

國立交通大學

資訊科學系

碩士論文

以 XML 為基礎的行動應用程式開發工具

XML-based Mobile Application Development Kit

研究生：沈聖博

指導教授：袁賢銘 教授

中華民國九十三年六月


# 以 XML 為基礎的行動應用程式開發工具

研究生：沈聖博

指導教授：袁賢銘

國立交通大學資訊科學研究所

## 摘要



隨著行動裝置(mobile device)近年來的蓬勃發展，造就了現在行動裝置的多樣性。於是，在行動裝置上可以執行不同程式語言所撰寫出來的行動應用程式(mobile application)，例如：WAP 的行動應用程式、J2ME MIDP 的行動應用程式.....等等。也由於行動裝置的多樣性，造就了多種不同的裝置規格和計算能力，因此，在不同的行動裝置上，行動應用程式也會有不同的使用者介面呈現和執行表現。

由於以上的原因，對程式設計師所造成的困擾是：若想在不同行動執行環境上執行相同功能的行動應用程式的話，就必須以不同的程式語言撰寫多份不同版本的程式碼。因此，本論文提出以 XML 為基礎的行動應用程式開發工具(XML-based Mobile Application Development Kit)，試圖從不同的角度及觀點來解決問題，並減少撰寫和維護行動應用程式的成本。


# XML-based Mobile Application Development Kit

Student : Sheng-Po Shen

Advisor : Shyan-Ming Yuan

Department of Computer and Information Science  
National Chiao Tung University

## Abstract



As the mobile devices are developed vigorously in recent years, they have all kinds of variety. Therefore, many mobile applications are coded by many different programming languages. For example, WAP mobile applications and J2ME MIDP mobile applications can be run on various mobile devices. According to diversity of mobile devices, diverse specifications and computing capability of mobile devices are individually developed. Therefore, mobile applications have various presentations of user interfaces and computational performance on different mobile devices.

As a result of previous facts, programmers have following trouble. **If they want to run mobile applications with same function on various mobile execution environments, then they have to code various versions of program by using different programming languages.** Therefore, this thesis proposes **XML-based Mobile Application Development Kit** to reduce the cost of coding and maintaining mobile applications and solve the problem with different points of view.

---

## Acknowledgements

---

首先感謝我的指導教授 - 袁賢銘教授，謝謝老師在我完成論文的這段期間給予我許多的建議與指導，更要謝謝老師給我很大的發展空間，讓我能朝自己的理想邁進，真的非常感謝。

再來要謝謝旨峰和淑卿兩位賢伉儷，謝謝你們平時的照顧、幫忙和關心，讓煩悶無聊的生活充滿了溫暖，非常高興能認識你們。也謝謝研一時，一同參加比賽的學長同學，邱繼弘、林書慶、李宣鋒、洪崇凱，那是一段非常棒的回憶，很高興能和你們一同參與。也謝謝一同在分散室系統實驗室打拼的同學，林奕宇、劉昀昇、陸振恩、蘇科旭、洪崇凱、蔡明耀，平時受到你們很多的幫忙，謝謝你們，希望你們在未來的道路上能平順、發達。還有實驗室小組的成員，高子漢、高啟涵、吳仁凱、於之鈞，平時從大家的討論中獲益良多，謝謝你們，祝你們在求學的過程中有很傑出的表現。

最後，謝謝我的父母，謝謝你們一直支持著我，讓我能隨心所欲地朝自己的夢想前進，謝謝你們，謝謝。



---

# Table of Contents

---

<b>Abstract in Chinese</b>	.....	<b>I</b>
<b>Abstract in English</b>	.....	<b>II</b>
<b>Acknowledgements</b>	.....	<b>III</b>
<b>Table of Contents</b>	.....	<b>IV</b>
<b>List of Figures</b>	.....	<b>VI</b>
<b>List of Tables</b>	.....	<b>VII</b>
<b>Chapter 1 Introduction</b>	.....	<b>1</b>
1.1 前言	.....	1
1.2 動機	.....	1
1.3 論文架構	.....	2
<b>Chapter 2 Background and Related Works</b>	.....	<b>3</b>
2.1 背景介紹	.....	3
2.1.1 MExE (Mobile Execution Environment)	.....	3
2.1.2 WAP (Wireless Application Protocol)	.....	4
2.1.3 J2ME (Java 2 Platform, Micro Edition)	.....	5
2.1.4 XML (Extensible Markup Language)	.....	6
2.1.5 XSLT / XPath(Extensible Stylesheet Language for Transformation / XML Path Language)	.....	6
2.2 相關研究文獻	.....	7
2.2.1 XUL (XML User-interface Language)	.....	7
2.2.2 UIML (User Interface Markup Language)	.....	7
2.2.3 XForms	.....	8
2.2.4 JavaML	.....	9
2.2.5 Adaptive User Interface for Mobile Devices	.....	9
2.2.6 Developing Adaptable User Interfaces for Component-based Systems	.....	10
2.2.7 Mobile Adaptive Application for Ubiquitous Collaboration in Heterogeneous Environments	.....	10
2.2.8 可適性的行動應用程式開發平台	.....	11
<b>Chapter 3 XML-based Mobile Application Development Kit</b>	.....	<b>12</b>
3.1 設計理念	.....	12
3.2 Development Kit 裡頭的成員	.....	13

<b>Chapter 4 PGML / PUML</b>	<b>14</b>
4.1 為什麼要設計 PGML 和 PUML ?	14
4.1.1 為什麼要將行動應用程式拆解成兩部份 ?	14
4.1.2 為什麼要使用 XML 做為抽象層級的描述方法 ?	14
4.1.3 為什麼要設計 PGML 和 PUML ?	15
4.2 PGML – <u>P</u> ervasive <u>l</u> o <u>G</u> ic <u>M</u> arkup <u>L</u> anguage	16
4.2.1 PGML 簡介	16
4.2.2 PGML 的設計理念	16
4.2.3 PGML 所含有的 elements	18
4.2.4 PGML 標準函式庫	22
4.2.5 4.2.5 PGML 範例	23
4.3 PUML – <u>P</u> ervasive <u>U</u> ser-interface <u>M</u> arkup <u>L</u> anguage	27
4.3.1 PUML 簡介	27
4.3.2 PUML 的設計理念	27
4.3.3 PUML 所含有的 elements	28
4.3.4 PUML 範例	32
<b>Chapter 5 Comparison</b>	<b>37</b>
5.1 使用者介面描述語言的比較	37
5.2 邏輯運算描述語言的比較	40
<b>Chapter 6 Mobile Application Transformation</b>	<b>41</b>
6.1 準備做行動應用程式的轉換	41
6.2 WAP Transformation Pattern	41
6.3 J2ME Transformation Pattern	51
<b>Chapter 7 Mobile Application Example</b>	<b>58</b>
7.1 使用者介面導向的程式技巧	58
7.2 兩個 PUML / PGML 行動應用程式的例子	58
<b>Chapter 8 Conclusion and Future Work</b>	<b>67</b>
8.1 結論	67
8.1.1 達到行動應用程式僅需撰寫一次的目標	67
8.1.2 減少行動應用程式開發與維護的成本	67
8.1.3 PUML / PGML 提供了一個新的方向	68
8.1.4 轉換樣式(transformation pattern)是一個好的指導方針	68
8.2 未來發展	69
8.2.1 回顧過往	69
8.2.2 開創未來	69
<b>Reference</b>	
<b>Appendix</b>	

---

## List of Figures

---

圖 2-1	WAP Programming Model .....	4
圖 2-2	WAP Architecture .....	4
圖 2-3	J2ME Architecture .....	5
圖 2-4	XForms Model .....	8
圖 2-5	User interface data flow and application data flow result in the application.....	11
圖 3-1	XML-based Mobile Application Development Kit 架構圖 .....	12
圖 4-1	PGML 文件結構示意圖 .....	24
圖 4-2	PGML 簡單範例 .....	24
圖 4-3	PGML 簡單範例的相對應 J2ME 程式碼 .....	24
圖 4-4	PGML 複雜的範例 .....	25
圖 4-5	轉換之後所產生的 J2ME 程式碼(經過排版) .....	26
圖 4-6	轉換之後所產生的 WMLScript 程式碼(經過排版) .....	26
圖 4-7	PUML 文件結構示意圖 .....	33
圖 4-8	PUML 簡單範例 .....	33
圖 4-9	class : BMngr_LoginInput .....	34
圖 4-10	class : B_loginBoard .....	35
圖 4-11	class : B_mainBoard .....	35
圖 6-1	change( ) function 架構圖 .....	50
圖 6-2	PGMLchange.wmls .....	50
圖 6-3	BMngr_Interface interface .....	51
圖 6-4	changeBoard( ) method 的 J2ME MIDP 範例程式碼片段 .....	53
圖 6-5	getBoard( ) method 的 J2ME MIDP 範例程式碼片段 .....	54
圖 6-6	多選字串的範例程式碼 .....	57

---

## List of Tables

---

表 4-1	PGML 所有的 elements 說明	.....18
表 4-2	PGML 標準函式庫說明	.....22
表 4-3	PUML 所有 elements 的說明	.....30
表 5-1	各種使用者介面描述語言的比較	.....39
表 5-2	PUML 和 UIML 的比較	.....39
表 7-1	Example1 在執時期的畫面(WML/WMLScript 和 J2ME MIDP)	.....59
表 7-2	Example2 在執時期的畫面(WML/WMLScript 和 J2ME MIDP)	.....63





---

# Chapter 1 Introduction

---

## 1.1 前言

隨著行動裝置(mobile device)近年來的蓬勃發展，造就了現在行動裝置的多樣性。我們可以依據類型將這些行動裝置分成 PDA、Mobile Phone、Java Phone、Smart Phone.....等等的幾個大類，而這些行動裝置彼此間的規格與計算能力(computing capability)都不大相同，像是顯示螢幕的大小、顯示色彩為彩色或黑白、記憶體容量大小、CPU 運算能力，以及可用來執行行動應用程式(mobile application)的行動執行環境(Mobile Execution Environment, 簡稱 MExE).....等等。

也就是因為行動裝置彼此間的種種差異，使得在行動應用程式的撰寫上也呈現了多樣性。因此，我們可以在行動裝置上執行 WAP 的行動應用程式、J2ME 的行動應用程式或是其他程式語言所撰寫出來的行動應用程式。

本篇論文從撰寫這些行動應用程式的程式設計師的角度，來探討“行動應用程式多樣性”所衍生出來的問題，並針對這些問題提出一個解決方案—以 XML 為基礎的行動應用程式開發工具 (XML-based Mobile Application Development Kit)。

## 1.2 動機

造成行動應用程式多樣性的原因，除了是因為行動裝置彼此間的規格和能力不同之外，主要是因為行動執行環境的不同。在 MExE<sub>[45][46]</sub>的定義中，行動執行環境可以分成五類(5 MExE classmarks)：WAP、PJava、J2ME、CLI、VP，(將在 2.1 節中，對 MExE 做背景介紹)。在這五種分類中，分別可以執行以不同程式語言所撰寫出來的行動應用程式，像是 WML / WMLScript、J2ME MIDP.....等等。

由於以上的原因，對程式設計師所造成的困擾是：若想在不同行動執行環境上執行相同功能的行動應用程式的話，就必須以不同的程式語言撰寫多份不同版本的程式碼。這將對程式設計師造成不小困擾，因為他們必須額外付出維護行動應用程式的成本，和撰寫程式碼的成本。

因此，若行動應用程式只需要撰寫一次，便可以在各種行動執行環境上執行的話，便可以大大地減少必須付出的成本。基於這樣的動機，便在本篇論文中提出

一個解決方案 — 以 XML 為基礎的行動應用程式開發工具 (XML-based Mobile Application Development Kit)。在本篇論文中，您將可以看到如何運用所提出的方法來達到“行動應用程式僅需撰寫一次”，以及“減少開發、維護行動應用程式的成本”的目標。

### 1.3 論文架構

在做過簡單介紹之後，本篇論文的架構安排如下：在第二章中，做背景及相關研究文獻的介紹。從第三章開始，正式進入本篇論文的核心，也就是我們所提出的解決方案所使用的相關技術。在第三章裡，將概括性地介紹本篇論文所提出的 XML-based Mobile Application Development Kit，包括其設計理念和其組成的成員。第四章介紹兩種在抽象層級(abstract level)中，用來描述行動應用程式的方法 — PGML 和 PUML，並且在第五章中分別將 PUML、PGML 和目前提出的幾個描述語言做比較。利用第四章所提及的方法，描述一個完整的行動應用程式之後，必須將此行動應用程式轉換成各種可以在行動執行環境上執行的程式碼，該轉換的機制與方法將在第六章中介紹。在第七章中，提出兩個例子來做整個轉換流程的說明。最後，將在第八章中做整篇論文的結論，以及討論未來還可以進行的方向。參考資料以及附錄資料分別放在論文的最後篇幅，提供必要的資訊。



---

## Chapter 2 Background and Related Works

---

### 2.1 背景介紹

#### 2.1.1 MExE (Mobile Execution Environment)

MExE<sub>[45]</sub>是 3GPP<sub>[46]</sub>(3rd Generation Partnership Project)針對目前行動裝置(mobile device)上的執行環境，所提出的主張。在 MExE 裡頭，將行動裝置上的執行環境分成五種類型，如果該行動裝置的執行環境屬於五種類型之一，則該行動裝置也稱為 MExE device。

MExE 的目標是，希望能透過 MExE 的定義，來規範所有的行動裝置上的執行環境，使這些行動執行環境能夠標準化，使往後開發行動應用程式(mobile application)或是軟體服務(service)能夠更方便及快速。

MExE 根據每種行動裝置的執行環境和計算能力(computing capability)，將執行環境分成五種類型(classmark)：

##### 1. classmark 1 – WAP execution environment

擁有這類型執行環境的行動裝置，通常都含有一個輕量型的瀏覽器，可以觀看、執行由 WML / WMLScript<sub>[36]</sub>所撰寫出來的網頁(web pages)或是應用程式(web application)。這類型的執行環境必須遵守 WAP(Wireless Application Protocol<sub>[36]</sub>)的規範，應用在 GSM、GPRS，或是相容的無線網路環境上。

##### 2. classmark 2 – Personal Java execution environment

擁有這類型執行環境的行動裝置，通常擁有比一般行動裝置較高的計算能力(computing capability)和硬體能力(例如：較多的記憶體、較大的螢幕)。這類型的執行環境(也就是 Java Virtual Machine，簡稱 JVM)，可以執行由 Personal Java(PJava)<sub>[42]</sub>所撰寫出來的行動應用程式，支援較複雜的使用者介面及較多的 APIs，例如：Java AWT、Java Applet.....等等。

##### 3. classmark 3 – J2ME/CLDC execution environment

擁有這類型執行環境的行動裝置，其計算能力和硬體能力較受限制，且通常螢幕較小，無法顯示複雜的使用者介面。因此，在這類型的執行環境(也就是 Kilobyte Virtual Machine，簡稱 KVM)中，可以執行由 J2ME<sub>[42]</sub>所撰出來的行動應用程式。在 J2ME 中，特別額外設計了專屬的 APIs。

##### 4. classmark 4 – CLI compact execution environment

這類型的執行環境是中立性質的，利用 Microsoft 系列的程式語言(例如：

embedded VC++)所寫出來的應用程式，都可以在這類型的執行環境上執行。

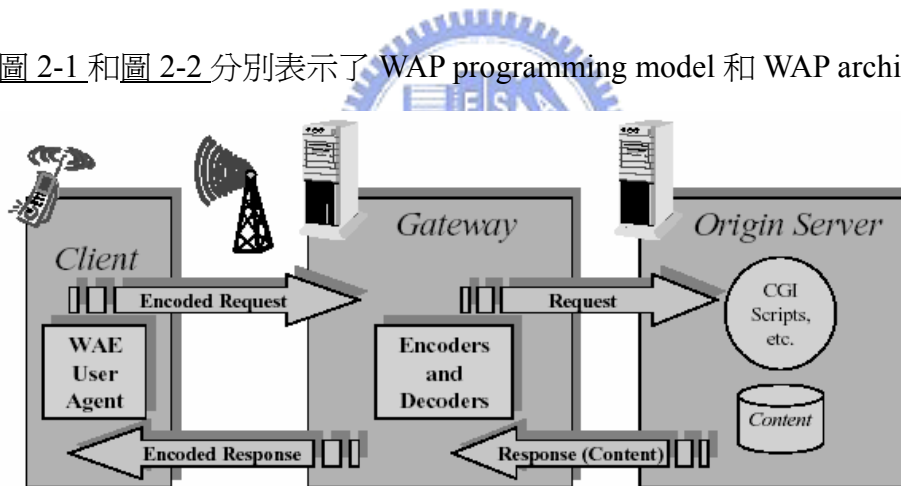
### 5. classmark 5 – Virtual PalmTop network execution environment

這類型的執行環境採用 client-server 架構，在行動裝置上只顯示使用者介面(client 端)，而行動裝置上所輸入的資料透過網路傳輸，送到遠端的 server 主機做邏輯運算的處理，處理完之後再將結果透過網路，送回到行動裝置上，再以適當的使用者介面顯示出來。另外，這類型的執行環境也可能跟之前的四種類型一起搭配使用。

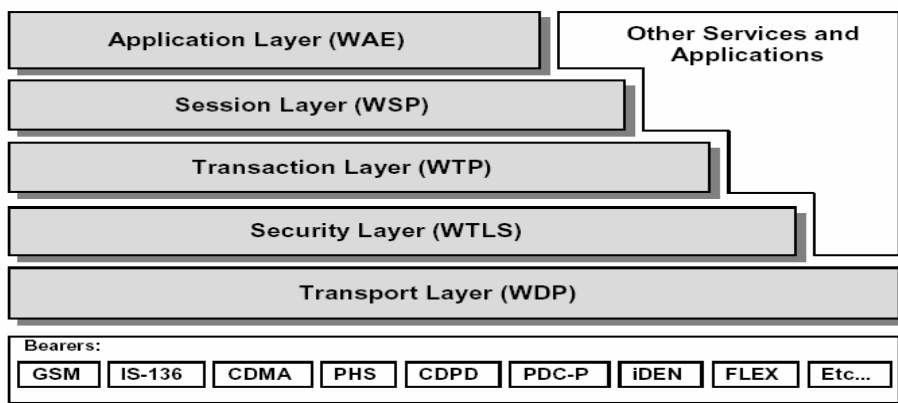
### 2.1.2 WAP (Wireless Application Protocol)

WAP<sub>[36]</sub>是一種應用在無線網路環境上的一種協定(protocol)，允許擁有無線裝置(wireless device)的使用者能夠在無線網路中，快速地存取資訊和服務。WAP 是由產業界所共同制定的一種公開的標準，使用 WML / WMLScript 來撰寫 WAP 應用程式，能夠應用在下列的無線網路環境中：CDMA、GSM、PHS、GPRS、CDPD、PDC、TDMA.....等等。

圖 2-1 和圖 2-2 分別表示了 WAP programming model 和 WAP architecture。



[圖 2-1 WAP Programming Model]



[圖 2-2 WAP Architecture]

### 2.1.3 J2ME (Java 2 Platform, Micro Edition)

J2ME<sub>[42]</sub>是一種專門爲了消費性嵌入式裝置所設計的 Java 平台，例如：行動電話(mobile phone)、個人數位助理(PDA, Personal Digital Assistant)、電視盒(set-top box)、smart card.....等等。在 J2ME platform 裡頭，包含了由多個專家組織(experts group)共同制定的標準 Java APIs。這些專家組織包括了硬體裝置的領導廠商，軟體廠商，以及軟體服務提供者(service provider)。

J2ME 的架構組織包含三個部份(參考圖 2-3): configurations、profiles、optional packages。這三個部份建立起一個完整的 Java 執行環境(Java runtime environment)。三個部份的內容分別是：

#### 1. configurations

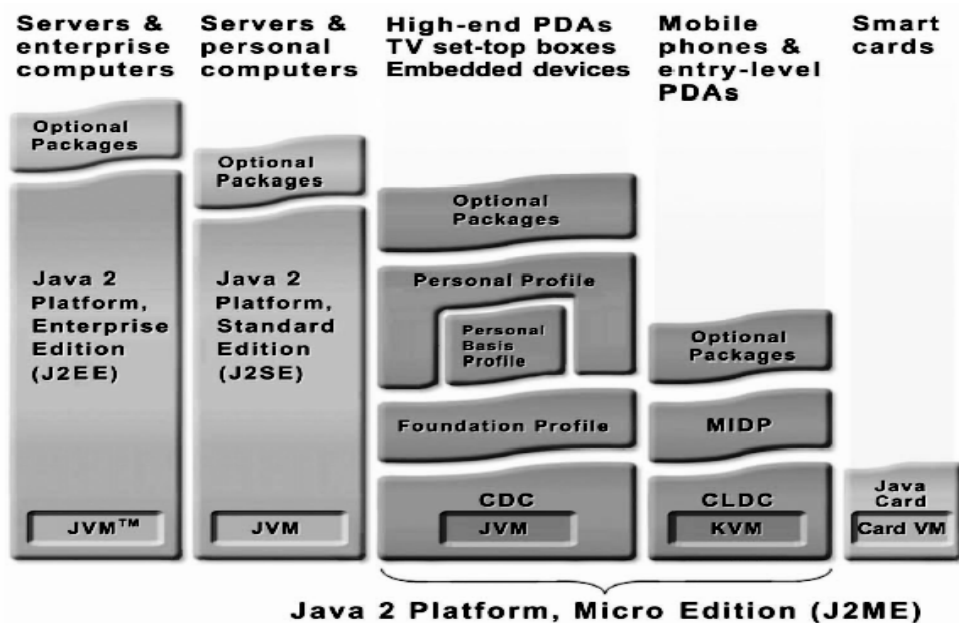
對於所有 J2ME 針對的裝置，提供基本的能力，包含了最小集合的 APIs。在 J2ME 裡頭，有兩種 configurations：分別是 CLDC(Connected Limited Device Configuration)，和 CDC(Connected Device Configuration)。

#### 2. profile

包含了更高階層的 APIs，針對不同裝置的能力，會有少許不同。可以跟 CDC 或 CLDC 建立起一個完整的執行環境。在 profile 的部份，定義了應用程式的生命週期(life cycle)、使用者介面、以及裝置的資料存取方式。

#### 3. optional package

是一種擴充用的 APIs。爲了滿足市場需求或是裝置本身特殊的能力，也可能是爲了擴充 J2ME 本身能力而設計的。例如：Bluetooth、Web Services.....等等。



[圖 2-3 J2ME Architecture]

### 2.1.4 XML (Extensible Markup Language)

XML<sub>[39]</sub>是一個開放的，以純文字為基礎的後設語言(meta-language)，以標籤(tag)的形式提供結構的，以及和語意相關的後設資料(metadata)。因為是以純文字為基礎的，所以可以很容易地被人類閱讀並了解，另外，它所提供的組織、描述和結構化資料的能力，使它能被機器(即應用程式)所解讀。

因為 XML 是一種後設語言，所以也能用來創造其他的語言，產生各式各樣的應用。現今有很多 XML 的應用，例如：XSLT、MathML、XPath、SOAP、WML.....等等。

### 2.1.5 XSLT / XPath

#### (Extensible Stylesheet Language for Transformation / XML Path Language)

XSLT<sub>[40][41]</sub>是 XSL(eXtensible Style Language)的其中一個分支。XSL 擁有強大的能力，能將一份 XML 文件以不同的方式呈現出來，或是將一份 XML 文件中的結構，重新組織成一份結構不同的 XML 文件。XSL 包含兩個主要的部份：

#### 1. a set of formatting objects

能將一份 XML 文件所描述的內容，以不同的呈現方式表現出來。這一個部份被獨立出來，但仍稱作 XSL 或 XSL-FO(XSL Formatting Objects)。

#### 2. transformation language

能將一份 XML 文件原本的內容結構，重新組織成內容結構完全不同的另一份 XML 文件。這一個部份也被獨立出來，稱之為 XSLT(eXtensible Stylesheet Language for Transformation)。

XSLT 具有以下的能力：

- XSLT 可以描述轉換之後的目的文件的格式樣板(template)，而樣本內容透過 XPath(eXtensible Path Language)<sub>[41]</sub>從原本的 XML 文件獲得。
- 可以從原本的 XML 文件中獲得 elements、attributes、text、processing instructions 和 comments。也可以在目的文件中穿插所需要的文字字串。
- 簡單地說，XSLT 文件就是轉換規則的集合(a set of transformation rules)，這些轉換規則可以重複使用，或是從外部檔案引入(import)。
- 在 XSLT 的轉換規則中，可以使用類似一般程式語言的條件判斷敘述，例如：if-else 敘述。
- 能夠使用變數(variable)或參數(parameter)，參數可由外部應用程式傳入。
- 能夠處理 XML 文件中，內容結構彼此的關係。
- 能夠處理簡單的字串、數字、布林值的運算。

## 2.2 相關研究文獻

### 2.2.1 XUL (XML User-interface Language)

XUL<sub>[11][34]</sub>是跨平台(cross-platform)的使用者介面描述語言，一開始是爲了使 Mozilla 這個瀏覽器，能夠更容易且快速地開發，而被創造出來的。是一種以 XML 爲基礎的使用者介面描述語言，所以具備了 XML 所有的特性與能力。因此，在 XUL 文件中也可以嵌入其他類型的 XML 語言，例如：XHTML、MathML.....<sub>[39]</sub>等等。

使用 XUL，可以去描述大部份圖形化的使用者介面元件，例如：

- 與輸入資料相關的 Text Field' checkbox' textbox.....等等。
- 工具列(toolbar)、按鈕(button)。
- 選單(menu)、選單列(menu bar)、跳出式選單(pop up menu)。
- 樹狀結構(tree)、表格(table)。
- 鍵盤快速鍵(keyboard shortcuts)。
- 其他.....。

### 2.2.2 UIML (User Interface Markup Language)

UIML<sub>[8][23][31][32]</sub>是一個以 XML 爲基礎的使用者介面描述語言。是一種位於抽象層級(abstract level)的使用者介面描述語言，可用來描述任何一種裝置平台上的使用者介面。因爲是以 XML 爲基礎的使用者介面描述語言，所以具備了 XML 所有的特性與能力。因此，在 UIML 文件中也可以嵌入其他類型的 XML 語言，例如：XHTML、MathML.....等等。

UIML 的設計目標(理念)：

- 所描述出來的使用者介面跟目的端的裝置平台無關(device independence)
- 減少裝置平台上的使用者介面開發時間。
- 將使用者介面部份和邏輯運算部份分開。
- 非專業的程式設計師也能容易地開發使用者介面。
- 能快速地開發出(描述出)使用者介面的雛型。
- 能有效率地透過網路下載使用者介面。
- 安全機制的加強。
- 具有擴充能力。

### 2.2.3 XForms

在一般的網頁應用程式中(web application)，均是透過“Form”這個使用者介面來跟使用者作互動。使用者在 Form 中輸入資料，然後資料透過網路被送 server 端作進一步的處理，然後將處理過後的資訊再透過網路，回傳到 client 端的瀏覽器上顯示出來。XForms<sub>[33]</sub>對這樣子的機制提出更進一步的想法。XForms 希望在網頁中，“Form”這樣子的一個使用者介面能：

- 提供更豐富的使用者介面元件。
- 需要使用者提供資料的部份，能夠跟其顯示方式區分開來。也就是說，“Form”這個使用者介面能夠以不同的呈現方式呈現，但都提供相同的使用者介面語意。
- 使用者所輸入的資料，能夠以 XML 的格式儲存並送出。
- client 端和 server 端彼此間的網路傳輸次數能夠減少。

XForms 是一種 XML 的應用，所以擁有 XML 所有的特性，相對的，也必須遵守 XML 所有的規範，因此，在 XForms 也可以嵌入其他類型的 XML 語言，例如：XHTML。XForms 所代表的是“下一世代網頁應用程式的使用者介面”。XForms 將這樣的使用者介面分成三個部份(參考圖 2-4)：

#### 1. XForms model

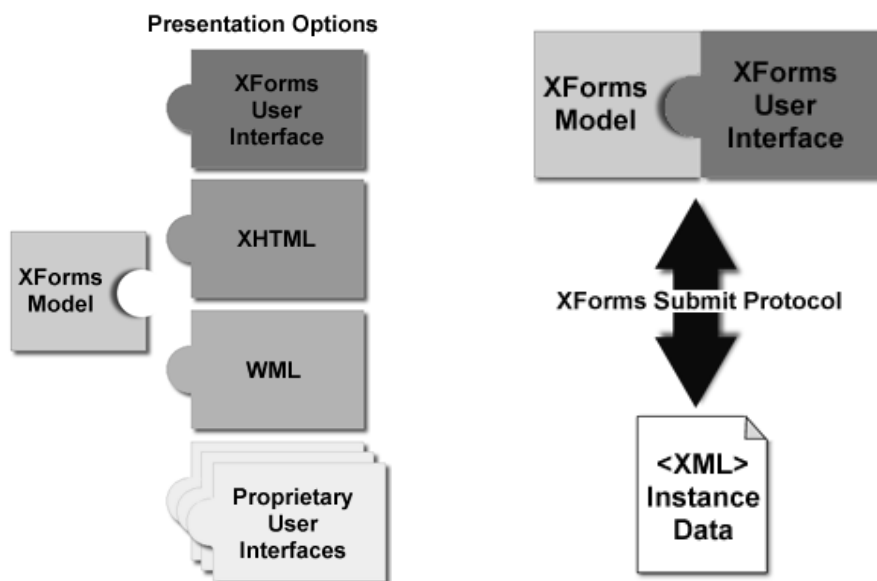
“Form”這個使用者介面的語意，包含需要使用者提供資料的部份。

#### 2. instance data

將使用者輸入的資料，儲存成 XML 格式。

#### 3. user interface

“Form”這個使用者介面呈現的方式。



[圖 2-4 XForms Model]



## 2.2.4 JavaML

JavaML<sub>[9][37]</sub>是以 XML 為基礎，以 XML 的方式來描述一份 Java 的原始程式碼。JavaML 是被設計用來幫助軟體工程的分析。

傳統程式設計師寫程式的方式，是以純文字的方式加上程式設計師自己的運算邏輯來完成的。以軟體工程的角度，當我們要去分析這一份純文字程式碼的結構時，需要經過一連串複雜的前置處理，包括語彙分析(lexical analysis)、語法分析(syntax analysis)、語意分析(semantic analysis)，也有可能需要產生 AST(abstract syntax tree)來做分析。在每一個前置階段也都可以搭配不同的分析工具，來完成所需要的分析工作，但花費的成本不低，結果也未必能滿意。

因此 JavaML 便產生，提供以不同的角度來做軟體工程的分析。首先針對 Java 語言，將 Java 的原始程式碼表示成 XML 文件，然後利用 XML 本身具有結構化的特性，以及現存很多 XML 相關的工具，來完成所需要做的軟體分析工作，這樣的一個過程，所需的成本較低，也往往能獲得較佳的結果。

## 2.2.5 Adaptive User Interface for Mobile Devices <sub>[11]</sub>

提出一個以 mobile agent 為基礎的系統架構，在這個架構上的所有裝置(包括桌上型電腦，行動裝置.....等等)，其使用者介面都可以靠遊走於這個架構上的 mobile agent 自動轉換產生。使用 XUL 來描述使用者介面，mobile agent 負責依據不同的情況，將 XUL 轉換成適當的使用者介面呈現：

- 當使用者所使用的裝置的運算能力夠強，且能執行 Java 程式時(例如：Java 手機或 PDA)，mobile agent 會遊走到該裝置上，並啟動使用 jXUL 這個中介軟體(middleware)，將 XUL 轉換成 Java Swing 的使用者介面，並執行起來。
- 當使用者所使用的裝置的運算能力夠強，但不能執行 Java 程式時(例如：沒有 JVM 的 PDA)，mobile agent 會遊走到該裝置上，並使用 XSLT 的轉換技術，將 XUL 轉換成 HTML 或 WML 的使用者介面，並執行起來。
- 當使用者所使用的裝置的運算能力不夠強時(當然也不能執行 Java 程式，例如：WAP 手機)，mobile agent 不會遊走到該裝置上，而是停留在整個架構中的某台主機上，當做 proxy 來被該裝置使用，該裝置透過網路將該 mobile agent 所轉換好的使用者介面(例如：WML)，下載到裝置本身並執行起來。

## 2.2.6 Developing Adaptable User Interfaces for Component-based Systems [3]

Component-based system 是由很多個 software component 所組成的，這篇論文是特別針對其中具有使用者介面的 software component 而提出的。當兩個以上具有使用者介面的 software component 需要組合起來，形成一個新的 software component 時，這時候便需要考慮這些來自不同 software component 的使用者介面，在組合後，新的 software component 的整體使用者介面是否適合或具有一致性？

這篇論文提出 aspect-oriented component engineering 的方式，來完成新的使用者介面適性化(adaptation)，也就是使來自不同 software component 的使用者介面能夠適當地組合起來，並且具有一致性。在這篇論文中也歸納出四種使用者介面適性化的類型(extension、composition、reconfiguration、adaptation to user, role and subtask)，幫助做更好的適性化的調整。

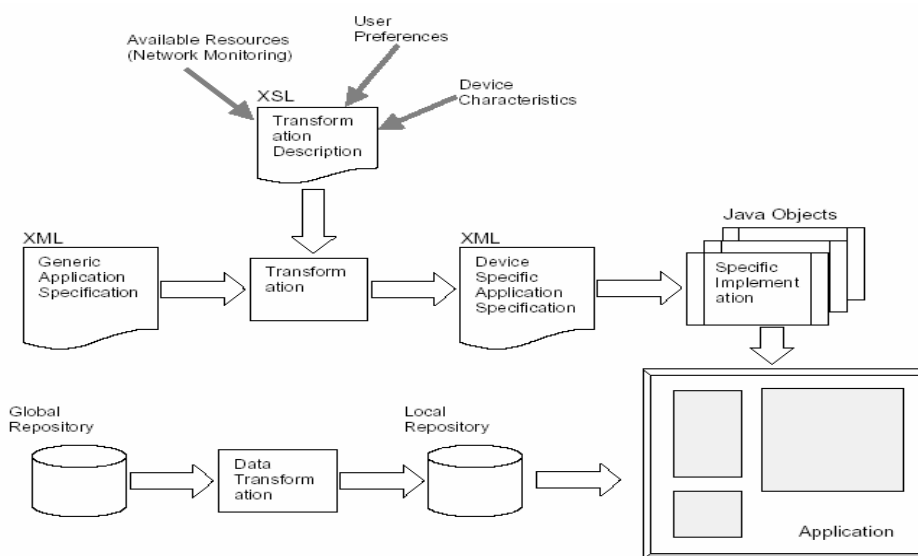
## 2.2.7 Mobile Adaptive Application for Ubiquitous Collaboration in Heterogeneous Environments [4]

在異質環境中要進行協同作業，例如甲員工使用桌上型電腦，乙員工使用 PDA，甲乙兩員工要使用同一種應用程式去共同完成一件工作。首先要克服的便是應用程式呈現的問題，也就是該應用程式的使用者介面在不同裝置上的呈現問題，因此應用程式的使用者介面需要被轉換。

這篇論文提出一個架構來解決這樣子的問題(參考圖 2-5)：

- 在 server 端以 XML 文件來描述應用程式的兩個部份：使用者介面和資料 (真實資料只有一份，且儲存在 server 端)。
- 在 server 端一開始利用 XML 描述的使用者介面和資料，是以一般性 (generic)的方式描述。
- 在轉換的過程中，使用者介面和資料這兩部份原本的 XML 描述，會再以 XML 的方式針對目的裝置做更準確的表達(device-specific)。
- 最後目的裝置會將之前所產生的 XML 文件轉換成真正可以執行的應用程式。

經過以上的處理，同樣的應用程式變可以產生不同的使用者介面，在異質環境中也可以完成協同作業了。



[圖 2-5 User interface data flow (top row) and application data flow (bottom row) result in the application.]

## 2.2.8 可適性的行動應用程式開發平台 [29]

### (An Adaptive Mobile Application Development Framework)

這篇論文提出一個可適性的行動應用程式開發平台，稱為 ART(Adaptive Remote Terminal)。該平台是一 client-server，並且分層的架構，讓程式設計人員可以快速的發展行動應用程式。基本的概念是將使用者介面與程式邏輯分開(在這篇論文中利用 XUL 來做使用者介面的描述，使用 Java 語言來描述程式邏輯)，利用桌上型電腦強大的運算能力與手持裝置的便利性，透過網路將兩者的優點結合並發揮出來。簡單來說就是延伸了手持裝置的計算能力。

ART 可以方便的新增通訊協定和使用者環境，除了達到可適性的目的外，也減少了程式開發與移植的功夫。

---

## Chapter 3 XML-based Mobile Application Development Kit

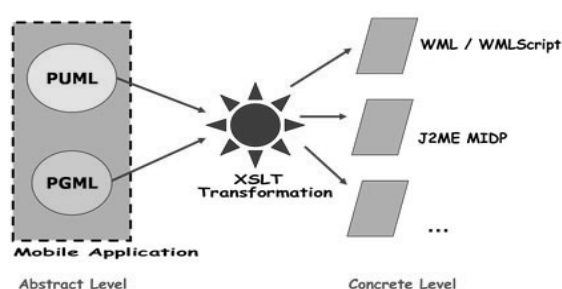
---

### 3.1 設計理念

基於“行動應用程式僅需撰寫一次，就可以在各種行動執行環境中執行”的想法，以及想要“減少行動應用程式開發與維護的成本”，所以有了本篇論文的產生，並且提出了一個解決方案，此解決方案便是 XML-based Mobile Application Development Kit。現在對此解決方案做概略性說明，詳細的討論將放在第四章和第六章。

由之前的介紹可以知道，行動執行環境(MExE)可以分成五大類，所以有多種程式語言可以用來撰寫行動應用程式，像是 WML / WMLScript、J2ME.....等等。因此，要達到“行動應用程式僅需撰寫一次”目標的話，就必須想辦法統合這些諸多可用來撰寫行動應用程式的程式語言。

本篇論文提出的方式是：在抽象層級(abstract level)中，額外設計用來描述行動應用程式的方法，而利用此方法所描述出來的行動應用程式的文件(或稱為程式碼)，可以利用轉換機制來將其轉換為各種可以在行動執行環境中執行的程式碼，例如：WML / WMLScript、J2ME MIDP.....等等。



[圖 3-1 XML-based Mobile Application Development Kit 架構圖]

從上圖的架構圖(圖 3-1)中可以看出，在描述行動應用程式方面，將行動應用程式分成兩部份：使用者介面部份(User-Interface)和邏輯運算部份(Logic Computing)。在抽象層級中，分別針對這兩個部份，提出了以 XML 為基礎的描述方式：使用者介面部份以 PUML(Pervasive User-interface Markup Language)來做描述，邏輯運算部份以 PGML(Pervasive loGic Markup Language)來做描述。這

樣做法的好處是，可以將行動應用程式的兩個部份交由兩組不同的程式設計師們負責，如此，可以加快開發速度，減少開發的時間成本。在維護行動應用程式方面，兩組程式設計師可以專門負責屬於自己的部份，減少維護行動應用程式的額外負擔，減少不必要的維護成本。

在轉換機制方面，採用 XSLT 的轉換機制，可以將在抽象層級中所描述的行動應用程式程式碼(或稱為文件)，轉換成各種實際可以在行動執行環境上執行的目的端程式碼。

在本論文中，主要的重點放在兩種目的端程式語言的轉換。分別是 MExE 五大分類中，WAP 類型下的 WML 和 WMLScript，以及 J2ME 分類下的 MIDP 程式碼的轉換。而在第七章中，也是以轉換成這兩種目的端程式語言為例子。

整體而言，XML-based Mobile Application Development Kit 的設計理念為：

1. 將行動應用程式分成使用者介面和邏輯運算，兩部份。
2. 利用以 XML 為基礎的描述方法，來描述行動應用程式。
3. 利用 XSLT 的轉換機制，達成行動應用程式僅需撰寫一次的目標。
4. 減少開發和維護行動應用程式的成本。

### 3.2 Development Kit 裡頭的成員

既然將所提出的解決方案稱之為 Development Kit，那麼就必須說明這個 Development Kit 中含有哪些工具。

在這個 Development Kit 中，主要包含下列三種工具：

1. 以 XML 為基礎的使用者介面描述語言 – PUML
2. 以 XML 為基礎的邏輯描述語言 – PGML
3. XSLT 的轉換機制 (本論文實作了將 PUML/PGML 轉換成爲 WML/WMLScript 和 J2ME MIDP 程式碼的 XSLT style sheet 文件)

在接下來的章節中，將會分別對這三種工具做詳細的說明。

---

## Chapter 4 PGML / PUML

---

### 4.1 爲什麼要設計 PGML 和 PUML ?

在這一節中，將更進一步說明 XML-based Mobile Application Development Kit 的前兩個設計理念：

- 1) 將行動應用程式分成使用者介面和邏輯運算，兩部份。
- 2) 利用以 XML 爲基礎的描述方法，來描述行動應用程式。

經過這一節的說明之後，將更清楚地可以知道爲什麼要設計 PGML 和 PUML 這兩個描述方法了。

#### 4.1.1 爲什麼要將行動應用程式拆解成兩部份？

因爲行動裝置本身必須具備很高的行動性，所以行動裝置的體積都很小，以方便攜帶。也因爲如此，行動裝置用來顯示資訊以及用來進行人機互動的螢幕，就相對地比桌上型電腦或筆記型電腦小了很多。

在各式各樣的行動裝置上，它們的螢幕大小以及顯示方式都不盡相同。但我們可以理解並想像到，相同功能的行動應用程式在不同的行動裝置上執行時（即在不同的行動執行環境上執行），它們的差別就僅在於顯示方式的不同，其運行於程式背後的邏輯都是相同的。

由於行動裝置的多樣性，也使得行動裝置的使用者介面也呈現多樣性。因此便想到，若將行動應用程式拆解成兩大部份的話 – 使用者介面的呈現和運行於背後的邏輯運算，必定可以大大地增加撰寫行動應用程式的彈性。

在使用者介面部份，可以依據行動執行環境的不同，做適合使用者介面的調整與改變。在邏輯運算方面，在不同的行動執行環境上執行時，主要的邏輯運算思維是相同的。

#### 4.1.2 爲什麼要使用 XML 做爲抽象層級的描述方法？

由於希望行動應用程式僅需撰寫一次，便可以在各種行動執行環境上執行。因此，需要將所有可以用來撰寫行動應用程式的程式語言做一個統合。

本篇論文所用的方式，是在所有可用的程式語言之上，加上一個抽象層級的描述方法。此描述方法必需足以代表所有在它下面的程式語言，也就是說，此描述方法具備了所有程式語言的共通性。如此，該描述方法才能透過轉換機制，將所描述的抽象行動應用程式，轉換為實際可以在行動執行環境上執行的，目的端行動應用程式。

由於這樣的想法，本篇論文選用了 XML 來做為此抽象層級描述方法的基本技術。充分利用 XML 本身的特性：跨平台、結構化、人類機器皆可閱讀……等等，加上現存很多 XML 相關的工具可以運用，例如本篇論文使用的 XSLT 轉換工具，它能將一份 XML 文件轉換成另一份結構不同的 XML 文件或純文字檔。在 XML 技術被普遍運用的今天，採用 XML 來做為抽象層級的描述方法，是最直接且可以直接受益於許多 XML 相關工具的做法了。

### 4.1.3 為什麼要設計 PGML 和 PUML ?

首先，針對使用者介面來討論，使用 XML 技術來描述使用者介面的相關文獻以及研究都已經進行多年了，但大都僅限於網頁內容呈現的調適性 (content adaptation)、應用程式執行環境的改變 (Browser、PC、PDA) 或是單純用來描述網頁使用者介面以及視窗程式的使用者介面。並沒有專門為行動裝置的環境所設計的描述方法，加上這些文獻研究所提出的觀點及描述方式，大多專為視窗環境或是高階瀏覽器所設計的。因此，本篇論文參考這些前輩門的研究，並加上自己的見解，設計了更加符合行動應用程式的使用者介面描述語言 – PUML。

至於邏輯運算方面，在所參考的文獻研究中，僅有 JavaML 是使用 XML 技術來描述邏輯運算的，但侷限於 Java (J2SE) 程式碼的描述。JavaML 並不符合需要，因為其所描述的是 Java 程式碼，其描述語言太過複雜，不適合使用來當做行動應用程式中，邏輯運算部份的描述方法。以 WMLScript 和 J2ME 為例，其程式語言的複雜度遠小於 J2SE 的 Java 程式碼，尤其是 WMLScript，其語法更是精簡，能力也較 Java 單薄許多。因此，使用 JavaML 來描述代表 J2ME 和 WMLScript 的邏輯運算的話，並不適合，有殺雞用牛刀之感。另外，JavaML 的本意是用來做軟體工程分析用的，出發點跟本篇論文所希望的“單純的邏輯描述”不同，自然 JavaML 也不符本篇論文所需了。最後，整理了所有可以用來撰寫行動應用程式的程式語言的特性，加上自己的見解，設計了符合行動應用程式的邏輯運算描述語言 – PGML。

## 4.2 PGML – Pervasive loGic Markup Language

### 4.2.1 PGML 簡介

PGML – Pervasive loGic Markup Language，是一個用來描述邏輯運算的語言，負責對行動應用程式中，邏輯運算的部份做描述。也由於 PGML 是一個以 XML 為基礎(XML-based)的描述語言，所以 PGML 必須遵守 XML 的所有規範，並且繼承了 XML 所擁有的特性。

PGML 具有簡單的物件觀念：在一份 PGML 文件中，你可以宣告多個用來做邏輯運算的 methods，而一份 PGML 文件你可以將之視為一個物件(object)。當你在使用某個 method 時，就像是在物件的世界中，呼叫物件中某個 method 一般。

如果你熟悉其他任何種類的程式語言的話，那麼，你也能夠很快地熟悉 PGML，不同的是，PGML 是 XML-based 的描述方法，是一種 XML 文件。當你使用 PGML 來做邏輯運算的描述時，同時也是在撰寫一份 XML 文件，所以也必須遵守 XML 的規範，除此之外，和平常你在使用的其他程式語言並沒有什麼不同了。

一份 PGML 文件經過 XSLT 的轉換機制之後，便可以轉換成各種可以在行動執行環境上執行的程式語言，例如：WMLScript、J2ME 的 MIDP.....等等。也就是說，你在 PGML 上所描述的邏輯運算將會對應到所轉換的目的程式語言上，當這被轉換而成的程式在行動執行環境上執行時，便會執行你在 PGML 文件中所描述的邏輯運算。

### 4.2.2 PGML 的設計理念

#### A. 引用一般程式語言的概念

因為 PGML 是用來做邏輯描述的語言，所以它最基本的能力要能夠做到像是：變數宣告、數學式的表示、function (method)的宣告、流程控制以及條件控制.....等等。因此，在參考了幾個常用的程式語言，以及行動執行環境上可以使用的程式語言(例如：C、Java、WMLScript.....等等)之後，試圖從這些程式語言中整理出一般常使用的概念，或是整理出，身為一種能夠描述邏輯運算的語言所必須具備的能力。

整理之後，PGML 所必須具備的能力如下：

- 可宣告、使用全域變數(global variable)或區域變數(local variable)。



- 有流程控制(flow control)，例如：for 迴圈的敘述。
- 可做條件判斷，例如：if-else 敘述。
- 可表達數學運算式。
- 可做邏輯比較，例如：大於、小於、等於。
- 可做邏輯運算，例如：and、or、not。
- 可宣告、呼叫 method。

### B. 精簡扼要的資料型態

因為 PGML 可以做變數宣告與使用、數學式的表示、邏輯運算、邏輯比較……等等。那麼，就一定會遇到資料型態的問題。這也是其他程式語言所必須處理的一個重要的議題。那麼，就必須去訂定屬於 PGML 的資料型態。

因為所撰寫的 PGML 文件必須經過 XSLT 的轉換機制，轉換成各種可以在行動執行環境上執行的程式碼。所以，在 PGML 上使用的資料型態，也必須能夠對應回各種可以在行動執行環境上執行的程式語言的資料型態。也就是說，各種可以在行動執行環境上執行的程式語言所“共同擁有”的資料型態，就是 PGML 的資料型態了。如此，PGML 的資料型態才能正確無誤地對應回各種程式語言的資料型態。

經過整理歸納之後，PGML 必須擁有三種資料型態：(1) 布林，boolean (2) 整數，int (3) 字串，String。

### C. 使用簡單的物件的概念

在 PGML 中，加入了簡單的物件概念。將一份 PGML 文件當做一個物件(object)來看待，在這個物件中，可以描述其 field members 和 methods，分別代表該物件的全域變數以及該物件所提供的 method。

加入物件概念的好處是，在撰寫 PGML 時可以以物件的角度來思考並撰寫 PGML 文件，會使得撰寫比較容易。並建議(並非一定)，以類似 Java Bean 的方式來寫 PGML 文件，會使得 PGML 文件容易撰寫、使用以及維護。

### D. 利用函式庫(library)來增加 PGML 的擴充性以及能力

另一方面，為了增加 PGML 的擴充性以及能力，我們制定了一個屬於 PGML 的標準函式庫(PGML Standard Library)。在這函式庫中，可以做字串的處理(例如：字串比較、取出子字串、獲得字串長度……等等)以及型別轉換。

這個標準函式庫具有可擴充性，以後可以陸續增加新的函式庫(library)或函

式(function)，也可以由別人制定屬於他們的函式庫。這裡的訣竅在於 XSLT 的轉換機制，只要將函式庫相對應的 XSLT style sheet 撰寫好，經過轉換機制轉換之後，就能夠轉換為各種行動執行環境上可執行的程式語言。

### 4.2.3 PGML 所含有的 elements

根據討論出來的 PGML 所必須具備的能力，以及為了方便使用 PGML 的程式設計師，能夠很快地熟悉 PGML 的語法，所以 PGML 的 elements 採用一般程式語言語法的命名規則。下面將所有的 elements 分類列出(表 4-1)：

[表 4-1 PGML 所有的 elements 說明]

Element	Description
<b><i>Core elements</i></b>	
<object>	每一份 PGML 文件的 root element (document element)，說明這是一份可視為一個物件(object)的 PGML 文件。<object>的 element content 即是所要描述的邏輯運算 (logic computing)。
<method>	PGML 最主要的 element，每個 <method>的 element content 代表的是想要執行的邏輯描述。另一方面來看，一個對外開放的<method>就相當於 PGML 對外的一個門戶，提供一個介面(interface)。
<variable>	在 PGML 中若要宣告變數或是參數，便是使用這個 element。
<operand>	在 PGML 中，這也是很重要的 element。要使用常數或使用變數都要靠這個 element 來描述。為邏輯中最基本的運算子，可代表常數或變數。
<b><i>Common elements in many places</i></b>	
<declaration>	用來宣告 global variable 和 local variable 的 element。<declaration>的 element content 即是<variable>的集合。可為 <object> 或 <method> 的 child element。

<test>	<test> element content 所代表的語意是：邏輯判斷。在邏輯上，<test>的 element content 的結果只有兩種：True 或 False。可為<for>、<if>或<while>的 child element。
<action>	<action>的 element content 所代表的語意是：啟動邏輯運算。可為<for>、<if>、<while>和<method>的 child element。也就是說，在流程控制時、經過條件判斷後以及呼叫某 method 時，所要做的動作，都是<action>的 element content 的內容。
<b><i>Flow control elements</i></b>	
<while>	代表做 flow control 的 element，語意相當於一般程式語言的 while-loop statement。要完成流程控制的工作必須搭配<test>和<action>等 child elements。
<for>	代表做 flow control 的 element，語意相當於一般程式語言的 for-loop statement。要完成流程控制的工作必須搭配<init>、<test>、<step>以及<action>等 child elements。
<init>	<for> element 的 child element。<init>的 element content 在語意上即是代表在做 for-loop 之前的初始值設定。
<step>	<for> element 的 child element。<step>的 element content 在語意上即是代表每做一次 for-loop 之後的下個階段的狀態。
<b><i>Condition control elements</i></b>	
<if>	代表做 condition control 的 element。語意相當於一般程式語言的 if-else statement。要完成條件控制的工作必須搭配<test>和<action>這兩個 child elements，如果需要的話也必須搭配<else>這個 child

	element。
<else>	在邏輯的語意上，<else>的 element content 即表示當不滿足之前的條件時，便執行<else> 的 element content 裡的邏輯運算描述。
<b>Mathematic operation elements</b>	
<add>	代表加法的 element。必須要有兩個 child elements，在語意上即代表將這兩個 child elements 相加。
<sub>	代表減法的 element。必須要有兩個 child elements，在語意上即代表將這兩個 child elements 相減。
<mult>	代表乘法的 element。必須要有兩個 child elements，在語意上即代表將這兩個 child elements 相乘。
<div>	代表除法的 element。必須要有兩個 child elements，在語意上即代表將這兩個 child elements 相除之後取所得的商數。
<mod>	代表 modulus 的 element。必須要有兩個 child elements，在語意上即代表將這兩個 child elements 相除之後取所得餘數。
<b>Logic comparison elements</b>	
<greater-than>	代表邏輯比較中，”大於”的 element。必須要有兩個 child elements，在語意上即是比較是否第一個 child element 大於第二個 child element。
<lesser-than>	代表邏輯比較中，”小於”的 element。必須要有兩個 child elements，在語意上即是比較是否第一個 child element 小於第二個 child element。
<equal-to>	代表邏輯比較中，”等於”的 element。必須要有兩個 child elements，在語意上即是比較是否第一個 child element 等於第二個 child

	element。
<not-equal>	代表邏輯比較中，”不等於”的 element。必須要有兩個 child elements，在語意上即是比較是否第一個 child element 不等於第二個 child element。
<greater-or-equal>	代表邏輯比較中，”大於或等於”的 element。必須要有兩個 child elements，在語意上即是比較是否第一個 child element 大於或等於第二個 child element。
<lesser-or-equal>	代表邏輯比較中，”小於或等於”的 element。必須要有兩個 child elements，在語意上即是比較是否第一個 child element 小於或等於第二個 child element。
<b><i>Logic operation elements</i></b>	
<and>	代表邏輯運算中的”and”的 element。必須要有兩個 child elements，在語意上即是滿足第一個 child element 且滿足第二個 child element。
<or>	代表邏輯運算中的”or”的 element。必須要有兩個 child elements，在語意上即是滿足第一個 child element 或滿足第二個 child element。
<not>	代表邏輯運算中的”not”的 element。只能有一個 child elements，在語意上即是將 child element 的邏輯顛倒。例如：將 True 變成 False，將 False 變成 True。
<b><i>Method-call and Library- function-call elements</i></b>	
<call-method>	用來表示要呼叫物件本身 method 的 element。
<lib>	用來表示要使用 library 裡頭的某個 function 的 element。
<b><i>Other elements</i></b>	
<in>	為<method>的 child element。用來描

	述一個 method 所需的參數。child element 為<variable>。
<return>	屬於<action>或<else>的 element content 中的 element，也相當於在<method>的 element content 之中。當 method 需要回傳東西時，便在<action>或<else>的 element content 做 return 的描述。
<assign>	<assign>的 element content 表示將某個數值或變數的值 assign 給某一變數。

#### 4.2.4 PGML 標準函式庫

在制定 PGML 的標準函式庫(PGML Standard Library)上，目前是針對三個資料型態(boolean、int、String)來制定，主要是提供型別轉換以及各型別專屬的函式(functions)。在標準函式庫中，我們將所有的函式分成三大類：(1) Integer (2) String (3) Boolean。

在以後，若有需要的話，可以新增標準函式庫裡的分類或是函式。每個人也可以制定屬於自己的函式庫(library)。因為，只要能將函式庫相對應的 XSLT style sheet 撰寫好，經過轉換機制轉換之後，就能夠轉換為各種可以在行動執行環境上執行的程式語言了。

表 4-2 為 PGML 的標準函式庫(PGML Standard Library)的整理：

[表 4-2 PGML 標準函式庫說明]

Function	Description
<b>Integer</b>	
toString	將整數值轉換成相對應的字串。例如：1234 → “1234”。
Abs	取得整數的絕對值。
Max	取得兩個整數中的最大值。
Min	取得兩個整數中的最小值。
maxInt	取得系統所支援的整數最大值。
minInt	取得系統所支援的整數最小值。
<b>String</b>	
parseInt	將字串轉換成相對應的整數值。例

	如：“1234” → 1234。
charAt	取得字串中某個位置的單一字元。(在 PGML 中，把單一字元當長度為 1 的字串處理)。
compare	按照一般程式語言比較字串的方式比較兩個字串。
Length	取得字串的長度。
substring	取得字串中的子字串。
Trim	刪除字串頭尾的空白字元。
concat	將兩個字串連接成一個字串。
<b>Booelan</b>	
toString	將布林值轉換成相對應的字串。例如：true → “true”。false → “false”。

## 4.2.5 PGML 範例

### A. PGML 文件結構

基本上，如果使用過其他任何一種程式語言的話，要撰寫 PGML 文件並不難，也可以說是很容易上手。因為在描述邏輯運算的概念上，PGML 和其他的程式語言並沒有什麼不同。唯一不同的是，因為 PGML 是以 XML 為基礎 (XML-based) 的描述方法，所以使用 PGML 來描述邏輯運算時，同時也是在撰寫 XML 文件，所以必須遵守 XML 的規範，必須使用標籤(tag)的方式來做邏輯運算的描述，除此之外，並沒有什麼不同了。

在觀念上，只需將一份 PGML 文件視為一個物件(object)，而這個物件可以提供 methods 出來使用，而所有的邏輯運算都在 method 裡頭做描述即可。也就是說，對於 PUML(Pervasive User-interface Markup Language，用來描述行動應用程式中，使用者介面的描述語言)文件來說，只需要知道這個物件(即 PGML 文件)所提供出來的 methods(或是稱為介面，interface)即可。

再來，簡單介紹一下 PGML 文件結構的骨架。

PGML 的 root element (document element)為<object>，<object>包含兩種 child element，分別為<declaration>和<method>。在<object>的 element content 中最多只能有一個<declaration>，最少一個<method>。而其他之前提到的 elements 則是放在<declaration>和<method>的 element content 中，做邏輯運算描述用的。每種 elements 的詳細用法及說明，請參考表 4-1 和本論文中的範例。

以下為 PGML 文件結構的示意圖(圖 4-1)：

```
<?xml version="1.0"?>
<object name="..." version="1.0"
  xmlns="http://dcs3.cis.nctu.edu.tw/Project/Pervasive/PGML/">
  <declaration>
    ...
  </declaration>
  <method name="..." return-type="...">
    ...
  </method>
  <method name="...">
    ...
  </method>
  ...
</object>
```

[圖 4-1 PGML 文件結構示意圖]

### B. 簡單的範例

用 PGML 寫個簡單的邏輯運算描述，如下(圖 4-2)：

```
<?xml version="1.0"?>
<object name="Increase5" version="1.0"
  xmlns="http://dcs3.cis.nctu.edu.tw/Project/Pervasive/PGML/">
  <declaration>
    <variable type="int" name="result" value="0"/>
  </declaration>
  <method name="incFive" visibility="public" return-type="int">
    <in>
      <variable type="int" name="num"/>
    </in>
    <action>
      <add result="result">
        <operand type="int" value="5"/>
        <operand select="num"/>
      </add>
    </action>
    <return select="result"/>
  </method>
</object>
```

[圖 4-2 PGML 簡單範例]

這個例子是描述一個名為 Increase5 的物件，提供了一個 method – incFive。呼叫這個 method 之後，便會將傳入值加五之後回傳。相對應的 J2ME 程式碼如下(圖 4-3)：

```
public class Increase5{
    int result = 0;

    public int IncFive(int num){
        result = 5 + num;
        return this.result;
    }
}
```

[圖 4-3 PGML 簡單範例的相對應 J2ME 程式碼]



### C. 複雜的範例

使用 PGML 做一個比較複雜的邏輯運算的描述。在這個例子(圖 4-4)中，PGML 的描述採取類似 Java Bean 的做法：在一開始，宣告了兩個 properties(兩個全域變數) – userName 和 userPassword。隨後描述了兩個 set methods 和兩個 get methods，分別是 setUserName( )、setUserPassword( )、getUserName( )以及 getUserPassword( )四個 methods。

```
<?xml version="1.0"?>
<object name="UserInfo" version="1.0"
  xmlns="http://dcs.w3.cis.nctu.edu.tw/Project/Pervasive/PGML/">
  <declaration>
    <variable type="String" name="userName" value="" />
    <variable type="String" name="userPassword" value="" />
  </declaration>
  <method name="setUserName" visibility="public" return-type="void">
    <in>
      <variable type="String" name="user_name" />
    </in>
    <action>
      <assign arg1="userName" arg2="user_name" />
    </action>
  </method>
  <method name="setUserPassword" visibility="public" return-type="void">
    <in>
      <variable type="String" name="passwd" />
    </in>
    <action>
      <assign arg1="userPassword" arg2="passwd" />
    </action>
  </method>
  <method name="getUserName" visibility="public" return-type="String">
    <action>
      <return select="userName" />
    </action>
  </method>
  <method name="getUserPassword" visibility="public" return-type="String">
    <action>
      <return select="userPassword" />
    </action>
  </method>
</object>
```

[圖 4-4 PGML 複雜的範例]

這個 PGML 範例經過 XSLT 的轉換機制轉換之後，所產生的 J2ME 程式碼 (圖 4-5) 以及 WMLScript 程式碼 (圖 4-6) 分別如下：

```

/*J2ME "import" declarations*/
public class UserInfo {
    /*global variables' declaration*/
    private String userName="";
    private String userPassword="";

    /*constructor*/
    public UserInfo(){ }

    /*all methods are here...*/
    public void setUserName( String user_name ){
        /*local variables' declaration*/

        this.userName=user_name;
    }

    public void setUserPassword( String passwd ){
        /*local variables' declaration*/

        this.userPassword=passwd;
    }

    public String getUserName( ){
        /*local variables' declaration*/

        return this.userName;
    }

    public String getUserPassword( ){
        /*local variables' declaration*/

        return this.userPassword;
    }
}

```

[圖 4-5 轉換之後所產生的 J2ME 程式碼(經過排版)]

```

/*some pragma are declared here...*/
/*WMLScript interface to set declared global variables*/
extern function globalVarDeclaration() {
    /*setting all global variables in this logic object*/
    WMLBrowser.setVar("UserInfo_userName", "");
    WMLBrowser.setVar("UserInfo_userPassword", "");
}

/*all methods are here...*/
extern function setUserName (user_name){
    WMLBrowser.setVar("UserInfo_userName", user_name );    }

extern function setUserPassword (passwd){
    WMLBrowser.setVar("UserInfo_userPassword", passwd );    }

extern function getUserName (){
    return WMLBrowser.getVar("UserInfo_userName");    }

extern function getUserPassword (){
    return WMLBrowser.getVar("UserInfo_userPassword");    }

/*WMLScript interface to change value of UI component on some WML card*/
extern function change(component, newValue, callMethod){
    if(callMethod == ""){
        WMLBrowser.setVar(component, newValue);    }
    else{
        if(callMethod == "getUserName"){
            WMLBrowser.setVar(component, getUserName());    }

        if(callMethod == "getUserPassword"){
            WMLBrowser.setVar(component, getUserPassword());    }
    }
}

```

[圖 4-6 轉換之後所產生的 WMLScript 程式碼(經過排版)]

## 4.3 PUML – Pervasive User-interface Markup Language

### 4.3.1 PUML 簡介

PUML 是一種描述使用者介面(User Interface, 簡稱 UI)的描述語言。也就是將行動應用程式拆成使用者介面以及邏輯運算兩部份之後, PUML 就是專門負責做使用者介面的描述。

PUML 是一種以 XML 為基礎(XML-based)的使用者介面的描述語言。所以 PUML 必須遵守 XML 的所有規範, 並且繼承了 XML 所擁有的特性。PUML 是從 XUL、UIML 以及 WML 之中獲得靈感, 但為了使 PUML 更適合運用在行動裝置上, 並且成爲一個足以位於抽象層級(abstract level)、代表著在所有行動執行環境中, 所可以呈現的使用者介面。所以, PUML 除了吸收三家之長外, 也擁有自己的設計理念, 這將在後面做介紹。

一份 PUML 文件經過 XSLT 的轉換機制之後, 便可以轉換成各種在行動執行環境中可以執行的程式語言, 例如: WML、J2ME 的 MIDP.....等等。也就是說, 在 PUML 上所描述的使用者介面的呈現方式將會對應到所轉換的目的程式語言上, 當這被轉換而成的程式, 在行動執行環境上執行時, 便會呈現在 PUML 文件中所希望的使用者介面的呈現方式。

### 4.3.2 PUML 的設計理念

#### A. 從 XUL、UIML 以及 WML 中獲得靈感

在設計 PUML 之前, 參考了幾個也是以 XML 為基礎(XML-based)的使用者介面描述語言, 包括: XUL、UIML 以及 WML。在整理、比較過這三種描述方法之後, 歸納出以下特性, 而這些特性是 PUML 必須具備的:

- element 數目必須精簡。
- element 名稱, 跟相對應的各種行動執行環境中的目標程式語言的關係, 必須盡量減低。
- 必須要有變數(variable)的觀念, 以便記錄各個使用者介面中, 每個元件的狀態(state), 也可當作邏輯運算的參數。
- 必須要有事件觸發(event-driven)的概念, 以便跟邏輯運算部份做連繫。
- 因爲是行動裝置的使用者介面的描述語言, 所以使用者介面中的每個元件之間, 彼此的關係必須簡單。
- 採用類似 WML 中 card 的概念(實際上: 在 PUML 中使用 board 的觀念來表達, 之後會再說明), 以符合行動裝置螢幕較小的特性。

## B. 平坦式的使用者介面結構設計

在 PUML 的設計理念中，另一個重點便是，在使用者介面的描述上所做的特別要求：**平坦式的使用者介面結構**。

一個完整的使用者介面是由多個元件(component)所構成的，這些元件即是使用者介面元件(UI component)，而統一放置這些元件的場所，稱為使用者介面容器(UI container)。也就是說，要完成使用者介面的描述，便是在使用者介面容器(UI container)中，有次序地排放使用者介面元件。在比較複雜的程式語言中(例如：C++、Java.....等等)，或是在一些以 XML 為基礎的描述語言中(例如：XUL、UIML.....等等)，某些元件也可以當作容器使用，也就是會出現**巢狀的使用者介面結構**。例如在一般視窗程式中的多層選單，便是一種巢狀的使用者介面結構。

而 PUML 所採用的**平坦式的使用者介面結構**，便是不允許**巢狀的使用者介面結構**的產生。也就是說，任何一種使用者介面元件都不能當做使用者介面容器使用，所有的使用者介面元件都以相等地位，依序地排放在使用者介面容器中，不會有巢狀的使用者介面結構產生。

使用平坦式的使用者介面結構的好處是：

- PUML 的程式設計師能夠簡單且輕易地描述出想要的使用者介面，而不需考慮到複雜的使用者介面的設計，可以減少程式發生錯誤以及除錯的時間。
- 因為 PUML 文件最後會透過 XSLT 的轉換機制，轉換成各種可以在行動執行環境上執行的程式語言。而這些程式語言所能描寫出來的使用者介面的複雜度均不相同，所以採用平坦式使用者介面結構的 PUML，便可以輕易地在轉換過程中，組合出複雜度不同的，目的端的使用者介面描述(target UI description)。若是 PUML 允許巢狀的使用者介面結構的話，所描述出來的使用者介面將會很複雜，那麼，若要將複雜的使用者介面轉換成不同複雜度的使用者介面的話，在轉換完成之後，出現跟原先所希望的使用者介面語意(UI semantic)不合的情況將會大大地增加。所以，採用**平坦式的使用者介面結構**是有其考量以及好處的。

### 4.3.3 PUML 所含有的 elements

由於要滿足 PUML 所必須具備的其中一個特性：*element 名稱，跟相對應的各種行動執行環境中的目標程式語言的關係，必須盡量減低*。所以要從另一個角度來思考 element 的名稱，這個名稱要夠抽象，必須不同於各個具體的目標程式語言中，所使用的使用者介面元件的名稱，但也要能充分地表達出

所要表達的使用者介面元件。而這些 PUML 的 element 必須也要能夠對應回各種目標程式語言的使用者介面元件。

所以，設計 PUML 的 element 的方法是：(1)整理出各種可以在行動執行環境中執行的程式語言(WAP、Personal Java、J2ME、CLI)的共同使用者介面元件 (2)將這些共同的使用者介面元件加以抽象化(abstraction)，成為 PUML 的 elements。

首先，我們粹取出的共同使用者介面元件有：UI container、image、label、text field、choice group(or list)以及跟事件觸發相關的 button 或 command。再來便是將這些共同的使用者介面元件加以抽象化，我們將這個抽象化的過程對應到現實生活中的情況：

在很多家庭裡，冰箱門上總是貼了許許多多的便利貼(例如：3M Post-it)，或是用磁鐵吸附了各式各樣的照片或是便條紙。也就是說，冰箱門相當於家裡的小小公佈欄，公佈欄上說明了要做的事或是提醒某件事記得去做……等等，造成了家人之間彼此間的互動。若是把使用者介面的描述，引進公佈欄的概念的話，將會是什麼情形呢？公佈欄的那塊板子便相當於 UI container，我們可以在上面黏貼標籤(label)，便條紙(text note)或照片(picture)，也可以釘上勾選單(check list paper)。這些黏貼上去的便條紙、照片、勾選單所形成的，便相當於使用者介面元件的排版。如此，使用者介面元件的抽象化便完成了。

在 PUML 中，由以上抽象化過程而來的 elements 有：<board>、<picture>、<label>、<textnote>、<listpaper>以及<item>。這些 element 代表的義意簡單說明如下：

- <board> → 相對應於公佈欄，即 UI container
- <picture> → 相對應於公佈欄上的照片，即圖檔，image
- <label> → 相對應於公佈欄上的標籤，即文字(唯讀)
- <textnote> → 相對應於公佈欄上的便條紙，即 text filed (可編輯)
- <listpaper> → 相對應於公佈欄上的勾選單，即 choice group
- <item> → 相對應於勾選單上的選項，即 choice group 的選項

除了以上基本的 elements 之外，在 PUML 中也有用來描述跟使用者介面相關額外訊息的 elements，例如：<user-interface>、<layout>，以及在 PUML 中，所會使用到和邏輯運算相關的 elements，例如：<logic-objects>、<object>。

以及跟事件觸發相關的 elements，例如：<action>、<use-object>、<param>、<change>、<nextboard>。詳細的 elements 說明請參考表 4-3。

[表 4-3 PUML 所有 elements 的說明]

Element	Description
<b><i>Core UI Related Elements</i></b>	
<user-interface>	每一份 PUML 文件的 root element (document element)，說明這是一份描述使用者介面(user interface)的 PUML 文件。
<layout>	PUML document 的 metadata，說明在同一份 PUML document 中的所有 <board>s 要如何做排版。用於以後做適性化(adaptation)之用，現階段並沒有任何作用。
<board>	PUML 中核心的 element，代表最基本的 UI container 單位。
<picture>	用來表示要在使用者介面上，顯示圖片的 element。相當於 window 視窗上的 Image。
<label>	用來表示要在使用者介面上，顯示文字的 element。相當於 window 視窗上的 Label。
<textnote>	用來表示要在使用者介面上，顯示單行文字輸入欄的 element。相當於 window 視窗上的 TextField。
<listpaper>	用來表示要在使用者介面上，顯示勾選列表的 element。相當於 window 視窗上的 Choice Group 或 Choice List。
<item>	用來表示在使用者介面上，顯示勾選列表中選項的 element。
<b><i>Logic Object Related Elements</i></b>	
<logic-objects>	<logic-objects>的 element content，是用來宣告在 PUML 中所要使用的邏輯物件(logic object)。依照宣告方式不同，邏輯物件有全域性和區域性兩種(global logic object 和 local logic object)。child element 只有<object>一

	種。
<object>	用來指定所要使用的 logic object(即是由 PGML 所描述出來的邏輯物件)。parent element 為<logic-objects>。
<b><i>Event Driven Related Elements</i></b>	
<action>	<p>有三種類型的 child elements : &lt;use-object&gt;、&lt;change&gt;、&lt;nextboard&gt;。在&lt;action&gt;的 element content 中，三種類型的 child element 可依照所需要的邏輯運算處理，或是使用者介面瀏覽的進行方式來做搭配使用。</p> <p>&lt;action&gt;這個 element，是將 PUMML 和 PGML 連結起來的橋樑。將所有的邏輯運算，透過&lt;use-object&gt;和&lt;param&gt;，交給 PGML 中的 method 來做。在使用者介面上的顯示方式，多以 Button 或 Command 的形態出現。</p>
<use-object>	<p>用來表示所要使用的 logic object，並可指定要用該 logic object 的哪個 method。</p> <p>可有兩種 parent element 的可能：&lt;action&gt;或&lt;change&gt;。</p>
<param>	用來表示邏輯運算時，所呼叫的 method 所需要的參數。邏輯運算的實體定義在相對應的 PGML 文件中。
<change>	用來更改使用者介面元件 (UI component)狀態的 element。例如：更改呈現在螢幕上的字串值
<nextboard>	用來表示在 PUMML 中，boards 彼此之間的連結關係。以這個 element 來達成使用者介面切換的功能。

### 4.3.4 PUML 範例

#### A. PUML 文件結構

其實撰寫 PUML 文件相當容易。PUML 的程式設計師只要有公佈欄的抽象概念，然後只要想著要在公佈欄上張貼什麼訊息即可。也就是說，若要在公佈欄上貼圖片的話，就只要在<board>的 element content 中加上<picture>即可，想要在公佈欄上貼上標籤的話，就只要在<board>的 element content 中加上<label>即可，其他以此類推。若想使用邏輯運算來完成某些特定的事情的話，就只要在<board>的 element content 中加上<logic-objects>，然後在<logic-objects>的 element content 中，使用<object>來指定要使用哪個邏輯物件(logic object，也就是 PGML 文件)即可。若想要觸發事件以進行邏輯運算的話，在<board>的 element content 中加上<action>，然後在<action>的 element content 中，根據你想執行的順序，分別加上<use-object>、<change>、<nextboard>即可，相當直覺。

也就是說，在 PUML 中，使用者介面容器(UI container)就是<board> element，在<board>的 element content 中可以放入<label>、<textnote>、<logic-objects>、<action>.....等等的 elements。PUML 的程式設計師在設計使用者介面時，便以 board 為基本單位，來設計行動應用程式中所有可能出現的使用者介面。

再來，簡單介紹一下 PUML 文件結構的骨架。

PUML 的 root element(document element)為<user-interface>，<user-interface>包含三種 child element，分別為<layout>、<logic-objects>和<board>。在<user-interface>的 element content 中最多只能有一個<layout>，而<logic-objects>最多也只能有一個，另外<board>則最少要有一個。<board>包含了六種 child element，分別為<picture>、<label>、<textnote>、<listpaper>，以及<logic-objects>和<action>，而在<board>中最多只能有一個<logic-objects>。<listpaper>的 child element 為<item>，<logic-objects>的 child element 為<object>，<action>的 child element 為<use-object>、<change>或<nextboard>。每種 elements 的詳細用法及說明，請參考表 4-3 和本論文中的範例。



以下為 PUML 文件結構的示意圖(圖 4-7)：

```
<?xml version="1.0"?>
<user-interface name="..." version="1.2"
  xmlns="http://dcs3.cis.nctu.edu.tw/Project/Pervasive/PUML/">
  <layout>...</layout>
  <logic-objects>
    <object name="..." source="...">
      ...
    </object>
  </logic-objects>

  <board name="..." title="..." seqNO="...">
    <picture/>
    <label/>
    <textnote/>
    <listpaper>
      <item/>
      ...
    </listpaper>
    <action name="..." showText="...">
      <use-object name="..." method="...">
        <param/>
        ..
      </use-object>
      <change container="..." component="..." update="...">
        ...
      </change>
      <nextboard goto="...">
    </action>
  </board>
  <board>...</board>
  ...
</user-interface>
```

[圖 4-7 PUML 文件結構示意圖]

## B. 簡單的範例

用 PUML 寫個簡單的使用者介面描述，如下(圖 4-8)：

```
<user-interface name="LoginInput" version="1.2"
  xmlns="http://dcs3.cis.nctu.edu.tw/Project/Pervasive/PUML/">
  <logic-objects>
    <object name="login" source="Login.pgml"/>
  </logic-objects>

  <board name="loginBoard" title="login page" seqNO="0">
    <label name="showMsg1" showText="input your name: "/>
    <textnote type="text" name="UserName"/>
    <label name="showMsg2" showText="select your sex: "/>
    <listpaper iname="isex" mode="single">
      <item showText="Boy"/>
      <item showText="Girl"/>
    </listpaper>

    <action name="enter" showText="ENTER">
      <use-object name="login" method="verify">
        <param select="UserName"/>
        <param select="isex"/>
        <param type="String" value="Location-NCTU"/>
      </use-object>
      <nextboard goto="mainBoard"/>
    </action>
  </board>

  <board name="mainBoard" title="main page">
    <!--content-->
  </board>
</user-interface>
```

[圖 4-8 PUML 簡單範例]

這個例子是描述一個登入的使用者介面。在這個所描述的使用者介面中，會要求使用者輸入姓名以及性別，然後在事件觸發後(按下 button 或 command 之後)，會將使用者在使用者介面上輸入的資料，送給後端的邏輯運算部份

做運算，運算完之後會將畫面切換到另一個描述的使用者介面。

相對應的 J2ME 程式碼如下(圖 4-9、圖 4-10、圖 4-11)：

(共產生三個 class：BMngr\_LoginInput、B\_loginBoard、B\_mainBoard)

```
import javax.microedition.midlet.*; import javax.microedition.lcdui.*;

interface BMngr_Interface {
    public void changeBoard(String boardName);
    public Displayable getBoard(String boardName);
}

public class BMngr_LoginInput extends MIDlet implements BMngr_Interface{
    public static BMngr_LoginInput instance;          private Display display;
    /*global logic objects*/
    public Login lgObj_login = new Login();
    /*all displayables corresponding to boards' declaration*/
    private B_loginBoard loginBoard = new B_loginBoard();  private B_mainBoard mainBoard = new B_mainBoard();
    /*Constructor*/
    public BMngr_LoginInput(){ instance = this;  }
    /*Main method*/
    public void startApp(){ display = Display.getDisplay(this); display.setCurrent(loginBoard);  }
    /*Handle pausing the MIDlet*/
    public void pauseApp(){  }
    /*Handle destroying the MIDlet*/
    public void destroyApp(boolean unconditional){  }
    /*Quit the MIDlet*/
    public static void quitApp(){ instance.destroyApp(true); instance.notifyDestroyed(); instance = null;  }
    /*change UI presentation*/
    public void changeBoard(String boardName){
        if(boardName.compareTo("loginBoard") == 0){ display.setCurrent(loginBoard);  }
        if(boardName.compareTo("mainBoard") == 0){ display.setCurrent(mainBoard);  }
    }
    /*get particular UI container*/
    public Displayable getBoard(String boardName){
        Displayable displayable = null;
        if(boardName.compareTo("loginBoard") == 0){ displayable = loginBoard;  }
        if(boardName.compareTo("mainBoard") == 0){ displayable = mainBoard;  }
        return displayable;
    }
}
```

[圖 4-9 class：BMngr\_LoginInput]

```

class B_loginBoard extends Form implements CommandListener {
    /*local logic objects*/
    /*UI components*/
    public StringItem showMsg1; public TextField UserName; public StringItem showMsg2; public ChoiceGroup isex;
    /*commands*/
    private Command enter = new Command("ENTER", Command.SCREEN, 1);
    /*Constructor*/
    public B_loginBoard() {
        /* initialize UI component */
        super("login page"); showMsg1 = new StringItem("input your name: ", "");
        UserName = new TextField("", "", 50, TextField.ANY);
        showMsg2 = new StringItem("select your sex: ", "");
        isex = new ChoiceGroup("", ChoiceGroup.EXCLUSIVE);
        isex.append("Boy", null); isex.append("Girl", null);

        /* set up listeners */
        this.setCommandListener(this);

        /* append UI component */
        this.append(showMsg1); this.append(UserName); this.append(showMsg2); this.append(isex);

        /* add commands */
        this.addCommand(enter);
    }

    /*Handle command events*/
    public void commandAction(Command command, Displayable displayable){
        String selectedStr = null; String oldPath = null; String newPath = null; String align = null;

        if(command == enter){
            BMngr_LoginInput.instance.lgObj_login.verify(UserName.getString(), isex.getSelectedIndex()+1, "Location-NCTU");
            BMngr_LoginInput.instance.changeBoard("mainBoard");
        }
    }
}

```

[圖 4-10 class : B\_loginBoard]



```

class B_mainBoard extends Form implements CommandListener {
    /*local logic objects*/
    /*UI components*/
    /*commands*/
    /*Constructor*/
    public B_mainBoard() {
        /* initialize UI component */
        super("main page");

        /* set up listeners */
        this.setCommandListener(this);

        /* append UI component */
        /* add commands */
    }

    /*Handle command events*/
    public void commandAction(Command command, Displayable displayable){
        String selectedStr = null; String oldPath = null; String newPath = null; String align = null;
    }
}

```

[圖 4-11 class : B\_mainBoard]

### C. 複雜的範例

使用 PUMML 做一個比較複雜的使用者介面描述。在這個例子中，使用四個 `<board>` 來描述四個使用者介面，而整個例子主要的用意是，展示所有在 PUMML 中可以描述出來的使用者介面元件。

第一個 `<board>` 主要展示 PUMML 可以描述圖片、標籤以及輸入欄位等使用者介面元件，並且可以將使用者輸入的資料送到後端邏輯部份去處理，處理完之後，使用者將可以在第二個 `<board>` 所描述的使用者介面中看到所輸入的資料。

第二個 `<board>` 主要展示在 PUMML 所描述出來的使用者介面中，可以隨時反應出使用者介面元件的變化，例如：將使用者在第一個使用者介面中所輸入的資料，透過更新使用者介面元件顯示的方式，將所輸入的資料顯示在第二個所描述的使用者介面上。另外，在這個使用者介面中，可以透過事件驅動的方式(按下 `button` 或 `command`)，選擇要進入第三個或第四個使用者介面。

第三個 `<board>` 主要展示 PUMML 也可以描述單選模式的選單(`single-selection choice group` 或 `single-selection choice list`)，並可以將使用者的選擇結果顯示在同一個使用者介面上。

第四個 `<board>` 主要展示 PUMML 也可以描述多選模式的選單(`multi-selection choice group` 或 `multi-selection choice list`)，並可以將使用者的選擇結果顯示在同一個使用者介面上。

因為篇幅排版的關係，這個範例的 PUMML 原始文件檔，以及經過 XSLT 轉換機制，所轉換出來的 J2ME 程式碼和 WML 程式碼，均放在 appendix 中。另外，本範例也會在 7.2 節中的 Example1 出現，將提供本範例所描述出來的使用者介面在執行時期的面貌。

---

## Chapter 5 Comparison

---

### 5.1 使用者介面描述語言的比較

在這一節中，把 PUML 和幾個同樣也是以 XML 為基礎，且用來描述使用者介面的描述語言拿來做比較。將分別列出每個描述語言的優缺點，最後再以兩個表格做結論。

#### A. *XUL (XML User-interface Language)*

- **優點：**

對於所有會出現在視窗上的使用者介面元件，定義的非常完整與清楚，而且每個使用者介面元件，在 XUL 中都有相對應的 `element` 表示。對於視窗中使用者介面元件的排版(layout)，也有相對應的 `element` 來表示。因此，XUL 可描述出非常複雜的視窗型態的使用者介面。

- **缺點：**

是專門設計給 PC 上的視窗程式使用的使用者介面描述方法，其 `element` 的名稱都是相對於視窗上的使用者介面元件名稱，像是 `<window>`、`<menu>`、`<deck>`.....等等。因此，若把 XUL 整個套用過來，用做於描述行動應用程式上的使用者介面並不適合，因為行動裝置並無法表現出複雜的視窗型態的使用者介面。

#### B. *UIML (User Interface Markup Language)*

- **優點：**

在 UIML 中，文件的結構非常完整，將使用者介面的排版(layout)，以及在使用者介面元件上可能觸發的事件，以及使用者介面在各個平台或語言上的呈現方式，以及每個使用者介面元件的屬性參數(attribute/ parameter)，或者事件觸發後所要進行的邏輯運算部份等等，都能分別做描述。也就是說，UIML 將一個應用程式的“使用者介面”和“邏輯運算”兩個部份拆解的相當清楚，並且對於使用者介面部份，也將使用者介面解析的更加詳細：包含使用者介面元件彼此間的結構安排(layout、structure)、使用者介面元件的屬性參數、使用者介面元件的呈現方式.....等等。另外，UIML 不是特別為哪一種執行平台或裝置所設計的，所以可以通用於 PC、行動裝置或是其他可能的目的端裝置。

- **缺點：**

1. 在 UIML 中，使用者介面結構(layout、structure)的安排上，可以描述複雜度很高的使用者介面，例如可以有“巢狀的使用者介面結構”。這會造成使用者介面在轉換上的困難，因為要轉換成各式各樣的使用者介面的話，每種使用者介面的結構安排並不一定相同，甚至差異很大，因此轉換間會有困難。
2. UIML 文件描述完之後，必須靠轉譯器(render)來做使用者介面的轉換。在轉換過程中，必須倚靠 UIML 文件中所提供的，轉換前及轉換後的使用者介面元件間的對應(mapping)資訊。目前的 UIML 缺乏足夠的資訊來提供給 render 做足夠好的轉換。
3. 要轉換到一種使用者介面，便需要一個相對應的 render。因此，撰寫 render 所花費的成本，會隨著所要轉換的使用者介面的增多而增加。另外，各種使用者介面的複雜度不同，要從 UIML 獲得足夠的資訊有限，若是更改 UIML 的小部份描述，可能就需要大幅修改相對應的 render。

### C. **WML (Wireless Markup Language)**

- **優點：**

專門為行動裝置所設計的使用者介面描述語言，根據行動裝置螢幕較小的特性，特別使用了 card 的概念。WML 所訂定的 element 簡單許多，能夠符合各種行動裝置在呈現使用者介面時的能力限制。

- **缺點：**

因為我們要設計的使用者介面描述語言，是要能代表各種行動執行環境上的各種語言的使用者介面的表示方式，而 WML 是包含在 WAP 這個分類(行動執行環境的分類之一)之內的，所以直接拿 WML 來用是不適合的。我們需要的是一個足以位於抽象層級(abstract level)、代表著在所有行動執行環境中，所可以呈現的使用者介面，PUML 能夠滿足這樣的需求。

### D. **XForms**

- **優點：**

1. 在網頁中，為“Form”提供了更豐富的使用者介面元件。
2. 將使用者介面呈現，輸入資料的邏輯運算以及資料儲存，三個部份分開。
3. 利用 XML 格式來儲存所輸入的資料，建立結構化的資料格式。
4. 對於所輸入的資料，能夠有功能較強的邏輯運算，甚至在 client 端便可以處理完畢，而不用透過網路傳送到 server 端處理。
5. 資料在網路傳輸上，能夠有暫停傳輸，或是續傳的能力。
6. 在 client 端便可以處理掉一些原本要在 server 端處理的邏輯運算，所以

可以減少 server 端邏輯運算的量及程式碼的撰寫。

● **缺點：**

最大的缺點便是，僅能被應用在擁有瀏覽器的裝置中，沒有辦法單獨以完整的行動應用程式的姿態存在。

將以上所討論的使用者介面描述語言，透過表格做個結論(表 5-1)：

[表 5-1 各種使用者介面描述語言的比較]

	<b>PUML</b>	<b>XUL</b>	<b>UIML</b>	<b>WML</b>	<b>XForms</b>
<b>Element 的數量</b>	少	多	少	少	少
<b>可描述的使用者介面複雜度</b>	平坦式的使用者介面結構	巢狀的使用者介面結構	巢狀的使用者介面結構	平坦式的使用者介面結構	巢狀的使用者介面結構
<b>專門為視窗程式或瀏覽器所設計(裝置顯示螢幕大)</b>	否	是	否	否	否
<b>專門為行動應用程式所設計(裝置顯示螢幕小)</b>	否	否	否	是	否
<b>專門為跨平台的使用者介面所設計</b>	是	否	是	否	是 (該平台需有瀏覽器)

另外，針對 PUML 和 UIML 這兩個性質上相似的使用者介面描述語言再做個比較(表 5-2)：

[表 5-2 PUML 和 UIML 的比較]

	<b>PUML</b>	<b>UIML</b>
<b>使用者介面的轉換機制</b>	使用 XSLT	使轉譯器(render)
<b>使用者介面轉換所需付出的成本</b>	低	高
<b>轉換出來的使用者介面的可能性</b>	多	多
<b>轉換出來的使用者介面在語意上的滿足性</b>	高	會有誤差

## 5.2 邏輯運算描述語言的比較

先拋開 JavaML 原本是被設計來做軟體工程分析的本意，而把焦點放在它可以完整地用來描述 Java 語言上面。當一個行動應用程式被拆解成兩大部份：使用者介面部份，以及邏輯運算部份之後，便需要一個能夠描述邏輯運算的方法。以一個描述邏輯運算的角度來看待 JavaML 的話，它能夠描述出相當複雜的邏輯運算，甚至能使用非常完整的物件導向的觀念，但最後並不採用 JavaML，反而另外設計 PGML 的原因是在：

- JavaML 完全針對 Java 語言所設計，所以描述的方式相當複雜，不適合學習。
- 本篇論文的目的，是在於將所描述的行動應用程式，轉換成各種目的端的行動應用程式。而用來表示目的端行動應用程式的邏輯運算部份的程式語言，複雜度差異相當大，例如：J2ME 和 WMLScript。因此，若使用 JavaML 做為描述邏輯運算部份的描述方法的話，由於 Java 和 WMLScript 天生的差異性，會使得用 JavaML 所描述好的邏輯運算，轉換不出適當的 WMLScript 程式碼，因為 WMLScript 程式碼太過於簡單，而不能表現出原本在 Java 中可以輕易做到的能力。例如：Java 可以使用陣列(Array)，並做相關運算，但 WMLScript 並沒有陣列(Array)的觀念，因此在轉換上變會產生困難，或是語意上的錯誤。





---

## Chapter 6 Mobile Application Transformation

---

### 6.1 準備做行動應用程式的轉換

當利用 PUML / PGML 撰寫好一個位於抽象層級的行動應用程式之後，本篇論文使用 XSLT 強大的轉換機制，將所描述的行動應用程式，轉換成實際上可執行的目的端行動應用程式。在轉換的過程中，爲了確保轉換出來的行動應用程式能正確無誤地被執行，因此，設計了一些轉換的規則。

在做轉換時，希望能依循一些規則來做轉換，一方面是爲了使撰寫 XSLT style sheet 變的較容易一點。另一方面是，當依據這些規則所轉換出來的行動應用程式有問題發生時，能夠在程式除錯上，或是 XSLT style sheet 的除錯上，提供一個除錯的方向。根據制定好的轉換規則，所轉換出來的程式碼，都必須滿足轉換規則中所有的規範。因此，在這些轉換出來的程式碼中可以觀察到，它們都具有一再使用的轉換規則，我們將這些經常出現的規則，稱爲樣式(pattern)。

在 6.2 節和 6.3 節中，將針對本論文主要要轉換的目的程式語言，提出兩種轉換樣式(transformation pattern)，分別是 WAP Transformation Pattern 和 J2ME Transformation Pattern。在這兩種轉換樣式中，都具有相當嚴謹的轉換規則與要求，以確保所轉換出來的行動應用程式，能夠在相對應的行動執行環境上正確無誤的執行。

### 6.2 WAP Transformation Pattern

在本轉換樣式中，要說明的是：將 PUML 文件轉換成 WML 程式碼的轉換規則，以及將 PGML 文件轉換成 WMLScript 程式碼的轉換規則。以上兩種轉換規則合稱 WAP Transformation Pattern。

使用 PUML / PGML 所描述的行動應用程式，經過轉換成 WML / WMLScript 程式碼之後，所有轉換出來的.wml 檔和.wmls 檔，以及相關的圖檔都放在同一個資料夾下。

#### A. 在 PUML / PGML 中宣告的變數，其轉換之後的統一命名法則

在 WAP 中，所有的變數都是放在同一個 WMLBrowser context 中。所以，

爲了避免在同一份 PUML 文件中，在不同的 board 中出現相同的變數名，或是在不同的 PGML 文件中，出現有相同名稱的全域變數。因爲這些擁有相同名稱的變數，在 WAP 程式執行時，因爲共用同一份 WMLBrowser context，所以可能會產生邏輯語意不正確的情形。爲了要避免這樣的狀況，就必須確定每個要放在 WMLBrowser context 中的變數名稱，必須是獨一無二的。方法如下：

### [轉換在 PUML 中宣告的變數]

在 PUML 文件中，變數的類型有兩種。一種是跟邏輯物件相關的變數，透過 <object> 這個 element 的 attribute – **name** 來宣告，這類型的變數僅代表所要使用的 WMLScript 檔(由相對應的 PGML 文件產生)，所以不會被放到 WMLBrowser context 中，因此不額外做變數命名的處理。

另一種類型的變數，來自於使用者介面元件(UI component)的宣告描述，也就是說，在 PUML 中每描述一種使用者介面元件，都隱性地透過 attribute – **name** 或 **iname**，宣告了屬於該使用者介面元件的變數。這些變數都會被放到 WMLBrowser context 中，所以必須做變數命名的處理。

整體而言，代表使用者介面元件的變數，在轉換之後的命名法則爲：

**Board name\_UI component name\_UI component attribute**

Board name 是用來說明，該代表使用者介面元件的變數是位於哪個 <board> 的 element content 之中。UI component name 即是使用者介面元件的 attribute – **name** 或 **iname** 的屬性值。在 PUML 中，使用者介面元件的 element 可以有種屬性(attribute)，這些屬性和其屬性值可能也需要被當成變數和變數值，放到 WMLBrowser context 中，所以在上面的命名法則，也把 UI component attribute 加入，但不一定會使用到。經過這樣的命名法則之後，便可以確保在 WMLBrowser context 中的變數名稱都是唯一的。

以下舉例說明：

使用者介面元件的宣告描述	<picture name="demo" source="demo.jpg" altText="Demo Image" align="center"/> [屬於<board name="welcomeBoard">的 element content]
轉換之後所產生的變數名稱	無
相對應的 WML 程式碼	

備註	因為在 WML 中，<img>的 attribute – <i>src</i> 這個 attribute 的屬性值無法用變數來取代，也就是說 是不允許的，所以執行於行動應用程式背後的邏輯運算，無法在行動應用程式執行期間更改圖片。因此，不對此使用者介面元件做任何的變數名稱的處理。
----	--

使用者介面元件的宣告描述	<label name="hintMsg" showText="input bellow informaions: "/> [屬於<board name="welcomeBoard">的 element content]
轉換之後所產生的變數名稱	welcomeBoard_hintMsg_showText
相對應的 WML 程式碼	<b>\$(welcomeBoard_hintMsg_showText)</b>
備註	因為<label>在使用者介面上的表現，是一串唯讀的文字。而希望行動應用程式在執行期間，能透過後端的邏輯運算，去更改其顯示在使用者介面上的文字，所以產生這樣子的變數，用來儲存<label>的 attribute – <i>showText</i> 的屬性值。

使用者介面元件的宣告描述	<textnote name="userName" type="text" title="name: "/> [屬於<board name="welcomeBoard">的 element content]
轉換之後所產生的變數名稱	1. welcomeBoard_userName 2. welcomeBoard_userName_title
相對應的 WML 程式碼	<p>\$( welcomeBoard_userName_title)<input name="welcomeBoard_userName" type="text"/></p>
備註	此使用者介面會產生兩種變數。第一種變數名稱直接會被使用在 WML 中的<input>的 attribute – <i>name</i> 中，也就是在 WML 中，用來記錄使用者輸入資料的變數。  此使用者介面元件可以透過 attribute – <i>title</i> ，來在使用者介面的呈現上補充說明此使用者介面元件的描述，例如輸入字串的提示。因為也希望這樣子的提示說明，也能夠透過後端邏輯運算來做更改，因此也將它以變數的型態存放在 WMLBrowser context 中，所以產生第二種的變

	數名稱，用來儲存<textnote>的 attribute – <b>title</b> 的屬性值。
--	--

使用者介面元件的宣告描述	<pre>&lt;listpaper title="select one fruit:" iname="iFruit" mode="single"&gt;   &lt;item showText="apple"/&gt; &lt;/listpaper&gt;</pre> <p>[屬於&lt;board name="fruitBoard"&gt;的 element content]</p>
轉換之後所產生的變數名稱	<ol style="list-style-type: none"> <li>1. fruitBoard_iFruit</li> <li>2. fruitBoard_iFruit_title</li> <li>3. fruitBoard_iFruit_item1</li> </ol>
相對應的 WML 程式碼	<pre>&lt;p&gt;\$( fruitBoard_iFruit_title) &lt;select iname="fruitBoard_iFruit" multiple="false"&gt;   &lt;option&gt;\$(fruitBoard_iFruit_item1)&lt;/option&gt; &lt;/select&gt;&lt;/p&gt;</pre>
備註	<p>此使用者介面會產生三種變數。第一種變數名稱直接會被使用在 WML 中的&lt;select&gt;的 attribute – <b>iname</b> 中，也就是在 WML 中，用來記錄使用者選擇了哪些選項的變數。</p> <p>此使用者介面元件可以透過 attribute – <b>title</b>，來在使用者介面的呈現上補充說明此使用者介面元件的描述，例如選擇項目的提示。因為也希望這樣子的提示說明，也能夠透過後端邏輯運算來做更改，因此也將它以變數的型態存放在 WMLBrowser context 中，所以產生第二種的變數名稱，用來儲存&lt;listpaper&gt;的 attribute – <b>title</b> 的屬性值。</p> <p>此使用者介面元件在最後的使用者介面的呈現上，是以選單的型式出現。在行動應用程式執行期間，也希望能夠透過後端邏輯運算的處理，來更改選單中選項的值，所以產生第三種變數名稱。在&lt;listpaper&gt;中可能會有多個&lt;item&gt;，所以每個&lt;item&gt;相對應的變數名稱都會加上編號，例如第一個&lt;item&gt;對應的變數名稱為 xxx_xxx_item1，第二個為 xxx_xxx_item2，以此類推。</p>

#### [轉換在 PGML 中宣告的變數]

為了避免在不同的 PGML 文件中有相同名稱的全域變數產生，導致 WAP 行動應用程式執行時，邏輯語意可能產生錯誤的情況，所以希望在轉換之後能夠產生唯一的變數名稱。

在 PGML 文件中，代表全域變數的變數名稱，在轉換之後的命名法則為：

**Logic object name\_global variable name**

Logic object name 是來自於 PGML 文件的<object>這個 root element 中的 attribute – *name*，用來表示該全域變數是屬於哪個 PGML 文件，也就是用來表示該全域變數是屬於哪個邏輯物件。global variable name 是來自於被宣告成全域變數的<variable>的 attribute – *name*。經過這樣的命名法則之後，便可以確保在 WMLBrowser context 中的變數名稱都是唯一的。

以下舉例說明：

邏輯物件(PGML 文件)中的全域變數	相對應的 WMLScript 變數名稱
UserInfo.pgml 中的 <b>userName</b>	<b>UserInfo_userName</b>
UserInfo.pgml 中的 <b>userPassword</b>	<b>UserInfo_userPassword</b>
FruitInfo.pgml 中的 <b>choosed</b>	<b>FruitInfo_choosed</b>
.....	.....

## B. PUML 轉成 WML 的轉換規則

接下來以條例式的方式，列出 PUML 轉成 WML 的轉換規則。

- PUML 文件轉換成相對應的 WML 檔案之後，該 WML 檔的檔案名稱必須跟 PUML 文件的檔案名稱一樣，而副檔名由.puml 改成.wml。  
例如：demo.puml → demo.wml
- 在轉換之後的 WML 程式碼中，不會使用<template>這個 element，來表示每個 WML card 中重覆、相同的邏輯，或是使用者介面元件。因為每個使用者介面，都在 PUML 的<board> element content 來各別描述。
- 在轉換之後的 WML 程式碼中，每個 card 中都不會有<prev/> (回上一頁) 這個 task。因為在 PUML 中，使用<nextboard>這個 element 來控制全部的使用者介面的瀏覽順序。所以在產生 WML 程式碼時，全部使用<go> 這個 element 來表示所要瀏覽的下個使用者介面。

因此，也不會有以下<template>產生：

```
<template>  
  <do type="prev" name="UIBack"><prev/></do>  
</template>
```

因為在每個使用者介面都能回到上一頁的話，是不符合整個行動應用程式的使用者介面瀏覽語意的。

- 在轉換之後的 WML 程式碼中，有兩個必要的<card> element，其 attribute – *id* 的屬性值分別是：
  - **@user-interfaceName\_index**。  
(@user-interfaceName 是指相對應的 PUML 文件中，<user-interface>的 attribute – *name* 的屬性值。)
  - **preSetting**。

例如：

將 DemoUI.puml (其 root element 即是<user-interface name="DemoUI">) 轉換成 DemoUI.wml 後，WML 程式碼中會包含兩個 card。

一個是<card id=" DemoUI\_index">，另一個是<card id=" preSetting">。

- attribute – *id* 的屬性值為**@user-interfaceName\_index** 的 card，為第一個必要的 card，也就是.wml 檔中第一個出現的 card。負責設定 PUML 文件中，所有用到的邏輯物件(global 或 local 的邏輯物件都要)的全部全域變數。
  - 利用每個邏輯物件對應的 WMLScript 檔的 globalVarDeclaration( ) function，來設定該邏輯物件的全域變數。
  - 每個邏輯物件都有一個相對應的 inner card。利用一連串的 inner card 來完成所有邏輯物件的全域變數的設定 (chain of inner cards)。

[詳細作法如下：]

- a) 每個邏輯物件(logic object)都有一個相對應的 inner card，其 attribute – *id* 的命名規則為：
  - 若在 PUML 中，為 global logic object 者，其對應的 inner card 的 attribute – *id* 的屬性值 **inner\_@user-interfaceName\_@name**。
  - 若在 PUML 中，為 local logic object 者，其對應的 inner card 的 attribute – *id* 的屬性值為 **inner\_@boardName\_@name**。  
( **@boardName** 是該 local logic object 所在的<board>的 attribute – *name* 的屬性值。**@name** 是代表邏輯物件的<object>的 attribute – *name* 的屬性值 )
- b) 在轉換出來的 WML 程式碼中，inner card 出現的順序跟其相對應的邏輯物件，出現在 PUML 文件中的順序一樣。而這一連串的 inner card，依照其出現順序來一一執行，因此稱呼這一連串的 inner card 為 **chain of inner cards**。
- c) attribute – *id* 的屬性值為**@user-interfaceName\_index** 的 card, 沒有 onenterforward event, 但有 ontimer event (timer=1)。

- 若有 inner card 的話，則 ontimer event 觸發後，便要跑到第一個 inner card 來執行。
  - 若無 inner card 的話，則 ontimer event 觸發後，便要跑到 attribute – *id* 的屬性值為 preSetting 的 card 來執行。
- d) 每個 inner card 都會有兩個 event：
- 一個為 onenterforward event。此 event 觸發後，要執行相對應的邏輯物件的相對應 WMLScript 檔中的 globalVarDeclaration()。
  - 一個為 ontimer event。此 event 觸發後，要依照順序執行下一個 inner card。若無下一個 inner card 的話，則要執行 attribute – *id* 的屬性值為 preSetting 的 card (timer= 1)。
- attribute – *id* 的屬性值為 preSetting 的 card 為第二個必要的 card。有兩個 event：onenterforward event 和 ontimer event：
    - a) 在 onenterforward event 觸發後，要做的事為：設定出現在 PUML 文件中的，所有使用者介面元件(UI component)的相關變數。
    - b) 在 ontimer event 觸發後，要執行 PUML 文件中所設定的，第一個要呈現的使用者介面所對應到的 card。
  - 轉換成 WML / WMLScript 形式的行動應用程式後，在開始執行前必須有一些前置作業，這些前置作業必需由一些特殊的 cards 來完成。這些 cards 的執行順序為：attribute – *id* 的屬性值為 @user-interfaceName\_index 的 card → chain of inner cards → attribute – *id* 的屬性值為 preSetting 的 card → PUML 文件中設定的第一個要呈現的使用者介面，所對應的 card。
  - PUML 文件中，<action> element content 中的每一個 child element 都會對應到一個 temp card，該 temp card 的 attribute – *id* 的屬性值為 **@boardName\_@actionName\_temp[child position]** (**@actionName** 是該 child element 所在的<action>的 attribute – *name* 的屬性。[**child position**]的值，來自於該 child element 在<action>的 element content 中出現的順序。)
  - a) 在轉換出來的 WML 程式碼中，temp card 出現的順序跟其相對應的 child element，出現在<action>的 element content 中的順序一樣。而這一連串的 temp card 會依照其出現的順序來一一執行，因此稱呼這一連串的 temp card 為 **chain of temp cards**。
  - b) 由<nextboard>對應的 temp card，僅會驅動一個 onenterforward event。此事件驅動之後，便會執行該<nextboard>的 attribute – *goto*

的屬性值所指定的使用者介面，所對應到的 card。

- c) 由<use-object>、<change>對應的 temp card，會驅動兩個 event：onenterforward event 和 ontimer event。且 timer value 為 1 (timer=1)。
- d) 當<use-object>對應的 temp card 的 onenterforward event 被驅動時，便根據該<use-object>的 attribute – **name** 的屬性值，來找到這個屬性值所指的邏輯物件的相對應的 WMLScript 檔，然後再根據 attribute – **method** 的屬性值，來知道要使用該 WMLScript 檔的哪個 function，如果該 function 需要傳入參數的話，則可以由該<use-object>的 child element <param>的 attribute – **select** 的屬性值，來獲得所要傳入的參數值。

當<use-object>對應的 temp card 的 ontimer event 被驅動時，便會根據所在的 chain of temp cards 的執行順序，依序執行下一個 temp card。

- e) 當<change>對應的 temp card 的 onenterforward event 被驅動時，會依據該<change>的 child element 不同，而做不同的處理。
  - child element 為<use-object>時：  
執行該<use-object>所指的邏輯物件所對應的 WMLScript 檔中的 chsnge( ) function。此 function 需要三個參數，第一個參數 (component)的參數值，為<change>的三個 attribute – **container**、**component**、**update** 所描述的使用者介面元件相對應的變數名稱 (此變數的命名規則，請參考 A 小節)。第二個參數(newValue)的參數值為空字串。第三個參數(callMethod)的參數值，為<use-object>的 attribute – **method** 的屬性值。
  - child element 為 text child 時：  
執行第一個在 PUML 文件中出現的 global logic object 所對應的 WMLScript 檔的 change( ) function，若無 global logic object 的話，則執行第一個在 PUML 文件中出現的 local logic object 所對應的 WMLScript 檔的 change( ) function，若無任何的 logic object 的話，則使用 PGMLchange.wmls(轉換系統自己產生的 WMLScript 檔)中的 change( ) function。

此 function 需要三個參數，第一個參數(component)的參數值，為<change>的三個 attribute – **container**、**component**、**update** 所描述的使用者介面元件相對應的變數名稱 (此變數的命名規則，請參考 A 小節)。第二個參數(newValue)的參數值，即為 text child 所表示的字串。第三個參數(callMethod)的參數值為空字串。



- f) 如果在 chain of temp cards 中，最後一個 child element 所對應的 temp card，不是由<nextboard>所對應而來的話，則最後一個 temp card 必須靠 ontimer event 的觸發，來回到該 child element 所在的 <board> element content 所描述的使用者介面，所對應的 card。
  - g) 因為所做的 WML 轉換，是轉換到 WML 1.3 的版本，所以任何會使用到的圖檔，都要轉換成 wbmp 格式(.wbmp 檔)。
  - h) 所使用到的圖檔，和所有轉換出來的 WML 檔以及 WMLScript 檔，都放同一個資料夾下。
  - i) PUMML 的<picture>的 attribute – *align* 對應到 WML 的<img>的 attribute – *align*。其屬性值的對應方式是：  
(left, center, right)  $\leftrightarrow$  (top, middle, bottom)
- 其他 WML 程式碼的轉換部份，則根據 WML 的語法，對照著 PUMML 的 element 做很直覺的轉換，可參考 appendix 中所附的 PUMML 的原始碼，以及其轉換出來的 WML 程式碼。

### C. PGML 轉成 WMLScript 的轉換規則

接下來以條列式的方式，列出 PGML 轉成 WMLScript 的轉換規則。

- 由於在 WMLScript 中，沒有全域變數的概念(global variable)，但在 PGML 文件中卻可以描述宣告全域變數來使用。因此，轉換成 WMLScript 之後，使用 WMLBrowser.setVar( )和 WMLBrowser.getVar( )來將 PGML 中所使用的全域變數放到 WMLBrowser context 中，或是從 WMLBrowser context 中取出所要使用的全域變數，達到存取全域變數的效果。
- 每個轉換出來的 WMLScript 檔都要有一個對外開放的 function – globalVarDeclaration( )。在這個 function 裡頭要做的事，便是利用 WMLBrowser.setVar( )來將相對應的 PGML 文件所描述宣告的全域變數，存放到 WMLBrowser context 中，完成全域變數的設定。(PGML 文件的全域變數，轉換成 WMLBrowser context 中的變數的命名規則，請參考 A 小節)
- 每個轉換出來的 WMLScript 檔都要有一個對外開放的 function – change(component, newValue, callMethod)。這個 function 是用來改變使用者介面元件的狀態值的，因此有三個參數：第一個參數(component)所指的是要被改變的使用者介面元件，第二個參數(newValue)是單純地透過參數傳遞，來指出使用者介面元件所要被更新的狀態值，第三個參

數(callMethod)是透過參數傳遞，來指出要使用位於此 WMLScript 檔的哪個 function，而希望該 function 的執行結果就是使用者介面元件的新狀態值。而第二個和第三個參數傳遞的狀況，則根據相對應的 PUML 文件中的<change> element 的使用狀況來做決定，其中會有一個空字串的傳遞。可以參考 B 小節的介紹。

因此可以看出，若希望某 function 的執行結果就是使用者介面元件的新狀態值的話，那麼所要呼叫的 function，必須不必傳入任何參數值，因為每個 function 所需要的參數數量不一定相同。因此，為了簡化設計起見，在 **change( ) function** 的設計上，僅使用三個參數(component、newValue、callMethod)，不再額外使用其它參數來表示所要呼叫的 function 的參數。以下列出 change( ) function 的架構圖(圖 6-1)：

```
extern function change(component, newValue, callMethod){
    if(callMethod == ""){
        WMLBrowser.setVar(component, newValue);
    }
    else{
        if(callMethod == "method1"){
            WMLBrowser.setVar(component, method1());
        }
        if(callMethod == "method2"){
            WMLBrowser.setVar(component, method2());
        }
        //...
    }
}
```

[圖 6-1 change( ) function 架構圖]

- 使用 PUML / PGML 所描述的行動應用程式，經過轉換成 WML / WMLScript 之後，所有轉換出來的.wml 檔和.wmls 檔，以及相關的圖檔都放在同一個資料夾下。另外，在同一個資料夾下，轉換系統必須另外產生一個 WMLScript 檔 – PGMLchange.wmls，該檔案就只有一個 change( ) function，檔案內容如下(圖 6-2)：

```
extern function change(component, newValue, nouse){
    WMLBrowser.setVar(component, newValue);
}
```

[圖 6-2 PGMLchange.wmls]

由檔案內容可以看出，第三個參數(nouse)是沒有作用的，參數傳入值在轉換機制中是設定為空字串。

- 在 PGML 文件中描述數學運算時，若使用了代表全域變數的<variable>，或是使用了<call-method>、<lib>的話，在轉換成 WMLScript 檔之後，要在相對應的程式碼中，加上 Lang.parseInt( )這個 WMLScript

的 library function 的呼叫，確保轉換之後的數學運算無誤。

- 其他 WMLScript 程式碼的轉換部份，則根據 WMLScript 的語法，對照著 PGML 的 element 做很直覺的轉換，可參考 appendix 中所附的 PGML 的原始碼，以及其轉換出來的 WMLScript 程式碼。

## 6.3 J2ME Transformation Pattern

在本轉換樣式(transformation pattern)中，要說明的是：將 PUML 文件轉換成 J2ME MIDP 程式碼的轉換規則，以及將 PGML 文件轉換成 J2ME MIDP 程式的轉換規則。以上兩種轉換規則合稱 J2ME Transformation Pattern。

使用 PUML / PGML 所描述的行動應用程式，經過轉換成 J2ME MIDP 程式碼之後，所有轉換出來的.class，都必須在同個 package 下(即在同一個資料夾下)。

### A. PUML 轉成 J2ME MIDP 的轉換規則

接下來以條列式的方式，列出 PUML 轉成 J2ME MIDP 的轉換規則。

- 一份 PUML 文件轉換成相對應的 J2ME MIDP 之後，會產生兩種類型的 class，以及一個 interface。轉換系統會產生一個 interface – **BMngr\_Interface**。<user-interface>會對應到一個繼承自 **MIDlet** 的 class，且此 class 必須實作轉換系統所產生的 **BMngr\_Interface** interface (extends MIDlet implements BMngr\_Interface)。每個<board>會對應到一個繼承自 **Form** 並實做 **CommandListener** interface 的 class (extends Form implements CommandListener)。
- **BMngr\_Interface** interface 有兩個 method 需要被實做(implementation)。一個是 **changeBoard(String boardName)** method，透過參數傳遞，指出所要切換的使用者介面，並在此 method 被執行之後，畫面切換到所指定的使用者介面。

另一個是 **getBoard(String boardName)** method，透過參數傳遞，指出所要取得的使用者介面，並在此 method 被執行之後，可以取得代表該使用者介面的變數，可以對此變數作操作，例如：更改該使用者介面中的，某個使用者介面元件的狀態值。

```
interface BMngr_Interface {  
    public void changeBoard(String boardName);  
    public Displayable getBoard(String boardName);  
}
```

[圖 6-3 BMngr\_Interface interface]

- 將一份 PUML 文件做轉換之後，只產生一份.java 檔，裡頭包含了所有的 class(第 1 點所提到的兩種 class)，以及 **BMngr\_Interface** 這個 interface。在這份.java 檔中：
  - a) <user-interface>所對應的 class(即繼承 MIDlet 的 class)，必須為 main class。
  - b) main class 必須是 public class，其他的 class、interface 不使用 **public** 這個關鍵字。也就是說，在同一份.java 檔中的所有 class 們，僅能有一個 class 是 **public**，且 interface 不能是 **public**。
  - c) .java 檔的檔名要跟 main class 的 class name 一樣。
  
- 在轉換成 J2ME MIDP 程式碼之後，<user-interface> element 會對應到一個繼承自 MIDlet 的 class，且該 class 實作轉換系統產生的一個 interface – **BMngr\_Interface**。<user-interface> 所對應到的 class 名稱為：**BMngr\_@user-interfaceName**。這裡的@user-interfaceName 指的是<user-interface>的 attribute – *name* 的屬性值。
  
- 在轉換成 J2ME MIDP 程式碼之後，<board> element 會對應到一個繼承自 Form 的 class，且該 class 實作一個 interface – **CommandListener**。<board> 所對應到的 class 名稱為：**B\_@boardName**。這裡的@boardName 指的是<board>的 attribute – *name* 的屬性值。
  
- 在 **BMngr\_@user-interfaceName** class 中：
  - a) 在 PUML 文件中描述宣告的 global logic object 所對應的 class，在轉換成 J2ME MIDP 程式碼之後，必須在此 **BMngr\_@user-interfaceName** class 中，一開始就被宣告並產生物件出來(使用 **new** 這個關鍵字)，並且對外開放(使用 **public** 這個關鍵字)。由於邏輯物件(logic object)對應的 class，其建構子(constructor)不需要參數，所以在產生物件的過程中，不需給與初始值。

[例如]

在 PUML 文件中有一個 global logic object 的描述宣告為：

**<object name="userInfo" source="UserInfo.pgml"/>**。

這個描述相對應的程式碼為：

**public UserInfo lgObj\_userInfo = new UserInfo();**

- b) logic object 的變數名要加上 “**lgObj\_**” 這個前置敘述(prefix)，用來表示此變數代表的是一個 logic object。即 **lgObj\_@objectName**，這裡的@objectName 指的是相對應的<object>的 attribute – *name* 的屬

性值。

- c) 在 PUMML 文件中描述的<board>所對應的 class，在轉換成 J2ME MIDP 程式碼之後，必須在此 **BMngr\_@user-interfaceName** class 中，一開始就被宣告並產生物件出來(使用 **new** 這個關鍵字)，但不對外開放(使用 **private** 這個關鍵字)。由於<board>對應的 class，其建構子(constructor)不需要參數，所以在產生物件的過程中，不需給與初始值。

[例如]

在 PUMML 文件中有一個<board>的描述為：

```
<board name="welcomeBoard" title="Simple Demo">。
```

這個描述相對應的程式碼為：

```
private B_welcomeBoard welcomeBoard = new B_welcomeBoard();
```

- d) 在 **BMngr\_@user-interfaceName** class 中，要有下列變數宣告敘述：

```
public static BMngr_@user-interfaceName instance;  
private Display display;
```

(@user-interfaceName 為<user-interface>的 attribute – *name* 的屬性值)

- e) 若在 PUMML 文件中沒有特別說明的話，以第一個<board>所對應的 class，當第一個使用者介面呈現。在轉換成 J2ME MIDP 程式碼後，以 *display.setCurrent()* 這個 method 來設定第一個要呈現的使用者介面。
- f) 實作 **BMngr\_Interface** interface 的 *changeBoard(String boardName)* method，實作方法如下：

比較傳入的參數值，然後利用 *display.setCurrent()* 這個 method 來切換所要呈現的使用者介面。在轉換過程中，會將所有的<board>所對應的使用者介面，都列入 J2ME MIDP 的程式比較中，下面是 J2ME MIDP 程式碼的片段(圖 6-4)：

```
public void changeBoard(String boardName){  
    if(boardName.compareTo("welcomeBoard") == 0){  
        display.setCurrent(welcomeBoard);  
    }  
  
    if(boardName.compareTo("infoBoard") == 0){  
        display.setCurrent(infoBoard);  
    }  
  
    if(boardName.compareTo("fruitBoard") == 0){  
        display.setCurrent(fruitBoard);  
    }  
  
    if(boardName.compareTo("colorsBoard") == 0){  
        display.setCurrent(colorsBoard);  
    }  
  
}
```

[圖 6-4 changeBoard() method 的 J2ME MIDP 範例程式碼片段]

- g) 實作 **BMngr\_Interface** interface 的 *getBoard(String boardName)*

method，實作方法如下：

比較傳入的參數值，然後回傳參數值所指定的使用者介面的變數 (data type 為 displayable)。在轉換過程中，會將所有的<board>所對應的使用者介面，都列入 J2ME MIDP 的程式比較中，下面是 J2ME MIDP 程式碼的片段(圖 6-5)：

```
public Displayable getBoard(String boardName){
    Displayable displayable = null;

    if(boardName.compareTo("welcomeBoard") == 0){
        displayable = welcomeBoard;
    }

    if(boardName.compareTo("infoBoard") == 0){
        displayable = infoBoard;
    }

    if(boardName.compareTo("fruitBoard") == 0){
        displayable = fruitBoard;
    }

    if(boardName.compareTo("colorsBoard") == 0){
        displayable = colorsBoard;
    }

    return displayable;
}
```

[圖 6-5 getBoard() method 的 J2ME MIDP 範例程式碼片段]

- 在 **B\_@boardName** class 中：
  - a) 建構子(constructor)不需參數。
  - b) 在 PUMML 文件中描述宣告的 local logic object 所對應的 class，在轉換成 J2ME MIDP 程式碼之後，必須在此 **B\_@boardName** class 中，一開始就被宣告並產生物件出來(使用 **new** 這個關鍵字)，但不對外開放(使用 **private** 這個關鍵字)。由於邏輯物件(logic object)對應的 class，其建構子(constructor)不需要參數，所以在產生物件的過程中，不需給與初始值。

[例如]

在 PUMML 文件中有一個 local logic object 的描述宣告為：

```
<object name="fruitInfo" source="FruitInfo.pgml"/>。
```

這個描述相對應的程式碼為：

```
private FruitInfo lgObj_fruitInfo = new FruitInfo();
```

- c) logic object 的變數名要加上 "**lgObj\_**" 這個前置敘述(prefix)，用來表示此變數代表的是一個 logic object。即 **lgObj\_@objectName**，這裡的**@objectName** 指的是相對應的<object>的 attribute – **name** 的屬性值。
- d) 在 PUMML 文件中描述的使用者介面元件，在轉換成 J2ME MIDP 程式碼之後，必須在此 **B\_@boardName** class 中，一開始就被宣告，

並且對外開放(使用 **public** 這個關鍵字)。

[例如]

在 PUML 文件中有一個使用者介面的描述為：

```
<label name="helloMsg" showText="Hello, "/>。
```

這個描述相對應的程式碼為：

```
public StringItem helloMsg;
```

- e) 在 PUML 文件中描述的<action> element，在轉換成 J2ME MIDP 程式碼之後，必須在此 **B\_@boardName** class 中，一開始就被宣告並產生物件出來(使用 **new** 這個關鍵字)，但不對外開放(使用 **private** 這個關鍵字)。<action>對應於 Command 物件，且 command type 使用 **Command.SCREEN**，而 priority 設成 1。

[例如]

在 PUML 文件中有一個使用者介面的描述為：

```
<action name="singleFruit" showText="Choose a Fruit">。
```

這個描述相對應的程式碼為：

```
private Command singleFruit =  
new Command("Choose a Fruit", Command.SCREEN, 1);
```

- f) 在建構子(constructor)中必須做的事有：
- 初始化使用者介面元件
  - 設定 Command Listener  
(Command Listener 即是本身 **B\_@boardName** class，因為此 class 實作了 **CommandListener** interface)
  - 把使用者介面元件，排放到包含它的使用者介面之中
  - 在使用者介面中，加入用來驅動事件的 Command

[例如]

在 PUML 文件中，使用者介面描述的片段為：

```
<board name="infoBoard" title="User Information">
```

```
    <label name="helloMsg" showText="Hello, "/>
```

```
    .....
```

```
    <action name="singleFruit" showText="Choose a Fruit">
```

```
    .....
```

```
</board>
```

這個描述片段在建構子中，相對應的程式碼為：

```

public B_infoBoard() {
    /* initialize UI component */
    super("User Information");
    helloMsg = new StringItem("Hello, ", "");
    .....
    /* set up listeners */
    this.setCommandListener(this);
    /* append UI component */
    this.append(helloMsg);
    .....
    /* add commands */
    this.addCommand(singleFruit);
    .....
}

```

- PUML 文件中的<textnote>，在轉換成 J2ME MIDP 程式碼時，對應到的是 **TextField** class。再產生相對應的 TextField 的物件時，預設可供使用者輸入資料的欄位長度是 50。
- 任何會使用到的圖檔，都要轉換成 png 格式(.png 檔)。
- 在轉換成 J2ME MIDP 程式碼時，要使用 PUML 文件裡，所描述的某個使用者介面元件的狀態值時，記得要使用該使用者介面元件相對應的 class，裡頭的 method 來獲得，不要直接使用代表該使用者介面元件的變數 (此為轉換時可能犯的錯誤)。
- 在 PUML 文件中，可能會透過<action> element content 的描述，來將 <listpaper>所代表的使用者介面元件的狀態值，交給 PUML 文件中所指定的邏輯物件來處理。而處理的方式便是呼叫該邏輯物件相對應的 class 中的 method，使用者介面元件的狀態值便被當成參數，傳入該 method 之中。這個時候，必須分兩種狀況來做轉換的處理：
  - 當<listpaper>的 attribute – *mode* 的屬性值為 “single” 時(單選) 使用 <listpaper> 相對應的 **ChoiceGroup** class 中的 **getSelectedIndex( )** method，來取得使用者介面元件的狀態值。由於 PGML 文件所描述出來的邏輯物件，在處理<listpaper>所對應到的使用者介面元件時，選單中選項的編號是從 1 開始，而在 J2ME MIDP 中，選項編號是從 0 開使。所以利用所提到的 method 來取得狀態值之後，必須再加上 1，才能讓邏輯物件來做邏輯運算的處理。



- 當<listpaper>的 attribute – *mode* 的屬性值為 “**multi**” 時(多選) 要將<listpaper>所對應的使用者介面元件的狀態值，轉化成一個 “多選字串” 之後，才能交給邏輯物件來做邏輯運算的處理。所謂 “多選字串”，即是將使用者在選單中，所選擇的多個項目的項目編號組合成一個字串，該字串稱為 “多選字串”，例如：使用者選擇了編號分別為 1、2、3、4 的項目，則該 “多選字串” 便是 “1;2;3;4”。
- “多選字串” 必須在做邏輯運算之前產生出來，產生多選字串的範例程式碼如下(圖 6-6)：

```

selectedStr = ""; //selectedStr是String
for(int i=0; i < iColors.size(); i++){ //iColors是ChoiceGroup
    if(iColors.isSelected(i)){
        if(selectedStr != ""){
            selectedStr = selectedStr + ";" + Integer.toString(i+1);
        }
        else{
            selectedStr = selectedStr + Integer.toString(i+1);
        }
    }
}

```

[圖 6-6 多選字串的範例程式碼]

同樣地，由於 PGML 文件所描述出來的邏輯物件，在處理<listpaper>所對應到的使用者介面元件時，選單中選項的編號是從 1 開始，而在 J2ME MIDP 中，選項編號是從 0 開始。所以利用所提到的 method 來取得狀態值之後，必須再加上 1，才能讓邏輯物件來做邏輯運算的處理。

- 其他 J2ME MIDP 程式碼的轉換部份，則根據 J2ME MIDP 的語法，對照著 PUMML 的 element 做很直覺的轉換，可參考 appendix 中所附的 PUMML 的原始碼，以及其轉換出來的 J2ME MIDP 程式碼。

## B. PGML 轉成 J2ME MIDP 的轉換規則

接下來以條列式的方式，列出 PGML 轉成 J2ME MIDP 的轉換規則。

- 一個 PGML 文件所描述的邏輯物件，在轉換成 J2ME MIDP 之後，會對應到一個對外開放的 class (使用 **public** 這個關鍵字)，而該 class 的名稱即為 PGML 文件中<object>的 attribute – *name* 的屬性值。且這個 class 的建構子(constructor)不需任何參數。
- 這部份程式碼的轉換，並沒有特別的規則要遵守。則根據 J2ME MIDP 的語法，對照著 PGML 的 element 做很直覺的轉換即可。可參考 appendix 中所附的 PGML 的原始碼，以及其轉換出來的 J2ME MIDP 程式碼。

---

## Chapter 7 Mobile Application Examples

---

### 7.1 使用者介面導向的程式技巧

在使用 PUML / PGML 來撰寫行動應用程式時，可以依循這裡所建議的程式寫作方法來撰寫(並非一定要照此方式來撰寫行動應用程式)，將可以加快行動應用程式的開發。

由於受限於行動裝置螢幕較小的因素，並且希望行動應用程式的使用者能夠有較好的使用者經驗(user experience)，所以在使用者介面的設計上就必須格外地用心。因此，若能以使用者介面的設計為主要思考，邏輯部份為輔，對於行動應用程式的設計應能事半功倍。所以，在撰寫行動應用程式上，本篇論文提出一個寫作的建議，那就是：**使用者介面導向的程式技巧(UI-oriented programming)**。

使用者介面導向的程式技巧可大致分成三大步驟：

1. 首先，一開始便規劃好一個行動應用程式可能所需要的全部使用者介面，並將這些使用者介面彼此間的關係規劃好(例如：使用者介面瀏覽的順序)。
2. 第二，確定每個使用者介面中，有哪些使用者介面元件的狀態(例如：使用者輸入的字串)需要被送到後端的邏輯運算部份做處理。
3. 最後，選擇適當的邏輯物件(logic object，即 PGML 所描寫的物件，即 PGML 文件)來處理所希望的邏輯運算，若無適當的邏輯物件存在，則利用 PGML 描述所需的邏輯物件來使用。

### 7.2 兩個 PUML / PGML 行動應用程式的例子

這邊舉出兩個實際利用 PUML / PGML 撰寫的行動應用程式的例子。

在 Simple Demo 的例子中，充分結合 PUML 和 PGML 的能力，做一個整合的展示，盡量地展現出 PUML 和 PGML 能做到的事，是一個淺顯易懂的示範行動應用程式。在這個應用程式中，PUML 方面，主要是顯示各個使用者介面，更改使用者介面元件的狀態值，使用邏輯物件以及切換顯示畫面(切換使用者介面)。在 PGML 方面，則是提供必要的邏輯運算。

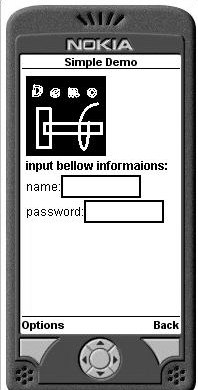
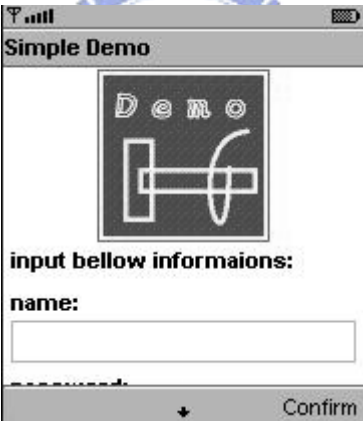
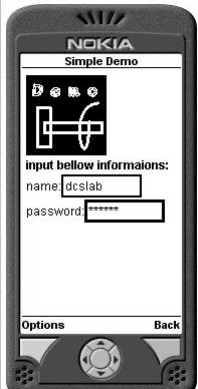
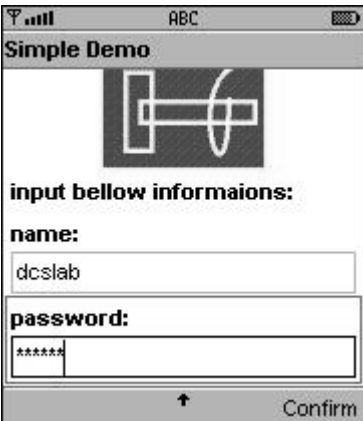
在 Calculator 的例子中，展示出來的是一個小算盤的行動應用程式。這個程式可以做簡單的"正負整數"運算：可做加、減、乘、除、mod，並有 M+、MC、MR、MS、C、=，以及+/- (正負號轉換)等功能。另外，當輸入錯誤時，會顯示錯誤訊息給使用者知道，是一個功能還算完備的“整數運算小算盤”。


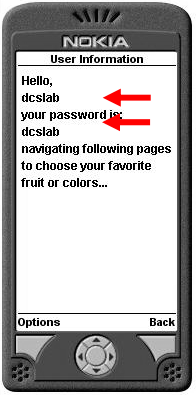


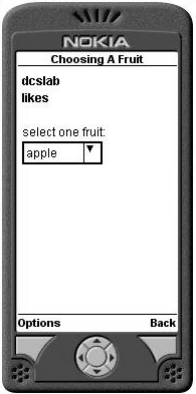

以下的兩個行動應用程式的展示，均只展示執行時期的畫面，分別是執行 WAP 應用程式(使用 Nokia Mobile Browser Simulator 4.0，NMBS 4.0)，以及執行 J2ME MIDP 應用程式(使用 Java Wireless TooKit 2.0 中的預設模擬器)時的執行畫面。兩個行動應用程式的程式碼(PUML / PGML)均放在 Appendix 中，以提供參考。

### A. Example 1 – Simple Demo

表 7-1，比較在 Simple Demo 這個行動應用程式執行時，使用者相同的動作或指令，在不同的行動執行環境上所呈現出來的樣子和效果。

[表 7-1 Example1 在執時期的畫面(WML/WMLScript 和 J2ME MIDP)]

WML / WMLScript	J2ME MIDP	簡單說明
		行動應用程式執行時的開始畫面(第一個使用者介面)。
		輸入資料。

		<p>確定所輸入的資料。 (按下[Confirm])</p>
		<p>將之前所輸入的資料顯示出來。</p>
		<p>進入功能選單。 (選擇 <b>Choose a Fruit</b>)</p>
		<p>切換到 <b>Choosing a Fruit</b> 的使用者介面。</p>

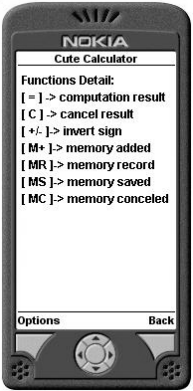
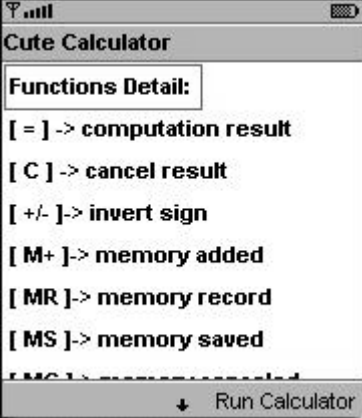
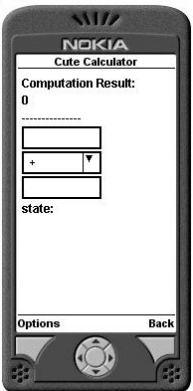
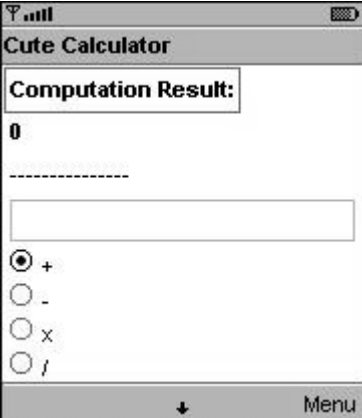
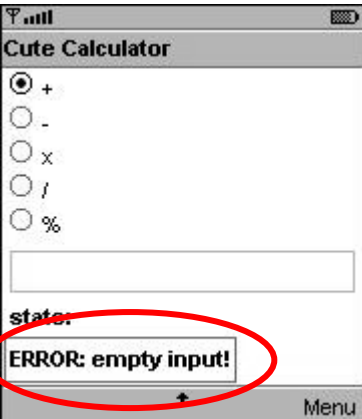
		<p>選擇喜歡的水果。</p>
		<p>選了芒果(mango)。</p>
		<p>確定選擇芒果(mango)。</p>
		<p>顯示選擇結果。</p>

		<p>切換到 <b>Choosing Colors</b> 的使用者介面。</p>
		<p>選擇喜歡的顏色。</p>
		<p>顯示選擇結果。</p>

**B. Example 2 – Calculator**

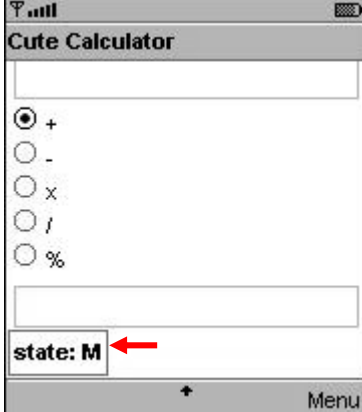
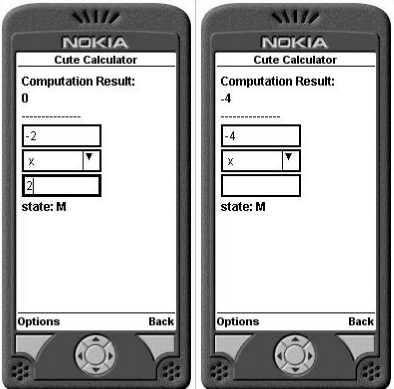
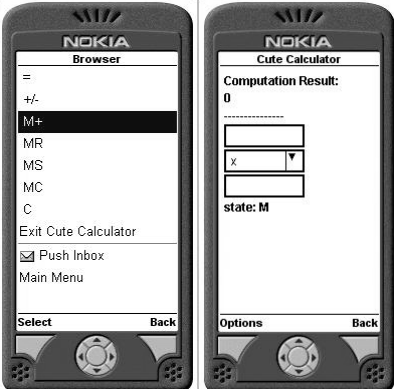

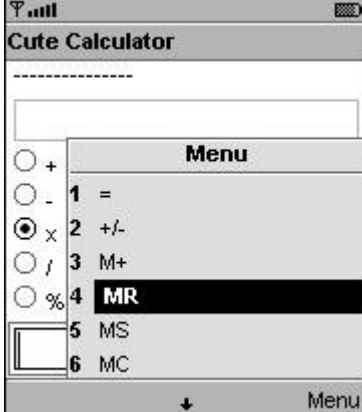
表 7-2，比較在 Calculator 這個行動應用程式執行時，使用者相同的動作或指令，在不同的行動執行環境上所呈現出來的樣子和效果。

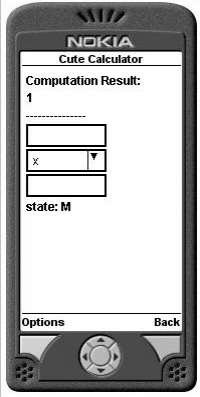
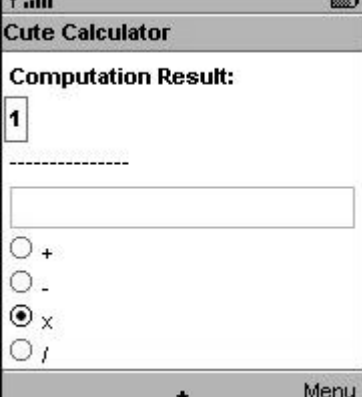

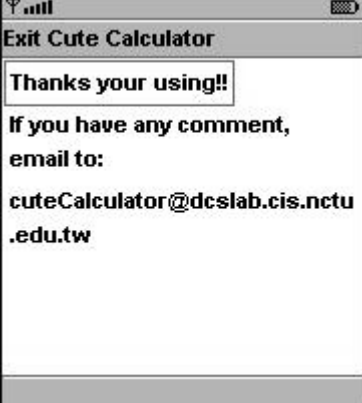
[表 7-2 Example2 在執時期的畫面(WML/WMLScript 和 J2ME MIDP)]

WML / WMLScript	J2ME MIDP	簡單說明
		<p>行動應用程式執行時的開始畫面(第一個使用者介面)。</p>
		<p>Calculator的使用者介面。</p>
		<p>沒有輸入任何資料便進行計算，會出現錯誤訊息。</p>

		<p>進行 2 + 3 的計算。</p>
		<p>在功能選單中選擇 等於(=)。</p>
		<p>等於 5。(2+3 的計 算結果)</p>
		<p>將 5 存放到 Calculator 的記憶體 中。( 按下 MS )</p>



		<p>Calculator的狀態值為 <b>M</b>，表示記憶體中有資料被記憶著。</p>
		<p><math>-2 \times 2 = -4</math></p>
		<p>將 <math>-4</math> 加到 Calculator 的記憶體中。(按下 <b>M+</b>)</p>
		<p>觀看 Calculator 的記憶體中，目前的資料結果。(按下 <b>MR</b>)</p>

		<p>記憶體中的資料結果為 1。</p>
		<p>離開 Calculator。 ( 按下 Exit Cute Calculator )</p>
		<p>行動應用程式的結束畫面。</p>

### 8.1 結論

#### 8.1.1 達到行動應用程式僅需撰寫一次的目標

利用 PUML 和 PGML，加上 XSLT 的轉換機制，達到“行動應用程式僅需撰寫一次，就可以在各種行動執行環境中執行”的目標。

使用 PUML 和 PGML 在抽象層級中(abstract level)，分別描述一個行動應用程式的使用者介面，以及邏輯運算部份，然後再將此描述完成的行動應用程式透過 XSLT 的轉換機制，轉換成各種可以在行動執行環境中執行的程式碼。達到行動應用程式僅需撰寫一次，便可以在各種執行環境中執行的目標。(在本篇論文中，以實作轉換成 WML/WMLScript 程式碼和 J2ME MIDP 程式碼為主。)

#### 8.1.2 減少行動應用程式開發與維護的成本

行動應用程式開發與維護的成本能夠減少，原因有二：一為使用 XSLT 的轉換機制，另一個便是將一個完整的行動應用程式拆成兩部份做處理。

##### 1. 使用 XSLT 的轉換機制

因為 XSLT 轉換機制的使用，使得行動應用程式僅需撰寫一次，便可以在各種行動執行環境上執行。如此，行動應用程式的程式設計師便不再需要替不同的行動執行環境，撰寫不同版本但功能相同的行動應用程式了，大大地節省了開發行動應用程式所需的時間成本。

##### 2. 將行動應用程式拆成使用者介面和邏輯運算兩部份

因為將行動應用程式拆成兩部份來看待，如此，在撰寫行動應用程式時便可以進行責任區分：負責使用者介面部份的程式設計師，只要專心於使用者介面的撰寫即可，負責邏輯部份的程式設計師，只要專心於邏輯部份的撰寫即可。當需要互相溝通討論時，便可以對自己所負責的部份做詳盡的思考，以促進彼此的溝通協調。

因為將行動應用程式拆成兩部份，所以也使得責任區分能夠形成，如此，可以加快行動應用程式的撰寫速度，節省開發所需的時間成本，以及使得整體

行動應用程式的結構設計更加縝密。另外，在維護行動應用程式方面，因為責任區分的關係，使得維護變得較容易，節省維護成本。

### 8.1.3 PUML / PGML 提供了一個新的方向

#### 1. PUML 是一個為行動應用程式而生的使用者介面描述語言

由 PUML 的設計理念可以看出，PUML 吸取了 XUL、UIML 和 WML 的優點，並改進了在 XUL 和 UIML 中，包含了太多專為視窗系統設計的描述方法的缺點，還加上了平坦式的使用者介面結構的設計理念。在在地完全考量了行動應用程式在撰寫方面的限制，以及考量了行動裝置在顯示使用者介面上的限制。所以，PUML 是一個專為行動應用程式所設計的使用者介面描述語言。

#### 2. PGML 是一個為行動應用程式而生的邏輯運算描述語言

由於行動裝置本身能力的限制，例如：記憶體較小、CPU 能力較弱、不一定支援符點數運算.....等等，所以行動應用程式本身不能夠進行較複雜的邏輯運算。而 PGML 所提供的邏輯描述，都是行動裝置本身能力限制之內可以執行的邏輯運算，並透過 PGML Standard Library 來做到一些比較複雜的邏輯運算，而且 PGML Standard Library 還具備擴充的特性，以便滿足未來所需(例如：行動裝置的能力提升，可以做更複雜的邏輯運算)。所以，PGML 是一個專為行動應用程式所設計的邏輯運算描述語言。

### 8.1.4 轉換樣式(transformation pattern)是一個好的指導方針

在透過 XSLT 做轉換時，所依循的轉換樣式(transformation pattern)，對於以後要做其他程式碼的轉換，或是其他相關的研究，都是一個相當好的指導方針。從本論文提出的兩個轉換樣式 – WAP Transformation Pattern 和 J2ME Transformation Pattern 可以看出，這兩個轉換樣式都具有相當嚴謹的轉換規則與要求，以確保所轉換出來的行動應用程式，能夠在相對應的行動執行環境上正確無誤的執行。

轉換樣式的另一個好處是，規範良好的轉換規則，在要對轉換結果除錯(debug)時，也能達到事半功倍的效果。所以，本論文所提出的轉換樣式的想法，對於要做相關研究以及其他程式語言轉換，將會是一個好的指導方針。

## 8.2 未來發展

### 8.2.1 回顧過往

到目前為止，很多的研究文獻都對“轉換(transformation)”或是“適性化(adaptation)”這個議題相當有興趣，但大部份都侷限在網頁內容的轉換(或是稱為網頁內容的適性化)，例如將 HTML 網頁轉換為 WML 的頁面，或是 XHTML 的頁面型態。有的研究也專注在，如何將網頁內容中的資訊加以處理，以能夠美觀地呈現在螢幕較小的行動裝置上。有的研究則是把重點放在，當執行應用程式的環境不同時，應用程式的適性化，例如在瀏覽器上執行的話，以 HTML 的型式呈現 web-based 的應用程式，如果在 PC 上的話，則把應用程式轉換成 Java 的應用程式來執行。

從以往的文獻研究來看，在內容的轉換上，大部份的研究重點都放在 PC 瀏覽器上，對行動裝置相關方面的研究著墨甚少。在應用程式的轉換上，也大都以 PC 平台為主，也有行動裝置方面的研究，但所有的文獻研究對於邏輯運算部份的轉換交代不清，甚至沒有交代，主力重點也是放在使用者介面的轉換以及呈現。



### 8.2.2 開創未來

本論文所提出的 XML-based Mobile Application Development Kit 完全改正過去相關研究文獻的缺點以及不足，並完全針對行動應用程式提出新的思考方向。但畢竟是新的思維，整體架構雖然已經完備，但仍可以朝以下目標加以改進，讓 XML-based Mobile Application Development Kit 更加完整。

- 使用 IDE 介面的開發工具來撰寫 PUML / PGML，避免去直接撰寫 XML 文件。
- 根據以後需要，可以新增、刪除 PUML / PGML 中的 elements。
- 利用新增 PGML Standard Library 的方式來提升 PGML 的能力。例如：新增資料庫相關的 library functions，讓 PGML 可以連結資料庫，對資料庫做操作。
- 撰寫各種 XSLT style sheet，讓用 PUML / PGML 所撰寫出來的行動應用程式可以轉換成各種目的端程式碼。例如：XHTML、ASP.NET……等等。
- 利用 PUML 做前端使用者介面的描述，後端可搭配現存的，能力較強的邏輯描述語言，例如：Java、C++……等等，不限於使用 PGML。

- 將轉換機制提升到，不只可以轉換成各種適用於小螢幕的行動應用程式，也可以轉換成各種適用於不同螢幕大小的應用程式。也就是說，將本篇論文所提出的方法，提升到不僅僅是在於行動應用程式的撰寫與轉換而已，而能夠滿足所有應用程式的轉換需求，所轉換出來的應用程式可以跨越任何一種執行平台。



---

## Reference

---

### 研究文獻(Research Theses and Reports)

- [1] Nikola Mitrović and Eduardo Mena, "Adaptive User Interface for Mobile Devices", IIS Department, University of Zaragoza, Maria de Luna 3, 50018 Zaragoza, Spain
- [2] Johan Plomp, Robbie Schaefer, and Wolfgang Mueller, "Comparing Transcoding Tools for Use with a Generic User Interface Format", Extreme Markup Languages 2002 Conference
- [3] John Grundy and John Hosking, "Developing Adaptable User Interfaces for Component-based Systems", User Interface Conference 2000, AUIC 2000 (IEEE conference)
- [4] Allan Meng Krebs, Ivan Marsic and Bogdan Dorohonceanu, "Mobile Adaptive Application for Ubiquitous Collaboration in Heterogeneous Environments", Proceedings of the 22nd ICDCSW'02
- [5] Roland A. Merrick, "Device Independent User Interfaces in XML", IBM Corporation
- [6] Richard Berry, Scott Isensee, John Mullaly, and Dave Roberts, "OVID: an overview", © 1997-1998, IBM Corporation
- [7] Pedro Azevedo, Roland Merrick, and Dave Roberts, "OVID to AUIML - User-Oriented Interface Modeling", IBM UK Ltd
- [8] Didier Courtaud, "Basis of UIML", UIML Europe 2001 conference, 8 March 2001
- [9] Greg J. Badros, "JavaML: A Markup Language for Java Source Code", Dept. of Computer Science and Engineering, University of Washington
- [10] Angel Puerta and Jacob Eisenstein, "XIML: A Common Representation for Interaction Data", RedWhale Software
- [11] Neil Deakin, "XUL Tutorial"
- [12] 3GLAB TRIGENIX, "a Universal Graphical User Interface Engine for Mobile Devices"
- [13] 3GLAB TRIGENIX, "The Total Handset User Experience, A Technical White Paper", September 2002
- [14] Guido Menkhaus and Wolfgang Pree, "A Hybrid Approach to Adaptive User Interface Generation", Software Research Lab, Constance University
- [15] Adnana Al-bar, "A Survey to Adaptive Application in Mobile Computing", SoftSys talk COGS, April 4th 2001
- [16] Jacob Eisenstein, Jean Vanderdonckt, and Angel Puerta, "Adapting to Mobile Contexts with User-Interface Modeling", RedWhale Software Corporation

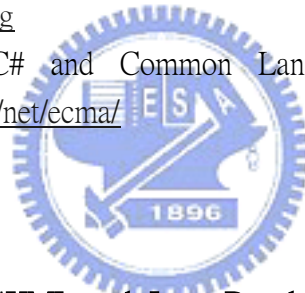
- [17] Jacob Eisenstein, Jean Vanderdonckt, and Angel Puerta, "**Applying Model-Based Techniques to the Development of UIs for Mobile Computers**", RedWhale Software Corporation
- [18] JEAN VANDERDONCKT, MURIELLE FLORINS, and FRÉDÉRIC OGER, "**Model-Based Design of Mobile User Interfaces**", Université catholique de Louvain
- [19] Mark Tierney, "**Adaptive Information Filtering for Mobile Users**", Human Computer Interaction and Language Engineering Laboratory, Swedish Institute of Computer Science
- [20] Angel R. Puerta, Eric Cheng, Tunhow Ou, and Justin Min, "**MOBILE: User-Centered Interface Building**", Stanford University
- [21] Andreas Müller, Peter Forbrig, and Clemens Cap, "**Model-Based User Interface Design Using Markup Concepts**", Department of Computer Science, University of Rostock
- [22] Robbie Schaefer and Wolfgang Mueller, "**Multimodal Interactive User Interfaces for Mobile Multi-Device Environments**", Paderborn University/C-LAB, Fuerstenallee 11 Paderborn, Germany
- [23] Mir Farooq Ali, Marc Abrams, and Manuel A. Pérez-Quiñones, "**Multi-Platform User Interface Construction with Transformations using UIML**"
- [24] Kris Luyten and Karin Coninx, "**An XML-based runtime user interface description language for mobile computing devices**", 2001 Springer-Verlag
- [25] Morris Sloman, "**Will Pervasive Computing be Manageable?**", OPENVIEW 2001: OPENVIEW USER'S CONFERENCE
- [26] Tangis Corporation, "**Context Aware Computing**", A Tangis white paper on the future of mobile/wireless computing
- [27] Anna Maria Jankowska, and Andrzej Dabkowski, "**Content Adaptation Tag Library - An Approach for User Interface Adaptation for Different Devices**", European University Viadrina, Chair of Business Informatics
- [28] Wataru Okada, Fumihiko Kato, Kazuhiro Kitagawa, and Tatsuya Hagino, "**Applying CC/PP to User's Environmental Information for Web Service Customization**"
- [29] 姚立三, "**An Adaptive Mobile Application Development Framework**", 國立交通大學, 碩士, 民國 92 年 6 月

#### 參考網站(Reference Web Sites)

- [30] IBM OVID, [http://www-3.ibm.com/ibm/easy/eou\\_ext.nsf/Publish/104](http://www-3.ibm.com/ibm/easy/eou_ext.nsf/Publish/104)
- [31] UIML 2.0a spec, <http://www.uiml.org/specs/docs/uiml20-17Jan00.pdf>, published on January 17, 2000
- [32] UIML.org, <http://www.uiml.org/index.php>



- [33] XForms 1.0 spec, <http://www.w3.org/TR/2002/CR-xforms-20021112/>, W3C candidate recommendation, 11/12/2002
- [34] eXtensible User Interface Language (XUL), <http://www.xulplanet.com/tutorials/xultu/>
- [35] eXtensible interface Markup Language (XIML), <http://www.ximl.org>
- [36] WAP Forum, OMA, Open Mobile Alliance, <http://www.wapforum.org/index.htm>
- [37] JavaML, <http://www.cs.washington.edu/homes/gjb/JavaML/>
- [38] Extreme Markup Languages Conference, <http://www.extrememarkup.com/extreme/>
- [39] W3C, <http://www.w3.org/>
- [40] XSL Transformations (XSLT) Version 1.0, <http://www.w3.org/TR/xslt>, W3C Recommendation 16 November 1999
- [41] XML Path Language (XPath) Version 1.0, <http://www.w3.org/TR/xpath>, W3C Recommendation 16 November 1999
- [42] Java Technology at Sun, <http://java.sun.com/>
- [43] Forum Nokia, <http://www.forum.nokia.com/main.html>
- [44] 3GLAB TRIGENIX, <http://www.trigenix.com/>
- [45] MexE forum, <http://www.MExEforum.org>
- [46] 3GPP, <http://www.3gpp.org>
- [47] ECMA and ISO/IEC C# and Common Language Infrastructure Standards, <http://msdn.microsoft.com/net/ecma/>



#### 參考書籍(Reference Books)

- [48] Hiroshi Maruyama etc., "XML and Java: Developing Web Applications, Second Edition"
- [49] Martin Frost, "Learning WML & WMLScript "
- [50] Chris Von See and Nitin Keskar, "XSLT Developer's Guide"

---

# Appendix

## Transformation Pattern Summary

---

### [WAP Transformation Pattern]

#### 1. 變數的命名法則

在轉換到 WML/WMLScript 時，要注意的是在 WAP 中，所有的變數均放置在 WMLBrowser context 中。而在 PUML/PGML 中可以宣告及使用變數，所以在轉換之後，必須確保 WMLBrowser context 中的所有變數名稱是獨一無二的，以免造成轉換後的 WAP 行動應用程式在執行時發生錯誤。

有兩種類型的變數，需要使用轉換後的變數命名法則：(1)PUML 中代表 UI 元件的變數 (2)PGML 中所使用的全域變數。第一種類型使用法則一，第二種類型使用法則二。

[法則一]

所在的 UI 容器(board 名稱)\_UI 元件名稱(變數名稱)\_UI 元件屬性[選擇性的]

[法則二]

所在的邏輯物件名稱\_全域變數名稱

#### 2. PUML 轉換到 WML

以下列出重要的轉換規則。

- 不會有<template>、<prev/>
- 產生五種類型的 card
  - (1) index card [必須]
  - (2) preSetting card [必須]
  - (3) inner card
  - (4) temp card
  - (5) general card
- index card

- WAP 行動應用程式第一張要執行的 card。
- 做 card 的導向。導到 inner card 串鍊，或是 preSetting card。
- preSetting card
  - 觸發 onenterforward event → 設定 UI 元件相關變數。
  - 觸發 ontimer event → 執行 WAP 行動應用程式真正的第一個 UI。
- inner card
  - PUML 中的每個邏輯物件皆對應一張 inner card。
  - 觸發 onenterforward event → 設定邏輯物件的全域變數(呼叫相對應的 globalVarDeclaration())。
  - 觸發 ontimer event → 執行下一張 inner card(形成 inner card 串鍊)，或執行 preSetting card。
- temp card
  - PUML 中，<action>的每一個 child element (<use-object>、<change>、<nextboard>)皆會對應到一張 temp card。
  - 觸發 onenterforward event → 執行相對應的邏輯運算。
  - 觸發 ontimer event → 執行下一張 temp card(形成 temp card 串鍊)。
- general card
  - PUML 中的每個<board>皆對應到一張一般 WML 中常見的 card。
  - PUML 中描述的 UI 元件轉換到 WML 中相對應的 UI 元件。
  - PUML 中的<action>轉換成 WML 中的<do>。



### 3. PGML 轉換到 WMLScript

以下列出重要的轉換規則。

- 利用 WMLBrowser.setVar()和 WMLBrowser.getVar()，模擬 PGML 中所指定的全域變數的存取。
- 產生兩個 interface (即每個產生的 WMLScript 檔都有以下兩個 function)
  - (1) *globalVarDeclaration()* → 設定該邏輯物件的全域變數。
  - (2) *change()* → 更改 UI 元件的狀態值。
- 額外產生 PGMLChange.wmls。
- 每個 WMLScript function 對應到一個 PGML 的<method>。PGML 所描述的邏輯運算，皆對應到適當的 WMLScript 的邏輯運算。

## [J2ME Transformation Pattern]

### 1. PUML 轉換到 J2ME MIDP

以下列出重要的轉換規則。

- 會產生一個 interface，兩種 class。
  - *BMngr\_Interface* interface
  - **BMngr class** (extends MIDlet, implements BMngr\_Interface)
  - **UI class** (extends Form, implements CommandListener)
- BMngr class
  - 轉換過程中產生的 class。extends MIDlet，implements BMngr\_Interface
  - 控制 UI 的瀏覽順序(透過實作 *BMngr\_Interface* interface 中的 *changeBoard()* method)。
  - 取得某 UI 元件，進行 UI 元件狀態值的修改(透過實作 *BMngr\_Interface* interface 中的 *getBoard()* method)。
  - *new* 出 PUML 文件中所指定的全域邏輯物件(global logic object)。
  - *new* 出 PUML 文件中，<board>所對應的 UI class 的 instance。
- UI class
  - PUML 文件中的每個<board>都會對應到一個 UI class。extends Forms, implements CommandListener
  - *new* 出<board>中的區域邏輯物件(local logic object)。
  - 代表 UI 元件的變數必須是 public。
  - PUML 中的<action>轉換成 command 物件。
  - 在 constructor 中要做的事有：對所有的 UI 元件做初始化的動作、設定 Command Listener，以及把 UI 元件和 command 加到 UI 呈現上。
  - event-driven 的邏輯運算，透過實作 *CommandListener* interface 中的 *commandAction()* method 來完成。

### 2. PGML 轉換到 J2ME MIDP

這部份的轉換規則較簡單。

- 每份 PGML 文件皆轉換到一個 public class，且 constructor 不需參數。
- PGML 所描述的邏輯運算，轉換到適當的 J2ME MIDP 的邏輯運算。

---

# Appendix

## Example 1 – Simple Demo

---

### [ Source PUML Code – DemoUI.puml ]

```
<?xml version="1.0"?>

<user-interface name="DemoUI" version="1.2" xmlns="http://dcsw3.cis.nctu.edu.tw/Project/Pervasive/PUML/">

  <logic-objects>
    <object name="userInfo" source="UserInfo.puml"/>
  </logic-objects>

  <board name="welcomeBoard" title="Simple Demo">
    <picture name="demo" source="demo.jpg" altText="Demo Image" align="center"/>
    <label name="hintMsg" showText="input bellow informaions: "/>
    <textnote name="userName" type="text" title="name: "/>
    <textnote name="userPassword" type="password" title="password: "/>

    <action name="enter" showText="Confirm">
      <use-object name="userInfo" method="setUserName">
        <param select="userName"/>
      </use-object>

      <use-object name="userInfo" method="setUserPassword">
        <param select="userPassword"/>
      </use-object>

      <change container="infoBoard" component="showName" update="showText">
        <use-object name="userInfo" method="getUserName"/>
      </change>

      <change container="infoBoard" component="showPasswd" update="showText">
        <use-object name="userInfo" method="getUserPassword"/>
      </change>

      <nextboard goto="infoBoard"/>
    </action>
  </board>

  <board name="infoBoard" title="User Information">
    <label name="helloMsg" showText="Hello, "/>
    <label name="showName" showText=""/>
    <label name="passwdMsg" showText="your password is: "/>
    <label name="showPasswd" showText=""/>
    <label name="infoMsg" showText="navigating following pages to choose your favorite fruit or colors..."/>

    <action name="singleFruit" showText="Choose a Fruit">
      <change container="fruitBoard" component="showName" update="showText">
        <use-object name="userInfo" method="getUserName"/>
      </change>

      <nextboard goto="fruitBoard"/>
    </action>

    <action name="multiColors" showText="Choose Colors">
      <change container="colorsBoard" component="showName" update="showText">
        <use-object name="userInfo" method="getUserName"/>
      </change>

      <nextboard goto="colorsBoard"/>
    </action>

    <action name="backWelcome" showText="Go to Start">
      <nextboard goto="welcomeBoard"/>
    </action>
  </board>
```

```

<board name="fruitBoard" title="Choosing A Fruit">
  <logic-objects>
    <object name="fruitInfo" source="FruitInfo.pgml"/>
  </logic-objects>

  <label name="showName" showText=""/>
  <label name="msg1" showText="likes"/>
  <label name="fruitMsg" showText=""/>
  <listpaper title="select one fruit." iname="iFruit" mode="single">
    <item showText="apple"/>
    <item showText="banana"/>
    <item showText="watermelon"/>
    <item showText="lemon"/>
    <item showText="strawberry"/>
    <item showText="mango"/>
  </listpaper>

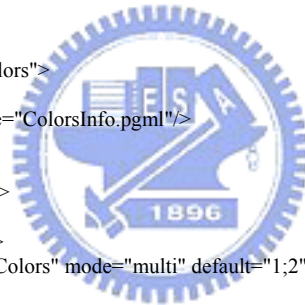
  <action name="fruitOK" showText="Select OK">
    <use-object name="fruitInfo" method="setChoosing">
      <param select="iFruit"/>
    </use-object>

    <change container="fruitBoard" component="fruitMsg" update="showText">
      <use-object name="fruitInfo" method="getChoosingInfo"/>
    </change>

    <nextboard goto="fruitBoard"/>
  </action>

  <action name="backInfo" showText="Go to Back">
    <nextboard goto="infoBoard"/>
  </action>
</board>

```



```

<board name="colorsBoard" title="Choosing Colors">
  <logic-objects>
    <object name="colorsInfo" source="ColorsInfo.pgml"/>
  </logic-objects>

  <label name="showName" showText=""/>
  <label name="msg1" showText="likes"/>
  <label name="colorsMsg" showText=""/>
  <listpaper title="select colors:" iname="iColors" mode="multi" default="1;2">
    <item showText="Red"/>
    <item showText="Blue"/>
    <item showText="Green"/>
    <item showText="Yellow"/>
    <item showText="White"/>
    <item showText="Black"/>
  </listpaper>

  <action name="colorsOK" showText="Select OK">
    <use-object name="colorsInfo" method="setChoosing">
      <param select="iColors"/>
    </use-object>

    <change container="colorsBoard" component="colorsMsg" update="showText">
      <use-object name="colorsInfo" method="getChoosingInfo"/>
    </change>

    <nextboard goto="colorsBoard"/>
  </action>

  <action name="backInfo" showText="Go to Back">
    <nextboard goto="infoBoard"/>
  </action>
</board>
</user-interface>

```

## [Source PGML Code – UserInfo.pgml]

```
<?xml version="1.0"?>
<object name="UserInfo" version="1.0" xmlns="http://dcsw3.cis.nctu.edu.tw/Project/Pervasive/PGML/">
  <declaration>
    <variable type="String" name="userName" value=""/>
    <variable type="String" name="userPassword" value=""/>
  </declaration>
  <method name="setUserName" visibility="public" return-type="void">
    <in>
      <variable type="String" name="user_name"/>
    </in>
    <action>
      <assign arg1="userName" arg2="user_name"/>
    </action>
  </method>
  <method name="setUserPassword" visibility="public" return-type="void">
    <in>
      <variable type="String" name="passwd"/>
    </in>
    <action>
      <assign arg1="userPassword" arg2="passwd"/>
    </action>
  </method>
  <method name="getUserName" visibility="public" return-type="String">
    <action>
      <return select="userName"/>
    </action>
  </method>
  <method name="getUserPassword" visibility="public" return-type="String">
    <action>
      <return select="userPassword"/>
    </action>
  </method>
</object>
```



## [Source PGML Code – FruitInfo.pgml]

```
<?xml version="1.0"?>
<object name="FruitInfo" version="1.0" xmlns="http://dcsw3.cis.nctu.edu.tw/Project/Pervasive/PGML/">
  <declaration>
    <variable type="int" name="choosed" value="-1"/>
  </declaration>
  <method name="setChoosing" visibility="public" return-type="void">
    <in>
      <variable type="int" name="index"/>
    </in>
    <action>
      <assign arg1="choosed" arg2="index"/>
    </action>
  </method>
  <method name="getChoosingInfo" visibility="public" return-type="String">
    <action>
      <if>
        <test>
          <equal-to>
            <operand select="choosed"/>
            <operand type="int" value="1"/>
          </equal-to>
        </test>
      </if>
    </action>
  </method>
</object>
```

```

<action>
<return type="String" value="apple"/>
</action>

<else>
<if>
  <test>
  <equal-to>
  <operand select="choosed"/>
  <operand type="int" value="2"/>
  </equal-to>
  </test>

  <action>
  <return type="String" value="banana"/>
  </action>

  <else>
  <if>
    <test>
    <equal-to>
    <operand select="choosed"/>
    <operand type="int" value="3"/>
    </equal-to>
    </test>

    <action>
    <return type="String" value="watermelon"/>
    </action>

    <else>
    <if>
      <test>
      <equal-to>
      <operand select="choosed"/>
      <operand type="int" value="4"/>
      </equal-to>
      </test>

      <action>
      <return type="String" value="lemon"/>
      </action>

      <else>
      <if>
        <test>
        <equal-to>
        <operand select="choosed"/>
        <operand type="int" value="5"/>
        </equal-to>
        </test>

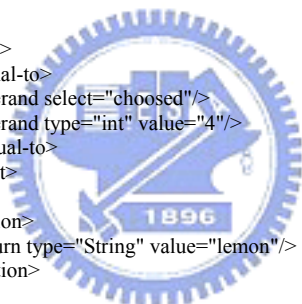
        <action>
        <return type="String" value="strawberry"/>
        </action>

        <else>
        <if>
          <test>
          <equal-to>
          <operand select="choosed"/>
          <operand type="int" value="6"/>
          </equal-to>
          </test>

          <action>
          <return type="String" value="mango"/>
          </action>

          <else>
          <return type="String" value=""/>
          </else>
        </if>
      </else>
    </if>
  </else>
</if>
</else>

```





```
                </if>
            </else>
        </if>
    </else>
</if>
</if>
</if>
</if>
</if>
</if>
</action>
</method>
</object>
```

**[Source PGML Code – ColorsInfo.pgml ]**

<?xml version="1.0" ?>

<object name="ColorsInfo" version="1.0" xmlns="http://dcsw3.cis.nctu.edu.tw/Project/Pervasive/PGML/">

```
<declaration>
    <variable type="String" name="choosed" value="" />
</declaration>

<method name="setChoosing"  visibility="public" return-type="void">
    <in>
        <variable type="String" name="selectionList" />
    </in>
```

<action>

```
    <assign arg1="choosed">
        <operand type="String" value="" />
    </assign>
```

<for>

```
    <init>
        <variable type="int" name="index" value="0" />
    </init>
```

<test>

```
    <lessor-or-equal>
        <operand select="index" />
        <lib call="pgml.String.length">
            <operand select="selectionList" />
        </lib>
    </lessor-or-equal>
</test>
```

<step>

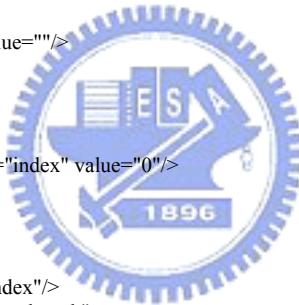
```
    <add result="index">
        <operand select="index" />
        <operand type="int" value="2" />
    </add>
</step>
```

<action>

```
    <assign arg1="choosed">
        <lib call="pgml.String.concat">
            <operand select="choosed" />
            <call-method name="getColor">
                <lib call="pgml.String.parseInt">
                    <lib call="pgml.String.charAt">
                        <operand select="selectionList" />
                        <operand select="index" />
                    </lib>
                </lib>
            </call-method>
        </lib>
```

```
    </assign>
```

```
    <assign arg1="choosed">
        <lib call="pgml.String.concat">
            <operand select="choosed" />
```



```

        <operand type="String" value=", "/>
      </lib>
    </assign>
  </action>
</for>
</action>
</method>

<method name="getChoosingInfo" visibility="public" return-type="String">
  <action>
    <return select="chosen"/>
  </action>
</method>

<method name="getColor" visibility="private" return-type="String">
  <in>
    <variable type="int" name="colorIndex"/>
  </in>

  <action>
    <if>
      <test>
        <equal-to>
          <operand select="colorIndex"/>
          <operand type="int" value="1"/>
        </equal-to>
      </test>

      <action>
        <return type="String" value="Red"/>
      </action>

    <else>
      <if>
        <test>
          <equal-to>
            <operand select="colorIndex"/>
            <operand type="int" value="2"/>
          </equal-to>
        </test>

        <action>
          <return type="String" value="Blue"/>
        </action>

      <else>
        <if>
          <test>
            <equal-to>
              <operand select="colorIndex"/>
              <operand type="int" value="3"/>
            </equal-to>
          </test>

          <action>
            <return type="String" value="Green"/>
          </action>

        <else>
          <if>
            <test>
              <equal-to>
                <operand select="colorIndex"/>
                <operand type="int" value="4"/>
              </equal-to>
            </test>

            <action>
              <return type="String" value="Yellow"/>
            </action>

          <else>
            <if>
              <test>
                <equal-to>

```





```

        <setvar name="infoBoard_infoMsg_showText" value="navigating following pages to choose your favorite fruit or colors..."/>
        <setvar name="fruitBoard_showName_showText" value=""/>
        <setvar name="fruitBoard_msg1_showText" value="likes"/>
        <setvar name="fruitBoard_fruitMsg_showText" value=""/>
        <setvar name="fruitBoard_iFruit_title" value="select one fruit:"/>
        <setvar name="fruitBoard_iFruit_item1" value="apple"/>
        <setvar name="fruitBoard_iFruit_item2" value="banana"/>
        <setvar name="fruitBoard_iFruit_item3" value="watermelon"/>
        <setvar name="fruitBoard_iFruit_item4" value="lemon"/>
        <setvar name="fruitBoard_iFruit_item5" value="strawberry"/>
        <setvar name="fruitBoard_iFruit_item6" value="mango"/>
        <setvar name="colorsBoard_showName_showText" value=""/>
        <setvar name="colorsBoard_msg1_showText" value="likes"/>
        <setvar name="colorsBoard_colorsMsg_showText" value=""/>
        <setvar name="colorsBoard_iColors_title" value="select colors:"/>
        <setvar name="colorsBoard_iColors_item1" value="Red"/>
        <setvar name="colorsBoard_iColors_item2" value="Blue"/>
        <setvar name="colorsBoard_iColors_item3" value="Green"/>
        <setvar name="colorsBoard_iColors_item4" value="Yellow"/>
        <setvar name="colorsBoard_iColors_item5" value="White"/>
        <setvar name="colorsBoard_iColors_item6" value="Black"/>
    </refresh>
    </onevent>
    <timer value="1"/>
</card>

```

```

<card id="welcomeBoard" title="Simple Demo">
    <p></p>
    <p><b>$(welcomeBoard_hintMsg_showText)</b></p>
    <p>name: <input name="welcomeBoard_userName" type="text"/></p>
    <p>password: <input name="welcomeBoard_userPassword" type="password"/></p>

```

```

        <do type="enter" label="Confirm">
            <go href="#welcomeBoard_enter_temp1"/>
        </do>
</card>

```

```

<card id="welcomeBoard_enter_temp1" onenterforward="UserInfo.wmls#setUserName('${welcomeBoard_userName}')
ontimer="#welcomeBoard_enter_temp2">
    <timer value="1"/>
</card>

```

```

<card id="welcomeBoard_enter_temp2" onenterforward="UserInfo.wmls#setUserPassword('${welcomeBoard_userPassword}')
ontimer="#welcomeBoard_enter_temp3">
    <timer value="1"/>
</card>

```

```

<card id="welcomeBoard_enter_temp3" onenterforward="UserInfo.wmls#change('infoBoard_showName_showText', 'getUserName')"
ontimer="#welcomeBoard_enter_temp4">
    <timer value="1"/>
</card>

```

```

<card id="welcomeBoard_enter_temp4" onenterforward="UserInfo.wmls#change('infoBoard_showPasswd_showText', 'getUserPassword')"
ontimer="#welcomeBoard_enter_temp5">
    <timer value="1"/>
</card>

```

```

<card id="welcomeBoard_enter_temp5" onenterforward="#infoBoard"/>

```

```

<card id="infoBoard" title="User Information">
    <p><b>$(infoBoard_helloMsg_showText)</b></p>
    <p><b>$(infoBoard_showName_showText)</b></p>
    <p><b>$(infoBoard_passwdMsg_showText)</b></p>
    <p><b>$(infoBoard_showPasswd_showText)</b></p>
    <p><b>$(infoBoard_infoMsg_showText)</b></p>

```

```

        <do type="singleFruit" label="Choose a Fruit">
            <go href="#infoBoard_singleFruit_temp1"/>
        </do>

```

```

        <do type="multiColors" label="Choose Colors">
            <go href="#infoBoard_multiColors_temp1"/>
        </do>

```

```

        <do type="backWelcome" label="Go to Start">

```



```

        <go href="#infoBoard_backWelcome_temp1"/>
    </do>
</card>

<card id="infoBoard_singleFruit_temp1" onenterforward="UserInfo.wmls#change('fruitBoard_showName_showText', 'getUserName')"
ontimer="#infoBoard_singleFruit_temp2">
    <timer value="1"/>
</card>

<card id="infoBoard_singleFruit_temp2" onenterforward="#fruitBoard"/>

<card id="infoBoard_multiColors_temp1" onenterforward="UserInfo.wmls#change('colorsBoard_showName_showText', 'getUserName')"
ontimer="#infoBoard_multiColors_temp2">
    <timer value="1"/>
</card>

<card id="infoBoard_multiColors_temp2" onenterforward="#colorsBoard"/>

<card id="infoBoard_backWelcome_temp1" onenterforward="#welcomeBoard"/>

<card id="fruitBoard" title="Choosing A Fruit">
    <p><b>$(fruitBoard_showName_showText)</b></p>
    <p><b>$(fruitBoard_msg1_showText)</b></p>
    <p><b>$(fruitBoard_fruitMsg_showText)</b></p>
    <p>select one fruit:<select iname="fruitBoard_iFruit" multiple="false">
        <option>$(fruitBoard_iFruit_item1)</option>
        <option>$(fruitBoard_iFruit_item2)</option>
        <option>$(fruitBoard_iFruit_item3)</option>
        <option>$(fruitBoard_iFruit_item4)</option>
        <option>$(fruitBoard_iFruit_item5)</option>
        <option>$(fruitBoard_iFruit_item6)</option>
    </select>
    <p>

    <do type="fruitOK" label="Select OK">
        <go href="#fruitBoard_fruitOK_temp1"/>
    </do>

    <do type="backInfo" label="Go to Back">
        <go href="#fruitBoard_backInfo_temp1"/>
    </do>
</card>

<card id="fruitBoard_fruitOK_temp1" onenterforward="FruitInfo.wmls#setChoosing('$(fruitBoard_iFruit)')" ontimer="#fruitBoard_fruitOK_temp2">
    <timer value="1"/>
</card>

<card id="fruitBoard_fruitOK_temp2" onenterforward="FruitInfo.wmls#change('fruitBoard_fruitMsg_showText', 'getChoosingInfo')"
ontimer="#fruitBoard_fruitOK_temp3">
    <timer value="1"/>
</card>

<card id="fruitBoard_fruitOK_temp3" onenterforward="#fruitBoard"/>

<card id="fruitBoard_backInfo_temp1" onenterforward="#infoBoard"/>

<card id="colorsBoard" title="Choosing Colors">
    <p><b>$(colorsBoard_showName_showText)</b></p>
    <p><b>$(colorsBoard_msg1_showText)</b></p>
    <p><b>$(colorsBoard_colorsMsg_showText)</b></p>
    <p>select colors:<select iname="colorsBoard_iColors" ivalue="1;2" multiple="true">
        <option>$(colorsBoard_iColors_item1)</option>
        <option>$(colorsBoard_iColors_item2)</option>
        <option>$(colorsBoard_iColors_item3)</option>
        <option>$(colorsBoard_iColors_item4)</option>
        <option>$(colorsBoard_iColors_item5)</option>
        <option>$(colorsBoard_iColors_item6)</option>
    </select>
    <p>

    <do type="colorsOK" label="Select OK">
        <go href="#colorsBoard_colorsOK_temp1"/>
    </do>

    <do type="backInfo" label="Go to Back">

```



```

        <go href="#colorsBoard_backInfo_temp1"/>
    </do>
</card>

<card id="colorsBoard_colorsOK_temp1" onenterforward="ColorsInfo.wmls#setChoosing('${colorsBoard_iColors}')"
ontimer="#colorsBoard_colorsOK_temp2">
    <timer value="1"/>
</card>

<card id="colorsBoard_colorsOK_temp2" onenterforward="ColorsInfo.wmls#change('colorsBoard_colorsMsg_showText', 'getChoosingInfo')"
ontimer="#colorsBoard_colorsOK_temp3">
    <timer value="1"/>
</card>

<card id="colorsBoard_colorsOK_temp3" onenterforward="#colorsBoard"/>

<card id="colorsBoard_backInfo_temp1" onenterforward="#infoBoard"/>
</wml>

```

### [Transformed WAP Code – UserInfo.wmls (indented) ]

```

/*some pragma are declared here...*/

/*WMLScript interface to set declared global variables*/
extern function globalVarDeclaration() {
    /*setting all global variables in this logic object*/
    WMLBrowser.setVar("UserInfo_userName", "");
    WMLBrowser.setVar("UserInfo_userPassword", "");
}

/*all methods are here...*/
extern function setUsername (user_name){
    WMLBrowser.setVar("UserInfo_userName", user_name );
}

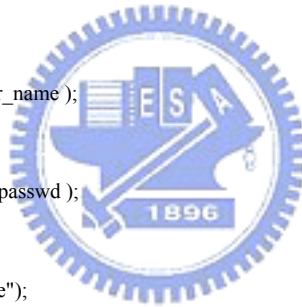
extern function setPassword (passwd){
    WMLBrowser.setVar("UserInfo_userPassword", passwd );
}

extern function getUsername (){
    return WMLBrowser.getVar("UserInfo_userName");
}

extern function getPassword (){
    return WMLBrowser.getVar("UserInfo_userPassword");
}

/*WMLScript interface to change value of UI component on some WML card*/
extern function change(component, newValue, callMethod){
    if(callMethod == ""){
        WMLBrowser.setVar(component, newValue);
    }
    else{
        if(callMethod == "getUsername"){
            WMLBrowser.setVar(component, getUsername());
        }
        if(callMethod == "getPassword"){
            WMLBrowser.setVar(component, getPassword());
        }
    }
}

```



### [Transformed WAP Code – FruitInfo.wmls (indented) ]

```

/*some pragma are declared here...*/

/*WMLScript interface to set declared global variables*/
extern function globalVarDeclaration() {
    /*setting all global variables in this logic object*/

```

```

        WMLBrowser.setVar("FruitInfo_choosed", -1);
    }

    /*all methods are here...*/
    extern function setChoosing (index){
        WMLBrowser.setVar("FruitInfo_choosed", index );
    }

    extern function getChoosingInfo (){
        if(WMLBrowser.getVar("FruitInfo_choosed")==1){
            return "apple";
        }
        else{
            if(WMLBrowser.getVar("FruitInfo_choosed")==2){
                return "banana";
            }
            else{
                if(WMLBrowser.getVar("FruitInfo_choosed")==3){
                    return "watermelon";
                }
                else{
                    if(WMLBrowser.getVar("FruitInfo_choosed")==4){
                        return "lemon";
                    }
                    else{
                        if(WMLBrowser.getVar("FruitInfo_choosed")==5){
                            return "strawberry";
                        }
                        else{
                            if(WMLBrowser.getVar("FruitInfo_choosed")==6){
                                return "mango";
                            }
                            else{
                                return "";
                            }
                        }
                    }
                }
            }
        }
    }
}

/*WMLScript interface to change value of UI component on some WML card*/
extern function change(component, newValue, callMethod){
    if(callMethod == ""){
        WMLBrowser.setVar(component, newValue);
    }
    else{
        if(callMethod == "getChoosingInfo"){
            WMLBrowser.setVar(component, getChoosingInfo());
        }
    }
}

```



### [Transformed WAP Code – ColorsInfo.wmls (indented) ]

```

/*some pragma are declared here...*/

/*WMLScript interface to set declared global variables*/
extern function globalVarDeclaration() {
    /*setting all global variables in this logic object*/
    WMLBrowser.setVar("ColorsInfo_choosed", "");
}

/*all methods are here...*/
extern function setChoosing (selectionList){
    WMLBrowser.setVar("ColorsInfo_choosed", "");

    for(var index=0; index<=String.length(selectionList); index=( Lang.parseInt(index) + 2 )){

```





## [Transformed J2ME Code – BMngr\_DemoUI.java (indented)]

```
package simpledemo;
import javax.microedition.midlet.*;
import javax.microedition.lcdui.*;

interface BMngr_Interface {
    public void changeBoard(String boardName);
    public Displayable getBoard(String boardName);
}

public class BMngr_DemoUI extends MIDlet implements BMngr_Interface{
    public static BMngr_DemoUI instance;
    private Display display;

    /*global logic objects*/
    public UserInfo lgObj_userInfo = new UserInfo();

    /*all displayables corresponding to boards' declaration*/
    private B_welcomeBoard welcomeBoard = new B_welcomeBoard();
    private B_infoBoard infoBoard = new B_infoBoard();
    private B_fruitBoard fruitBoard = new B_fruitBoard();
    private B_colorsBoard colorsBoard = new B_colorsBoard();

    /*Constructor*/
    public BMngr_DemoUI(){
        instance = this;
    }

    /*Main method*/
    public void startApp(){
        display = Display.getDisplay(this);
        display.setCurrent(welcomeBoard);
    }

    /*Handle pausing the MIDlet*/
    public void pauseApp(){
    }

    /*Handle destroying the MIDlet*/
    public void destroyApp(boolean unconditional){
    }

    /*Quit the MIDlet*/
    public static void quitApp(){
        instance.destroyApp(true);
        instance.notifyDestroyed();
        instance = null;
    }

    /*change UI presentation*/
    public void changeBoard(String boardName){
        if(boardName.compareTo("welcomeBoard") == 0){
            display.setCurrent(welcomeBoard);
        }
        if(boardName.compareTo("infoBoard") == 0){
            display.setCurrent(infoBoard);
        }
        if(boardName.compareTo("fruitBoard") == 0){
            display.setCurrent(fruitBoard);
        }
        if(boardName.compareTo("colorsBoard") == 0){
            display.setCurrent(colorsBoard);
        }
    }

    /*get particular UI container*/
    public Displayable getBoard(String boardName){
        Displayable displayable = null;

        if(boardName.compareTo("welcomeBoard") == 0){
            displayable = welcomeBoard;
        }
    }
}
```



```

        if(boardName.compareTo("infoBoard") == 0){
            displayable = infoBoard;
        }
        if(boardName.compareTo("fruitBoard") == 0){
            displayable = fruitBoard;
        }
        if(boardName.compareTo("colorsBoard") == 0){
            displayable = colorsBoard;
        }

        return displayable;
    }
}

```

```

class B_welcomeBoard extends Form implements CommandListener {
    /*local logic objects*/

    /*UI components*/
    public ImageItem demo;
    public StringItem hintMsg;
    public TextField userName;
    public TextField userPassword;

    /*commands*/
    private Command enter = new Command("Confirm", Command.SCREEN, 1);

    /*Constructor*/
    public B_welcomeBoard() {
        /* initialize UI component */
        super("Simple Demo");
        try{
            demo = new ImageItem("", Image.createImage("/demo.png"), ImageItem.LAYOUT_CENTER, "Demo Image");
        } catch(Exception e){e.printStackTrace();}
        hintMsg = new StringItem("input bellow informaions: ", "");
        userName = new TextField("name: ", "", 50, TextField.ANY);
        userPassword = new TextField("password: ", "", 50, TextField.PASSWORD);

        /* set up listeners */
        this.setCommandListener(this);

        /* append UI component */
        this.append(demo);
        this.append(hintMsg);
        this.append(userName);
        this.append(userPassword);

        /* add commands */
        this.addCommand(enter);
    }

    /*Handle command events*/
    public void commandAction(Command command, Displayable displayable){
        String selectedStr = null;
        String oldPath = null;
        String newPath = null;
        String align = null;

        if(command == enter){
            BMngr_DemoUI.instance.lgObj_userInfo.setUserName(userName.getString());
            BMngr_DemoUI.instance.lgObj_userInfo.setUserPassword(userPassword.getString());
            ((B_infoBoard)BMngr_DemoUI.instance.getBoard("infoBoard")).showName.
                setLabel(BMngr_DemoUI.instance.lgObj_userInfo.getUserName());
            ((B_infoBoard)BMngr_DemoUI.instance.getBoard("infoBoard")).showPasswd.
                setLabel(BMngr_DemoUI.instance.lgObj_userInfo.getUserPassword());
            BMngr_DemoUI.instance.changeBoard("infoBoard");
        }
    }
}

```



```

class B_infoBoard extends Form implements CommandListener {
    /*local logic objects*/

    /*UI components*/

```

```

public StringItem helloMsg;
public StringItem showName;
public StringItem passwdMsg;
public StringItem showPasswd;
public StringItem infoMsg;

/*commands*/
private Command singleFruit = new Command("Choose a Fruit", Command.SCREEN, 1);
private Command multiColors = new Command("Choose Colors", Command.SCREEN, 1);
private Command backWelcome = new Command("Go to Start", Command.SCREEN, 1);

/*Constructor*/
public B_infoBoard() {
    /* initialize UI component */
    super("User Information");
    helloMsg = new StringItem("Hello, ", "");
    showName = new StringItem("", "");
    passwdMsg = new StringItem("your password is: ", "");
    showPasswd = new StringItem("", "");
    infoMsg = new StringItem("navigating following pages to choose your favorite fruit or colors...", "");

    /* set up listeners */
    this.setCommandListener(this);

    /* append UI component */
    this.append(helloMsg);
    this.append(showName);
    this.append(passwdMsg);
    this.append(showPasswd);
    this.append(infoMsg);

    /* add commands */
    this.addCommand(singleFruit);
    this.addCommand(multiColors);
    this.addCommand(backWelcome);
}

```



```

/*Handle command events*/
public void commandAction(Command command, Displayable displayable){
    String selectedStr = null;
    String oldPath = null;
    String newPath = null;
    String align = null;

    if(command == singleFruit){
        ((B_fruitBoard)BMngr_DemoUI.instance.getBoard("fruitBoard")).showName.
            setLabel(BMngr_DemoUI.instance.lgObj_userInfo.getUserName());
        BMngr_DemoUI.instance.changeBoard("fruitBoard");
    }
    if(command == multiColors){
        ((B_colorsBoard)BMngr_DemoUI.instance.getBoard("colorsBoard")).showName.
            setLabel(BMngr_DemoUI.instance.lgObj_userInfo.getUserName());
        BMngr_DemoUI.instance.changeBoard("colorsBoard");
    }
    if(command == backWelcome){
        BMngr_DemoUI.instance.changeBoard("welcomeBoard");
    }
}
}

```

```

class B_fruitBoard extends Form implements CommandListener {
    /*local logic objects*/
    private FruitInfo lgObj_fruitInfo = new FruitInfo();

    /*UI components*/
    public StringItem showName;
    public StringItem msg1;
    public StringItem fruitMsg;
    public ChoiceGroup iFruit;

    /*commands*/
    private Command fruitOK = new Command("Select OK", Command.SCREEN, 1);
    private Command backInfo = new Command("Go to Back", Command.SCREEN, 1);
}

```

```

/*Constructor*/
public B_fruitBoard() {
    /* initialize UI component */
    super("Choosing A Fruit");
    showName = new StringItem("", "");
    msg1 = new StringItem("likes", "");
    fruitMsg = new StringItem("", "");
    iFruit = new ChoiceGroup("select one fruit:", ChoiceGroup.EXCLUSIVE);
    iFruit.append("apple", null);
    iFruit.append("banana", null);
    iFruit.append("watermelon", null);
    iFruit.append("lemon", null);
    iFruit.append("strawberry", null);
    iFruit.append("mango", null);

    /* set up listeners */
    this.setCommandListener(this);

    /* append UI component */
    this.append(showName);
    this.append(msg1);
    this.append(fruitMsg);
    this.append(iFruit);

    /* add commands */
    this.addCommand(fruitOK);
    this.addCommand(backInfo);
}

/*Handle command events*/
public void commandAction(Command command, Displayable displayable){
    String selectedStr = null;
    String oldPath = null;
    String newPath = null;
    String align = null;

    if(command == fruitOK){
        lgObj_fruitInfo.setChoosing(iFruit.getSelectedIndex()+1);
        ((B_fruitBoard)BMngr_DemoUI.instance.getBoard("fruitBoard")).fruitMsg.
        setLabel(lgObj_fruitInfo.getChoosingInfo());
        BMngr_DemoUI.instance.changeBoard("fruitBoard");
    }
    if(command == backInfo){
        BMngr_DemoUI.instance.changeBoard("infoBoard");
    }
}
}

```



```

class B_colorsBoard extends Form implements CommandListener {
    /*local logic objects*/
    private ColorsInfo lgObj_colorsInfo = new ColorsInfo();

    /*UI components*/
    public StringItem showName;
    public StringItem msg1;
    public StringItem colorsMsg;
    public ChoiceGroup iColors;

    /*commands*/
    private Command colorsOK = new Command("Select OK", Command.SCREEN, 1);
    private Command backInfo = new Command("Go to Back", Command.SCREEN, 1);

    /*Constructor*/
    public B_colorsBoard() {
        /* initialize UI component */
        super("Choosing Colors");
        showName = new StringItem("", "");
        msg1 = new StringItem("likes", "");
        colorsMsg = new StringItem("", "");
        iColors = new ChoiceGroup("select colors:", ChoiceGroup.MULTIPLE);
        iColors.append("Red", null);
        iColors.append("Blue", null);
        iColors.append("Green", null);
        iColors.append("Yellow", null);
    }
}

```

```

iColors.append("White", null);
iColors.append("Black", null);

String defaultSelected = "1;2";
for(int index=0; index <= defaultSelected.length(); index=index+2){
    String str = String.valueOf(defaultSelected.charAt(index));
    iColors.setSelectedIndex(Integer.parseInt(str)-1, true);
}

/* set up listeners */
this.setCommandListener(this);

/* append UI component */
this.append(showName);
this.append(msg1);
this.append(colorsMsg);
this.append(iColors);

/* add commands */
this.addCommand(colorsOK);
this.addCommand(backInfo);
}

/*Handle command events*/
public void commandAction(Command command, Displayable displayable){
    String selectedStr = null;
    String oldPath = null;
    String newPath = null;
    String align = null;

    if(command == colorsOK){
        selectedStr = "";
        for(int i=0; i < iColors.size(); i++){
            if(iColors.isSelected(i)){
                if(selectedStr != ""){
                    selectedStr = selectedStr + ";" + Integer.toString(i+1);
                }
                else{
                    selectedStr = selectedStr + Integer.toString(i+1);
                }
            }
        }

        lgObj_colorsInfo.setChoosing(selectedStr); ((B_colorsBoard)BMngr_DemoUI.instance.getBoard("colorsBoard")).colorsMsg.
            setLabel(lgObj_colorsInfo.getChoosingInfo());
        BMngr_DemoUI.instance.changeBoard("colorsBoard");
    }

    if(command == backInfo){
        BMngr_DemoUI.instance.changeBoard("infoBoard");
    }
}
}

```



### [Transformed J2ME Code – **UserInfo.java** (indented)]

```

package simpledemo;
/*J2ME "import" declarations*/

public class UserInfo {
    /*global variables' declaration*/
    private String userName="";
    private String userPassword="";

    /*constructor*/
    public UserInfo(){}

    /*all methods are here...*/
    public void setUserName( String user_name ){
        /*local variables' declaration*/

        this.userName=user_name;
    }
}

```

```

}

public void setUserPassword( String passwd ){
    /*local variables' declaration*/

    this.userPassword=passwd;
}

public String getUserName( ){
    /*local variables' declaration*/

    return this.userName;
}

public String getUserPassword( ){
    /*local variables' declaration*/

    return this.userPassword;
}
}

```

### [Transformed J2ME Code – FruitInfo.java (indented)]

```

package simpledemo;
/*J2ME "import" declarations*/

public class FruitInfo {
    /*global variables' declaration*/
    private int choosed=-1;

    /*constructor*/
    public FruitInfo(){

    /*all methods are here...*/
    public void setChoosing( int index ){
        /*local variables' declaration*/

        this.choosed=index;
    }

    public String getChoosingInfo( ){
        /*local variables' declaration*/

        if(this.choosed==1){
            return "apple";
        }
        else{
            if(this.choosed==2){
                return "banana";
            }
            else{
                if(this.choosed==3){
                    return "watermelon";
                }
                else{
                    if(this.choosed==4){
                        return "lemon";
                    }
                    else{
                        if(this.choosed==5){
                            return "strawberry";
                        }
                        else{
                            if(this.choosed==6){
                                return "mango";
                            }
                            else{
                                return "";
                            }
                        }
                    }
                }
            }
        }
    }
}

```



```
    }
  }
}
```

### [Transformed J2ME Code – ColorsInfo.java (indented)]

```
package simpledemo;
/*J2ME "import" declarations*/

public class ColorsInfo {
  /*global variables' declaration*/
  private String choosed="";

  /*constructor*/
  public ColorsInfo(){

  /*all methods are here...*/
  public void setChoosing( String selectionList ){
    /*local variables' declaration*/

    this.choosed="";

    for(int index=0; index<=selectionList.length(); index=( index + 2 )){
      this.choosed=this.choosed.concat(getColor(java.lang.Integer.
        parseInt(String.valueOf(selectionList.charAt(index)))));
      this.choosed=this.choosed.concat(", ");
    }
  }

  public String getChoosingInfo( ){
    /*local variables' declaration*/

    return this.choosed;
  }

  private String getColor( int colorIndex ){
    /*local variables' declaration*/

    if(colorIndex==1){
      return "Red";
    }
    else{
      if(colorIndex==2){
        return "Blue";
      }
      else{
        if(colorIndex==3){
          return "Green";
        }
        else{
          if(colorIndex==4){
            return "Yellow";
          }
          else{
            if(colorIndex==5){
              return "White";
            }
            else{
              if(colorIndex==6){
                return "Black";
              }
              else{
                return "";
              }
            }
          }
        }
      }
    }
  }
}

}
```



---

# Appendix

## Example 2 – Calculator

---

### [ Source PUML Code – CalculatorUI.puml ]

```
<?xml version="1.0"?>

<user-interface name="CalculatorUI" version="1.2" xmlns="http://dcs.w3.cis.nctu.edu.tw/Project/Pervasive/PUML/">

  <logic-objects>
    <object name="memoryCal" source="MemoryCal.pgml"/>
  </logic-objects>

  <board name="welcomeBoard" title="Cute Calculator">
    <label name="msg1" showText="Functions Detail: "/>
    <label name="msg2" showText="[ = ] -> computation result"/>
    <label name="msg3" showText="[ C ] -> cancel result"/>
    <label name="msg4" showText="[ +/- ] -> invert sign"/>
    <label name="msg5" showText="[ M+ ] -> memory added"/>
    <label name="msg6" showText="[ MR ] -> memory record"/>
    <label name="msg7" showText="[ MS ] -> memory saved"/>
    <label name="msg8" showText="[ MC ] -> memory canceled"/>

    <action name="runCal" showText="Run Calculator">
      <nextboard goto="calculatorBoard"/>
    </action>
  </board>

  <board name="calculatorBoard" title="Cute Calculator">
    <logic-objects>
      <object name="generalCal" source="GeneralCal.pgml"/>
    </logic-objects>

    <label name="resultMsg" showText="Computation Result:"/>
    <label name="calResult" showText="0"/>
    <label name="lineMsg" showText="-----"/>

    <textnote name="operand1" type="text"/>
    <listpaper iname="iOP" mode="single" default="1">
      <item showText=" + "/>
      <item showText=" - "/>
      <item showText=" x "/>
      <item showText=" / "/>
      <item showText=" % "/>
    </listpaper>
    <textnote name="operand2" type="text"/>
    <label name="stateMsg" showText="state: "/>
    <label name="errorMsg" showText=""/>

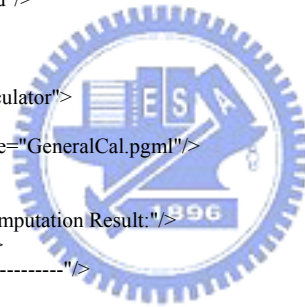
    <action name="equalCal" showText=" = ">
      <use-object name="generalCal" method="doOP">
        <param select="operand1"/>
        <param select="iOP"/>
        <param select="operand2"/>
      </use-object>

      <change container="calculatorBoard" component="calResult" update="showText">
        <use-object name="generalCal" method="getCalResult"/>
      </change>

      <change container="calculatorBoard" component="operand1" update="value">
        <use-object name="generalCal" method="getCalResult"/>
      </change>

      <change container="calculatorBoard" component="operand2" update="value"/>

      <change container="calculatorBoard" component="errorMsg" update="showText">
        <use-object name="generalCal" method="getErrorMsg"/>
      </change>
    </action>
  </board>
</user-interface>
```





```

    </change>

    <nextboard goto="calculatorBoard"/>
</action>

<action name="invertCal" showText=" +/- ">
  <use-object name="generalCal" method="doInvert">
    <param select="operand1"/>
  </use-object>

  <change container="calculatorBoard" component="calResult" update="showText">
    <use-object name="generalCal" method="getCalResult"/>
  </change>

  <change container="calculatorBoard" component="operand1" update="value">
    <use-object name="generalCal" method="getCalResult"/>
  </change>

  <change container="calculatorBoard" component="operand2" update="value"/>

  <change container="calculatorBoard" component="errorMsg" update="showText">
    <use-object name="generalCal" method="getErrorMsg"/>
  </change>

  <nextboard goto="calculatorBoard"/>
</action>

<action name="mplusCal" showText=" M+ ">
  <use-object name="memoryCal" method="doMplus">
    <param select="calResult"/>
  </use-object>

  <change container="calculatorBoard" component="calResult" update="showText">
    <use-object name="generalCal" method="doCancel"/>
  </change>

  <change container="calculatorBoard" component="operand1" update="value"/>

  <change container="calculatorBoard" component="operand2" update="value"/>

  <change container="calculatorBoard" component="stateMsg" update="showText">
    <use-object name="memoryCal" method="getMState"/>
  </change>

  <nextboard goto="calculatorBoard"/>
</action>

<action name="mrCal" showText=" MR ">
  <change container="calculatorBoard" component="calResult" update="showText">
    <use-object name="memoryCal" method="doMR"/>
  </change>

  <change container="calculatorBoard" component="operand1" update="value"/>

  <change container="calculatorBoard" component="operand2" update="value"/>

  <change container="calculatorBoard" component="stateMsg" update="showText">
    <use-object name="memoryCal" method="getMState"/>
  </change>

  <nextboard goto="calculatorBoard"/>
</action>

<action name="msCal" showText=" MS ">
  <use-object name="memoryCal" method="doMS">
    <param select="calResult"/>
  </use-object>

  <change container="calculatorBoard" component="calResult" update="showText">
    <use-object name="generalCal" method="doCancel"/>
  </change>

  <change container="calculatorBoard" component="operand1" update="value"/>

  <change container="calculatorBoard" component="operand2" update="value"/>

```

```

    <change container="calculatorBoard" component="stateMsg" update="showText">
      <use-object name="memoryCal" method="getMState"/>
    </change>

    <nextboard goto="calculatorBoard"/>
  </action>

  <action name="mcCal" showText=" MC ">
    <use-object name="memoryCal" method="doMC"/>

    <change container="calculatorBoard" component="stateMsg" update="showText">
      <use-object name="memoryCal" method="getMState"/>
    </change>

    <nextboard goto="calculatorBoard"/>
  </action>

  <action name="cCal" showText=" C ">
    <change container="calculatorBoard" component="calResult" update="showText">
      <use-object name="generalCal" method="doCancel"/>
    </change>

    <change container="calculatorBoard" component="operand1" update="value"/>

    <change container="calculatorBoard" component="operand2" update="value"/>

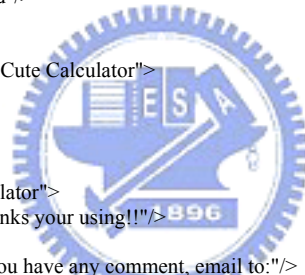
    <change container="calculatorBoard" component="errorMsg" update="showText">
      <use-object name="generalCal" method="getErrorMsg"/>
    </change>

    <nextboard goto="calculatorBoard"/>
  </action>

  <action name="exitCal" showText="Exit Cute Calculator">
    <nextboard goto="endBoard"/>
  </action>
</board>

<board name="endBoard" title="Exit Cute Calculator">
  <label name="endMsg1" showText="Thanks your using!!"/>
  <label name="blankMsg" showText=""/>
  <label name="endMsg2" showText="If you have any comment, email to:"/>
  <label name="endMsg3" showText="cuteCalculator@dcslab.cis.nctu.edu.tw"/>
</board>
</user-interface>

```



## [ Source PGML Code – GeneralCal.pgml ]

```

<?xml version="1.0"?>
<object name="GeneralCal" version="1.0" xmlns="http://dcsw3.cis.nctu.edu.tw/Project/Pervasive/PGML/">
  <declaration>
    <variable type="int" name="calResult" value="0"/>
    <variable type="String" name="errorMsg" value=""/>
  </declaration>

  <method name="doOP" visibility="public" return-type="void">
    <in>
      <variable type="String" name="operand1"/>
      <variable type="int" name="opIndex"/>
      <variable type="String" name="operand2"/>
    </in>

    <declaration>
      <variable type="int" name="opd1" value="0"/>
      <variable type="int" name="opd2" value="0"/>
    </declaration>

    <action>
      <if>

```

```

<test>
<not>
  <equal-to>
    <lib call="pgml.String.compare">
      <operand select="operand1"/>
      <operand type="String" value=""/>
    </lib>
    <operand type="int" value="0"/>
  </equal-to>
</not>
</test>

<action>
  <assign arg1="opd1">
    <lib call="pgml.String.parseInt">
      <operand select="operand1"/>
    </lib>
  </assign>

  <assign arg1="errorMsg">
    <operand type="String" value=""/>
  </assign>
</action>

<else>
  <assign arg1="errorMsg">
    <operand type="String" value="ERROR: empty input!"/>
  </assign>
</else>
</if>

<if>
  <test>
  <not>
    <equal-to>
      <lib call="pgml.String.compare">
        <operand select="operand2"/>
        <operand type="String" value=""/>
      </lib>
      <operand type="int" value="0"/>
    </equal-to>
  </not>
  </test>

  <action>
    <assign arg1="opd2">
      <lib call="pgml.String.parseInt">
        <operand select="operand2"/>
      </lib>
    </assign>

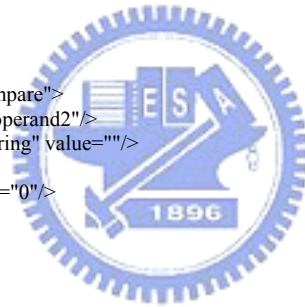
    <assign arg1="errorMsg">
      <operand type="String" value=""/>
    </assign>
  </action>

  <else>
    <assign arg1="errorMsg">
      <operand type="String" value="ERROR: empty input!"/>
    </assign>
  </else>
</if>

<if>
  <test>
  <equal-to>
    <lib call="pgml.String.compare">
      <operand select="errorMsg"/>
      <operand type="String" value=""/>
    </lib>
    <operand type="int" value="0"/>
  </equal-to>
  </test>

  <action>

```



```

<if>
  <test>
  <equal-to>
  <operand select="opIndex"/>
  <operand type="int" value="1"/>
  </equal-to>
  </test>

  <action>
  <add result="calResult">
    <operand select="opd1"/>
    <operand select="opd2"/>
  </add>
  </action>

  <else>
  <if>
    <test>
    <equal-to>
    <operand select="opIndex"/>
    <operand type="int" value="2"/>
    </equal-to>
    </test>

    <action>
    <sub result="calResult">
      <operand select="opd1"/>
      <operand select="opd2"/>
    </sub>
    </action>

    <else>
    <if>
      <test>
      <equal-to>
      <operand select="opIndex"/>
      <operand type="int" value="3"/>
      </equal-to>
      </test>

      <action>
      <mult result="calResult">
        <operand select="opd1"/>
        <operand select="opd2"/>
      </mult>
      </action>

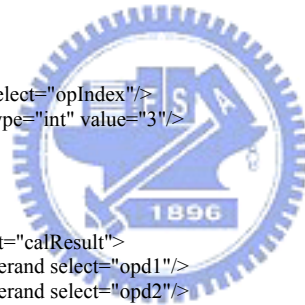
      <else>
      <if>
        <test>
        <equal-to>
        <operand select="opIndex"/>
        <operand type="int" value="4"/>
        </equal-to>
        </test>

        <action>
        <div result="calResult">
          <operand select="opd1"/>
          <operand select="opd2"/>
        </div>
        </action>

        <else>
        <if>
          <test>
          <equal-to>
          <operand select="opIndex"/>
          <operand type="int" value="5"/>
          </equal-to>
          </test>

          <action>
          <mod result="calResult">
            <operand select="opd1"/>

```



```

        <operand select="opd2"/>
    </mod>
</action>

    <else>
    <assign arg1="calResult">
    <operand type="int" value="0"/>
    </assign>
    </else>
</if>
</else>
</if>
</else>
</if>
</else>
</if>
</else>
</if>
</else>
</if>
</action>
</if>
</action>
</method>

```

```

<method name="doCancel" visibility="public" return-type="String">
    <action>
        <assign arg1="calResult">
            <operand type="int" value="0"/>
        </assign>

        <assign arg1="errorMsg">
            <operand type="String" value=""/>
        </assign>

        <return type="String" value="0"/>
    </action>
</method>

```

```

<method name="doInvert" visibility="public" return-type="void">
    <in>
        <variable type="String" name="invertee"/>
    </in>

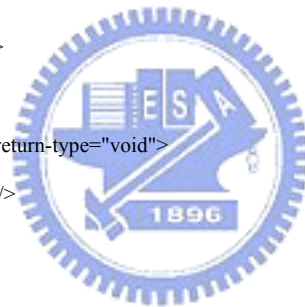
    <action>
    <if>
        <test>
            <not>
                <equal-to>
                <lib call="pgml.String.compare">
                    <operand select="invertee"/>
                    <operand type="String" value=""/>
                </lib>
                <operand type="int" value="0"/>
            </equal-to>
            </not>
        </test>

        <action>
            <assign arg1="calResult">
                <lib call="pgml.String.parseInt">
                    <operand select="invertee"/>
                </lib>
            </assign>

            <assign arg1="errorMsg">
                <operand type="String" value=""/>
            </assign>
        </action>

        <else>
            <assign arg1="errorMsg">
                <operand type="String" value="ERROR: empty input!"/>
            </assign>
        </else>
    </if>

```



```

</if>
  <test>
    <and>
      <equal-to>
        <lib call="pgml.String.compare">
          <operand select="errorMsg"/>
          <operand type="String" value=""/>
        </lib>
        <operand type="int" value="0"/>
      </equal-to>

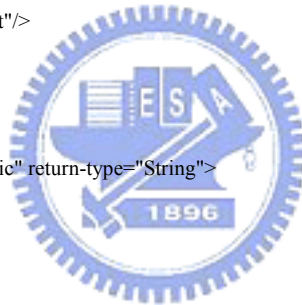
      <not-equal>
        <operand select="calResult"/>
        <operand type="int" value="0"/>
      </not-equal>
    </and>
  </test>

  <action>
    <mult result="calResult">
      <operand type="int" value="-1"/>
      <operand select="calResult"/>
    </mult>
  </action>
</if>
</action>
</method>

<method name="getCalResult" visibility="public" return-type="String">
  <action>
    <return>
      <lib call="pgml.Integer.toString">
        <operand select="calResult"/>
      </lib>
    </return>
  </action>
</method>

<method name="getErrorMsg" visibility="public" return-type="String">
  <action>
    <return select="errorMsg"/>
  </action>
</method>
</object>

```



### [ Source PGML Code – MemoryCal.pgml ]

```

<?xml version="1.0"?>

<object name="MemoryCal" version="1.0" xmlns="http://dcs3.cis.nctu.edu.tw/Project/Pervasive/PGML/">

  <declaration>
    <variable type="int" name="calMemory" value="0"/>
    <variable type="String" name="mState" value="state: "/>
  </declaration>

  <method name="setMState" visibility="private" return-type="void">
    <in>
      <variable type="String" name="memoryState"/>
    </in>

    <action>
      <assign arg1="mState" arg2="memoryState"/>
    </action>
  </method>

  <method name="getMState" visibility="public" return-type="String">
    <action>
      <return select="mState"/>
    </action>
  </method>

```

```

<method name="doMplus" visibility="public" return-type="void">
  <in>
    <variable type="String" name="tempOperand"/>
  </in>

  <action>
    <add result="calMemory">
      <operand select="calMemory"/>
      <lib call="pgml.String.parseInt">
        <operand select="tempOperand"/>
      </lib>
    </add>

    <call-method name="setMState">
      <operand type="String" value="state: M"/>
    </call-method>
  </action>
</method>

<method name="doMR" visibility="public" return-type="String">
  <action>
    <return>
      <lib call="pgml.Integer.toString">
        <operand select="calMemory"/>
      </lib>
    </return>
  </action>
</method>

<method name="doMS" visibility="public" return-type="void">
  <in>
    <variable type="String" name="tempOperand"/>
  </in>

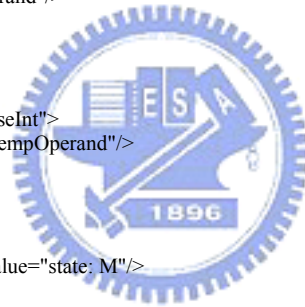
  <action>
    <assign arg1="calMemory">
      <lib call="pgml.String.parseInt">
        <operand select="tempOperand"/>
      </lib>
    </assign>

    <call-method name="setMState">
      <operand type="String" value="state: M"/>
    </call-method>
  </action>
</method>

<method name="doMC" visibility="public" return-type="void">
  <action>
    <assign arg1="calMemory">
      <operand type="int" value="0"/>
    </assign>

    <call-method name="setMState">
      <operand type="String" value="state: "/>
    </call-method>
  </action>
</method>
</object>

```



### [ Transformed WAP Code – CalculatorUI.wml (indented)]

```

<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE wml PUBLIC "-//WAPFORUM//DTD WML 1.3/EN" "http://www.wapforum.org/DTD/wml13.dtd">

<wml>
<card id="CalculatorUI_index" ontimer="#inner_CalculatorUI_memoryCal">
  <timer value="1"/>
</card>

<card id="inner_CalculatorUI_memoryCal" onenterforward="MemoryCal.wmls#globalVarDeclaration()"
ontimer="#inner_calculatorBoard_generalCal">

```

```

<timer value="1"/>
</card>

<card id="inner_calculatorBoard_generalCal" onenterforward="GeneralCal.wmls#globalVarDeclaration()" ontimer="#preSetting">
  <timer value="1"/>
</card>

<card id="preSetting" ontimer="#welcomeBoard">
  <onevent type="onenterforward">
    <refresh>
      <setvar name="welcomeBoard_msg1_showText" value="Functions Detail: "/>
      <setvar name="welcomeBoard_msg2_showText" value="[ = ] -&gt; computation result"/>
      <setvar name="welcomeBoard_msg3_showText" value="[ C ] -&gt; cancel result"/>
      <setvar name="welcomeBoard_msg4_showText" value="[ +/- ]-&gt; invert sign"/>
      <setvar name="welcomeBoard_msg5_showText" value="[ M+ ]-&gt; memory added"/>
      <setvar name="welcomeBoard_msg6_showText" value="[ MR ]-&gt; memory record"/>
      <setvar name="welcomeBoard_msg7_showText" value="[ MS ]-&gt; memory saved"/>
      <setvar name="welcomeBoard_msg8_showText" value="[ MC ]-&gt; memory concealed"/>
      <setvar name="calculatorBoard_resultMsg_showText" value="Computation Result:"/>
      <setvar name="calculatorBoard_calResult_showText" value="0"/>
      <setvar name="calculatorBoard_lineMsg_showText" value="-----"/>
      <setvar name="calculatorBoard_iOP_item1" value=" + "/>
      <setvar name="calculatorBoard_iOP_item2" value=" - "/>
      <setvar name="calculatorBoard_iOP_item3" value=" x "/>
      <setvar name="calculatorBoard_iOP_item4" value=" / "/>
      <setvar name="calculatorBoard_iOP_item5" value=" % "/>
      <setvar name="calculatorBoard_stateMsg_showText" value="state: "/>
      <setvar name="calculatorBoard_errorMsg_showText" value=""/>
      <setvar name="endBoard_endMsg1_showText" value="Thanks your using!!"/>
      <setvar name="endBoard_blankMsg_showText" value=""/>
      <setvar name="endBoard_endMsg2_showText" value="If you have any comment, email to:"/>
      <setvar name="endBoard_endMsg3_showText" value="cuteCalculator@dcslab.cis.nctu.edu.tw"/>
    </refresh>
  </onevent>
  <timer value="1"/>
</card>

<card id="welcomeBoard" title="Cute Calculator">
  <p><b>$(welcomeBoard_msg1_showText)</b></p>
  <p><b>$(welcomeBoard_msg2_showText)</b></p>
  <p><b>$(welcomeBoard_msg3_showText)</b></p>
  <p><b>$(welcomeBoard_msg4_showText)</b></p>
  <p><b>$(welcomeBoard_msg5_showText)</b></p>
  <p><b>$(welcomeBoard_msg6_showText)</b></p>
  <p><b>$(welcomeBoard_msg7_showText)</b></p>
  <p><b>$(welcomeBoard_msg8_showText)</b></p>

  <do type="runCal" label="Run Calculator">
    <go href="#welcomeBoard_runCal_temp1"/>
  </do>
</card>

<card id="welcomeBoard_runCal_temp1" onenterforward="#calculatorBoard"/>

<card id="calculatorBoard" title="Cute Calculator">
  <p><b>$(calculatorBoard_resultMsg_showText)</b></p>
  <p><b>$(calculatorBoard_calResult_showText)</b></p>
  <p><b>$(calculatorBoard_lineMsg_showText)</b></p>
  <p><input name="calculatorBoard_operand1" type="text"/></p>
  <p>
    <select name="calculatorBoard_iOP" ivalue="1" multiple="false">
      <option>$(calculatorBoard_iOP_item1)</option>
      <option>$(calculatorBoard_iOP_item2)</option>
      <option>$(calculatorBoard_iOP_item3)</option>
      <option>$(calculatorBoard_iOP_item4)</option>
      <option>$(calculatorBoard_iOP_item5)</option>
    </select>
  </p>
  <p><input name="calculatorBoard_operand2" type="text"/></p>
  <p><b>$(calculatorBoard_stateMsg_showText)</b></p>
  <p><b>$(calculatorBoard_errorMsg_showText)</b></p>

  <do type="equalCal" label=" = ">
    <go href="#calculatorBoard_equalCal_temp1"/>
  </do>

```





```

<do type="invertCal" label=" +/- ">
  <go href="#calculatorBoard_invertCal_temp1"/>
</do>

<do type="mplusCal" label=" M+ ">
  <go href="#calculatorBoard_mplusCal_temp1"/>
</do>

<do type="mrCal" label=" MR ">
  <go href="#calculatorBoard_mrCal_temp1"/>
</do>

<do type="msCal" label=" MS ">
  <go href="#calculatorBoard_msCal_temp1"/>
</do>

<do type="mcCal" label=" MC ">
  <go href="#calculatorBoard_mcCal_temp1"/>
</do>

<do type="cCal" label=" C ">
  <go href="#calculatorBoard_cCal_temp1"/>
</do>

<do type="exitCal" label="Exit Cute Calculator">
  <go href="#calculatorBoard_exitCal_temp1"/>
</do>
</card>

```

```

<card id="calculatorBoard_equalCal_temp1" onenterforward="GeneralCal.wmls#doOP('$calculatorBoard_operand1', '$calculatorBoard_iOP'), '$calculatorBoard_operand2)'" ontimer="#calculatorBoard_equalCal_temp2">
  <timer value="1"/>
</card>

```

```

<card id="calculatorBoard_equalCal_temp2" onenterforward="GeneralCal.wmls#change(calculatorBoard_calResult_showText', ', 'getCalResult)'"
ontimer="#calculatorBoard_equalCal_temp3">
  <timer value="1"/>
</card>

```

```

<card id="calculatorBoard_equalCal_temp3" onenterforward="GeneralCal.wmls#change(calculatorBoard_operand1', ', 'getCalResult)'"
ontimer="#calculatorBoard_equalCal_temp4">
  <timer value="1"/>
</card>

```

```

<card id="calculatorBoard_equalCal_temp4" onenterforward="PGMLchange.wmls#change(calculatorBoard_operand2', ', ')"
ontimer="#calculatorBoard_equalCal_temp5">
  <timer value="1"/>
</card>

```

```

<card id="calculatorBoard_equalCal_temp5" onenterforward="GeneralCal.wmls#change(calculatorBoard_errorMsg_showText', ', 'getErrorMsg)'"
ontimer="#calculatorBoard_equalCal_temp6">
  <timer value="1"/>
</card>

```

```

<card id="calculatorBoard_equalCal_temp6" onenterforward="#calculatorBoard"/>

```

```

<card id="calculatorBoard_invertCal_temp1" onenterforward="GeneralCal.wmls#doInvert('$calculatorBoard_operand1)'"
ontimer="#calculatorBoard_invertCal_temp2">
  <timer value="1"/>
</card>

```

```

<card id="calculatorBoard_invertCal_temp2" onenterforward="GeneralCal.wmls#change(calculatorBoard_calResult_showText', ', 'getCalResult)'"
ontimer="#calculatorBoard_invertCal_temp3">
  <timer value="1"/>
</card>

```

```

<card id="calculatorBoard_invertCal_temp3" onenterforward="GeneralCal.wmls#change(calculatorBoard_operand1', ', 'getCalResult)'"
ontimer="#calculatorBoard_invertCal_temp4">
  <timer value="1"/>
</card>

```

```

<card id="calculatorBoard_invertCal_temp4" onenterforward="PGMLchange.wmls#change(calculatorBoard_operand2', ', ')"
ontimer="#calculatorBoard_invertCal_temp5">
  <timer value="1"/>

```



```

</card>

<card id="calculatorBoard_invertCal_temp5" onenterforward="GeneralCal.wmls#change('calculatorBoard_errorMsg_showText', 'getErrorMsg')"
ontimer="#calculatorBoard_invertCal_temp6">
  <timer value="1"/>
</card>

<card id="calculatorBoard_invertCal_temp6" onenterforward="#calculatorBoard"/>

<card id="calculatorBoard_mplusCal_temp1" onenterforward="MemoryCal.wmls#doMplus('${calculatorBoard_calResult_showText}')"
ontimer="#calculatorBoard_mplusCal_temp2">
  <timer value="1"/>
</card>

<card id="calculatorBoard_mplusCal_temp2" onenterforward="GeneralCal.wmls#change('calculatorBoard_calResult_showText', 'doCancel!')"
ontimer="#calculatorBoard_mplusCal_temp3">
  <timer value="1"/>
</card>

<card id="calculatorBoard_mplusCal_temp3" onenterforward="PGMLchange.wmls#change('calculatorBoard_operand1', '')"
ontimer="#calculatorBoard_mplusCal_temp4">
  <timer value="1"/>
</card>

<card id="calculatorBoard_mplusCal_temp4" onenterforward="PGMLchange.wmls#change('calculatorBoard_operand2', '')"
ontimer="#calculatorBoard_mplusCal_temp5">
  <timer value="1"/>
</card>

<card id="calculatorBoard_mplusCal_temp5" onenterforward="MemoryCal.wmls#change('calculatorBoard_stateMsg_showText', 'getMState')"
ontimer="#calculatorBoard_mplusCal_temp6">
  <timer value="1"/>
</card>

<card id="calculatorBoard_mplusCal_temp6" onenterforward="#calculatorBoard"/>

<card id="calculatorBoard_mrCal_temp1" onenterforward="MemoryCal.wmls#change('calculatorBoard_calResult_showText', 'doMR')"
ontimer="#calculatorBoard_mrCal_temp2">
  <timer value="1"/>
</card>

<card id="calculatorBoard_mrCal_temp2" onenterforward="PGMLchange.wmls#change('calculatorBoard_operand1', '')"
ontimer="#calculatorBoard_mrCal_temp3">
  <timer value="1"/>
</card>

<card id="calculatorBoard_mrCal_temp3" onenterforward="PGMLchange.wmls#change('calculatorBoard_operand2', '')"
ontimer="#calculatorBoard_mrCal_temp4">
  <timer value="1"/>
</card>

<card id="calculatorBoard_mrCal_temp4" onenterforward="MemoryCal.wmls#change('calculatorBoard_stateMsg_showText', 'getMState')"
ontimer="#calculatorBoard_mrCal_temp5">
  <timer value="1"/>
</card>

<card id="calculatorBoard_mrCal_temp5" onenterforward="#calculatorBoard"/>

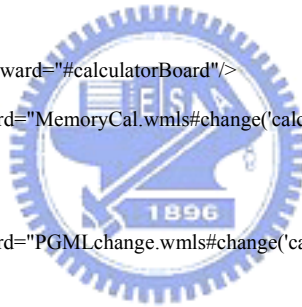
<card id="calculatorBoard_msCal_temp1" onenterforward="MemoryCal.wmls#doMS('${calculatorBoard_calResult_showText}')"
ontimer="#calculatorBoard_msCal_temp2">
  <timer value="1"/>
</card>

<card id="calculatorBoard_msCal_temp2" onenterforward="GeneralCal.wmls#change('calculatorBoard_calResult_showText', 'doCancel!')"
ontimer="#calculatorBoard_msCal_temp3">
  <timer value="1"/>
</card>

<card id="calculatorBoard_msCal_temp3" onenterforward="PGMLchange.wmls#change('calculatorBoard_operand1', '')"
ontimer="#calculatorBoard_msCal_temp4">
  <timer value="1"/>
</card>

<card id="calculatorBoard_msCal_temp4" onenterforward="PGMLchange.wmls#change('calculatorBoard_operand2', '')"
ontimer="#calculatorBoard_msCal_temp5">

```



```

        <timer value="1"/>
</card>

<card id="calculatorBoard_msCal_temp5" onenterforward="MemoryCal.wmls#change('calculatorBoard_stateMsg_showText', '', 'getMState')"
ontimer="#calculatorBoard_msCal_temp6">
    <timer value="1"/>
</card>

<card id="calculatorBoard_msCal_temp6" onenterforward="#calculatorBoard"/>

<card id="calculatorBoard_mcCal_temp1" onenterforward="MemoryCal.wmls#doMC()" ontimer="#calculatorBoard_mcCal_temp2">
    <timer value="1"/>
</card>

<card id="calculatorBoard_mcCal_temp2" onenterforward="MemoryCal.wmls#change('calculatorBoard_stateMsg_showText', '', 'getMState')"
ontimer="#calculatorBoard_mcCal_temp3">
    <timer value="1"/>
</card>

<card id="calculatorBoard_mcCal_temp3" onenterforward="#calculatorBoard"/>

<card id="calculatorBoard_cCal_temp1" onenterforward="GeneralCal.wmls#change('calculatorBoard_calResult_showText', '', 'doCancel!')"
ontimer="#calculatorBoard_cCal_temp2">
    <timer value="1"/>
</card>

<card id="calculatorBoard_cCal_temp2" onenterforward="PGMLchange.wmls#change('calculatorBoard_operand1', '', '')"
ontimer="#calculatorBoard_cCal_temp3">
    <timer value="1"/>
</card>

<card id="calculatorBoard_cCal_temp3" onenterforward="PGMLchange.wmls#change('calculatorBoard_operand2', '', '')"
ontimer="#calculatorBoard_cCal_temp4">
    <timer value="1"/>
</card>

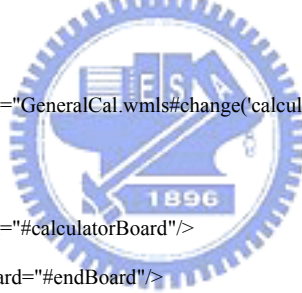
<card id="calculatorBoard_cCal_temp4" onenterforward="GeneralCal.wmls#change('calculatorBoard_errorMsg_showText', '', 'getErrorMsg')"
ontimer="#calculatorBoard_cCal_temp5">
    <timer value="1"/>
</card>

<card id="calculatorBoard_cCal_temp5" onenterforward="#calculatorBoard"/>

<card id="calculatorBoard_exitCal_temp1" onenterforward="#endBoard"/>

<card id="endBoard" title="Exit Cute Calculator">
    <p><b>$(endBoard_endMsg1_showText)</b></p>
    <p><b>$(endBoard_blankMsg_showText)</b></p>
    <p><b>$(endBoard_endMsg2_showText)</b></p>
    <p><b>$(endBoard_endMsg3_showText)</b></p>
</card>
</wml>

```



## [ Transformed WAP Code – GeneralCal.wmls (indented) ]

```

/*some pragma are declared here...*/

/*WMLScript interface to set declared global variables*/
extern function globalVarDeclaration() {
    /*setting all global variables in this logic object*/
    WMLBrowser.setVar("GeneralCal_calResult", 0);
    WMLBrowser.setVar("GeneralCal_errorMsg", "");
}

/*all methods are here...*/
extern function doOP (operand1, opIndex, operand2){
    var opd1=0;    var opd2=0;

    if(!(String.compare(operand1, "")==0)){
        opd1=Lang.parseInt(operand1);
        WMLBrowser.setVar("GeneralCal_errorMsg", "");
    }
}

```

```

else{
    WMLBrowser.setVar("GeneralCal_errorMsg", "ERROR: empty input!");
}

if(!(String.compare(operand2, "")==0)){
    opd2=Lang.parseInt(operand2);
    WMLBrowser.setVar("GeneralCal_errorMsg", "");
}
else{
    WMLBrowser.setVar("GeneralCal_errorMsg", "ERROR: empty input!");
}

if(String.compare(WMLBrowser.getVar("GeneralCal_errorMsg"), "")==0){
    if(opIndex==1){
        WMLBrowser.setVar("GeneralCal_calResult", ( Lang.parseInt(opd1) + Lang.parseInt(opd2) ));
    }
    else{
        if(opIndex==2){
            WMLBrowser.setVar("GeneralCal_calResult", ( Lang.parseInt(opd1) - Lang.parseInt(opd2) ));
        }
        else{
            if(opIndex==3){
                WMLBrowser.setVar("GeneralCal_calResult", Lang.parseInt(opd1) * Lang.parseInt(opd2) );
            }
            else{
                if(opIndex==4){
                    WMLBrowser.setVar("GeneralCal_calResult", Lang.parseInt(opd1) / Lang.parseInt(opd2) );
                }
                else{
                    if(opIndex==5){
                        WMLBrowser.setVar("GeneralCal_calResult", Lang.parseInt(opd1) % Lang.parseInt(opd2) );
                    }
                    else{
                        WMLBrowser.setVar("GeneralCal_calResult", 0);
                    }
                }
            }
        }
    }
}

extern function doCancel (){
    WMLBrowser.setVar("GeneralCal_calResult", 0);
    WMLBrowser.setVar("GeneralCal_errorMsg", "");

    return "0";
}

extern function doInvert (invertee){
    if(!(String.compare(invertee, "")==0) ){
        WMLBrowser.setVar("GeneralCal_calResult", Lang.parseInt(invertee));
        WMLBrowser.setVar("GeneralCal_errorMsg", "");
    }
    else{
        WMLBrowser.setVar("GeneralCal_errorMsg", "ERROR: empty input!");
    }

    if((String.compare(WMLBrowser.getVar("GeneralCal_errorMsg"), "")==0 && WMLBrowser.getVar("GeneralCal_calResult")!=0) ){
        WMLBrowser.setVar("GeneralCal_calResult", -1* Lang.parseInt(WMLBrowser.getVar("GeneralCal_calResult")));
    }
}

extern function getCalResult (){
    return String.toString(WMLBrowser.getVar("GeneralCal_calResult"));
}

extern function getErrorMsg (){
    return WMLBrowser.getVar("GeneralCal_errorMsg");
}

/*WMLScript interface to change value of UI component on some WML card*/
extern function change(component, new Value, callMethod){

```



```

if(callMethod == ""){
    WMLBrowser.setVar(component, newValue);
}
else{
    if(callMethod == "doCancel"){
        WMLBrowser.setVar(component, doCancel());
    }
    if(callMethod == "getCalResult"){
        WMLBrowser.setVar(component, getCalResult());
    }
    if(callMethod == "getErrorMsg"){
        WMLBrowser.setVar(component, getErrorMsg());
    }
}
}
}

```

## [ Transformed WAP Code – MemoryCal.wmls (indented) ]

/\*some pragma are declared here...\*/

```

/*WMLScript interface to set declared global variables*/
extern function globalVarDeclaration() {
    /*setting all global variables in this logic object*/
    WMLBrowser.setVar("MemoryCal_calMemory", 0);
    WMLBrowser.setVar("MemoryCal_mState", "state: ");
}

```

/\*all methods are here...\*/

```

function setMState (memoryState){
    WMLBrowser.setVar("MemoryCal_mState", memoryState );
}

extern function getMState (){
    return WMLBrowser.getVar("MemoryCal_mState");
}

extern function doMplus (tempOperand){
    WMLBrowser.setVar("MemoryCal_calMemory", (Lang.parseInt(WMLBrowser.getVar("MemoryCal_calMemory"))
    + Lang.parseInt(Lang.parseInt(tempOperand))));
    setMState("state: M");
}

extern function doMR (){
    return String.toString(WMLBrowser.getVar("MemoryCal_calMemory"));
}

extern function doMS (tempOperand){
    WMLBrowser.setVar("MemoryCal_calMemory", Lang.parseInt(tempOperand));
    setMState("state: M");
}

extern function doMC (){
    WMLBrowser.setVar("MemoryCal_calMemory", 0);
    setMState("state: ");
}

```

/\*WMLScript interface to change value of UI component on some WML card\*/

```

extern function change(component, newValue, callMethod){
    if(callMethod == ""){
        WMLBrowser.setVar(component, newValue);
    }
    else{
        if(callMethod == "getMState"){
            WMLBrowser.setVar(component, getMState());
        }
        if(callMethod == "doMR"){
            WMLBrowser.setVar(component, doMR());
        }
        if(callMethod == "doMC"){
            WMLBrowser.setVar(component, doMC());
        }
    }
}

```



```

    }
}

```

### [ Transformed WAP Code – PGMLchange.wmls (indented) ]

```

extern function change(component, newValue, nouse){
    WMLBrowser.setVar(component, newValue);
}

```

### [ Transformed J2ME Code – BMngr\_CalculatorUI.java (indented) ]

```

package calculator;
import javax.microedition.midlet.*;
import javax.microedition.lcdui.*;

```

```

interface BMngr_Interface {
    public void changeBoard(String boardName);
    public Displayable getBoard(String boardName);
}

```

```

public class BMngr_CalculatorUI extends MIDlet implements BMngr_Interface{
    public static BMngr_CalculatorUI instance;
    private Display display;

```

```

    /*global logic objects*/
    public MemoryCal lgObj_memoryCal = new MemoryCal();

```

```

    /*all displayables corresponding to boards' declaration*/
    private B_welcomeBoard welcomeBoard = new B_welcomeBoard();
    private B_calculatorBoard calculatorBoard = new B_calculatorBoard();
    private B_endBoard endBoard = new B_endBoard();

```

```

    /*Constructor*/
    public BMngr_CalculatorUI(){
        instance = this;
    }

```

```

    /*Main method*/
    public void startApp(){
        display = Display.getDisplay(this);
        display.setCurrent(welcomeBoard);
    }

```

```

    /*Handle pausing the MIDlet*/
    public void pauseApp(){
    }

```

```

    /*Handle destroying the MIDlet*/
    public void destroyApp(boolean unconditional){
    }

```

```

    /*Quit the MIDlet*/
    public static void quitApp(){
        instance.destroyApp(true);
        instance.notifyDestroyed();
        instance = null;
    }

```

```

    /*change UI presentation*/
    public void changeBoard(String boardName){
        if(boardName.compareTo("welcomeBoard") == 0){
            display.setCurrent(welcomeBoard);
        }
        if(boardName.compareTo("calculatorBoard") == 0){
            display.setCurrent(calculatorBoard);
        }
        if(boardName.compareTo("endBoard") == 0){
            display.setCurrent(endBoard);
        }
    }
}

```



```

/*get particular UI container*/
public Displayable getBoard(String boardName){
    Displayable displayable = null;
    if(boardName.compareTo("welcomeBoard") == 0){
        displayable = welcomeBoard;
    }
    if(boardName.compareTo("calculatorBoard") == 0){
        displayable = calculatorBoard;
    }
    if(boardName.compareTo("endBoard") == 0){
        displayable = endBoard;
    }

    return displayable;
}
}

```

```

class B_welcomeBoard extends Form implements CommandListener {
    /*local logic objects*/

```

```

/*UI components*/

```

```

public StringItem msg1;
public StringItem msg2;
public StringItem msg3;
public StringItem msg4;
public StringItem msg5;
public StringItem msg6;
public StringItem msg7;
public StringItem msg8;

```

```

/*commands*/

```

```

private Command runCal = new Command("Run Calculator", Command.SCREEN, 1);

```

```

/*Constructor*/

```

```

public B_welcomeBoard() {
    /* initialize UI component */
    super("Cute Calculator");
    msg1 = new StringItem("Functions Detail: ", "");
    msg2 = new StringItem("[ = ]-> computation result", "");
    msg3 = new StringItem("[ C ]-> cancel result", "");
    msg4 = new StringItem("[ +/- ]-> invert sign", "");
    msg5 = new StringItem("[ M+ ]-> memory added", "");
    msg6 = new StringItem("[ MR ]-> memory record", "");
    msg7 = new StringItem("[ MS ]-> memory saved", "");
    msg8 = new StringItem("[ MC ]-> memory canceled", "");

```

```

/* set up listeners */

```

```

this.setCommandListener(this);

```

```

/* append UI component */

```

```

this.append(msg1);
this.append(msg2);
this.append(msg3);
this.append(msg4);
this.append(msg5);
this.append(msg6);
this.append(msg7);
this.append(msg8);

```

```

/* add commands */

```

```

this.addCommand(runCal);
}

```

```

/*Handle command events*/

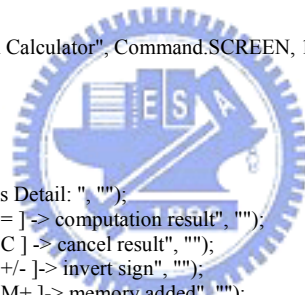
```

```

public void commandAction(Command command, Displayable displayable){
    String selectedStr = null;
    String oldPath = null;
    String newPath = null;
    String align = null;

    if(command == runCal){
        BMngr_CalculatorUI.instance.changeBoard("calculatorBoard");
    }
}

```



```

    }
}

```

```

class B_calculatorBoard extends Form implements CommandListener {
    /*local logic objects*/
    private GeneralCal lgObj_generalCal = new GeneralCal();

    /*UI components*/
    public StringItem resultMsg;
    public StringItem calResult;
    public StringItem lineMsg;
    public TextField operand1;
    public ChoiceGroup iOP;
    public TextField operand2;
    public StringItem stateMsg;
    public StringItem errorMsg;

    /*commands*/
    private Command equalCal = new Command(" = ", Command.SCREEN, 1);
    private Command invertCal = new Command(" +/- ", Command.SCREEN, 1);
    private Command mplusCal = new Command(" M+ ", Command.SCREEN, 1);
    private Command mrCal = new Command(" MR ", Command.SCREEN, 1);
    private Command msCal = new Command(" MS ", Command.SCREEN, 1);
    private Command mcCal = new Command(" MC ", Command.SCREEN, 1);
    private Command cCal = new Command(" C ", Command.SCREEN, 1);
    private Command exitCal = new Command("Exit Cute Calculator", Command.SCREEN, 1);

    /*Constructor*/
    public B_calculatorBoard() {
        /* initialize UI component */
        super("Cute Calculator");
        resultMsg = new StringItem("Computation Result: ", "");
        calResult = new StringItem("0", "");
        lineMsg = new StringItem("-----", "");
        operand1 = new TextField("", "", 50, TextField.ANY);
        iOP = new ChoiceGroup("", ChoiceGroup.EXCLUSIVE);
        iOP.append(" + ", null);
        iOP.append(" - ", null);
        iOP.append(" x ", null);
        iOP.append(" / ", null);
        iOP.append(" % ", null);
        iOP.setSelectedIndex(1-1, true);
        operand2 = new TextField("", "", 50, TextField.ANY);
        stateMsg = new StringItem("state: ", "");
        errorMsg = new StringItem("", "");

        /* set up listeners */
        this.setCommandListener(this);

        /* append UI component */
        this.append(resultMsg);
        this.append(calResult);
        this.append(lineMsg);
        this.append(operand1);
        this.append(iOP);
        this.append(operand2);
        this.append(stateMsg);
        this.append(errorMsg);

        /* add commands */
        this.addCommand(equalCal);
        this.addCommand(invertCal);
        this.addCommand(mplusCal);
        this.addCommand(mrCal);
        this.addCommand(msCal);
        this.addCommand(mcCal);
        this.addCommand(cCal);
        this.addCommand(exitCal);
    }

    /*Handle command events*/
    public void commandAction(Command command, Displayable displayable){
        String selectedStr = null;

```





```

String oldPath = null;
String newPath = null;
String align = null;

if(command == equalCal){
    lgObj_generalCal.doOP(operand1.getString(), iOP.getSelectedIndex()+1, operand2.getString());
    ((B_calculatorBoard)BMngr_CalculatorUI.instance.getBoard("calculatorBoard")).calResult.
        setLabel(lgObj_generalCal.getCalResult());
    ((B_calculatorBoard)BMngr_CalculatorUI.instance.getBoard("calculatorBoard")).operand1.
        setString(lgObj_generalCal.getCalResult());
    ((B_calculatorBoard)BMngr_CalculatorUI.instance.getBoard("calculatorBoard")).operand2.setString("");
    ((B_calculatorBoard)BMngr_CalculatorUI.instance.getBoard("calculatorBoard")).errorMsg.
        setLabel(lgObj_generalCal.getErrorMsg());
    BMngr_CalculatorUI.instance.changeBoard("calculatorBoard");
}

if(command == invertCal){
    lgObj_generalCal.doInvert(operand1.getString());
    ((B_calculatorBoard)BMngr_CalculatorUI.instance.getBoard("calculatorBoard")).calResult.
        setLabel(lgObj_generalCal.getCalResult());
    ((B_calculatorBoard)BMngr_CalculatorUI.instance.getBoard("calculatorBoard")).operand1.
        setString(lgObj_generalCal.getCalResult());
    ((B_calculatorBoard)BMngr_CalculatorUI.instance.getBoard("calculatorBoard")).operand2.setString("");
    ((B_calculatorBoard)BMngr_CalculatorUI.instance.getBoard("calculatorBoard")).errorMsg.
        setLabel(lgObj_generalCal.getErrorMsg());
    BMngr_CalculatorUI.instance.changeBoard("calculatorBoard");
}

if(command == mplusCal){
    BMngr_CalculatorUI.instance.lgObj_memoryCal.doMplus(calResult.getLabel());
    ((B_calculatorBoard)BMngr_CalculatorUI.instance.getBoard("calculatorBoard")).calResult.
        setLabel(lgObj_generalCal.doCancel());
    ((B_calculatorBoard)BMngr_CalculatorUI.instance.getBoard("calculatorBoard")).operand1.setString("");
    ((B_calculatorBoard)BMngr_CalculatorUI.instance.getBoard("calculatorBoard")).operand2.setString("");
    ((B_calculatorBoard)BMngr_CalculatorUI.instance.getBoard("calculatorBoard")).stateMsg.
        setLabel(BMngr_CalculatorUI.instance.lgObj_memoryCal.getMState());
    BMngr_CalculatorUI.instance.changeBoard("calculatorBoard");
}

if(command == mrCal){
    ((B_calculatorBoard)BMngr_CalculatorUI.instance.getBoard("calculatorBoard")).calResult.
        setLabel(BMngr_CalculatorUI.instance.lgObj_memoryCal.doMR());
    ((B_calculatorBoard)BMngr_CalculatorUI.instance.getBoard("calculatorBoard")).operand1.setString("");
    ((B_calculatorBoard)BMngr_CalculatorUI.instance.getBoard("calculatorBoard")).operand2.setString("");
    ((B_calculatorBoard)BMngr_CalculatorUI.instance.getBoard("calculatorBoard")).stateMsg.
        setLabel(BMngr_CalculatorUI.instance.lgObj_memoryCal.getMState());
    BMngr_CalculatorUI.instance.changeBoard("calculatorBoard");
}

if(command == msCal){
    BMngr_CalculatorUI.instance.lgObj_memoryCal.doMS(calResult.getLabel());
    ((B_calculatorBoard)BMngr_CalculatorUI.instance.getBoard("calculatorBoard")).calResult.
        setLabel(lgObj_generalCal.doCancel());
    ((B_calculatorBoard)BMngr_CalculatorUI.instance.getBoard("calculatorBoard")).operand1.setString("");
    ((B_calculatorBoard)BMngr_CalculatorUI.instance.getBoard("calculatorBoard")).operand2.setString("");
    ((B_calculatorBoard)BMngr_CalculatorUI.instance.getBoard("calculatorBoard")).stateMsg.
        setLabel(BMngr_CalculatorUI.instance.lgObj_memoryCal.getMState());
    BMngr_CalculatorUI.instance.changeBoard("calculatorBoard");
}

if(command == mcCal){
    BMngr_CalculatorUI.instance.lgObj_memoryCal.doMC();
    ((B_calculatorBoard)BMngr_CalculatorUI.instance.getBoard("calculatorBoard")).stateMsg.
        setLabel(BMngr_CalculatorUI.instance.lgObj_memoryCal.getMState());
    BMngr_CalculatorUI.instance.changeBoard("calculatorBoard");
}

if(command == cCal){
    ((B_calculatorBoard)BMngr_CalculatorUI.instance.getBoard("calculatorBoard")).calResult.
        setLabel(lgObj_generalCal.doCancel());
    ((B_calculatorBoard)BMngr_CalculatorUI.instance.getBoard("calculatorBoard")).operand1.setString("");
    ((B_calculatorBoard)BMngr_CalculatorUI.instance.getBoard("calculatorBoard")).operand2.setString("");
    ((B_calculatorBoard)BMngr_CalculatorUI.instance.getBoard("calculatorBoard")).errorMsg.
        setLabel(lgObj_generalCal.getErrorMsg());
    BMngr_CalculatorUI.instance.changeBoard("calculatorBoard");
}

```

```

    }

    if(command == exitCal){
        BMngr_CalculatorUI.instance.changeBoard("endBoard");
    }
}

```

```

class B_endBoard extends Form implements CommandListener {
    /*local logic objects*/

    /*UI components*/
    public StringItem endMsg1;
    public StringItem blankMsg;
    public StringItem endMsg2;
    public StringItem endMsg3;

    /*commands*/

    /*Constructor*/
    public B_endBoard() {
        /* initialize UI component */
        super("Exit Cute Calculator");
        endMsg1 = new StringItem("Thanks your using!!", "");
        blankMsg = new StringItem("", "");
        endMsg2 = new StringItem("If you have any comment, email to:", "");
        endMsg3 = new StringItem("cuteCalculator@dcslab.cis.nctu.edu.tw", "");

        /* set up listeners */
        this.setCommandListener(this);

        /* append UI component */
        this.append(endMsg1);
        this.append(blankMsg);
        this.append(endMsg2);
        this.append(endMsg3);

        /* add commands */
    }

    /*Handle command events*/
    public void commandAction(Command command, Displayable displayable) {
        String selectedStr = null;
        String oldPath = null;
        String newPath = null;
        String align = null;
    }
}

```



### [ Transformed J2ME Code – GeneralCal.java (indented) ]

```

package calculator;
/*J2ME "import" declarations*/

public class GeneralCal {
    /*global variables' declaration*/
    private int calResult=0;    private String errorMsg="";

    /*constructor*/
    public GeneralCal(){}

    /*all methods are here...*/
    public void doOP( String operand1, int opIndex, String operand2 ){
        /*local variables' declaration*/
        int opd1=0;    int opd2=0;

        if(!(operand1.compareTo("")==0) ){
            opd1=java.lang.Integer.parseInt(operand1);
            this.errorMsg="";
        }
        else{

```

```

        this.errorMsg="ERROR: empty input!";
    }

    if(!(operand2.compareTo("")==0)){
        opd2=java.lang.Integer.parseInt(operand2);
        this.errorMsg="";
    }
    else{
        this.errorMsg="ERROR: empty input!";
    }

    if(this.errorMsg.compareTo("")==0){
        if(opIndex==1){
            this.calResult=( opd1 + opd2 );
        }
        else{
            if(opIndex==2){
                this.calResult=( opd1 - opd2 );
            }
            else{
                if(opIndex==3){
                    this.calResult=opd1*opd2;
                }
                else{
                    if(opIndex==4){
                        this.calResult=opd1/opd2;
                    }
                    else{
                        if(opIndex==5){
                            this.calResult=opd1%opd2;
                        }
                        else{
                            this.calResult=0;
                        }
                    }
                }
            }
        }
    }
}

public String doCancel( ){
    /*local variables' declaration*/

    this.calResult=0;
    this.errorMsg="";
    return "0";
}

public void doInvert( String invertee ){
    /*local variables' declaration*/

    if(!(invertee.compareTo("")==0) ){
        this.calResult=java.lang.Integer.parseInt(invertee);
        this.errorMsg="";
    }
    else{
        this.errorMsg="ERROR: empty input!";
    }

    if((this.errorMsg.compareTo("")==0 && this.calResult!=0) ){
        this.calResult=-1*
        this.calResult;
    }
}

public String getCalResult( ){
    /*local variables' declaration*/

    return java.lang.Integer.toString(this.calResult);
}

public String getErrorMsg( ){
    /*local variables' declaration*/

```



```

    }
    return this.errorMsg;
}

```

### [ Transformed J2ME Code – MemoryCal.java (indented) ]

```

package calculator;
/*J2ME "import" declarations*/

public class MemoryCal {
    /*global variables' declaration*/
    private int calMemory=0;    private String mState="state: ";

    /*constructor*/
    public MemoryCal(){

    /*all methods are here...*/
    private void setMState( String memoryState ){
        /*local variables' declaration*/

        this.mState=memoryState;
    }

    public String getMState( ){
        /*local variables' declaration*/

        return this.mState;
    }

    public void doMplus( String tempOperand ){
        /*local variables' declaration*/

        this.calMemory=(
        this.calMemory + java.lang.Integer.parseInt(tempOperand) );
        setMState("state: M");
    }

    public String doMR( ){
        /*local variables' declaration*/

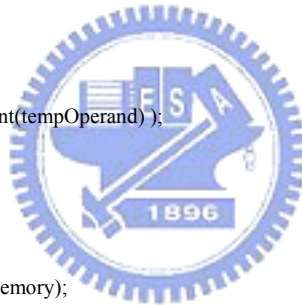
        return java.lang.Integer.toString(this.calMemory);
    }

    public void doMS( String tempOperand ){
        /*local variables' declaration*/

        this.calMemory=java.lang.Integer.parseInt(tempOperand);
        setMState("state: M");
    }

    public void doMC( ){
        /*local variables' declaration*/
        this.calMemory=0;
        setMState("state: ");
    }
}

```



---

# Appendix

## ALL XSLT Transformation Files

---

### [ For PGML – PGML2MIDP.xsl (indented) ]

```
<?xml version="1.0"?>

<!--this is an XSLT stylesheet of PGML 1.0 to transform PGML into J2ME MIDP without error control.-->

<xsl:stylesheet version="1.0" xmlns:xsl="http://www.w3.org/1999/XSL/Transform"
xmlns:pgml="http://dcs.w3.cis.nctu.edu.tw/Project/Pervasive/PGML/">

  <!--for PGML standard library-->
  <xsl:import href="PGMLStdLib2MIDP.xsl"/>

  <!--for package path information-->
  <xsl:param name="packagePath"/>

  <!--because MIDP is an text document-->
  <xsl:output method="text"/>

  <xsl:template match="/">
    <xsl:if test="$packagePath">
      package <xsl:value-of select="$packagePath"/>;
    </xsl:if>
    <xsl:text> </xsl:text>
    <xsl:text> /*J2ME &quot;import&quot; declarations*/</xsl:text>
    <xsl:text> </xsl:text>

    <xsl:apply-templates select="pgml:object"/>
  </xsl:template>

  <xsl:template match="pgml:object">
    public class <xsl:value-of select="@name"/> {
      /*global variables' declaration*/
      <xsl:apply-templates select="pgml:declaration"/>

      /*constructor*/
      public <xsl:value-of select="@name"/>(){}

      /*all methods are here...*/
      <xsl:for-each select="pgml:method">
        <xsl:apply-templates select="."/>

        <xsl:text> </xsl:text>
      </xsl:for-each>
    }
  </xsl:template>

  <xsl:template match="pgml:declaration">
    <xsl:for-each select="pgml:variable">
      <xsl:apply-templates select="."/>
      <xsl:text> </xsl:text>
    </xsl:for-each>
  </xsl:template>

  <xsl:template match="pgml:method">
```



```

        <xsl:value-of select="@visibility"/><xsl:text> </xsl:text><xsl:value-of select="@return-type"/><xsl:text> </xsl:text><xsl:value-of
select="@name"/>( <xsl:apply-templates select="pgml:in"/> ){
    /*local variables' declaration*/
    <xsl:apply-templates select="pgml:declaration"/>

    <xsl:apply-templates select="pgml:action"/>
}
<xsl:text>    </xsl:text>
</xsl:template>

```

```

<xsl:template match="pgml:variable">
  <xsl:if test="name(..)='object'">
    <xsl:text>private </xsl:text>
  </xsl:if>

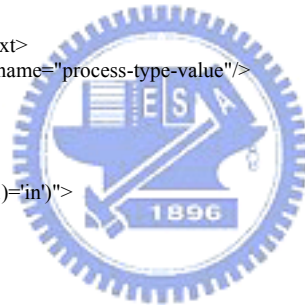
  <xsl:choose>
    <xsl:when test="node()">
      <xsl:value-of select="@type"/>
      <xsl:text> </xsl:text>
      <xsl:value-of select="@name"/>
      <xsl:text>=</xsl:text>
      <xsl:apply-templates/>
    </xsl:when>

    <xsl:otherwise>
      <xsl:value-of select="@type"/>
      <xsl:text> </xsl:text>
      <xsl:value-of select="@name"/>

      <xsl:if test="@value">
        <xsl:text>=</xsl:text>
        <xsl:call-template name="process-type-value"/>
      </xsl:if>
    </xsl:otherwise>
  </xsl:choose>

  <xsl:if test="not(name(..)='init' or name(..)='in')">
    <xsl:text>; </xsl:text>
  </xsl:if>
</xsl:template>

```



```

<xsl:template match="pgml:operand">
  <xsl:choose>
    <xsl:when test="@select">
      <xsl:call-template name="get-local-or-global-variable">
        <xsl:with-param name="varName" select="@select"/>
      </xsl:call-template>
    </xsl:when>

    <xsl:otherwise>
      <xsl:call-template name="process-type-value"/>
    </xsl:otherwise>
  </xsl:choose>
</xsl:template>

```

```

<xsl:template name="get-local-or-global-variable">
  <xsl:param name="varName">defaultVarName</xsl:param>

  <xsl:choose>
    <!--get local variable-->
    <xsl:when test="ancestor::pgml:method/pgml:declaration/pgml:variable[@name=$varName]">
      <xsl:value-of select="$varName"/>
    </xsl:when>

    <xsl:when test="ancestor::pgml:method/pgml:in/pgml:variable[@name=$varName]">
      <xsl:value-of select="$varName"/>
    </xsl:when>
  </xsl:choose>

```

```

    <xsl:when test="ancestor::pgml:for/pgml:init/pgml:variable[@name=$varName]">
      <xsl:value-of select="$varName"/>
    </xsl:when>

    <!--get global variable-->
    <xsl:otherwise>
      this.<xsl:value-of select="$varName"/>
    </xsl:otherwise>
  </xsl:choose>
</xsl:template>

<xsl:template name="process-type-value">
  <xsl:choose>
    <!--String is special datatype, it needs to be quoted-->
    <xsl:when test="@type='String'">
      <xsl:text>&quot;</xsl:text><xsl:value-of select="@value"/><xsl:text>&quot;</xsl:text>
    </xsl:when>

    <xsl:otherwise>
      <xsl:value-of select="@value"/>
    </xsl:otherwise>
  </xsl:choose>
</xsl:template>

<xsl:template match="pgml:in">
  <xsl:for-each select="pgml:variable">
    <xsl:apply-templates select="."/>

    <xsl:if test="not(position()=last())">
      <xsl:text>,</xsl:text>
    </xsl:if>
  </xsl:for-each>
</xsl:template>

<xsl:template match="pgml:action">
  <xsl:for-each select="node()">
    <xsl:apply-templates select="."/>
    <xsl:text> </xsl:text>
  </xsl:for-each>
</xsl:template>

<xsl:template match="pgml:return">
  <xsl:text> return </xsl:text>
  <xsl:choose>
    <xsl:when test="@*">
      <xsl:choose>
        <xsl:when test="@select">
          <xsl:call-template name="get-local-or-global-variable">
            <xsl:with-param name="varName" select="@select"/>
          </xsl:call-template>
        </xsl:when>

        <xsl:otherwise>
          <xsl:call-template name="process-type-value"/>
        </xsl:otherwise>
      </xsl:choose>
    </xsl:when>

    <xsl:otherwise>
      <xsl:apply-templates/>
    </xsl:otherwise>
  </xsl:choose>
  <xsl:text>; </xsl:text>
</xsl:template>

```



```

<xsl:template match="pgml:for">
  for(<xsl:apply-templates select="pgml:init"/>; <xsl:apply-templates select="pgml:test"/>; <xsl:apply-templates select="pgml:step"/>){
    <xsl:apply-templates select="pgml:action"/>
  }
  <xsl:text>    </xsl:text>
</xsl:template>

```

```

<xsl:template match="pgml:init">
  <xsl:apply-templates/>
</xsl:template>

```

```

<xsl:template match="pgml:test">
  <xsl:apply-templates/>
</xsl:template>

```

```

<xsl:template match="pgml:step">
  <xsl:apply-templates/>
</xsl:template>

```

```

<xsl:template match="pgml:while">
  while(<xsl:apply-templates select="pgml:test"/>){
    <xsl:apply-templates select="pgml:action"/>
  }
  <xsl:text>    </xsl:text>
</xsl:template>

```

```

<xsl:template match="pgml:if">
  if(<xsl:apply-templates select="pgml:test"/>){
    <xsl:apply-templates select="pgml:action"/>
  }
  <xsl:if test="pgml:else">
    <xsl:apply-templates select="pgml:else"/>
  </xsl:if>
</xsl:template>

```



```

<xsl:template match="pgml:else">
  else{
    <xsl:for-each select="node()">
      <xsl:apply-templates select="."/>
      <xsl:text>    </xsl:text>
    </xsl:for-each>
  }
  <xsl:text>    </xsl:text>
</xsl:template>

```

```

<!--check if a math-statement is end-->
<xsl:template name="math-statement-end">
  <xsl:if test="not(name(..)='init' or name(..)='step' or name(..)='return' or name(..)='variable')">
    <xsl:text>; </xsl:text>
  </xsl:if>
</xsl:template>

```

```

<xsl:template match="pgml:add">
  <xsl:if test="@result">
    <xsl:call-template name="get-local-or-global-variable">
      <xsl:with-param name="varName" select="@result"/>
    </xsl:call-template>
    <xsl:text>=</xsl:text>
  </xsl:if>
</xsl:template>

```



```

</xsl:if>

<xsl:text>( </xsl:text>
<xsl:apply-templates select="*[1]"/>
<xsl:text> + </xsl:text>
<xsl:apply-templates select="*[2]"/>
<xsl:text> )</xsl:text>

<xsl:if test="@result">
  <xsl:call-template name="math-statement-end"/>
</xsl:if>
</xsl:template>

```

```

<xsl:template match="pgml:sub">
  <xsl:if test="@result">
    <xsl:call-template name="get-local-or-global-variable">
      <xsl:with-param name="varName" select="@result"/>
    </xsl:call-template>
    <xsl:text>=</xsl:text>
  </xsl:if>

  <xsl:text>( </xsl:text>
  <xsl:apply-templates select="*[1]"/>
  <xsl:text> - </xsl:text>
  <xsl:apply-templates select="*[2]"/>
  <xsl:text> )</xsl:text>

  <xsl:if test="@result">
    <xsl:call-template name="math-statement-end"/>
  </xsl:if>
</xsl:template>

```

```

<xsl:template match="pgml:mult">
  <xsl:if test="@result">
    <xsl:call-template name="get-local-or-global-variable">
      <xsl:with-param name="varName" select="@result"/>
    </xsl:call-template>
    <xsl:text>=</xsl:text>
  </xsl:if>

  <xsl:apply-templates select="*[1]"/>
  <xsl:text>*</xsl:text>
  <xsl:apply-templates select="*[2]"/>

  <xsl:if test="@result">
    <xsl:call-template name="math-statement-end"/>
  </xsl:if>
</xsl:template>

```



```

<xsl:template match="pgml:div">
  <xsl:if test="@result">
    <xsl:call-template name="get-local-or-global-variable">
      <xsl:with-param name="varName" select="@result"/>
    </xsl:call-template>
    <xsl:text>=</xsl:text>
  </xsl:if>

  <xsl:apply-templates select="*[1]"/>
  <xsl:text></xsl:text>
  <xsl:apply-templates select="*[2]"/>

  <xsl:if test="@result">
    <xsl:call-template name="math-statement-end"/>
  </xsl:if>
</xsl:template>

```

```

<xsl:template match="pgml:mod">

```

```

<xsl:if test="@result">
  <xsl:call-template name="get-local-or-global-variable">
    <xsl:with-param name="varName" select="@result"/>
  </xsl:call-template>
  <xsl:text>=</xsl:text>
</xsl:if>

<xsl:apply-templates select="*[1]"/>
<xsl:text>%</xsl:text>
<xsl:apply-templates select="*[2]"/>

<xsl:if test="@result">
  <xsl:call-template name="math-statement-end"/>
</xsl:if>
</xsl:template>

```

```

<xsl:template match="pgml:greater-than">
  <xsl:apply-templates select="*[1]"/>
  <xsl:text>&gt;</xsl:text>
  <xsl:apply-templates select="*[2]"/>
</xsl:template>

```

```

<xsl:template match="pgml:lesser-than">
  <xsl:apply-templates select="*[1]"/>
  <xsl:text>&lt;</xsl:text>
  <xsl:apply-templates select="*[2]"/>
</xsl:template>

```

```

<xsl:template match="pgml:equal-to">
  <xsl:apply-templates select="*[1]"/>
  <xsl:text>=</xsl:text>
  <xsl:apply-templates select="*[2]"/>
</xsl:template>

```



```

<xsl:template match="pgml:not-equal">
  <xsl:apply-templates select="*[1]"/>
  <xsl:text>!=</xsl:text>
  <xsl:apply-templates select="*[2]"/>
</xsl:template>

```

```

<xsl:template match="pgml:greater-or-equal">
  <xsl:apply-templates select="*[1]"/>
  <xsl:text>&gt;=</xsl:text>
  <xsl:apply-templates select="*[2]"/>
</xsl:template>

```

```

<xsl:template match="pgml:lesser-or-equal">
  <xsl:apply-templates select="*[1]"/>
  <xsl:text>&lt;=</xsl:text>
  <xsl:apply-templates select="*[2]"/>
</xsl:template>

```

```

<xsl:template match="pgml:and">
  <xsl:text> (</xsl:text>
  <xsl:apply-templates select="*[1]"/>
  <xsl:text> &amp;&amp; </xsl:text>
  <xsl:apply-templates select="*[2]"/>
  <xsl:text> )</xsl:text>
</xsl:template>

```

```

<xsl:template match="pgml:or">
  <xsl:text> (</xsl:text>
  <xsl:apply-templates select="*[1]"/>
  <xsl:text> || </xsl:text>
  <xsl:apply-templates select="*[2]"/>
  <xsl:text>) </xsl:text>
</xsl:template>

```

```

<xsl:template match="pgml:not">
  <xsl:text> !(</xsl:text>
  <xsl:apply-templates/>
  <xsl:text>) </xsl:text>
</xsl:template>

```

```

<xsl:template match="pgml:assign">
  <xsl:call-template name="get-local-or-global-variable">
    <xsl:with-param name="varName" select="@arg1"/>
  </xsl:call-template>
  <xsl:text>=</xsl:text>

  <xsl:choose>
    <xsl:when test="@arg2">
      <xsl:call-template name="get-local-or-global-variable">
        <xsl:with-param name="varName" select="@arg2"/>
      </xsl:call-template>
    </xsl:when>

    <xsl:otherwise>
      <xsl:apply-templates/>
    </xsl:otherwise>
  </xsl:choose>

  <xsl:if test="not(name(..)='init' or name(..)='step')">
    <xsl:text>; </xsl:text>
  </xsl:if>
</xsl:template>

```



```

<xsl:template match="pgml:call-method">
  <xsl:value-of select="@name"/>
  <xsl:text>(</xsl:text>

  <xsl:for-each select="*">
    <xsl:apply-templates select="."/>
    <xsl:if test="not(position()=last())">
      <xsl:text>, </xsl:text>
    </xsl:if>
  </xsl:for-each>

  <xsl:text>)</xsl:text>

  <xsl:if test="name(..)='action' or name(..)='else'">
    <xsl:text>; </xsl:text>
  </xsl:if>
</xsl:template>

```

```

<xsl:template match="pgml:lib">
  <!--finding out corresponding method in J2ME-->
  <xsl:apply-imports/>

  <xsl:if test="name(..)='action' or name(..)='else'">
    <xsl:text>; </xsl:text>
  </xsl:if>
</xsl:template>

```

```

</xsl:stylesheet>

```

## [ For PGML – PGMLStdLib2MIDP.xml (indented)]

```
<?xml version="1.0"?>
```

```
<!--
```

1. this is an XSLT stylesheet of PGML standard library 1.0 to transform PGML standard library into corresponding J2ME code.
2. PGMLStdLib2MIDP.xml is imported by PGML2MIDP.xml

```
-->
```

```
<xsl:stylesheet version="1.0" xmlns:xsl="http://www.w3.org/1999/XSL/Transform" xmlns:pgml="http://dcs.w3.cis.nctu.edu.tw/Project/Pervasive/PGML/">
```

```
<!--because MIDP is a text document-->
```

```
<xsl:output method="text"/>
```

```
<xsl:template match="pgml:lib">
```

```
<xsl:choose>
```

```
<xsl:when test="@call='pgml.Integer.toString'">
```

```
<xsl:if test="not(count(*)=1)">
```

```
<xsl:message terminate="yes">
```

```
in method named &quot;<xsl:value-of select="ancestor::pgml:method/@name"/>&quot;, &lt;lib&gt; has wrong number of parameters!
```

```
&quot;<xsl:value-of select="@call"/>&quot; must have one parameter.
```

```
</xsl:message>
```

```
</xsl:if>
```

```
<xsl:text>java.lang.Integer.toString</xsl:text>
```

```
<xsl:call-template name="tail-part"/>
```

```
</xsl:when>
```

```
<xsl:when test="@call='pgml.Integer.abs'">
```

```
<xsl:if test="not(count(*)=1)">
```

```
<xsl:message terminate="yes">
```

```
in method named &quot;<xsl:value-of select="ancestor::pgml:method/@name"/>&quot;, &lt;lib&gt; has wrong number of parameters!
```

```
&quot;<xsl:value-of select="@call"/>&quot; must have one parameter.
```

```
</xsl:message>
```

```
</xsl:if>
```

```
<xsl:text>java.lang.Math.abs</xsl:text>
```

```
<xsl:call-template name="tail-part"/>
```

```
</xsl:when>
```

```
<xsl:when test="@call='pgml.Integer.max'">
```

```
<xsl:if test="not(count(*)=2)">
```

```
<xsl:message terminate="yes">
```

```
in method named &quot;<xsl:value-of select="ancestor::pgml:method/@name"/>&quot;, &lt;lib&gt; has wrong number of parameters!
```

```
&quot;<xsl:value-of select="@call"/>&quot; must have two parameters.
```

```
</xsl:message>
```

```
</xsl:if>
```

```
<xsl:text>java.lang.Math.max</xsl:text>
```

```
<xsl:call-template name="tail-part"/>
```

```
</xsl:when>
```

```
<xsl:when test="@call='pgml.Integer.min'">
```

```
<xsl:if test="not(count(*)=2)">
```

```
<xsl:message terminate="yes">
```

```
in method named &quot;<xsl:value-of select="ancestor::pgml:method/@name"/>&quot;, &lt;lib&gt; has wrong number of parameters!
```

```
&quot;<xsl:value-of select="@call"/>&quot; must have two parameters.
```

```
</xsl:message>
```

```
</xsl:if>
```

```
<xsl:text>java.lang.Math.min</xsl:text>
```

```
<xsl:call-template name="tail-part"/>
```

```
</xsl:when>
```

```
<xsl:when test="@call='pgml.Integer.maxInt'">
```

```
<xsl:if test="*">
```

```
<xsl:message terminate="yes">
```

```
in method named &quot;<xsl:value-of select="ancestor::pgml:method/@name"/>&quot;, &lt;lib&gt;
```

```

        &lt;lib&gt; has wrong number of parameters!
        &quot;<xsl:value-of select="@call"/&quot; must have no any parameter.
    </xsl:message>
</xsl:if>

    <xsl:text>java.lang.Integer.MAX_VALUE</xsl:text>
</xsl:when>

<xsl:when test="@call='pgml.Integer.minInt'">
    <xsl:if test="*">
        <xsl:message terminate="yes">
            in method named &quot;<xsl:value-of select="ancestor::pgml:method/@name"/&quot;,
            &lt;lib&gt; has wrong number of parameters!
            &quot;<xsl:value-of select="@call"/&quot; must have no any parameter.
        </xsl:message>
    </xsl:if>

    <xsl:text>java.lang.Integer.MIN_VALUE</xsl:text>
</xsl:when>

<xsl:when test="@call='pgml.String.parseInt'">
    <xsl:if test="not(count(*)=1)">
        <xsl:message terminate="yes">
            in method named &quot;<xsl:value-of select="ancestor::pgml:method/@name"/&quot;,
            &lt;lib&gt; has wrong number of parameters!
            &quot;<xsl:value-of select="@call"/&quot; must have one parameter.
        </xsl:message>
    </xsl:if>

    <xsl:text>java.lang.Integer.parseInt</xsl:text>
    <xsl:call-template name="tail-part"/>
</xsl:when>

<xsl:when test="@call='pgml.String.charAt'">
    <xsl:if test="not(count(*)=2)">
        <xsl:message terminate="yes">
            in method named &quot;<xsl:value-of select="ancestor::pgml:method/@name"/&quot;,
            &lt;lib&gt; has wrong number of parameters!
            &quot;<xsl:value-of select="@call"/&quot; must have two parameters.
        </xsl:message>
    </xsl:if>

    String.valueOf(<xsl:apply-templates select="*[1]"/>.charAt(<xsl:apply-templates select="*[2]"/>))
</xsl:when>

<xsl:when test="@call='pgml.String.compare'">
    <xsl:if test="not(count(*)=2)">
        <xsl:message terminate="yes">
            in method named &quot;<xsl:value-of select="ancestor::pgml:method/@name"/&quot;,
            &lt;lib&gt; has wrong number of parameters!
            &quot;<xsl:value-of select="@call"/&quot; must have two parameters.
        </xsl:message>
    </xsl:if>

    <xsl:apply-templates select="*[1]"/>.compareTo(<xsl:apply-templates select="*[2]"/>)
</xsl:when>

<xsl:when test="@call='pgml.String.length'">
    <xsl:if test="not(count(*)=1)">
        <xsl:message terminate="yes">
            in method named &quot;<xsl:value-of select="ancestor::pgml:method/@name"/&quot;,
            &lt;lib&gt; has wrong number of parameters!
            &quot;<xsl:value-of select="@call"/&quot; must have one parameter.
        </xsl:message>
    </xsl:if>

    <xsl:apply-templates select="*[1]"/>.length()
</xsl:when>

<xsl:when test="@call='pgml.String.substring'">
    <xsl:if test="not(count(*)=3)">
        <xsl:message terminate="yes">
            in method named &quot;<xsl:value-of select="ancestor::pgml:method/@name"/&quot;,
            &lt;lib&gt; has wrong number of parameters!
            &quot;<xsl:value-of select="@call"/&quot; must have three parameter.
    </xsl:if>

```

```

        </xsl:message>
    </xsl:if>

    <xsl:apply-templates select="*[1]"/>.subString(<xsl:apply-templates select="*[2]"/>, <xsl:apply-templates
select="*[2]"/>+<xsl:apply-templates select="*[3]"/>)
    </xsl:when>

    <xsl:when test="@call='pgml.String.trim'">
        <xsl:if test="not(count(*)=1)">
            <xsl:message terminate="yes">
                in method named &quot;<xsl:value-of select="ancestor::pgml:method/@name"/>&quot;,
                &lt;lib&gt; has wrong number of parameters!
                &quot;<xsl:value-of select="@call"/>&quot; must have one parameter.
            </xsl:message>
        </xsl:if>

        <xsl:apply-templates select="*[1]"/>.trim()
    </xsl:when>

    <xsl:when test="@call='pgml.String.concat'">
        <xsl:if test="not(count(*)=2)">
            <xsl:message terminate="yes">
                in method named &quot;<xsl:value-of select="ancestor::pgml:method/@name"/>&quot;,
                &lt;lib&gt; has wrong number of parameters!
                &quot;<xsl:value-of select="@call"/>&quot; must have two parameter.
            </xsl:message>
        </xsl:if>

        <xsl:apply-templates select="*[1]"/>.concat(<xsl:apply-templates select="*[2]"/>)
    </xsl:when>

    <xsl:when test="@call='pgml.Boolean.toString'">
        <xsl:if test="not(count(*)=1)">
            <xsl:message terminate="yes">
                in method named &quot;<xsl:value-of select="ancestor::pgml:method/@name"/>&quot;,
                &lt;lib&gt; has wrong number of parameters!
                &quot;<xsl:value-of select="@call"/>&quot; must have one parameter.
            </xsl:message>
        </xsl:if>

        new Boolean(<xsl:apply-templates select="*[1]"/>).toString()
    </xsl:when>

    <xsl:otherwise>
        <xsl:message terminate="yes">
            in method named &quot;<xsl:value-of select="ancestor::pgml:method/@name"/>&quot;,
            &lt;lib&gt; has wrong library function call!
            you use a non-existing PGML standard library : <xsl:value-of select="@call"/>
        </xsl:message>
    </xsl:otherwise>
</xsl:choose>
</xsl:template>

<xsl:template name="tail-part">
    <!--passing needed parameters into method-->
    <xsl:text></xsl:text>

    <xsl:for-each select="*">
        <xsl:apply-templates select="."/>

        <xsl:if test="not(position()=last())">
            <xsl:text>, </xsl:text>
        </xsl:if>
    </xsl:for-each>

    <xsl:text></xsl:text>
</xsl:template>
</xsl:stylesheet>

```

## [ For PGML – PGML2WMLS.xsl (indented)]

```
<?xml version="1.0"?>
```

```
<!--this is an XSLT stylesheet of PGML 1.0 to transform PGML into WMLScript without error control.-->
```

```
<xsl:stylesheet version="1.0" xmlns:xsl="http://www.w3.org/1999/XSL/Transform"
xmlns:pgml="http://dcs.w3.cis.nctu.edu.tw/Project/Pervasive/PGML/">
```

```
  <!--for PGML standard library-->
```

```
  <xsl:import href="PGMLStdLib2WMLS.xsl"/>
```

```
  <!--because MIDP is a text document-->
```

```
  <xsl:output method="text"/>
```

```
  <xsl:template match="/">
```

```
    <xsl:text>/*some pragma are declared here...*/</xsl:text>
```

```
    <xsl:text>    </xsl:text>
```

```
    <xsl:apply-templates select="pgml:object"/>
```

```
  </xsl:template>
```

```
  <xsl:template match="pgml:object">
```

```
    /*WMLScript interface to set declared global variables*/
```

```
    extern function globalVarDeclaration() {
```

```
      /*setting all global variables in this logic object*/
```

```
      <xsl:apply-templates select="pgml:declaration"/>
```

```
    }
```

```
    /*all methods are here...*/
```

```
    <xsl:for-each select="pgml:method">
```

```
      <xsl:apply-templates select="."/>
```

```
      <xsl:text>    </xsl:text>
```

```
    </xsl:for-each>
```

```
    /*WMLScript interface to change value of UI component on some WML card*/
```

```
    <xsl:call-template name="change-fuction-interface"/>
```

```
  </xsl:template>
```

```
  <xsl:template match="pgml:declaration">
```

```
    <xsl:choose>
```

```
      <xsl:when test="name(.)='object'">
```

```
        <xsl:call-template name="global-variables-setting"/>
```

```
      </xsl:when>
```

```
      <xsl:when test="name(.)='method'">
```

```
        <xsl:for-each select="pgml:variable">
```

```
          <xsl:apply-templates select="."/>
```

```
          <xsl:text>    </xsl:text>
```

```
        </xsl:for-each>
```

```
      </xsl:when>
```

```
    </xsl:choose>
```

```
  </xsl:template>
```

```
  <xsl:template match="pgml:method">
```

```
    <xsl:choose>
```

```
      <xsl:when test="@visibility='public'">
```

```
        <xsl:text>extern function </xsl:text>
```

```
      </xsl:when>
```

```
      <xsl:when test="@visibility='private'">
```

```
        <xsl:text>function </xsl:text>
```

```
      </xsl:when>
```

```
    </xsl:choose>
```

```
  </xsl:template>
```



```

    <xsl:value-of select="@name"/> (<xsl:apply-templates select="pgml:in"/>){
      <xsl:apply-templates select="pgml:declaration"/>

      <xsl:apply-templates select="pgml:action"/>
    }
  <xsl:text> </xsl:text>
</xsl:template>

```

```

<xsl:template name="global-variables-setting">
  <xsl:for-each select="pgml:variable">
    <xsl:choose>
      <xsl:when test="node()">
        WMLBrowser.setVar(&quot;<xsl:value-of select="/pgml:object/@name"/>_<xsl:value-of
select="@name"/>&quot;; <xsl:apply-templates/>);
      </xsl:when>

      <xsl:otherwise>
        <xsl:choose>
          <xsl:when test="@value">
            WMLBrowser.setVar(&quot;<xsl:value-of select="/pgml:object/@name"/>_<xsl:value-of
select="@name"/>&quot;; <xsl:call-template name="process-type-value"/>);
          </xsl:when>

          <xsl:otherwise>
            WMLBrowser.setVar(&quot;<xsl:value-of select="/pgml:object/@name"/>_<xsl:value-of
select="@name"/>&quot;; &quot;&quot;);
          </xsl:otherwise>
        </xsl:choose>
      </xsl:choose>
    </xsl:for-each>
  <xsl:text> </xsl:text>
</xsl:template>

```



```

<xsl:template match="pgml:variable">
  <xsl:choose>
    <xsl:when test="name(.)='in'">
      <xsl:value-of select="@name"/>
    </xsl:when>

    <xsl:otherwise>
      <xsl:text>var </xsl:text><xsl:value-of select="@name"/>

      <xsl:choose>
        <xsl:when test="node()">
          <xsl:text>=</xsl:text>
          <xsl:apply-templates/>
        </xsl:when>

        <xsl:otherwise>
          <xsl:if test="@value">
            <xsl:text>=</xsl:text>
            <xsl:call-template name="process-type-value"/>
          </xsl:if>
        </xsl:otherwise>
      </xsl:choose>

      <xsl:if test="not(name(.)='init' or name(.)='in')">
        <xsl:text>; </xsl:text>
      </xsl:if>
    </xsl:otherwise>
  </xsl:choose>
</xsl:template>

```

```

<xsl:template match="pgml:operand">
  <xsl:choose>
    <xsl:when test="@select">
      <xsl:if test="name(.)='add' or name(.)='sub' or name(.)='mult' or name(.)='div' or name(.)='mod'">

```



```

        <xsl:text> Lang.parseInt(</xsl:text>
    </xsl:if>

    <xsl:call-template name="get-local-or-global-variable">
        <xsl:with-param name="varName" select="@select"/>
    </xsl:call-template>

    <xsl:if test="name(..)='add' or name(..)='sub' or name(..)='mult' or name(..)='div' or name(..)='mod'">
        <xsl:text> </xsl:text>
    </xsl:if>
</xsl:when>

<xsl:otherwise>
    <xsl:call-template name="process-type-value"/>
</xsl:otherwise>
</xsl:choose>
</xsl:template>

<xsl:template name="get-local-or-global-variable">
    <xsl:param name="varName">defaultVarName</xsl:param>

    <xsl:choose>
        <!--get local variable-->
        <xsl:when test="ancestor::pgml:method/pgml:declaration/pgml:variable[@name=$varName]">
            <xsl:value-of select="$varName"/>
        </xsl:when>

        <xsl:when test="ancestor::pgml:method/pgml:in/pgml:variable[@name=$varName]">
            <xsl:value-of select="$varName"/>
        </xsl:when>

        <xsl:when test="ancestor::pgml:for/pgml:init/pgml:variable[@name=$varName]">
            <xsl:value-of select="$varName"/>
        </xsl:when>

        <!--get global variable-->
        <xsl:otherwise>
            WMLBrowser.getVar(&quot;<xsl:value-of select="/pgml:object[@name=">_<xsl:value-of select="$varName"/>&quot;)&quot;);
        </xsl:otherwise>
    </xsl:choose>
</xsl:template>

<xsl:template name="process-type-value">
    <xsl:choose>
        <!--String is speial datatype, it need to be quoted-->
        <xsl:when test="@type='String'">
            <xsl:text>&quot;<xsl:text><xsl:value-of select="@value"/><xsl:text>&quot;</xsl:text>
        </xsl:when>

        <xsl:otherwise>
            <xsl:value-of select="@value"/>
        </xsl:otherwise>
    </xsl:choose>
</xsl:template>

<xsl:template match="pgml:in">
    <xsl:for-each select="pgml:variable">
        <xsl:apply-templates select="."/>

        <xsl:if test="not(position()=last())">
            <xsl:text>, </xsl:text>
        </xsl:if>
    </xsl:for-each>
</xsl:template>

<xsl:template match="pgml:action">
    <xsl:for-each select="node()">

```



```

        <xsl:apply-templates select="."/>
        <xsl:text>    </xsl:text>
    </xsl:for-each>
</xsl:template>

```

```

<xsl:template match="pgml:return">
    <xsl:text> return </xsl:text>
    <xsl:choose>
        <xsl:when test="@*">
            <xsl:choose>
                <xsl:when test="@select">
                    <xsl:call-template name="get-local-or-global-variable">
                        <xsl:with-param name="varName" select="@select"/>
                    </xsl:call-template>
                </xsl:when>

                <xsl:otherwise>
                    <xsl:call-template name="process-type-value"/>
                </xsl:otherwise>
            </xsl:choose>
        </xsl:when>

        <xsl:otherwise>
            <xsl:apply-templates/>
        </xsl:otherwise>
    </xsl:choose>
    <xsl:text>; </xsl:text>
</xsl:template>

```

```

<xsl:template match="pgml:for">
    for(<xsl:apply-templates select="pgml:init"/>; <xsl:apply-templates select="pgml:test"/>; <xsl:apply-templates select="pgml:step"/>){
        <xsl:apply-templates select="pgml:action"/>
    }
    <xsl:text>    </xsl:text>
</xsl:template>

```



```

<xsl:template match="pgml:init">
    <xsl:apply-templates/>
</xsl:template>

```

```

<xsl:template match="pgml:test">
    <xsl:apply-templates/>
</xsl:template>

```

```

<xsl:template match="pgml:step">
    <xsl:apply-templates/>
</xsl:template>

```

```

<xsl:template match="pgml:while">
    while(<xsl:apply-templates select="pgml:test"/>){
        <xsl:apply-templates select="pgml:action"/>
    }
    <xsl:text>    </xsl:text>
</xsl:template>

```

```

<xsl:template match="pgml:if">
    if(<xsl:apply-templates select="pgml:test"/>){
        <xsl:apply-templates select="pgml:action"/>
    }

    <xsl:if test="pgml:else">

```

```

        <xsl:apply-templates select="pgml:else"/>
    </xsl:if>
</xsl:template>

```

```

<xsl:template match="pgml:else">
    else{
        <xsl:for-each select="node()">
            <xsl:apply-templates select="."/>
            <xsl:text> </xsl:text>
        </xsl:for-each>
    }
    <xsl:text> </xsl:text>
</xsl:template>

```

```

<!--check if a math-statement is end-->
<xsl:template name="math-statement-end">
    <xsl:if test="not(name(..)='init' or name(..)='step' or name(..)='return' or name(..)='variable')">
        <xsl:text>; </xsl:text>
    </xsl:if>
</xsl:template>

```

```

<xsl:template name="set-local-or-global-variable-head">
    <xsl:param name="varName">defaultVarName</xsl:param>

    <xsl:choose>
        <!--local variable-->
        <xsl:when test="ancestor::pgml:method/pgml:declaration/pgml:variable[@name=$varName]">
            <xsl:value-of select="$varName"/>
            <xsl:text>=</xsl:text>
        </xsl:when>

        <xsl:when test="ancestor::pgml:method/pgml:in/pgml:variable[@name=$varName]">
            <xsl:value-of select="$varName"/>
            <xsl:text>=</xsl:text>
        </xsl:when>

        <xsl:when test="ancestor::pgml:for/pgml:init/pgml:variable[@name=$varName]">
            <xsl:value-of select="$varName"/>
            <xsl:text>=</xsl:text>
        </xsl:when>

        <!--global variable-->
        <xsl:otherwise>
            <xsl:text>WMLBrowser.setVar(&quot;</xsl:text><xsl:value-of select="/pgml:object/@name"/>_<xsl:value-of
select="$varName"/><xsl:text>&quot;; </xsl:text>
        </xsl:otherwise>
    </xsl:choose>
</xsl:template>

```

```

<xsl:template name="set-local-or-global-variable-tail">
    <xsl:param name="varName">defaultVarName</xsl:param>

    <!--global variable-->
    <xsl:if test="not(ancestor::pgml:method/pgml:declaration/pgml:variable[@name=$varName])">
        <xsl:if test="not(ancestor::pgml:method/pgml:in/pgml:variable[@name=$varName])">
            <xsl:if test="not(ancestor::pgml:for/pgml:init/pgml:variable[@name=$varName])">
                <xsl:text> )</xsl:text>
            </xsl:if>
        </xsl:if>
    </xsl:if>
</xsl:template>

```

```

<xsl:template match="pgml:add">
    <xsl:if test="@result">
        <xsl:call-template name="set-local-or-global-variable-head">

```

```

        <xsl:with-param name="varName" select="@result"/>
    </xsl:call-template>
</xsl:if>

<xsl:text>( </xsl:text>
<xsl:apply-templates select="*[1]"/>
<xsl:text> + </xsl:text>
<xsl:apply-templates select="*[2]"/>
<xsl:text> )</xsl:text>

<xsl:if test="@result">
    <xsl:call-template name="set-local-or-global-variable-tail">
        <xsl:with-param name="varName" select="@result"/>
    </xsl:call-template>

    <xsl:call-template name="math-statement-end"/>
</xsl:if>
</xsl:template>

```

```

<xsl:template match="pgml:sub">
    <xsl:if test="@result">
        <xsl:call-template name="set-local-or-global-variable-head">
            <xsl:with-param name="varName" select="@result"/>
        </xsl:call-template>
    </xsl:if>

    <xsl:text>( </xsl:text>
    <xsl:apply-templates select="*[1]"/>
    <xsl:text> - </xsl:text>
    <xsl:apply-templates select="*[2]"/>
    <xsl:text> )</xsl:text>

    <xsl:if test="@result">
        <xsl:call-template name="set-local-or-global-variable-tail">
            <xsl:with-param name="varName" select="@result"/>
        </xsl:call-template>

        <xsl:call-template name="math-statement-end"/>
    </xsl:if>
</xsl:template>

```



```

<xsl:template match="pgml:mult">
    <xsl:if test="@result">
        <xsl:call-template name="set-local-or-global-variable-head">
            <xsl:with-param name="varName" select="@result"/>
        </xsl:call-template>
    </xsl:if>

    <xsl:apply-templates select="*[1]"/>
    <xsl:text>*</xsl:text>
    <xsl:apply-templates select="*[2]"/>

    <xsl:if test="@result">
        <xsl:call-template name="set-local-or-global-variable-tail">
            <xsl:with-param name="varName" select="@result"/>
        </xsl:call-template>

        <xsl:call-template name="math-statement-end"/>
    </xsl:if>
</xsl:template>

```

```

<xsl:template match="pgml:div">
    <xsl:if test="@result">
        <xsl:call-template name="set-local-or-global-variable-head">
            <xsl:with-param name="varName" select="@result"/>
        </xsl:call-template>
    </xsl:if>

    <xsl:apply-templates select="*[1]"/>

```

```

<xsl:text>/</xsl:text>
<xsl:apply-templates select="*[2]"/>

<xsl:if test="@result">
  <xsl:call-template name="set-local-or-global-variable-tail">
    <xsl:with-param name="varName" select="@result"/>
  </xsl:call-template>

  <xsl:call-template name="math-statement-end"/>
</xsl:if>
</xsl:template>

```

```

<xsl:template match="pgml:mod">
  <xsl:if test="@result">
    <xsl:call-template name="set-local-or-global-variable-head">
      <xsl:with-param name="varName" select="@result"/>
    </xsl:call-template>
  </xsl:if>

  <xsl:apply-templates select="*[1]"/>
  <xsl:text>%</xsl:text>
  <xsl:apply-templates select="*[2]"/>

  <xsl:if test="@result">
    <xsl:call-template name="set-local-or-global-variable-tail">
      <xsl:with-param name="varName" select="@result"/>
    </xsl:call-template>

    <xsl:call-template name="math-statement-end"/>
  </xsl:if>
</xsl:template>

```

```

<xsl:template match="pgml:greater-than">
  <xsl:apply-templates select="*[1]"/>
  <xsl:text>&gt;</xsl:text>
  <xsl:apply-templates select="*[2]"/>
</xsl:template>

```



```

<xsl:template match="pgml:lesser-than">
  <xsl:apply-templates select="*[1]"/>
  <xsl:text>&lt;</xsl:text>
  <xsl:apply-templates select="*[2]"/>
</xsl:template>

```

```

<xsl:template match="pgml:equal-to">
  <xsl:apply-templates select="*[1]"/>
  <xsl:text>=</xsl:text>
  <xsl:apply-templates select="*[2]"/>
</xsl:template>

```

```

<xsl:template match="pgml:not-equal">
  <xsl:apply-templates select="*[1]"/>
  <xsl:text>!=</xsl:text>
  <xsl:apply-templates select="*[2]"/>
</xsl:template>

```

```

<xsl:template match="pgml:greater-or-equal">
  <xsl:apply-templates select="*[1]"/>
  <xsl:text>&gt;=</xsl:text>
  <xsl:apply-templates select="*[2]"/>
</xsl:template>

```

```

<xsl:template match="pgml:lesser-or-equal">
  <xsl:apply-templates select="*[1]"/>
  <xsl:text>&lt;=</xsl:text>
  <xsl:apply-templates select="*[2]"/>
</xsl:template>

```

```

<xsl:template match="pgml:and">
  <xsl:text> (</xsl:text>
  <xsl:apply-templates select="*[1]"/>
  <xsl:text> &amp;&amp; </xsl:text>
  <xsl:apply-templates select="*[2]"/>
  <xsl:text>) </xsl:text>
</xsl:template>

```

```

<xsl:template match="pgml:or">
  <xsl:text> (</xsl:text>
  <xsl:apply-templates select="*[1]"/>
  <xsl:text> || </xsl:text>
  <xsl:apply-templates select="*[2]"/>
  <xsl:text>) </xsl:text>
</xsl:template>

```

```

<xsl:template match="pgml:not">
  <xsl:text> !(</xsl:text>
  <xsl:apply-templates/>
  <xsl:text>) </xsl:text>
</xsl:template>

```

```

<xsl:template match="pgml:assign">
  <xsl:call-template name="set-local-or-global-variable-head">
    <xsl:with-param name="varName" select="@arg1"/>
  </xsl:call-template>

  <xsl:choose>
    <xsl:when test="@arg2">
      <xsl:call-template name="get-local-or-global-variable">
        <xsl:with-param name="varName" select="@arg2"/>
      </xsl:call-template>
    </xsl:when>

    <xsl:otherwise>
      <xsl:apply-templates/>
    </xsl:otherwise>
  </xsl:choose>

  <xsl:call-template name="set-local-or-global-variable-tail">
    <xsl:with-param name="varName" select="@arg1"/>
  </xsl:call-template>

  <xsl:if test="not(name(..)='init' or name(..)='step')">
    <xsl:text>; </xsl:text>
  </xsl:if>
</xsl:template>

```



```

<xsl:template match="pgml:call-method">
  <xsl:if test="name(..)='add' or name(..)='sub' or name(..)='mult' or name(..)='div' or name(..)='mod'">
    <xsl:text> Lang.parseInt(</xsl:text>
  </xsl:if>

  <xsl:value-of select="@name"/>
  <xsl:text>(</xsl:text>

  <xsl:for-each select="*">

```

```

        <xsl:apply-templates select="."/>
        <xsl:if test="not(position()=last())">
            <xsl:text>, </xsl:text>
        </xsl:if>
    </xsl:for-each>

    <xsl:text></xsl:text>

    <xsl:if test="name(.)='add' or name(.)='sub' or name(.)='mult' or name(.)='div' or name(.)='mod'">
        <xsl:text> </xsl:text>
    </xsl:if>

    <xsl:if test="name(.)='action' or name(.)='else'">
        <xsl:text>; </xsl:text>
    </xsl:if>
</xsl:template>

```

```

<xsl:template match="pgml:lib">
    <xsl:if test="name(.)='add' or name(.)='sub' or name(.)='mult' or name(.)='div' or name(.)='mod'">
        <xsl:text> Lang.parseInt(</xsl:text>
    </xsl:if>

```

```

<!--finding out corresponding standard library in WMLScript-->
<xsl:apply-imports/>

```

```

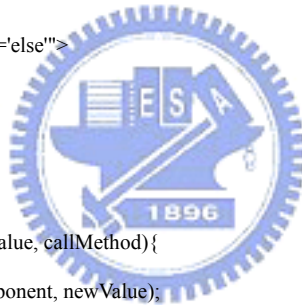
<xsl:if test="name(.)='add' or name(.)='sub' or name(.)='mult' or name(.)='div' or name(.)='mod'">
    <xsl:text> </xsl:text>
</xsl:if>

```

```

<xsl:if test="name(.)='action' or name(.)='else'">
    <xsl:text>; </xsl:text>
</xsl:if>
</xsl:template>

```



```

<xsl:template name="change-fuction-interface">
    extern function change(component, newValue, callMethod){
        if(callMethod == ""){
            WMLBrowser.setVar(component, newValue);
        }
        else{
            <xsl:for-each select="pgml:method">
                <xsl:if test="not(pgml:in)">
                    if(callMethod == &quot;<xsl:value-of select="@name"/>&quot;){
                        WMLBrowser.setVar(component, <xsl:value-of select="@name"/>());
                    }
                <xsl:text> </xsl:text>
            </xsl:if>
        </xsl:for-each>
    }
}
<xsl:text> </xsl:text>
</xsl:template>

```

```

</xsl:stylesheet>

```

## [ For PGML – PGMLStdLib2WMLS.xml (indented) ]

```
<?xml version="1.0"?>
```

```
<!--
```

1. this is an XSLT stylesheet of PGML standard library 1.0 to transform PGML standard library into corresponding WMLScript code.
2. PGMLStdLib2WMLS.xml is imported by PGML2WMLS.xml

```
-->
```

```
<xsl:stylesheet version="1.0" xmlns:xsl="http://www.w3.org/1999/XSL/Transform" xmlns:pgml="http://dcsw3.cis.nctu.edu.tw/Project/Pervasive/PGML/">
```

```
<!--because MIDP is a text document-->
```

```
<xsl:output method="text"/>
```

```
<xsl:template match="pgml:lib">
```

```
<xsl:choose>
```

```
<xsl:when test="@call='pgml.Integer.toString'">
```

```
<xsl:if test="not(count(*)=1)">
```

```
<xsl:message terminate="yes">
```

```
in method named &quot;<xsl:value-of select="ancestor::pgml:method/@name"/>&quot;, &lt;lib&gt; has wrong number of parameters!
```

```
&quot;<xsl:value-of select="@call"/>&quot; must have one parameter.
```

```
</xsl:message>
```

```
</xsl:if>
```

```
<xsl:text>String.toString</xsl:text>
```

```
<xsl:call-template name="tail-part"/>
```

```
</xsl:when>
```

```
<xsl:when test="@call='pgml.Integer.abs'">
```

```
<xsl:if test="not(count(*)=1)">
```

```
<xsl:message terminate="yes">
```

```
in method named &quot;<xsl:value-of select="ancestor::pgml:method/@name"/>&quot;, &lt;lib&gt; has wrong number of parameters!
```

```
&quot;<xsl:value-of select="@call"/>&quot; must have one parameter.
```

```
</xsl:message>
```

```
</xsl:if>
```

```
<xsl:text>Lang.abs</xsl:text>
```

```
<xsl:call-template name="tail-part"/>
```

```
</xsl:when>
```

```
<xsl:when test="@call='pgml.Integer.max'">
```

```
<xsl:if test="not(count(*)=2)">
```

```
<xsl:message terminate="yes">
```

```
in method named &quot;<xsl:value-of select="ancestor::pgml:method/@name"/>&quot;, &lt;lib&gt; has wrong number of parameters!
```

```
&quot;<xsl:value-of select="@call"/>&quot; must have two parameters.
```

```
</xsl:message>
```

```
</xsl:if>
```

```
<xsl:text>Lang.max</xsl:text>
```

```
<xsl:call-template name="tail-part"/>
```

```
</xsl:when>
```

```
<xsl:when test="@call='pgml.Integer.min'">
```

```
<xsl:if test="not(count(*)=2)">
```

```
<xsl:message terminate="yes">
```

```
in method named &quot;<xsl:value-of select="ancestor::pgml:method/@name"/>&quot;, &lt;lib&gt; has wrong number of parameters!
```

```
&quot;<xsl:value-of select="@call"/>&quot; must have two parameters.
```

```
</xsl:message>
```

```
</xsl:if>
```

```
<xsl:text>Lang.min</xsl:text>
```

```
<xsl:call-template name="tail-part"/>
```

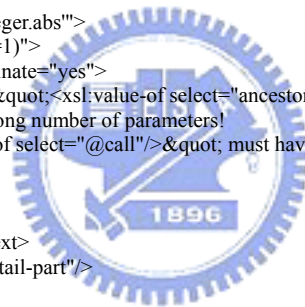
```
</xsl:when>
```

```
<xsl:when test="@call='pgml.Integer.maxInt'">
```

```
<xsl:if test="*">
```

```
<xsl:message terminate="yes">
```

```
in method named &quot;<xsl:value-of select="ancestor::pgml:method/@name"/>&quot;, &lt;lib&gt;
```





```

        &lt;lib&gt; has wrong number of parameters!
        &quot;<xsl:value-of select="@call"/&quot; must have no any parameter.
    </xsl:message>
</xsl:if>

    <xsl:text>Lang.maxInt()</xsl:text>
</xsl:when>

<xsl:when test="@call='pgml.Integer.minInt'">
    <xsl:if test="*">
        <xsl:message terminate="yes">
            in method named &quot;<xsl:value-of select="ancestor::pgml.method/@name"/&quot;,
            &lt;lib&gt; has wrong number of parameters!
            &quot;<xsl:value-of select="@call"/&quot; must have no any parameter.
        </xsl:message>
    </xsl:if>

    <xsl:text>Lang.minInt()</xsl:text>
</xsl:when>

<xsl:when test="@call='pgml.String.parseInt'">
    <xsl:if test="not(count(*)=1)">
        <xsl:message terminate="yes">
            in method named &quot;<xsl:value-of select="ancestor::pgml.method/@name"/&quot;,
            &lt;lib&gt; has wrong number of parameters!
            &quot;<xsl:value-of select="@call"/&quot; must have one parameter.
        </xsl:message>
    </xsl:if>

    <xsl:text>Lang.parseInt</xsl:text>
    <xsl:call-template name="tail-part"/>
</xsl:when>

<xsl:when test="@call='pgml.String.charAt'">
    <xsl:if test="not(count(*)=2)">
        <xsl:message terminate="yes">
            in method named &quot;<xsl:value-of select="ancestor::pgml.method/@name"/&quot;,
            &lt;lib&gt; has wrong number of parameters!
            &quot;<xsl:value-of select="@call"/&quot; must have two parameters.
        </xsl:message>
    </xsl:if>

    <xsl:text>String.charAt</xsl:text>
    <xsl:call-template name="tail-part"/>
</xsl:when>

<xsl:when test="@call='pgml.String.compare'">
    <xsl:if test="not(count(*)=2)">
        <xsl:message terminate="yes">
            in method named &quot;<xsl:value-of select="ancestor::pgml.method/@name"/&quot;,
            &lt;lib&gt; has wrong number of parameters!
            &quot;<xsl:value-of select="@call"/&quot; must have two parameters.
        </xsl:message>
    </xsl:if>

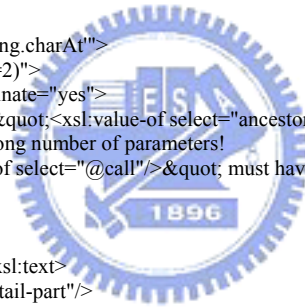
    <xsl:text>String.compare</xsl:text>
    <xsl:call-template name="tail-part"/>
</xsl:when>

<xsl:when test="@call='pgml.String.length'">
    <xsl:if test="not(count(*)=1)">
        <xsl:message terminate="yes">
            in method named &quot;<xsl:value-of select="ancestor::pgml.method/@name"/&quot;,
            &lt;lib&gt; has wrong number of parameters!
            &quot;<xsl:value-of select="@call"/&quot; must have one parameter.
        </xsl:message>
    </xsl:if>

    <xsl:text>String.length</xsl:text>
    <xsl:call-template name="tail-part"/>
</xsl:when>

<xsl:when test="@call='pgml.String.substring'">
    <xsl:if test="not(count(*)=3)">
        <xsl:message terminate="yes">

```



```

                in method named &quot;<xsl:value-of select="ancestor::pgml:method/@name"/>&quot;,
                &lt;lib&gt; has wrong number of parameters!
                &quot;<xsl:value-of select="@call"/>&quot;, must have three parameter.
            </xsl:message>
        </xsl:if>

        <xsl:text>String.subString</xsl:text>
        <xsl:call-template name="tail-part"/>
    </xsl:when>

    <xsl:when test="@call='pgml.String.trim'">
        <xsl:if test="not(count(*)=1)">
            <xsl:message terminate="yes">
                in method named &quot;<xsl:value-of select="ancestor::pgml:method/@name"/>&quot;,
                &lt;lib&gt; has wrong number of parameters!
                &quot;<xsl:value-of select="@call"/>&quot;, must have one parameter.
            </xsl:message>
        </xsl:if>

        <xsl:text>String.trim</xsl:text>
        <xsl:call-template name="tail-part"/>
    </xsl:when>

    <xsl:when test="@call='pgml.String.concat'">
        <xsl:if test="not(count(*)=2)">
            <xsl:message terminate="yes">
                in method named &quot;<xsl:value-of select="ancestor::pgml:method/@name"/>&quot;,
                &lt;lib&gt; has wrong number of parameters!
                &quot;<xsl:value-of select="@call"/>&quot;, must have two parameter.
            </xsl:message>
        </xsl:if>

        <xsl:apply-templates select="*[1]"/> + <xsl:apply-templates select="*[2]"/>
    </xsl:when>

    <xsl:when test="@call='pgml.Boolean.toString'">
        <xsl:if test="not(count(*)=1)">
            <xsl:message terminate="yes">
                in method named &quot;<xsl:value-of select="ancestor::pgml:method/@name"/>&quot;,
                &lt;lib&gt; has wrong number of parameters!
                &quot;<xsl:value-of select="@call"/>&quot;, must have one parameter.
            </xsl:message>
        </xsl:if>

        <xsl:text>String.toString</xsl:text>
        <xsl:call-template name="tail-part"/>
    </xsl:when>

    <xsl:otherwise>
        <xsl:message terminate="yes">
            in method named &quot;<xsl:value-of select="ancestor::pgml:method/@name"/>&quot;,
            &lt;lib&gt; has wrong library function call!
            you use a non-existing PGML standard library : <xsl:value-of select="@call"/>
        </xsl:message>
    </xsl:otherwise>
</xsl:choose>
</xsl:template>

<xsl:template name="tail-part">
    <!--passing needed parameters into method-->
    <xsl:text></xsl:text>

    <xsl:for-each select="*">
        <xsl:apply-templates select="."/>

        <xsl:if test="not(position()=last())">
            <xsl:text>, </xsl:text>
        </xsl:if>
    </xsl:for-each>

    <xsl:text></xsl:text>
</xsl:template>
</xsl:stylesheet>

```

## [ For PUML – PUML2MIDP.xsl (indented)]

```
<?xml version="1.0"?>
```

```
<!--this is an XSLT stylesheet of PUML 1.2 to transform PUML into J2ME MIDP without error control.-->
```

```
<xsl:stylesheet version="1.0" xmlns:xsl="http://www.w3.org/1999/XSL/Transform"
xmlns:puml="http://dcs.w3.cis.nctu.edu.tw/Project/Pervasive/PUML/">
```

```
<!--for package path information-->
```

```
<xsl:param name="packagePath"/>
```

```
<!--because MIDP is a text document-->
```

```
<xsl:output method="text"/>
```

```
<xsl:template match="/">
```

```
<xsl:if test="$packagePath">
```

```
package <xsl:value-of select="$packagePath"/>;
```

```
</xsl:if>
```

```
<xsl:text> </xsl:text>
```

```
<xsl:text>import javax.microedition.midlet.*; </xsl:text>
```

```
<xsl:text>import javax.microedition.lcdui.*; </xsl:text>
```

```
<xsl:text> </xsl:text>
```

```
<xsl:text> </xsl:text>
```

```
interface BMngr_Interface {
    public void changeBoard(String boardName);
    public Displayable getBoard(String boardName);
}
```

```
<xsl:text> </xsl:text>
```

```
<xsl:text> </xsl:text>
```

```
<xsl:apply-templates select="puml:user-interface"/>
```

```
</xsl:template>
```

```
<xsl:template match="puml:user-interface">
```

```
public class BMngr_<xsl:value-of select="@name"/> extends MIDlet implements BMngr_Interface {
```

```
public static BMngr_<xsl:value-of select="@name"/> instance;
```

```
private Display display;
```

```
/*global logic objects*/
```

```
<xsl:apply-templates select="puml:logic-objects"/>
```

```
/*all displayables corresponding to boards' declaration*/
```

```
<xsl:for-each select="puml:board">
```

```
private B_<xsl:value-of select="@name"/><xsl:text> </xsl:text><xsl:text><xsl:value-of select="@name"/> = new B_<xsl:value-of
```

```
select="@name"/>());
```

```
<xsl:text> </xsl:text>
```

```
</xsl:for-each>
```

```
/*Constructor*/
```

```
public BMngr_<xsl:value-of select="@name"/>(){
```

```
instance = this;
```

```
}
```

```
/*Main method*/
```

```
public void startApp(){
```

```
display = Display.getDisplay(this);
```

```
<xsl:call-template name="set-first-UI"/>
```

```
}
```

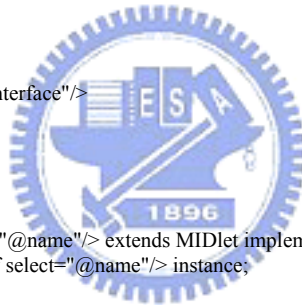
```
/*Handle pausing the MIDlet*/
```

```
public void pauseApp(){
```

```
}
```

```
/*Handle destroying the MIDlet*/
```

```
public void destroyApp(boolean unconditional){
```



```

    }

    /*Quit the MIDlet*/
    public static void quitApp(){
        instance.destroyApp(true);
        instance.notifyDestroyed();
        instance = null;
    }

    /*change UI presentation*/
    public void changeBoard(String boardName){
        <xsl:for-each select="puml:board">
            if(boardName.compareTo("<xsl:value-of select="@name"/>&quot;") == 0){
                display.setCurrent(<xsl:value-of select="@name"/>);
            }
        </xsl:for-each>
    }

    /*get particular UI container*/
    public Displayable getBoard(String boardName){
        Displayable displayable = null;

        <xsl:for-each select="puml:board">
            if(boardName.compareTo("<xsl:value-of select="@name"/>&quot;") == 0){
                displayable = <xsl:value-of select="@name"/>;
            }
        </xsl:for-each>

        return displayable;
    }
}

<xsl:text> </xsl:text>
<xsl:text> </xsl:text>

<xsl:for-each select="puml:board">
    <xsl:apply-templates select="."/>
</xsl:for-each>
</xsl:template>

<xsl:template name="set-first-UI">
    <xsl:choose>
        <xsl:when test="board[@seqNO=0]">
            display.setCurrent(<xsl:value-of select="puml:board[@seqNo=0]/@name"/>);
        </xsl:when>

        <xsl:otherwise>
            display.setCurrent(<xsl:value-of select="puml:board[1]/@name"/>);
        </xsl:otherwise>
    </xsl:choose>
</xsl:template>

<xsl:template match="puml:layout">
    <!--be used for future-->
</xsl:template>

<xsl:template match="puml:logic-objects">
    <!--global logic objects-->
    <xsl:if test="name(.)='user-interface'">
        <xsl:for-each select="puml:object">
            public <xsl:apply-templates select="."/>
        </xsl:for-each>
    </xsl:if>

    <!--local logic objects-->

```



```

<xsl:if test="name(.)='board'">
  <xsl:for-each select="puml:object">
    private <xsl:apply-templates select="."/>
  </xsl:for-each>
</xsl:if>
</xsl:template>

```

```

<xsl:template match="puml:object">
  <xsl:variable name="className" select="substring-before(@source, '.pgml')"/>
  <xsl:value-of select="$className"/> IgObj_ <xsl:value-of select="@name"/> = new <xsl:value-of select="$className"/>();
  <xsl:text>    </xsl:text>
</xsl:template>

```

```

<xsl:template match="puml:board">
  class B <xsl:value-of select="@name"/> extends Form implements CommandListener {
    /*local logic objects*/
    <xsl:apply-templates select="puml:logic-objects"/>

    /*UI components*/
    <xsl:call-template name="UI-components-declaration"/>

    /*commands*/
    <xsl:for-each select="puml:action">
      private Command <xsl:value-of select="@name"/> = new Command("<xsl:value-of select="@showText"/>"&quot;,
Command.SCREEN, 1);
      <xsl:text>    </xsl:text>
    </xsl:for-each>

```

```

    /*Constructor*/
    public B <xsl:value-of select="@name"/>() {
      /* initialize UI component */
      super("<xsl:value-of select="@title"/>"&quot;);
      <xsl:call-template name="UI-components-initialization"/>

      /* set up listeners */
      this.setCommandListener(this);

      /* append UI component */
      <xsl:call-template name="append-UI-components"/>

      /* add commands */
      <xsl:for-each select="puml:action">
        this.addCommand(<xsl:value-of select="@name"/>);
        <xsl:text>      </xsl:text>
      </xsl:for-each>
    }

```

```

    /*Handle command events*/
    public void commandAction(Command command, Displayable displayable){
      String selectedStr = null;
      String oldPath = null;
      String newPath = null;
      String align = null;

      <xsl:for-each select="puml:action">
        <xsl:apply-templates select="."/>
      </xsl:for-each>
    }
  }

```

```

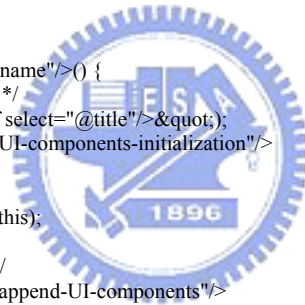
  <xsl:text>    </xsl:text>
  <xsl:text>    </xsl:text>
</xsl:template>

```

```

<xsl:template name="UI-components-declaration">
  <xsl:for-each select="node()">
    <xsl:choose>

```



```

<xsl:when test="name(.)='picture'">
    public ImageItem <xsl:value-of select="@name"/>;
</xsl:when>

<xsl:when test="name(.)='label'">
    public StringItem <xsl:value-of select="@name"/>;
</xsl:when>

<xsl:when test="name(.)='textnote'">
    public TextField <xsl:value-of select="@name"/>;
</xsl:when>

<xsl:when test="name(.)='listpaper'">
    public ChoiceGroup <xsl:value-of select="@iname"/>;
</xsl:when>
</xsl:choose>

<xsl:text> </xsl:text>
</xsl:for-each>
</xsl:template>

```

```

<xsl:template name="UI-components-initialization">
    <xsl:for-each select="node()">
        <xsl:choose>
            <xsl:when test="name(.)='picture'">
                try{
                    <xsl:value-of select="@name"/> = new ImageItem(&quot;&quot;,
                        Image.createImage(&quot;<xsl:value-of select="substring-before(@source, '.')">.png&quot;),
                        <xsl:call-template name="picture-position-param"/>, &quot;<xsl:value-of select="@altText"/>&quot;);
                } catch(Exception e){e.printStackTrace();}
            </xsl:when>

            <xsl:when test="name(.)='label'">
                <xsl:value-of select="@name"/> = new StringItem (&quot;<xsl:value-of select="@showText"/>&quot;,
                &quot;&quot;);
            </xsl:when>

            <xsl:when test="name(.)='textnote'">
                <xsl:if test="@type='text'">
                    <xsl:value-of select="@name"/> = new TextField(&quot;<xsl:value-of select="@title"/>&quot;,
                        &quot;<xsl:value-of select="@value"/>&quot;, 50, TextField.ANY);
                </xsl:if>

                <xsl:if test="@type='password'">
                    <xsl:value-of select="@name"/> = new TextField(&quot;<xsl:value-of select="@title"/>&quot;,
                        &quot;<xsl:value-of select="@value"/>&quot;, 50, TextField.PASSWORD);
                </xsl:if>
            </xsl:when>

            <xsl:when test="name(.)='listpaper'">
                <xsl:choose>
                    <xsl:when test="@mode='single'">
                        <xsl:value-of select="@iname"/> = new ChoiceGroup(&quot;<xsl:value-of
select="@title"/>&quot;, ChoiceGroup.EXCLUSIVE);
                    </xsl:when>

                    <xsl:when test="@mode='multi'">
                        <xsl:value-of select="@iname"/> = new ChoiceGroup(&quot;<xsl:value-of
select="@title"/>&quot;, ChoiceGroup.MULTIPLE);
                    </xsl:when>

                    <xsl:otherwise>
                        <xsl:value-of select="@iname"/> = new ChoiceGroup(&quot;<xsl:value-of
select="@title"/>&quot;, ChoiceGroup.EXCLUSIVE);
                    </xsl:otherwise>
                </xsl:choose>

                <xsl:for-each select="puml:item">
                    <xsl:value-of select="../@iname"/>.append(&quot;<xsl:value-of select="@showText"/>&quot;, null);
                    <xsl:text> </xsl:text>
                </xsl:for-each>

                <xsl:if test="@default">

```

```

        <xsl:choose>
            <xsl:when test="@mode='single'">
                <xsl:value-of select="@iname"/>.setSelectedIndex(<xsl:value-of select="@default"/>-1,
true);
            </xsl:when>

            <xsl:when test="@mode='multi'">
                String defaultSelected = &quot;<xsl:value-of
select="@default"/>&quot;;<xsl:text></xsl:text>
                for(int index=0; index &lt;= defaultSelected.length(); index=index+2){
                    String str = String.valueOf(defaultSelected.charAt(index));
                    <xsl:value-of select="@iname"/>.setSelectedIndex(Integer.parseInt(str)-1, true);
                }
            </xsl:when>

            <xsl:otherwise>
                <xsl:value-of select="@iname"/>.setSelectedIndex(<xsl:value-of select="@default"/>-1,
true);
            </xsl:otherwise>
        </xsl:choose>
    </xsl:if>
</xsl:when>
</xsl:choose>
<xsl:text> </xsl:text>
</xsl:for-each>
</xsl:template>

```

```

<xsl:template name="picture-position-param">
    <xsl:choose>
        <xsl:when test="@align='left'">
            ImageItem.LAYOUT_LEFT
        </xsl:when>

        <xsl:when test="@align='center'">
            ImageItem.LAYOUT_CENTER
        </xsl:when>

        <xsl:when test="@align='right'">
            ImageItem.LAYOUT_RIGHT
        </xsl:when>

        <xsl:otherwise>
            ImageItem.LAYOUT_DEFAULT
        </xsl:otherwise>
    </xsl:choose>
</xsl:template>

```



```

<xsl:template name="append-UI-components">
    <xsl:for-each select="node()">
        <xsl:choose>
            <xsl:when test="name(.)='picture'">
                this.append(<xsl:value-of select="@name"/>);
            </xsl:when>

            <xsl:when test="name(.)='label'">
                this.append(<xsl:value-of select="@name"/>);
            </xsl:when>

            <xsl:when test="name(.)='textnote'">
                this.append(<xsl:value-of select="@name"/>);
            </xsl:when>

            <xsl:when test="name(.)='listpaper'">
                this.append(<xsl:value-of select="@iname"/>);
            </xsl:when>
        </xsl:choose>
        <xsl:text> </xsl:text>
    </xsl:for-each>
</xsl:template>

```

```

<xsl:template match="puml:action">
  if(command == <xsl:value-of select="@name"/>){
    <xsl:for-each select="node()">
      <xsl:apply-templates select="."/>
      <xsl:text> </xsl:text>
    </xsl:for-each>
  }
</xsl:template>

```

```

<xsl:template match="puml:use-object">
  <xsl:if test="name(.)='action'">
    <xsl:call-template name="multi-selection-param-preprocessing"/>
  </xsl:if>

  <xsl:variable name="objectName" select="@name"/>

  <xsl:choose>
    <!--use local logic object-->
    <xsl:when test="ancestor::puml:board/puml:logic-objects/puml:object[@name=$objectName]">
      lgObj_<xsl:value-of select="$objectName"/>.<xsl:value-of select="@method"/>
    </xsl:when>

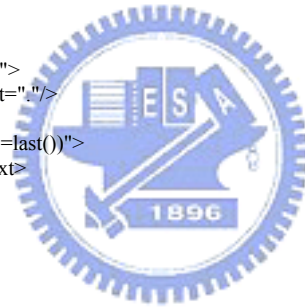
    <!--use global logic object-->
    <xsl:otherwise>
      BMngr_<xsl:value-of select="/puml:user-interface/@name"/>.instance.lgObj_<xsl:value-of
select="$objectName"/>.<xsl:value-of select="@method"/>
    </xsl:otherwise>
  </xsl:choose>

  <xsl:text></xsl:text>
  <xsl:for-each select="puml:param">
    <xsl:apply-templates select="."/>

    <xsl:if test="not(position()=last())">
      <xsl:text>, </xsl:text>
    </xsl:if>
  </xsl:for-each>
  <xsl:text></xsl:text>

  <xsl:if test="name(.)='action'">
    <xsl:text>; </xsl:text>
  </xsl:if>
</xsl:template>

```



```

<xsl:template match="puml:param">
  <xsl:choose>
    <xsl:when test="@select">
      <xsl:call-template name="check-param-UI-component-type"/>
    </xsl:when>

    <xsl:otherwise>
      <xsl:choose>
        <xsl:when test="@type='String'">
          &quot;<xsl:value-of select="@value"/>&quot;
        </xsl:when>

        <xsl:otherwise>
          <xsl:value-of select="@value"/>
        </xsl:otherwise>
      </xsl:choose>
    </xsl:otherwise>
  </xsl:choose>
</xsl:template>

```

```

<xsl:template name="multi-selection-param-preprocessing">
  <xsl:for-each select="./puml:param">
    <xsl:if test="./@select=ancestor::puml:board/puml:listpaper[@mode='multi']/@iname">

```



```

        selectedStr = &quot;&quot;;<xml:text></xml:text>
        for(int i=0; i <&lt; <xml:value-of select="@select"/>.size(); i++){
            if(<xml:value-of select="@select"/>.isSelected(i)){
                if(selectedStr != ""){
                    selectedStr = selectedStr + "," + Integer.toString(i+1);
                }
                else{
                    selectedStr = selectedStr + Integer.toString(i+1);
                }
            }
        }
    }
</xml:if>
</xml:for-each>

<xml:text>    </xml:text>
</xml:template>

```

```

<xml:template name="check-param-UI-component-type">
    <xml:choose>
        <xml:when test="./@select=ancestor::puml:board/puml:label/@name">
            <xml:value-of select="@select"/>.getLabel()
        </xml:when>

        <xml:when test="./@select=ancestor::puml:board/puml:textnote/@name">
            <xml:value-of select="@select"/>.getString()
        </xml:when>

        <xml:when test="./@select=ancestor::puml:board/puml:listpaper[@mode='single']/@iname">
            <xml:value-of select="@select"/>.getSelectedIndex()+1
        </xml:when>

        <xml:when test="./@select=ancestor::puml:board/puml:listpaper[@mode='multi']/@iname">
            selectedStr
        </xml:when>
    </xml:choose>
</xml:template>

```



```

<xml:template match="puml:change">
    <xml:for-each select="puml:use-object">
        <xml:call-template name="multi-selection-param-preprocessing"/>
    </xml:for-each>

    <xml:variable name="ContainerName" select="@container"/>
    <xml:variable name="ComponentName" select="@component"/>
    <xml:variable name="Update" select="@update"/>
    <xml:variable name="ItemIndex" select="@itemIndex"/>
    <xml:variable name="Child" select="node()"/>
    <xml:variable name="IFUseObject">
        <xml:choose>
            <xml:when test="puml:use-object">YES</xml:when>
            <xml:otherwise>NO</xml:otherwise>
        </xml:choose>
    </xml:variable>

    <xml:for-each select="/puml:user-interface/puml:board">
        <xml:if test="@name=$ContainerName">
            <xml:for-each select="node()">
                <xml:if test="@name=$ComponentName">
                    <xml:choose>
                        <xml:when test="name(.)='picture'">
                            <xml:call-template name="process-picture-update"/>
                        </xml:when>

                        <xml:when test="name(.)='label'">
                            <xml:call-template name="process-label-update">
                                <xml:with-param name="container" select="$ContainerName"/>
                                <xml:with-param name="component" select="$ComponentName"/>
                                <xml:with-param name="update" select="$Update"/>
                                <xml:with-param name="child" select="$Child"/>
                            </xml:call-template>
                        </xml:when>
                    </xml:choose>
                </xml:if>
            </xml:for-each>
        </xml:if>
    </xml:for-each>

```

```

        <xsl:with-param name="ifUseObject" select="$IFUseObject"/>
    </xsl:call-template>
</xsl:when>

    <xsl:when test="name(.)='textnote'">
        <xsl:call-template name="process-textnote-update">
            <xsl:with-param name="container" select="$ContainerName"/>
            <xsl:with-param name="component" select="$ComponentName"/>
            <xsl:with-param name="update" select="$Update"/>
            <xsl:with-param name="child" select="$Child"/>
            <xsl:with-param name="ifUseObject" select="$IFUseObject"/>
        </xsl:call-template>
    </xsl:when>

    <xsl:when test="name(.)='listpaper'">
        <xsl:call-template name="process-listpaper-update">
            <xsl:with-param name="container" select="$ContainerName"/>
            <xsl:with-param name="component" select="$ComponentName"/>
            <xsl:with-param name="update" select="$Update"/>
            <xsl:with-param name="itemIndex" select="$ItemIndex"/>
            <xsl:with-param name="child" select="$Child"/>
            <xsl:with-param name="ifUseObject" select="$IFUseObject"/>
        </xsl:call-template>
    </xsl:when>
</xsl:choose>
</xsl:if>
</xsl:for-each>
</xsl:if>
</xsl:for-each>
</xsl:template>

```

```

<xsl:template match="puml.nextboard">
    BMngr_<xsl:value-of select="/puml:user-interface/@name"/>.instance.changeBoard(&quot;<xsl:value-of select="@goto"/>&quot;);
    <xsl:text>    </xsl:text>
</xsl:template>

```



```

<!--
<xsl:template name="process-picture-update">
    <xsl:param name="container"/>
    <xsl:param name="component"/>
    <xsl:param name="update"/>
    <xsl:param name="child"/>
    <xsl:param name="ifUseObject"/>
    <xsl:variable name="BMngrName" select="/puml:user-interface/@name"/>

    <xsl:choose>
        <xsl:when test="$ifUseObject='YES'">
            <xsl:if test="$update='source'">
                try {
                    oldPath=<xsl:apply-templates select="$child"/>;
                    newPath=oldPath.substring(0,oldPath.indexOf('.'));
                    ((B_<xsl:value-of select="$container"/>)BMngr_<xsl:value-of
select="$BMngrName"/>.instance.getBoard(&quot;<xsl:value-of select="$container"/>&quot;)).<xsl:value-of
select="$component"/>.setImage(Image.createImage(newPath.concat(&quot;.png&quot;)));
                } catch(Exception e){e.printStackTrace();}
            </xsl:if>

            <xsl:if test="$update='altText'">
                ((B_<xsl:value-of select="$container"/>)BMngr_<xsl:value-of
select="$BMngrName"/>.instance.getBoard(&quot;<xsl:value-of select="$container"/>&quot;)).<xsl:value-of
select="$component"/>.setAltText(<xsl:apply-templates select="$child"/>);
            </xsl:if>

            <xsl:if test="$update='align'">
                align=<xsl:apply-templates select="$child"/>;
                if(align=="left"){
                    ((B_<xsl:value-of select="$container"/>)BMngr_<xsl:value-of
select="$BMngrName"/>.instance.getBoard(&quot;<xsl:value-of select="$container"/>&quot;)).<xsl:value-of
select="$component"/>.setLayout(ImageItem.LAYOUT_LEFT);
                }
                if(align=="center"){
                    ((B_<xsl:value-of select="$container"/>)BMngr_<xsl:value-of

```

```

select="$BMngrName"/>.instance.getBoard(&quot;<xsl:value-of select="$container"/>&quot;)).<xsl:value-of
select="$component"/>.setLayout(ImageItem.LAYOUT_CENTER);
        }
        if(align=="right"){
            ((B_<xsl:value-of select="$container"/>)BMngr_<xsl:value-of
select="$BMngrName"/>.instance.getBoard(&quot;<xsl:value-of select="$container"/>&quot;)).<xsl:value-of
select="$component"/>.setLayout(ImageItem.LAYOUT_RIGHT);
        }
    </xsl:if>
</xsl:when>

<xsl:otherwise>
    <xsl:if test="$update='source'">
        try{
            ((B_<xsl:value-of select="$container"/>)BMngr_<xsl:value-of
select="$BMngrName"/>.instance.getBoard(&quot;<xsl:value-of select="$container"/>&quot;)).<xsl:value-of
select="$component"/>.setImage(Image.createImage(&quot;<xsl:value-of select="substring-before($child,')"/>.png&quot;));
        }catch(Exception e){e.printStackTrace();}
    </xsl:if>

    <xsl:if test="$update='altText'">
        ((B_<xsl:value-of select="$container"/>)BMngr_<xsl:value-of
select="$BMngrName"/>.instance.getBoard(&quot;<xsl:value-of select="$container"/>&quot;)).<xsl:value-of
select="$component"/>.setAltText(&quot;<xsl:value-of select="$child"/>&quot;);
    </xsl:if>

    <xsl:if test="$update='align'">
        <xsl:if test="string($child)='left'">
            ((B_<xsl:value-of select="$container"/>)BMngr_<xsl:value-of
select="$BMngrName"/>.instance.getBoard(&quot;<xsl:value-of select="$container"/>&quot;)).<xsl:value-of
select="$component"/>.setLayout(ImageItem.LAYOUT_LEFT);
        </xsl:if>

        <xsl:if test="string($child)='center'">
            ((B_<xsl:value-of select="$container"/>)BMngr_<xsl:value-of
select="$BMngrName"/>.instance.getBoard(&quot;<xsl:value-of select="$container"/>&quot;)).<xsl:value-of
select="$component"/>.setLayout(ImageItem.LAYOUT_CENTER);
        </xsl:if>

        <xsl:if test="string($child)='right'">
            ((B_<xsl:value-of select="$container"/>)BMngr_<xsl:value-of
select="$BMngrName"/>.instance.getBoard(&quot;<xsl:value-of select="$container"/>&quot;)).<xsl:value-of
select="$component"/>.setLayout(ImageItem.LAYOUT_RIGHT);
        </xsl:if>
    </xsl:if>
</xsl:otherwise>
</xsl:choose>

    <xsl:text>    </xsl:text>
</xsl:template>
-->

```

```

<xsl:template name="process-label-update">
    <xsl:param name="container"/>
    <xsl:param name="component"/>
    <xsl:param name="update"/>
    <xsl:param name="child"/>
    <xsl:param name="ifUseObject"/>
    <xsl:variable name="BMngrName" select="/puml:user-interface/@name"/>

    <xsl:choose>
        <xsl:when test="$ifUseObject='YES'">
            <xsl:if test="$update='showText'">
                ((B_<xsl:value-of select="$container"/>)BMngr_<xsl:value-of
select="$BMngrName"/>.instance.getBoard(&quot;<xsl:value-of select="$container"/>&quot;)).<xsl:value-of
select="$component"/>.setLabel(<xsl:apply-templates select="$child"/>);
            </xsl:if>
        </xsl:when>

        <xsl:otherwise>
            <xsl:if test="$update='showText'">
                ((B_<xsl:value-of select="$container"/>)BMngr_<xsl:value-of
select="$BMngrName"/>.instance.getBoard(&quot;<xsl:value-of select="$container"/>&quot;)).<xsl:value-of
select="$component"/>.setLabel(&quot;<xsl:value-of select="$child"/>&quot;);
            </xsl:if>
        </xsl:otherwise>
    </xsl:choose>

```

```

        </xsl:if>
      </xsl:otherwise>
    </xsl:choose>

    <xsl:text>    </xsl:text>
  </xsl:template>

```

```

<xsl:template name="process-textnote-update">
  <xsl:param name="container"/>
  <xsl:param name="component"/>
  <xsl:param name="update"/>
  <xsl:param name="child"/>
  <xsl:param name="ifUseObject"/>
  <xsl:variable name="BMngrName" select="/puml:user-interface/@name"/>

  <xsl:choose>
    <xsl:when test="$ifUseObject='YES'">
      <xsl:if test="$update='title'">
        ((B_<xsl:value-of select="$container"/>)BMngr_<xsl:value-of
select="$BMngrName"/>.instance.getBoard(&quot;<xsl:value-of select="$container"/>&quot;)).<xsl:value-of
select="$component"/>.setLabel(<xsl:apply-templates select="$child"/>);
        </xsl:if>

        <xsl:if test="$update='value'">
          ((B_<xsl:value-of select="$container"/>)BMngr_<xsl:value-of
select="$BMngrName"/>.instance.getBoard(&quot;<xsl:value-of select="$container"/>&quot;)).<xsl:value-of
select="$component"/>.setString(<xsl:apply-templates select="$child"/>);
          </xsl:if>
        </xsl:when>

      <xsl:otherwise>
        <xsl:if test="$update='title'">
          ((B_<xsl:value-of select="$container"/>)BMngr_<xsl:value-of
select="$BMngrName"/>.instance.getBoard(&quot;<xsl:value-of select="$container"/>&quot;)).<xsl:value-of
select="$component"/>.setLabel(&quot;<xsl:value-of select="$child"/>&quot;);
          </xsl:if>

          <xsl:if test="$update='value'">
            ((B_<xsl:value-of select="$container"/>)BMngr_<xsl:value-of
select="$BMngrName"/>.instance.getBoard(&quot;<xsl:value-of select="$container"/>&quot;)).<xsl:value-of
select="$component"/>.setString(&quot;<xsl:value-of select="$child"/>&quot;);
            </xsl:if>
          </xsl:otherwise>
        </xsl:choose>

        <xsl:text>    </xsl:text>
      </xsl:template>

```

```

<xsl:template name="process-listpaper-update">
  <xsl:param name="container"/>
  <xsl:param name="component"/>
  <xsl:param name="update"/>
  <xsl:param name="itemIndex"/>
  <xsl:param name="child"/>
  <xsl:param name="ifUseObject"/>
  <xsl:variable name="BMngrName" select="/puml:user-interface/@name"/>

  <xsl:choose>
    <xsl:when test="$ifUseObject='YES'">
      <xsl:if test="$update='title'">
        ((B_<xsl:value-of select="$container"/>)BMngr_<xsl:value-of
select="$BMngrName"/>.instance.getBoard(&quot;<xsl:value-of select="$container"/>&quot;)).<xsl:value-of
select="$component"/>.setLabel(<xsl:apply-templates select="$child"/>);
        </xsl:if>

        <xsl:if test="$update='item'">
          ((B_<xsl:value-of select="$container"/>)BMngr_<xsl:value-of
select="$BMngrName"/>.instance.getBoard(&quot;<xsl:value-of select="$container"/>&quot;)).<xsl:value-of
select="$component"/>.set(<xsl:value-of select="$itemIndex"/>-1, <xsl:apply-templates select="$child"/>, null);
          </xsl:if>
        </xsl:when>

```

```

        <xsl:otherwise>
            <xsl:if test="$update='title'">
                ((B_<xsl:value-of select="$container"/>)BMngr_<xsl:value-of
                select="$BMngrName"/>.instance.getBoard(&quot;<xsl:value-of select="$container"/>&quot;)).<xsl:value-of
                select="$component"/>.setLabel(&quot;<xsl:value-of select="$child"/>&quot;);
            </xsl:if>

            <xsl:if test="$update='item'">
                ((B_<xsl:value-of select="$container"/>)BMngr_<xsl:value-of
                select="$BMngrName"/>.instance.getBoard(&quot;<xsl:value-of select="$container"/>&quot;)).<xsl:value-of
                select="$component"/>.set(<xsl:value-of select="$itemIndex"/>-1, <xsl:value-of select="$child"/>, null);
            </xsl:if>
        </xsl:otherwise>
    </xsl:choose>

    <xsl:text>    </xsl:text>
</xsl:template>

</xsl:stylesheet>

```

## [ For PUML – PUML2WML.xsl (indented) ]

```
<?xml version="1.0"?>
```

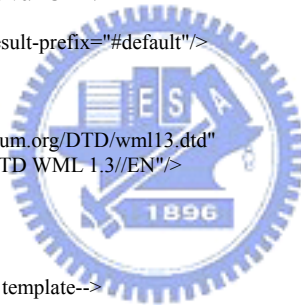
```
<!--this is an XSLT stylesheet of PUML 1.2 to transform PUML into WML with error control.-->
```

```
<xsl:stylesheet version="1.0" xmlns:xsl="http://www.w3.org/1999/XSL/Transform"
xmlns:puml="http://dcs.w3.cis.nctu.edu.tw/Project/Pervasive/PUML/">
```

```
<xsl:namespace-alias stylesheet-prefix="puml" result-prefix="#default"/>
```

```
<!--because WML is an XML document-->
```

```
<xsl:output method="xml" indent="yes"
doctype-system="http://www.wapforum.org/DTD/wml13.dtd"
doctype-public="-//WAPFORUM//DTD WML 1.3/EN"/>
```



```
<xsl:template match="/">
    <xsl:element name="wml">
        <!--no need 'prev' task in common template-->

        <xsl:apply-templates select="puml:user-interface"/>
    </xsl:element>
</xsl:template>
```

```
<xsl:template match="puml:user-interface">
    <xsl:element name="card">
        <xsl:attribute name="id"><xsl:value-of select="@name"/>_index</xsl:attribute>

        <xsl:choose>
            <!--no needing inner cards-->
            <xsl:when test="not(/puml:logic-objects or /puml:board/puml:logic-objects)">
                <xsl:attribute name="ontimer">#preSetting</xsl:attribute>
            </xsl:when>

            <!--needing inner cards-->
            <xsl:otherwise>
                <xsl:call-template name="check-first-inner-card"/>
            </xsl:otherwise>
        </xsl:choose>

        <xsl:element name="timer">
            <xsl:attribute name="value">1</xsl:attribute>
        </xsl:element>
    </xsl:element>

```

```
<!--check if there is any inner card-->
```

```

<xsl:if test="/puml:logic-objects or ./puml:board/puml:logic-objects">
  <xsl:call-template name="chain-of-inner-cards"/>
</xsl:if>

<xsl:call-template name="preSetting-card"/>

<xsl:for-each select="puml:board">
  <xsl:apply-templates select="."/>
</xsl:for-each>
</xsl:template>

<xsl:template name="check-first-inner-card">
  <xsl:variable name="board_inner_cards" select="puml:board/puml:logic-objects"/>

  <xsl:choose>
    <xsl:when test="puml:logic-objects">
      <xsl:attribute name="ontimer">#inner_<xsl:value-of select="/puml:user-interface/@name"/>_<xsl:value-of
select="puml:logic-objects/puml:object[1]/@name"/></xsl:attribute>
    </xsl:when>

    <xsl:otherwise>
      <xsl:attribute name="ontimer">#inner_<xsl:value-of select="$board_inner_cards[1]/../@name"/>_<xsl:value-of
select="$board_inner_cards[1]/puml:object[1]/@name"/></xsl:attribute>
    </xsl:otherwise>
  </xsl:choose>
</xsl:template>

<xsl:template name="chain-of-inner-cards">
  <xsl:variable name="user-interfaceName" select="@name"/>
  <xsl:variable name="board_inner_cards" select="puml:board/puml:logic-objects"/>

  <xsl:if test="puml:logic-objects">
    <xsl:for-each select="puml:logic-objects/puml:object">
      <xsl:element name="card">
        <xsl:attribute name="id">inner_<xsl:value-of select="$user-interfaceName"/>_<xsl:value-of
select="@name"/></xsl:attribute>

        <xsl:attribute name="onenterforward"><xsl:value-of select="substring-before(@source,
'.pgml')"/>.wmls#globalVarDeclaration(</xsl:attribute>

        <xsl:choose>
          <xsl:when test="position()=last()">
            <xsl:choose>
              <xsl:when test="$board_inner_cards">
                <xsl:attribute name="ontimer">#inner_<xsl:value-of
select="$board_inner_cards[1]/../@name"/>_<xsl:value-of select="$board_inner_cards[1]/puml:object[1]/@name"/></xsl:attribute>
              </xsl:when>

              <xsl:otherwise>
                <xsl:attribute name="ontimer">#preSetting</xsl:attribute>
              </xsl:otherwise>
            </xsl:choose>
          </xsl:when>

          <xsl:otherwise>
            <xsl:attribute name="ontimer">#inner_<xsl:value-of
select="$user-interfaceName"/>_<xsl:value-of select="following-sibling::puml:object/@name"/></xsl:attribute>
          </xsl:otherwise>
        </xsl:choose>

        <xsl:element name="timer">
          <xsl:attribute name="value">1</xsl:attribute>
        </xsl:element>
      </xsl:element>
    </xsl:for-each>
  </xsl:if>

  <xsl:if test="$board_inner_cards">
    <xsl:for-each select="$board_inner_cards">

```

```

        <xsl:choose>
            <xsl:when test="position()=last()">
                <xsl:call-template name="local_inner_chain">
                    <xsl:with-param name="current-board" select="."/>
                    <xsl:with-param name="end-inner-chain" select=""YES""/>
                </xsl:call-template>
            </xsl:when>

            <xsl:otherwise>
                <xsl:call-template name="local_inner_chain">
                    <xsl:with-param name="current-board" select="."/>
                    <xsl:with-param name="next-board"
select="$board_inner_cards[count(preceding-sibling::*)+2]"/>
                    <xsl:with-param name="end-inner-chain" select=""NO""/>
                </xsl:call-template>
            </xsl:otherwise>
        </xsl:choose>
    </xsl:for-each>
</xsl:if>
</xsl:template>

<xsl:template name="local_inner_chain">
    <xsl:param name="current-board"/>
    <xsl:param name="next-board"/>
    <xsl:param name="end-inner-chain" select=""NO""/>

    <xsl:if test="$end-inner-chain=""NO"">
        <xsl:for-each select="$current-board/puml:object">
            <xsl:element name="card">
                <xsl:attribute name="id">inner_<xsl:value-of select="../../@name"/>_<xsl:value-of
select="@name"/></xsl:attribute>

                <xsl:attribute name="onenterforward"><xsl:value-of select="substring-before(@source,
'pgml')"/>.wmls#globalVarDeclaration(</xsl:attribute>

                <xsl:choose>
                    <xsl:when test="position()=last()">
                        <xsl:attribute name="ontimer">#inner_<xsl:value-of
select="$next-board/../../@name"/>_<xsl:value-of select="$next-board/puml:object[1]/@name"/></xsl:attribute>
                    </xsl:when>

                    <xsl:otherwise>
                        <xsl:attribute name="ontimer">#inner_<xsl:value-of select="../../@name"/>_<xsl:value-of
select="following-sibling::puml:object/@name"/></xsl:attribute>
                    </xsl:otherwise>
                </xsl:choose>

                <xsl:element name="timer">
                    <xsl:attribute name="value">1</xsl:attribute>
                </xsl:element>
            </xsl:element>
        </xsl:for-each>
    </xsl:if>

    <xsl:if test="$end-inner-chain=""YES"">
        <xsl:for-each select="$current-board/puml:object">
            <xsl:element name="card">
                <xsl:attribute name="id">inner_<xsl:value-of select="../../@name"/>_<xsl:value-of
select="@name"/></xsl:attribute>

                <xsl:attribute name="onenterforward"><xsl:value-of select="substring-before(@source,
'pgml')"/>.wmls#globalVarDeclaration(</xsl:attribute>

                <xsl:choose>
                    <xsl:when test="position()=last()">
                        <xsl:attribute name="ontimer">#preSetting</xsl:attribute>
                    </xsl:when>

                    <xsl:otherwise>
                        <xsl:attribute name="ontimer">#inner_<xsl:value-of select="../../@name"/>_<xsl:value-of
select="following-sibling::puml:object/@name"/></xsl:attribute>
                    </xsl:otherwise>
                </xsl:choose>
            </xsl:element>
        </xsl:for-each>
    </xsl:if>

```





```

        <xsl:attribute name="name"><xsl:value-of
select="ancestor::puml:board/@name"/>_<xsl:value-of select="@iname"/>_title</xsl:attribute>

        <xsl:attribute name="value"><xsl:value-of select="@title"/></xsl:attribute>
    </xsl:element>
</xsl:if>

    <xsl:for-each select="puml:item">
        <xsl:element name="setvar">
            <xsl:attribute name="name"><xsl:value-of
select="ancestor::puml:board/@name"/>_<xsl:value-of select="../@iname"/>_item<xsl:value-of select="position()"/></xsl:attribute>

            <xsl:attribute name="value"><xsl:value-of select="@showText"/></xsl:attribute>
        </xsl:element>
    </xsl:for-each>
</xsl:when>
</xsl:choose>
</xsl:for-each>
</xsl:for-each>
</xsl:template>

```

```

<xsl:template match="puml:layout">
    <!--be used for future-->
</xsl:template>

```

```

<xsl:template match="puml:logic-objects">
    <!--no use in here-->
</xsl:template>

```

```

<xsl:template match="puml:object">
    <!--no use in here-->
</xsl:template>

```



```

<xsl:template match="puml:board">
    <xsl:element name="card">
        <xsl:attribute name="id"><xsl:value-of select="@name"/></xsl:attribute>
        <xsl:if test="@title">
            <xsl:attribute name="title"><xsl:value-of select="@title"/></xsl:attribute>
        </xsl:if>

```

```

        <xsl:for-each select="node()">
            <xsl:choose>
                <xsl:when test="name(.)='picture'">
                    <xsl:element name="p">
                        <xsl:element name="img">
                            <xsl:attribute name="src"><xsl:value-of select="substring-before(@source,
'.')"/>.wbmp</xsl:attribute>

                            <xsl:attribute name="alt"><xsl:value-of select="@altText"/></xsl:attribute>

                            <xsl:if test="@align">
                                <xsl:call-template name="picture-position-param"/>
                            </xsl:if>
                        </xsl:element>
                    </xsl:element>
                </xsl:when>

                <xsl:when test="name(.)='label'">
                    <xsl:element name="p">
                        <xsl:element name="b">${<xsl:value-of select="ancestor::puml:board/@name"/>_<xsl:value-of
select="@name"/>_showText}</xsl:element>
                    </xsl:element>
                </xsl:when>

                <xsl:when test="name(.)='textnote'">
                    <xsl:element name="p">
                        <xsl:if test="@title">

```

```

        <xsl:value-of select="@title"/>
    </xsl:if>

    <xsl:element name="input">
        <xsl:attribute name="name"><xsl:value-of select="ancestor::puml:board/@name"/>_<xsl:value-of
select="@name"/></xsl:attribute>

        <xsl:attribute name="type"><xsl:value-of select="@type"/></xsl:attribute>

        <xsl:if test="@value">
            <xsl:attribute name="value"><xsl:value-of select="@value"/></xsl:attribute>
        </xsl:if>
    </xsl:element>
</xsl:element>
</xsl:when>

<xsl:when test="name(.)='listpaper'">
    <xsl:element name="p">
        <xsl:if test="@title">
            <xsl:value-of select="@title"/>
        </xsl:if>

        <xsl:element name="select">
            <xsl:attribute name="iname"><xsl:value-of
select="ancestor::puml:board/@name"/>_<xsl:value-of select="@iname"/></xsl:attribute>

            <xsl:if test="@default">
                <xsl:attribute name="ivalue"><xsl:value-of select="@default"/></xsl:attribute>
            </xsl:if>

            <xsl:choose>
                <xsl:when test="@mode='single'">
                    <xsl:attribute name="multiple">false</xsl:attribute>
                </xsl:when>
                <xsl:when test="@mode='multi'">
                    <xsl:attribute name="multiple">>true</xsl:attribute>
                </xsl:when>
                <xsl:otherwise>
                    <xsl:attribute name="multiple">false</xsl:attribute>
                </xsl:otherwise>
            </xsl:choose>

            <xsl:for-each select="puml:item">
                <xsl:element name="option">${<xsl:value-of
select="ancestor::puml:board/@name"/>_<xsl:value-of select="."/>@iname"/>_item<xsl:value-of select="position()"/></xsl:element>
            </xsl:for-each>
        </xsl:element>
    </xsl:element>
</xsl:when>
</xsl:choose>
</xsl:for-each>

    <xsl:for-each select="puml:action">
        <xsl:apply-templates select="."/>
    </xsl:for-each>
</xsl:element>

<xsl:for-each select="puml:action">
    <xsl:call-template name="chain-of-temp-cards"/>
</xsl:for-each>
</xsl:template>

<xsl:template name="picture-position-param">
    <xsl:choose>
        <xsl:when test="@align='left'">
            <xsl:attribute name="align">top</xsl:attribute>
        </xsl:when>

        <xsl:when test="@align='center'">
            <xsl:attribute name="align">middle</xsl:attribute>
        </xsl:when>
    </xsl:choose>

```

```

        <xsl:when test="@align='right'">
            <xsl:attribute name="align">bottom</xsl:attribute>
        </xsl:when>

        <xsl:otherwise>
            <xsl:attribute name="align">bottom</xsl:attribute>
        </xsl:otherwise>
    </xsl:choose>
</xsl:template>

<xsl:template match="puml:action">
    <xsl:element name="do">
        <xsl:attribute name="type"><xsl:value-of select="@name"/></xsl:attribute>

        <xsl:attribute name="label"><xsl:value-of select="@showText"/></xsl:attribute>

        <xsl:element name="go">
            <xsl:attribute name="href">#<xsl:value-of select="ancestor::puml:board/@name"/>_<xsl:value-of
select="@name"/>_temp1</xsl:attribute>
        </xsl:element>
    </xsl:element>
</xsl:template>

<xsl:template name="chain-of-temp-cards">
    <xsl:for-each select="node()">
        <xsl:apply-templates select="."/>
    </xsl:for-each>
</xsl:template>

<xsl:template match="puml:use-object">
    <xsl:variable name="objectName" select="@name"/>

    <xsl:variable name="scriptName">
        <xsl:choose>
            <!--use local logic object-->
            <xsl:when test="ancestor::puml:board/puml:logic-objects/puml:object[@name=$objectName]">
                <xsl:value-of select="ancestor::puml:board/puml:logic-objects/puml:object[@name=$objectName]/@source"/>
            </xsl:when>

            <!--use global logic object-->
            <xsl:otherwise>
                <xsl:value-of select="/puml:user-interface/puml:logic-objects/puml:object[@name=$objectName]/@source"/>
            </xsl:otherwise>
        </xsl:choose>
    </xsl:variable>

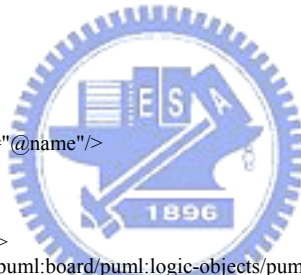
    <xsl:element name="card">
        <xsl:attribute name="id"><xsl:value-of select="ancestor::puml:board/@name"/>_<xsl:value-of
select="../@name"/>_temp<xsl:value-of select="count(preceding-sibling::*)+1"/></xsl:attribute>

        <xsl:attribute name="onenterforward">
            <xsl:value-of select="substring-before($scriptName, '.pgml')"/>.wmls#<xsl:value-of select="@method"/>
        </xsl:attribute>
        <xsl:text></xsl:text>
        <xsl:for-each select="puml:param">
            <xsl:apply-templates select="."/>

            <xsl:if test="not(position()=last())">
                <xsl:text>,</xsl:text>
            </xsl:if>
        </xsl:for-each>
        <xsl:text></xsl:text>
    </xsl:element>

    <xsl:choose>
        <xsl:when test="not(following-sibling::*)">
            <xsl:attribute name="ontimer">#<xsl:value-of select="ancestor::puml:board/@name"/></xsl:attribute>
        </xsl:when>
    </xsl:choose>

```



```

        <xsl:otherwise>
            <xsl:attribute name="ontimer">#<xsl:value-of select="ancestor::puml:board/@name"/>_<xsl:value-of
select="../@name"/>_temp<xsl:value-of select="count(preceding-sibling::*)+2"/></xsl:attribute>
        </xsl:otherwise>
    </xsl:choose>

    <xsl:element name="timer">
        <xsl:attribute name="value">1</xsl:attribute>
    </xsl:element>
</xsl:element>
</xsl:template>

```

```

<xsl:template match="puml:param">
    <xsl:choose>
        <xsl:when test="@select">
            <xsl:call-template name="check-param-UI-component-type"/>
        </xsl:when>

        <xsl:otherwise>
            <xsl:choose>
                <xsl:when test="@type='String'">
                    <xsl:text>&apos;</xsl:text>
                    <xsl:value-of select="@value"/>
                    <xsl:text>&apos;</xsl:text>
                </xsl:when>

                <xsl:otherwise>
                    <xsl:text>&apos;</xsl:text>
                    <xsl:value-of select="@value"/>
                    <xsl:text>&apos;</xsl:text>
                </xsl:otherwise>
            </xsl:choose>
        </xsl:otherwise>
    </xsl:choose>
</xsl:template>

```



```

<xsl:template name="check-param-UI-component-type">
    <xsl:choose>
        <xsl:when test="..@select=ancestor::puml:board/puml:label/@name">&apos;$(<xsl:value-of
select="ancestor::puml:board/@name"/>_<xsl:value-of select="@select"/>_showText)&apos;</xsl:when>

        <xsl:when test="..@select=ancestor::puml:board/puml:textnote/@name">&apos;$(<xsl:value-of
select="ancestor::puml:board/@name"/>_<xsl:value-of select="@select"/>)&apos;</xsl:when>

        <xsl:when test="..@select=ancestor::puml:board/puml:listpaper[@mode='single']/@iname">&apos;$(<xsl:value-of
select="ancestor::puml:board/@name"/>_<xsl:value-of select="@select"/>)&apos;</xsl:when>

        <xsl:when test="..@select=ancestor::puml:board/puml:listpaper[@mode='multi']/@iname">&apos;$(<xsl:value-of
select="ancestor::puml:board/@name"/>_<xsl:value-of select="@select"/>)&apos;</xsl:when>
    </xsl:choose>
</xsl:template>

```

```

<xsl:template match="puml:change">
    <xsl:variable name="objectName" select="puml:use-object/@name"/>

    <xsl:variable name="scriptName">
        <xsl:choose>
            <xsl:when test="puml:use-object">
                <xsl:choose>
                    <!--use local logic object-->
                    <xsl:when test="ancestor::puml:board/puml:logic-objects/puml:object[@name=$objectName]">
                        <xsl:value-of
select="ancestor::puml:board/puml:logic-objects/puml:object[@name=$objectName]/@source"/>
                    </xsl:when>

                    <!--use global logic object-->
                    <xsl:otherwise>

```

```

        <xsl:value-of
select="/puml:user-interface/puml:logic-objects/puml:object[@name=$ObjectName]/@source"/>
        </xsl:otherwise>
    </xsl:choose>
</xsl:when>

    <xsl:otherwise>
        <xsl:choose>
            <xsl:when test="/puml:user-intefcae/puml:logic-objects">
                <xsl:value-of select="/puml:user-intefcae/puml:logic-objects/puml:object[1]/@source"/>
            </xsl:when>

            <xsl:when test="/puml:user-intefcae/puml:board/puml:logic-objects">
                <xsl:value-of
select="/puml:user-intefcae/puml:board/puml:logic-objects/puml:object[1]/@source"/>
            </xsl:when>

            <xsl:otherwise>
                <xsl:text>PGMLchange.pgml</xsl:text>
            </xsl:otherwise>
        </xsl:choose>
    </xsl:otherwise>
</xsl:choose>
</xsl:variable>

    <xsl:element name="card">
        <xsl:attribute name="id"><xsl:value-of select="ancestor::puml:board/@name"/>_<xsl:value-of
select="../@name"/>_temp<xsl:value-of select="count(preceding-sibling::*)+1"/></xsl:attribute>

        <xsl:attribute name="onenterforward">
            <xsl:choose>
                <xsl:when test="puml:use-object"><xsl:value-of select="substring-before($scriptName,
'.pgml')"/>.wmls#change(&apos;<xsl:call-template name="chang-UI-component-value"/>&apos;, &apos;&apos;, &apos;&apos;<xsl:value-of
select="puml:use-object/@method"/>&apos;)</xsl:when>

                <xsl:otherwise><xsl:value-of select="substring-before($scriptName,
'.pgml')"/>.wmls#change(&apos;<xsl:call-template name="chang-UI-component-value"/>&apos;, &apos;&apos;<xsl:value-of select="text()"/>&apos;,
&apos;&apos;)</xsl:otherwise>
            </xsl:choose>
        </xsl:attribute>

        <xsl:choose>
            <xsl:when test="not(following-sibling::*)">
                <xsl:attribute name="ontimer">#<xsl:value-of select="ancestor::puml:board/@name"/></xsl:attribute>
            </xsl:when>

            <xsl:otherwise>
                <xsl:attribute name="ontimer">#<xsl:value-of select="ancestor::puml:board/@name"/>_<xsl:value-of
select="../@name"/>_temp<xsl:value-of select="count(preceding-sibling::*)+2"/></xsl:attribute>
            </xsl:otherwise>
        </xsl:choose>

        <xsl:element name="timer">
            <xsl:attribute name="value">1</xsl:attribute>
        </xsl:element>
    </xsl:element>
</xsl:template>

<xsl:template name="chang-UI-component-value">
    <xsl:variable name="ContainerName" select="@container"/>
    <xsl:variable name="ComponentName" select="@component"/>
    <xsl:variable name="Update" select="@update"/>
    <xsl:variable name="ItemIndex" select="@itemIndex"/>

    <xsl:for-each select="/puml:user-interface/puml:board">
        <xsl:if test="@name=$ContainerName">
            <xsl:for-each select="node()">
                <xsl:if test="@name=$ComponentName">
                    <xsl:choose>
                        <xsl:when test="name(.)='picture'">

```

```

        <!--reserved for future-->
    </xsl:when>

    <xsl:when test="name(.)='label'">
        <xsl:if test="$Update='showText'"><xsl:value-of
select="$ContainerName"/>_<xsl:value-of select="$ComponentName"/>_showText</xsl:if>
        </xsl:when>

        <xsl:when test="name(.)='textnote'">
            <xsl:if test="$Update='title'"><xsl:value-of select="$ContainerName"/>_<xsl:value-of
select="$ComponentName"/>_title</xsl:if>

            <xsl:if test="$Update='value'"><xsl:value-of select="$ContainerName"/>_<xsl:value-of
select="$ComponentName"/></xsl:if>

            </xsl:when>

        <xsl:when test="name(.)='listpaper'">
            <xsl:if test="$Update='title'"><xsl:value-of select="$ContainerName"/>_<xsl:value-of
select="$ComponentName"/>_title</xsl:if>

            <xsl:if test="$Update='item'"><xsl:value-of select="$ContainerName"/>_<xsl:value-of
select="$ComponentName"/>_item<xsl:value-of select="$ItemIndex"/></xsl:if>
            </xsl:when>
        </xsl:choose>
    </xsl:if>
</xsl:for-each>
</xsl:if>
</xsl:for-each>
</xsl:template>

```

```

<xsl:template match="puml:nextboard">
    <xsl:element name="card">
        <xsl:attribute name="id"><xsl:value-of select="ancestor:puml:board/@name"/>_<xsl:value-of
select="./@name"/>_temp<xsl:value-of select="count(preceding-sibling::*)+1"/></xsl:attribute>
        <xsl:attribute name="onenterforward">#<xsl:value-of select="@goto"/></xsl:attribute>
    </xsl:element>
</xsl:template>

```

```

<!--other templates-->

```

```

</xsl:stylesheet>

```

