「SBC 架構」與多媒體設計

孫述平*、趙善中、邱俊龍

*義守大學數位多媒體設計學系
（高雄市大樹區學城路一段一號、07-6577711-8451、spsun@isu.edu.tw）

摘　要

「SBC 架構」是一種先進的系統定義語言或者系統塑模語言。「SBC 架構」基本上是透過六個金律：(A)架構階層圖，(B)框架圖，(C)構件操作圖，(D)構件連結圖，(E)結構行爲合一圖，(F)互動流程圖等來達成任何一個系統的定義工作。「SBC 架構」應用於多媒體設計，其最大的好處就在於不論多媒體系統的複雜度如何，「SBC 架構」都可以很輕易地將其設計架構展現出來。

一. 什麼是系統

英文中系統(System)一詞來源於古代希臘文(Systēma)，意爲部分組合而成的整體，我們在日常生活中，常常用到或聽到系統這個名詞。針對任何一個系統，一千個人可能會有一千種說帖，眾說紛紜，沒有任何共識可言。所以我們必須要給系統一個明確的定義，如此針對任何一個系統，一千個人也只會有一種解釋，眾口一致，達成共識。

以下是一個傳統的系統定義：所謂系統，指的就是某一群物質或非物質，有形或無形，實體(Concrete)或抽象(Abstract)的構件(Component)經由相互聯繫、相互作用所整合而成的全體，如圖 1-1 所示。

傳統的系統定義：

所謂系統，指的就是某一群物質或非物質，有形或無形，實體或抽象的構件經由相互聯繫、相互作用所整合而成的全體。

圖 1-1.　傳統的系統定義

綜觀傳統的系統定義，我們可以發現到其強調任何系統是一個全體。除此之外，系統定義亦強調系統的組態(Configuration)，即是說系統的全體是由構件整合而成，如圖 1-2 所示。

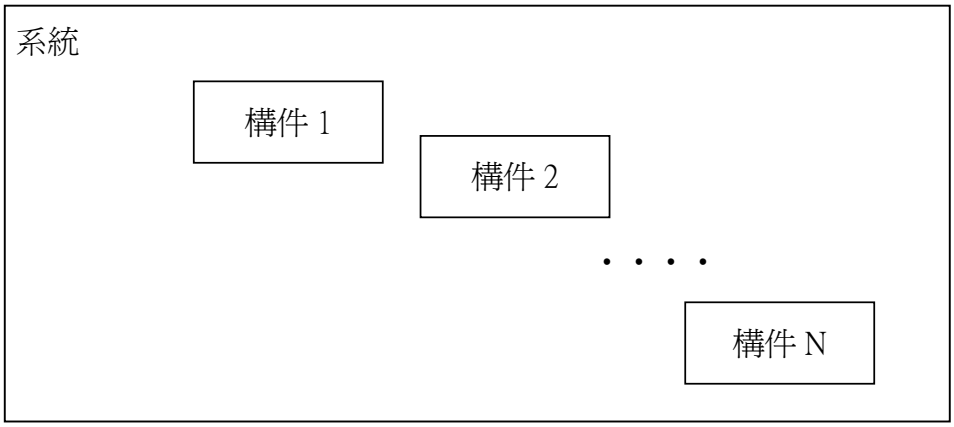

圖 1-2.　系統是由許多構件組合而成

探討傳統的系統定義後，讓我們舉些例子解釋什麼是系統。譬如說，一隻馬是一個系統，一輛汽車也是一個系統，如圖 1-3 所示。其中，馬是一個具有「野地狂奔」、「吃草」、「草地慢跑」等行爲的整體，馬的構件包括「馬頭」、「馬軀」兩項。然後，汽車是一個具有「轉彎」、「行駛公路」、「倒車入庫」等行爲的整體，汽車的構件包括「車身」、「引擎」、「輪胎」三項。

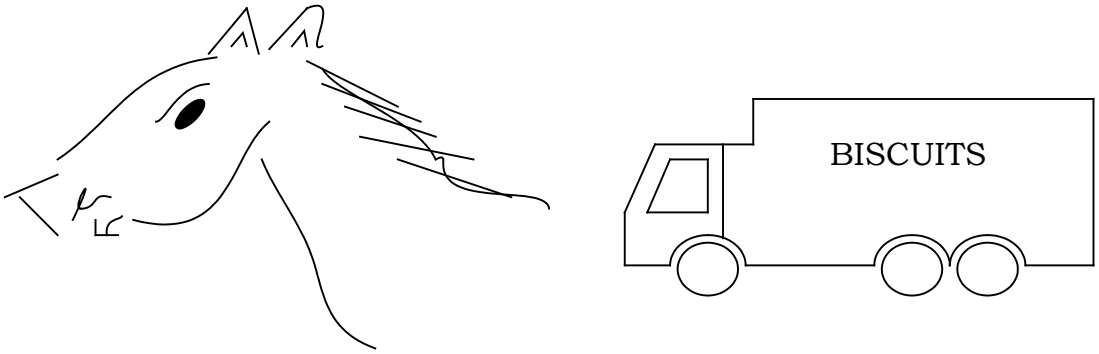

圖 1-3.　一隻馬是一個系統，一輛汽車也是一個系統

另外，一隻鉛筆，一個電燈泡，一部電腦硬碟機，或者一架販賣機是小型系統的一些典型範例。一家炸雞速食店，一座體育館，一個保全系統，或者一個物流業的倉儲管理則屬於中型的系統。而飛機場內的航空管制系統，一個大眾捷運系統，一條全國性的高速公路，或者整個網際網路(Internet)就算是規模比較大的系統了。超大型的系統可以星球來說明：整個地球在火星人來看不過是一個系統罷了。甚至於一個太陽系，或者整個銀河系也是可以把它當成一個系統來處理，如圖 1-4 所示。

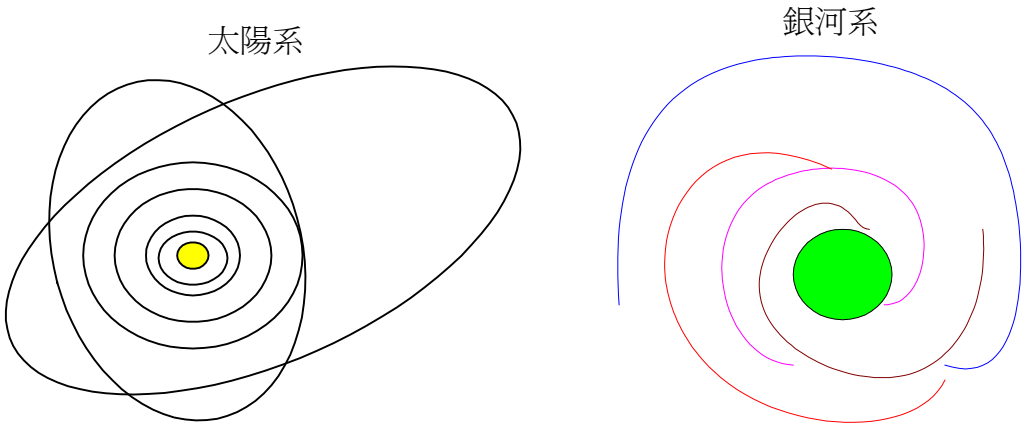

圖 1-4.　太陽系，或者銀河系也都是系統

二. 什麼是 SBC 架構

一個系統是一個整合而成的全體。若要能整合系統全體，則必須先整合系統的結構(Structure)和行爲(Behavior)。依據如此的說法，我們須要付予系統一個比較先進的定義：所謂系統，指的就是某一群物質或非物質，有形或無形，實體或抽象的構件經由相互聯繫、相互作用，並透過結構行爲合一(Structure-Behavior Coalescence，簡稱爲 SBC)所整合而成的全體，如圖 2-1 所示。

系統定義：

所謂系統，指的就是某一群物質或非物質，有形或無形，實體或抽象的構件經由相互聯繫、相互作用，並透過結構行爲合一所整合而成的全體。

圖 2-1.　系統定義

綜觀以上的系統定義，我們可以發現到系統必須要透過結構行爲合一才有辦法整合全體，如圖 2-2 所示，唯有「結構行爲合一」方能定義出系統。

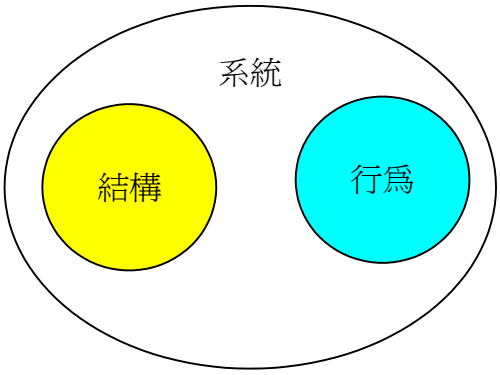

圖 2-2.　「結構行爲合一」方能定義出系統

SBC 架構[1, 2, 3, 4]就是一種先進的系統定義語言(System Definition Language)或者系統塑模語言(System Modeling Language)。SBC 架構基本上是透過六個金律：(A)架構階層圖(Architecture Hierarchy Diagram，簡稱爲 AHD)，(B)框架圖(Framework

Diagram，簡稱爲 FD)，(C)構件操作圖(Component Operation Diagram，簡稱爲 COD)，(D)構件連結圖(Component Connection Diagram，簡稱爲 CCD)，(E)結構行爲合一圖(Structure-Behavior Coalescence Diagram，簡稱爲 SBCD)，(F)互動流程圖(Interaction Flow Diagram，簡稱爲 IFD)等來達成任何一個系統的定義工作。

架構階層圖可以讓我們看出一個系統之多階級(Multi-Level)的分解與組合。如圖 2-3 顯示在「多媒體 KTV」系統的架構階層圖裡，首先「多媒體 KTV」會分解出「選歌介面」和「歌曲影片」，然後「歌曲影片」再分解出「歌曲 1」和「歌曲 2」。(架構階層圖是達成 SBC 架構的第一個金律。)

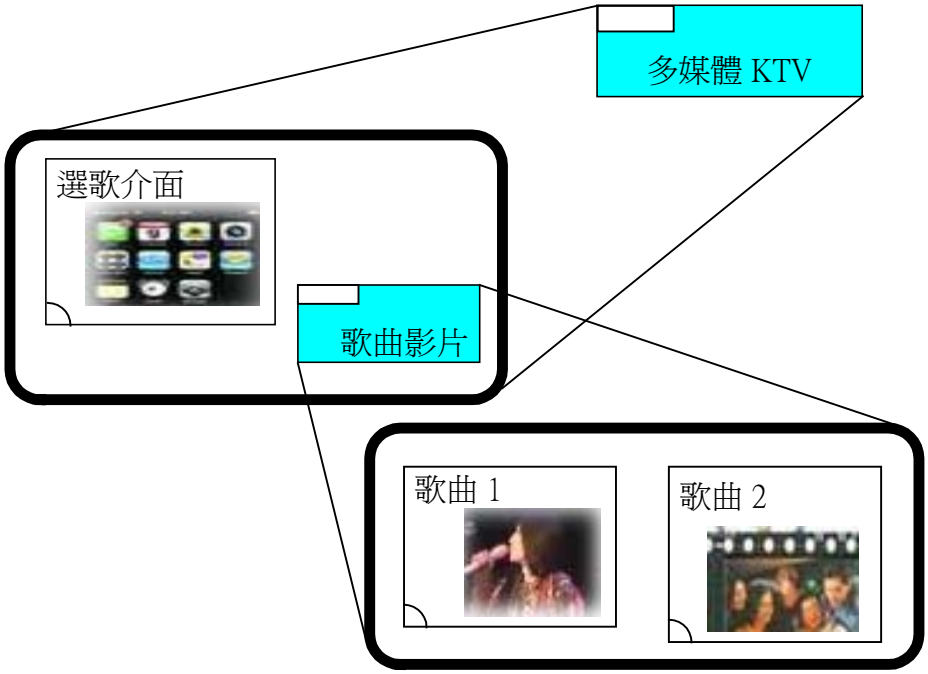

圖 2-3.　「多媒體 KTV」的架構階層圖

框架圖可以讓我們看出一個系統之多層次(Multi-Tier)或者多層級(Multi-Layer)的分解與組合。如圖 2-4 顯示在「多媒體 KTV」系統的框架圖裡，「選歌介面」、「歌曲 1」、「歌曲 2」構件是由「多媒體」分解出來的。(框架圖是達成 SBC 架構的第二個金律。)

圖 2-4. 「多媒體KTV」的框架圖

圖 2-5 利用構件操作圖來顯示「多媒體 KTV」系統所有構件的操作。其中，「選歌介面」有「選第 1 首歌」和 「選第 2 首歌」等二個操作，「歌曲 1」有「播放第 1 首歌」和 「跟唱第 1 首歌」等二個操作，「歌曲 2」「播放第 2 首歌」和 「跟唱第 2 首歌」等二個操作。(構件操作圖是達成 SBC 架構的第三個金律。)

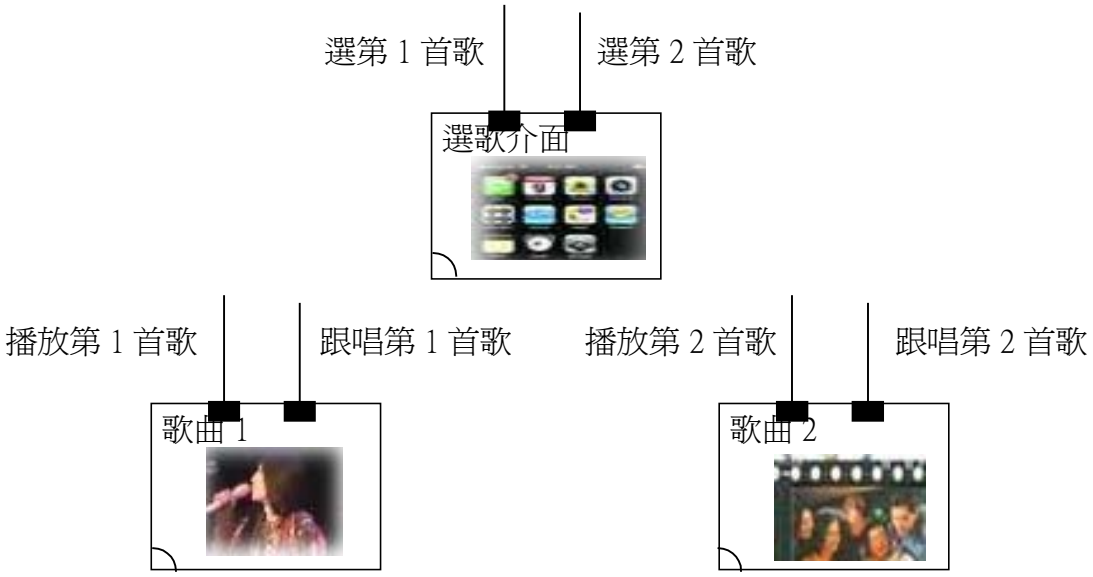

圖 2-5. 「多媒體 KTV」的構件操作圖

結構觀點除了列出構件與操作外，對於這些構件之間或者和外界環境(External Environment)的連結(Connection)也會加以詳細描述。圖 2-6 利用構件連結圖來顯示在「多媒體」系統裡，外界環境「歌唱者」和構件「選歌介面」、「歌曲 1」、「歌曲 2」彼此之間的連結。(構件連結圖是達成 SBC 架構的第四個金律。)

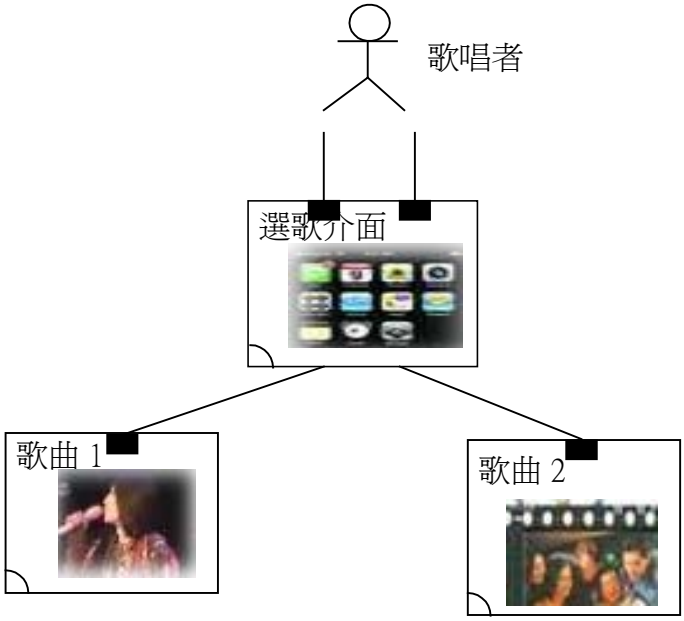

圖 2-6. 「多媒體 KTV」的構件連結圖

在圖 2-6 裡，我們可以看到外界環境「歌唱者」和構件「選歌介面」有連結，構件「選歌介面」和構件「歌曲 1」、「歌曲 2」等等有連結。有了構件連結圖以後，「多媒體 KTV」系統的樣式(Pattern)會呈現出來，因而「多媒體 KTV」系統的系統結構會變得更清晰。

在「多媒體 KTV」系統裡，若是構件之間、以及它們和外界環境有所互動，則這些互動就會產生系統行爲出來。如此結構導引出行爲的狀況下，則結構和行爲兩者合而爲一。圖 2-7 顯示「多媒體 KTV」系統的結構行爲合一圖，外界環境「歌唱者」和「選歌介面」、「歌曲 1」之間互動產生「卡拉 OK 第 1 首歌」行爲；外界環境「歌唱者」和「選歌介面」、「歌曲 2」之間互動產生「卡拉 OK 第 2 首歌」行爲。(結構行爲合一圖是達成 SBC 架構的第五個金律。)

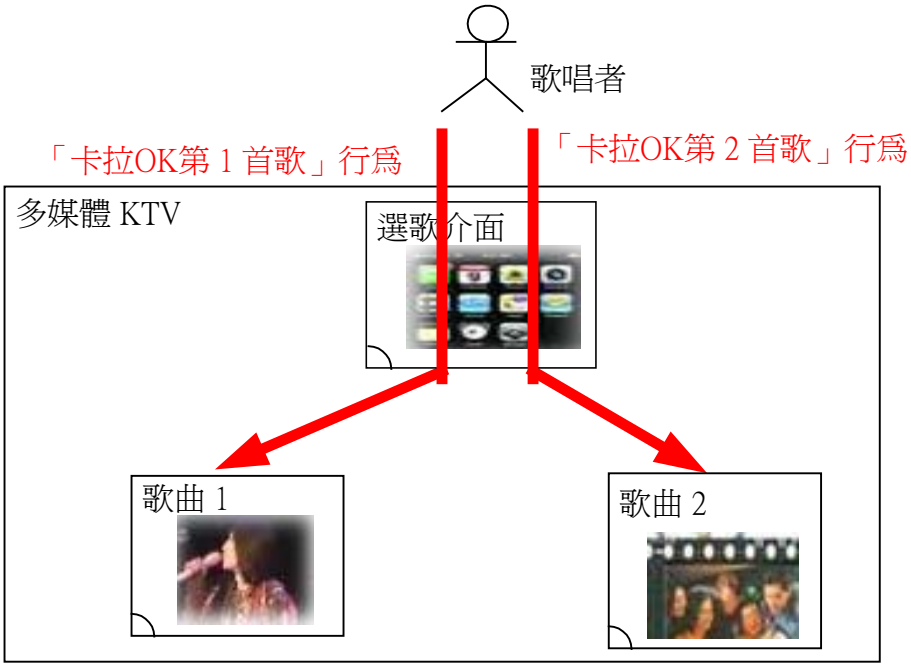

圖 2-7.　「多媒體 KTV」的結構行爲合一圖

一個系統的行爲乃是其個別的行爲總合起來。例如，「多媒體 KTV」系統的整體系統行爲包括「卡拉 OK 第 1 首歌」和「卡拉 OK 第 2 首歌」等二個個別的行爲。換句話說，「卡拉 OK 第 1 首歌」和「卡拉 OK 第 2 首歌」等二個個別的行爲總合起來就等於「多媒體 KTV」系統的整體系統行爲。

(附註：「卡拉 OK 第 1 首歌」行爲和「卡拉 OK 第 2 首歌」行爲二者彼此之間是相互獨立，沒有任何牽連的。由於它們彼此之間沒有任何瓜葛，因而這二個行爲可以同時交錯進行(Concurrently Execute)，互不干擾。)

採用 SBC 架構，最主要的目標就是只會有一個整合模型(Coalescence Model)，而不會有各自分離的結構模型和行爲模型。在圖 2-7 中，我們可以看到，「多媒體 KTV」系統的結構和行爲都一起存在其系統架構的整合模型裡面。換句話說，在「多媒體 KTV」系統的系統架構整合模型裡，我們不但看到它的系統結構，也同時看到它的系統行爲。

一個系統的整體行爲包括許多個別的行爲。每一個個別的行爲代表系統一個情境(Scenario)的執行路徑。每個執行路徑可以說就是一個互動流程圖。執行路徑可以說是將系統的內部細節互動串接起來。互動流程圖強調的是這些串接起來的互動

之先後次序。(互動流程圖是達成 SBC 架構的第六個金律。)

「多媒體 KTV」系統的互動流程圖共有二個，我們會將它們分別繪製出來。圖 2-8 說明「卡拉 OK 第 1 首歌」行爲的互動流程圖。外界環境「歌唱者」和「選歌介面」會發生「選第 1 首歌」操作呼叫的互動。再來，「選歌介面」和「歌曲 1」會發生「播放第 1 首歌」操作呼叫的互動。最後，外界環境「歌唱者」和「歌曲 1」會發生「跟唱第 1 首歌」操作呼叫的互動。

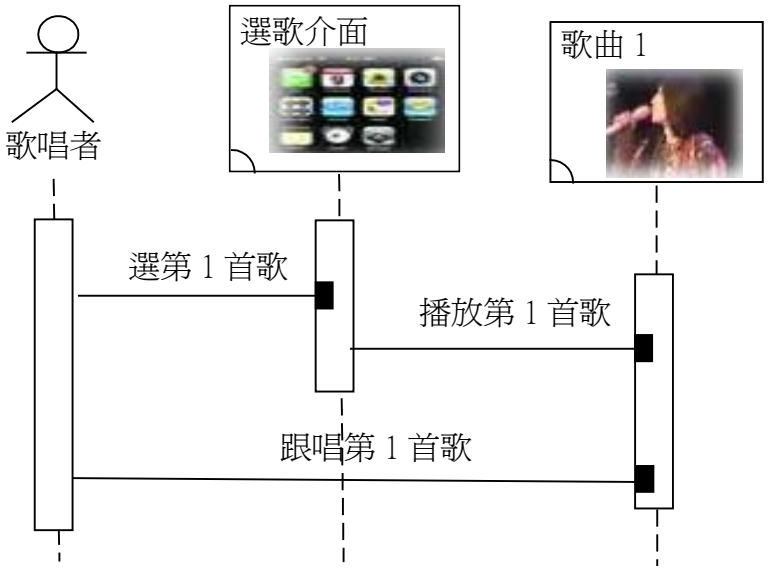

圖 2-8.　「卡拉OK第 1 首歌」行爲的互動流程圖

圖 2-9 說明「卡拉 OK 第 2 首歌」行爲的互動流程圖。外界環境「歌唱者」和「選歌介面」會發生「選第 2 首歌」操作呼叫的互動。再來，「選歌介面」和「歌曲 2」會發生「播放第 2 首歌」操作呼叫的互動。最後，外界環境「歌唱者」和「歌曲 2」會發生「跟唱第 2 首歌」操作呼叫的互動。

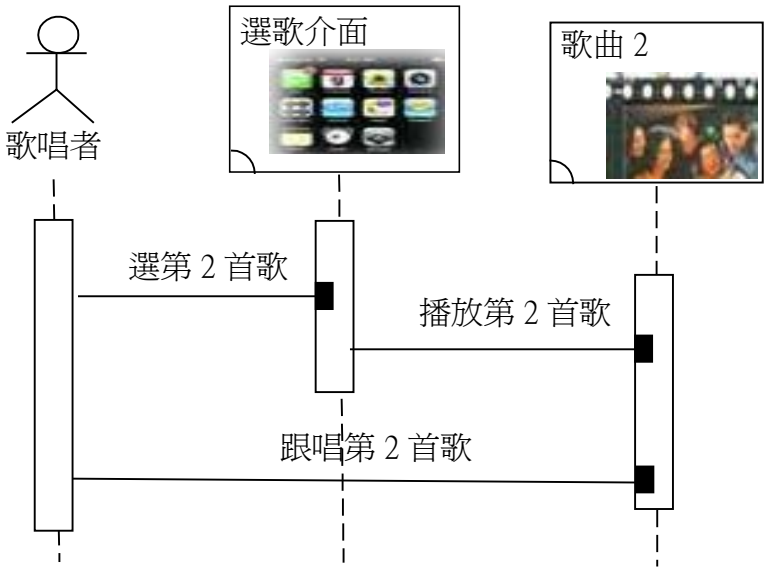

圖 2-9.　「卡拉OK第 2 首歌」行爲的互動流程圖

結構行爲合一(SBC)專注於解決設計者如何在一個可持續發展的基礎上，開發和交付創新產品所關聯的所有重大問題。結構行爲合一包括了充分利用跨越供應鏈的產品知識資產來實現產品創新的最大化，改善產品研發速度和精確性，以最大限度滿足客戶的需求。以結構行爲合一爲核心的設計模型要突出可持續發展的佈局與攻防，支持連續創新，充分利用設計領域的知識資產，開創產品設計永續經營的契機。

三. 多媒體設計

針對任何一類型的多媒體設計，我們都可以用 SBC 架構將其描繪出來。由於使用 SBC 架構所繪製出來的設計藍圖簡單易懂，所有利害關係者(Stakeholders)都可以一目瞭然，減低了不少溝通上的障礙。

例如前面所討論的「多媒體 KTV」系統的 SBC 架構，如圖 3-1 所示。所有利害關係者可以很快地得到共識，完全不會掉入雞同鴨講的困境。

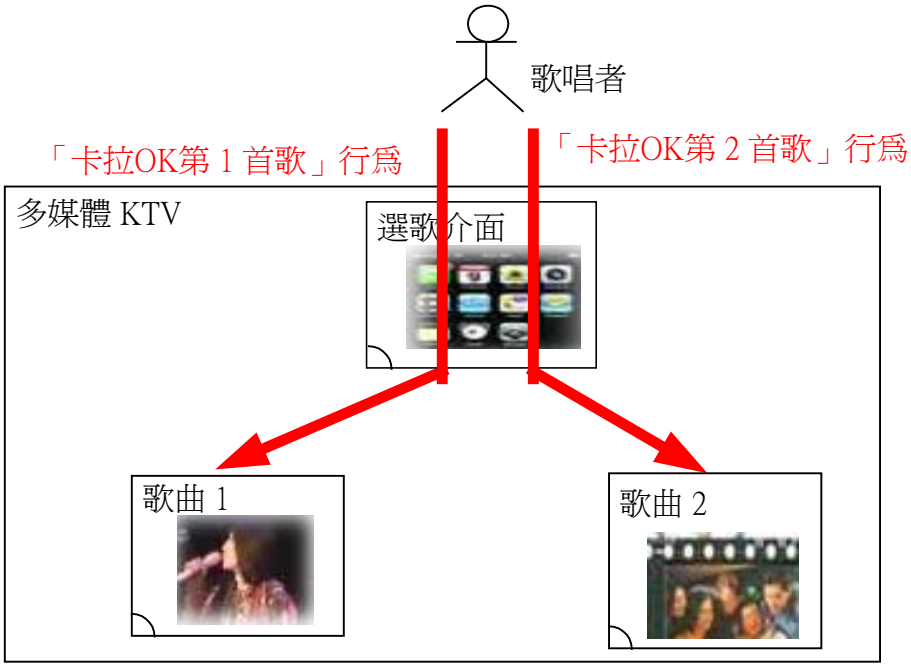

圖 3-1.　「多媒體 KTV」的結構行爲合一圖

四. 結論

本文所介紹之架構理論已廣泛應用於各類產品的設計與開發，其最大的好處就在於不論產品的複雜度如何，架構導向的產品設計皆能顯而易見的呈現於產品開發工作上。

針對多媒體設計，「SBC 架構」亦是採用六個金律：(A)架構階層圖，(B)框架圖，(C)構件操作圖，(D)構件連結圖，(E)結構行爲合一圖，(F)互動流程圖等很輕易地將其設計架構展現出來。

五. 參考文獻

1. William S. Chao， Systems Architecture: SBC Software Architecture at Work，LAP Lambert Academic Publishing, 2012。

2. William S. Chao, Software Architecture: SBC Architecture at Work, National Sun Yat-Sen University Press, 2011。

3. William S. Chao，System Analysis and Design: SBC Software Architecture in Practice，LAP Lambert Academic Publishing，2009。
4. 趙善中，《系統架構學：軟體架構，企業架構，知識架構，思考架構》，科技圖書公司，2008。