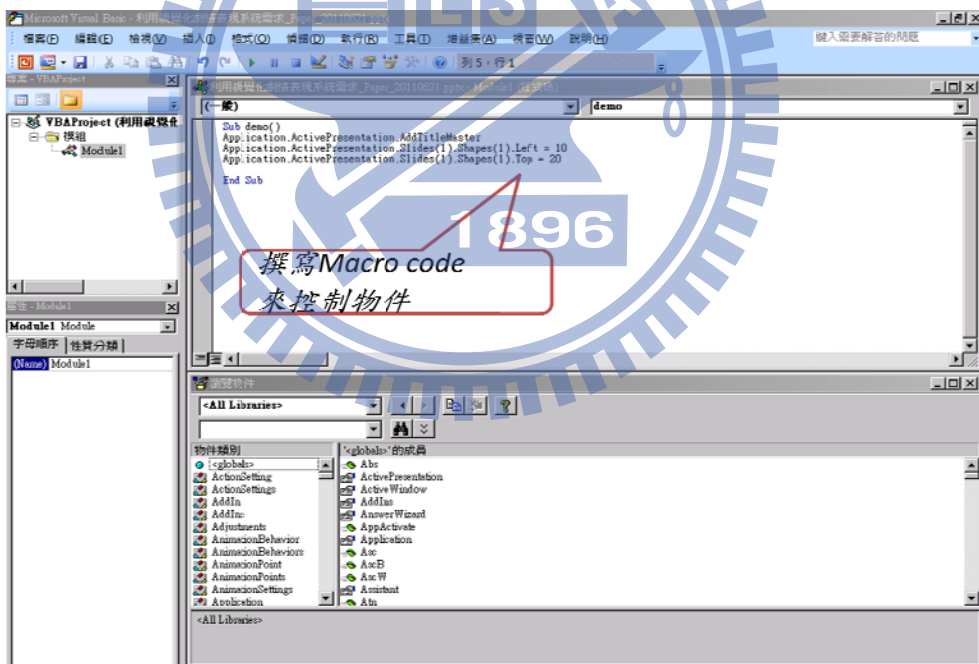


圖 9- Microsoft PowerPoint VBA macro code editor.

## 2.2.2 使用 Adobe Flash 編輯

Adobe Flash 是專門用來設計互動式多媒體動畫的軟體。以時間軸為基礎的多媒體編輯工具(timeline based)，將每一種既定的多媒體元件如圖片、聲音、影像...等，以成員角色(Symbol)的方式展現，配合舞台(Stage)的觀念，讓各種原件呈現。[圖 10]

由於 Adobe Flash 編輯工具功能強大，用來進行”多媒體呈現需求”的編輯是可以達到的。但是由於其專業功能太多，需要有一些電腦與美工的基礎，且對於時間軸的觀念要很清楚才能正確編輯。此外，若要在 Flash 中產生互動功能，使用者必須透過撰寫 Action Script 程式碼來定義物件之間的互動。[圖 11]

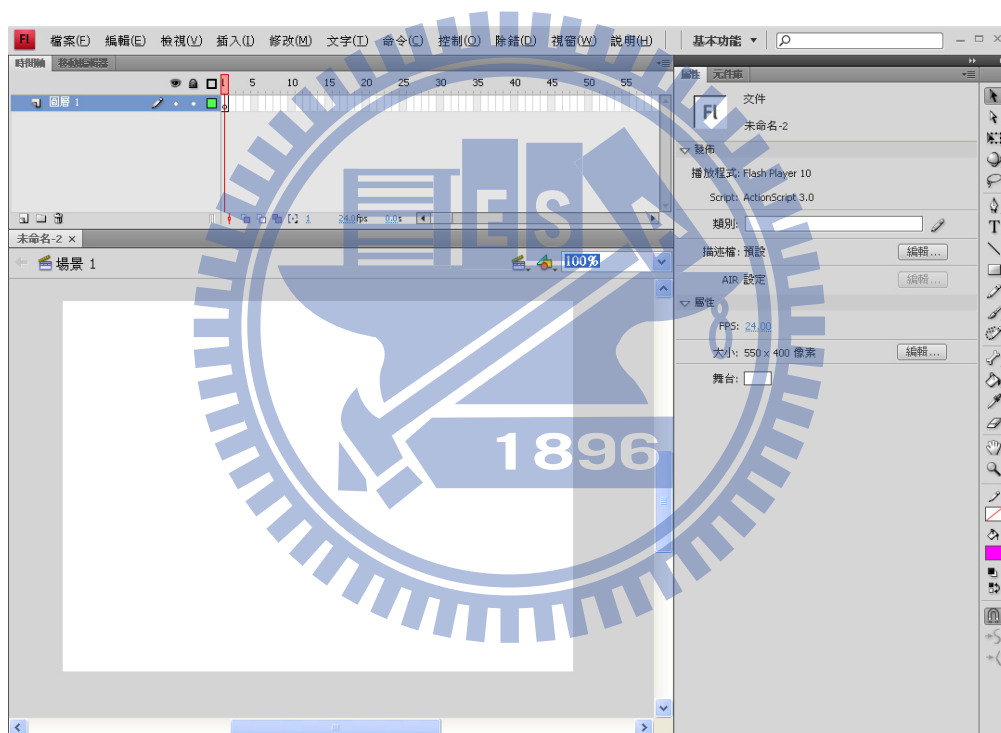


圖 10- Adobe Flash 編輯畫面[13]

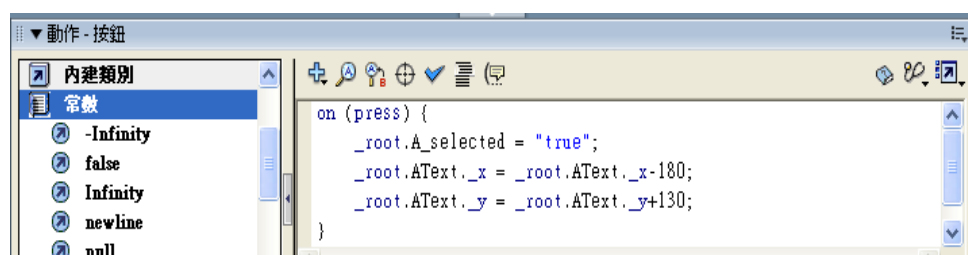


圖 11- Adobe Flash Action Script 編輯畫面

## 2.2.3 使用 Authoring Tools 編輯

於 2.1.3 小節中我們所提到的”Multimedia 方式呈現需求”的論文中-“Visual Software Requirements Definition Environment / Atsushi Ohnishi”，是一個很明確應用於以視覺化方式來呈現需求的一個工具。以下是其編輯工具畫面[圖 12]；使用者透過右邊的 icon 區，拖曳出所需的已定義的 icon 圖式[圖 13]，建立 icon 之間的關連，完成所謂的”Visual SRS”。

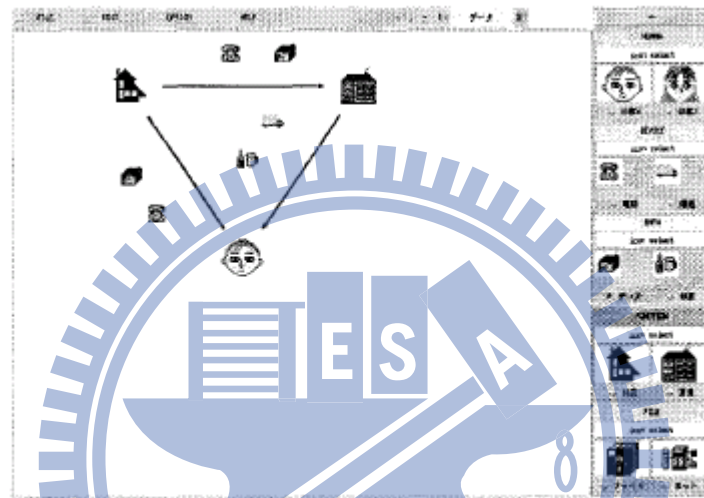


Fig.3: VRDL example  
1896

圖 12- VRDL 編輯畫面

Table 6: Icons definition

Icon	semantics (label, type)	
	customer	human
	clerk	human
	warehouse	function
	liquor shop	function
	order DB	file
	order form	data
	liquor	data
	fax	device
	track	device

圖 13- Icons Definition

## 2.1.3 綜合比較

綜合以上 3 種工具應用於編輯出”視覺化劇情呈現需求”的結果，我們分別就其各種支援度來做一個比較[表 4]，結果如下表所示，專業的編輯工具雖然有提供比較強大的編輯能力，不過需要手動撰寫文字程式碼，且操作上比較複雜，無適合沒有程式撰寫能力的使用者上手。

表 4- 編輯工具比較

使用工具	優勢	劣勢	特性
Microsoft PowerPoint	<ul style="list-style-type: none"> <li>● 容易上手</li> </ul>	<ul style="list-style-type: none"> <li>● 須撰寫文字 VBA macro code 完成互動功能。</li> </ul>	<ul style="list-style-type: none"> <li>● 內建圖庫</li> </ul>
Adobe Flash	<ul style="list-style-type: none"> <li>● 功能完整強大</li> </ul>	<ul style="list-style-type: none"> <li>● 一般使用者無法掌握複雜功能操作。</li> <li>● 須撰寫文字 Action Script 完成互動功能。</li> </ul>	<ul style="list-style-type: none"> <li>● Timeline based</li> </ul>
authoring tools	<ul style="list-style-type: none"> <li>● 容易上手</li> </ul>	<ul style="list-style-type: none"> <li>● 只能撥放、缺少互動機制。</li> <li>● Icon 定義特定，無擴充機制。</li> </ul>	<ul style="list-style-type: none"> <li>● Icon based</li> <li>● 內建圖庫</li> </ul>

## 2.3 理想的多媒體編輯工具應用於視覺化劇情呈現需求

在分析和評估目前幾種多媒體編輯工具應用於編輯出以”視覺化劇情呈現需求”過程特性之後，我們綜合出一個應用於編輯”視覺化劇情呈現需求”的多媒體編輯工具應該要符合下列三個特性：

- (1) 視覺化編輯環境
- (2) 支援條件控制
- (3) 支援互動功能

以下我們就對這三個特性做詳細的介紹。

### 1. 【視覺化編輯環境】：

互動式多媒體越來越普遍，使用互動式多媒體編輯工具的人也越來越多，一般使用者不見得具有撰寫文字程式的能力，因此在系統設計考量上，我們希望能夠以視覺化為基礎的編輯環境，讓使用者可以很方便的編輯，不需要陡峭的學習曲線就能夠編輯出”視覺化劇情呈現需求”的互動式多媒體成品。

### 2. 【支援條件控制】：

由於一般需求內容一定會包含條件式流程。因此，除了循序式的表達之外，必須提條件式流程之能力，才能滿足基本的需求內容。

### 3. 【支援互動功能】：

現行的多媒體編輯工具越來越強大，都具備有建立互動功能的機制。一般的靜態的播放展示，只是單向的內容呈現，已經無法滿足使用者的需求了。因此，支援建立互動的功能，能讓使用者編輯出來的”多媒體呈現需求”有互動的機制是非常重要的基本功能。



## 三、 多媒體編輯工具系統需求分析

### 3.1 視覺化的編輯方法

如何讓一般沒有程式背景的人都能夠很容易操作，不需撰寫撰寫文字程式碼即可輕輕鬆鬆編輯出豐富的多媒體劇情?為了達到這個目的，我們必須提供一個視覺化的編輯方法，使用者利用圖式瞭解元件的意義，不需要像傳統程式語言須透過記憶程式語言的指令與語法，透過文字編輯介面來編輯。由此來幫助一般的使用者不需手動撰寫文字程式，就能夠編輯出豐富的劇情。

#### 3.1.1 視覺化語言

關於視覺化程式語言(Visual Language)的研究已經發展多年，其簡單的定義如下:[9]

「使用者利用視覺化程式工具定義出基本原件，將基本原件組合成一個有意義的視覺化程式，之後透過視覺化程式系統，編譯且執行所編輯的程式。」

另外在 Visual Object-Oriented Programming Concepts And Environments 書中提出: [10]

「視覺化程式是被分別在本身如何被描述的和如何描述，被描述的是程式，按照程式語言的句法和語意學表達。」

因此視覺化程式語言有以下的特性: [9]

1. 具體性(Concreteness):具體的利用特定的圖示或線條等方法來表達一段程式。
2. 直接性(Directness):減少使用者達成目標所需的動作。
3. 清晰性(Explicitness):清楚直接的表達語意，不需要使用者去猜想。
4. 立即性(Immediate Visual Feedback):利用視覺化編輯工具編輯後，可以對內容直接顯現得到立即的回饋。

綜合以上對於”視覺化程式語言”的定義與特性，傳統撰寫程式的方法是用文字模式來撰寫，使用者利用鍵盤輸入依照自己的需求輸入符合程式語法的文字；然而，”視覺化語言”提供圖形為表達工具，使用對話表單、視窗、圖示等比喻來

和使用者溝通，使用者則利用滑鼠、鍵盤等設備來編輯所要表達的程式，系統會依使用者編輯完的內容產生一相對應的程式。

### 3.1.2 視覺化語言的實現

我們要如何實現以”視覺化語言”來做為編輯的主要方式呢?傳統程式語言的語法是由文字所描述，而視覺化語言的編輯方式就是把那些文字式描述的語法轉換成視覺化的語法，所謂的視覺化語法就是以圖示(Icon)、表格、動畫等模式來代表文字式語法。

使用圖示(Icon)來建立視覺化語言的優點包括：

- (1) 可讀性高(Readable)：人對具體事物的了解比較容易，所以直接用 圖形來呈現程式的流程、物件的關係等等，將文字程式的部份以實體狀態呈現在你眼前，而不必憑抽象的程式碼去想像最終的結果，可以讓寫作者能輕鬆寫作一個程式。
- (2) 使用者介面友善：圖形使用介面(GUI)讓使用者在操作上更加方便、學習上的速度也加快，讓初學者和非專業人員也可以輕鬆的使用。

### 3.1.3 視覺化的編輯模式

在編輯過程當中的一些條件式設定，例如流程控制，我們需要設計一個編輯環境讓使用者不需要去撰寫程式碼也不需要去看到它們的描述語言。因此這些控制我們必須以圖式的方式來表示，並以滑鼠拖拉的方式加上對話表單(Dialog Window)的協助，幫助使用者完成編輯動作。[11]

我們所設計的視覺化編輯方式如下：

- (1) 將所有控制 icon 方式來編輯：  
每個控制都以一個對應的圖示來代替，讓使用者在編輯時容易辨識、不需要記憶語法和指令。
- (2) 以拖拉及點擊的方式進行編輯：  
在編輯時，只需將 icon 拖拉至相對應編輯區域即可。

- (3) 以對話表單(Dialog Form)來輸入參數：  
應用對話表單，以適當的 Dialog Form 來幫助引導使用者輸入該指令的參數。
- (4) 條列編輯結果：  
系統會將使用者編輯的結果以表格條列的方式顯示在編輯畫面當中，不僅能幫助使用者觀看編輯結果的順序，也能幫助使用者之後進行修改調整順序。

## 3.2 多媒體的組成與軟體需求

綜合上述的研究，我們提出了以視覺化語言為基礎的編輯概念，設計以 icon 為編輯”視覺化語言”的基礎原件的編輯架構。為了方便使用者的編輯，我們設計了拖拉及點擊滑鼠的操作方式進行”視覺化需求呈現”的編輯。以上是針對使用對於系統的操作行為進行研究，接下來我們要探討如何將文字化的需求轉化成以視覺化呈現需求。

### 3.2.1 互動式多媒體的組成

我們的最終目標是將一個文字模式的需求，透過一個編輯工具，轉換成以”視覺化多媒體”的方式呈現。因此，我們先來了解一般互動式多媒體的架構是如何組成的。

一個互動式多媒體的組成包含了許多場景，場景就像是一個容器，每一個場景裡面包含了許多角色以及角色之間的劇情互動，每個角色是由其素材、動畫及位置大小所組成。當中素材指的其實就是所使用的多媒體 - 包含文字、圖片、聲音或是影片…等。[11][圖 14]

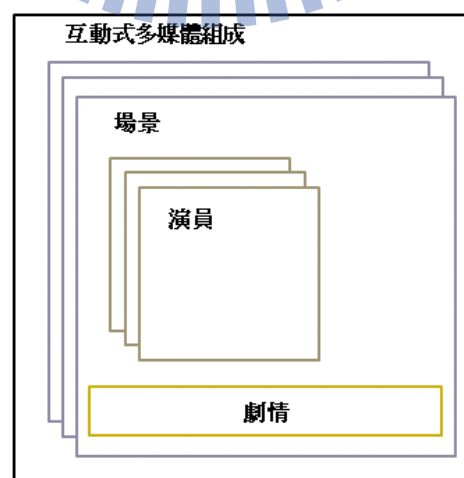


圖 14- 互動式多媒體的組成

## 3.2.2 Authoring Tool For Visual Software Requirement

### Representation

根據此篇論文” A visual and reuse-based paradigm for software construction” [3]的研究，一個以視覺化呈現的需求，是以四個基本元素所組成:[圖 15]

#### (1)演員(Actor)

- 用來表達軟體需求中的最小單位元素，稱之為 MRC (Multimedia Reusable Components)。

#### (2)場景(Scenes)

- 場景可視為一個可裝載演員的容器，也代表一部分的軟體需求的描述，透過場景中的演員來描述軟體需求。場景中也包含了” 場景的布局”，用來描述場景的陳列與展現。另一方面，透過” 場景劇本”(Scenario)演員與演員之間的演出順序，位置…等時間與空間的關係。

#### (3)關聯(Relationships)

- 關聯可分為兩類：
  - i. Actor 與 Actor 之間的關連：  
描述 actor 之間的出場順序，例如循序或是平行展現。
  - ii. Scene 與 Actor 之間的關連：  
描述 actor 在 scenes 中的大小位置…等狀態。

#### (4)需求專案(Requirement Project)

- 由於每個場景代表一個需求的描述，組合多個場景就可以組成一份需求專案。

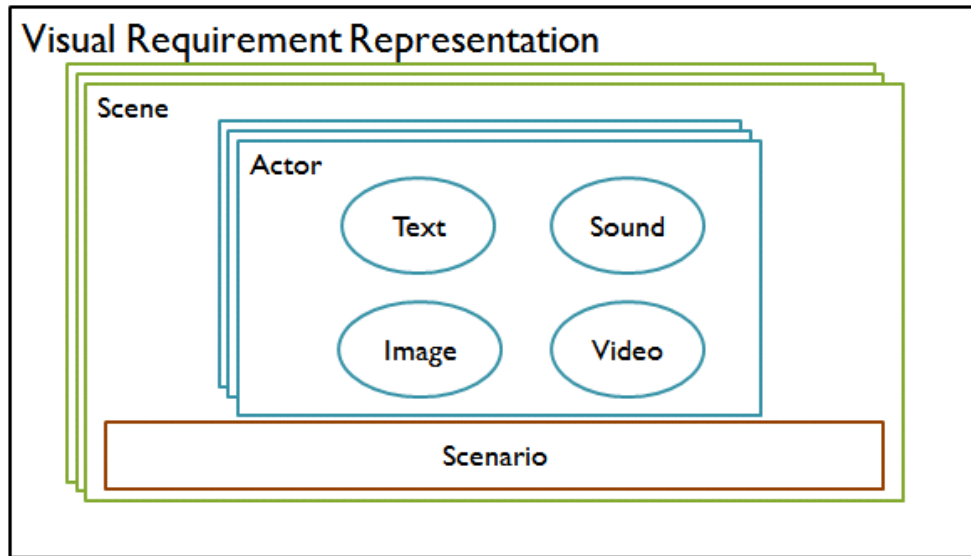


圖 15- Visual Requirement Representation

### 3.2.2 文字需求的轉換過程

根據多媒體組成的架構與視覺化呈現需求的基本元素與組成架構，我們就可以根據分析文字需求，並且加以分類出符合上述的基本元素，就可以利用編輯器編輯出以“視覺化劇情”來呈現需求。

我們首先來分析 ATM 提款功能的文字需求，我們發現我們可以將這份需求所描述的內容經由簡單的分析流程[圖 16]，分成 4 個類別，分別為“人”、“事”、“地”、“物”四個部分。

以下是我們分類的步驟：

- (1) 首先，我們分析名詞部分的詞彙；例如，“客戶”、“ATM”、“提款卡”...等。
- (2) 接者我們分析有關動作行為的部分；例如，“插入提款卡”、“輸入密碼”、“輸入提款金額”...等。
- (3) 之後根據上述的簡單分析，將分析出來的元素放在適當的 4 個類別中。

**銀行 ATM 系統需求(範例):**

- **提款**

客戶可以在 ATM 上提款，提款時先插入提款卡，輸入密碼。若輸入密碼錯誤，系統會秀出密碼錯誤的訊息，若密碼正確，系統秀出要使用者輸入提款金額數。客戶輸入金額，提款成功。若提款金額大於帳戶金額，則系統秀出金額不足的錯誤訊息。

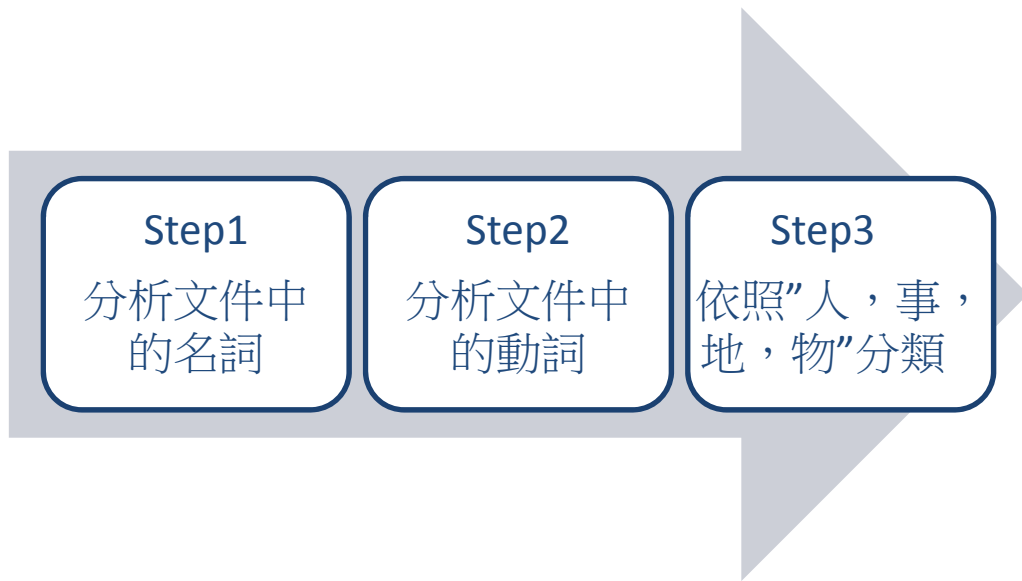


圖 16- 文字需求分析流程

因此經由上面的分析後，我們可以將一個文字描述的需求，經過分析後，以下列分類方式呈現：

類別	元素
人	客戶
事	插入提款卡，輸入密碼，輸入提款金額
地	銀行
物	ATM，提款卡，錢，訊息

因過分類後的元素在和視覺化呈現的需求的四個基本元素-演員(Actor)、場景(Scenes)、關聯(Relationships)、需求專案(Requirement Project)彼此做對應而產生下列的對應結果。[圖 17]

- (1) 演員: 提款人、提款機、提款卡、錢...等。
- (2) 場景: 銀行。
- (3) 關聯: 插入提款卡，輸入密碼，輸入提款金額。
- (4) 需求專案: 提款功能。

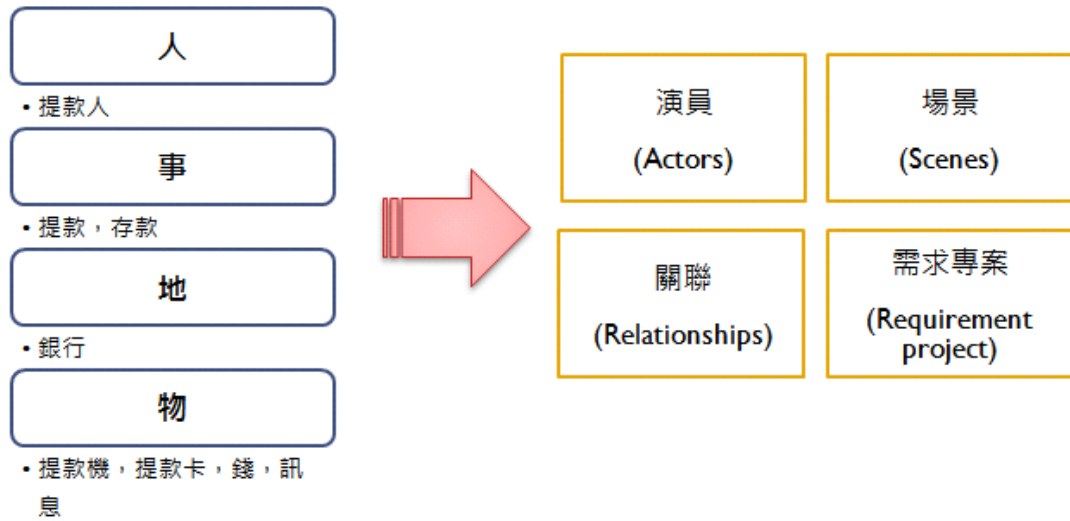


圖 17- 視覺化呈現的需求的四個基本元素對應



### 3.3 多媒體編輯工具系統需求

在前兩小節中我們探討了視覺化的編輯方法，與探討視覺化語言，也根據視覺化劇情呈現需求的四個基本元素，將文字需求分析歸類到對應的基本元素內。因此，我們要如何編輯一個以視覺化劇情呈現需求的多媒體內容呢?可以下列流程來編輯。[圖 18]

- (1) 文字需求: 使用者以文字需求為基礎。
- (2) 選取場景: 使用者根據分析後的結果，根據四大基本元四中的”場景”，選取適當的背景圖。
- (3) 選取 icon: 使用者根據分析後的結果，根據四大基本元四中的”演員”，從 icon 圖庫拖曳出所需 icon 當作演員。將所選的 icon 演員，以滑鼠拖曳的方式在場景中擺放適當的位置。
- (4) 編輯需求流程: 使用者依據需求的內容，設定演員之間的關連性與順序性，並且一步一步的將此關係加入需求流程列中。
- (5) 播放: 使用者按播放功能，觀看編輯結果。
- (6) 產生循序圖: 系統依據使用者條列的流程結合角色，幫使用者產生一份循序圖提供一個完整的流程預覽供使用者參考。

另外，使用者在編輯過程中，可在步驟 2 到步驟 5 之間循環修正。

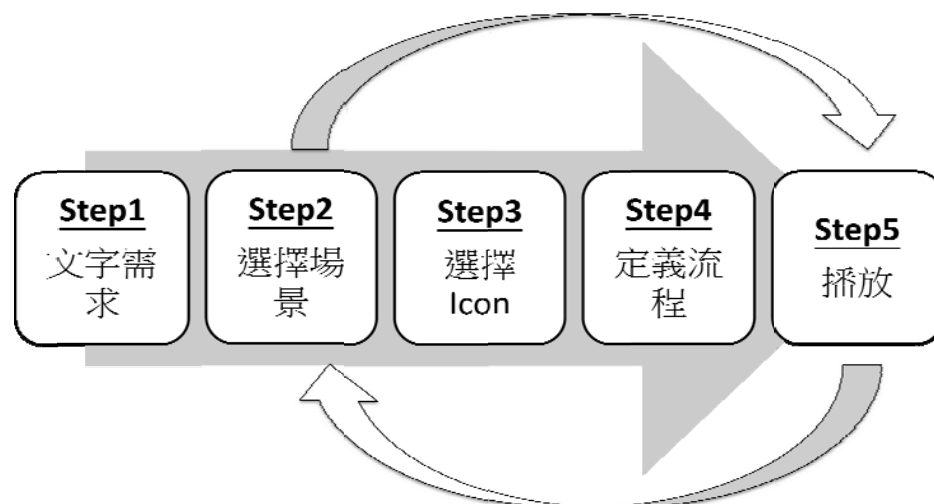


圖 18- 視覺化劇情呈現需求編輯流程



### 3.3.1 編輯機制

根據上述的編輯流程，我們要如何設計一個方便使用者的編輯機制呢？  
本系統以下列編輯機制提供使用者編輯：

- (1) 將所有控制”圖示化”方式來編輯：編輯時容易辨識、不需要記憶語法和指令。
- (2) 以拖拉及點擊的方式進行編輯：不需手動撰寫程式碼即可完成編輯。
- (3) 以對話表單(Dialog Form)來輸入參數：以適當的 Dialog Form 來幫助使用者輸入該指令的參數。
- (4) 條列編輯的結果：較直覺、方便閱讀。

### 3.3.2 系統功能

本系統在功能上，根據“Visual Software Requirements Definition Environment / Atsushi Ohnishi”這篇研究做了下列的功能加強：

- (1) 增加互動機制：  
使用者透過互動行為(拖曳，click)，來表現需求的內容。
- (2) Icon 定義彈性：  
Icon 無特定意義，允許使用者在建構時給予特定角色。
- (3) Icon 擴充容易：  
擴充 Icon 時，只需將圖檔複製到特定 Icon 分類目錄即可。
- (4) 產生條列式需求：  
根據使用者編輯的流程，產生條列式需求，方便使用者總覽整個需求流程。

## 四、多媒體編輯工具系統設計與實作

### 4.1 編輯畫面設計

根據上節，我們已經分析了”多媒體編輯工具系統”的編輯機制與編輯流程，以下我們先來介紹本系統的編輯畫面：

(1) 主畫面：主畫面分成三大部分：

- 功能鍵：功能鍵包含了”編輯流程”，”撥放流程”，”變更背景”。
- 主要編輯區：主要編輯區可讓使用者挑選背景，並從 icon 圖庫中拖曳出所需的 icon 圖示，並移動到適當的位置。
- 分類 Icon：分類 icon 以 group 方式將使用者所常用的 icon 分類到不同 tab 中。使用者可以切換 tab，以拖曳方式選去所需的 icon。



圖 19- 多媒體編輯工具系統主畫面

(2) 選定 Actor:

使用者從根據所需的 icon，以拖曳方式選取到編輯區。



圖 20- 多媒體編輯工具系統選取 actor

(3) 編輯流程:依據不同的 Action 輸入不同的 Actor。

- Action=RightClick:當 Action=RightClick 時,必須輸入 Actor 1, Actor 2, Menu 1, Menu 2, Actor3, Actor4, 其中 Menu 1 代表使用者觸發後會出現 Actor 3, Menu 2 代表使用者觸發後會出現 Actor 4。



圖 21- 多媒體編輯工具系統定義流程(Action=RightClick)

- Action=DoubleClick:當 Action=DoubleClick，使用者必須輸入 Actor 1, Actor 2 與 Actor 3。當使用者 double click Actor 1 時，會將 Actor 3 秀出。

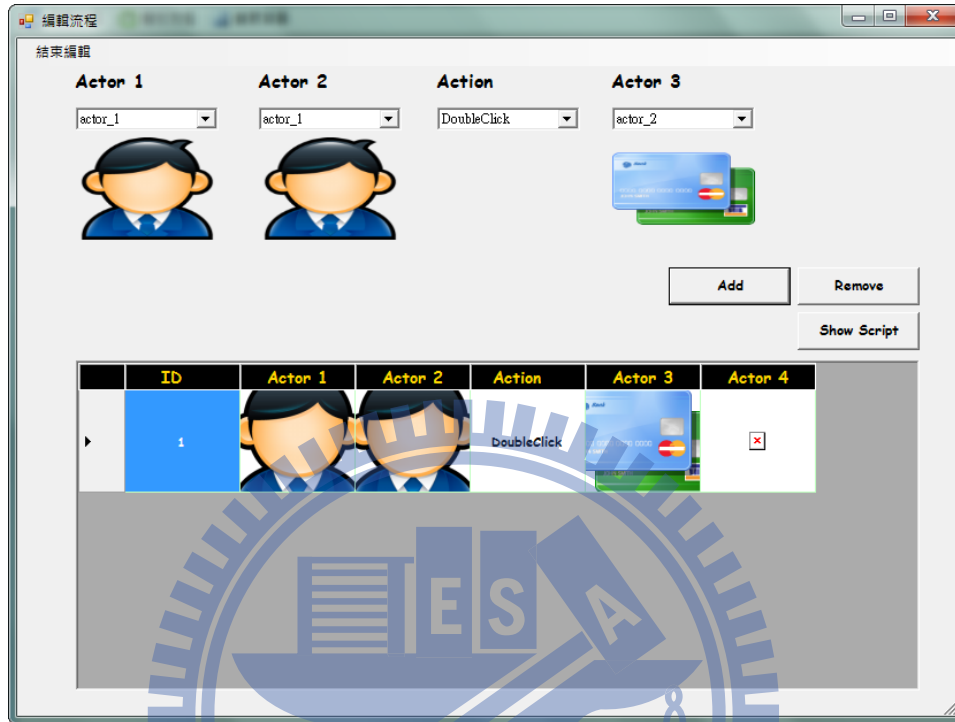


圖 22 -多媒體編輯工具系統定義流程(Action=DoubleClick)

- Action=DragDrop:當 Action=DragDrop，使用者必須輸入 Actor 1，Actor 2 與 Actor 3。當使用者將 Actor 1 拖曳到 Actor 2 時，會秀出 Actor 3

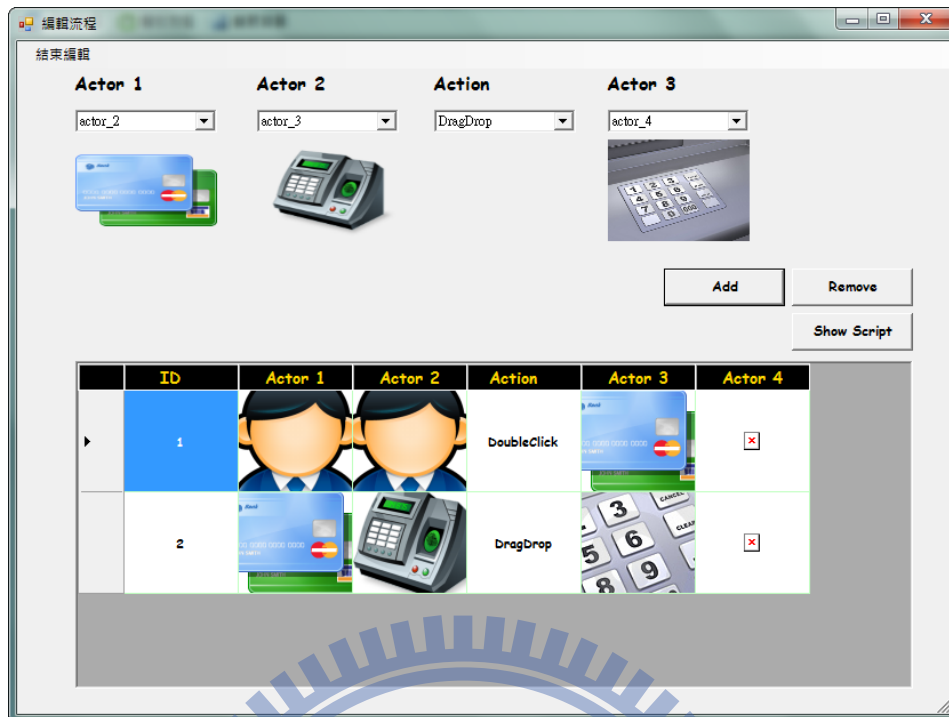


圖 23-多媒體編輯工具系統定義流程(Action=DragDrop)

(4) 播放互動流程:

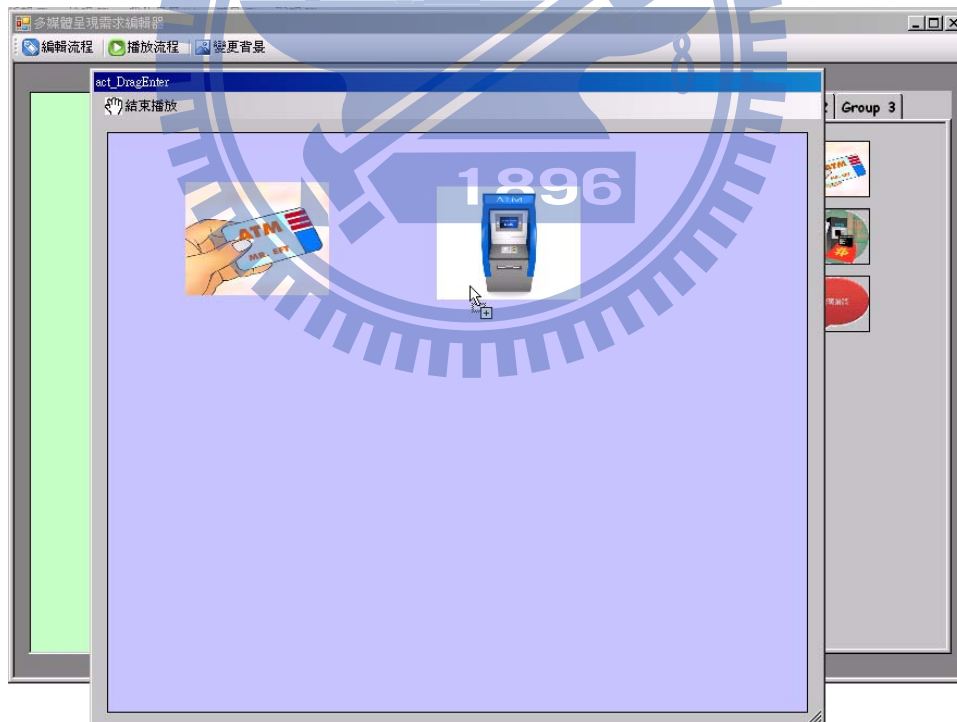


圖 24- 多媒體編輯工具系統播放互動流程

(5) Script 預覽:

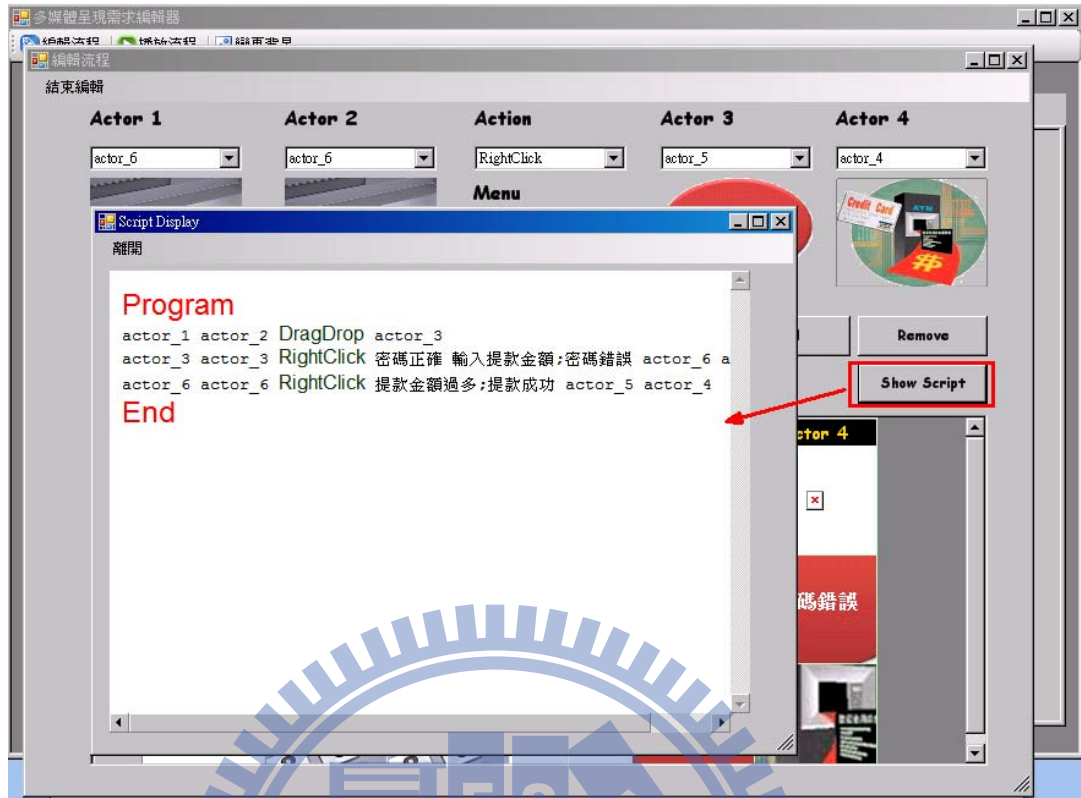


圖 25- Script 預覽

## 4.2 系統架構設計

上一小節，我們設計了系統的操作介面，但為了達到視覺化的編輯，使用者不會有輸入文字程式碼的動作，因此系統要將使用者的編輯動作轉化成為系統所能認知的描述。我們由以下的系統架構圖來說明各個原件的功能。[圖 26]

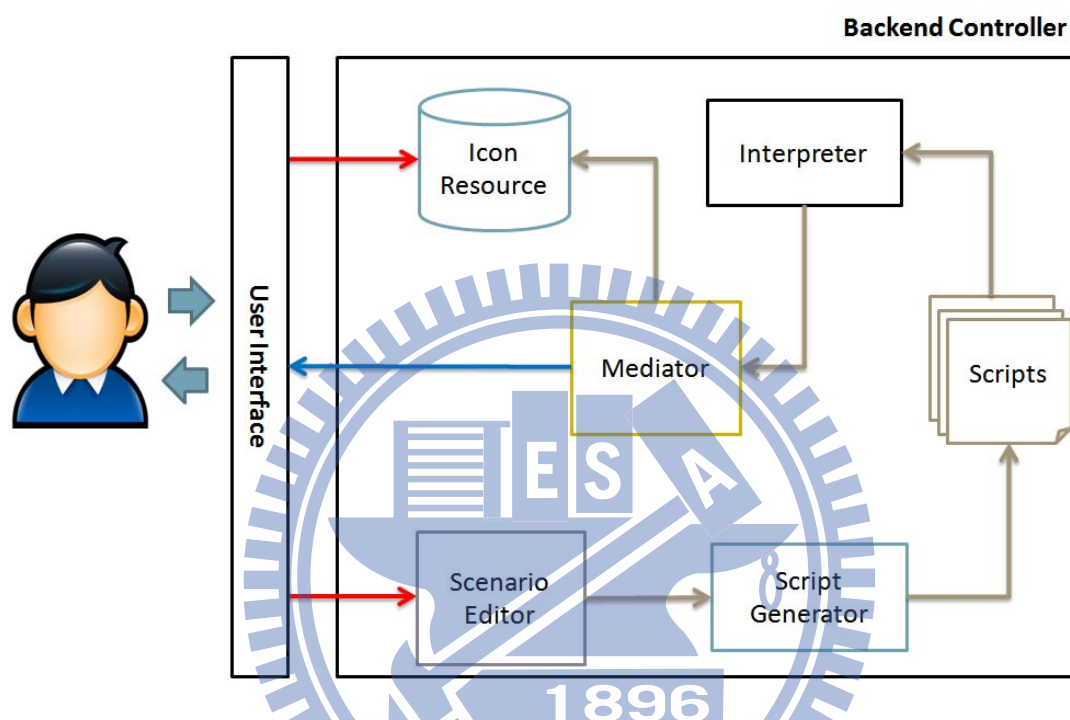


圖 26- 多媒體編輯工具系統架構

我們可以看到，此系統最主要有 2 大部分，分別是 User Interface 與 ”後端控制元件” (Backend Controller)。以下針對”後端控制元件”做一個詳細的介紹：

- (1) Scenario Editor:  
為流程編輯原件，主要功能是讓使用者編輯需求流程。
- (2) Script Generator:  
負責辨識需求流程，並根據定義的語法將辨識後的結果轉化成定義的 script language。
- (3) Scripts:  
一個 Scripts 暫存區，儲存經由 Script Generator 所產生的 Scripts。

- (4) Interpreter:  
Script Language 的直譯器，Interpreter 會將從 Scripts 讀入的程式碼解譯成相對應的動作行為與角色。
- (5) Mediator:  
根據 Interpreter 所解譯出的互動行為及角色，到 Icon Resource 中取出對應的角色。當使用者進行撥放或互動行為時，則負責中央控管 actor 之間的互動，而非由各自 actor 彼此交錯控制互動行為。
- (6) Icon Resource:  
Icon 的儲存中心，提供 icon 分類儲存，並提供使用者在編輯時所需要的 icon。

舉一個例子，使用者根據“Icon Resource”這個原件所提供的 icon，透過介面拖曳選取，接下來利用“Scenario Editor”所提供的編輯劇情功能，編輯出條列式的流程並轉交給“Script Generator”處理。經由“Script Generator”將條列式流程轉譯成為定義的“Scripts”後便完成了編輯動作。當使用者進行播放功能時，系統將“Scripts”經由“Interpreter”轉譯成系統的控制程式，透過“Mediator”來控制 icon(actor)之間的互動行為。

回顧視覺化呈現的需求的四個基本元素-演員(Actor)、場景(Scenes)、關聯(Relationships)、需求專案(Requirement Project)我們可以將上述的系統元件的功能做個對應。演員與場景都是透過 Icon Resource 來提供、關聯就是使用者透過本系統設計出的劇情，並轉換成 actor 之間的互動並由 script 紀錄、而專案需求則是透過透過本系統設計出的某一段劇情。

## 4.2.1 系統語法定義

“Script Generator”將條列式的流程透過定義的語法規則，將其轉換成“Scripts”；以下介紹本系統所定義的語法。



```

<program> ::= program <command>* end
<command> ::= <actor><actor><action><actor>
<actor> ::= actor<id>
<id> ::= a|b|c|...|z|0|1|2|...|9||A||B||C|...|Z|
<action> ::= drag | right_click | double_lick | mouse_over

```

以下是透過上述定義的語法所編輯出的 script 範例:

```

program
  drag actor_a actor_b actor_c
  right_click actor_c
  mouse_over actor_a
  drag actor_b actor_d actor_e
end

```

## 4.2.2 系統設計樣式

設計模式這個術語是由 Erich Gamma 等人在 1990 年代從建築設計領域引入到計算機科學的。它是對軟體設計中普遍存在（反覆出現）的各種問題，所提出的解決方案。

設計模式並不直接用來完成程式碼的編寫，而是描述在各種不同情況下，要怎麼解決問題的一種方案。物件導向設計模式通常以類別或物件來描述其中的關係和相互作用，但不涉及用來完成應用程式的特定類別或物件。設計模式能使不穩定依賴於相對穩定、具體依賴於相對抽象，避免會引起麻煩的緊耦合，以增強軟體設計面對並適應變化的能力。

在本系統中的兩個主要的元件”Interpreter”與”Mediator”是根據軟體工程領域中的設計樣式(Design Pattern)[12,14,15]為基礎應用於本系統的實作中。

我們分別簡單的探討這兩個設計模式是如何在本系統中應用:

### (1) Interpreter Pattern:

利用自訂的語言來表現程式要執行的內容，但是執行時必須先經過翻譯程式分析自訂語言並執行，這個翻譯程式稱作”直譯器”。使用此 pattern 前必須先定義好語言規格，例如常見的 BNF(Backus-Naur Form)用來描述程式語言的文法。以下

是”Interpreter” design pattern 的 UML class diagram[圖 28]。"AbstractExpression" 一個抽象的 expression，定義 interpret 這個方法的介面。"TerminalExpression"非終點的表示式，"NonterminalExpression"終點表示式。"Context"則提供文法解析所需要的方法。

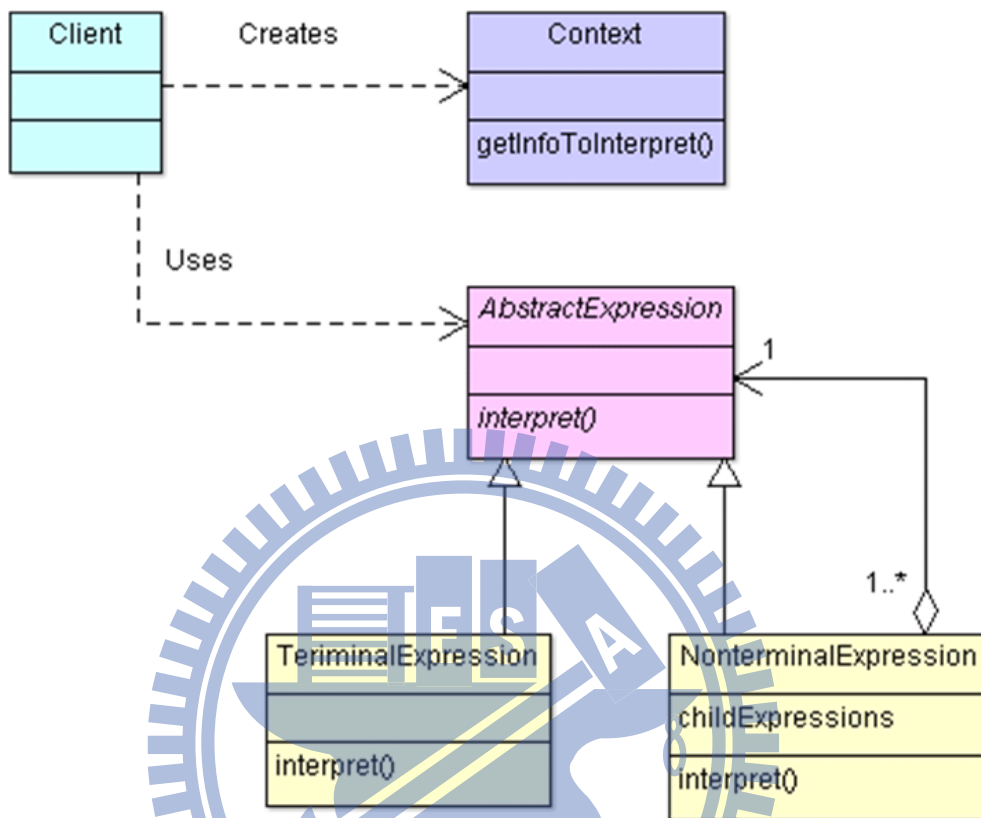


圖 27- Interpreter class diagram

## (2) Mediator Pattern:

利用仲介者物件來封裝一系列關於物件交互行為，負責調節、傳遞的工作。在組織物件工作的時候，物件彼此之間可能相互依賴，這就使得物件之間的耦合性相對的提高，最差的情況下，所有的物件都知道彼此的存在，這會使得系統的重用性降低。以本系統為例，當使用者定義各個 actor 之間的互動行為後，彼此並無直接耦合關聯，因為所有 actor 都是可以重用的元件，因此，必須將彼此的直接關聯性透過仲介者來處理。以下是”Mediator” design pattern 的 UML class diagram。[圖 28]

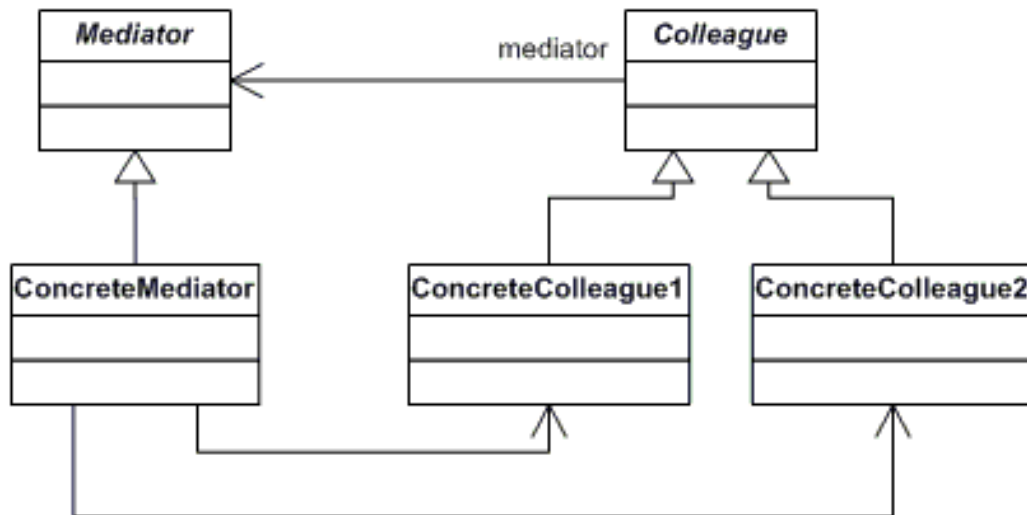


圖 28- Mediator class diagram

### 4.3 系統 Class Diagram

- frmMain: 為系統主要呈現視窗，提供使用者拖曳所需 icon 與進入編輯或播放式窗。
- frmEdit: 提供使用者編輯需求流程。
- frmScript: 依據使用者編輯的流程所產生的 script 秀出，並將關鍵字上色。
- frmPlay: 依據使用者編輯的結果播放並與使用者操作產生互動。
- modUtil: 提供各 form module 所需的 common method.

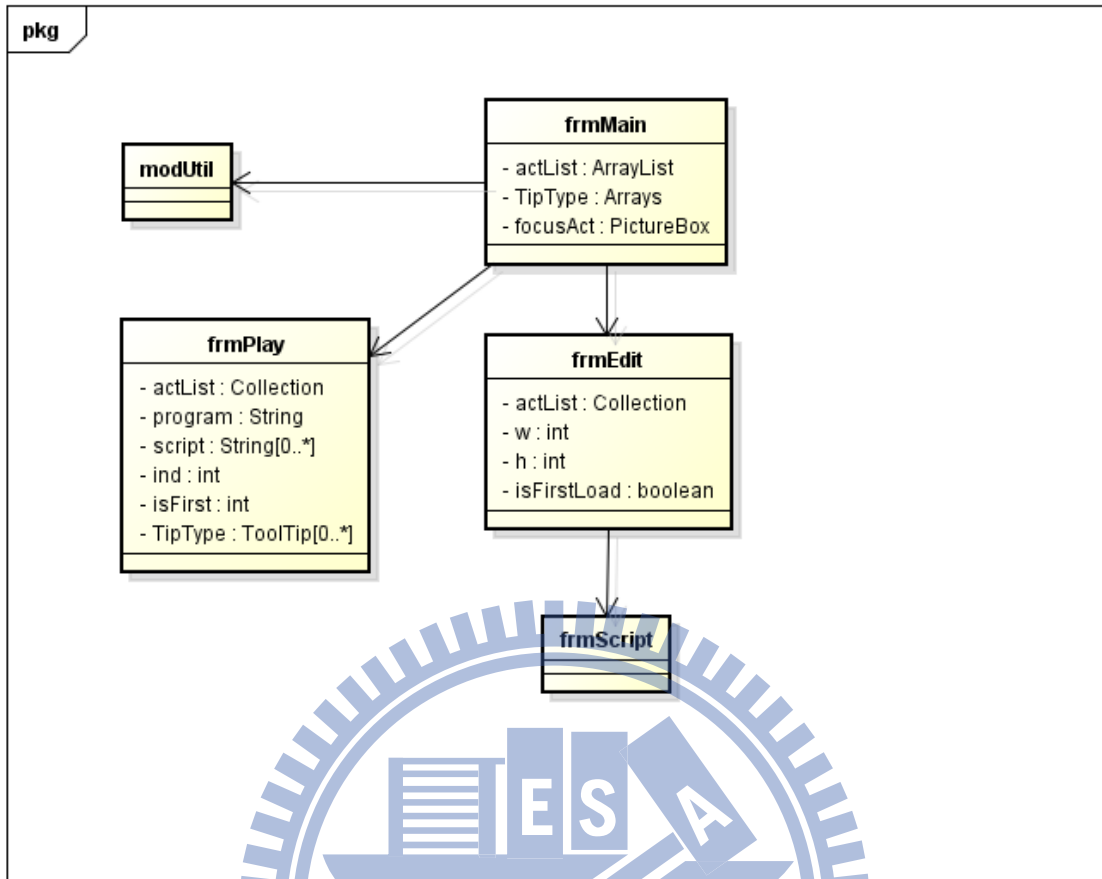


圖 29- System Class Diagram

## 4.4 系統 Sequence Diagram

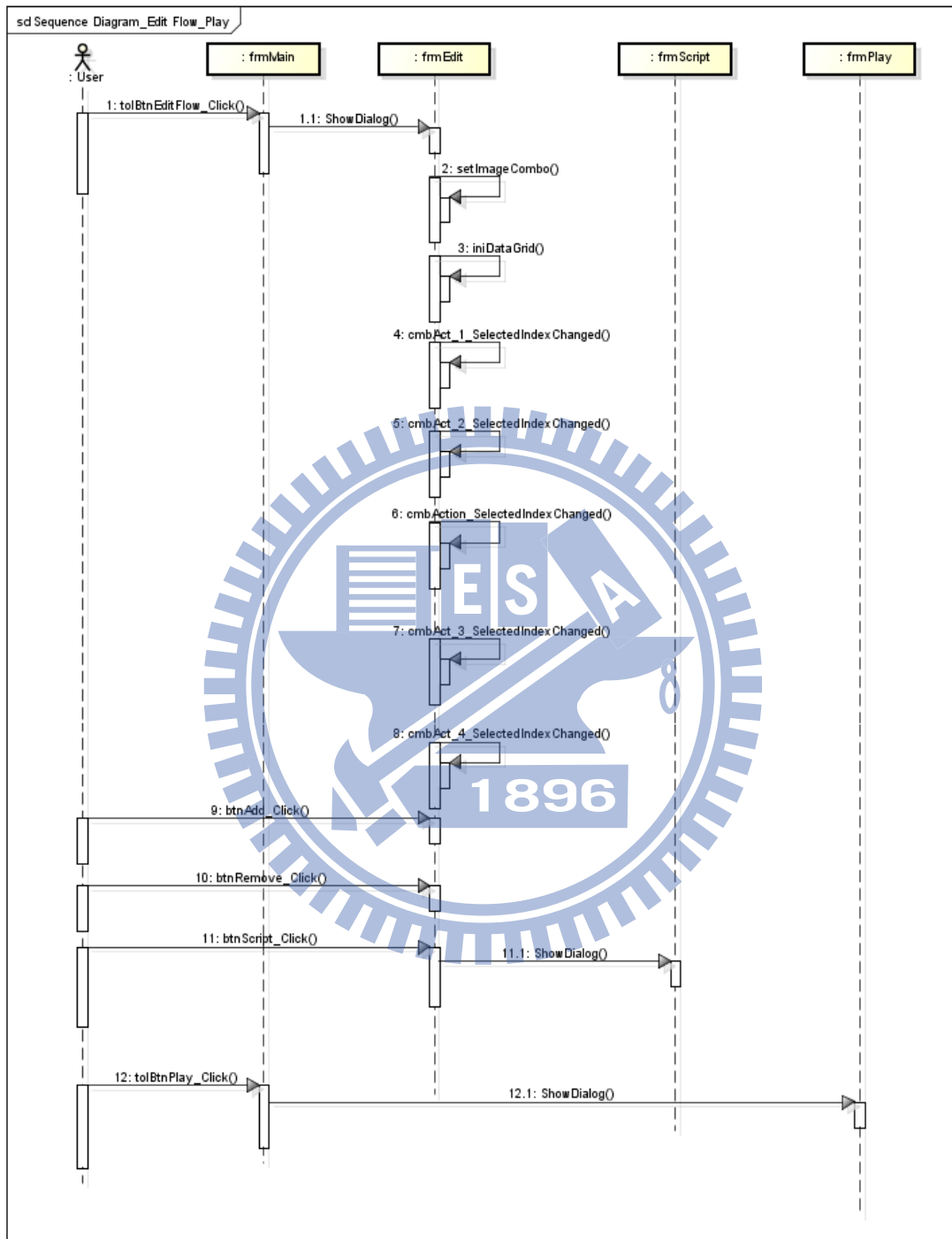


圖 30 - System Sequence Diagram

## 五、 實作範例

### 5.1 範例

在這一節當中我們會以 ATM 提款流程為例子，介紹如何用我們的視覺化劇情呈現需求編輯器實際編輯出需求的內容。

#### 銀行 ATM 系統需求(範例):

- 提款

客戶可以在 ATM 上提款，提款時先插入提款卡，輸入密碼。若輸入密碼錯誤，系統會秀出密碼錯誤的訊息，若密碼正確，系統秀出要使用者輸入提款金額。客戶輸入金額，提款成功。若提款金額大於帳戶金額，則系統秀出金額不足的錯誤訊息。

#### (1) 新增 icon:

使用者在本系統的目錄中建立 icon 分類目錄:

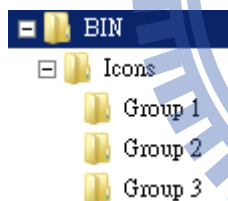


圖 31- 新增 icon

#### (2) 選取 icon:

拖曳出使用者，提款機，提款卡，輸入鍵盤，money，錯誤訊息等圖示。



圖 32- 選取並拖曳 icon

(3) 定義流程:

根據文字需求，以下列流程編排。

1. 提款卡插入提款機，show 出輸入密碼畫面。
2. 輸入密碼錯誤時，show 出錯誤訊息。
3. 輸入密碼正確時，且提款成功時，show 出鈔票圖示。



圖 33-定義流程:提款卡插入提款機，show 出輸入密碼畫面



圖 34-定義流程:提款卡插入提款機，show 出輸入密碼畫面





圖 35-定義流程:輸入密碼錯誤時, show 出錯誤訊息

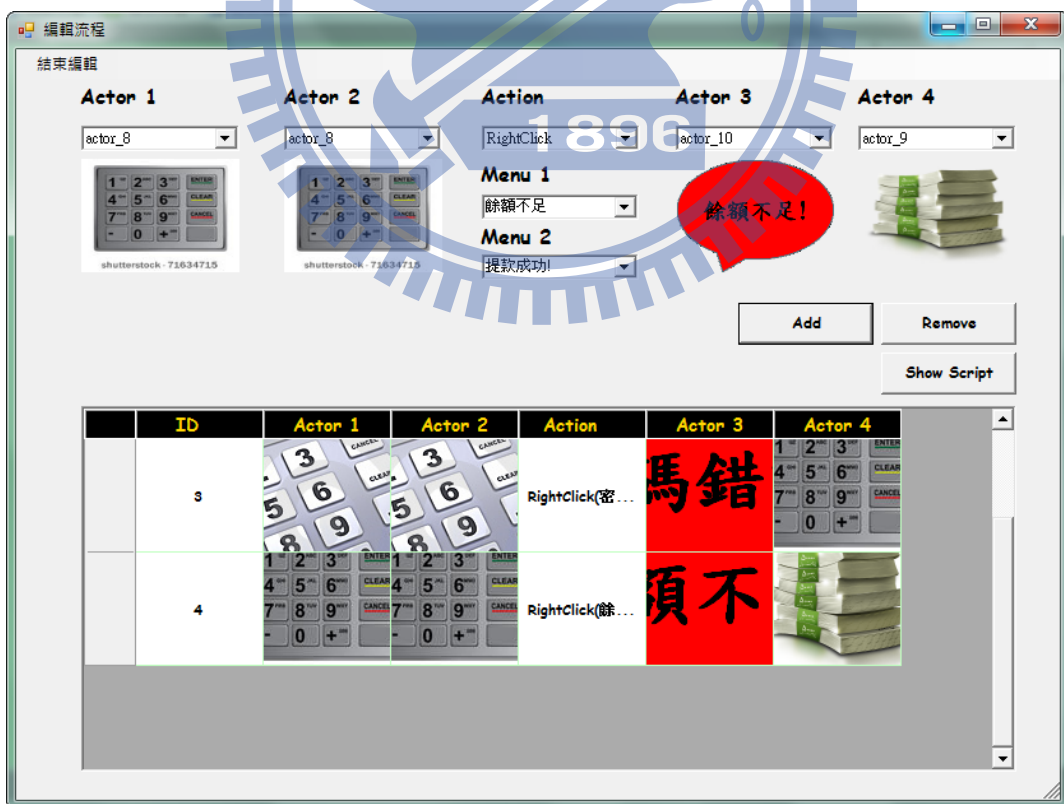


圖 36-定義流程:輸入密碼正確時, 且提款成功時, show 出鈔票圖示

(4) 檢視 Script: Highlight 語法部分。

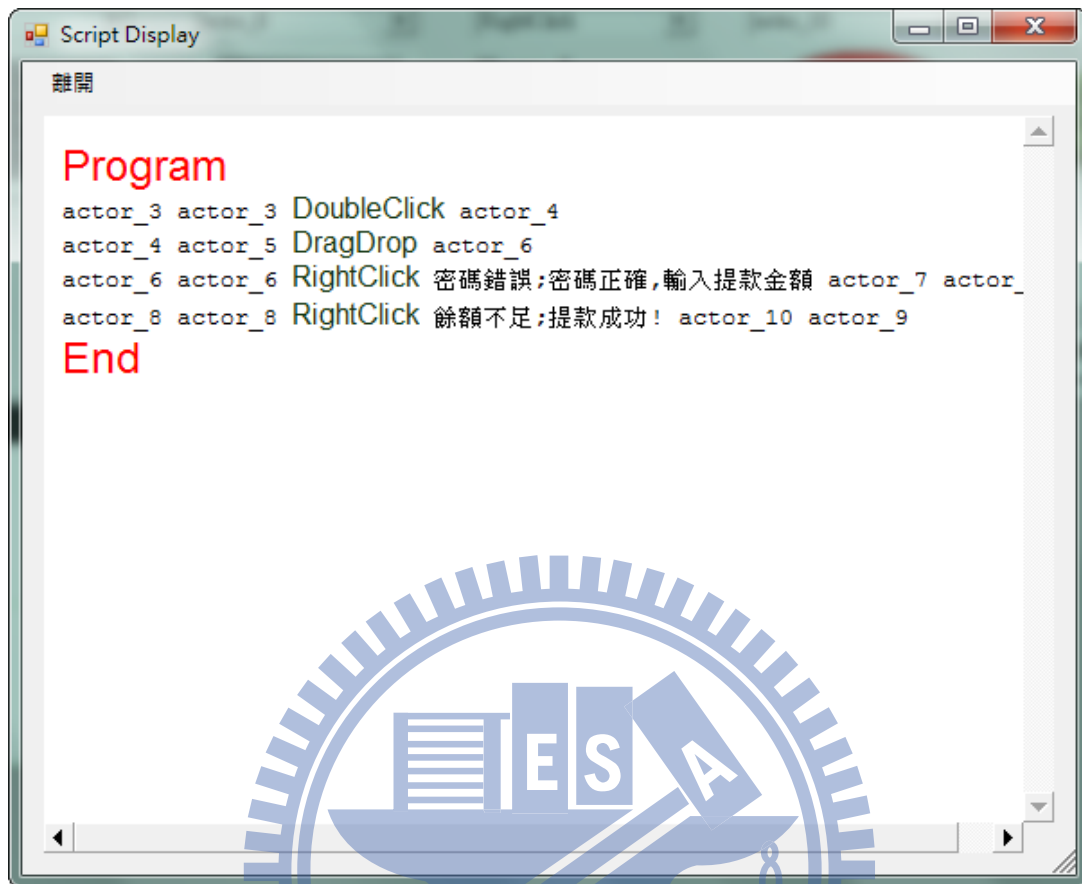


圖 37-檢視 Script

(5) 播放流程:

1. Double click 使用者 icon，系統 show 出提款卡。
2. 拖曳提款卡插入提款機，show 出輸入密碼畫面。
3. 右鍵輸入密碼錯誤時，show 出錯誤訊息。
4. 右鍵輸入密碼正確時，show 出輸入提款金額。
5. 右鍵輸入餘額不足時，show 出錯誤訊息。
6. 右鍵輸入時提款成功時，show 出金錢圖示。

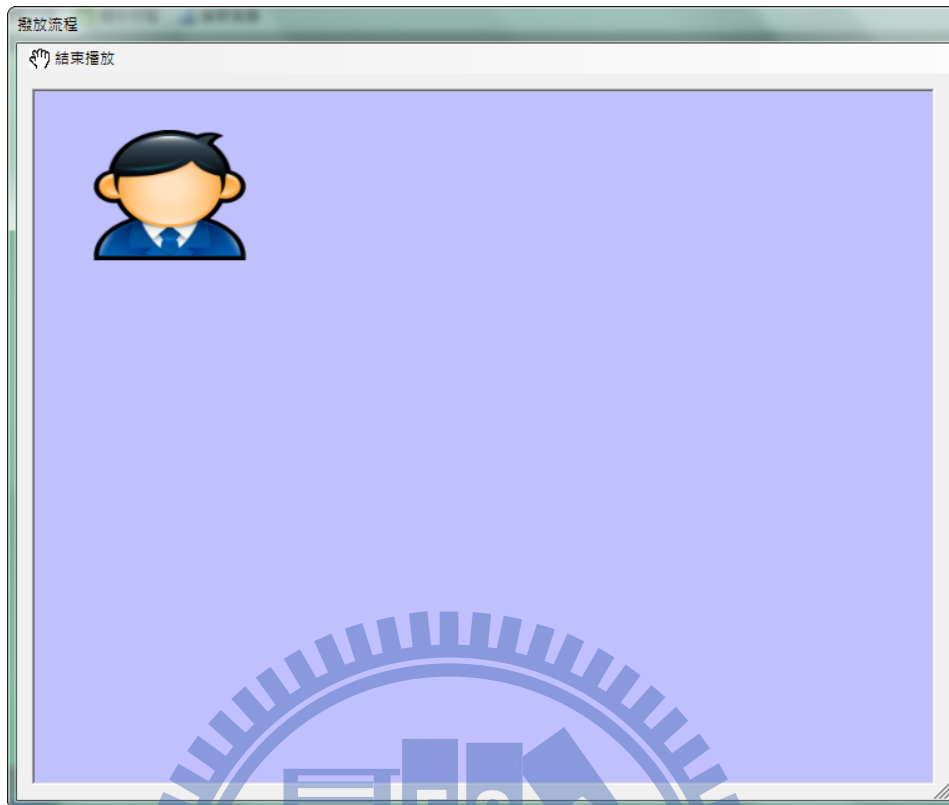


圖 38-播放流程:Double click 使用者 icon，系統 show 出提款卡

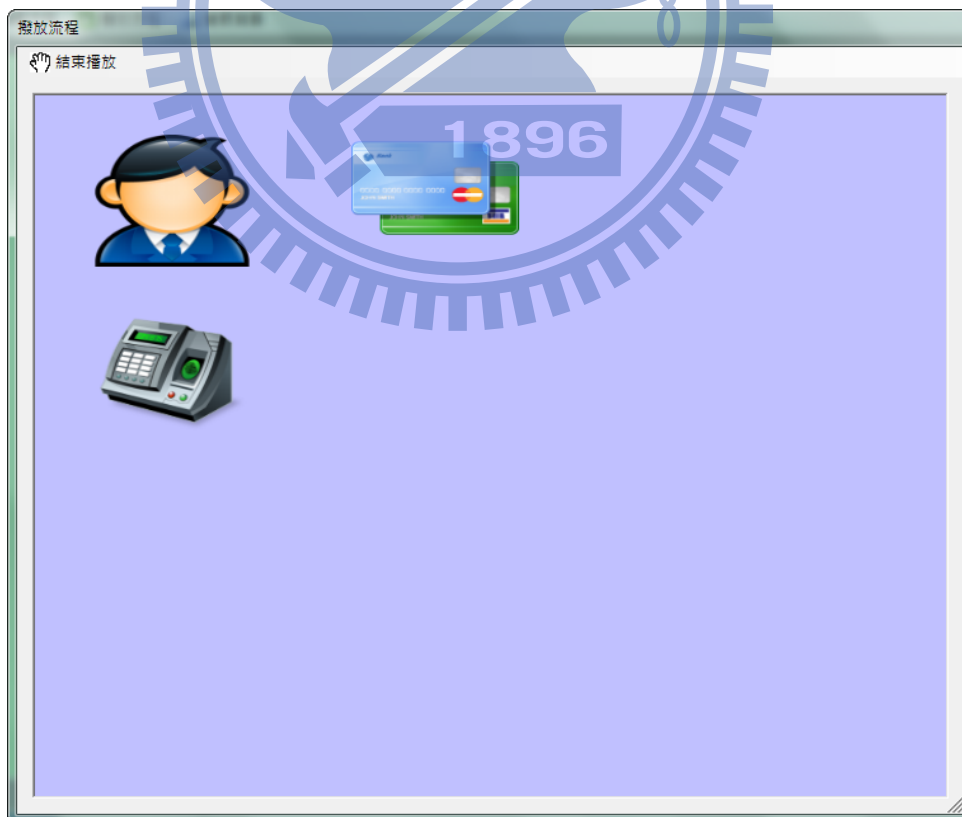


圖 39-播放流程:拖曳提款卡插入提款機，show 出輸入密碼畫面

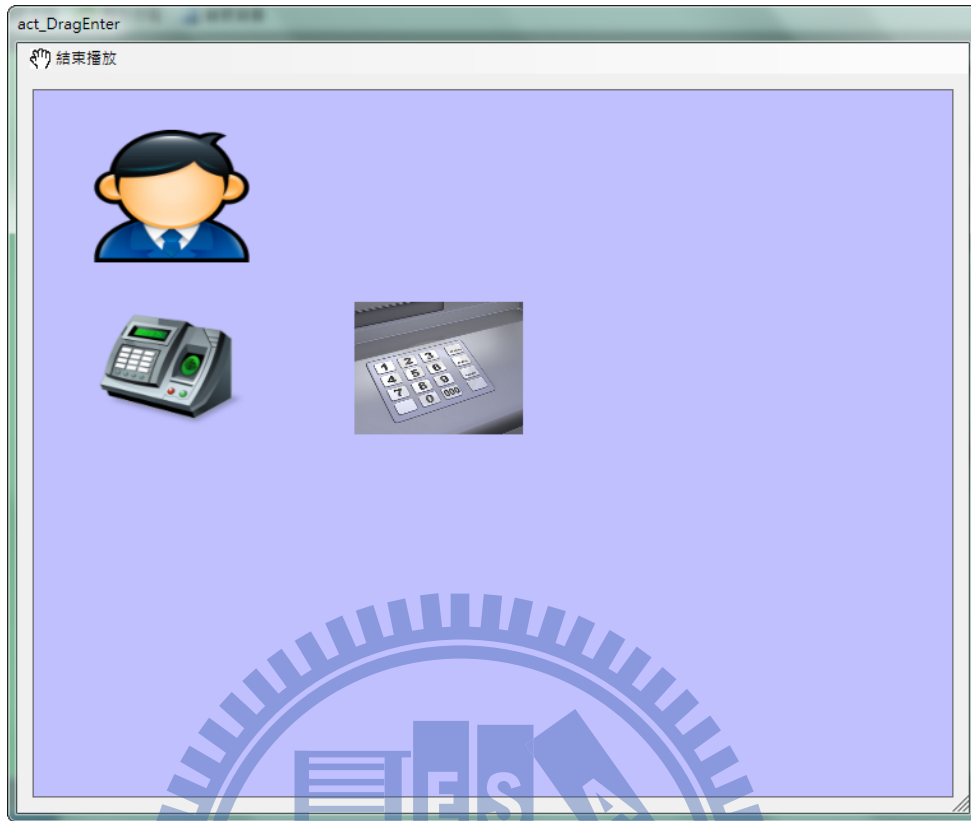


圖 40-播放流程:拖曳提款卡插入提款機，show 出輸入密碼畫面



圖 41-播放流程:右鍵輸入密碼錯誤時，show 出錯誤訊息

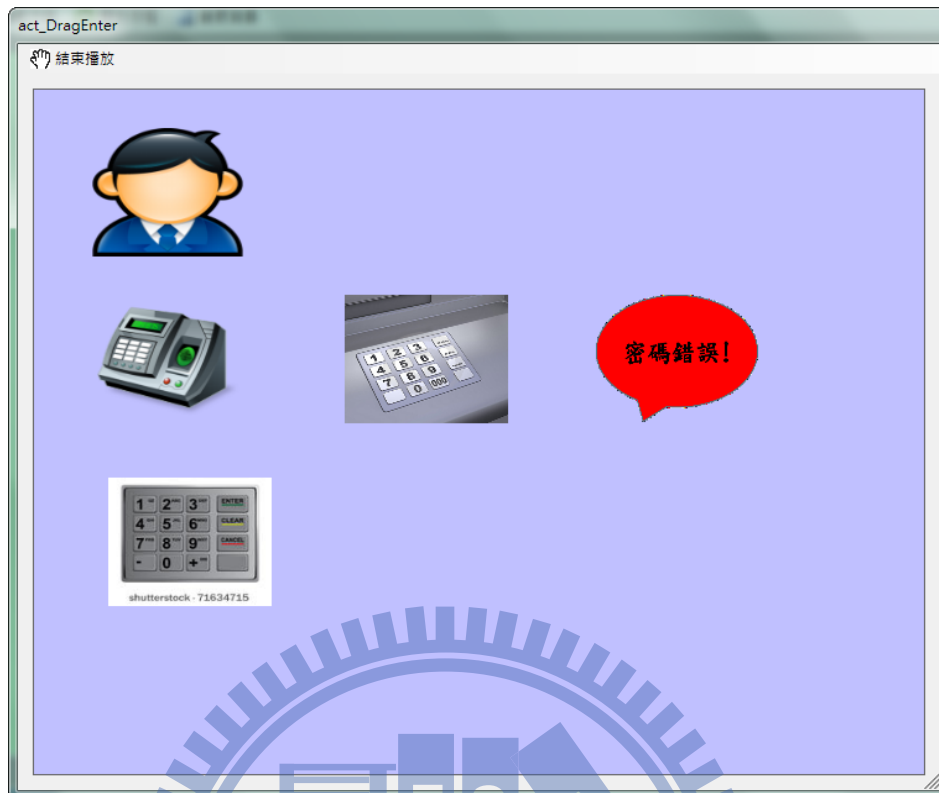


圖 42-播放流程:右鍵輸入密碼錯誤時，show 出錯誤訊息



圖 43-播放流程:右鍵輸入餘額不足時，show 出錯誤訊息

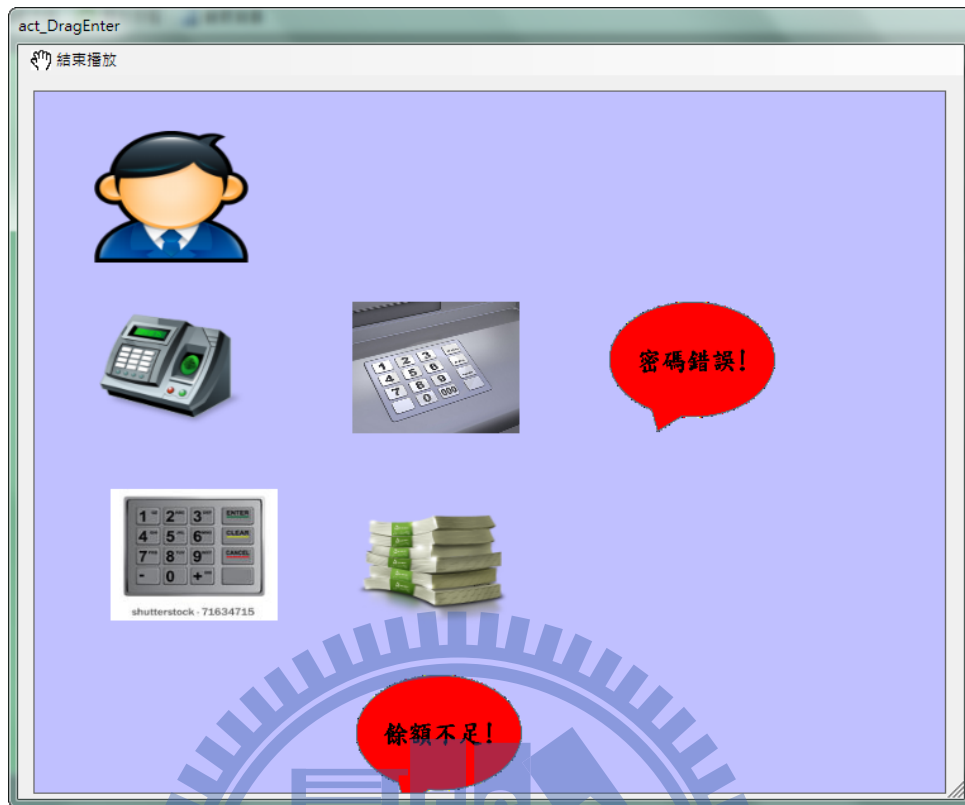


圖 44-播放流程:右鍵輸入時提款成功時，show 出金錢圖示

## 六、 結論

### 6.1 總結

利用互動式多媒體的來編輯以視覺化劇情呈現需求使的使用者與系統開發人員之間的溝通更為自然與直覺，而編輯的方式也利用視覺化程式語言的特性，利用圖示及拖曳方式進行編輯，使得不會文字程式語言的使用者也能編輯出具有互動性與邏輯性的互動式多媒體成品來表現需求。以下是本論文的總結：

- 本論文探討需求的呈現方式與比較，並實作編輯工具，以銀行 ATM 提款機提款流程為例，編輯出以視覺化呈現需求。
- 以編輯工具而言：我們建立了一個視覺化劇情編輯環境，使用者不需要撰寫文字式的程式而運用視覺化的圖示或對話表單來編輯以視覺化劇情呈現需求。
- 在視覺化呈現需求加上互動機制，除了靜態的流程表達，也讓使用者更能體會需求內容。



## 6.2 未來發展方向

我們提出以下可以繼續發展的方向，讓本論文的系統增加新的功能以及更廣泛的應用：

- 以此工具進行實驗，分成兩組，一組以一般文字模式呈現需求、另一組以本工具編輯出視覺化劇情需求，以證明在文字化與視覺化兩者的意義是相同的。
- 在多媒體編輯器中，提供儲存機制，方便使用者可以暫存編輯內容。
- 除了以 icon 來呈現，應擴充以聲音，影片型式來呈現需求。
- 可將編輯成果輸出，並可以隨時撥放。
- 本系統是使用 Window 為發展環境，故只能適用於 Windows 系統之上。將來可往其他不同平台上發展，如在 LINUX 或其他作業系統(OS)上。
- 本實驗室所研發的編輯手是一個完整的多媒體編輯工具，因此可以將此概念移植到編輯手擴充其功能。





## 參考文獻或資料

- [1] Ian Sommerville, “Software Engineering 6<sup>th</sup> Edition”.
- [2] Hughes Department of Defense Composite Software Error History.
- [3] Wu-Chi Chen ; Dr. Deng-Jyi Chen , A visual and reuse-based paradigm for software construction,1998
- [4] Marina Jirotko , “Ethnography by Video for Requirements Capture” ,1995.
- [5] Jacob Buur , “Ethnographic Video as Design Specs” ,2010.
- [6] Sascha Konrad , “Visualizing Requirements in UML Models”, 2006.
- [7] Atsushi Ohnishi , “Visual Software Requirements Definition Environment”, 2002.
- [8] Chorng-Shiuh Koong, “A Component-based Visual Scenario Construction Environment for Non-Programming Users to Create Interactive Electronic Books”, PHD Thesis, N.C.T.U. Taiwan, October, 2000.
- [9] Burnett, Margaret, Goldberg, Adele, and Lewis Ted, "Visual object-oriented programming Concepts and environments," Manning, Greenwich, 1995.
- [10] Chang, S.K, et al., ”The future of visual languages”, Visual Languages, 1999. Proceedings. 1999 IEEE Symposium on, 1999
- [11] Shu-Ying Chiang, “The Design and Implementation of a Multimedia Test Question Template System Based on an Enhanced Visual Scenario Authoring Tool”, Master Thesis, N.C.T.U. Taiwan, 2005.
- [12] Gamma, Johnson, Helm, Vlissides, “Design Patterns: Elements of Reusable Object-Oriented Software”, Addison Wesley Longman, 1995
- [13] Adobe, [On-Line]. Available:www.adobe.com
- [14] James William Cooper, “Visual Basic design patterns: VB 6.0 and VB.NET”, Addison-Wesley Professional, 2002

- [15] Yu-Wei Wang, "On the Study of Reusable Visual Scenario", Master Thesis, N.C.T.U. Taiwan, 2011
- [16] Cia-Yu Chiu, "The Design and Implementation of the Visual Requirement Representation Template and its Customization Web-based System - Using Multimedia Yearbook as an example", Master Thesis, N.C.T.U. Taiwan, 2009

