支援視覺化編輯工具所產生之互動式多媒體內容在 PDA 環境下的 MPEG-4 播放器製作

學生：葉京荃　　指導教授：陳登吉 博士

國立交通大學資訊工程學系碩士班

摘要

MPEG-4 是一個由 MPEG 組織所發表用於視訊壓縮與呈現的多媒體標準。它採用了物件導向的觀念，主要擁有的優點：低位元率、高度壓縮率並且具有互動能力。有鑑於智慧型手機、PDA 的普及，在手持行動裝置上的互動式多媒體應用越來越多，如賀卡、簡訊、學習類教材或是試題。如何快速有效的編輯出 MPEG-4 規格的多媒體呈現，以及支援其在手持系統上的播放為一重要挑戰。本研究着重於設計與實作一在 PDA 上可播放互動式的多媒體呈現的 MPEG-4 播放器。

在這篇研究裡面，我和另一位同學共同合作研究。另一位同學負責研究 MPEG-4 的轉繹器，將視覺化編輯工具產生 MPEG-4 檔案，最後在我所設計研究的 MPEG-4 播放器上播放。MPEG-4 的播放器使用了開放源碼來完成設計。將 PC 上的程式碼改編到 PDA 上是有價值的，了解 MPEG-4 的核心程式是一件不易的工作。數個多媒體應用的例子用來展現 MPEG-4 播放器的可行性與應用性。

The Design and Implementation of an MPEG-4 player under the PDA environment for the multimedia presentation

Student : *Ching-Chuan Yeh* Advisor: *Dr. Deng-Jyi Chen*

Department of Computer Science and Information Engineering
National Chiao Tung University

Abstract

MPEG-4 is a current standard for video encoding and presentation announced by MPEG organization. It adopts the object-objected concept. Low bit-rate and high file compression ratio are its major advantage with supporting of object interaction. Smart phones and PDA have become very popular pervasive devices today due to many multimedia applications on them become feasible. These multimedia applications include gamming, e-mailing, messaging, multimedia presentation… to name a few. How to take the advantages of MPEG-4 to realize it on PC and PDA environment becomes a challenge work.

In this thesis research, we teamed up, with the other classmate who is responsible for the implementation of an MPEG-4 file translator based on a visual authoring tool, to implement the MPEG-4 player for PDA and PC devices. Specifically, both the MPEG-4 player for PC and PDA are designed and implemented using open source library code. Porting experiences from this project is valuable and the understanding of the MPEG-4 kernel is non-trivia task. Several multimedia application examples are used to demonstrate the feasibility and applicability of the MPEG-4 player.

誌謝

感謝神，在我研究所兩年的日子裡，我曾經碰到令我幾乎難以承受的打擊，父親所生的病幾乎摧毀了我們家，我感謝祂在我絕望的時候，祂安慰了我；在生病的時候，我得到醫治；在我論文找不到方向的時候，祂給了我方向。

各位同學、學弟學妹，若我有這個榮幸讓你們看到我這篇論文的話，請你們聽聽我下面的描述。我父親有精神方面的疾病，在研究所的歷程當中，從碩一剛開始的時候他就曾經生過一次重病，到碩二快結束時又生了另一次，這一次重病是最重的一次，他病到要自殺。當然在這個景況下，我們全家都非常緊張，深怕一個沒注意到他就自殺身亡了，在這個過程當中，母親、妹妹和我必須每天輪流守候，還有教會的弟兄姊妹用愛心的付出，才挽回了父親的性命。

在精神病的病房中，看到許許多多的病人，有很多人都待了很久的時間，我父親卻在一個月之內就好了起來，我實在不得不說是神蹟，這件事不就是你們曾經聽過的”死裡復活的神蹟”嗎? 有一位神祂創造天地萬有，並且想和人建立親密的關係，但是人因為犯罪的關係，就與神隔絕，要走向滅亡的地獄之路。只要你曾經做過虧欠良心的事就是犯罪。但是神愛這個世界上的所有人，就差派祂的愛子耶穌為我們的罪釘死在十字架上，並且在三天後復活了。每個人都需要這位救主，我們要向祂承認自己的罪，並且相信祂，讓祂來引導你一生，祂必賜給你全新的生命且是豐盛的生命。

若你決定要來認識這一位上帝，請你開始讀聖經、禱告並且去你家附近的教會聚會，祂一定會引導你的道路，因為是祂找到了你，他要愛你，把最好的東西都賜給你。

本論文多蒙指導老師 陳登吉教授的耐心指導及多方教誨，幫助我度過許多難關，論文得以順利完成，在此致上無限的謝意。

此外，感謝所有曾教導我、幫助我的師長、朋友、同學及軟體工程實驗室中每一位成員，尤其是實驗室同窗：周宜靖、王宇涵、江書瑩、吳直穎，實驗室學弟妹：韓整賢、余筱薇、洪啟彰，在各方面所給予的寶貴意見與協助。

最後，感謝不斷支持我的家人，特別是養育我、栽培我的父母，讓我全心全意地完成學業與論文，也才能有今天的我，在此獻上我最衷心的感謝。

目錄

表目錄

圖目錄

一、緒論

由於計算能力、記憶體、和網路頻寬的增加，使用者對於多媒體的應用愈來愈多。直到最近，因著各式各樣多媒體科技的發明，多媒體應用在過去幾年來愈趨於複雜。最近的多媒體應用期望把各樣不同的多媒體物件做一整合，並且達到和使用者互動的效果。這些多媒體物件不再是靜態的，可能會隨著與使用的互動或者外在環境做改變。因此，這些多媒的內容必須具有互動式的邏輯，而這項能力必須由描述語言(the script language)來完成。

因些，MPEG committee 試圖將這些多媒體技術整合成為單一標準，這就是我們所熟知的 MPEG-4 標準。MPEG-4 是 MPEG-1 和 MPEG-2 這兩個視訊技術的推進，在 1998 年被 ISO 標準組織設計出來。MPEG-4 相較於前兩個技術，從原來的 frame-based 的技術，接受了新的物件導向的觀念。

另外，隨著 PC 的運算能力提升，手持裝置的運算能力也跟著提升，因此手持裝置上的互動式多媒體也有正在發展起來的趨勢，我們系統的發展環境會針對此一方向。

在這一章裡面，我們會介紹 MPEG 技術的演進，目前手持裝置上互動式多媒體的發展，最後我們會詳述這篇論文的研究動機與目標。

1.1 MPEG Committee

MPEG (Moving Picture Experts Group) 組織為一世界知名的制定多媒體技術標準的組織，這些多媒體技術包括視訊、音訊的壓縮技術、多媒體應用與通訊。MPEG 是在 ISO(International Organization for Standardization)的支持下運作，一年開會 3 次，參加的人平均約有 350 個，這些專家來自 200 多個公司、20 多個不同的國家。

MPEG 家族制定了許多世界知名的標準，包括我們所熟知的 MPEG-1, MPEG-2 和 MPEG-4，這些標準被規定在 ISO/IEC-11172, ISO/IEC-13818 和 ISO/IEC-14496 這些文件之中。

1.1.1 MPEG-1

MPEG-1 是 MPEG 所研發出來的第一個標準，在 1992 年被發展出來。它的目標是想要達到在低位元率(low bit rates)下，對相當品質的影像和聲音多媒體內容做壓縮。MPEG-1 標準的要求對視訊到達 1.2Mbps 而音訊到達 250kbps。MPEG-1 定義了 3 層，這 3 層針對不同的音訊品質和壓縮複雜度而設計。一般人

所熟知的 MP3 格式，正是 MPEG-1 Layer III。這個格式可以對 5 分鐘的立體音訊效果做壓縮，在人耳可以接受的品質下，只有 5Mbytes 的大小。

1.1.2 MPEG-2

1994 年 MPEG-2 標準被設計出來，為的是擁有更高的位元率(4-9 Mbits/sec)。MPEG-2 屬於較先進的影像壓縮技術，具有十分優異的壓縮效能，相對的硬體需求也比較高；目前 MPEG-2 編碼技術並未全面開放使用，任何 MPEG-2 轉換都必須付費使用。由於 MPEG-2 影片具有優異的壓縮效能，因此被高畫質的 DVD 所採用。在今日，MPEG-2 在數位電視、DVDs、PCs、衛星、和 set-up boxes 這些應用大量被使用。

1.1.3 MPEG-3

MPEG-3 當初設計的目的是為了涵蓋 HDTV 這個領域，提供更好的位元率(20-40M bit/sec)。但是沒過多久，他們就發現 MPEG-2 本身就可以達到 HDTV 的需要，所以 MPEG-3 這個標準就被廢止了。

1.1.4 MPEG-4

MPEG-4 在 1998 年被制定完成，在 1999 年通過 ISO 認證。MPEG-1 和 MPEG-2 強調於 framed-based 的音訊視訊多媒體內容，而 MPEG-4 是將這些各種不同的多媒體型態做一個整合，並且提供非常多的多媒體應用也讓使用者可對這些多媒體內容做互動。MPEG-4 使用的不是舊有的 frame-based 技術，而是新的物件導向的概念，在一個架構下整合了現有的多媒體技術包括 2D/3D 圖形、動畫、視訊 codec，多媒體串流，互動能力等。

將這些多媒體技術整合之後，MPEG-4 就能提供這些應用: 數位電視、互動式圖形應用和互動式多媒體(WWW)。MPEG-4 是一個標準格式和機制來儲存和傳送這些多媒體，並且將這些多媒體做一個整合並且再使用那些多媒體素材。

1.2 MPEG-4 概念

為了了解 MPEG-4，我們將 MPEG-2 和 MPEG-4 處理流程做一個比較。然後

我們會對 MPEG-4 的特性和功能作一介紹。[10]

MPEG-2 所使用的是一種靜態(static)展示的方法，所呈現的畫面都已經是固定的，無法與使用者有任何的互動。MPEG-2 的數位內容，在壓縮之前就先把這些原始資料組成一個一個的框架(frame)，這些原始資料是由視訊、圖形、文字、影像所組成的。所以當使用者播放這 MPEG-2 格式的多媒體時，只能把每一個框架(frame)依序解出，卻沒有辦法再一次的重組這些原始資料，因為這在壓縮那一端就已經組合(compose)完成了。

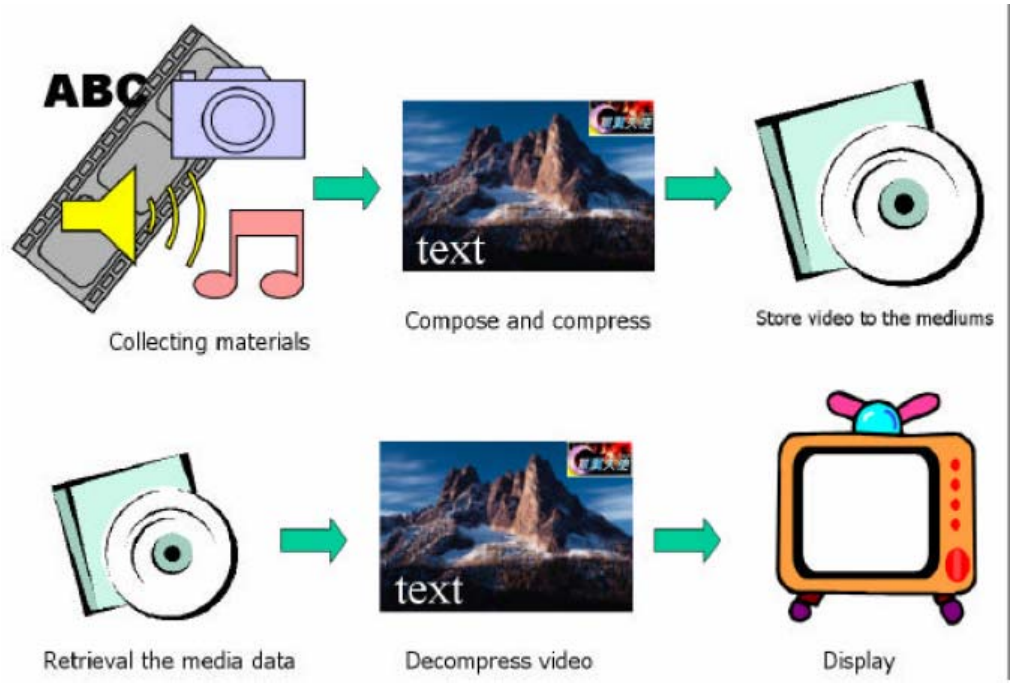


圖 1：MPEG-2 對多媒體資料的處理方式

在 MPEG-4 的標準裡面，所有的多媒體資料是分別被壓縮及傳送，在展示之前，這些資料串流(stream)才被重新組合成一個畫面。所以，MPEG-2 是在壓縮之前就先組合每一個框架(frame)，而 MPEG-4 是在解壓縮每一個資料串流之後，才將之組合成每一個場景(scene)。這樣的設計有下面兩點好處:

- 由於 MPEG-4 是在展示之前才把畫面展示出來，這樣就可以達到與使用者互動功能，當使用者輸入一些資訊到場景裡面，場景馬上可以被重新組合成新的畫面。
- 若是在一個場景之中，有重覆使用到的資料串流，比如說一張圖被重覆使用了好幾次，在 MPEG-4 的設計之中，因為每一次的展示都要重新組合畫面，所以若是在場景中有重覆出現的資料，只要使用同一分的資料串流就可以了，這樣就可以節省大量的壓縮空間。

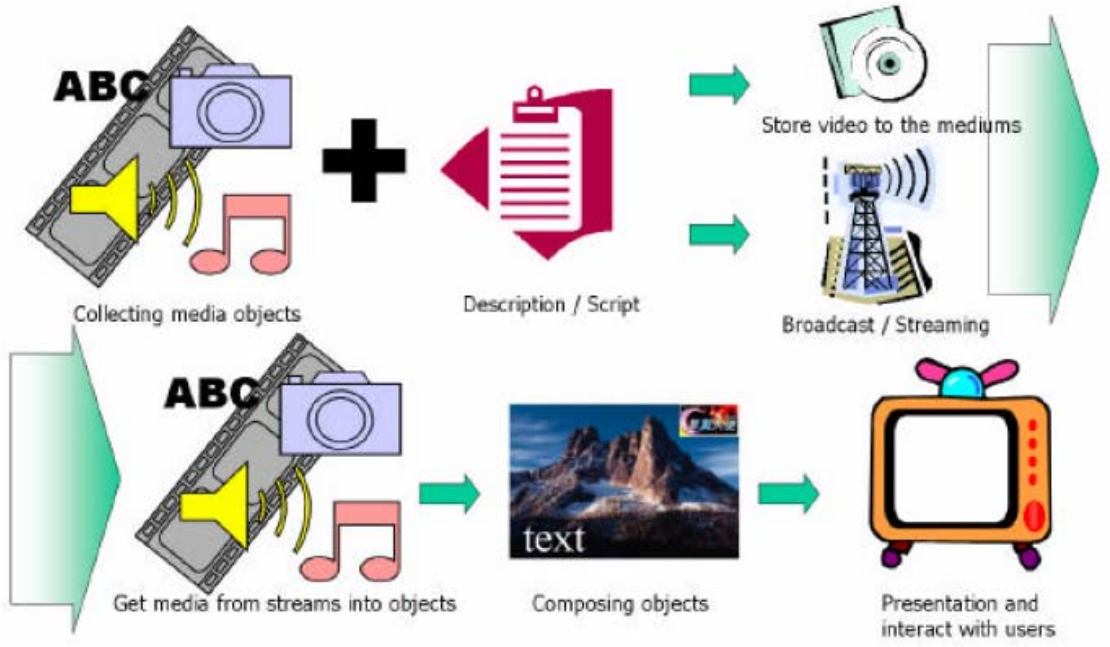


圖 2：MPEG-4 多媒體資料的處理方式

1.3 互動式多媒體在 PC 及 PDA 上之發展

一般我們所認知的多媒體為: 圖片、聲音、影片、3D Model、文字等，或者稱為具有聲光效果的呈現。多媒體，具體而言，是結合了許多種類型的媒體元素，包括影像、聲音、及文字，而最重要的是藉由電腦來掌控呈現，因此更精確地說，應稱之為"電腦多媒體"。當這些多媒體互相整合起來，並且具有和使用者互動的功能時，我們稱為互動式多媒體。

互動式多媒體的定義如下：

- 可以呈現文字、圖形、照片、動畫、影片、音效等不同型式的資料。
- 所有訊息皆轉成數位化的型態。
- 以電腦來掌控其播放呈現的方式和內容的安排。
- 具有與使用者互動的功能。

PC 上的互動式多媒體擁有比以前更多采多姿的各種應用，如電子賀卡、學習類教材、遊戲、互動式虛擬實境、互動式網頁等。目前市面在 PC 上，較為大眾所知的互動式多媒體格式有下列幾種：

- flash：flash 是 Macromedia 公司所推出的軟體，它是專門用來設計互動式多媒體動畫的軟體，它可以為你的網頁加入專業且漂亮的互動式按鈕及向量式的動畫圖案特效，它是目前製作網頁動畫最熱門的軟體。
- sparkle：微軟將會在不久後推出新一代動畫制作工具 Sparkle，其功能與架構似乎都學習自 flash。Sparkle 將會繼承在微軟的下一代作業系統 Longhorn 產品及

相關技術中。

- MPEG-4：MPEG-4 為一由 MPEG 所制定的新的多媒體格式，不但對 video、audio 本身有較 MPEG-1, MPEG-2 有更好的壓縮率，更重要的它具有前兩個格式所沒有的互動功能，可以讓使用者對一個場景中的多媒體物件做互動。

目前互動式多媒體日漸普及，在 PC 上看得到的應用已有許多，但手持裝置上的應用還不是非常普遍，最常見的為遊戲類，而手機上的 MMS 簡訊目前大受使用者歡迎，但其雖有多媒體的效果，並無互動的能力。

手持裝置支援多媒體的能力越來越強，因此除了開發遊戲以外，其他的應用，如賀卡、簡訊、學習類教材或是試題，甚至與家庭監視系統整合，提供即時簡訊或 e-mail 通知服務等等，也都能提高其應用價值。

目前所知道在手持裝置上已有的互動式多媒體應用有：

- Flash Lite: 是專為行動電話開發的 Macromedia Flash 設定檔。 其成長的動力來自強大的 Flash 轉換引擎，不論使用何種作業系統、處理器和螢幕大小，都能提供一致的體驗。 而且還有世界各地強有力的 Flash 開發人員社群提供支援。 初步的意見反應相當明確： Flash 可大幅加速開發適用於行動電話的內容與介面。
- MPEG-4: 由於 MPEG-4 有極佳之壓縮率，所以 MPEG-4 的多媒體檔案大小較小，對於手持裝置的記憶體較 PC 環境受限這個因素，是個絕佳的條件。然而由於 MPEG-4 的互動功能，必須由 BIFS 這個場景描述語言來完成，所以造成一般的數位內容製作者耗時、費力。

1.4 研究動機與目標

1.4.1 MPEG-4 現有問題

MPEG-4 包含了所有最先進的多媒體技術，是眾所皆知的一個世界性技術。但是由於它是由很多的系統元件組成具大的架構，並且橫跨許多的領域，造成它實作上的困難。事實上，現有的 MPEG-4 應用程式都是取 MPEG-4 規格的一部分功能，大部分的公司和研究單位都是把 MPEG-4 當作一種新一代的視訊音訊壓縮技術。他們甚至將 MPEG-4 存成他們自己的檔案格式。

我們發現 MPEG-4 不能被廣為使用的因素有下面幾點:

1. MPEG-4 的系統過於龐大和複雜。

MPEG-4 由各式不同的系統元件所構成，但是其實每個系統元件都有其本身所有的一些問題。即使我們將所有的系統元件都完成了，整合在一起的時候仍會

發生許多的問題，比如說在一個 3D 的空間裡要如何呈現一個視訊(video)?並且因為MPEG-4太過複雜，造成在業界各家的解讀有所不同，造成各家完成的MPEG-4系統互不相容，這就是一個非常嚴重的問題。

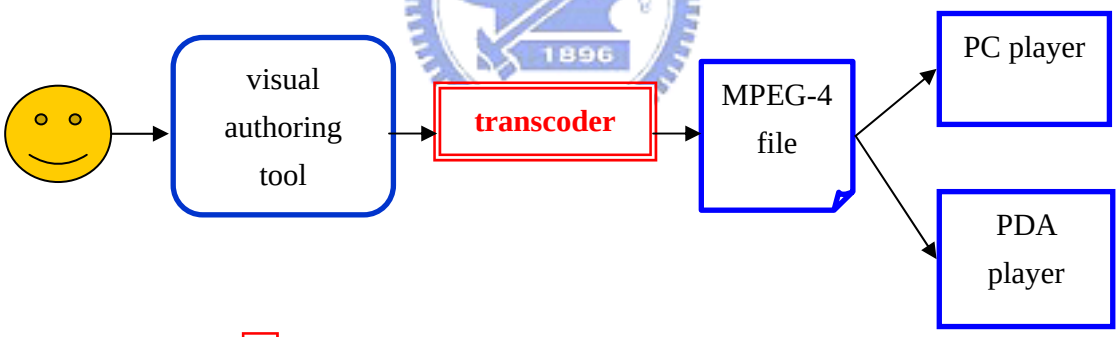
2. 不易製作 MPEG-4 格式的數位內容。

MPEG-4 是一種互動式的多媒體，故它使用了一種場景描述語言(BIFS)來完成這項功能。然而，目前並沒有一項強而有力的視覺化編輯工具可以用來產生MPEG-4 數位內容。

1.4.2 解決方式

經由以上的研究與分析後，我們發現要完成一個完整的 MPEG-4 系統是相當困難的，在 PC 環境上各樣的資源較充足，問題比較少，但是在 PDA 的環境上，限制較多，並且手持裝置的使用者要求也較低，因此我們決定只完成一部分的多媒體呈現功能，包括文字、圖形、聲音這 3 種。

如下圖所示，我們使用一視覺化編輯工具來幫助我們輕鬆的編輯出大量的MPEG-4 多媒體內容，視覺化編輯工具的多媒體格式經過一 MPEG-4 轉繹器(transcoder)轉換成 MPEG-4 檔案，最後使用者使用這些播放器播放這些檔案。本研究和同組同學周宜靖的論文研究分工製作以下 MPEG-4 系統：



註：□ 由周宜靖同學設計製作

圖 3：論文系統架構圖

1.5 研究方法、步驟

為了達成一個以多媒體為內容之互動式行動教學評量系統，我們將以物件導向的技術來設計與規劃整個系統架構，初步可分為下列步驟。

1. 本研究技術之探討:

首先研究 MPEG-4 的系統架構，場景描述語言，還有檔案格式等。

2. 需求分析:

根據相關文獻的內容，及對現有系統作研究分析的過程中，逐步歸納出一理想系統的規格需求。

3. 規劃系統架構:
   根據需求分析，找出系統的功能需求之後，規劃系統所需要用到的模組和物件，來架構出整個系統。
4. 系統設計與實作:
   根據所規劃出之系統架構，逐步將每個模組建構完成，並設計一系統主模組，用以控制系統中各模組呼叫的流程及傳遞模組運作所需之資訊。
5. 提出論文結論與未來發展方向。

1.6 章節概要

在本章中我們先對 MPEG 的技術演進及互動式多媒體及其在手持裝置上的應用做一介紹與探討。最後提出此論文中系統設計製作的動機與研究方法。

接著在第二章中，我們將探討 MPEG-4 的相關技術，另外還有對於 MPEG-4 播放器與編輯手播放器之研究。

在第三章中，我們會將一個理想系統的規格需求歸納整理出來。

第四章中，整個系統的模組設計與分析。

第五章中，我們針對系統中，重要功能、模組及流程加以細部的解說。

第六章中，我們將呈現系統功能範例。

第七章為總結，我們提出此篇論文的結論及未來發展方向。

二、本研究技術之探討

2.1 MPEG-4 標準

MPEG-4 是最新的國際多媒體通訊標準，是由 Moving Picture Expert Group(MPEG)是發展出來的。它是使用一種物件為導向的方式來做多媒體呈現，也就是一個場景是由各樣多媒體物件(media objects)所組成，所有它有很強的整合性功能。MPEG-4 為四個階層的架構，分別為傳輸層、同步層、壓縮層、整合層。這個標準主要是由 Systems[4], Video, Audio 這 3 個部分所組成，systems 提到的是整個系統細部的分工組成，還有場景的構成。後面兩個部分別提到 video 和 audio 的壓縮演算法。由於我的研究並不着重於壓縮的演算法，所以我只介紹 systems 的部分。

2.1.1 MPEG-4 目標

MPEG-4 的目標為提供一個全新的多媒體影音呈現標準。並且支援新的傳輸與存取方式、互動性、不同的傳輸協定，所以它可以達到各式各樣不同的應用，如網路視訊、多媒體廣播(broadcasting)、數位內容資料庫的存取、遊戲、在手持裝置上的多媒體通訊等等。

MPEG-4 最大的進步在於它讓使用者不再處於被動的狀態，它讓使用者可以對場景中的多媒體物件做互動的功能，不論這些物件的來源，可能是圖片、文字、影片、聲音。數位內容的生產者可以讓使用者去改變場件中物件的屬性，或者是物件的行為模式。

2.1.2 MPEG-4 系統架構

ISO/IEC 14496-1[4]這份文件定義了 MPEG-4 的系統架構，這分文件包含下面這些項目:

1. 如何表示多媒體物件(audio-video objects)。
2. 這些多媒體物件在一場景中空間和時間的位置和它們的行為。
3. 如何以位元碼表示處理資料流的資訊。
4. 一個通用性的資料流傳輸介面。

在這一小節中，我們將介紹 MPEG-4 的系統架構。

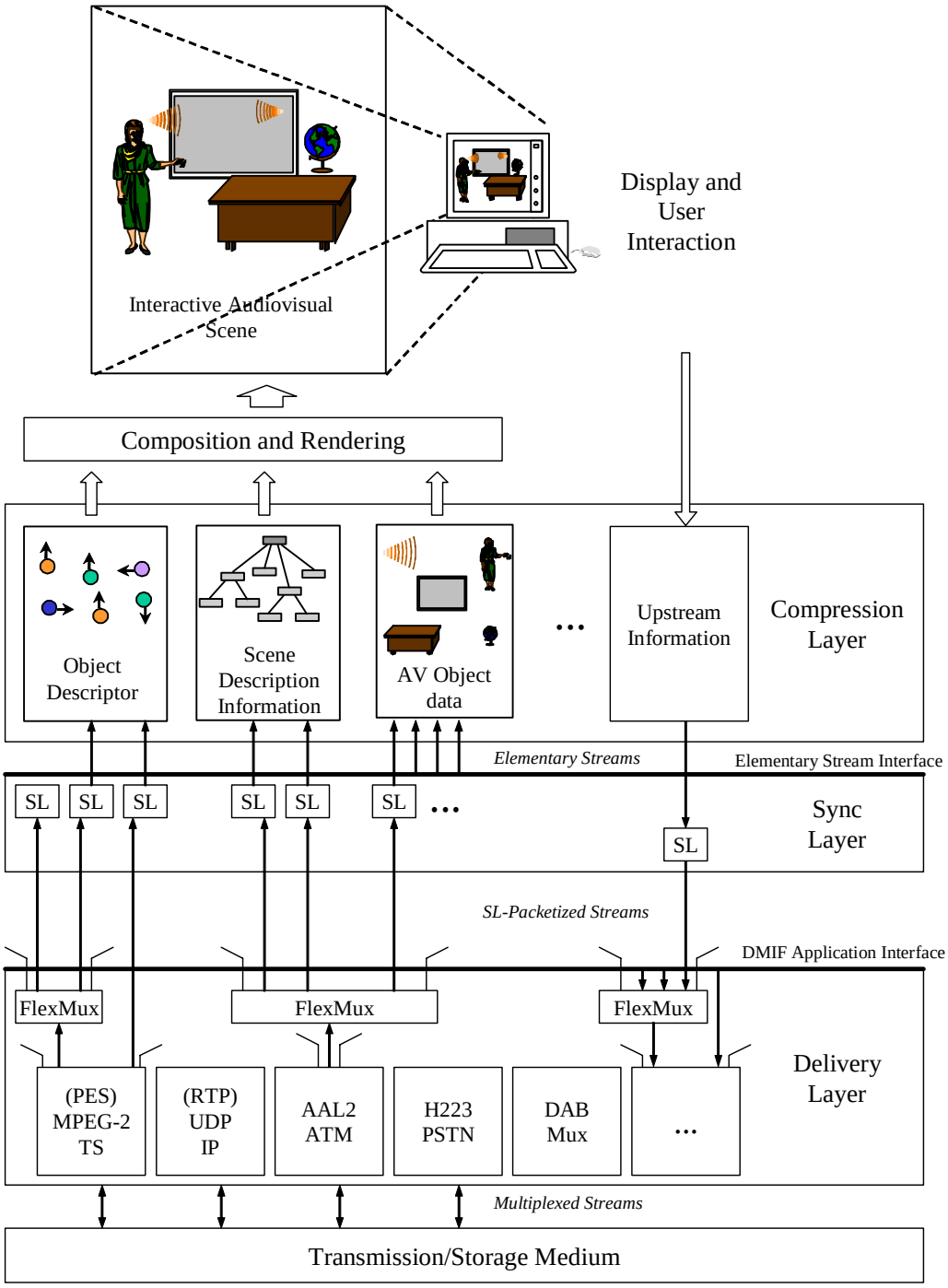


圖 4：MPEG-4 系統架構

MPEG-4 為四層的架構，目的是為了讓每一層可以彼此的分工分明。皆下來以 MPEG-4 資料流的流向來分項訴說各層的功能。

1. 傳輸層(Delivery Layer)

這一層的目的是為了抽象化底層的傳輸協定，在最底層可以包含各種不同的傳輸協定，但是上面幾層不必理會最下面到底是使用那種傳輸協定。因此，

MPEG-4 定義了全新的傳輸架構稱為 Delivery Multimedia Integration Framework(DMIF)。

在 DMIF 中定義了兩種介面 DMIF-Application Interface(DAI)及 DMIF Network Interface(DNI)。資料流由網路實體層往上經由 DNI 進入傳遞層，網路實體層是網路實際的協定，透過這一層使得傳輸層和下層網路所使用的網路協定互相獨立，下層可依需要使用各式的網路協定，如 RTP、ATM、H.223。

通過這層 DNI 介面的資料流我們稱為 FlexMux 資料流，一個 FlexMux 資料流可以是參雜具有同樣網路參數的一個或多個具備同步資訊的物件封包資料流(SL-Packetized stream)。傳遞層會將這些參雜的資料流分為一個一個獨立的物件封包資料流，再經由 DAI 介面與同步層連接。

2. 同步層(Sync Layer)

主要是負責每個封包的切割、時間、連續性等資訊。Elementary Stream 會被切割成最小的資料單位稱為 access units，而只有 access units 才能被賦予時間的資訊，這個動作有點類似為一個信封被貼上郵票的動作，這樣才知道這個資料要在什麼時候被解碼和構成。切割之後的 access units 又會合而成為一個 elementary stream，然後送到下一層。

3. 壓縮層(Compression Layer)

從底層上來的資料在這一層被解碼，被解碼出來的資料最後會被組成及繪出整個畫面。下面列出解碼出來的資料種類:

(1) OD(Object Description) :

描述一個多媒體物件是由那些基本資料流構成，提供基本資料流和場景描述間的連結。一個物件描述式是基本資料流描述式的集合，這些描述式是在描述一個資料流的配置和物件本身或者是場景描述的資訊。物件描述本身就是以基本資料流的方式傳輸，每一個物件描述式都有一個識別碼(object descriptor ID)，利用這個識別碼就可以提供和場景描述的連結。每一個基本資料流也有一個識別碼(elementary stream ID)，透過這個識別碼可以得到每一個資料流。

根據下圖，我們透過物件描述識別碼來得到一個物件描述式，從裡面得到基本資料流描述式，就知道每一個基本資料流的識別碼。

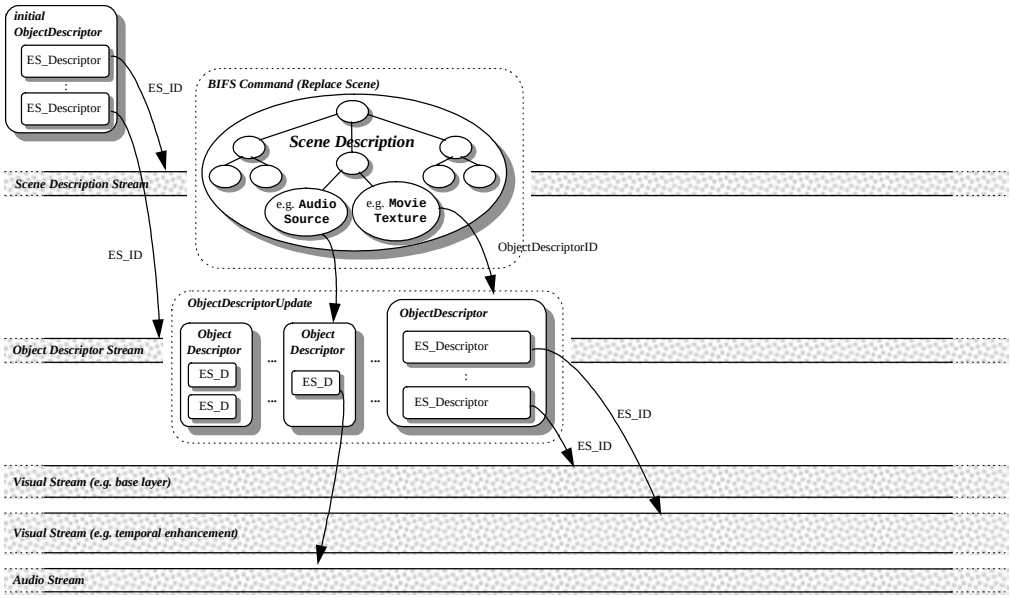


圖 5：OD 使得 SD 和 ES 做一連結

(2) SD(Scene Description)：場景描述資訊是由場景描述語言(scene description language)來完成。場景描述語言是用來描述物件之間的行為，彼如說物件的位置、大小、物件和物件之間的關係等等。MPEG-4 所使用的場景描述語言為 BIFS(Binary Format for Scene)。

- 針對 MPEG-4 system 的 binary format 場景描述語言，根據 VRML[15]的語法所延伸
- Node、Field 為構成場景的基本元素
- 三種 BIFS command (Replace, Insert, Delete)控制場景劇情
- Route 和 Event 控制場景內物件之間訊息的傳遞
- 使用 JavaScript 來記錄較複雜的狀態。

在下一節中會有更詳細的介紹

4. 組成層(Composition Layer)

MPEG4 並沒有規定這一層是用何種方式實作。這一層實作的方式是和 OS 本身是有關的, 如在 windows 下可以使用的繪圖函式有 GDIPlus 及 DirectX2D 等。若是在 PDA 上，就有 pocket pc 2003 的繪圖函式。利用下面幾層的資訊，包括解碼出來的原始資料、物件描述及場景描述，來繪出最後所要呈現給使用者的場景

2.2 MPEG-4 場景描述---BIFS

MPEG-4 內容是由一個或多個 MPEG-4 場景組成，而一個場景由多個多媒體物件所構成，這些多媒體可以是文字、2D 圖形、3D 圖形、聲音或影像。對於這

些物件所構成的場景，需要額外的系統資訊來描述物件間時間與空間的關係，如物件特效與物件的合成還有物件互動的資訊。這些建構場景所需要的系統資訊，稱為場景描述(Scene Description)。如圖所示，這是一個場景的例子，若是將這個場景以樹狀結構來描述就會如圖所示。

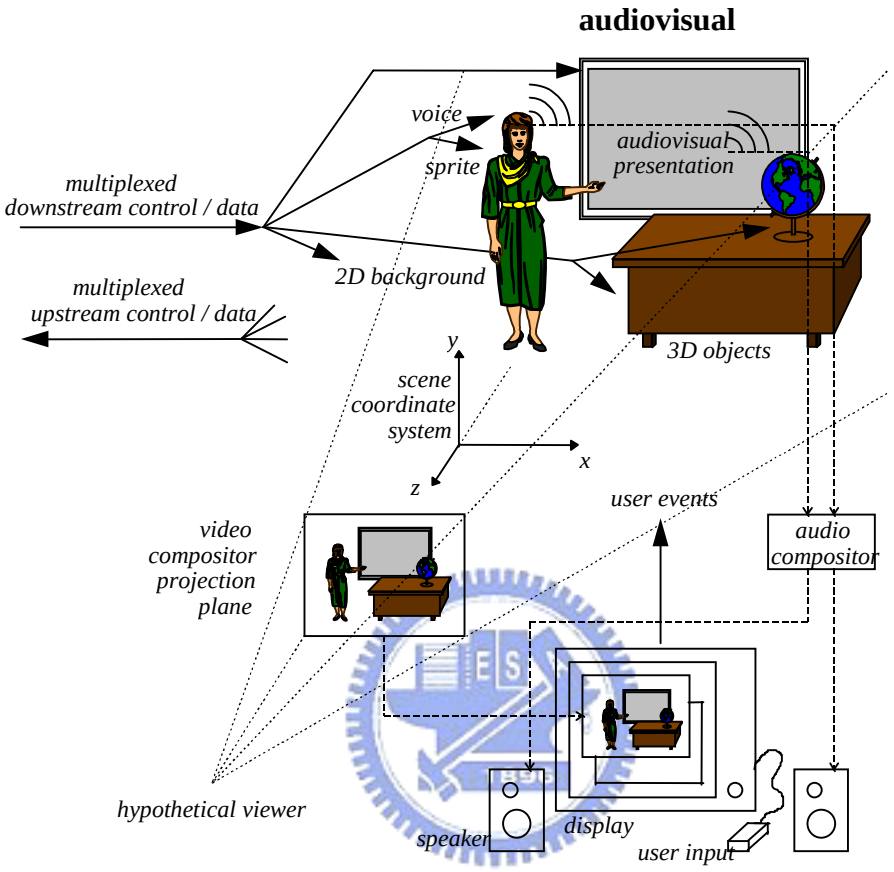


圖 6：一多媒體物件所組成的場景

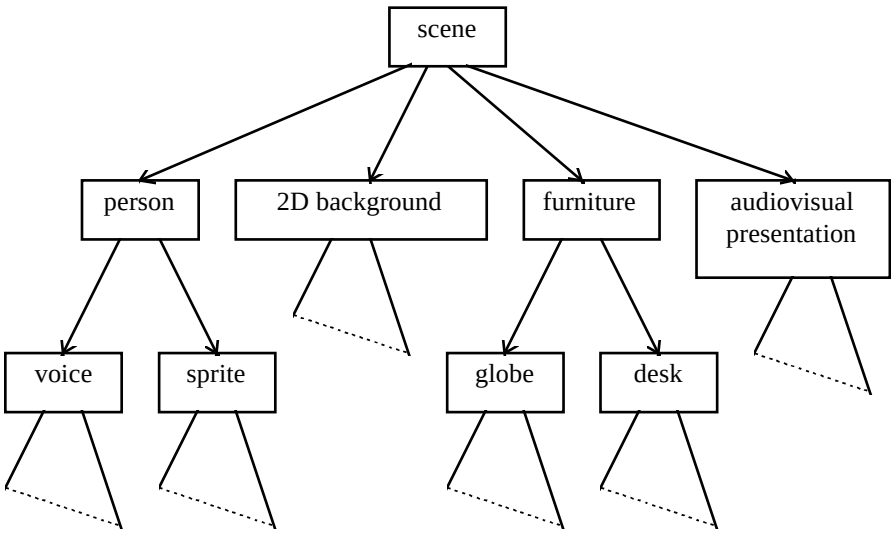


圖 7：上圖中物件所構成的場景樹

在 MPEG-4 媒體資料，是以 BIFS(Binary Format for Scene)來描述場景資訊，

BIFS 是 VRML 語言的擴充，並且 MPEG-4 新增了數個 VRML 所沒有的機制: 資料串流、場景更新、資料壓縮。

在 BIFS 的定義中，一個場景主要由 nodes、fields、routes 所構成。經由各種大小不同的 node 和 field 相構成就可以成為一個多媒體物件(MediaObject)，而多媒體物件的互動行為則必須由 routes 來完成。

- Node : 場景構成的基本單位，可是看為是一種物件
- Field : field 為 node 的屬性，舉例來說 FontStyle node 所擁有的 fields 有以下: family, style, size, spacing。
- Event :
  - 大部分的 Node 有能力去處理進入事件(eventIns)，靠著 eventIns 指示， Node 可改變目前的狀態
  - Node 可以對改變的狀態有所反應，藉著送出一些發生事件 (eventOuts)
- Route : 為 field 之間互相傳遞事件(event)的管道。有些特別的功能必須利用 routes 才能完成，如使物件在場景中的移動、接收滑鼠的 click event 等。

在場景描述的架構中，場景可以隨時間產生不同的變化，所以 BIFS 定義了場景描述命令(BIFS Command)來完成這項工作，可以插入、刪除、取代節點和取代整個節點，透過這些命令可以改變整個場景的樹狀結構及節點內容，達到場景轉換的目的。

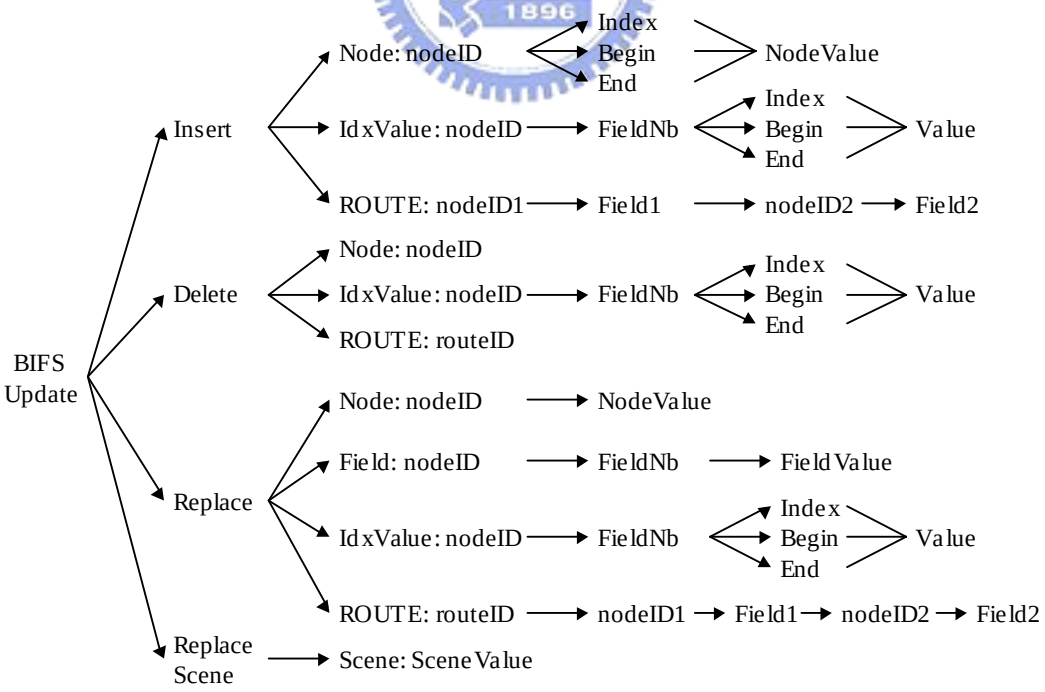


圖 8：BIFS 指令的種類

下面這個例子是用route來達成動畫的效果，這裡的動畫是一多媒體物件沿著它所設的路徑移動。TimeSensor節點產生時間資訊給PositionInterpolator2D節點，

PositionInterpolator2D節點產生位置資訊給Transform2D節點，這樣這個多媒體物件就會延著動畫路徑移動。

BIFS 語法:

<Route fromNode="Node1" fromField="field1" toNode="Node2" toField="field2"/>

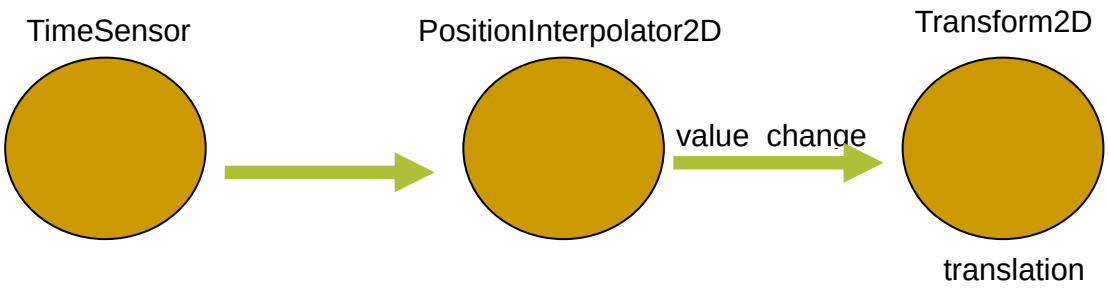


圖 9：Route 的運作模式

2.3 MPEG-4 檔案格式

MPEG-4 定義專屬的檔案格式，是以 QuickTime 的檔案格式做一個延伸，將檔案格式視為一個容器，以容納各種不同多媒體實體資料，如 JPEG 圖片檔，WAV、MP3 音效檔，MPEG-1、MPEG-2 圖片檔，另外使用者也可以用自訂的檔案格式來放進 MPEG-4 檔案裡面，因此 MPEG-4 檔案格式有易於管理、具有彈性等特點。

MPEG-4 檔案的資料結構是由 **Atom** 所構成。Atom 可以視為是一種容器(container)，每一 Atom 皆可含更小的數個 Atoms。MPEG4 檔案主要分為兩個 Atom: mdat 和 moov。mdat 主要擺放所有的原始資料，即是多媒體物件經過壓縮後會以 tracks 的形式放在這個 atom 裡面，而 moov 內含所有描述那些 track 的額外資訊，如：track 本身的大小和距檔案的距離。

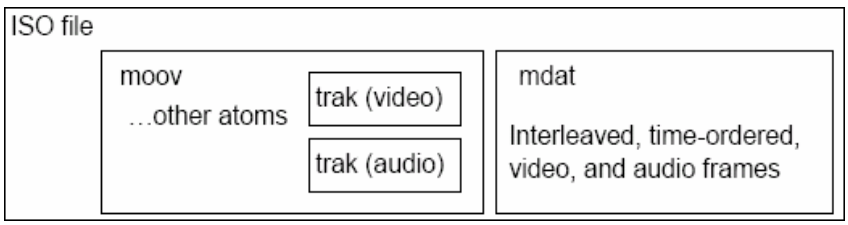


圖 10：MPEG-4 檔案格式

- mdat 內含四種 type 的 track chunk
  - sdsm : 所有 BIFS 的 Binary Code
  - odsm : 所有 Object Description 的 Binary Code

- vide : 呈現此影像的 raw data (圖片、影片)
- sound : 呈現此音效的 raw data (聲音)

- moov 內含 3 種 atom
  - mvhd :一些額外資訊(如 creation time)
  - iods : 初使化物件描述子(InitialObjectDescriptor)資訊，根據此資訊可取得另外兩個系統資料流，就是 SD 與 OD。
  - trak : 有四種 type(odsm, sdsm, vide, sound)。必須從此獲得關於每個 track 的相關資訊。最重 要的兩個資訊為 track 的**本身大小(size)**與**距檔頭的距離(offset)**，才可從 mdat 中把 track 的原始資料取出。

2.4 MPEG-4 相關工作

MPEG-4 是個既龐大且複雜的多媒體標準，在接下來的部分我們會討論的相關工作:

1. MPEG-4 player
2. MPEG-4 場景編輯工具
3. 其它相關工作

大部分這些系統實作缺乏對系統架構和系統功能的詳細說明。在 MPEG-4 的標準包含了一個實作的參考軟體 IM1(Implementation Model 1)。IM1 是目前最重要的系統實作，並且它被業界和學術界拿來做他們系統的核心。所以接下來我們會詳細介紹 IM1 以及資策會的播放器，其它部分簡略帶過。

2.4.1 Implementation Model 1(IM1)

在 1997 年 5 月，MPEG 組織建立兩個特別的團隊 : System Software Implementaion 1 and 2, 就是我們所知道的 IM1 和 IM2。這個兩團隊專注於 MPEG-4 系統部分的開發，而 IM1 決定用 C++來開發系統，IM2 決定用 Java 來開發系統。以下是這兩個團體的目標 :

1. 驗證規格的可應用性
2. 展示標準的規格
3. 讓開發當作一個參考

然而，IM2 沒有成功，所以接下來的說明我們會針對 IM1。

2.4.1.1 IM1 的工具

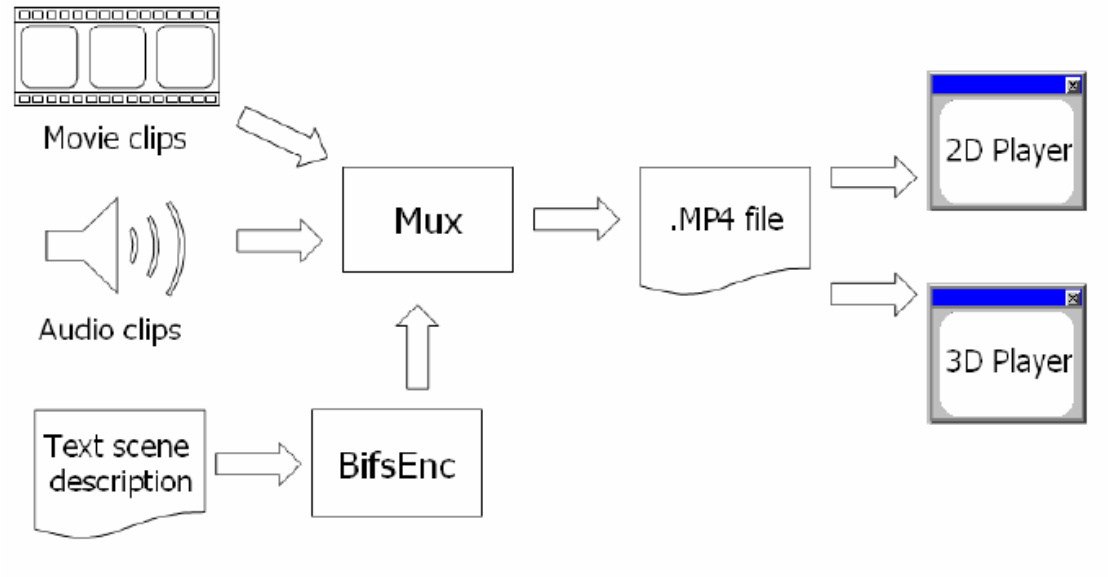


圖 11：IM1 MPEG-4 的壓縮工具

IM1 開發了以下數個工具(如圖所示)，這些是：

- BifsEnc

  BifsEnc 是一個用來將 MPEG-4 的場景描述文字檔壓縮成 BIFS 和 OD 資料。當一個 MPEG-4 的內容產生者想去產生一些數位內容時，他們必須了解 BIFS 的語法然後用來描述場景。

- Mux Tool

  此工具會將一個描述 MPEG-4 檔裡每一個追縱(track)的描述檔，壓縮成為一個 MP4 檔案。Mux tool 會將視訊(video)檔案、音訊(audio)檔案和 BIFS 檔案(由 BifsEnc 壓縮而成)成為一個完整的 MP4 檔案。

- 2D Player

  IM1 2D 播放器是用來播放只有包含 BIFS 的 2D 的節點(node)MPEG-4 內容。

- 3D Player

  當一個 MPEG-4 內容含有 3D 的節點時，是用 3D 播放器來播放。

要產生一個 MPEG-4 內容，根據上圖，作者必須把視訊壓成 G.723 的檔案格式和把音訊壓成 H.263 檔案格式，但是 IM1 並沒有提供這些壓縮工具。而且作者必須用 BIFS 語言來描述場景資訊和 BIFS 指令。BifsEnc 是將這些文字檔壓成 BIFS 追縱(track)檔案格式，然後使用 Mux tool 來將這些追縱(track)資料合成一個完整的 MP4 檔案。最後，我們用 2D 播放器來播放 2D MPEG-4 內容，用 3D 的播放器來播放 3D MPEG-4 內容。

2.4.1.2 IM1 系統架構

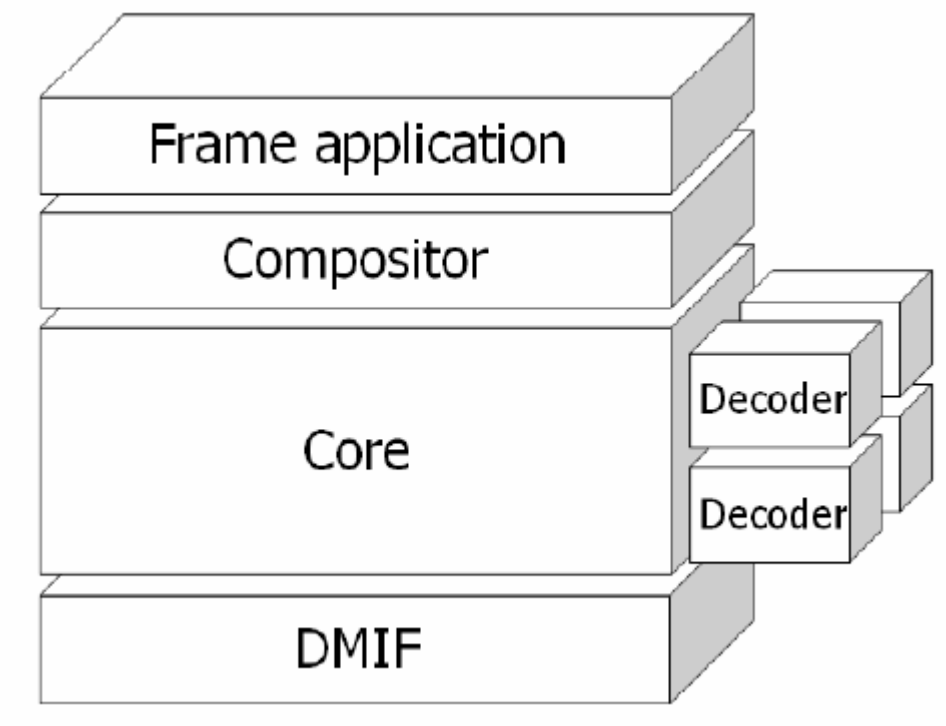


圖 12：IM1 系統架構佟

IM1 2D/3D 播放器擁有不少模組，主要的模組有下面這些:

1. 核心模組(Core module):
   核心模組是在 IM1 中最重要的模組，這個模組就是 IM1 系統的核心，它擁有以下子模組：
   - 執行模組(Executive module): 用來控制 IM1 系統的執行流程。
   - 同步層模組(Sync Layer module): 控制緩衝區和 SL 封包的時間。
   - BIFS/OD 解碼模組(BIFS/OD decoder module): 解開場景資訊和物件描述資訊。
   - 場景管理模場: 處理場景路徑(routes)和事件(events)。
2. 解碼模組(Decoder module):
   解碼模組將從解碼緩衝區得到的存取單位(access units)解壓縮，並且將這些解壓縮出來的資料存放到組成緩衝區(composition buffer)，最後組成並且繪成整個畫面。
3. 組成模組(Compositor module):
   這個模組會去走訪放在場景管理(scene manager)中場景樹的資料，而且將組成單位從組成緩衝區取出，然後組成並且繪出整個場景畫面。
4. 架構應用模組(Frame application module):
   負責 IM1 播放器的外觀展現，這個模組偵察出使用者動作(比如滑鼠的點擊和動作)還有提供圖形使都者介面(GUI)。
5. DMIF 模組

此模組會連到遠端並且接收從網路傳送過來的 SL 封包。此模組處理 IM1 系統中的傳輸服務和分工服務(demultiplex-services)。

IM1 player 在一開始的架構是希望能夠做到各平台獨立(platform independent)，但是現在它還是都在 MicroSoft Windows 平台下運作較多。IM1 將許多現有的軟體工具作一個整合的工作，所以它可以幫助我們深入了解 MPEG-4 標準。下面是 IM1 player 的系統模組圖：

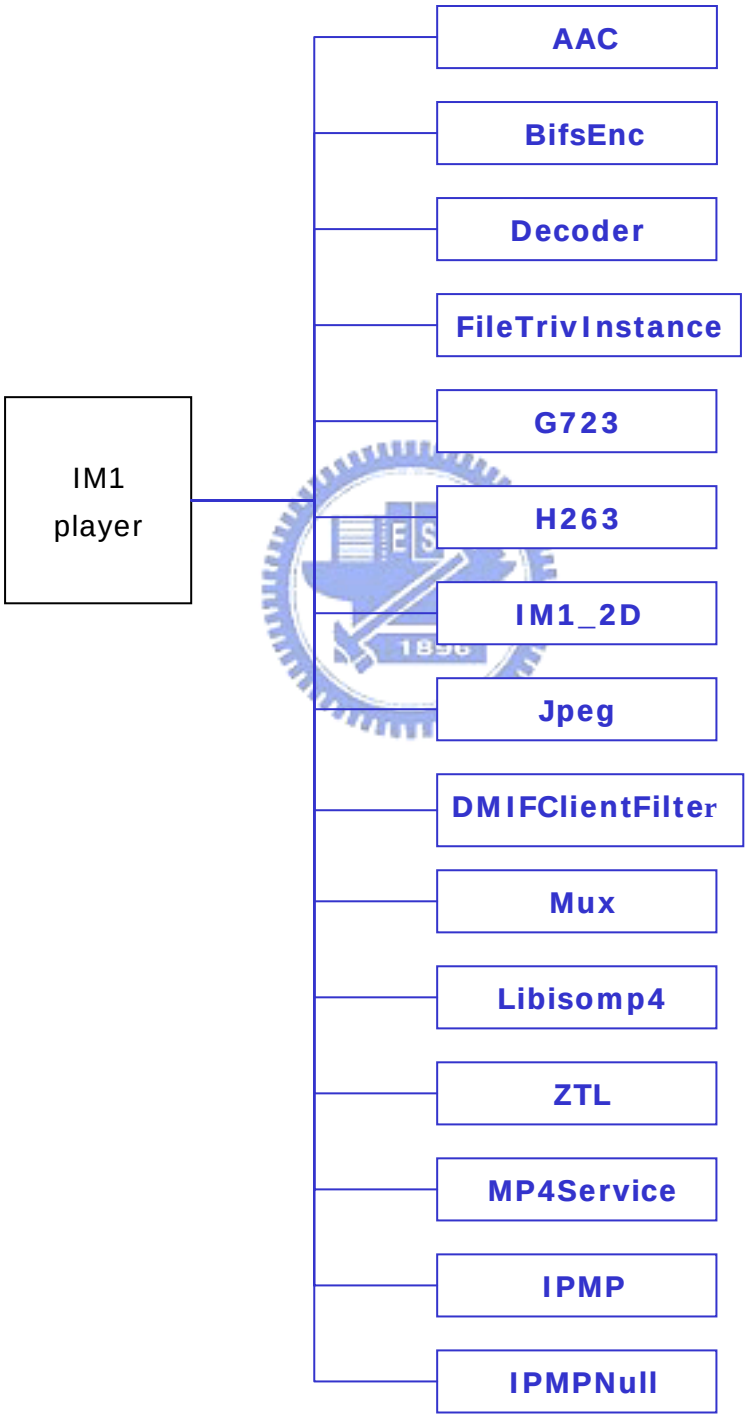


圖 13：IM1 播放器的系統模組圖

IM1 只支援四種 codec，分別是 Jpeg AAC H263 G723

1. Jpeg --- 圖片壓縮格式 jpg
2. AAC --- MPEG-4 Audio 縮壓格式 aac
3. H263 --- MPEG-4 Video 縮壓格式 h263
4. G723 --- 語音壓縮格式

下圖為 IM1 player 的系統流程圖，列出主要物件之間的流程：[1]

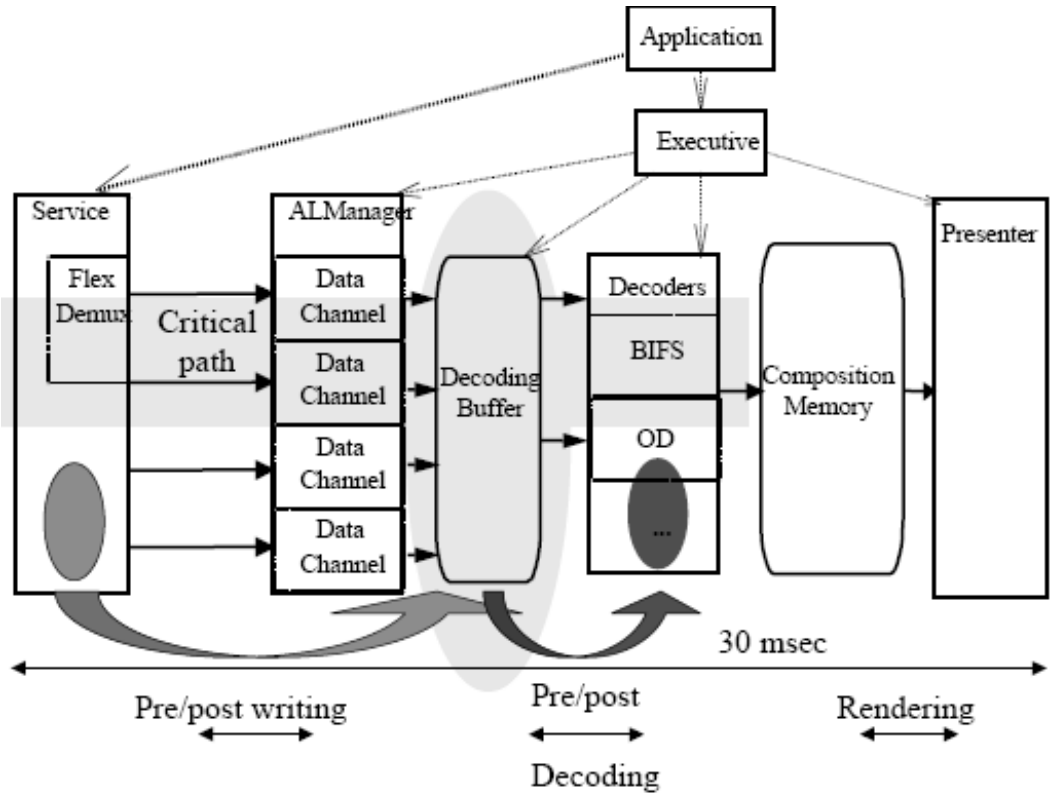


圖 14：IM1 播放器的系統流程圖

程式的進入點為 Application 物件，Application 會依序執行以下工作：

1. 產生顯示視窗
2. 起始 Executive 物件
3. 取得一 DMIF session
4. 起始一個 Executive session
5. 將控制權交給 Executive 物件
6. 將 session 關閉
7. 結束 Executive 物件
8. 結束整個程式

Executive 物件控制 session 中資料的流向，它擁有自己的執行緒並且會執行以下工作：

1. 起始一 Presenter 物件
2. 起始 ALManager 物件，此物件會起始一 Service 物件

3. 解析從 Service 傳送過來的 iods
4. 起始 BIFSDecoders 物件(負責解析 BIFS 和 OD)
5. 將 BIFS 和 OD 資料流傳送給 BIFSDecoders 物件
6. 必要之時，將會產生額外的 ALManager 物件來處理一些 URL 定位的資料流
7. 將控權交給 Presenter 物件
8. 結束 session
9. 告知 Application 物件此 session 已結束

DMIF 層是由 Service 類別完成，它是用來處理和底層傳輸協定的溝通並且負責傳送資料封包。而一些網路上的事件會交給 ALManager 物件處理。ALManager 類別與 DataChannel 類別是 MPEG-4 的 adaption 層，它接收由 Service 物件傳來的封包，將封包解開並傳送 AU 給合適的元件。

Presenter 類別是用來控制場景的呈現，並且是平台獨立的，下面是場景的控制流程：

1. 由 Executive 物件是由 Presenter 物件所起始
2. 當 BIFSDecoder 把場景建立完成之後，才呼叫 Presnter 的起始函式
3. Presenter 的執行緒其實是一個迴圈，它每過幾個百萬分之一秒就會重繪整個畫面
4. 每一個 MediaObject 物件會繪出它自己及它自己的子節點
5. 最後 Presenter 物件結束

控制場景的類別有很多，其中最重要的類別是 MediaObject。MediaObject 是所有的基礎類別，所有在 BIFS 中的節點都是繼承此類別。MediaObject 擁有以下特性：

1. 一個 MediaObject 的物件有零或多個欄位(fields)，欄位是由 NodeField 類別為其資料結構。
2. 一個 MediaObject 的物件可以是零或多個其它多媒體物件的父類別。
3. 一個 MediaObject 的物件可繪出它自己和它的子類別。
4. 一個 MediaObject 的物件必須擁有具集指令，存放解析後的 BIFS 指令解析

在 MPEG-4 之欄位和欄位之間透過路徑傳遞訊息，路徑是由 Route 類別來處理，它負責兩個 NodeField 物件的連接。

資策會的 MPEG-4 播放器[24]是由 IM1 MPEG-4 播放器修改而來，裡面大部分的模組是從 IM1 播放器繼承修改而來，所以大部分一些基本的功能是與 IM1 類似，而系統的程式流程也是非常地類似。以下列出資策會 MPEG-4 player 新增的系統模組。

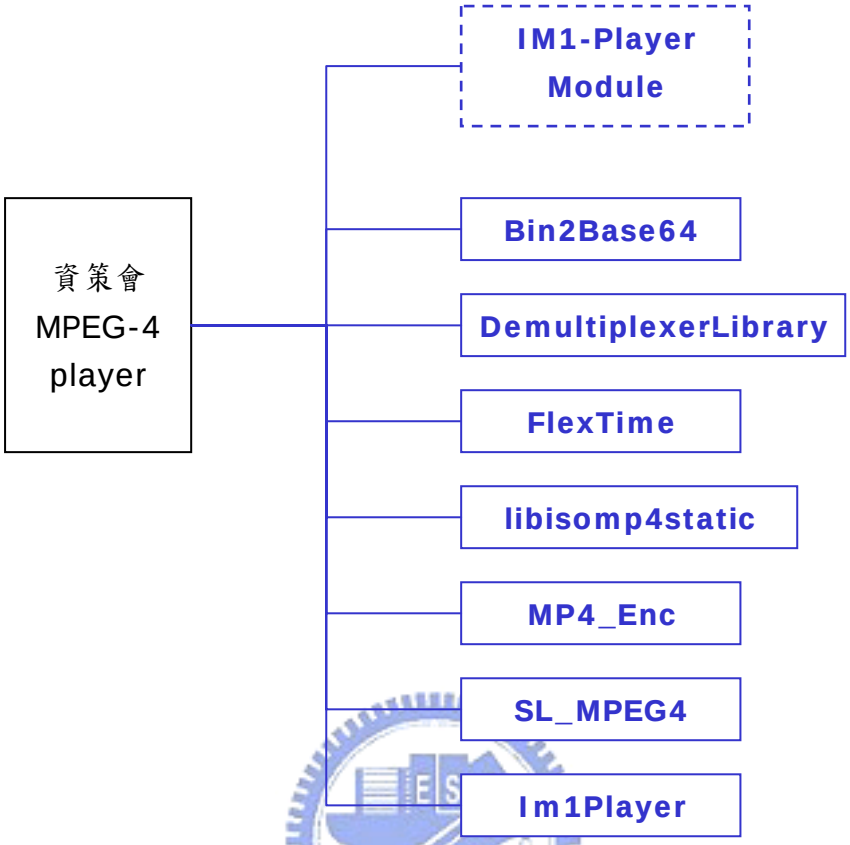


圖 15：MPEG-4 系統模組

2.4.2 MPEG-4 相關研究

在這一節中我們將會列出 MPEG-4 的相關研究工作。MPEG-4 的研究工作分為兩個大方向，一個是關於視訊音訊的壓縮演算法的研究，另外一個是關於 MPEG-4 的系統部分也就是場景和互動的相關研究。我們接下來介紹的研究著重於第二個。

2.4.2.1 學術界

- **E.N.S.T**(Ecole Nationale Superieure des Telecommunications, Paris) 是一個有名的 MPEG-4 研究中心，發展出好幾套 MPEG-4 工具。

1. MPEG-Pro

MPEG-Pro 是一個 MPEG-4 的編輯工具和展示工具，此工具提供了視覺化編輯介面。

2. mp4edit

MPEG-4 文字編輯器，必須輸入幾個文字檔來產生一個 MP4 檔。

3. Harmonia

此工具為一視覺化工具讓使用者可以編輯一張又一張的投影片，最後轉成 MPEG-4 格式。它提供了視覺化的編輯介面還有樣版系統，讓使用者可以輕鬆的產生最後的 MP4 檔案。

4. OSMO4

OSMO4 是用來呈現 MPEG-4 2D 場景的呈現工具。OSMO4 外掛 Xvid 模組(MPEG-4 視訊)，MAD 模組(MP3 音訊)和 JPEG、PNG(靜態圖片)。

- **MPEG-4 Virtual Studio**

倫敦大學的資訊科學系發展出一套虛擬實際的影片製作工具。在這個計畫中，他們嘗試將 3D 的環境整合到電視的虛擬實境。

- **ViDe MPEG-4 Working Group**

這個團隊在 2000 年 7 月的時候成立，他們相信 MPEG-4 在線上學習、合作、和資源共享上有很大的潛力。這個組織的研究目標主要是找出在工業界願意一起研究發 MPEG-4 的夥伴；了解目前 MPEG-4 技術發展的近況；推展 MPEG-4 的應用在高等教育中的應用；並且推動 MPEG-4 在學術界的知名度。

2.4.2.1 工業界

- **ISMA(Internet Stream Media Alliance)**

ISMA[23]的目標是加速業界對於新的串流多媒體標準的接受度。主要是使用 MPEG-4 為解決
方案，它主要想要對抗的對象為 Microsoft。

ISMA Specifications(ISMA 規格書)

- Presentation Control: BIFS
- File Format: MP4
- Codecs:
  - Speech: CELP
  - Audio: MP3
  - Video: MPEG-4 SP@L1(profile 0), ASP@L3b(profile 1)
- Control Protocol: RTSP/SDP
- Transport Protocol: RTP/RTCP

- **Envivo**

Envivio[19]自 1999 年從法國電信獨立出來後，發起人Julien Signe's便將其在法國電信R&D部門超過三年以上的工程經驗加以延展。Envivio針對廣播界、有線電視業及內容開發商市場…等，提供MPEG4 串流媒體解決方案－兼具技術及商業考量的產品，公司目標為在專業媒體產業上成為主宰MPEG4 解決方案的供應者。

Envivio的團隊陣容堅強，其重要核心人物不是在MPEG4 定義階段表現活躍，就是曾經開發過市場領先的產品及公司，並集合一群在MPEG4 標準、圖像處理及網路系統方面出類拔粹的專家。

Envivio率先提出Mpeg-4 串流影音解決方案，成為市場上技術的領先者，這樣end-to-end的Mpeg-4 串流解決方案包括了互動式媒體、各項管理元件、online或offline的影音轉檔，皆可在envivio最新的Mpeg-4 技術中找到。目前Envivio擁有 11 個在MPEG4 標準上的專利，包括在法國電信領時期超過一千萬美金的投資、智慧財產及二代MPEG4 軟體，如今全都屬於Envivio的資產。

- **iVAST**

iVAST MPEG-4 平臺改進了互動式多媒體內容在寬頻網路上的傳輸。iVAST 平臺是建立在 MPEG-4 標準上，提供一整套的軟體來解決產生、傳輸、和呈現互動式多媒體，並且提供在娛樂、企業、教育等領域的解決方案。

iVAST 主要提供三种軟體：進行編碼壓縮的“iVAST Studio Encode”、多媒體伺服器“iVAST Media Server”及播放器“iVAST Experience Player Win-32 Basic”。

“iVAST Studio Encode”軟體可對音頻/視頻進行 MPEG-4 編碼，具有可進行 64kbit/秒的低位速度發送的高壓縮性能。“iVAST Media Server”為發送 MPEG-4 音頻/視頻的軟體，在發送內容方面增加了交互式與在線點播功能。“iVAST Experience Player Win-32 Basic”為用於內容播放的媒体播放器，其特點是內容發送者可以加上卡通畫面等進行剪切後提供給用戶。用戶可以通過使用 iVAST 的各內容發送者來下載這一播放器。

- **Tvia**

Tvia 是一專門製作多媒體顯示處理器晶片的公司，發表了消費性產品(DVD 播放器，互動視電視，機上盒，手持裝置)的嵌入式 MPEG-4 軟體。

它發表了一個全新的技術: TVM4，這是一整合 TV 和網際網路資源的互動式應用。典型的應用包括如下: VOD、互動式呈現、數位監視、網路瀏覽。

TVM4 全然支援 MPEG-4 工業規格。TVM4 主要為一嵌入式應用程式並且它針對資源和對特殊應用的需求來增加擴展它的功能。

2.5 智勝國際編輯手 player

智勝國際編輯手[3]player 是編輯手這一套編輯工具的播放器，它是由一幕一幕(scene)的場景所構成，而場景裡有自己的場景劇情來描述場景自己演員本身的行為模式還有和使用者互動的功能。編輯手擁有自己的描述語言來描述場景劇情，而由於這套工具是由本實驗室所研發出來，我們可以拿到原始碼來分析他的播放步驟。

其播放步驟如下：

1. Step1: 載入場景描述檔。這個步驟裡會經過 lex/yacc 這套工具來分析場景描述檔的語法，得到的資訊會存放在一個 tree structure 裡，然後經過 tree traversing，再把這些資訊擷取出來放在程式的資料結構之中。
   - ❑ 與場景有關的資訊
     - ■ 背景圖，背景音樂
   - ❑ 演員資訊
     - ■ 演員類型: 文字、圖片、聲音、影片
     - ■ 演員屬性: 位置、大小
   - ❑ 劇情
     - ■ 開場，退場劇情
     - ■ 互動劇情
2. Step2: 播放。根據 step1 的資訊，要把演員的劇情呈現出來，我們會把一些效果交給子系統去處裡，而互動式功能的部分，可以使用 windows 本身的 message dispatching 技術來做到。
   - ❑ Actor 本身的效果，由下面的系統來處理
     - ■ 繪圖系統
     - ■ 聲音系統
     - ■ 過場特效
   - ❑ 場景劇情
     - ■ 開退場劇情
     - ■ 互動式劇情

2.6 編輯手 player vs. MPEG-4 player

在上面兩節之中，我們分別介紹了編輯手 player 和 MPEG-4 player。接下來，我們就來分析這兩種 player 的差異和比較，下面這個圖表是編輯手 player 和 MPEG-4 的特性比較：

	編輯手 Player	MPEG-4 Player
架構	有 MPEG-4 的最上面兩層	擁有分工分明的四層架構
設計角度	使用者	系統
壓縮方式	皆無，所有多媒體檔案為分散	對場景描述、物件描述及多媒體資料
規格	較簡單	較複雜
多媒體支援	jpg, png, gif, wav, mp3,midi, mpg, wmv	jpg, aac, mpeg-1, mpeg-2
播放軸線	event-based	time-based
場景構成	開場、退場、互動劇情	Access Units
串流傳輸	無	有

表 1：編輯手 player 和 MPEG-4 player 之差異比較

2.7 PC 和 PDA 發展環境之差異

為了要在 PDA 環境[16]上發展 MPEG-4 播放器的程式，故要了解在 PDA 環境上會受到那些在 PC 環境上發展不會受到的限制。

	PC	PDA
記憶體大小	256MB ~1GB	10~100MB
CPU 速度	1GB~3GB	100MB~1GB
發展工具	MVS 6.0, MVS .Net	MS EVC++
顯示器大小	15~17 吋	6 cm × 7 cm
作業系統	Microsoft Windows	Microsoft WinCE
多媒體支援	jpg, png, gif, aac, mpeg-1, mpeg-2, mp3, wav, wmv,	Jpg, bmp, wav
資料匯流排	64 bits	32 bits
顏色	16 位元~64 位元	65536

表 2：PC 和 PDA 發展環境之差異

三、系統功能需求分析

3.1 PDA MPEG-4 播放器之功能需求

1. 檔案清單:
   列出在 PDA 裝置上所有的 MPEG-4 檔案，可供使用者點選要開啟的 MPEG-4 檔案。

2. 開啟檔案:
   開啟在上一個功能點選的 MPEG-4 檔案，並且讀取 MPEG-4 檔案裡面的內容，然後把這些資料載入記憶體。

3. 播放 MPEG-4 檔案
   在畫面上展示出多媒體呈現。

4. 暫停播放 MPEG-4 檔案
   暫止播放目前正在播放的檔案。

5. 顯示播放時間:
   顯示從開始播放到現在播放到第幾秒。

6. 視訊大小:
   可以讓使用選擇播放時顯示在畫面上的大小。提供兩種選擇，一種是原始大小(original size)，忠實顯示出原來 MPEG-4 檔案中所設定的場景大小；另外一種選擇是(full screen)，讓原來的場景大小被擴充或縮小到和螢幕大小一樣。

7. 顯示世界資訊(World Info):
   顯示一些關於 MPEG-4 本身的一些資訊，比如說這個數位內容的製作者、製作時間等等。

8. 調整播放速度:
   選擇播放時的速度，有三種選擇：原速、快一倍(200%)、慢一倍(50%)。

9. 結束時清除螢幕:
   當播放結束的時候是否要將螢幕清除。

10. 關閉播放器:

結束整個播放器程式。

3.2 系統功能設計及製作上之考量

在進入下一章系統架構規劃之前，我們先提出在系統的設計、製作時我們的考量點為何。

我們希望製作出來的系統在將來的維護、擴充及發展上能作到十分的有彈性，因而我們用物件導向的技術來設計及製作此系統。但是由於物件導向技術有一個對 PDA 非常不利的條件，就是物件導向技術所發展出來的程式效能會比程序導向的程式較差，而 PDA 的計算能力又較 PC 環境為差，所以在這個考量下，有些模組的確是用”物件”的方式來包裝的，而又有一些模組純粹是以程序來組成。這個的分別的方式是，若是這個模組的功能性很強，很有可能會一直被執行到，那麼這個模組就必須是以程序來完成的；相對的，如果這個模組主要是儲存一些資料，那麼就會以物件的方式來完成這個模組。

由於本研究的重點不在於 codec 壓縮和解壓縮的演算法，所以我們會使用許多現成的 codec 來達成我們的需求，系統中可能會用到的 codec 有 Jpeg, AAC, H263, 必須要到網路上去找開放源碼而且是在 pocket pc 平台上的程式。

在 PDA 上面的另外一個限制條件是記憶體容量較小，所以當圖片、影片這些資料在 PC 上由數位內容生產者(Content Creator)處理的時候，我們希望先做一些處理讓這些資料迎合 PDA 的大小。這是因為在 BIFS 裡有個語法，可以把影視類的多媒體物件做縮放的動作。舉個例子，比如說一張圖他的原大小是 100x100 pixel，為了要在 PDA 上被完整顯示，它的比率(scale)屬性會被調成 0.5x0.5，也就是它顯示出來的大小是 50x50 pixel。因此，為了節省記憶體空間，要先把這多媒體資料利用程式先縮放好之後，再用 MPEG-4 編碼器(encoder)壓成 MPEG-4 檔。

要解決這個問題，最後決定是在周宜靖的轉譯器裡面加入這一段功能。因為對數位內容生產者(content creator)這是一件很麻煩的事，而若要在 PDA 最後顯示時才處理的話，對於 PDA 本身是一種負擔，因為 PDA 的繪圖速度本來也較 PC 差，若是利用比率屬性來調整大小，會造成畫面上有動畫的物件嚴重閃爍(blinking)，且物件動畫的移動速度變慢。所以在壓縮成一 MPEG-4 檔案之前，就先把物件縮成適當大小，是最好的選擇。

四、系統架構

4.1 取得 Atom 資料

MPEG-4 播放器的第一步就是要知道如何去解析 MPEG-4 檔案，在第一章的時候我們介紹了 MPEG-4 的檔案格式，在這一個小節中我們將會介紹該如何用程式去解析 atom，得到這些 atom 的資料。

每個 atom 的前 8 個位元組，首先的 4 個位元組是表示 atom 的大小，後面的 4 個位元組是表是 atom 的型態。根據這 8 個位元我們就可以得到每個 atom 的二元碼資料，並且把這些資料存到它們所屬的資料結構中。

下圖是 atom 的首先 8 個位元組：

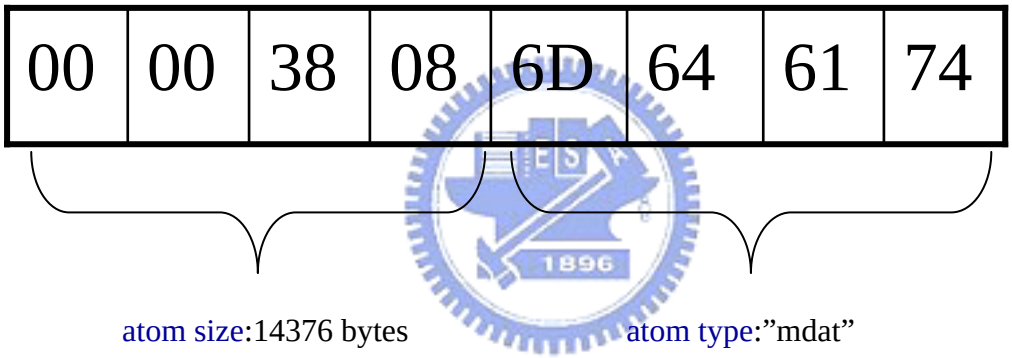


圖 16：每個 atom 的前 8 個位元組

4.2 多媒體物件原始資料

得到所有 atom 的 byte code 之後，我們就可以分析這些 byte code 中的資訊，來得到多媒體物件的原始資料。

在第一章的時候，我們知道一個 MPEG-4 檔案分成 mdat 和 moov 這兩個主要的 atom，mdat 放的主要是所有多媒體物件的原始資料，也包括 BIFS 和 OD 的二位元碼，而因為 mdat 本身並沒有資訊指出每一個多媒體物件的資料區段的界限，這個資料區段我們稱為資料軌(track)。每個在 mdat 中的資料軌在 moov 中都有一個相對應的資料軌，這個 moov 中的資料軌會描錄在 mdat 的資料軌的大小(size)和距檔頭的長度(offset)，這樣我們就可以完整的取得在 mdat 中多媒體物件的原始資料。

下圖是取得 mdat 每個資料軌(track)的示意圖：

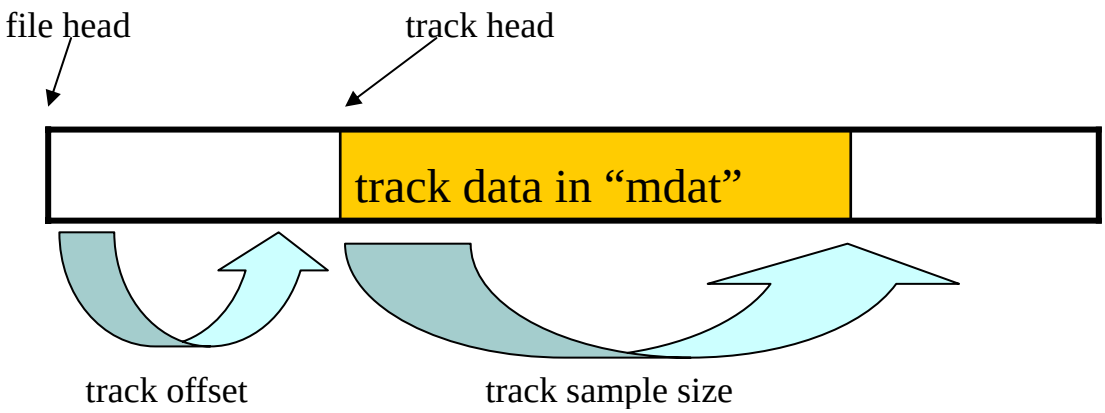


圖 17：取得資料軌(track)資料

4.3 解析物件資訊與場景資訊

要構成整個場景，我們必須要知道兩件事，一是物件本身的資訊，包括物件本身會用到那些資料軌(一個物件可由許多資料軌組成)，還有這些資料軌用的壓縮方法(codec)等等。另一件事是組成場景的資訊，舉個例子來說，一張圖的大小和在畫面上的位置，或者是物件的行為模式和物件與使用者之間的互動關係。

為了要得到這兩樣資訊，是放在 OD 和 BIFS 的資料軌之中。在上一小節中，雖然我們得到了每個資料軌的資料，但我們並不知道那一個是 OD 或 BIFS 的資料軌，但是在解析 moov 中的資料的時候我們會知道每一個資料軌的識別碼(track ID)。

經過下面這些步驟之後，我們會知道 OD 和 BIFS 的資料軌識別碼，並且我們就可以更進一步的去解析物件資訊和場景資訊。

1. step1: 解析 iods，得到 OD 和 BIFS 的資料軌識別碼。在一般的情況下，BIFS 的 track ID 為 1，而 OD 的 track ID 為 2。
2. step2: 解析 OD track 的二位元碼，得到所有物件有關的資訊。
3. step3: 解析 SD track 的二位元碼，得到組成場景的資訊。

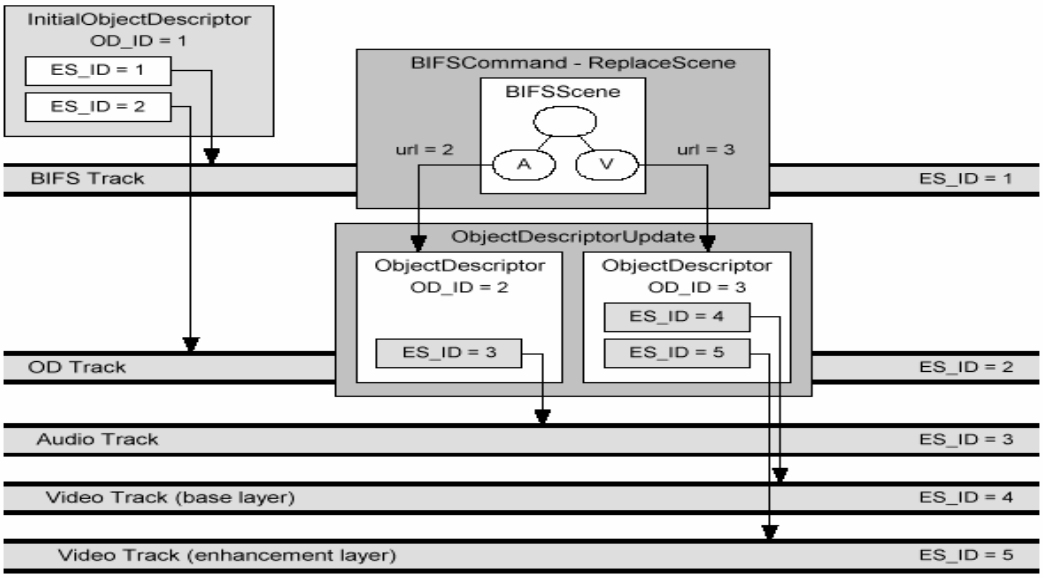


圖 18：連接物件描述與場景描述

4.4 系統架構圖

在此小節中，我們描述出整體系統的模組與架構。

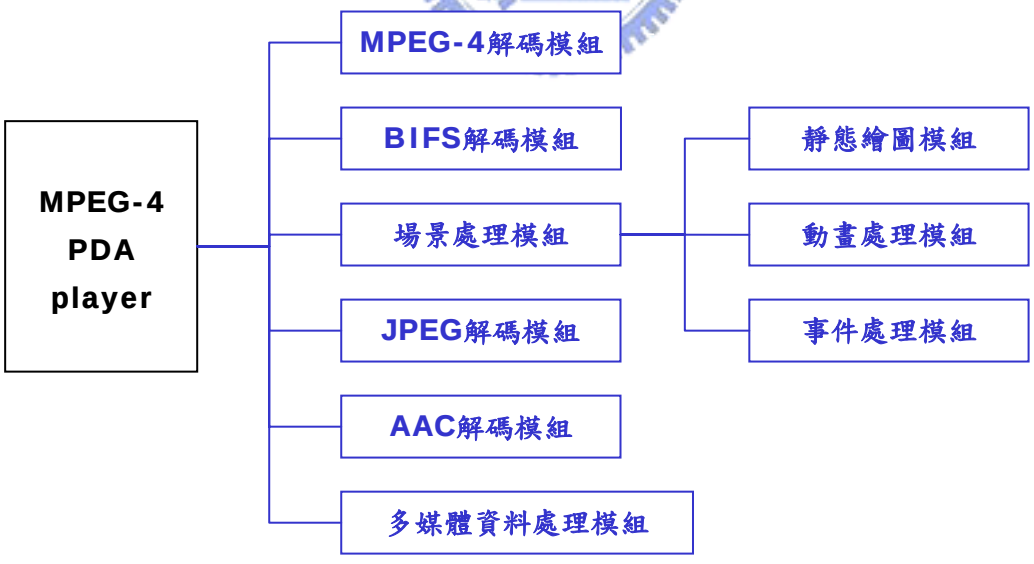


圖 19：系統架構圖

每個模組代表了一個獨立的功能機制，各個模組間相互獨立，由主系統規劃

在特定機制運作時，系統中所應呼叫的元件及流程。裡面有用到一些 codec: JPEG, AAC 這些對於多媒體資料的解碼演算法，為了節省程式開發的時間，我都使用現有的開放源碼來做資料解碼。凡是模組的名字裡有 decoder 的，都是和資料的處理有關，而 scene handler 和 event handler 是負責繪出整個場景和處理使用者對多媒體物件的互動。

4.5 系統資料流向

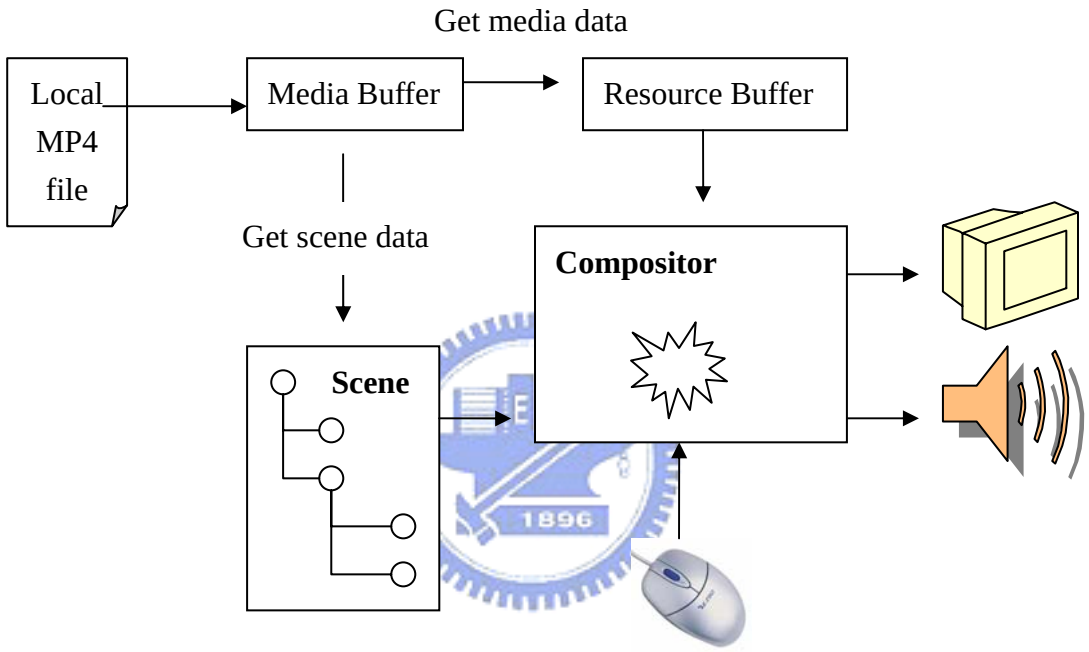


圖 20：系統的資料流向

依上圖所示，一個本地端的 MP4 檔案，經過 MP4 解碼模組解開之後，會將資料放在多媒體緩衝區(Media Buffer)裡面，然後 BIFS 解碼模組會把它所需的場景資料取出，然後解析這個場景資料。多媒體資料處理模組會把多媒體資料取出放在資源緩衝區(Resource Buffer)裡面。圖中的 Compositor 即是場景處理模場，負責根據場景資料和多媒體資料，把整個場景繪出以及動畫的處理，還有處理使用者的互動事件。

4.6 使用者介面

使用者介面如圖 21 所示，白色的部分即是使用者所接觸的程式介面，如同一般視窗應用程式(Windows Application)，我們利用功能選單的方式讓使用者選擇其所需之功能應用，我們將一簡單的流程描述如下:

當使用者將 player 啟動時的第一個畫面，會列出裝置裡面所有的 mp4 檔，然後當使用者點選了需要的 mp4 檔，就會進入第二個畫面，開始播放。若是使用者要用到一些其他的功能或者他要另外開啟別的檔案，就可以點選上面的功能鍵來使用這些功能。

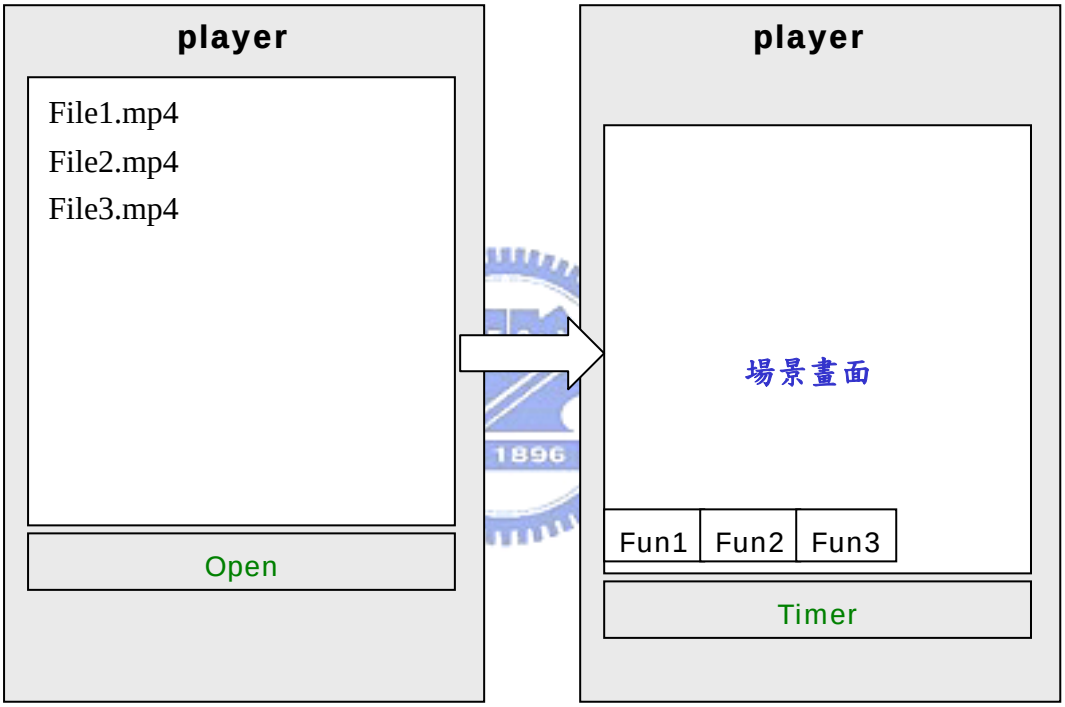


圖 21 系統使用者介面示意圖

五、系統設計與實作

5.1 主要功能設計

在此節中，我們將重要的模組、功能及機制作一詳細的描述。

5.1.1 系統程式流程

以下是系統的程式流程圖，首先使用者透過使用者介面選擇要播放的MPEG-4檔案，接著MPEG-4解析模組會將這個MPEG-4檔案解開，取出每一個atom的資料，並且將每一個追縱的原始資料(raw data)取出。BIFS解析模組負責解析場景的資料，

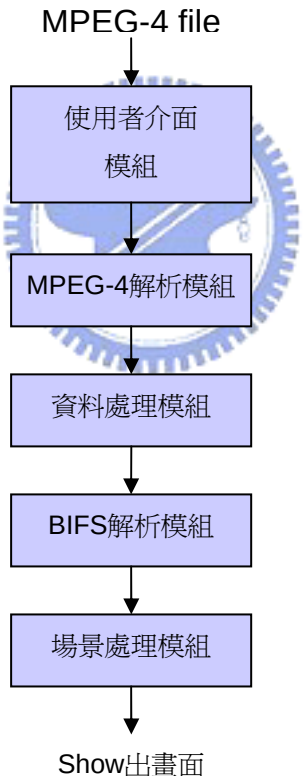


圖 22：系統程式流程圖

5.1.1 解析 atom 流程圖

在上一章的時候，我們已經知道，透過前 8 個 byte 我們就可以得到 atom 的資料。每一個 atom 都有可能會有子 atom，而且可能不只一個，所以我們必須一

直不斷的遞迴呼叫下去，當一個 atom 沒有子 atom 的時候，表示它是最底層的 atom 這時就可以把它的資料讀取出來存到它所屬的 atom 物件之中。

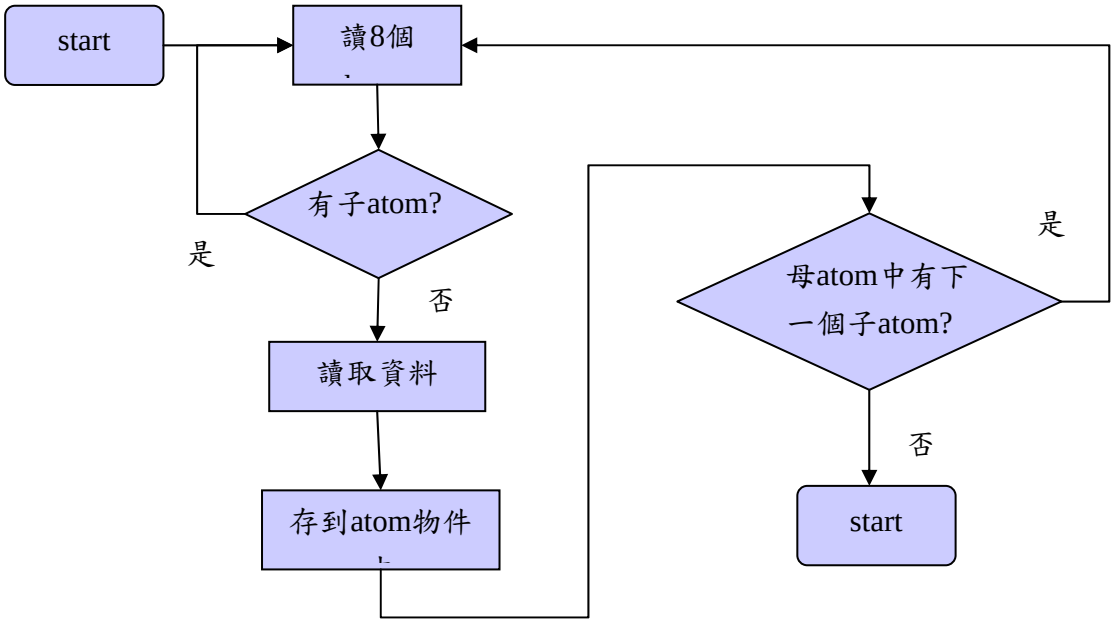


圖 23：解析 atom 流程圖

5.1.3 atom 類別圖

每個 atom 都有自己專屬的欄位，用來存放每個 atom 獨有資訊。我所設計的 atom 類別，都是由 FullAtom 類別繼承下來，故所有的 atom 都有共同的欄位如下：type, uuid, size, version, flag。且每個 atom 都有自己的 parse 函式，用來讀取 MPEG-4 檔案中的位元組資料，並且把它存在 atom 中。

由於 atom 的種類非常的多，一張圖無法畫得下，所以只畫出一些重要的 atom。

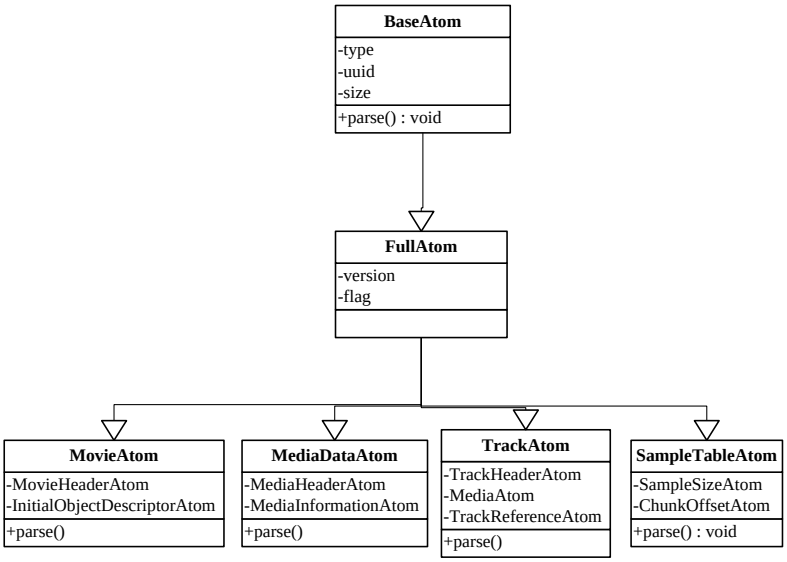


圖 24：atom 類別圖

5.1.4 解析 BIFS 二位元碼流程圖

我們得 BIFS track 的二位元碼的原始資料之後，就可以開始解析每個 bit 所代表的意義，最重要的資訊有 4 種，分別是 node id、node type、field type、和 field value，因為多媒體物件是由 node 所構成，所以經過解析之後，就可以知道每個多媒體物件是由那些 node 所構成，而每個 field 中的值，正是多媒體物件本身在場景之中的屬性。另外還有 route，這個資訊和動畫與互動這兩項功能有關。

GetBits 函式會根據所輸入的參數: 目前要從 BIFS 資料追縱裡得到幾個位元，來讀取位元資料。

BifsDecode 就是解析 BIFS 二位元碼的函式，首先它會處理最外層的 3 種節點：OrderedGroup, Group, Conditional，整個 BIFS 場景描述其實就是這三種最外層的節點所構成。這 3 種節點，都可內含更多的子節點，每一個節點都有自己的解析函式名為 NodeDecode 函式，Node 就是節點的名字，舉例來說 Group 的解析函式為 GroupDecode。

每一個 NodeDecode 函式由於知道目前正在解析的是那一種節點，所以就會照著那個節點所定義的欄位去解析。

Conditional 節點會紀錄很多動畫與互動的一些 BIFS-Command，所以在解開這些 BIFS-Command 時有一個函式為 DecodeCommandBuffer，我定義了 Scenario 類別來存放這些指令，使得場景執行的時候，只要去讀 Scenario 所存放的資訊就能使得場景被正確呈現出來。

另外，還有一 routing table 的資料結構，來存放有那些新增的路徑(route)，從這個節點的這個欄位到達那個節點的那個欄位的資訊記錄下來，當有某節點一欄位有事件產生出來之時(eventOut)會去檢查這個 table，是否有路徑導向另一個

節點的欄位，做出相對應的處理。

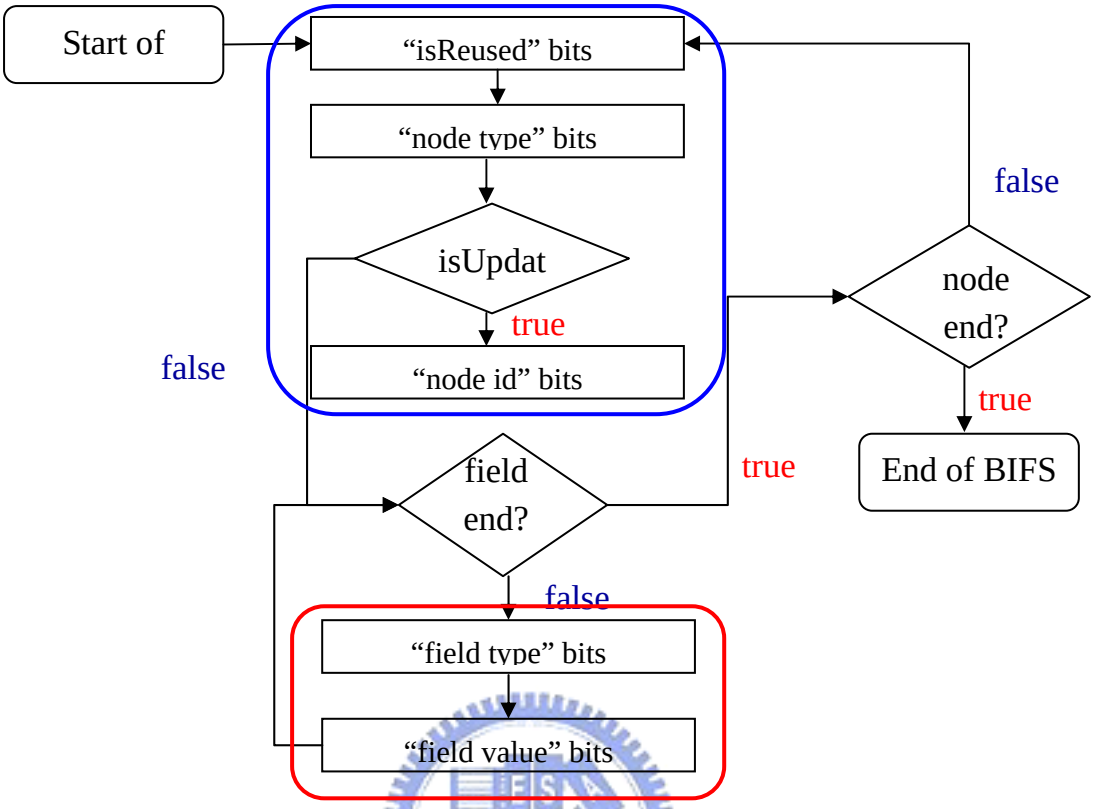


圖 25：解析 node 流程圖

每一個 Group 節點可視為一個多媒體物件，下面的圖說明一圖片多媒體物件用 BIFS 來呈現：

```
APPEND TO Sc053.children
 Group {
       children [
           Transform2D {  translation   -106.875 77.75
                children [
                    Shape {
                        appearance
                            Appearance {
                                material
                                    Material2D { filled   true }

                                texture
                                    ImageTexture { url   9 }
                            }
                        geometry
                            Rectangle { size   1.0 1.0 }
                    }
                ]
           }
       ]
 }
```

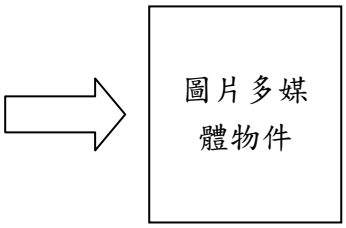


圖 26：BIFS 中一多媒體物件的語法

5.1.5 MediaObjects 類別圖

在解所 BIFS 的資料的同時，亦會把所有物件場景資訊存入資料結構中，此資料結構就是我所設計的 MediaObject。MediaObject 又分為 3 個子類別，分別是 ImageMediaObject, TextMediaObject, AudioMediaObject 分別存放的是圖片、文字、聲音的多媒體物件資料，這些資料包括了物件本身屬性如位置、大小、移動路徑、多媒體資料位址。

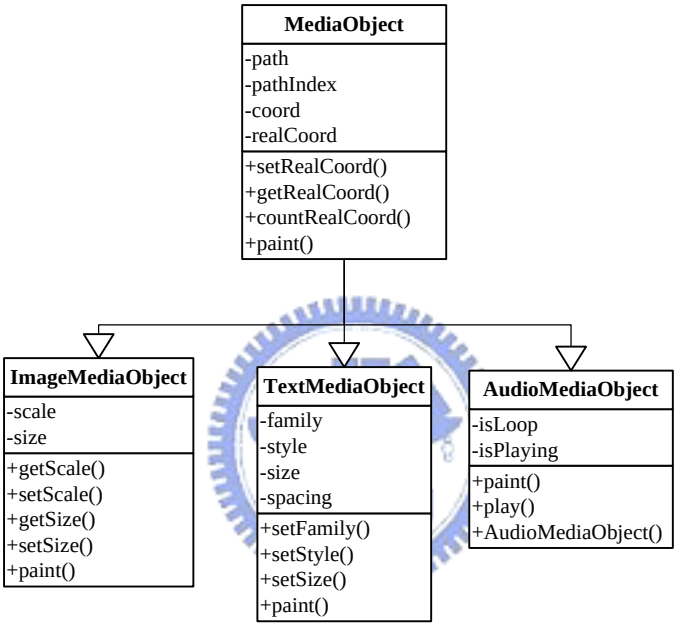


圖 27：多媒體物件類別圖

如此設計是使用到物件導向程式設計”多形”的特性，有些行為模式是所有子類別都有，但是各自的定義又有所不同，而且利用”多形”，就可以使用父類別的指標來做存取。舉例來說，在繪出整個場景的時候，其實是呼叫每個多媒體物件的”paint”函式，而 ImageMediaObject 和 TextMediaObject 的 paint 函式的實作方式是完全不同的，一個是把一張圖繪出而另一個是把文字顯示在畫面上，故他們都有從 MediaObject 繼承而來的 paint 函式，但是它們有各自的實作方法。用同樣的父指標指向它們，然後呼叫父指標的 paint 函式，那麼執行時就會執行各自的 paint 函式，最後構成整個場景。

六、系統功能展示

本研究製作一個在手持裝置上的 MPEG-4 播放器，其操作的流程與系統的外觀，將在本章做一個介紹。首先，使用者啟動播放器程式，接著選擇要播放的 MP4 檔案。操作順序如圖到圖，使用者可直接播放 MP4 檔案，暫停播放，或者結束整個播放程式，在播放器的右下角有顯示目前開始播放的時間。

6.1 系統功能說明

在這一節中，我們將用圖來說明，本系統所擁有的功能。

(1) 檔案選取畫面：下圖是開啟播放器程式後，會進入的第一個畫面，使用者點選要播放的 MPEG-4 檔案，然後按下開啟按鈕之後，就會進入下一個畫面。

圖 28：檔案選取畫面

(2)播放畫面：第二個畫面是播放 MPEG-4 的畫面，在這個使用者界面中共有 5 個功能，分別是播放、暫停、關閉、顯示播放時間、和互動功能，如下圖所示。

圖 29：播放畫面

6.2 系統操作流程

以下我們展示播放一個 MP4 檔案的例子：

(1) 開啟 MPEG-4 播放程式：

圖 30：開啟 MPEG-4 播放程式

(2) 開啟播放器程式之後，它會自動列出裝置內所有的 MP4 檔案，使者可以選取要播放的 MPEG-4 檔案，之後把它開啟。

圖 31：開啟 MPEG-4 檔案

(3) 進入到顯示的畫面，這時使用者只要按下”播放”鈕，就可以將剛剛所選取的 MPEG-4 檔案，呈現出來。

圖 32：進入顯示畫面

(4) 按下播放鈕開始播放

圖 33：按下播放按鈕

(5) 按下暫停鈕將播放暫停下來，這個時候右下角的計時會停上，而且使用者對於多媒體的互動功能也會失效。

圖 34：按下暫停鈕

(6) 按下播放鈕使播放繼續，計時器也會恢復計時。

圖 35：再次按下播放鈕

(8) 使用者和場景中的多媒體物件互動。

圖 36：使用者和場景中的多媒體物件互動

(8) 按下關閉鈕，關閉整個播放程式。

圖 37：關閉播放程式

七、結論

7.1 總結

當初在研究 MPEG-4 這個題目的時候，並不確定自己要實作出來的系統是什麼。以為 MPEG-4 這個新的多媒體技術應該會成為下一代的多媒體技術主流，在花費我不少時間之後(1 個多學期)才有一點頭緒，事實上光是要看懂那一分四百多頁的規格書，就讓人要耗盡心血。深入研究之後，發現 MPEG-4 的規格實在是太過於複雜，這是因為當初設計這套系統的人，想要把所有領域的多媒體都囊括進去，包括 2D, 3D, VRML, SMIL, SVG, streaming 等各領域的技術，結果反而造成系統既龐大又過於複雜，所以在業界目前並沒有被廣泛的使用。但是雖然在系統部分不為大眾所接受，但是它對於視訊和音訊的壓縮卻被廣泛的採納。我的結論是，要在短時間之內靠一人之力完成整個 MPEG-4 系統是非常困難的，所以最後是決定把我會用到的部分萃取出來，只實作這些需要的部分。

我們使用了視覺化多媒體編輯工具產生大量的多媒體內容，並且經過轉換成為 MPEG-4 格式，可以在 PC 或 PDA 環境上的播放器播放，這樣即是解決 MPEG-4 多媒體內容不易大量被產生的問題。

7.2 未來發展方向

在這一節中，我們會列出一些目前系統尚需補足的功能，或者是一些研究方向在本論文中未提及的:

1. 系統傳輸層: 在這一篇論文中，我們所針對的 MPEG-4 的資料處理，都只是針對本機檔案(local file)，也就是不考慮網路上的資料流傳輸。所以這一層是完全不在系統實作裡面的。

2. 系統同步層: 實作的系統沒有考慮資料的同步問題，這也是因為我所選擇的 MPEG-4 多媒體內容並沒有很複雜的結構，並且資料都是靜態的，所以也忽略了這一層。如果有實作傳輸層，這一層還是必要的。

3. JavaScript: BIFS 的互動功能是有限的，這是因為 BIFS 並沒有定義變數來儲存狀態(state)，所以有兩個以上的狀態要紀錄下來的時候，只靠 BIFS 就沒有辦法處理了，這時就必須借助其他的描述語言，在 MPEG-4 使用了 JavaScript 來完成這項不足點。

4. MPEG-J: 由於 BIFS 並不能支援如程式般複雜的邏輯控制，所以在 MPEG-4 Version 2 擴展了這項新的功能，它是直接以一種程式控制的機制來取代 BIFS 中以參數控制的機制。MPEG-J 提供了豐富的 API，不僅在場景互動的功能上還有播放器相關的一些功能。MPEG-J 是在 MPEG-4 內容中加上 Java code 的串流，所以這意味著它一樣具有 Java 跨平台的特性，如在 PC、PDA、Setup-Box。

5. 由於 PDA 上所使用的功能較少，我們實作出來的 MPEG-4 播放器只有符合規格的某部分，像是 BIFS 中關於 node 的定義，我們就只實作了一部分的 node，並且有些實作出來的 node 中的 field 也並沒有每一個 field 都實作出來，所以若是轉換出來的 BIFS 用到新的 node，那麼系統就必須加入新的 node。

參考文獻或資料

[1] Yang-Chih Huang, “A Study on Interactive Support of MPEG-4”, Master Thesis of N.C.T.U Taiwan, 2000.

[2] Jang-Jer Tsai, “MPEG-4 Systems and Its Simulations”, Master Thesis of N.C.T.U Taiwan, 1999.

[3] Chorng-Shiuh Koong, "A Component-based Visual Scenario construction envirenment for non-programming users to create interactive electronic books",N.C.T.U Taiwan, dissertation,2000

[4] ISO/IEC 14496-1, Information Technology – Coding of Audio-Visual Objects: Systems, 2d Edition, 2001.

[5] IM1 Core code + authoring tools Version 4.0 ISO/IEC JTC1/SC29/WG11 M5450

[6] ISO/IEC 14496-1, Information Technology – Coding of Audio-Visual Objects: Systems Amendment: Textual Format: XMT Chapters 14-17, 2d Edition, 2001.

[7] ISO/IEC 14496-5, Information Technology – Coding of Audio-Visual Objects: Reference Software, 2d Edition, 2001.

[8] International Standard ISO/IEC 14772-1:1997 – VRML 97: The Virtual Reality Modeling Language.

[9] Li-Hsien Chen, “The Implementation and Application of MPEG-4 Based Object Description Framework and Streaming Manager”, Master Thesis of N.T.U Taiwan, 2001.

[10] Meng-Chi Hsieh, “Design and Implementation of a MPEG-4 System”, Doctor Thesis of N.T.U Taiwan, 2003.

[11] Jen-Chieh Wang, “Entropy based Object Selection and Automatic Encryption Mechanism in MPEG-4 Streaming”, Master Thesis of N.C.K.U Taiwan, 2004.

[12] Po-Kai Chiu, “Design and Implement a MP4 Video-on- System Based on PC Cluster”, Master Thesis of N.C.K.U Taiwan, 2003.

[13] Jun-Rong Yang, “Implementing a MPEG-4 Interactive Environment with MPEG-J”, Master Thesis of N.C.K.U Taiwan, 2003.

[14] Grady Booch, James Rumbaugh, Ivar Jacobson著，UML使用手冊，張裕益譯，博碩文化，台北，2003。

[15] 嚴子翔著，VRML 虛擬實境網頁語言，知城數位，2001。

[16] 楊先民著，PocketPC 程式設計自學手冊，學貫行銷，2002。

[17] MPEG Home Page. http://mpeg.telecomitalialab.com

[18] IBM MPEG-4 Technologies，http://www.research.ibm.com/mpeg4/indexjs.htm

[19] EnvivoTV and Envivo Encoding Station web site: http://www.envivio.com

[20] GPAC Project web site: http://gpac.sourceforge.net/

[21] MPEG-4 @ ENST : http://www.comelec.enst.fr/~dufourd/mpeg-4/index.html

[22] MPEG Industry Forum: http://www.mpegif.org/

[23] ISMA: http://www.isma.tv/

[24] 資策會多媒體實驗室，http://www.iii.org.tw/special/

附錄 1:支援的 atom

每個 atom 資料前的 4 個位元組為 atom 的類型(type)，所有每個 atom 都有其相對應的縮寫，根據這些縮寫，我們就知道 atom 的類別。然而有一些 atom 我雖有定義儲存它的資料結構，然而卻沒有定義解析它的函式，這是因為這些 atom 中的資料不是我所需要的，所以在解析的過程中我把它忽略掉了。

下表列出本系統所支援的 atom 列表：

Supporting MPEG-4 Atom	Abbreviation	Supporting Decoding
ChunkOffsetAtomType	stco	Yes
DataInformationAtom	dinf	No
DataReferenceAtom	dref	No
HandlerAtom	hdlr	No
InitialObjectDescriptorAtom	iods	Yes
MediaAtom	mdia	Yes
MediaDataAtom	mdat	Yes
MediaHeaderAtom	mdhd	No
MediaInformationAtom	minf	Yes
MovieAtom	moov	Yes
MovieHeaderAtom	mvhd	Yes
SampleDescriptionAtom	stsd	Yes
SampleSizeAtom	stsz	Yes
SampleTableAtom	stbl	Yes
SampleToChunkAtom	stsc	No
SyncSampleAtom	stss	No
TimeToSampleAtom	stts	No
TrackAtom	trak	Yes
TrackHeaderAtom	tkhd	Yes
TrackReferenceAtom	tref	Yes

表 3：支援的 MPEG-4 atom

附錄 2: 支援的節點(nodes)和欄位(fields)

此章會列此本論文系統所支援的節點和欄位。實作這些節點的規格是 ISO/IEC 14496-1:2001，由於實作時間有限，並且我們實際上只針對編輯手播放器的功能來實作我們需要的節點，而實作的節點中的每一個欄位也並非都能被正確的解析出來。

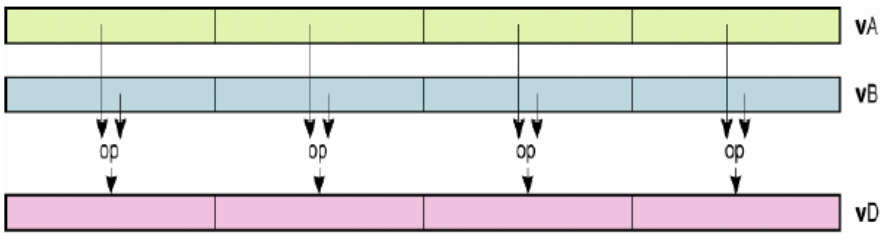
Supporting MPEG-4 Nodes	Supporting MPEG-4 Fields
Appearence	material, textture
AudioClip	loop, url
Bitmap	scale
Conditional	buffer
FontStyle	family, style, justify, size, spacing, leftToright
Geometry	bitmap, rectangle, text
Group	
ImageTexture	url
Material2D	emissiveColor, filled, transparency
MediaControl	url, mediaStartTime, mediaStopTime, mediaSpeed, loop, preRoll, mute, enabled
OrderedGroup	
PositionInterpolator2D	key, keyValue
Rect	size
Shape	appearance, geometry
Sound2D	source
Text	string, fontStyle
TimeSensor	enabled, cycleInterval
TouchSensor	enabled
Transform2D	translation

表 4：支援的節點和欄位

附錄 3: AAC Decoding 在手持裝置上的實作

現在 aac decoder 在手持裝置上的實作方式，都是直接與 DSP/FPGA 做結合。由於手持裝置的運算能力有限，會採用以下的方式來加速解碼的速度：

1. DSP 程式碼最佳化：
   - 程式寫作方式最佳化：是關於程式指令和運算元的最佳化，有下面幾種方式：
     i. 不同資料型態有不同的運算時間，位元數愈少的資料型態運算愈快，比如：8 bit char 的加法比 64 bit double 的加法速度快很多
     ii. 使用內部基本函式：有一些對於 DSP 架構設計出來的基本函式，可以減少運算時間
     iii. 利用 SIMD 特性：一個指令同時處理完好幾筆資料，如下圖所示：vA 和 vB 各有四筆資料，本來要經過依序做完四次運算，但是依此架構就可以平行處理完四筆資料，之後存放到 vD。



     iv. 編繹器的迴圈展開方式(loop unrolling)：展開愈大會造成程式碼大小愈大，但是運算速度會變得愈快。如下圖所示，(a)是原來的迴圈，(b)將迴圈展開一次，(c)將迴圈完全展開

```
(a)
int i,a=0,b=0;
for (i=0;i<10;i++)
{
        a+=i;
        b+=i;
}
```

```
(b)
int i,a=0,b=0;
for (i=0;i<10;i++)
        a+=i;
for (i=0;i<10;i++)
        b+=i;
```

```
(c)
int i=0,a=0,b=0;
a+=i; b+=i; i++;
a+=i; b+=i; i++;
a+=i; b+=i; i++;
a+=i; b+=i; i++;
a+=i; b+=i; i++;
a+=i; b+=i; i++;
a+=i; b+=i; i++;
a+=i; b+=i; i++;
a+=i; b+=i; i++;
a+=i; b+=i; i++;
```

v. 少用指標：盡量使用暫存器，而不用記憶體，這樣可以減少暫存器和記憶體傳送資料的時間。如下圖所示，將(a)的寫法轉換成(b)寫法。

```
(a)
int a,b,c;
a=1;
b=1;
c=a+b;
c=a*b;
```

```
(b)
int *a,*b,*c;
*a=1;
*b=1;
*c=*a+*b;
*c=*a**b;
```

- 演算法最佳化：這一部分是必須針對 aac 的演算法做修改，故必須深入了解 aac 所用的壓縮方法，再去找尋可以做到減少運算時間的地方。