

國立交通大學

土木工程學系

碩士論文

IFC 資料標準之結構物資訊擷取與建立

Set up and retrieve structural information

based on IFC model



研究生：樊啟勇

指導教授：林昌佑 博士

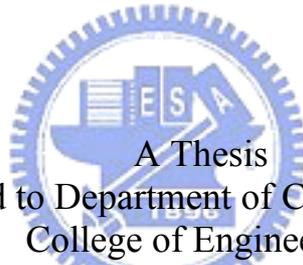
中華民國九十六年八月

IFC資料標準之結構物資訊擷取與建立
Set up and retrieve structural information based on IFC model

研究生：樊啟勇
指導教授：林昌佑 博士

Student : Chi-Yung Fan
Advisor : Dr.Chang-Yu Lin

國立交通大學
土木工程學系
碩士論文



A Thesis
Submitted to Department of Civil Engineering
College of Engineering
National Chiao Tung University
In Partial Fulfillment of the Requirements
for the Degree of
Master of Science

in
Civil Engineering
August 2007
Hsinchu, Taiwan, Republic of China

中華民國九十六年八月

IFC 資料標準之結構物資訊擷取與建立

研究生：樊啟勇

指導教授：林昌佑 博士

國立交通大學土木工程系所碩士班

摘要

為增加生產力，提高營建業整體效率，電子化與資訊標準化是必須的途徑。在世界主要工業國家，以國際組織 BuildingSmart(International Alliance for Interoperability/IAI)訂定的 IFC(Industry Foundation Classes)為主之建築資訊模式 (BIM, Building Information Model)已被廣泛採用為營建業資訊標準架構。IFC/BIM 在建築物之生命週期中，從規劃，建築設計，工程分析，估價發包，施工乃至營運維護各階段，不同分工間，皆可利用其來交換資訊乃至於知識與決策。IFC 是一種開放性資料格式作為資訊的交換以及共享使用於整個營建管理上，以純文字檔案存在，採用物件資料庫的概念來處理資料內容，避免資料傳遞時發生錯漏造成財務和時間上的損失。

為推動國內在此建築資訊標準之應用，本研究的目的在了解 IFC 資料結構內容並嘗試應用。由於市面上已有多家繪圖軟體宣稱支援 IFC 標準，本研究對於幾家市佔率較高者進行了其支援狀況的探討與比較。由於 IFC 資料能用於營造業各生命週期，特別是設計階段，故冀望結構設計時能提供所需求之資料來銜接建築與結構設計資料傳遞。因 IFC 涵蓋建築物生命週期各階段所需資料，頗為煩雜，針對結構設計所需結構元件如梁柱等之組成資料進行擷取，並建立結構分析相關部分資訊。

關鍵詞：BIM、IFC、標準化、資訊共享

Set up and retrieve structural information based on IFC model

Student : Chi-Yung Fan

Advisor : Dr.Chang-Yu Lin

Institute of Civil Engineering
National Chiao Tung University

Abstract

In order to increase productivity and improve the efficiency of building industry, electronization and standardization information are the essential way. Recently, in main industrial countries, the IFC standard has already been accepted as the information standard for buildings.

In order to promote the application of the IFC building information standards in Taiwan, this research is to comprehend IFC information and to utilize it to exchange necessary data. Through the research, compatibility of the public domain software and software from major CAD vendor will be examined. A prototype of user interface to establish and retrieve structural information of IFC data is set up in this research. Some examples will be obtained or derived to see the basic components. In addition, to reduce the complexity in the beginning, only the essential structural building element to form spaces is examined in this preliminary research.

Key Word: building information modeling, industry foundation classes, Standardization, Information sharing

誌謝

在交大兩年來求學與研究過程中，感謝恩師昌佑博士，在學業方面細心的教誨與栽培，在研究方面耐心的指引與建議，並提供良好的研究環境及設備，使論文得以順利完成，學生在此至上最深的敬意。

論文口試期間，承蒙洪士林和趙文成教授提供寶貴的意見，使得本文更臻完善，在此表達最由衷的謝意。

論文研究及撰寫期間，感謝同窗志偉、弘毅、雅晶，學長奕銘、益世，學弟秉廷、宗穎、銘浩給予的支持與鼓勵，使得我的研究生涯充滿溫馨。

在修課期間，感謝信宏、欣晏、志軒、俊成、巍羸、靜芸、振剛、耀邦這些朋友們在課業與生活上的陪伴與扶持。

最後要感謝祖父、父母和家人，於我求學期間無微不至的付出及關懷，讓我在學業上無後顧之憂，而能專心致力於課業與研究。在此將論文獻給身邊的所有人，願大家能與我共享這份喜悅。



目錄

摘要.....	i
Abstract.....	ii
誌謝.....	iii
目錄.....	iv
表目錄.....	vi
圖目錄.....	vii
第一章 緒論.....	1
1.1 研究背景與動機.....	1
1.2 研究目的.....	2
1.3 研究流程.....	2
1.4 研究內容.....	2
1.5 論文架構.....	3
第二章 文獻回顧.....	4
2.1 營建電子化.....	4
2.2 營建資訊標準化.....	5
2.2.1 BIM 建築資訊模型.....	5
2.2.2 IFC 資料標準.....	7
2.2.3 IFC 與 STEP 之異同.....	8
2.3 國內相關應用研究.....	9
2.4 支援 IFC 之軟體.....	10
第三章 研究相關技術.....	11
3.1 IFC 資料標準規則.....	11
3.1.1 IfcProject.....	13
3.1.2 IfcSite.....	14
3.1.3 IfcBuilding.....	16
3.1.4 IfcBuildingStorey.....	18
3.1.5 IfcBeam.....	19
3.1.6 IfcColumn.....	20
3.1.7 IFC 幾何形狀描述.....	21
3.2 相關程式語言.....	22
3.2.1 Fortran 程式語言.....	22
3.2.2 Visual Basic 程式語言.....	23
3.3 相關 BIM 軟體.....	24
3.3.1 AutoDesk-Revit.....	25
3.3.2 Bentley-Architecture.....	25
3.3.3 GraphiSoft-ArchiCAD.....	26
3.4 其他免費軟體.....	27
3.4.1 IfcStoreyView.....	27
3.4.2 Ifc Quick Browser.....	27
第四章 支援 IFC 格式繪圖軟體之探討.....	28
4.1 前言.....	28
4.2 Autodesk Revit.....	28
4.2.1 Revit→IFC→Revit→IFC.....	30

4.2.2 Revit→IFC→Bentley→IFC.....	30
4.2.3 Revit→IFC→ArchiCAD→IFC	31
4.3 Bentley Architecture.....	31
4.3.1 Bentley→IFC→Bentley→IFC.....	32
4.3.2 Bentley→IFC→ArchiCAD→IFC.....	32
4.3.3 Bentley→IFC→Revit→IFC.....	32
4.4 Graphisoft ArchiCAD.....	33
4.4.1 ArchiCAD→IFC→ArchiCAD→IFC.....	34
4.4.2 ArchiCAD→IFC→Revit→IFC.....	34
4.4.3 ArchiCAD→IFC→Bentley→IFC.....	34
4.5 其他 IFC 模型於三大軟體之分析	35
4.5.1 3D 鋼架模型	35
4.5.2 IFCServe 模型	36
4.6 小結	37
第五章 IFC 資料擷取與建立及 IFCUser 視窗開發	39
5.1 前言	39
5.2 IFC 檔案資訊擷取	39
5.2.1 程式建構策略	39
5.2.2 程式運作步驟	40
5.3 IFC 檔案資訊建立	40
5.3.1 程式建構策略	40
5.3.2 程式運作步驟	41
5.4 IFCUser 開發	41
5.4.1 程式建構目的	41
5.4.2 程式建構策略	41
5.4.3 程式運作步驟	42
5.5 擷取與建立操作	42
5.5.1 擷取步驟	42
5.5.2 實例示範—擷取 IFC 資料	43
5.5.3 建立步驟	44
5.5.4 實例示範—建立 IFC 資料	45
第六章 結論與建議	47
6.1 結論	47
6.2 建議	47
參考文獻	49



表目錄

表 2.1 IFC 與 STEP 之異同[3].....	51
表 2.2 支援 IFC 之軟體[11].....	51
表 4.1 繪圖軟體對不同斷面描述方法對照表.....	52
表 4.2 繪圖軟體支援 IFC 之轉換率比較.....	53



圖目錄

圖 1.1 研究流程圖	54
圖 3.1 IFC 架構之層級觀念[13]	55
圖 3.2 完整 IFC 檔必要類別樹狀圖	56
圖 3.3 I 形斷面細部示意圖[11]	57
圖 3.4 I 形斷面各點連線圖	57
圖 3.5 I 形梁以面組成示意圖	58
圖 4.1 軟體交叉探討示意圖	58
圖 4.2 Revit 建築柱斷面形狀選擇	59
圖 4.3 Revit 結構柱斷面形狀選擇	59
圖 4.4 Revit 梁斷面形狀選擇	60
圖 4.5 Autodesk Revit 柱樹狀圖	61
圖 4.6 Autodesk Revit 梁樹狀圖	62
圖 4.7 Revit 建模平面圖	63
圖 4.8 Revit 建模 3D 視圖	63
圖 4.9 Revit 建模匯出 IFC 檔以 Notepad 檢視	64
圖 4.10 Revit 建模匯出 IFC 檔以 IfcStoreyView 檢視平面圖	64
圖 4.11 Revit 建模匯出 IFC 檔以 IfcStoreyView 檢視立體圖	65
圖 4.12 Revit 建模匯出 IFC 再匯入 Revit 平面圖	65
圖 4.13 Revit 建模匯出 IFC 再匯入 Revit 3D 視圖	66
圖 4.14 IfcBuildingElementProxy 資料結構圖	67
圖 4.15 Revit_Revit_model 以 IfcStoreyView 檢視平面圖	68
圖 4.16 Revit_Revit_model 以 IfcStoreyView 檢視 3D 視圖	68
圖 4.17 Revit 建模匯出 IFC 再匯入 Bentley 四視圖	69
圖 4.18 R Revit_Bentley_model 以 IfcStoreyView 檢視平面圖	69
圖 4.19 Revit_Bentley_model 以 IfcStoreyView 檢視 3D 視圖	70
圖 4.20 Revit_Bentley_model 以 IfcStoreyView 檢視 I 斷面放大平面圖	70
圖 4.21 Revit 建模匯出 IFC 再匯入 ArchiCAD 平面圖	71
圖 4.22 Revit 建模匯出 IFC 再匯入 ArchiCAD 3D 視圖	71
圖 4.23 Revit_ArchiCAD_model 以 IfcStoreyView 檢視平面圖	72
圖 4.24 Revit_ArchiCAD_model 以 IfcStoreyView 檢視 3D 視圖	72
圖 4.25 Bentley 柱斷面形狀選擇圖	73
圖 4.26 Bentley 梁斷面形狀選擇圖	73
圖 4.27 Bentley Architecture 柱樹狀圖	74
圖 4.28 Bentley Architecture 梁樹狀圖	75
圖 4.29 Bentley 建模四視圖	76
圖 4.30 Bentley 建模匯出 IFC 檔以 Notepad 檢視	76
圖 4.31 Bentley 建模匯出 IFC 檔以 IfcStoreyView 檢視平面圖	77
圖 4.32 Bentley 建模匯出 IFC 檔以 IfcStoreyView 檢視立體圖	77
圖 4.33 Bentley 建模匯出 IFC 再匯入 ArchiCAD 平面圖	78
圖 4.34 Bentley 建模匯出 IFC 再匯入 ArchiCAD_3D 視圖	78

圖 4.35 Bentley→IFC→ArchiCAD→IFC 空心圓形梁.....	79
圖 4.36 Bentley→IFC→ArchiCAD→IFC 其他梁柱.....	79
圖 4.37 Bentley_ArchiCAD_model 以 IfcStoreyView 檢視平面圖.....	79
圖 4.38 Bentley_ArchiCAD_model 以 IfcStoreyView 檢視 3D 視圖.....	80
圖 4.39 Bentley 建模匯出 IFC 再匯入 Revit 平面圖.....	80
圖 4.40 Bentley 建模匯出 IFC 再匯入 Revit_3D 視圖.....	81
圖 4.41 Bentley_Revit_model 以 IfcStoreyView 檢視平面圖.....	81
圖 4.42 Bentley_Revit_model 以 IfcStoreyView 檢視 3D 視圖.....	82
圖 4.43 ArchiCAD 柱斷面形狀選擇圖.....	82
圖 4.44 ArchiCAD 柱之鋼材斷面形狀選擇圖.....	83
圖 4.45 ArchiCAD 梁斷面形狀選擇圖.....	83
圖 4.46 ArchiCAD 梁之鋼材斷面形狀選擇圖.....	84
圖 4.47 Graphisoft ArchiCAD 柱樹狀圖.....	85
圖 4.48 Graphisoft ArchiCAD 梁樹狀圖.....	86
圖 4.49 ArchiCAD 建模平面圖.....	87
圖 4.50 ArchiCAD 建模 3D 視圖.....	87
圖 4.51 ArchiCAD 建模匯出 IFC 檔以 Notepad 檢視.....	88
圖 4.52 ArchiCAD 建模匯出 IFC 檔以 IfcStoreyView 檢視平面圖.....	88
圖 4.53 ArchiCAD 建模匯出 IFC 檔以 IfcStoreyView 檢視立體圖.....	89
圖 4.54 ArchiCAD 建模匯出 IFC 再匯入 Revit 平面圖.....	89
圖 4.55 ArchiCAD 建模匯出 IFC 再匯入 Revit 3D 視圖.....	90
圖 4.56 ArchiCAD_Revit_model 以 IfcStoreyView 檢視平面圖.....	90
圖 4.57 ArchiCAD_Revit_model 以 IfcStoreyView 檢視 3D 視圖.....	91
圖 4.58 ArchiCAD 建模匯出 IFC 再匯入 Bentley 四視圖.....	91
圖 4.59 ArchiCAD_Bentley_model 以 IfcStoreyView 檢視平面圖.....	92
圖 4.60 ArchiCAD_Bentley_model 以 IfcStoreyView 檢視 3D 視圖.....	92
圖 4.61 Frame 平面圖.....	93
圖 4.62 Frame_3D 視圖.....	93
圖 4.63 Frame 匯入 Revit 平面圖.....	94
圖 4.64 Frame 匯入 Revit_3D 視圖.....	94
圖 4.65 Frame_Revit 以 IfcStoreyView 平面圖.....	95
圖 4.66 Frame_Revit 以 IfcStoreyView_3D 視圖.....	95
圖 4.67 Frame 匯入 Bentley 四視圖.....	96
圖 4.68 Frame_Bentley 以 IfcStoreyView 平面圖.....	96
圖 4.69 Frame_Bentley 以 IfcStoreyView_3D 視圖.....	97
圖 4.70 Frame 匯入 ArchiCAD 平面圖.....	97
圖 4.71 Frame 匯入 ArchiCAD_3D 視圖.....	98
圖 4.72 Frame_ArchiCAD 以 IfcStoreyView 平面圖.....	98
圖 4.73 Frame_ArchiCAD 以 IfcStoreyView_3D 視圖.....	99
圖 4.74 IFCServer_model 平面圖.....	99
圖 4.75 IFCServer_model_3D 視圖.....	100
圖 4.76 IFCServer_model 匯入 Revit 平面圖.....	100
圖 4.77 IFCServer_model 匯入 Revit_3D 視圖.....	101
圖 4.78 IFCServer_Revit_model 以 IfcStoreyView 平面圖.....	101

圖 4.79 IFCSrver_Revit_model 以 IfcStoreyView_3D 視圖	102
圖 4.80 IFCSrver_model 匯入 Bentley 四視圖	102
圖 4.81 IFCSrver_Bentley_model 以 IfcStoreyView 平面圖	103
圖 4.82 IFCSrver_Bentley_model 以 IfcStoreyView_3D 視圖	103
圖 4.83 IFCSrver_model 匯入 ArchiCAD 平面圖	104
圖 4.84 IFCSrver_model 匯入 ArchiCAD_3D 視圖	104
圖 4.85 IFCSrver_ArchiCAD_model 以 IfcStoreyView 平面圖	105
圖 4.86 IFCSrver_ArchiCAD_model 以 IfcStoreyView_3D 視圖	105
圖 5.1 多對一制對應圖	106
圖 5.2 IFCUser 柱、梁樹狀圖	107
圖 5.3 IFCUser 起始畫面	108
圖 5.4 擷取 IFC 檔案	108
圖 5.5 選擇載入 IFC 檔案	109
圖 5.6 載入 IFC 檔案成功	109
圖 5.7 擷取完成	110
圖 5.8 擷取 AutoDesk Revit 模型	110
圖 5.9 擷取 Bentley Architecture 模型	111
圖 5.10 擷取 Graphisoft ArchiCAD 模型	111
圖 5.11 建立	112
圖 5.12 建立對話盒	112
圖 5.13 建立成功	113
圖 5.14 選擇另存新檔	113
圖 5.15 儲存成功	114
圖 5.17 IFCUser_Frame 以 IfcStoreyView 檢視立體圖	115
圖 5.18 IFCUser_Frame_Revit 平面圖	115
圖 5.19 IFCUser_Frame_Revit 立體圖	116
圖 5.20 IFCUser_Frame_Bentley 四視圖	116
圖 5.21 IFCUser_Frame_Bentley 以 IfcStoreyView 檢視平面圖	117
圖 5.22 IFCUser_Frame_Bentley 以 IfcStoreyView 檢視立體圖	117
圖 5.23 IFCUser_model 平面圖	118
圖 5.24 IFCUser_model_3D 視圖	118
圖 5.25 IFCUser_model 匯入 Revit 平面圖	119
圖 5.26 IFCUser_model 匯入 Revit_3D 視圖	119
圖 5.27 IFCUser_Revit_model 以 IfcStoreyView 平面圖	120
圖 5.28 IFCUser_Revit_model 以 IfcStoreyView_3D 視圖	120
圖 5.29 IFCUser_model 匯入 Bentley 四視圖	121
圖 5.30 IFCUser_Bentley_model 以 IfcStoreyView 平面圖	121
圖 5.31 IFCUser_Bentley_model 以 IfcStoreyView_3D 視圖	122
圖 5.32 IFCUser_model 匯入 ArchiCAD 平面圖	122
圖 5.33 IFCUser_model 匯入 ArchiCAD_3D 視圖	123
圖 5.34 IFCUser_ArchiCAD_model 以 IfcStoreyView 平面圖	123
圖 5.35 IFCUser_ArchiCAD_model 以 IfcStoreyView_3D 視圖	124

第一章 緒論

1.1 研究背景與動機

一個營建工程專案的結構組織，具有垂直分工與水平分工的特性。就垂直分工而言，一個營建工程專案從規劃、設計、施工至使用階段，都有不同的組織成員、不同的目標。就水平分工而言，每一個體處於同一階段，需要專業化分工、追求高品質與效率。在複雜的組織需求下，資訊如何在垂直分工有效傳承，水平分工有效分享？所以資訊的儲存、交換與共享就顯得格外重要。

在傳承與分工的過程中工程圖檔扮演著極重要的溝通角色。然而每個專業人員所需要及習慣使用的軟體卻不盡相同，不同軟體間難以交流，以至於在不同階段圖檔總是需要重複建置而浪費人力與時間，且容易發生錯誤。建立資訊系統下之資料標準正可減少人為疏失。

所謂 IFC 即是一種開放性資料格式作為資訊的交換以及共享，可用於整個營建週期上，以 Express 為描述資料模型的標準語言，以純文字檔案存在，採用關聯性資料庫的概念來處理資料內容，避免資料傳遞時發生錯漏造成財務和時間上的損失，是目前先進國家之營建業所發展的必要趨勢。

國際標準組織 (ISO, International Standard Organization) 所制定的 STEP (Standard for Exchange of Product model data) 以及 IAI 組織 (International Alliance for Interoperability) 所制定的 IFC (Industry Foundation Classes) 分別為整個工業界以及營建產業制定了一套產品資料的交換標準。IFC 以物件導向技術進行 3D 圖元資訊的標準化為基礎，來制訂一個較 STEP 單純且考慮實務性的工業標準，因此，在營建產業，其雖較 STEP 起步晚，卻發展迅速，已有逐漸凌駕 STEP 之趨勢。IAI 組織的 13 個分會、22 個會員國正積極推廣 BuildingSmart 的觀念。數家著名的繪圖軟體公司 (包括 Autodesk、Bentley、Graphisoft 等) 皆已積極開發結合 IFC 模型標準之繪圖軟體，並隨著 IFC 之完善持續跟進。

然而台灣卻初入此領域，相關研究亦少之又少。有鑒於 IFC 為未來業界工程繪圖軟體須遵循之必然趨勢，相關知識的學習與研究實有其必要性。

1.2 研究目的

為推動國內在此模型資訊標準之應用，本研究的主旨在了解 IFC 資料結構內容並嘗試運用。由於 IFC 資料能用於營造業各生命週期，故冀望整體運作時能提供所需求之資料來銜接建築與結構分析資料傳遞。由於 IFC 標準定義廣泛且繁多，目前市面上販售的繪圖軟體繁多，宣稱支援 IFC 標準的也不少，但其實各家軟體的支援情形差異甚遠，沒有任何一家軟體廠商敢宣稱自己的軟體對 IFC 格式的支援是絕對完善的，且對於包裝好的應用軟體，我們無法一窺其內部究竟，我們只能做相對的驗證與了解，以作為我們使用與改善的參考與借鏡。有感其重要性，我們將針對目前市佔率較大的三家軟體公司 (AutoDesk、Bentley 及 GraphiSoft)，以他們自己宣稱支援 IFC 資訊標準最具代表性的軟體 (AutoDesk Revit、Bentley Architecture 及 GraphiSoft ArchiCAD) 來做探討。

為嘗試 IFC 之運用，本研究撰寫出一套程式，針對結構元件如梁、柱、板、Member 等之組成資料進行擷取與建立，且開發使用者介面視窗程式以利於 IFC 資料擷取與建立的操作。希望借此拋磚引玉，以幫助有心投入 IFC 研究之學者入門。

1.3 研究流程

本研究之流程，首先針對近年來營建領域電子化、資訊標準化、BIM 觀念、IFC 資訊標準、相關技術及國內外研究成果，進行文獻回顧，並擬定研究方向與預期目標；然後深入分析 IFC 資訊標準之語法規範及圖檔架構，選擇適當的軟體工具，確認 IFC 資訊標準之架構及流程，進行 IFC 資訊擷取程式的撰寫及實作。並針對在選擇工具時，因不同軟體工具對 IFC 的支援程度所產生的差異來進行探討與比較。再以這些差異做為參考與借鏡來進行 IFC 資訊建立程式的撰寫及實作。最後，開發 IFCUser 使用者介面系統結合 IFC 擷取與建立功能，以利操作與視覺化展示。詳細流程請參考圖 1.1 所示。

1.4 研究內容

本研究以 Fortran 程式語言擷取 IFC 內物件資訊並格式化輸出。由於 IFC 資料乃是以純文字檔案存在，每一筆資料皆有其描述意義與含有其他筆資料的相關性，即使沒有相關支援 IFC 之配套軟體，仍有其可讀性。但其資料結構相當複雜，且遵循一定的模式，故需開發程式幫助解讀。除了使 IFC 檔案的完整性所必須存有的資訊外，本程式所能擷取的物件只包括各種斷面形狀的梁、柱、板及 Member。

接下來，則探討與比較三大繪圖軟體公司所宣稱支援 IFC 格式之軟體，包含 AutoDesk 公司的 Revit、Bentley 公司的 Architecture 及 GraphiSoft 公司的 ArchiCAD。雖然這些軟體皆宣稱支援 IFC 格式，但 IFC 本身可描述之範圍廣大，單一繪圖軟體不可能全盤運用到，以至於各家軟體所運用的程度與範圍各有不同，因而產生出相容性、交換性等問題。

為了嘗試應用，以 Fortran 程式語言建立 IFC 模型資訊。在了解如何擷取 IFC 資訊後，仍須試著建立 IFC 資料才能達到充分運用的效果。本程式試著補足三大軟體所缺乏的斷面形狀設定，可建立的物件包含 IFC 已定義斷面形狀的梁與柱，依使用者的指令選擇建立各類型的梁或柱。

最後以 Visual Basic 程式語言建立 IFCUser 視窗程式。此視窗程式用以提供 IFC 資料擷取與建立程式的輸出輸入介面，筆者命名為 IFCUser，以列表式展示 Fortran 所擷取之物件實用資訊，及開發使用者介面視窗供使用者選擇要建立的物件類型、斷面形狀、坐落座標等 IFC 模型資訊。

1.5 論文架構

本論文共分六章。第一章為緒論，說明本研究之動機與背景、目的、流程、內容。第二章為文獻回顧，針對營建領域電子化、資訊標準化、國內外對 IFC/BIM 之推動、研究、應用等等進行整理與摘錄。第三章為研究相關技術，針對本研究應用之 IFC 資料標準規範、程式語言、應用軟體等等，進行簡介與說明。第四章針對市面上繪圖軟體應用 IFC 之現況，以三套市佔率較高之軟體為例進行交叉探討與比較。第五章針對本研究所開發之 IFCUser 軟體進行解說，包括對 IFC 檔案的擷取、建立與使用者視窗介面的開發觀念及策略等等。第六章為本研究之結論及對未來研究之建議與展望。

第二章 文獻回顧

2.1 營建電子化

所謂電子化，主要指使用者透過電腦與網路操作，管理企業內部資訊之資訊管理系統，透過軟體技術的輔助，藉以增加企業內部資訊之流通，同時減少企業處理流程的時間。所以真正的企業電子化，應涵蓋企業與企業間的電子化或稱產業電子化[1]。

近年來，由於勞力短缺及加入世界貿易組織後，接踵而來的國際競爭壓力，營建業必須提昇技術水準及生產管理效能來減少人力需求，以達到縮短工期、降低成本及提高品質之目標。為此行政院於 1989 年繼中華民國生產自動化推行計畫後，於 1991 年起實施為期十年中華民國產業自動化計畫，並將營建自動化納入自動化工作。1999 年行政院會議通過產業自動化及電子化推動方案，明訂營建業部分由內政部營建署負責，並於 2000 年起積極展開，其透過資訊整合，建立營建業上、中、下游廠商之關連體系，達成資源分享及有利作業環境，並擬定營建電子化白皮書[2]。

營建電子化導入可產生的效益，包含[1]：

1. 降低營運成本：營建電子化建立初期，系統建構及資料建立的初期成本較高，不過整體架構完成後，將大幅降低人事營運成本，例如常態性資料建檔及整理工作。
2. 加速資料傳遞：可透過電子郵件傳送工程相關資訊，節省傳統郵寄時間浪費及遺失風險，其資料散佈能力優於傳統電話傳達效率。
3. 加速資訊流通：一旦電子化上路，公司內部對工地或工地對工地的資訊傳遞會明顯加速，例如日報表傳送及工地施工視訊監控等。
4. 提升工作效率：可透過視訊系統作為公司內部或工地的協調會議，甚至業主、設計單位及施工單位三方面的溝通會議。
5. 提升形象：藉由電子化建立提供相關網頁諮詢服務，提升營建企業形象。

6. 強化文件管理：近年來，營建業大多經由 ISO 驗證制度，建立公司內部及工地的書面資料，惟傳統書面文件管理方式，除佔用空間外，資料整理及搜尋相當不易，若以電子化處理，資料可隨時透過檢索快速取得，同時亦可節省大量書面儲存空間。
7. 豐富的供應鏈資訊：電子化系統方便搜尋相關營建物料供應鏈廠商，以提供更多工程發包選擇機會，改善傳統比價及議價的缺點。
8. 強化知識系統：電子化可將公司營運及技術層面的資訊，以建檔方式，將其散佈於公司企業內部，作為教育之用，避免對工程經驗之過度依賴。

2.2 營建資訊標準化

在營建業電子化的同時，衍生出許多其他問題有待解決。就營建生命週期，營建工程專業分工介面多，且複雜，造成營建工程資訊繁雜而交換困難。若不能透過電子化與標準化整合各種工程資訊，易造成工程單位各行其事、資訊重覆建置、徒增錯誤、浪費資源、降低效率，進而影響工程品質。因此，建立一套能夠讓大家共同遵守的營建物資料模式標準，以期能順利交換是必要的。

資訊標準化的意義在於，採用一致的表達方式，進行資訊交換及共享，解決傳統作業不同格式的資料在不同系統間傳遞時產生的介面問題。圖檔資訊標準是不同圖檔間的共同語言，如同國際間採用英語進行溝通[3]。因此營建產業逐漸孕育出 BIM(Building Information Model)的觀念，將圖形與非圖形資訊整合於模型中，以期達成整個生命週期的需求。

2.2.1 BIM 建築資訊模型

經過多年來不斷的研發與改進，所有與建築產業相關之軟體正處於一個相當關鍵的時刻。專業工作者及事務所多年來不斷的使用電腦輔助進行建築設計，終於開始真正感受到將圖形與非圖形資訊整合於模型中所帶來的效益，並意識到這些資訊不只是可以應用設計施工階段，而是可以應用於建築物的整個生命週期 (Building life cycle) [4,5,6]。

這種先進的觀念常常以幾種不同的名稱出現，例如建築資訊模型(Building Information Model/BIM)、單一建築模型 (Single Building Model/SBM)、或是虛

擬建築模型 (Virtual Building Model/VBM)。這些名稱與意義都極為相近的，且縮寫容易混淆，而目前學術界與專家們所最能接受的就是以建築資訊模型(BIM)來描述這些觀念。

過去十幾年中，建築產業中的專家們與軟體廠商經過不斷的努力，逐漸的朝向 BIM 的方向前進，並從中提升了生產力。BIM 所能達成的目標依使用者對於 BIM 的需求可劃分為幾個不同的層級，由最基本的需求開始，分別為：

1. 輸入/使用/分析資料 (Enter/access/analyze information)：在建物的整個生命週期中，將會有許多不同專業的工作者在不同的階段需要輸入或變更建物資料，他們需要合適、有效率、簡單易學的工具，幫助他們得到適當且精確的資訊，使他們能夠做出正確的決定。
2. 分享資訊(Share information)：設計師需要將他們手中的資訊分享給其他人，例如後續工程或營運管理團隊。同時，這些共用的資訊必須能夠由不同的系統，或是由前一個專案匯入。
3. 同步分享資訊(Synchronize shared information)：設計師必須確認他們分享及接收的資料，都是已經由團隊中成員適當的更新或檢視，同時所有圖檔的編輯歷程都能被完整的紀錄，隨時能夠查詢誰在何時對於圖檔做了哪些編輯。
4. 為工作提供最佳的參考資料(Work in context)：設計師必須在工作時得到最佳的參考資料，使其得以做出正確的決定，提高其生產力。例如空調技師在使用『空調設計軟體』進行工作時，必須同時能夠檢視建物的結構模型及資訊，作為其輔助設計的背景資料。
5. 在安全的環境中進行完全的合作 (A secure environment for full collaboration)：所有的設計師必須確保他們的想法能夠被完整的傳達給其他工作伙伴，但此同時，其智慧財產權(Intellectual Property)也必須能被完全的保護。這兩個項目在數位整合的工作流程中是缺一不可的。

無庸置疑的，BIM 具有更多的優點，因為 BIM 能將的建築師、結構技師、營造廠、及建物之管理/營運團隊更有效的整合，進而產生下列的效益：

- 由模型中粹取更多的資訊（在基本之圖形資料上附加更多資訊）將能提高生產力及設計品質。
- 自動產生並整理文件資料將可大幅減少錯誤與遺漏。
- 在設計時能夠「跨領域/專業」的提供相關資訊（Design-in-context across disciplines），並以自動化的方式遵循專案繪圖準則，以大幅增加設計的精確度。
- 前所未有的作業方式：以最新的分析工具減少重複的設計工作。
- 設計歷程(Design History)記錄將能減少爭議與訴訟。
- 隨時提供正確的資訊，將有助於做出正確的決定、減少等待/追蹤資料(waiting/back-tracking)所浪費的時間。
- 提升使用者智慧財產權(IP)的保障將有助於資訊的自由流通。
- 由原始建築模型增加更多資訊，產生營運模型及設施圖，將能提升建物經營管理之效率。

由於 BIM 需要將大量的資料建立於模型之中，因此對於資料的管理便亦發重要。由技術的觀點而言，邁向 BIM 的成功關鍵就取決於其資料結構。一個 BIM 系統若是想要正確且穩定的運作，就必須要能處理並分享大量的、混合的、複雜的資料，這些資料還必須應付建物的整個生命週期所需，包括設計、檢閱、到最後滿足數百種不同使用的需求。

這些資料不但常常必須被不同的單位使用，並且是用許多不同的方法來執行，其複雜度不亞於銀行或是航空公司系統，這正是 BIM 最大的挑戰。而 IFC 模型資訊標準的發展正是朝 BIM 的觀念逐步發展努力當中。

2.2.2 IFC 資料標準

1994 年 8 月，12 家美國軟體公司聯合起來希望建立一套標準，目的是為了促成彼此之軟體能夠互換資訊以降低成本、提高生產力。1995 年 10 月正式成立 IAI 組織，並逐漸將他們的理想推廣到全世界，目前在全世界已擁有 13 個分會、22 個會員國。為展現出其遠大的目標，IAI 最近已改稱為 BuildingSmart。[7]

BuildingSmart(IAI)的使命：

1. 促進營建產業的效率
2. 支持產業中開放的資訊交換標準
3. 在全世界廣泛推廣 BuildingSmart 的理念
4. 發展產業中跨領域的轉譯模式(universal translator)：IFCs(Industry Foundation Classes)

由 BuildingSmart 針對營建工程提出的一套專屬於營建工程的資訊模型標準，稱為 IFC(Industry Foundation Classes)標準。BuildingSmart 所制定之 IFC 與 STEP[8]為相類似之研究成果，BuildingSmart 之組成成員為各營建產業相關公司或財團法人組織，由許多著名的繪圖軟體公司與學術界共同主導，其目的在於專為營建相關產業制定一標準資訊格式，使營建產業在其工程生命週期中建立一資訊共享之供應鏈架構，節省資訊轉換成本，提高資訊交換之效率。

BuildingSmart 所進行之標準制定雖包含整體營建產業，但目前仍以建築工程為主，而一般建築工程進行大致分為構想、設計、營造、驗收等四大階段，所以 IAI 乃以此四大階段為主軸，據以設計資訊模型，以構建出完整之建築工程架構。

IFC 是以物件導向技術作為 3D 圖元資訊的標準化基礎，配合各項軟體技術與工具之發展，來制定一個較 STEP 單純且考慮實務性的營建業標準，因此，就營建業領域而言，雖然 IFC 較 STEP 起步晚，卻發展迅速，歷經了 IFC 1.0, IFC 1.5, IFC 1.5.1, IFC 2.0, IFC 2x, IFC 2x2, IFC 2x3,IFCXML 等等版本，已有逐漸凌駕 STEP 之趨勢。

2.2.3 IFC 與 STEP 之異同

就制定的目標而言，IFC 和 STEP 的精神是一致的，亦即針對各產業的產品和作業所需的相關資訊，建立一套標準化的資料表達方式，以利於不同生命週期階段、不同軟體工具、以及不同人員之間有效率地進行資料交換與共享。基於相同的理念，ISO 制定了 STEP 標準，IAI 則制定了 IFC 標準。但是，由於兩個組織的成員特性、制定標準的起源背景有所不同，造成 IFC 和 STEP 的內容、架構和應用範圍並不盡相同。STEP 的需求起源於整合美國國防工業的產品資料，因此其制定範圍涵蓋整個工業界與製造業；IFC 則是營建相關產業的公司與機構，為了商業需求而制定，因此制定範圍針對營建產業與設施管理(AEC/FM)。此外，

由於 IAI 的成員中許多為軟體工具商，因此為了其自身的軟體需求和商業利益，IFC 的制定和更新速度都較 STEP 來得快。另一方面，由於 ISO STEP 是一套中性的產品資訊表達標準，並廣為各國政府所採用，因此在採用一項新技術之前，必須通盤考量並確認其正反面因素，故制定速度無法完全配合工業界在時間方面的需求。IFC 與 STEP 之異同整理如表 2.1。

2.3 國內相關應用研究

雖然國內在 IFC 標準及 STEP 標準的發展尚不成熟，但仍有少數研究應用在營建產業方面，其中有幾份觸及了此兩項或更多關於標準方面的技術。本研究挑出三份較近幾年的以作為參考之用：

1. 建築圖形資訊標準於營建業電子商務之應用研究[3]：

此研究之目的在於嘗試應用 IFC 資訊標準建立營建圖檔資訊之網路共享機制。使營建圖形資訊能夠透過網際網路進行解讀、存取及展示，使得在傳統作業下不同來源、不同格式的營建圖形資訊能夠透過網路機制進行交換及溝通與整合，以作為推展營建業電子商務之基礎。此研究對 IFC 及 STEP 做了些許介紹與比較，而後因 IFC 的制定及更新速度較快而選擇其為實作技術，結合物件導向、虛擬實境技術以及運用網路工具，達成其研究目標。

2. 建築物生命週期資訊共享之研究[9]：

此研究之目的在於建立建築物生命週期資訊共享機制，透過地方政府的建築管理單位，實現「建築物生命週期資訊共享」的理想目標。此研究為博士論文，研究方向著重於整體「營建資訊運籌管理」，只有在 5.2 節「建築物資訊標準化之制定」中對 IFC 及 STEP 稍做介紹，並未深入探討。

3. 建築設施在營運與維護階段資訊共享-以學校教室為例[10]：

此研究之目的在於建構建築物於營運與維護階段資訊共享的模型，以提升營運與維護階段管理之效率。此研究著重於「營運與維護」的構思與流程，只有在 2.2.1 節「資訊交換標準—圖形資訊」中對 IFC、STEP 及 FMC 稍做介紹，並未深入探討。

以上，除了第一篇論文對 IFC 標準探討較深入外，國內可供參考的資料實在不多，就算是國外的資料，能免費取得的也往往缺乏技術性參考價值。

2.4 支援 IFC 之軟體

IFC 是由多家軟體廠商所支持的 BuildingSmart 組織研發出來的標準，此組織本身並不生產軟體，僅研究、宣導與提供軟體廠商 IFC 諮詢服務，越來越多廠商爭相加入 IFC 的行列，似乎覺得沒有支援 IFC 格式是落伍的一件事。本研究已知幾家支援 IFC 的軟體公司及其軟體(表 2.2)如：Archimen Active3D、Autodesk Revit 等等。



第三章 研究相關技術

3.1 IFC 資料標準規則

IFC 標準由許多 Schema 所組成。這些 Schema 按照物件導向的觀念和原則，分為 Select、Class、Type 以及 Property Set 等四類；並且透過嚴格的參照與繼承關係，組成了一個包含四個觀念層級(Conceptual Layers)的架構(圖 3.1)：資源層(Resource Layer)、核心層(Core Layer)、介面層(Interoperability Layer)以及領域/應用層(Domain/Application Layer)[12,13]。

1. 資源層

資源層為 IFC 架構之最低層級。資源層所定義的類別(classes)，為一般性的低階觀念與物件，可以不依賴其他類別而獨立存在。

在 IFC Release2.0 中所制定的資源層綱目包含測量資料(measure)、輔助資料(utility)、幾何資料(geometry)、物件性質(property)、以及物件性質型別(property type)。在 Release 2.0 中，則針對 Release 1.5 的內容進行部份的重組，並進行部分擴充。

2. 核心層

核心層為 IFC 架構的第二層級。核心層所定義的類別可被介面層與領域/應用層的所有類別所參照(referenced)與特殊化(specialized)。核心層提供了 IFC 物件模型的基本結構並且定義了大部分的抽象觀念。

核心層本身包含了兩階抽象內容：基本核心(The Kernel)和核心延伸(Core Extensions)：

- 基本核心

基本核心提供了 IFC 物件模型所需的所有基本觀念，並且定義了物件模型的結構與組成成分。基本核心所定義的抽象觀念為高層級所必須用到的。基本核心並且包含了關於物件關係、型別定義、屬性及規則的基本觀念。

- 核心延伸

核心延伸的內容包含基本核心的延伸觀念或特殊觀念，提供營建/設施管理相關產業使用所需。每一個核心延伸為基本核心所定義類別之特殊化。

3. 介面層

介面層定義了不同的領域/應用層模型之間所共用的觀念或物件模組。介面層的綱目類別包含了共享建築元件(shared building elements)以及共享建築服務元件(shared building service elements)。

4. 領域/應用層

領域/應用層為 IFC 架構的最高層級，提供了營建和設施管理領域所需要的物件模型。目前 IFC 所定義的領域模型(Domain Models)包含建築(Architecture)、設施管理(Facility Management)、估價(Cost Estimating)、以及機電設備(HVAC)。

IFC 之架構遵循「階梯原則」(ladder principle)：每一個層級的類別可參照(reference)同一層級或較低層級的其他類別，但不能參照較高層級的類別。

階梯原則的內容如下[14]：

1. 資源類別(Resource Classes) 只能參照或使用其他資源類別。
2. 核心類別(Core Classes)能參照其他核心類別，以及無限制地參照或使用(use)資源層的資源類別。核心類別不能參照或使用介面層或領域/應用層的類別。
3. 核心層本身亦遵循階梯原則。因此，基本核心類別(Kernel Classes)能被核心延伸的類別所參照或使用，反之則否。基本核心類別不能參照核心延伸類別。
4. 介面層的類別能參照任何核心層或資源層的類別，但不能參照領域/應用層的類別。
5. 領域/應用層的類別能夠參照介面層、核心層以及資源層的任何類別。

一份完整的 IFC 檔案所必須包含的類別如圖 3.2 所示，以下將幾項對本研究較重要的類別 (IfcProject、IfcSite、IfcBuilding、IfcBuildingStorey、IfcBeam、IfcColumn)以及 IFC 幾何描述的方法分別進行簡單說明。

3.1.1 IfcProject

通常一個 IFC 檔案內只有一個 IfcProject，它代表一個建築計劃，通常會有一個 IfcRelAggregates 將其和 IfcSite 連結，以表示該 IfcSite 屬於此 IfcProject。一份完整的 IFC 檔案須參照或引用 IfcProject 內容的重要基本資訊[15]，包括：

- 單位系統
- 世界座標系統
- 空間座標維度
- 數值小數點精確度
- 真北方向與世界座標系統的關係

IfcProject 實例說明：

```
#5=IFCPROJECT('1inSikC8zBXvvUqXeXO2RS',#13,'DefaultProject','C:\Documents and Settings\',$, '*Project*','design',(#40,#117),#26);
```

範例中資料代碼為#5，以#加上一整數組成之代碼，以方便資料連結與參照。資料名稱為 IFCPROJECT，前三個字母必為 IFC，所有字母皆需大寫。括號內有 9 項屬性資料，其屬性內容依序指述如下：

1. GlobalId：在範例中為'1inSikC8zBXvvUqXeXO2RS'。以 22 個字元組成，由原匯出 IFC 軟體隨機亂數產生，以給於此 IfcProject 一個專屬的、重複機率極低名稱。
2. OwnerHistory：在範例中為#13。所參照的各項歷史紀錄之代碼，藉以參照檔案來源、建立時間、轉換媒介等等歷史紀錄。
3. 此 Project 的名稱：在範例中為'DefaultProject'。原匯出軟體所給予此 IfcProject 的一個簡短名稱，通常具有簡單的代表意義。此屬性也可以不給定 (Optional)，不給定時以\$表示。
4. 對此 Project 的描述：在範例中為'C:\Documents and Settings\'。原匯出軟體所給予的一串文字，通常具有簡單描述此 IfcProject 的意義。此屬性也可以不給定 (Optional)，不給定時以\$表示。

5. 此 Project 的類型：在範例中為\$。原匯出軟體所給予的一串文字，通常是對此 IfcProject 的類型作一個簡短的描述。此屬性也可以不給定(Optional)，不給定時以\$表示。
6. 此 Project 的用途：在範例中為'*Project*'。原匯出軟體所給予的一串文字，通常是對此 IfcProject 的用途作一個簡短的描述。此屬性也可以不給定(Optional)，不給定時以\$表示。
7. 此 Project 的工作階段：在範例中為'design'。原匯出軟體所給予的一串文字，通常是對工作階段作一個簡單的描述。此屬性也可以不給定(Optional)，不給定時以\$表示。
8. 此 Project 的相關幾何資訊代碼：在範例中為(#40,#117)。所參照的資訊連結代碼，例如：世界座標系統、真北方向的訂定等等皆由此參照而來。括號中的代碼最少一個，最多無限制，視需求增加。
9. 此 Project 的單位資訊代碼：在範例中為#26。所參照之單位集合的代碼，例如：時間單位、長度單位、面積單位、體積單位等等皆由此參照而來。

3.1.2 IfcSite

IfcSite 代表一個建地，含有一些簡單的地理資訊，包括在世界座標上的點位，經度、緯度及海拔等等。通常會有兩個 IfcRelAggregates 分別將其和 IfcProject 及 IfcBuilding 連結，以表示該 IfcSite 和 IfcProject 及 IfcBuilding 的關聯性。

IfcSite 實例說明：

```
#10=IFCSITE('25pHYjCjHEHfryUXKQ6EZI',#13,'DefaultSite',$,$,#61,$,'*Site*').ELEMENT.,(24,28,0),(54,25,0),0.,$,$);
```

範例中資料代碼為#10，以#加上一整數組成之代碼，以方便資料連結與參照。資料名稱為 IFCSITE，前三個字母必為 IFC，所有字母皆需大寫。括號內有 14 項屬性資料，其屬性內容依序指述如下：

1. GlobalId：在範例中為'25pHYjCjHEHfryUXKQ6EZI'。以 22 個字元組成，由原匯出 IFC 軟體亂數產生，以給於此 IfcSite 一個專屬的、重複機率極低名稱。

2. OwnerHistory : 在範例中為#13。所參照的各項歷史紀錄之代碼，藉以參照檔案來源、建立時間、轉換媒介等等歷史紀錄。
3. 此 Site 的名稱：在範例中為'DefaultSite'。原匯出軟體所給予此 IfcSite 的一個名稱，通常具有簡單的代表意義。此屬性也可以不給定(Optional)，不給定時以\$表示。
4. 對此 Site 的描述：在範例中為\$。原匯出軟體所給予的一串文字，通常具有簡單描述此 IfcSite 的意義。此屬性也可以不給定(Optional)，不給定時以\$表示。
5. 此 Site 的類型：在範例中為\$。原匯出軟體所給予的一串文字，通常是對此 IfcSite 的類型作一個簡單的描述。此屬性也可以不給定(Optional)，不給定時以\$表示。
6. IfcSite 相對於 IfcProject 的位置：在範例中為#61。所參照之相對位置與方向的代碼，由此可以知道這個 IfcSite 相對於所屬的 IfcProject 座標系統(世界座標系統)中，座落點與方向。
7. Representation : 在範例中為\$。原匯出軟體所給予的一串文字，通常描述此建地在工程架構內的安排。此屬性也可以不給定(Optional)，不給定時以\$表示。
8. 此 Site 的用途：在範例中為'*Site*'。原匯出軟體所給予的一串文字，通常是對此 IfcSite 的用途作一個簡短的描述。此屬性也可以不給定(Optional)，不給定時以\$表示。
9. 建地類型：在範例中為.ELEMENT.。對此建地類型的描述，有 3 種描述可以選擇，COMPLEX、ELEMENT、PARTIAL，COMPLEX 代表多塊建地組合而成的複合建地，ELEMENT 代表一般的建地類型，PARTIAL 代表一塊建地的某個區域。
10. 建地緯度：在範例中為(24,28,0)。該建地的緯度，24 是度、28 是分、0 是秒，赤道以北的緯度範圍為 0 至 90 度，赤道以南的緯度範圍為 0 至-90 度。此屬性也可以不給定(Optional)，不給定時以\$表示。

11. 建地經度：在範例中為(54,25,0)。該建地的經度，54 是度、25 是分、0 是秒，Greenwich(地名)本初子午線以西的經度範圍為 0 至 180 度，Greenwich 本初子午線以東的經度範圍為 0 至-180 度。此屬性也可以不給定(Optional)，不給定時以\$表示。
12. 建地海拔：在範例中為 0。該建地的海拔高度，此屬性也可以不給定(Optional)，不給定時以\$表示。
13. 建地編號：在範例中為\$。依各區域的編號系統編制。此屬性也可以不給定(Optional)，不給定時以\$表示。
14. 建地位址：在範例中為\$。通常是因郵政目的而給定。此屬性也可以不給定(Optional)，不給定時以\$表示。

3.1.3 IfcBuilding

IfcBuilding 代表一個建築物。通常會有兩個 IfcRelAggregates 分別將其和 IfcSite 及多個 IfcBuildingStorey 連結，以表示該 IfcBuilding 和 IfcSite 及 IfcBuildingStorey 的關聯性。



IfcBuilding 實例說明：

```
#20=IFCBUILDING('22xjcHSuT5uReWvdMftCm_',#13,'DefaultBuilding',,$,$,#33,$'
*Building*',,ELEMENT.,0.,0.,$);
```

範例中資料代碼為#20，以#加上一整數組成之代碼，以方便資料連結與參照。資料名稱為 IFCBUILDING，前三個字母必為 IFC，所有字母皆需大寫。括號內有 12 項屬性資料，其屬性內容依序指述如下：

1. GlobalId：在範例中為'22xjcHSuT5uReWvdMftCm_'。以 22 個字元組成，由原匯出 IFC 軟體亂數產生，以給於此 IfcBuilding 一個專屬的、重複機率極低名稱。
2. OwnerHistory：在範例中為#13。所參照的各項歷史紀錄之代碼，藉以參照檔案來源、建立時間、轉換媒介等等歷史紀錄。

3. 此 Building 的名稱：在範例中為'DefaultBuilding'。原匯出軟體所給予此 IfcBuilding 的一個名稱，通常具有簡單的代表意義。此屬性也可以不給定(Optional)，不給定時以\$表示。
4. 對此 Building 的描述：在範例中為\$。原匯出軟體所給予的一串文字，通常具有簡單描述此 IfcBuilding 的意義。此屬性也可以不給定(Optional)，不給定時以\$表示。
5. 此 Building 的類型：在範例中為\$。原匯出軟體所給予的一串文字，通常是對此 IfcBuilding 的類型作一個簡單的描述。此屬性也可以不給定(Optional)，不給定時以\$表示。
6. IfcBuilding 相對於 IfcSite 的位置：在範例中為#33。所參照之相對位置與方向的代碼，由此可以知道這個 IfcBuilding 相對於所屬的 IfcSite 座標系統中，座落點與方向。
7. Representation：在範例中為\$。原匯出軟體所給予的一串文字，通常描述此建築物在工程架構內的安排。此屬性也可以不給定(Optional)，不給定時以\$表示。
8. 此 Building 的用途：在範例中為'*Building*'。原匯出軟體所給予的一串文字，通常是對此 IfcBuilding 的用途作一個簡短的描述。此屬性也可以不給定(Optional)，不給定時以\$表示。
9. 建物類型：在範例中為.ELEMENT.。對此建築物類型的描述，有 3 種描述可以選擇，COMPLEX、ELEMENT、PARTIAL，COMPLEX 代表多塊建築物組合而成的複合建築物，例如兩座大樓間以某結構物連結在一起者。ELEMENT 代表一般的建築物類型，PARTIAL 代表一座建築物被垂直切割後的其中某個區塊。
10. 建物海拔：在範例中為 0.。此建築物的海拔高度，通常是由一樓地板算起。此屬性也可以不給定(Optional)，不給定時以\$表示。
11. 基底海拔：在範例中為 0.。此建築物基底的海拔高度。此屬性也可以不給定(Optional)，不給定時以\$表示。

12. 建物位址：在範例中為\$。通常是因郵政目的而給定。此屬性也可以不給定 (Optional)，不給定時以\$表示。

3.1.4 IfcBuildingStorey

IfcBuildingStorey 代表建築物中的某個樓層。通常會有一個 IfcRelAggregates 將一至多個 IfcBuildingStorey 和 IfcBuilding 連結，以表示此 IfcBuildingStorey 屬於該 IfcBuilding。

IfcBuildingStorey 實例說明：

```
#30=IFCBUILDINGSTOREY('1iTFK0l4XC8Om5I3Y_OZLe',#13,'GroundFloor','Module',$,#30,$,'3DModel',ELEMENT.,0.);
```

範例中資料代碼為#30，以#加上一整數組成之代碼，以方便資料連結與參照。資料名稱為 IFCBUILDINGSTOREY，前三個字母必為 IFC，所有字母皆需大寫。括號內有 10 項屬性資料，其屬性內容依序指述如下：

1. GlobalId：在範例中為'1iTFK0l4XC8Om5I3Y_OZLe'。以 22 個字元組成，由原匯出 IFC 軟體亂數產生，以給於此 IfcBuildingStorey 一個專屬的、重複機率極低名稱。
2. OwnerHistory：在範例中為#13。所參照的各項歷史紀錄之代碼，藉以參照檔案來源、建立時間、轉換媒介等等歷史紀錄。
3. 此 BuildingStorey 的名稱：在範例中為'GroundFloor'。原匯出軟體所給予此 IfcBuildingStorey 的一個名稱，通常具有簡單的代表意義。此屬性也可以不給定(Optional)，不給定時以\$表示。
4. 對此 BuildingStorey 的描述：在範例中為'Module'。原匯出軟體所給予的一串文字，通常具有簡單描述此 IfcBuildingStorey 的意義。此屬性也可以不給定(Optional)，不給定時以\$表示。
5. 此 BuildingStorey 的類型：在範例中為\$。原匯出軟體所給予的一串文字，通常是對此 IfcBuildingStorey 的類型作一個簡單的描述。此屬性也可以不給定(Optional)，不給定時以\$表示。

6. IfcBuildingStorey 相對於 IfcBuilding 的位置：在範例中為#30。所參照之相對位置與方向的代碼，由此可以知道這個 IfcBuildingStorey 相對於所屬的 IfcBuilding 座標系統中，座落點與方向。
7. Representation：在範例中為\$。原匯出軟體所給予的一串文字，通常描述此樓層在工程架構內的安排。此屬性也可以不給定(Optional)，不給定時以\$表示。
8. 此 BuildingStorey 的用途：在範例中為'3DModel'。原匯出軟體所給予的一串文字，通常是對此 IfcBuildingStorey 的用途作一個簡短的描述。此屬性也可以不給定(Optional)，不給定時以\$表示。
9. 樓層類型：在範例中為.ELEMENT。對此建築樓層類型的描述，有 3 種描述可以選擇，COMPLEX、ELEMENT、PARTIAL，COMPLEX 代表多個樓層組合而成的複合樓層。ELEMENT 代表一般的樓層類型。PARTIAL 代表一個樓層被水平切割後的其中某小層。
10. 樓層高度：在範例中為 0。通常指此層樓地板高度，相對於所屬建築物的一樓地板高度。此屬性也可以不給定(Optional)，不給定時以\$表示。

3.1.5 IfcBeam

IfcBeam 是一個水平或近乎水平的結構的構件，從結構的觀點來看，它承受較多的剪力與彎矩。

IfcBeam 實例說明：

```
#40=IFCBEAM('37ikST2zL9Ie933pCr0HCc',#13,'BMR-010','Model:2706','*default beam*',#814,#799,$);
```

範例中資料代碼為#40，以#加上一整數組成之代碼，以方便資料連結與參照。資料名稱為 IFCBEAM，前三個字母必為 IFC，所有字母皆需大寫。括號內有 8 項屬性資料，其屬性內容依序指述如下：

1. GlobalId：在範例中為'37ikST2zL9Ie933pCr0HCc'。以 22 個字元組成，由原匯出 IFC 軟體亂數產生，以給於此 IfcBeam 一個專屬的、重複機率極低名稱。

2. OwnerHistory：在範例中為#13。所參照的各項歷史紀錄之代碼，藉以參照檔案來源、建立時間、轉換媒介等等歷史紀錄。
3. 此 Beam 的名稱：在範例中為'BMR-010'。原匯出軟體所給予此 IfcBeam 的一個名稱，通常具有簡單的代表意義。此屬性也可以不給定(Optional)，不給定時以\$表示。
4. 對此 Beam 的描述：在範例中為'Model:2706'。原匯出軟體所給予的一串文字，通常具有簡單描述此 IfcBeam 的意義。此屬性也可以不給定(Optional)，不給定時以\$表示。
5. 此 Beam 的類型：在範例中為'*default beam*'。原匯出軟體所給予的一串文字，通常是對此 IfcBeam 的類型作一個簡單的描述。此屬性也可以不給定(Optional)，不給定時以\$表示。
6. 相對位置與方向代碼：在範例中為#814。所參照之相對位置與方向的代碼，由此可以知道這個 IfcBeam 相對於所屬的 IfcBuildingStorey 座標系統中，座落點與方向。
7. 此 Beam 的幾何描述代碼：在範例中為#799。所參照之對於此 IfcBeam 幾何描述的代碼，由此可知此 IfcBeam 的長度、斷面形狀、斷面尺寸等等資訊。
8. 此 Beam 的標籤：在範例中為\$。原匯出軟體所給予的一串文字，對此 IfcBeam 作一個標籤。此屬性也可以不給定(Optional)，不給定時以\$表示。

3.1.6 IfcColumn

IfcColumn 為一個垂直或近乎垂直的結構構件，從結構觀點來看，它承受較多的軸力。

IfcColumn 實例說明：

```
#50=IFCCOLUMN('2pfKXL$Q95XO3aVGD70cIV',#13,'BMR-012','Model:2708','*default column*',#843,#821,$);
```

範例中資料代碼為#50，以#加上一整數組成之代碼，以方便資料連結與參照。資料名稱為 IFCCOLUMN，前三個字母必為 IFC，所有字母皆需大寫。括號內有 8 項屬性資料，其屬性內容依序指述如下：

1. GlobalId : 在範例中為'2pfKXL\$Q95XO3aVGD70cIV'。以 22 個字元組成，由原匯出 IFC 軟體亂數產生，以給於此 IfcColumn 一個專屬的、重複機率極低名稱。
2. OwnerHistory : 在範例中為#13。所參照的各項歷史紀錄之代碼，藉以參照檔案來源、建立時間、轉換媒介等等歷史紀錄。
3. 此 Column 的名稱：在範例中為'BMR-012'。原匯出軟體所給予此 IfcColumn 的一個名稱，通常具有簡單的代表意義。此屬性也可以不給定(Optional)，不給定時以\$表示。
4. 對此 Column 的描述：在範例中為'Model:2708'。原匯出軟體所給予的一串文字，通常具有簡單描述此 IfcColumn 的意義。此屬性也可以不給定(Optional)，不給定時以\$表示。
5. 此 Column 的類型：在範例中為'*default column*'。原匯出軟體所給予的一串文字，通常是對此 IfcColumn 的類型作一個簡單的描述。此屬性也可以不給定(Optional)，不給定時以\$表示。
6. 相對位置與方向代碼：在範例中為#843。所參照之相對位置與方向的代碼，由此可以知道這個 IfcColumn 相對於所屬的 IfcBuildingStorey 座標系統中，座落點與方向。
7. 此 Column 的幾何描述代碼：在範例中為#821。所參照之對於此 IfcColumn 幾何描述的代碼，由此可知此 IfcColumn 的長度、斷面形狀、斷面尺寸等等資訊。
8. 此 Column 的標籤：在範例中為\$。原匯出軟體所給予的一串文字，對此 IfcColumn 作一個標籤。此屬性也可以不給定(Optional)，不給定時以\$表示。

3.1.7 IFC 幾何形狀描述

在 IFC 標準中有一些對於幾何形狀描述的類別，而同一個建築物件常常可以使用不同的類別來描述。本研究在進行梁柱資訊探討時，接觸到幾種對物件的幾何形狀描述方法，以下針對其描述的方式分類進行說明：

1. 特定斷面描述法：

針對建築常用的均勻(Uniform)斷面桿件，IFC 有定義一些獨特類別來描述其斷面，再給於桿件長度尺寸，便可描述成為立體物件。其中本研究所使用到的如：矩形使用 IfcRectangleProfileDef 描述、圓形使用 IfcCircleProfileDef 描述、空心矩形使用 IfcRectangleHollowProfileDef 描述、空心圓形使用 IfcCircleHollowProfileDef 描述、C 形使用 IfcCShapeProfileDef 描述、I 形使用 IfcIShapeProfileDef 描述、L 形使用 IfcLShapeProfileDef 描述、T 形使用 IfcTShapeProfileDef 描述、U 形使用 IfcUShapeProfileDef 描述、Z 形使用 IfcZShapeProfileDef 描述等等。這些獨特描述類別僅針對單一斷面形狀進行描述。以 I 形為例，如圖 3.3 所示，包含該斷面細部尺寸。描述方式精準易判斷，在傳遞時亦較不易出現誤差。

2. 斷面頂點連線描述法：

針對任意且均勻(Uniform)斷面形狀桿件，IFC 定義中如：IfcArbitraryClosedProfileDef 與 IfcArbitraryOpenProfileDef，此兩類別分別可以表示任意封閉形狀與任意開放形狀，描述時依固定順序將點與點做連線。以 I 形為例，如圖 3.4 所示，該斷面有 $n(12)$ 個頂點，就以 $n+1(13)$ 個頂點座標描述，起始點會再做為終點重複描述一次。但此種描述方式在不同軟體間轉換時容易因為連線順序問題而發生誤差。

3. 表平面組合描述法：

針對任意立體物件，IFC 將其表面切割為複數平面來描述，使用類別為 IfcFacetedBrep。以 I 形梁為例，如圖 3.5 所示，將此立方體以複數 IfcFace(表平面)組合而成，而每個 IfcFace 以自己所擁有的頂點座標描述該平面，與前一描述法不同之處在於，有 n 個頂點，就以 n 個頂點座標描述，起始點不重覆描述。此描述方法在本研究的實例中沒有使用到，僅供參考。

3.2 相關程式語言

3.2.1 Fortran 程式語言

Fortran 這個名詞是由 Formula Translate 這兩個字各取前面 3、4 個字母所組成的。所以，很明顯的可以看出這個語言當初的訴求，是為了應用在理工方面的

計算需求所發展出來的語言。Fortran 是由 IBM 公司的 John Backus 所率領的小組，從 1953 年開始發展，1956 年正式對外公開了一份 Fortran 的程式設計參考手冊。而在 1957 年 4 月，世界上有史以來的第一個 Fortran 編譯程式完工出廠，Fortran 到此時才算正式誕生[16]。

隨著世界不斷地進步，程式語言寫做理論屢屢創新。由 Fortran66、Fortran77、Fortran90 而至國際標準組織 ISO 在 1997 年公布 Fortran95 標準，Fortran 慢慢趨向於結構化，並加入了物件導向的觀念及工具、提供了指標、加強了陣列的功能、改良了舊的 Fortran 語法中的寫作版面格式以及加強了平行運算等等功能，使 Fortran 老而不衰，至今仍為工程界所廣為應用。

Fortran 的優缺點：

1. 現今的 Fortran 仍主要是程序導向、結構式撰寫為主，因為數值計算都是結構式演算法，雖然新版的 Fortran 也開始強調物件導向，但用 Fortran 寫物件導向是非常不便的事。
2. 由於 Fortran 早期是開發用以求解數值問題，所以 Fortran 在浮點運算方面是採截斷誤差，所以在這方面會比 C++ 與 Java 等等快很多。
3. Fortran 主要是以傳址呼叫為主，指標功能容易出錯且難以維護，而新世代的語言都比較傾向傳值呼叫。
4. 跨平台問題，Fortran 的跨平台能力滿差，如寫在微軟系統下的，放在 Linux 或 Unix，大概都會有問題。
5. 視窗程式撰寫麻煩，雖然 Fortran 也能撰寫視窗程式，但用 Fortran 寫視窗介面，是非常不便的。

3.2.2 Visual Basic 程式語言

BASIC 是在 1960 年代中期，由 Dartmouth 學院的 John Kemeny 教授與 Thomas Kurtz 教授所開發，用以撰寫簡單的程式。BASIC 的主要目標就是要讓初學者熟悉程式設計的技術。

在各種型態的電腦或硬體平台中受到廣泛使用的 BASIC，已經使得程式語言的效能大幅提昇。當比爾蓋茲建立微軟公司時，他在許多早期的個人電腦上實作

了 BASIC。由於在 1980 年代的晚期與 1990 年代早期，微軟 Windows 圖形使用者介面(GUI)的發展，BASIC 很自然的演變成微軟在 1991 年所引介的 Visual Basic，簡稱 VB。

直到 1991 年 Visual Basic 出現以前，開發以微軟為基礎的應用程式是困難而麻煩的過程。雖然 Visual Basic 是由 BASIC 程式語言發展而來，但是兩者是完全不同的程式語言，VB 提供許多強大的功能，例如圖形使用者介面、事件處理、使用 Win32 應用程式設計介面(Windows 32-bit Application Programming Interface；Win32 API)、物件導向的程式設計，以及錯誤處理。

微軟已經針對 .NET 設計出一種 Visual Basic。在 Visual Basic 早期的版本裡，並沒有物件導向功能，但是 Visual Basic .NET 提供增強的物件導向能力，包含強大的元件庫，使程式設計師以更快的方式開發應用程式。Visual Basic .NET 也使得增強的語言交互運用性可以使用：不同程式語言的軟體元件可以不同於以往地進行互動。開發人員甚至可以將舊版的軟體製作成新的 Visual Basic .NET 程式套件。而且，Visual Basic .NET 應用程式可以透過網際網路互動，使用諸如簡單物件存取協定(Simple Object Access Protocol；SOAP)與 XML 等業界標準。Visual Basic .NET 是微軟 .NET 策略很重要的一部份，使現存的 Visual Basic 開發人員可以很容易地合併到 .NET 中。.NET 與 Visual Basic .NET 更進一步的具體化將會產生新的程式設計樣式，其中從所謂的網頁服務(Web Services)的元件加以建立的應用程式可用於網際網路上。

Visual Basic .NET 執行的過程是以物件為主，當你使用滑鼠或鍵盤觸動那個物件就執行該物件所設定的動作。不像傳統 BASIC 程式設計是採程序導向設計，也就是說程式執行的流程在編寫程式時就決定了；至於 Visual Basic .NET 程式執行的流程是在執行時由操作者來決定其流程。在 Visual Basic .NET 中更支援物件導向的相關語法與運作機制(類別、繼承、多形、介面...)，讓程式設計師在處理複雜的軟體系統時，得以藉由物件導向的程式設計方式，簡化軟體設計的複雜度，增加軟體開發的效率與軟體的可維護性。

3.3 相關 BIM 軟體

3.3.1 AutoDesk-Revit

Revit 是 revise instantly(立即修改)的縮寫而來，Revit 軟體前身的研發開始於 1997 年末，是由當初將參數化 CAD 技術引進機械 CAD 市場的技術人員所創立。在不到十年的時間內，該軟體就將機械設計 CAD，從一個僅能處理 2D 圖形的產業，轉變成主要處理數位模型的產業[17]。

Revit 是專門針對建築資訊化模型(BIM)設計的，是最先引入建築社群並提供建築設計和文件管理支援的軟體。但其基礎技術、建築資訊化模型以及參數化變更引擎在經過設計和最佳化後，可以支援整個建築企業的資訊建立和管理。建築資訊化模型是一種先進的資料庫基礎結構，可滿足建築設計和製作團隊的資訊需求。Autodesk Revit 將此資訊基礎結構的功能擴大到建築專案的結構配置、構造和敷地設計工作中。

Autodesk Revit 與營建相關的軟體主要有兩種 Revit Building(Architecture)與 Revit Structure。Building 主要功能著重於建築與管線設備的規劃，它可以將設備管線呈現於設計建立完成的建築物空間中，所以當你以 3d 型態或是剖面去看就可以真實看到管線設備與建築的關係，較適合設備技師或廠商使用，如機電、空調、消防等。Building 本身的最後版本為 Revit Building 9.1，而在目前最新版本中 Building 這個名詞已被 Architecture 所取代，稱為 Revit Architecture 2008，在功能上是承接 Revit Building 9.1 的。Structure 主要功能著重於結構設計，它可以依據設定將配筋、鋼骨等物件畫在建築 3D 模型上，最重要的是依據它們原本設定的理想，這些結構物件是可以回饋至結構分析軟體做分析，然後在它們的數量統計程式中還可表列出結構材料的使用數量，然後再傳至 Excel 做預算，所以這應屬結構分析及數量計算使用，如結構技師或輔助營造廠對於欲施工建物的結構認知，目前最新版本為 Revit Structure 2008。此兩版本檔案是可以互通的。本研究所使用的是 Autodesk Revit Structure 9.1 與 Revit Architecture 2008，其宣稱支援 IFC 的版本為 IFC 2x2 及 IFC 2x3。

3.3.2 Bentley-Architecture

Bentley Architecture 是 Bentley 公司針對建築師的需求所設計，提供了最直觀的使用者介面、2D/3D 共用的工具以及高度的自動化設計輔助。在 Bentley

Architecture 中可以使用 2D、3D 或是 2D/3D 混合的工作流程，一切的編輯與修改都能在 2D/3D 間自動同步更新，大幅增加設計工作進行的速度及效率。

Bentley Architecture 提供了直覺化的使用者介面，強大的專業工具，以及符合建築專業需求的標準零件庫，在整個由設計概念到施工圖的流程中，隨時產生使用者需要的書圖報表。

Bentley Architecture 具有高度的彈性，能讓使用者輕易的將各種國際性或地區性的工業標準應用在設計工作中，或是針對使用者的需要輕易的進行客製化及擴充，以符合專業的需求。

Bentley Architecture 是根據建築資訊模型(BIM)概念而來的。以一個智慧的 3D 建築模型作為單一的資料來源以供應所有型式的輸出與輸入。所有的圖元及修改都是加諸於這個單一建築物模型，而工程圖面(包括建築之平面圖、剖面圖和高程)及文件(包括品質控管，估價報表及特性)都可以自此一單一建築物模型中自動取得及更新的。本研究所使用的是 Bentley Architecture V8 XM，其宣稱支援 IFC 的版本為 IFC 2x、IFC 2x2 及 IFC 2x3。

3.3.3 GraphiSoft-ArchiCAD

ArchiCAD 是 Graphisoft 公司的旗艦產品，其基於全三維的模型設計，擁有強大的剖/立面、設計圖檔、參數計算等自動生成功能，以及便捷的方案演示和圖形渲染，為建築師提供了一個無與倫比的「所見即所得」的圖形設計工具。ArchiCAD 內建的 PlotMaker 圖檔編輯軟體使出圖過程與圖檔管理的自動化水準大大提高，而智能化的工具也保證了每個細微的修改在整個圖冊中相關圖檔的自動更新，大大節省了傳統設計軟體大量的繪圖與圖紙編輯時間，使建築師能夠有更多的時間和精力專注於設計本身，創造出更多感動人心的設計精品。

同時對於各設計企業來說，ArchiCAD 不僅僅意味著設計生產力的提升，還能夠幫助企業更為高效科學地管理與檢索設計檔案，完善企業的設計標準，提高知識產品的使用價值。

ArchiCAD 完善的團隊協作功能為大型項目的多組織、多成員協同設計提供了高效的工具，團隊領導者可以根據不同區域、不同功能、不同建築元素等屬性將設計任務分解，而團隊成員可以依據權限在一個共同的可視化項目環境裡準確

無誤的完成協同工作；同時 ArchiCAD 創建的三维模型，通過 IFC 標準資訊平台的訊息交換，可以為建築設計、結構分析等提供強大的基礎模型，為多方專業協同設計提供了有效的保障。本研究所使用的是 ArchiCAD 10，其宣稱支援 IFC 的版本為 IFC 2x2 與 IFCXML 2x2。

3.4 其他免費軟體

在 BuildingSmart 官方網站內有提供一些免費軟體，以供使用者驗證或檢視 IFC 資訊。本研究使用 IfcStoreyView 來驗證圖形資訊，使用 Ifc Quick Browser 來檢視與搜尋文字資訊。以下對此兩軟體做簡單的介紹：

3.4.1 IfcStoreyView

IfcStoreyView 是由德國 Forschungszentrum Karlsruhe 公司所開發出來的一套免費軟體[18]，IfcStoreyView 能夠將 IFC 檔案內的物件以 2D 和在 3D 的圖形模式展示出來，採用 OpenGL 技術美化視覺效果。建築內的物件類別以樹狀方式排列，可讓人一目了然其中各物件與整體建築的從屬關係，並可查詢各物件基本資訊如樓層、座標、材料、體積、表面積等特性。

3.4.2 Ifc Quick Browser

Ifc Quick Browser 是由 G.E.M.(Gamal Kira, Eberhard Michaelis and Michael Kloß) Team Solutions 所開發出來的一套免費軟體[19]。此軟體除了將每筆資料以純文字方式列出之外，點選單筆資料可以直接列出其被參照與所參照的各筆資料，且以樹狀方式排列之，讓使用者在查詢時省去了大部分搜尋時間。

第四章 支援 IFC 格式繪圖軟體之探討

4.1 前言

IFC 本身乃是一種資訊標準，在了解它後，如何應用才是最重要的課題。目前市面上販售的繪圖軟體繁多，宣稱支援 IFC 標準的也不少，但其實各家軟體的支援情形差異甚遠。由於 IFC 標準定義廣泛且繁多，沒有任何一家軟體廠商敢宣稱自己的軟體對 IFC 格式的支援是絕對標準的。且對於包裝好的應用軟體，我們無法一窺其內部究竟，我們只能做相對的驗證與了解，以作為我們使用與改善的參考與借鏡。有感其重要性，故在本章中針對目前市佔率較大的三家軟體公司 (AutoDesk、Bentley 及 GraphiSoft)，以他們自己宣稱支援 IFC 資訊標準具代表性的軟體 AutoDesk Revit、Bentley Architecture 及 GraphiSoft ArchiCAD 來做探討。本研究在此後依序以 Revit、Bentley 及 ArchiCAD 稱之。

在 4.2 至 4.4 節中除了要針對三家軟體本身的支援程度與範圍探討之外，還會進行交叉匯出匯入探討(圖 4.1)，匯出 IFC 再匯入原軟體時可以了解該軟體本身的匯出匯入機制是否健全。匯入其他軟體，檢視資料是否有變化或遺失，探討兩兩軟體相互間轉換率、支援程度、影響因素等等。由於 IFC 尚無法定義節點性質，且為了方便檢視各物件在轉換前後的變化，所以本研究在以此三軟體建模時會將各物件分離呈現，為了討論的完整性，本研究儘可能納入該軟體所有可建立的梁柱斷面，但不包括重複的與 IFC 尚未定義的。

4.5 節會再以其他方式所建立的 IFC 檔案匯入三大軟體來做探討，包含網路上搜尋到他人所建好的 IFC 立體 3D 鋼架模型、筆者自己撰寫的 IFCUser(請參閱第五章)所建立的梁柱模型等等，將前三節所得到的推論做更進一步驗證與反驗證。

4.6 節針對以上推論做一個小結，將各軟體對不同斷面的描述方式整理為表 4.1，將各軟體間轉換率整理為表 4.2，以便於參考。

4.2 Autodesk Revit

本研究所使用的是 Autodesk Revit Building 9.1 與 Revit Architecture 2008，由於 Revit Architecture 2008 是在此部份研究完成後才正式推出，且經詳細比對後發現其對 IFC 格式的支援程度與 Autodesk Revit Building 9.1 並無明顯差異，故在圖片選用方面仍以 Autodesk Revit Building 9.1 為主。

軟體中柱分為兩類：建築柱、結構柱，兩者差別在於建築柱沒有材料屬性，結構柱則有，結構柱的材料可又分為混凝土、鋼材、木材。軟體中梁只有一類，性質類似結構柱含有材料屬性。若要建立模型，建築柱初始可選擇的斷面形狀只有矩形 1 種，若要增加種類須由資料庫載入(圖 4.2)，結構柱初始可選擇的斷面形狀只有 I 形 1 種，若要增加種類亦須由資料庫載入(圖 4.3)，梁初始可選擇的斷面形狀只有 I 形 1 種，若要增加種類同樣須由資料庫載入(圖 4.4)。本研究為了討論的充分性，在接下來建模時會載入資料庫中所有斷面形狀，但摒除重複的或 IFC 未定義的類型。

Revit 建模匯出 IFC 格式後柱與梁的資料分別以圖 4.5 與圖 4.6 的樹狀圖表示，其中實線代表必經路徑，虛線代表選擇路徑(單選)。可以發現矩形柱、梁斷面是以 IfcRectangleProfileDef 表示，圓形柱是以 IfcCircleProfileDef 表示；U 形、I 形柱及 U 形、I 形、L 形、T 形梁斷面是以 IfcArbitraryClosedProfileDef 表示；空心矩形與空心圓形柱、梁斷面是以 IfcArbitraryProfileDefWithVoids 表示。

本研究在 Revit 軟體中以其所有可選擇的梁柱斷面種類建出一個簡單的模型，平面圖與立體圖分別示於圖 4.7 與圖 4.8。其中建築柱由左至右包含矩形、圓形共 2 支；結構柱由左至右包含矩形、圓形、U 形、I 形、空心矩形、空心圓形共 6 支；梁由下至上包含矩形、U 形、I 形、L 形、T 形、空心矩形、空心圓形共 7 支。在平面視圖建梁時，不論建在何樓層，只會顯示其標示尺寸，而本身卻無法呈現，只有將顯示模式更改為 Site 才能正常顯示(圖 4.7)，而柱則無影響。將建好的模型匯出成為 IFC 檔案，因為是純文字檔，可以由 Notepad 開啟(圖 4.9)。再以 IfcStoreyView 檢視，平面圖(圖 4.10)與立體圖(圖 4.11)與原先資料一致，可以確定該軟體匯出的 IFC 資訊是符合 IFC 標準的。

4.2.1 Revit→IFC→Revit→IFC

將 Revit 匯出的 IFC 檔案再匯入 Revit 時，可以發現，平面圖中柱完整顯示，梁卻無法呈現(圖 4.12)。將圖 4.7 與圖 4.9 比較後發現 Site 模式在檔案轉換過程中消失了，這是造成梁無法呈現的主因；而立體圖中柱、梁皆與原建模圖無明顯差異(圖 4.13)。

這時候再匯出成 IFC 檔來與第一次匯出的 IFC 檔案做比對，發現第一份 IFC 檔案大小 111KB 到了第二份變成 109KB。仔細查看檔案中的資料發現 8 支 IfcColumn 資訊結構上無差異，而 7 支 IfcBeam 卻被 IfcBuildingElementProxy 類別取代了，IfcBuildingElementProxy 在 IFC 格式的設定中是代表尚未被 IFC 特別定義的建築物件，屬於一種過渡時期的定義方式，這代表 Revit 無法正確辨識出自己所匯出的梁物件，而把它們視為不明物體。我們以 Ifc Quick Browser 來看其中一筆 IfcBuildingElementProxy 類別與其連結資料(圖 4.14)。

以 IfcStoreyView 來檢視第二份 IFC 檔案，平面圖(圖 4.15)與立體圖(圖 4.16)所顯示出來的圖形與前述檔案資訊相符合，梁的著色依 IfcStoreyView 對 IfcBuildingElementProxy 的描述而呈現紅色，而柱的著色依 IfcStoreyView 對 IfcColumn 的描述而呈現綠色。

4.2.2 Revit→IFC→Bentley→IFC

將 Revit 第一次匯出的 IFC 檔匯入 Bentley，所有梁柱依然有所顯示，但是卻只是以圖元形式存在(如圖 4.17)，無法修改亦無法刪除。

再以 Bentley 匯出成 IFC 檔來與原 Revit 匯出的 IFC 檔案做比對，發現第一份 IFC 檔案大小 111KB 到了第二份變成 44KB。仔細查看檔案中的資料發現 8 支 IfcColumn 及 7 支 IfcBeam 中空心矩形的那兩支卻整個消失了，其他只是部分描述及連結改變。中空矩形梁的完全消失代表 Bentley 無法將 Revit 對中空矩形梁的描述方式再轉換出來。

以 IfcStoreyView 來觀看第二份 IFC 檔案，平面圖與立體圖分別示於圖 4.18 與圖 4.19。平面檢視圖中空心矩形柱與梁消失，與前述檔案資訊相符合。而立體檢視圖中除了空心矩形梁柱外，還缺少了 U 形、I 形柱與 U 形、I 型、L 型、T 形梁，這是因為 IfcArbitraryClosedProfileDef 在前面的轉換過程中部分點座標的轉

換不夠精準或連線錯誤，造成 IfcStoreyView 無法將其描繪成為封閉體積，以 I 斷面檢視圖放大為例(圖 4.20)，很明顯由連線錯誤造成。

4.2.3 Revit→IFC→ArchiCAD→IFC

將 Revit 第一次匯出的 IFC 檔匯入 ArchiCAD，就視覺上來說，物件的呈現與原建模圖無明顯差異，如圖 4.21 與圖 4.22。

再以 ArchiCAD 匯出成 IFC 檔來與原 Revit 匯出的 IFC 檔案做比對，發現第一份 IFC 檔案大小 111KB 到了第二份變成 157KB。查看檔案中梁、柱的資訊依然存在，只是描述的方式變為 ArchiCAD 本身的設定(請參考 4.4 節)，而梁、柱以外的資料大致上繼承了原 Revit 匯出的 IFC 檔案。

將 ArchiCAD 做第二次匯出的 IFC 檔以 IfcStoreyView 觀看，平面圖(圖 4.23)如前述檔案資訊相符，立體圖(圖 4.24)中唯獨空心矩形梁沒有顯示，乃是因為在轉換的過程中 IfcArbitraryClosedProfileDef 內部分點座標的轉換不夠精準或連線錯誤，造成 IfcStoreyView 無法將其描繪成封閉體積。

4.3 Bentley Architecture

本研究所使用的是 Bentley Architecture V8 XM 試用版，原程式中可選擇的柱的斷面形狀有九種(圖 4.25)：矩形、圓形、C 型、I 形、L 型、T 型、U 型、空心矩形、空心圓形，梁的斷面形狀也是九種(圖 4.26)，種類與柱相同。

Bentley 柱與梁的資料分別以圖 4.27 與圖 4.28 的樹狀圖表示，其中實線代表必經路徑，虛線代表選擇路徑(單選)。由此可以發現柱與梁皆是以 IfcBeam 來表示，只是用方向去控制它罷了，這是錯誤的描述方式，柱本身應以 IfcColumn 描述。在斷面形狀方面，可以發現圓形柱、梁斷面是以 IfcCircleProfileDef 表示；矩形、C 型、I 形、L 型、T 型、U 型柱、梁斷面是以 IfcArbitraryClosedProfileDef 表示；空心矩形與空心圓形柱、梁斷面是以 IfcArbitraryProfileDefWithVoids 表示。

本研究在 Bentley 軟體中以其九種柱與九種梁建出一個簡單的模型如圖 4.29，其中柱由左至右包含矩形、圓形、I 形、L 形、U 形、T 形、空心矩形、空心圓形、C 形共 9 支；梁由下至上順序同柱也是 9 支。將建好的模型匯出成為 IFC 檔案，因為是純文字檔，可以由 Notepad 開啟(圖 4.30)，再以 IfcStoreyView 檢視，

可以發現空心圓形在平面圖(圖 4.31)中顯示為矩形，在立體圖(圖 4.32)中無法顯示，這表示 Bentley 本身在匯出空心圓形斷面時即不符 IFC 標準，而其他物件資訊是符合標準的。

4.3.1 Bentley→IFC→Bentley→IFC

將 Bentley 匯出的 IFC 檔案後再匯入 Bentley，可以發現第一份 IFC 檔案大小 36.8KB 到了第二份變成 36.7KB 這個差距在 Bentley 軟體中完全看不出來，詳細比對兩份 IFC 檔後發現除了檔案順序稍微改變外，某些內容相同的 IfcAxis2Placement3D 共用了，使得檔案大小精簡了 0.1KB，如此算是自我匯入匯出完全轉換。

4.3.2 Bentley→IFC→ArchiCAD→IFC

將第一次匯出的 IFC 檔匯入 ArchiCAD，發現在平面圖(圖 4.33)中空心圓型柱消失，空心圓型梁變為一條線，由於斷面積為零體積為零，在立體圖(圖 4.34)中兩者皆無法顯示，其餘斷面形狀依然完好存在。

再以 ArchiCAD 匯出成 IFC 檔來與原 Bentley 匯出的 IFC 檔案做比對，發現第一份 IFC 檔案大小 36.8KB 到了第二份變成 35.3KB。查看檔案發現原本的 18 個 IfcBeam(柱+梁)消失了 1 個，變成 17 個 IfcBeam，消失的就是空心圓型柱，而 17 個 IfcBeam 其中空心圓型梁的資料以 Ifc Quick Browser 檢視如圖 4.35，另外 16 個斷面的資料其中一組如圖 4.36。空心圓形梁的 IfcShapeRepresentation 中的描述較其他斷面形狀缺少 BoundingBox 描述立體構成要素的部份，這是其之所以變成一條線的原因。

將 ArchiCAD 做第二次匯出的 IFC 檔以 IfcStoreyView 開啟，平面圖(圖 4.37)與立體圖(圖 4.38)所顯示出來的圖形與前述檔案資訊相符合。

4.3.3 Bentley→IFC→Revit→IFC

將第一次匯出的 IFC 檔匯入 Revit，就視覺上而言，平面圖(圖 4.39)與立體圖(圖 4.40)中所呈現的物件與原建模圖無明顯差異。但是以滑鼠點選發現所有物件的名稱已不再是柱或梁，而變成了「BuildingElement」，這表示 Revit 可能已經無法辨認物件類型，而以一般建築物件來命名。

再以 Revit 匯出成 IFC 檔來與原 Bentley 匯出的 IFC 檔案做比對，發現第一份 IFC 檔案大小 36.8KB 到了第二份變成 115.0KB。查看檔案中梁、柱的資訊，果然發現已看不到 IfcBeam 類別的存在，所有物件變成了以 IfcBuildingElementProxy 來表示，IfcBuildingElementProxy 在 IFC 格式的設定中是代表尚未被 IFC 特別定義的建築物件，屬於一種過渡時期的定義方式，而這種方式不如 IfcBeam 定義來的直接與簡潔，以致於檔案放大三倍之多。

將以 Revit 第二次匯出的 IFC 檔以 IfcStoreyView 檢視，平面圖(圖 4.41)與立體圖(圖 4.42)中所有物件的著色依 IfcStoreyView 對 IfcBuildingElementProxy 的描述而呈現紅色，與原本 IfcStoreyView 對 IfcColumn 或 IfcBeam 的描述應呈現的綠色顯著不同。

4.4 Graphisoft ArchiCAD

本研究所使用的是 ArchiCAD 十版(ArchiCAD 10)試用版，原程式中可選擇的柱的斷面形狀需與材料同時選擇，共分為三類 11 種(圖 4.43)：矩形混凝土、圓形混凝土、鋼材，其中鋼材類分為 9 種(圖 4.44)：C 形、I 形、L 形、T 形、U 形、Z 形、空心矩形，以及 2 種重複或 IFC 未定義的封閉形狀。梁的斷面形狀亦需與材料同時選擇，共分為兩類 13 種(圖 4.45)：矩形混凝土、鋼材，其中鋼材類分為 12 種(圖 4.46)：C 形、I 形、L 形、T 形、U 形、Z 形、空心矩形，以及 5 種重複或 IFC 未定義的封閉形狀。IFC 未特別定義的封閉斷面形狀在 IFC 檔案內皆以 IfcArbitraryClosedProfileDef 表示，故在此不額外探討。

ArchiCAD 柱與梁的資料分別以圖 4.47 與圖 4.48 的樹狀圖表示，其中實線代表必經路徑，虛線代表選擇路徑(單選)：我們可以發現矩形混凝土柱、梁斷面以 IfcRectangleProfileDef 表示，圓形混凝土柱斷面以 IfcCircleProfileDef 表示；而鋼材柱、梁斷面全都以 IfcArbitraryClosedProfileDef 表示。

本研究在 ArchiCAD 軟體中以其在 IFC 有定義的 2 種混凝土柱和 7 種鋼柱，以及 1 種混凝土梁和 7 種鋼梁建出一個簡單的模型，平面圖與立體圖分別示於圖 4.49 與圖 4.50，其中柱由左至右包含矩形、圓形、C 形、U 形、L 形、I 形、空心矩形、T 形、Z 形共 9 支；梁由下至上包含矩形、C 形、U 形、L 形、I 形、空心

矩形、T 形、Z 形共 8 支。將建好的模型匯出成為 IFC 檔案，因為是純文字檔，可以由 Notepad 開啟(圖 4.51)，再以 IfcStoreyView 檢視，平面圖(圖 4.52)與立體圖(圖 4.53)與原先資料一致，可以確定該軟體匯出的 IFC 資訊是符合 IFC 標準的。

4.4.1 ArchiCAD→IFC→ArchiCAD→IFC

將 ArchiCAD 匯出的 IFC 檔案後再匯入 ArchiCAD，我們可以發現第一份 IFC 檔案大小 41.5KB 到了第二份變成 41.4KB 這個差距在 ArchiCAD 軟體中檢視完全沒有差異，詳細比對兩份 IFC 檔後發現除了檔案順序稍微改變外，某些內容相同的 IfcAxis2Placement3D 共用了，使得檔案大小精簡了 0.1KB，如此算是自我匯入匯出完全轉換。

4.4.2 ArchiCAD→IFC→Revit→IFC

將 ArchiCAD 第一次匯出的 IFC 檔匯入 Revit，可以發現平面視圖(圖 4.54)下，9 支柱與 8 支梁中只有空心矩形斷面的無法顯示；立體圖(圖 4.55)中依然是空心矩形斷面無法顯示。這可能是 Revit 無法辨識 ArchiCAD 匯出之 IFC 檔案對空心矩形斷面的表達方式。

再以 Revit 匯出成 IFC 檔來與原 ArchiCAD 匯出的 IFC 檔案做比對，發現第一份 IFC 檔案大小 41.5KB 到了第二份變成 68.6KB。查看檔案中梁、柱的資訊，Ifccolumn 由原本的 9 個變成 8 個，消失的正是空心矩形柱，而原本的 8 個 IfcBeam 的其中 7 個變成了以 IfcBuildingElementProxy 來表示，消失的那個正是空心矩形梁。空心矩形梁、柱皆消失的原因是因為 ArchiCAD 與 Revit 對於空心矩形斷面的描述差異太大，以致完全無法轉換。

將 Revit 做第二次匯出的 IFC 檔以 IfcStoreyView 檢視，平面圖(圖 4.56)與立體圖(圖 4.57)中所有的梁著色依 IfcStoreyView 對 IfcBuildingElementProxy 的描述而呈現紅色，所有的柱著色依 IfcStoreyView 對 IfcColumn 的描述應呈現的綠色。

4.4.3 ArchiCAD→IFC→Bentley→IFC

將 ArchiCAD 第一次匯出的 IFC 檔匯入 Bentley，四視圖(圖 4.58)中所呈現的物件與原建模圖無明顯差異。

再以 Bentley 匯出成 IFC 檔來與原 ArchiCAD 匯出的 IFC 檔案做比對，發現第一份 IFC 檔案大小 41.5KB 到了第二份變成 38.1KB。詳細查看檔案中的資訊，發現梁柱的描述方式大致上還是依照 ArchiCAD 原本使用的類別描述。

我們再將 Bentley 做第二次匯出的 IFC 檔以 IfcStoreyView 開啟，平面圖(圖 4.59)與立體圖(圖 4.60)中所顯示出來的圖形與原 ArchiCAD 匯出之 IFC 資訊相符合。所有的柱與梁著色依 IfcStoreyView 對 IfcColumn 與 IfcBeam 的描述呈現綠色。

4.5 其他 IFC 模型於三大軟體之分析

4.5.1 3D 鋼架模型

本研究下載多個由 BuildingSmart 官方網站所提供之範例 IFC 檔案，其中一份檔案為維吉尼亞理工大學(Virginia Tech)所建立之 3D 鋼架模型，符合 IFC2x3 標準，資料量適中、結構型式較具代表性，以此作為本小節之主題，平面圖與立體圖分別示於圖 4.61 與圖 4.62。此鋼架有 6 支 IfcColumn、21 支 IfcBeam 及 16 支 IfcMember，IfcMember 是描述結構體中的雜項較小型物件。以本研究自己撰寫之 IFCUser(請參閱第五章)檢視可以馬上看出其中 2 支梁為 T 形，其餘梁柱皆為 I 形，柱皆為 I 形。以下我們將其匯入 3 套軟體討論之：

1. 3D 鋼架 IFC 檔→Revit→IFC

將鋼架 IFC 檔案匯入 Revit，發現平面圖(圖 4.63)中只有 6 支柱確定有呈現，短向的梁與小梁無法呈現，至於長向，梁與 Member 重疊的部份在此不易分辨。再來看 3D 視圖(圖 4.64)，視覺上所呈現的物件與原建模圖無明顯差異，平面圖中的情形曾發生於 4.2.1 小節，同樣是梁只能在 Site 模式顯示，卻因 Site 模式傳遞遺失無造成無法呈現。再匯出 IFC 檔案，詳細檢查其中的資訊發現 6 支 IfcColumn 外其餘 IfcBeam 及 IfcMember 全部變成由 IfcBuildingElementProxy 類別定義。再將 Revit 匯出 IFC 檔以 IfcStoreyView 開啟，平面圖(圖 4.65)與立體圖(圖 4.66)中所有的梁及 Member 的著色依 IfcStoreyView 對 IfcBuildingElementProxy 的描述而呈現紅色，所有的柱著色依 IfcStoreyView 對 IfcColumn 的描述應呈現的綠色。

2. 3D 鋼架 IFC 檔→Bentley→IFC

將鋼架 IFC 檔案匯入 Bentley(圖 4.67)，發現梁柱皆可呈現，而 Member 卻全部消失了。再匯出 IFC 檔案，詳細檢查其中的資訊發現果然 IfcMember 全部消失，柱與梁依然存在，因此推測 Bentley 可能無法辨識 IfcMember 類別物件。再將 Bentley 匯出 IFC 檔以 IfcStoreyView 開啟，平面圖(圖 4.68)與立體圖(圖 4.69)中所顯示出來的圖形與前述之資訊相符。

3. 3D 鋼架 IFC 檔→ArchiCAD→IFC

將鋼架 IFC 檔案匯入 ArchiCAD，平面圖與立體圖分別示於圖 4.70 與圖 4.71，發現所有物件皆可呈現，由圖中無法察覺有何異狀。再匯出 IFC 檔案，詳細檢查其中的資訊發現除了檔案有所精簡之外，所有資訊無遺失，對於此鋼架 ArchiCAD 是 3 套軟體中轉換率最高的。再將 ArchiCAD 匯出 IFC 檔以 IfcStoreyView 檢視，平面圖(圖 4.72)與立體圖(圖 4.73)中所顯示出來的圖形與原 3D 鋼架圖形相符。

4.5.2 IFCServe 模型

IFCServe 為一免費的 IFC 類別資料庫，搭配 C/C++、Java、VB 等等可以大大助於 IFC 資料擷取與建立的工作，而本研究之擷取與建立程式是使用 Fortran 撰寫，IFCServer 無法支援。在此藉助他人相關 IFCServer 研究的建立功能部分，來建立一簡單模型來幫助探討，平面圖與立體圖分別示於圖 4.74 與圖 4.75，此模型有 4 支 IfcColumn 及 4 支 IfcBeam，各 4 種斷面形狀，包括矩形、圓形、T 形及 I 形，皆依 IFC 獨特類別定義其斷面形狀。以下我們將其匯入 3 套軟體討論之：

1. IFCServe IFC 檔→Revit→IFC

將 IFCServe 產生之 IFC 檔案匯入 Revit，可以發現平面圖(圖 4.76)中只有矩形柱與矩形梁有呈現，而立體圖(圖 4.77)中則皆可呈現，平面圖的狀況應當也類似於 4.2.1 與 4.5.1 及 4.5.2 小節，因顯示模式而造成。再由 Revit 匯出 IFC 檔案，詳細檢查其中的資訊發現 4 支 IfcColumn 外其餘 4 支 IfcBeam 全部變成由 IfcBuildingElementProxy 類別定義。再將 Revit 匯出之 IFC 檔以 IfcStoreyView 檢視，平面圖(圖 4.78)與立體圖(圖 4.79)中所有的梁著色依 IfcStoreyView 對 IfcBuildingElementProxy 的描述而呈現紅色，所有的柱著色依 IfcStoreyView 對 IfcColumn 的描述而呈現綠色。

2. IFCServe IFC 檔→Bentley→IFC

將 IFCServe 產生之 IFC 檔案匯入 Bentley，四視圖(圖 4.80)中可以發現梁、柱都只顯示矩形、圓形與 I 形斷面，T 形無法呈現。再以 Bentley 匯出 IFC 檔案，詳細檢查其中的資訊發現 T 形梁柱消失，其餘 3 支梁與柱仍存在。再將 Bentley 匯出 IFC 檔以 IfcStoreyView 檢視，平面圖(圖 4.81)與立體圖(圖 4.82)中所顯示出來的圖形與前述之資訊相符。

3. IFCServe IFC 檔→ArchiCAD→IFC

將 IFCServe 產生之 IFC 檔案匯入 ArchiCAD，平面圖與立體圖分別示於圖 4.83 與圖 4.84，發現所有物件皆可呈現，由圖中無法察覺有何異狀。再由 ArchiCAD 匯出 IFC 檔案，詳細檢查其中的資訊，發現檔案內物件資訊相差無幾。再將 ArchiCAD 匯出 IFC 檔以 IfcStoreyView 檢視，平面圖(圖 4.85)與立體圖(圖 4.86)中與原 IFCServe 建模圖相符。

4.6 小結

本研究 4.2 至 4.5 節盡量把三大繪圖軟體對 IFC 標準的支援範圍、轉換情況等等做交叉比對，並推論各個問題的所在。由於這些軟體皆是包裝好的應用軟體，無法一窺其內部程式撰寫情形，所有問題只能以旁證推論其發生原因。各軟體對不同斷面的描述方式整理為表 4.1，而三大軟體對於不同出處之 IFC 檔案在接受度與轉換度方面，本研究將其總結於表 4.2。

在前面的討論中，常有物件資訊在不同軟體間轉換後，平面圖可呈現卻無法在 3D 視圖中呈現的情形，且都是發生在 IfcArbitraryClosedProfileDef 這類以點、線表示斷面形狀的描述下，足以證明目前業界強調 BIM 精神的重要性，這類易產生誤差的描述方式，除非是 IFC 仍未定義的斷面形狀，否則不建議以此類別做描述。

Bentley Architecture 在其本身建模匯出 IFC 檔案時，無論柱或梁皆是以 IfcBeam 表示，卻在 4.2.2 與 4.4.3 小節，由它軟體建立之模型匯入 Bentley 再匯出 IFC 檔案時，仍保留了原軟體大部分的描述方式。這說明了 Bentley 似乎對於其他軟體建立匯入的 IFC 檔案有另外一套獨特的匯入匯出程序控制。

各項比較：

1. 支援 IFC 範圍：Bentley > ArchiCAD > Revit
2. 類別使用適切性：Revit > ArchiCAD > Bentley
3. 自我匯出匯入完整性：ArchiCAD > Bentley > Revit
4. 它軟體匯出匯入完整性：Bentley > ArchiCAD > Revit



第五章 IFC 資料擷取與建立及 IFCUser 視窗開發

5.1 前言

IFC 標準的發展為了涵蓋建築業分工範圍，及未來可擴充性，所以在格式的要求上有相當的自由度。本研究所建立的程式必須能夠擷取來自各個不同軟體匯出的 IFC 檔案，如 Revit、Bentley、ArchiCAD 以及網路上抓取的範例，儘可能的擴大其適用性，以滿足大多數的使用者。而後筆者再依 IFC 標準之定義內較簡潔的方式示範建立 IFC 檔案。這些都是以 Fortran 程式語言寫成。

本研究為了使用者便於使用擷取與建立 IFC 資料的功能，開發 IFCUser 視窗程式使用者介面。由於 Fortran 在視窗程式的開發上非常複雜且不符合效益，故結合 Visual Basic 程式語言在視窗開發上的優勢來完成整套程式的運作。

5.2 IFC 檔案資訊擷取

5.2.1 程式建構策略

IFC 資料是以多個樹狀關係連結每筆資訊，所以程式在擷取資料時需由起始點(IfcProject、IfcColumn、IfcBeam 等等)出發，隨著樹枝的發展逐步擷取所需資訊。

在編寫主程式時，盡量以簡潔與常駐性功能為考量，而主要擷取功能則交由各個副程式去分工合作，以減少編寫時的錯誤機會與日後維護的方便。另外，有些主程式與副程式常用到的變數，設定一個 Module 將其儲存為全域變數，這樣不論在主程式或副程式中要使用這些變數時，只要使用 Use 這個指令即可使用，省去重複宣告與傳遞的困擾。IFC 檔案內每筆資料皆有其資料名稱，相同資料名稱的每筆資料皆有其固定格式。除主程式外，我們依照所用到的資料名稱逐一建立副程式，讓一份 IFC 檔案中每筆資料皆有其對應的副程式，採多對一制(圖 5.1)，以使本程式可以應用於任何含有柱、梁等的 IFC 檔案。

由不同軟體建立之 IFC 檔案內，雖有相同名稱的資料，格式卻有些微差異，如空格的數量與位置、長度過長時對於斷行的處理等等；而且隨著檔案內所描述物件的多寡，單筆資料所存在的資訊或連結量不盡相同，這使得同名之單筆資

料常常在長度上差異數倍，甚至數百倍，例如 IfcRelAssignsToGroup 這個資料名稱中有一項屬性叫作 RelatedObjects，這個屬性需連結整份檔案中的 IfcStructuralPointConnection，當檔案內物件越多、連結量越多，此筆資料就越長。這些在同名稱資料所發生的變化，我們在建立擷取程式時也必須在相對應的同一個副程式裡充分判讀無誤，才能適用於任意 IFC 檔案。

5.2.2 程式運作步驟

當程式開啟時，主程式會先判讀檔案內資料的筆數以及大小，使電腦給於適當的記憶體容量以儲存資料內容，再依資料名稱判斷是否為擷取的進入點。當讀到起始點時便開始呼叫副程式擷取對結構物件有所描述的資訊，再判斷接下來所應呼叫的其他副程式，如此循環產生一連串連結以及擷取的動作。當一個物件資訊擷取完成，運作步驟會再回到主程式區，尋找下一個起始點，直到檔案結束為止。

所有擷取出來的資訊會分為兩個檔案輸出，第一個儲存工程方面較關心的資訊，如物件種類、物件長度、斷面形狀、斷面積、I 值以及斷面的細部描述等等二十多項，會再由 IFUser 讀取後展示出來(IFUser 會在本章第四節做介紹)，其中斷面積、Ix、Iy 等在 IFC 檔案中並沒有描述，須在程式中寫入運算式，以檔案中已有的資訊運算出來。另一個輸出的檔案是儲存其他較不常用卻仍有擷取價值的資訊，可以供有心人士參考，或作為往後程式擴充維護時參考。

5.3 IFC 檔案資訊建立

5.3.1 程式建構策略

IFC 格式在定義中有相當的自由度，可依使用者的需求搭配，筆者以較簡潔的搭配方式來建立一個完整的 IFC 檔案。這部份算是示範性質的研究，因為在大多數情況下，不會有人喜歡以純輸入數據的方式建立物件圖模型，還是需要以繪圖軟體為平台，以其供應的虛擬空間來繪製 3D 模型才符合方便性。但是目前大多數有支援 IFC 格式的軟體所能描述的物件斷面種類少於 IFC 已經定義的，所以此部分的研究在於示範性建立大多數繪圖軟體所不足的斷面形狀。

本程式所能建立的斷面形狀包括矩形、圓形、C形、I形、L形、T形、U形和Z形，所需輸入的屬性值包括物件類別、斷面形狀、物件長度、斷面細部、坐落座標等等。

IFC 建立程式中副程式的部份大致上與 IFC 擷取程式在命名與呼叫順序上差不多，但是其內部功能卻全然不同。然而由於 IFC 截取程式須配合各家軟體所輸出的 IFC 檔案格式，使其內部判讀機制較為複雜，而 IFC 建立程式是以筆者認為較適當的格式輸出，格式單一，使得在程式撰寫時反而較容易些，其樹狀圖如圖 5.2 所示。

5.3.2 程式運作步驟

程式運作須一步一步讀入所要建立的物件訊息，如斷面形狀、各項尺寸、起始座標等等，而這些訊息都可經由 IFCUser 做輸入，會存在一個訊息文件內。如同擷取程式，建立時依然按照樹狀方式一步一步建立 IFC 資訊，由所讀入的訊息資訊選擇樹根並逐步發展，每當需要變數時，就會由訊息文件內讀取。

每筆 IFC 資料開頭皆有其特定的資料代碼，為使此代碼不重複，有在程式中設立一 Module 來儲存全域變數，以供計數之用，代碼間不需連號，也不須依大小排列，只要不重複即可。

5.4 IFCUser 開發

5.4.1 程式建構目的

1. 將擷取與建立的功能結合在一起，使兩個功能在同一介面下操作。
2. 方便使用者閱覽所擷取之物件資訊。
3. 方便使用者輸入建立 IFC 結構物資訊所必要之數據。
4. 在這個電腦 Windows 化的時代，發展視窗使用者介面較易使大眾接受與操作。

5.4.2 程式建構策略

此程式主要在於以它輸出輸入介面結合擷取與建立 IFC 程式，所以需在適當的時機啟動另外兩個已編譯好的執行檔，VB 中可以 Shell 指令執行此步驟，那麼

寫作程式的重心就在於輸出與輸入的規劃。輸出介面筆者是以列表形式作為擷取資料展方式，輸入介面是另闢一個對話盒，將輸入欄位列於其上。

5.4.3 程式運作步驟

由於 VB.NET 本身就屬於物件導向語言，所以本程式的運作是在執行時由操作者來決定其流程。

當要使用擷取功能時，使用者可以自由的選擇所想要閱覽的 IFC 檔案，選擇好後就會執行 Shell 指令以啟動擷取程式，擷取程式工作完後會輸出一份文件檔，再由 IFCUser 讀入以列表形式展現出來。

當要使用建立功能時，會開啟另一個對話盒，此對話盒上會提供使用者選擇物件類別、斷面形狀；輸入物件長度、斷面細部尺寸以及坐落座標，並且在使用者選擇斷面形狀時會有一參考圖說隨之改變，斷面細部各欄位亦會隨之啟用或停用。使用者可以加入多個物件後一次輸出成為一 IFC 檔案，並讓使用者自由輸入檔案名稱與選擇儲存位置。



5.5 擷取與建立操作

5.5.1 擷取步驟

1. 由於擷取與建立 IFC 的程式皆已由 IFCUser 使用者介面程式包裝起來，所以使用起來非常簡便，只需將 IFCUser.exe、IFC_MAIN.exe、IFC_CREATE.exe 這三個執行檔放在同一個資料夾，然後滑鼠左鍵雙擊 IFCUser.exe 即可開始程式的運作。
2. 起始畫面中(圖 5.3)，左上角有「檔案(F)」、「建立(C)」、「說明(H)」三個選項。
3. 若要擷取某 IFC 檔案資訊，請點選檔案→載入檔案，如圖 5.4 所示。
4. 然後程式會自動出現開啟舊檔畫面(圖 5.5)，讓使用者選取所要擷取的 IFC 檔案。
5. 若載入檔案成功，就會出現一個成功訊息，如圖 5.6 所示。

6. 然後程式中就會展示出此檔案內的物件資訊(圖 5.7)，所整理出來的資訊包含物件類型、斷面形狀、物件長度、斷面積、慣性矩、斷面細部尺寸等等結構資訊。

5.5.2 實例示範—擷取 IFC 資料

由於本擷取功能在設計時已將各家軟體匯出 IFC 格式的差異性納入考量，故在此以第四章三家軟體所建立之模型為例，依前小節步驟操作後，結果如下：

1. AutoDesk Revit：

擷取完成後資料如圖 5.8 所示，柱的部分資訊由上而下分別是矩形建築柱、圓形建築柱、矩形結構柱、圓形結構柱、U 形結構柱、I 形結構柱、空心矩形結構柱、空心圓形結構柱共 8 支；梁的部分資訊由上而下分別是矩形梁、U 形梁、I 形梁、L 形梁、T 形梁、空心矩形梁、空心圓形梁共 7 支。

資料中可以看出矩形柱、梁斷面是以 IfcRectangleProfileDef 表示，圓形柱是以 IfcCircleProfileDef 表示，U 形、I 形柱及 U 形、I 形、L 形、T 形梁斷面是以 IfcArbitraryClosedProfileDef 表示，空心矩形與空心圓形柱、梁斷面是以 IfcArbitraryProfileDefWithVoids 表示。將此截取出的資訊回到 4.2 節作比對，驗證擷取之資訊與建模時相同。

2. Bentley Architecture：

擷取完成後資料如圖 5.9 所示，柱的部分資訊由上而下分別是矩形柱、圓形柱、I 形柱、L 形柱、U 形柱、T 形柱、空心矩形柱、空心圓形柱、C 形柱共 9 支；梁的部分資訊由上而下分別是矩形梁、圓形梁、I 形梁、L 形梁、U 形梁、T 形梁、空心矩形梁、空心圓形梁、C 形梁共 9 支。

資料中可以看出圓形柱、梁斷面是以 IfcCircleProfileDef 表示，矩形、C 型、I 形、L 型、T 型、U 型柱、梁斷面是以 IfcArbitraryClosedProfileDef 表示，中空矩形與空心圓形柱、梁斷面是以 IfcArbitraryProfileDefWithVoids 表示，將此截取出的資訊回到 4.3 節作比對，驗證擷取之資訊與建模時相同。

3. Graphisoft ArchiCAD：

擷取完成後資料如圖 5.10 所示，柱的部分資訊由上而下分別是矩形柱、圓形柱、C 形柱、U 形柱、L 形柱、I 形柱、空心矩形柱、T 形柱、Z 形柱共 9 支；梁的部分資訊由上而下分別是矩形梁、C 形梁、U 形梁、L 形梁、I 形梁、空心矩形梁、T 形梁、Z 形梁共 8 支。

資料中可以看出矩形柱、梁斷面以 IfcRectangleProfileDef 表示，圓形柱斷面以 IfcCircleProfileDef 表示，C 形、I 形、L 形、T 形、U 形、Z 形與空心矩形柱、梁斷面全都以 IfcArbitraryClosedProfileDef 表示，將此截取出的資訊回到 4.3 節作比對，驗證擷取之資訊與建模時相同。

5.5.3 建立步驟

1. 點選視窗左上端的「建立模型」->「加入」就可以進入建立 IFC 功能，如圖 5.11 所示。
2. 然後會出現一個建立專用對話盒(圖 5.12)。物件類型可以選擇柱及梁，起始預設為柱。柱預設為垂直方向，而梁為水平，梁可在 XY 平面轉動，依使用者需要可輸入 0~360 度。斷面形狀有八種選擇，矩形、圓形、C 形、I 形、L 形、T 形、U 形和 Z 形，起始預設為矩形，長度欄可供使用者輸入數字，單位為公尺(M)，若空白未輸入則預設為 300；再來是斷面細部尺寸，斷面細部各欄的開放或停用會隨著斷面形狀的選擇而改變，以避免使用者浪費時間輸入用不到的資訊，若各欄都留下空白而未輸入，筆者亦有設定預設值，不會發生無值的狀況；最後是輸入物件的座落位置，若空白則預設為(0,0,0)。
3. 設定好數值後，按下右下方的「加入」就會將剛剛所設定的資訊記憶起來，若使用者要建立多重物件，只要重複前步驟即可。當使用者所需物件都已加入後，按下右下角的「建立」，程式即會開始建立 IFC 檔案，建立完畢會出現成功訊息，如圖 5.13 所示。
4. 然後程式會跳出另存新檔畫面(圖 5.14)，供使用者輸入所要儲存的檔案名稱以及儲存位置。
5. 儲存完畢後也會出現成功訊息(圖 5.15)，建立 IFC 過程即完成。

5.5.4 實例示範—建立 IFC 資料

在此依前小節操作步驟建立一個 Frame 模型，平面圖與立體圖分別示於圖 5.16 與圖 5.17，模型分上下兩層，共 12 支柱與 14 支梁。為了趨於真實性，統一以 I 型梁柱建立。將此 Frame 依第四章的方式將其匯入 3 套軟體討論之：

1. IFCUser Frame→Revit→IFC

將 IFCUser 產生之 Frame 的 IFC 檔案匯入 Revit，發現平面圖(圖 5.18)中只有柱呈現，而 3D 視圖(圖 5.19)中則皆可呈現，平面圖的狀況應當也類似於 4.2.1 與 4.5.1 小節，因顯示模式而造成。再匯出 IFC 檔案，詳細檢查其中的資訊發現 12 支 IfcColumn 外其餘 14 支 IfcBeam 全部變成由 IfcBuildingElementProxy 類別定義。此情形到目前為止已是 Revit 所發生的常態現象。

2. IFCUser Frame→Bentley→IFC

將 IFCUser 產生之 Frame 的 IFC 檔案匯入 Bentley，四視圖(圖 5.20)中可以發現梁、柱皆可呈現。再由 Bentley 匯出 IFC 檔案，詳細檢查其中的資訊發現沒有太大差異。再將 Bentley 匯出 IFC 檔以 IfcStoreyView 開啟，平面圖與立體圖分別示於圖 5.21 與圖 5.22，所顯示出來的圖形與前述之資訊相符。

3. IFCUser Frame→ArchiCAD→IFC

將 IFCUser 產生之 Frame 的 IFC 檔案匯入 ArchiCAD，可以發現所有物件皆可呈現，無法察覺有何異狀。再由 ArchiCAD 匯出 IFC 檔案，詳細檢查其中的資訊發現檔案內物件資訊與原檔案相差無幾。

IFCUser 產生之 Frame 在此比較中沒有額外異常情形可能只是因為斷面形狀單一的关系。

另外，IFCUser 可建立的梁柱斷面形狀共有 8 種，包括矩形、圓形、C 形、I 形、L 形、T 形、U 形及 Z 形，皆是依 IFC 所定義之獨特類別定義。本小節以此建立功能拿來與第四章呼應，納入對三家軟體支援 IFC 狀況的探討，並將在探討完畢後納入表 4.1 的轉換率比較。

首先建立一簡單模型，此模型有 8 支 IfcColumn 及 8 支 IfcBeam，平面圖與立體圖分別示於圖 5.23 與圖 5.24。各 8 種斷面形狀，由左下至右上包括矩形、圓形、

C 形、I 形、L 形、T 形、U 形及 Z 形，皆依 IFC 獨特類別定義其斷面形狀。以下再將其匯入 3 套軟體討論之：

1. IFCUser IFC 檔→Revit→IFC

將 IFCUser 產生之 IFC 檔案匯入 Revit，發現平面圖(圖 5.25)中只有矩形柱與矩形梁有呈現，而 3D 視圖(圖 5.26)中則皆可呈現，平面圖的狀況應當也類似於 4.2.1 與 4.5.1 小節，因顯示模式而造成。再匯出 IFC 檔案，詳細檢查其中的資訊發現 8 支 IfcColumn 外其餘 8 支 IfcBeam 全部變成由 IfcBuildingElementProxy 類別定義。再將 Revit 匯出 IFC 檔以 IfcStoreyView 開啟，平面圖與立體圖分別示於圖 5.27 與圖 5.28，所有的梁著色依 IfcStoreyView 對 IfcBuildingElementProxy 的描述而呈現紅色，所有的柱著色依 IfcStoreyView 對 IfcColumn 的描述而呈現綠色。然而梁與柱的 Z 座標相對位置發生等比例誤差，此誤差是 Revit 匯出時才發生的。

2. IFCUser IFC 檔→Bentley→IFC

將 IFCUser 產生之 IFC 檔案匯入 Bentley，四視圖(圖 5.29)可以發現梁、柱都只顯示矩形、圓形與 I 形斷面，其餘皆無法呈現。再由 Bentley 匯出 IFC 檔案，詳細檢查其中的資訊發現居然只剩下 1 個矩形 IfcBeam，其餘全部消失，這表示 Bentley 對於較精簡之 IFC 檔案比較無法轉換。再將 Bentley 匯出 IFC 檔以 IfcStoreyView 開啟，平面圖與立體圖分別示於圖 5.30 與圖 5.31，所顯示出來的圖形與前述之資訊相符。

3. IFCUser IFC 檔→ArchiCAD→IFC

將 IFCUser 產生之 IFC 檔案匯入 ArchiCAD，可以發現所有物件皆可呈現，平面圖與立體圖分別示於圖 5.32 與圖 5.33，由圖中無法察覺有何異狀。再由 ArchiCAD 匯出 IFC 檔案，詳細檢查其中的資訊發現檔案內 C 形柱與梁是依照原本 IFCUser 的描述方式(IfcCShapeProfileDef)匯出的，但是各項尺寸中只保留了長度與寬度，其他尺寸變成 0。而其餘物件資訊則與原檔案相差無幾。再將 ArchiCAD 匯出 IFC 檔以 IfcStoreyView 開啟，平面圖(圖 5.34)中 C 形柱無法呈現，立體圖(圖 5.35)中 C 形柱與梁皆無法呈現。其餘物件圖皆與原圖相符。

第六章 結論與建議

6.1 結論

1. 本研究以梁與柱為主要對象來探討 AutoDesk Revit、Bentley ArchiTecture 及 GraphiSoft ArchiCAD 三家軟體對 IFC 標準的支援狀況。建置其所有支援的斷面形狀，以匯出匯入 IFC 檔案的方式進行轉換與比對，以了解它們在 IFC 資訊內容敘述的差異性與轉換率。
2. 由於 IFC 尚在發展當中，而 IFC 格式的自由度亦相當的高，市面上許多家軟體雖然宣稱自己支援 IFC，然而其相互間的相容性卻沒有達到預期，這仍是有待進一步努力的。
3. 本研究在了解三家軟體對於 IFC 資訊內容敘述的差異性後，以 Fortran 和 VB 混合程式語言，開發 IFCUser 套件實作擷取與建立 IFC 資訊的功能。擷取功能部份配合不同來源之 IFC 檔案的差異性，以增進其適用性。建立部分則可建立包含矩形、圓形、C 形、I 形、L 形、T 形、U 形和 Z 形的梁柱斷面形狀，以 IFC 標準中對這些斷面形狀獨特的描述方式建立之。
4. 所有宣稱支援 IFC 格式的繪圖軟體，必須包含匯入與匯出 IFC 檔案的功能，匯出即在建立相關 IFC 檔案，匯入後使用者即可擷取相關結構資訊。本研究雖然是比較初階的研究，卻是我們充分應用 IFC 的重要條件之一。
5. 資訊標準必須被推廣與應用，雖然 IFC/BIM 的發展尚未成熟，但本研究期望能拋磚引玉，增進研究後進對 IFC 資訊標準的興趣與啟發，使他們能在 IFC/BIM 領域為我國開拓一片天。

6.2 建議

1. 本研究的主要研究對象只包含柱與梁，但目前 IFC 所定義有關營建方面的還包括牆、板、門、窗、樓梯、Member 等等，這是研究後進可以多加探討的方向。

2. 本研究在 IFCUser 套件的製作時是使用 Fortran 與 VB 混合語言，尚有許多改善空間。未來可嘗試使用不同的程式語言來撰寫，以期能有更高的效率與更多的功能。
3. 本研究所實作的 IFC 擷取與建立功能，未來可嘗試結合繪圖軟體進行匯出與匯入的工作，使 IFC 標準做更完整的應用。
4. 目前 IFC 標準已發展完整的最新版本為 2x3，雖然能使用在結構分析方面的定義尚不完整，但發展仍在進行中，將來在這方面是可以預期的。在 IFC 不足的部份，先行開發一套系統程式來補足之，將可開啟結構分析方面的應用。



參考文獻

- [1] 楊秉蒼，楊宜縈，「營建電子化發展」，
<http://www.arch.net.tw/modern/month/291/291-2.htm>，2004。
- [2] 行政院 iAeB 推動小組，「產業電子化白皮書」，2000。
- [3] 潘稟嘉，「建築圖形資訊標準於營建業電子商務之應用研究」，碩士論文，
國立台灣大學，台北，2000。
- [4] General Services Administration, “GSA Building Information Modeling Guide Series” ,2006.
- [5] National Building Information Model Standard, “Overview National BIM Standard” ,2006.
- [6] Bentley, 「賓特利系統建築資訊模型白皮書—營造產業的提昇需要從零開始?」，2006。
- [7] BuildingSmart, <http://www.buildingsmart.de/index.htm>,2006.
- [8] Georgia Tech,“ISO STEP Overview”,2000.
- [9] 郭榮欽，「建築物生命週期資訊共享之研究」，博士論文，國立台灣大學，
台北，2000。
- [10] 李兆平，「建築設施在營運與維護階段資訊共享-以學校教室為例」，碩士
論文，國立交通大學，新竹，2001。
- [11] 曾仁杰，「工程設計階段及生命週期資料交換標準發展與展望」，第二屆土
木水利工程專業軟體應用經驗交流研討會暨專業軟體展示會，2006。
- [12] IAI,IFC2x3_Final_HTML_distribution,
[http://www.iai-international.org/Model/IFC\(ifcXML\)Specs.html](http://www.iai-international.org/Model/IFC(ifcXML)Specs.html),2006.
- [13] IAI,IFC Model View Definition Format,
http://www.iai-singapore.org/doc/20060410_IFC_MVD_Format.pdf,2006.
- [14] IAI,Industry Foundation Classes - Release 2x,
[http://www.iai-international.org/Model/IFC\(ifcXML\)Specs.html](http://www.iai-international.org/Model/IFC(ifcXML)Specs.html),2000.
- [15] IFC,IFC 2x2 – Industry Foundation Classes,
[http://www.iai-international.org/Model/IFC\(ifcXML\)Specs.html](http://www.iai-international.org/Model/IFC(ifcXML)Specs.html),2005.

- [16] 彭國倫，「精通 Fortran 90 程式設計」，碁峰資訊股份有限公司，2000。
- [17] 李敬賢，「傑西科技有限公司(AutoCAD)技術經理 BIM 在土木工程之應用」，學術演講，國立交通大學工二館 117 室(階梯教室)，2007。
- [18] Forschungszentrum Karlsruhe GmbH,
<http://www.iai.fzk.de/www-extern/index.php?id=1134&L=1>,2006.
- [19] G.E.M. Team Solutions,http://www.team-solutions.de/?page_id=18,1999.
- [20] IAI,IAI – International Alliance for Interoperability,
[http://www.iai-international.org/Model/IFC\(ifcXML\)Specs.html](http://www.iai-international.org/Model/IFC(ifcXML)Specs.html),2005.
- [21] 楊潮瑞，「工程營建資訊系統資訊交換標準之研究-以發包採購為例」，碩士論文，國立台灣大學，台北，2002。
- [22] 呂翊民，「公共工程技術資料庫整合計畫整合性之探討」，國立交通大學土木工程研究所，碩士論文，2003。



表 2.1 IFC 與 STEP 之異同[3]

	IFC	STEP
制定機構	Industry Alliance for Interoperability(IAI)	國際標準組織(ISO)
需求起源	軟體工具資訊交換技術的商業需求	整合美國國防工業產品資料
制定目標	針對各產業的產品和作業資訊，建立標準化的資料表達方式，以利於資料交換與共享	針對各產業的產品和作業資訊，建立標準化的資料表達方式，以利於資料交換與共享
製定範圍	針對營建產業及設施管理(AECFM)	涵蓋整個製造業
制定速度	較快	較慢
描述語言	ISO 10303 Part11,Express	ISO 10303 Part11,Express
繪圖軟體支援	AutoDesk Revit、Bentley ArchiTecture、GraphiSoft ArchiCAD 等多家軟體	目前尚無繪圖軟體完整支援STEP圖檔的存取

表 2.2 支援 IFC 之軟體[11]

軟體公司	軟體名稱
Archimen	Active3D
Autodesk	AutoCAD Architecture
Autodesk	Revit Building、Structure、Architecture
Bentley	Bentley Architecture
DDS	DDS-CAD Building Service
Nemetschek	Allplan
Graphisoft	ArchiCAD
Solibri	Solibri Model Checker
Tekla	Tekla Structures
Progman	MagiCAD
Vizelia	Facility Online

表 4.1 繪圖軟體對不同斷面描述方法對照表

		描述形狀									
		矩形	圓形	C 形	I 形	L 形	T 形	U 形	Z 形	空心矩形	空心圓形
軟 體	Autodesk Revit	IfcRectangle ProfileDef	IfcCircle ProfileDef	無	IfcArbitrary Closed ProfileDef	IfcArbitrary Closed ProfileDef	IfcArbitrary Closed ProfileDef	IfcArbitrary Closed ProfileDef	無	IfcArbitrary ProfileDef WithVoids	IfcArbitrary ProfileDef WithVoids
	Bentley Architectur e	IfcArbitrary Closed ProfileDef	IfcCircle ProfileDef	IfcArbitrary Closed ProfileDef	IfcArbitrary Closed ProfileDef	IfcArbitrary Closed ProfileDef	IfcArbitrary Closed ProfileDef	IfcArbitrary Closed ProfileDef	無	IfcArbitrary ProfileDef WithVoids	IfcArbitrary ProfileDef WithVoids
	Graphisoft ArchiCAD	IfcRectangle ProfileDef	IfcCircle ProfileDef	IfcArbitrary Closed ProfileDef	無						
	IFCServer	IfcRectangle ProfileDef	IfcCircle ProfileDef	無	IfcIShape ProfileDef	無	IfcTShape ProfileDef	無	無	無	無
	IFCUser	IfcRectangle ProfileDef	IfcCircle ProfileDef	IfcCShape ProfileDef	IfcIShape ProfileDef	IfcLShape ProfileDef	IfcTShape ProfileDef	IfcUShape ProfileDef	IfcZShape ProfileDef	無	無

表 4.2 繪圖軟體支援 IFC 之轉換率比較

建模軟體→匯出 IFC 檔							
		Autodesk Revit	Bentley Architecture	Graphisoft ArchiCAD	立體鋼架結構	IFCUser	IFCServer
匯 入 IFC 檔 ↓ 軟 體 ↓ 匯 出 IFC 檔	Autodesk Revit	完整轉換 50% 其餘類別改變	完整轉換 0% 全部類別改變	完整轉換 45% 其餘類別改變 或消失	完整轉換 30% 其餘類別改變	完整轉換 50% 其餘類別改變	完整轉換 50% 其餘類別改變
	Bentley Architecture	完整轉換 30% 其餘點座標誤 差過大或物件 消失	完整轉換 100%	完整轉換 100%	完整轉換 70% 其餘類別消失	完整轉換 15% 其餘類別改變 或消失	完整轉換 75% 其餘類別消失
	Graphisoft ArchiCAD	完整轉換 80% 其餘消失	完整轉換 85% 其餘消失	完整轉換 100%	完整轉換 100%	完整轉換 90% 其餘類別消失	完整轉換 95% 其餘類別消失

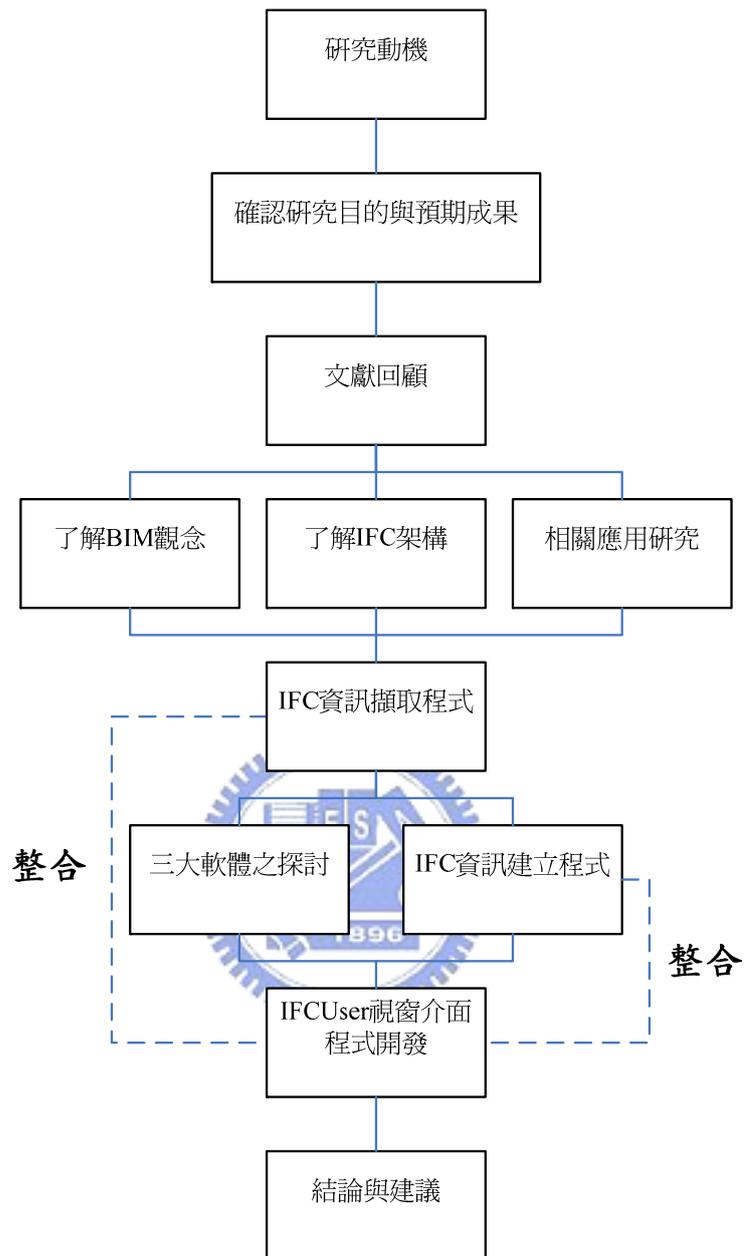


圖 1.1 研究流程圖

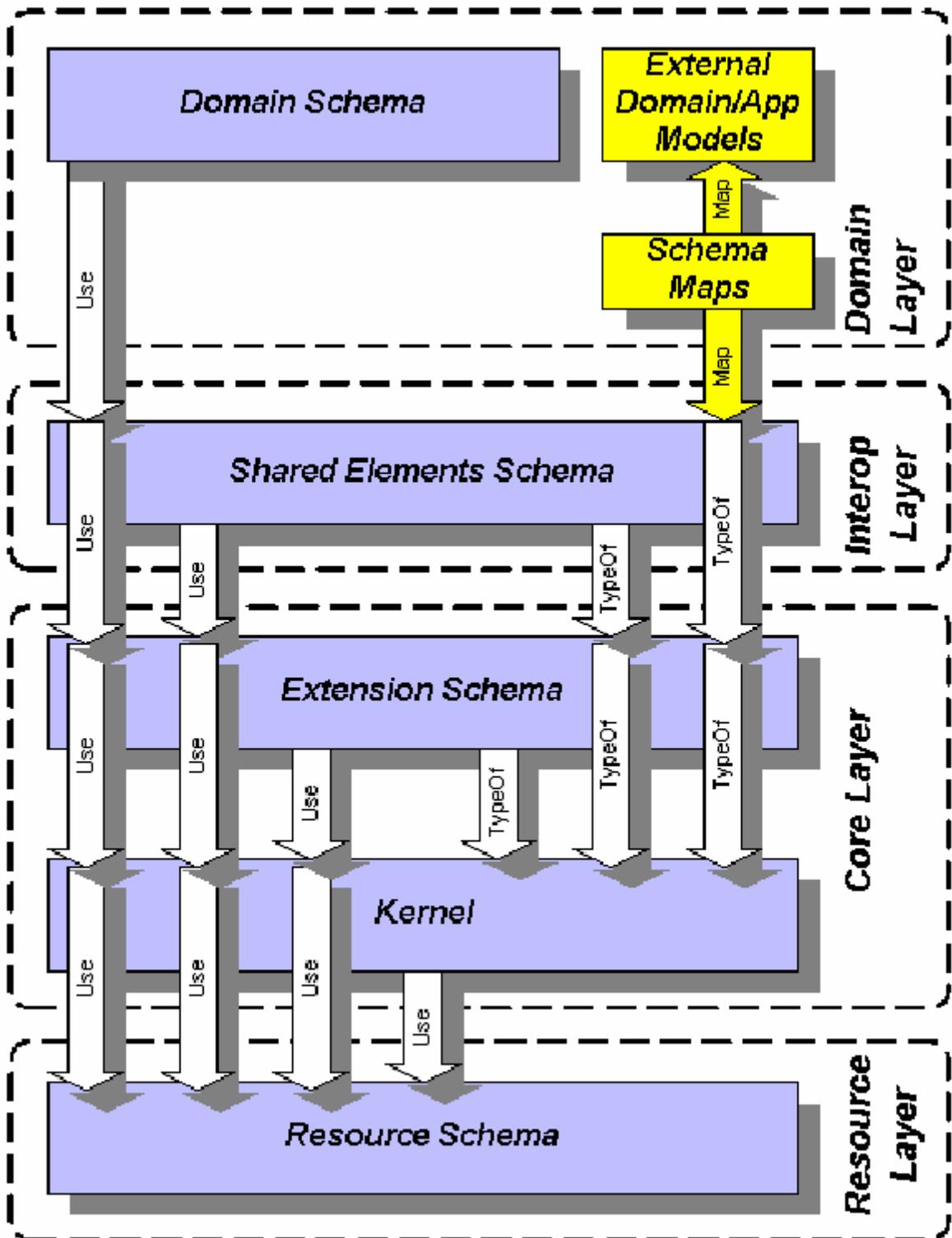


圖 3.1 IFC 架構之層級觀念[13]

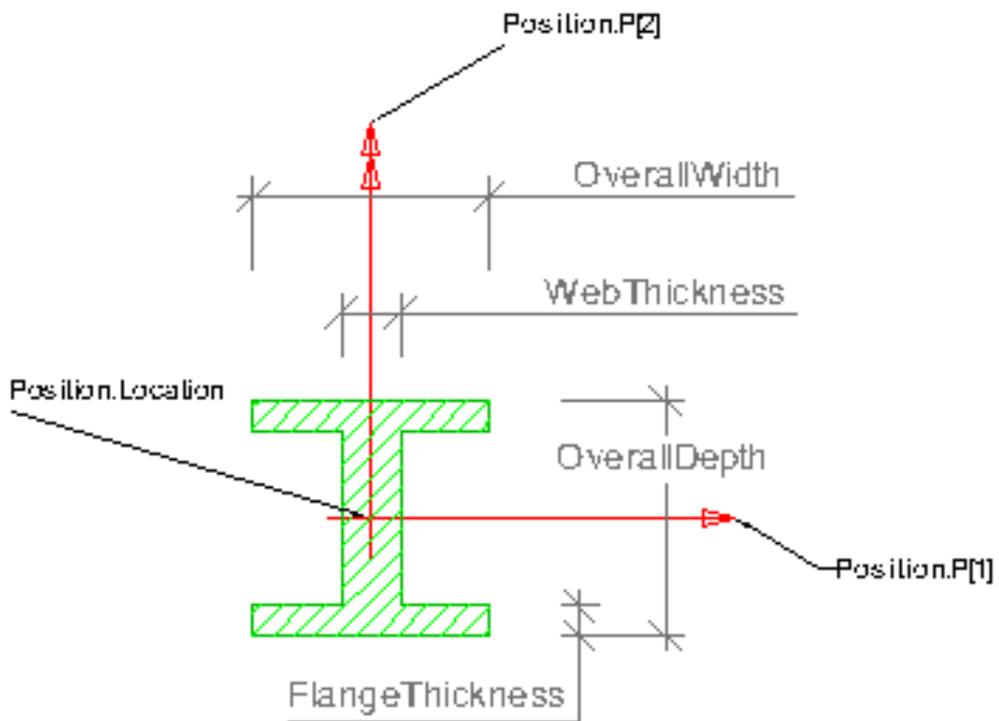
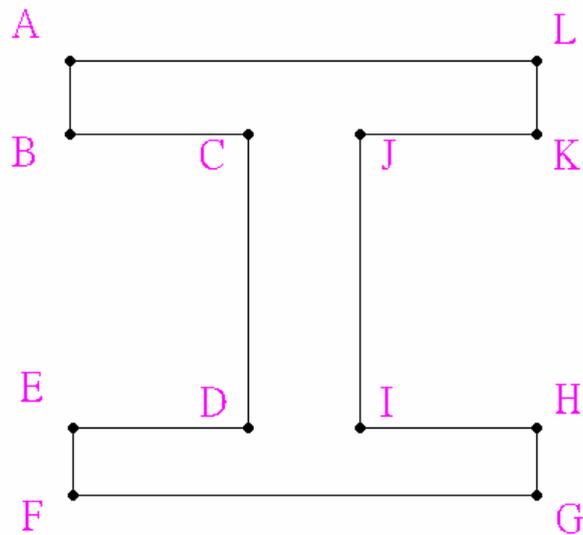
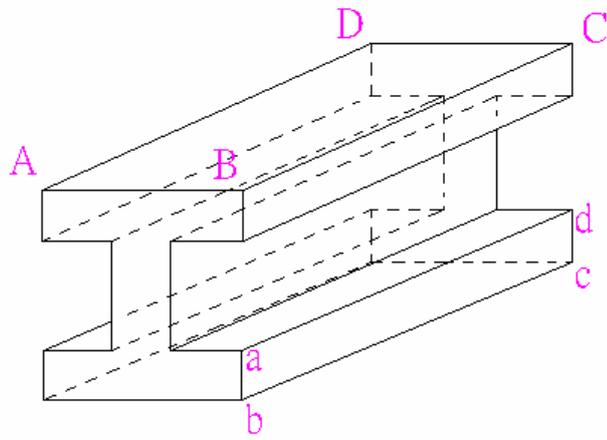


圖 3.3 I形斷面細部示意圖[11]



順序：A→B→C→D→E→F→G→H→I→J→K→L→A

圖 3.4 I形斷面各點連線圖



頂面：A、B、C、D
側面：a、b、c、d

圖 3.5 I 形梁以面組成示意圖

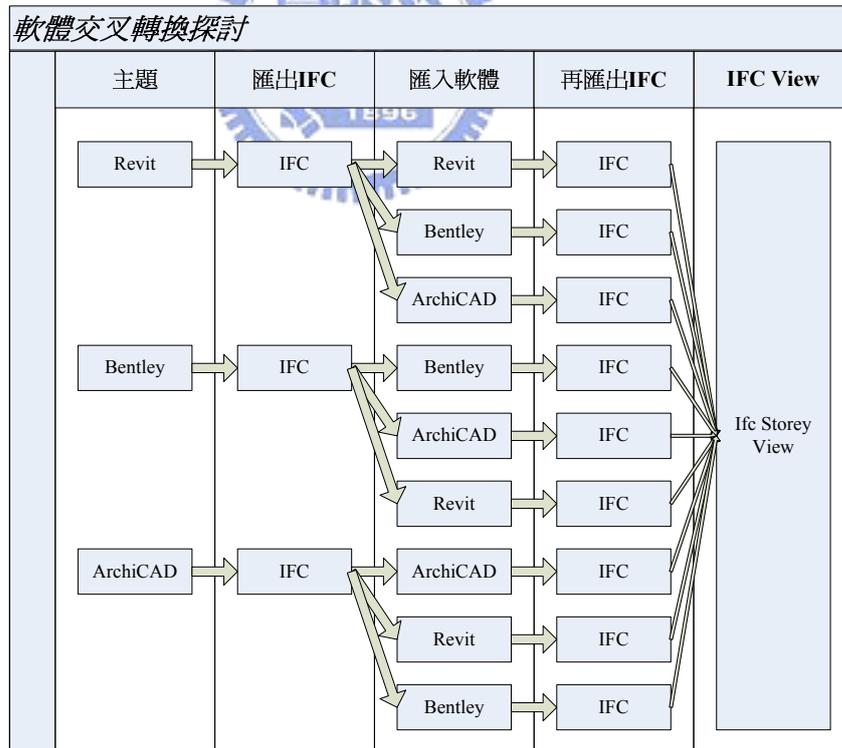


圖 4.1 軟體交叉探討示意圖

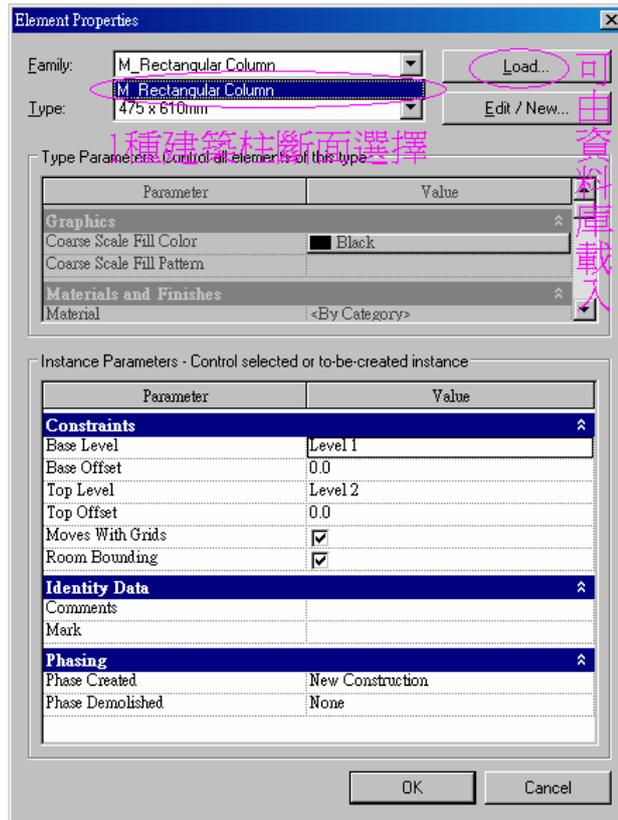


圖 4.2 Revit 建築柱斷面形狀選擇

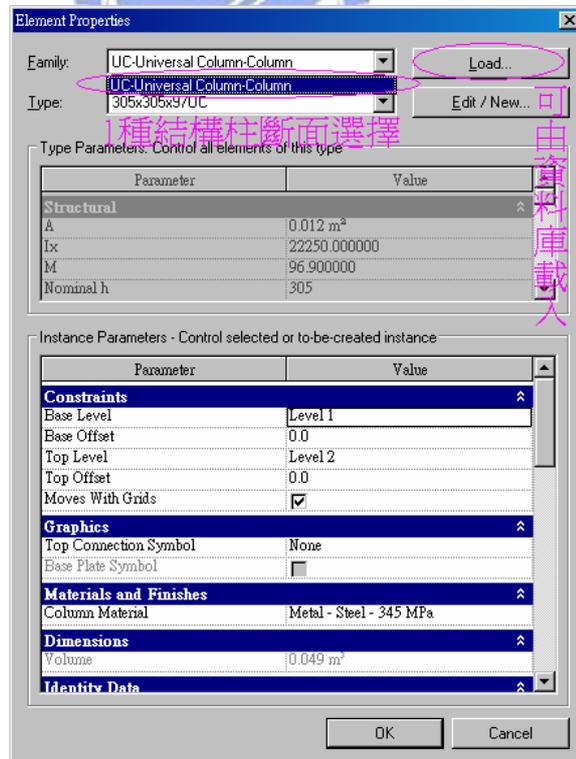


圖 4.3 Revit 結構柱斷面形狀選擇

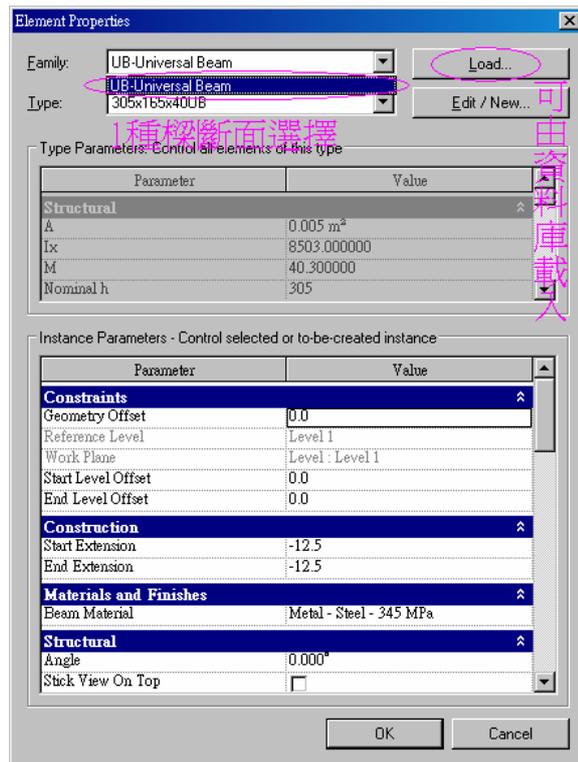


圖 4.4 Revit 梁斷面形狀選擇



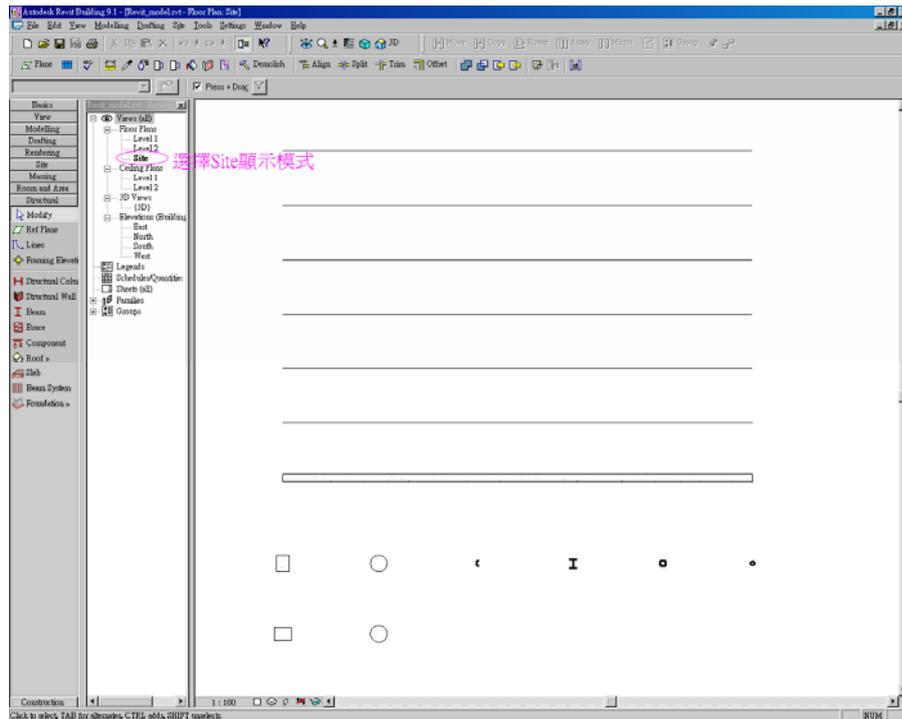


圖 4.7 Revit 建模平面圖

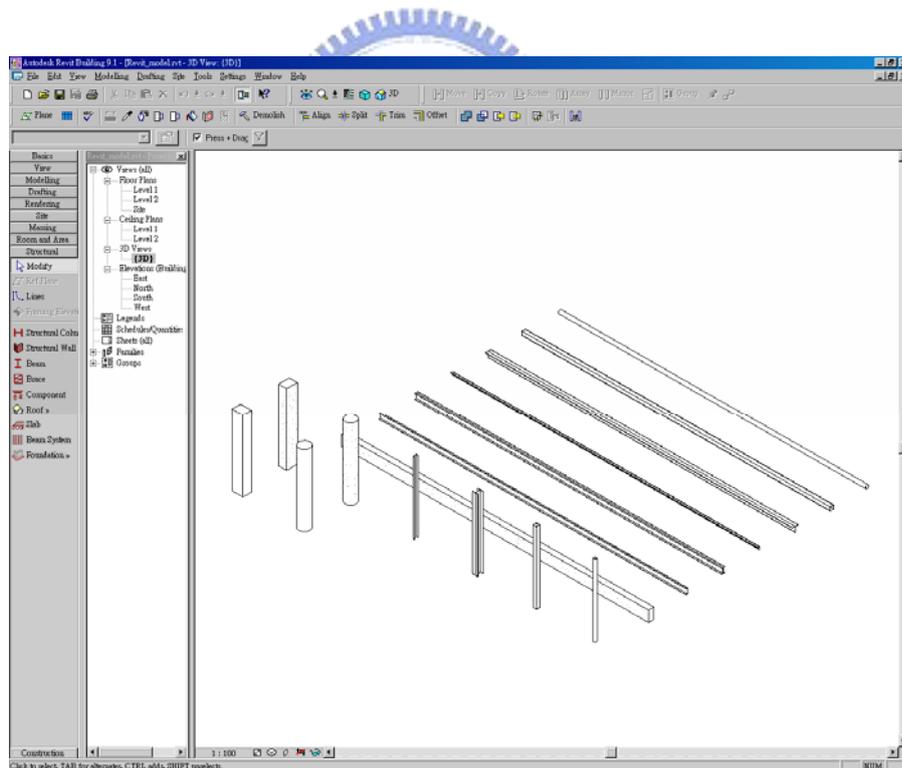


圖 4.8 Revit 建模 3D 視圖

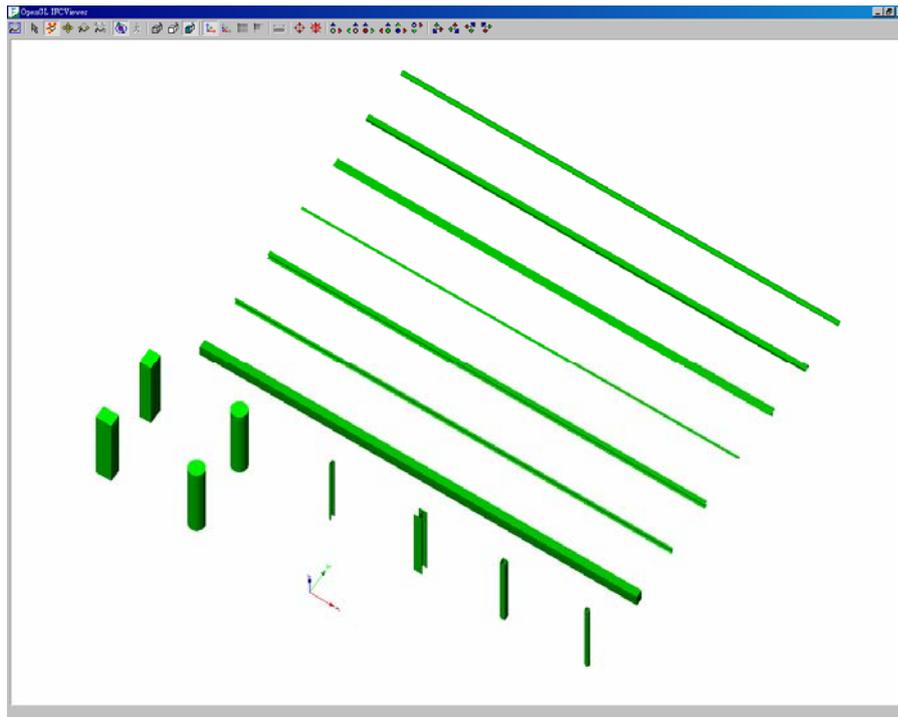


圖 4.11 Revit 建模匯出 IFC 檔以 IfcStoreyView 檢視立體圖

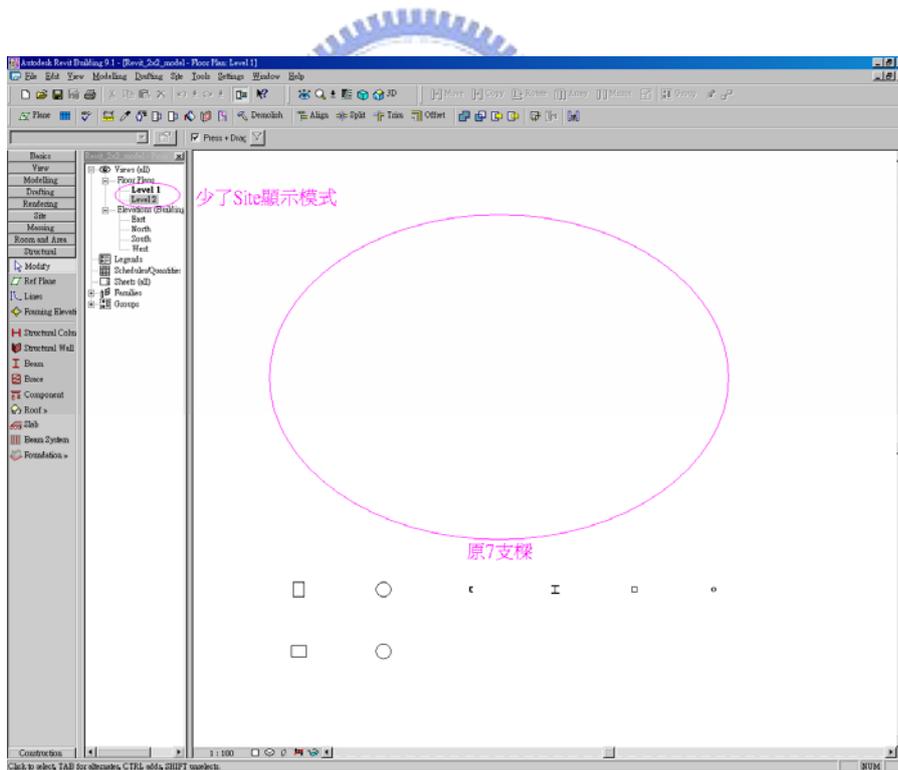


圖 4.12 Revit 建模匯出 IFC 再匯入 Revit 平面圖

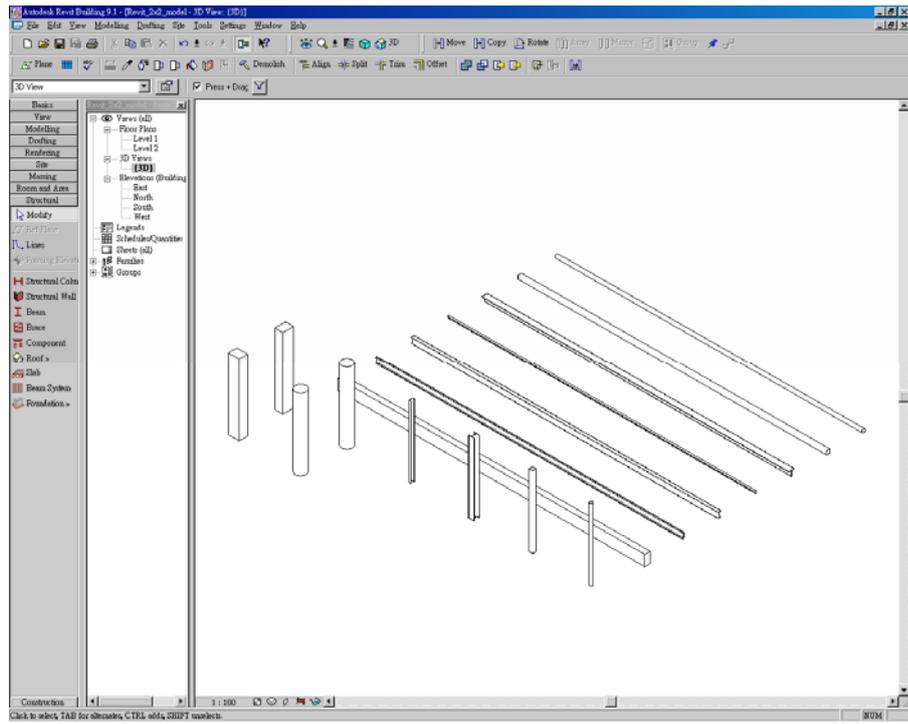


圖 4.13 Revit 建模匯出 IFC 再匯入 Revit 3D 視圖



```

#159=IFCBUILDINGELEMENTPROXY(2U2uIJUkD2chhh_PdG56MU',#19,M_Concrete-Rectangular Beam:300 x
├── #19=IFCOWNERHISTORY(#18,#2,$,NOCHANGE,$,$,$,0);
│   ├── #18=IFCPERSONANDORGANIZATION(#16,#17,$);
│   │   ├── #16=IFCPERSON($,$,fan,$,$,$,$,$);
│   │   ├── #17=IFCORGANIZATION($,$,$,$);
│   │   └── #2=IFCAPPLICATION(#1,Release 9,'Autodesk Revit Building 9','Revit');
│   │       └── #1=IFCORGANIZATION($,'Autodesk Revit Building 9',$,$,$);
├── #158=IFCLOCALPLACEMENT(#23,#157);
│   ├── #23=IFCLOCALPLACEMENT(#13,#22);
│   │   ├── #13=IFCLOCALPLACEMENT($,#12);
│   │   │   ├── #12=IFCAXIS2PLACEMENT3D(#3,$,$);
│   │   │   │   ├── #3=IFCCARTESIANPOINT((0,0,0));
│   │   │   └── #22=IFCAXIS2PLACEMENT3D(#3,$,$);
│   │   │       ├── #3=IFCCARTESIANPOINT((0,0,0));
│   │   └── #157=IFCAXIS2PLACEMENT3D(#154,#155,#156);
│   │       ├── #154=IFCCARTESIANPOINT((-9701.595272235934,-10151.59527223591,2398.40472776415));
│   │       ├── #155=IFCDIRECTION((0,0,1));
│   │       └── #156=IFCDIRECTION((1,0,0));
├── #153=IFCPRODUCTDEFINITIONSHAPE($,$,(#152));
│   ├── #152=IFCSHAPEREPRESENTATION(#15,'Body','MappedRepresentation',(#151));
│   │   ├── #15=IFCGEOMETRICREPRESENTATIONCONTEXT($,$,3,1.E-006,#14,$);
│   │   │   ├── #14=IFCAXIS2PLACEMENT3D(#3,$,$);
│   │   │   │   ├── #3=IFCCARTESIANPOINT((0,0,0));
│   │   │   └── #151=IFCMAPPEDITEM(#148,#150);
│   │   │       ├── #148=IFCREPRESENTATIONMAP(#147,#146);
│   │   │       │   ├── #147=IFCAXIS2PLACEMENT3D(#3,$,$);
│   │   │       │   │   ├── #3=IFCCARTESIANPOINT((0,0,0));
│   │   │       │   └── #146=IFCSHAPEREPRESENTATION(#15,'Body','Brep',(#145));
│   │   │       │       ├── #15=IFCGEOMETRICREPRESENTATIONCONTEXT($,$,3,1.E-006,#14,$);
│   │   │       │       │   ├── #14=IFCAXIS2PLACEMENT3D(#3,$,$);
│   │   │       │       │   │   ├── #3=IFCCARTESIANPOINT((0,0,0));
│   │   │       │       └── #145=IFCFACETEDBREP(#144);
│   │   │       │           ├── #144=IFCCLOSEDSHELL((#124,#131,#134,#137,#140,#143));
│   │   │       │           │   ├── #124=IFCFACE((#123));
│   │   │       │           │   ├── #131=IFCFACE((#130));
│   │   │       │           │   ├── #134=IFCFACE((#133));
│   │   │       │           │   ├── #137=IFCFACE((#136));
│   │   │       │           │   ├── #140=IFCFACE((#139));
│   │   │       │           │   └── #143=IFCFACE((#142));
│   │   │       └── #150=IFCCARTESIANTRANSFORMATIONOPERATOR3D($,$,#149,1,$);
│   │   │           └── #149=IFCCARTESIANPOINT((0,0,0));
└── #150=IFCCARTESIANTRANSFORMATIONOPERATOR3D($,$,#149,1,$);
    └── #149=IFCCARTESIANPOINT((0,0,0));

```

圖 4.14 IfcBuildingElementProxy 資料結構圖

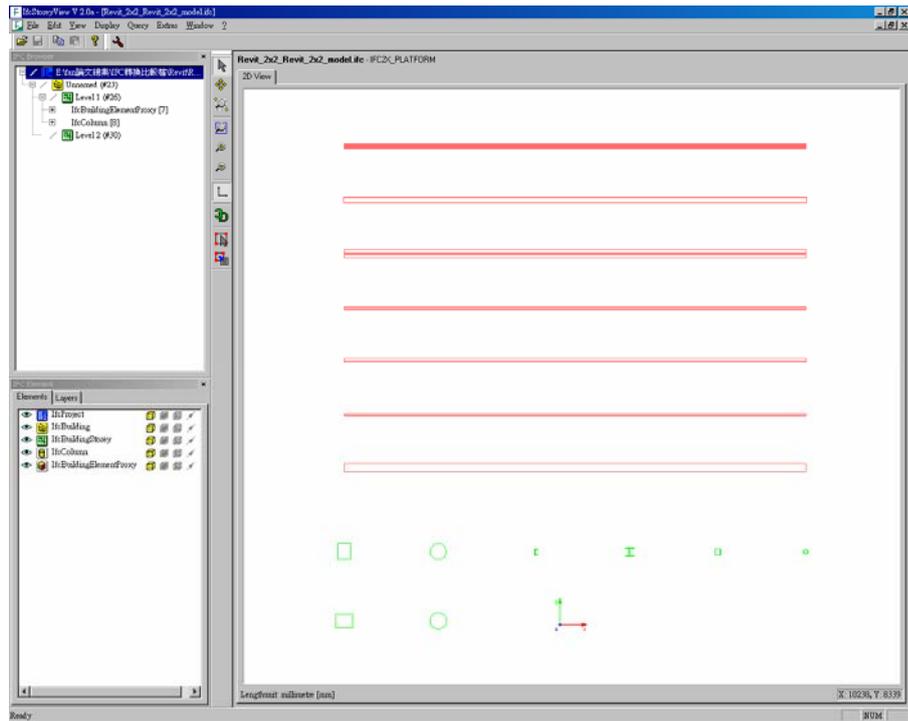


圖 4.15 Revit_Revit_model 以 IfcStoreyView 檢視平面圖

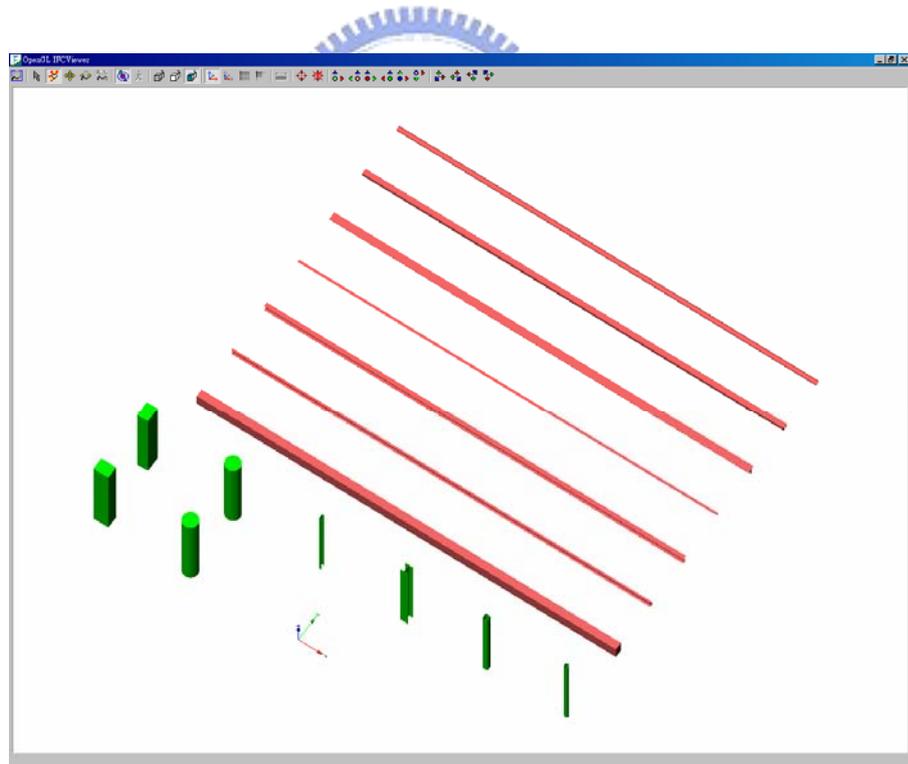


圖 4.16 Revit_Revit_model 以 IfcStoreyView 檢視 3D 視圖

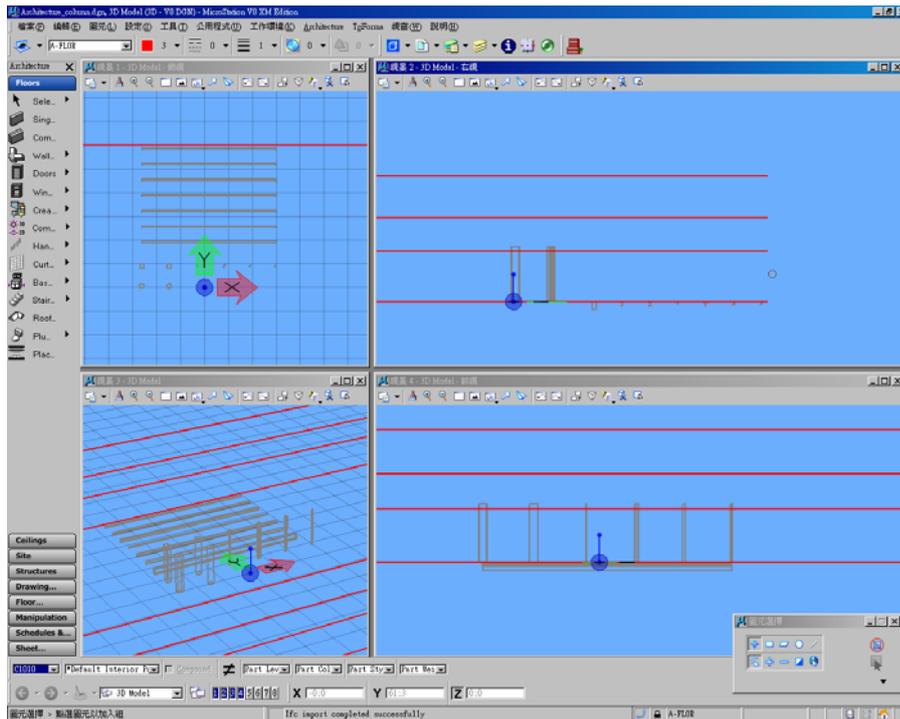


圖 4.17 Revit 建模匯出 IFC 再匯入 Bentley 四視圖

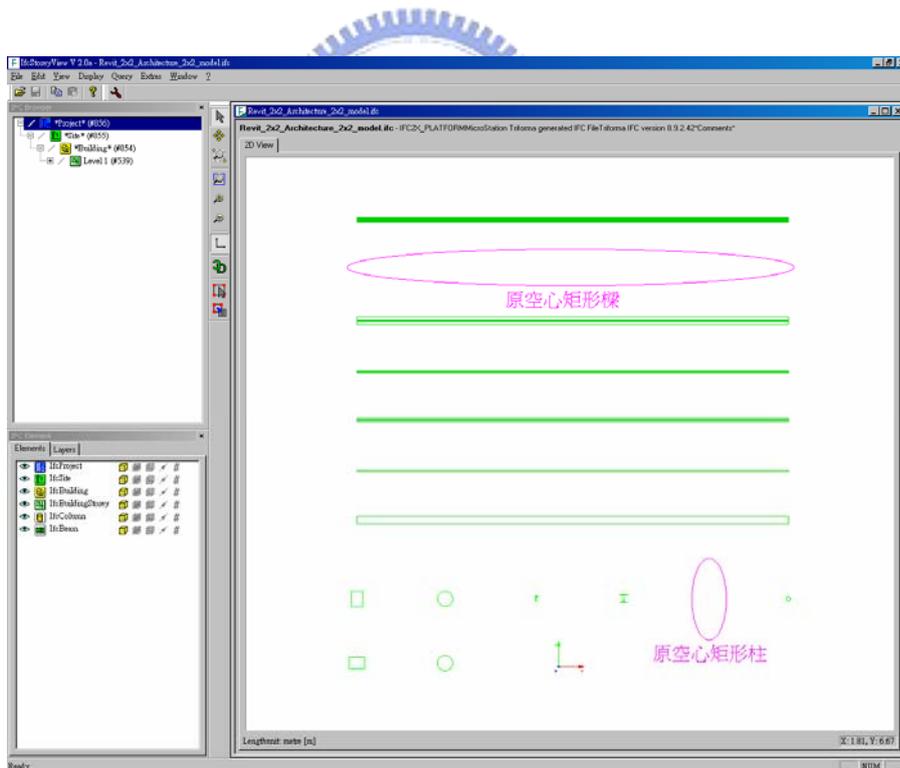


圖 4.18 R Revit_Bentley_model 以 IfcStoreyView 檢視平面圖

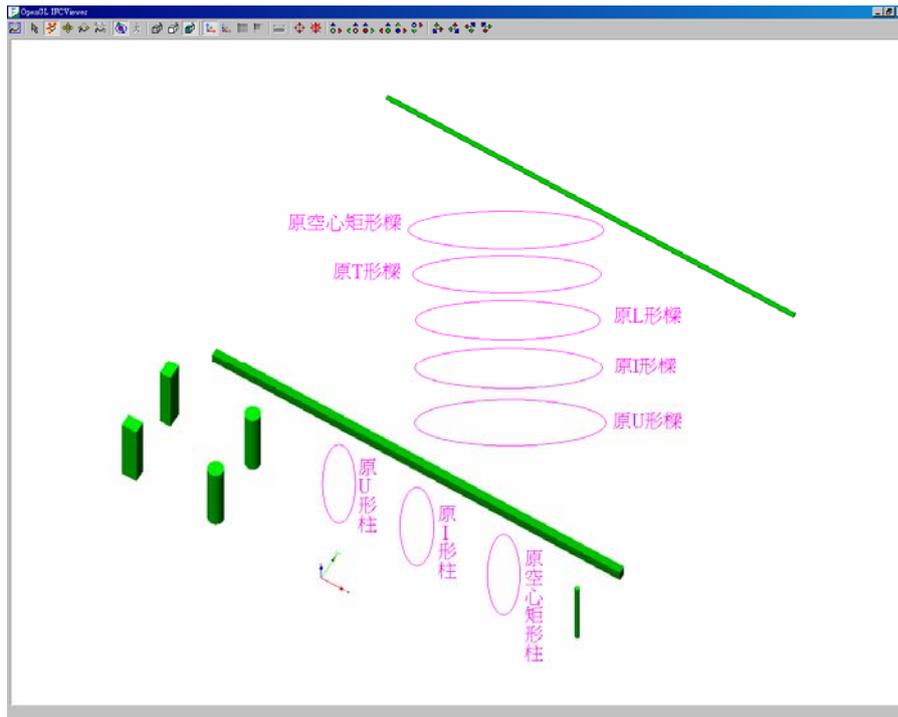


圖 4.19 Revit_Bentley_model 以 IfcStoreyView 檢視 3D 視圖

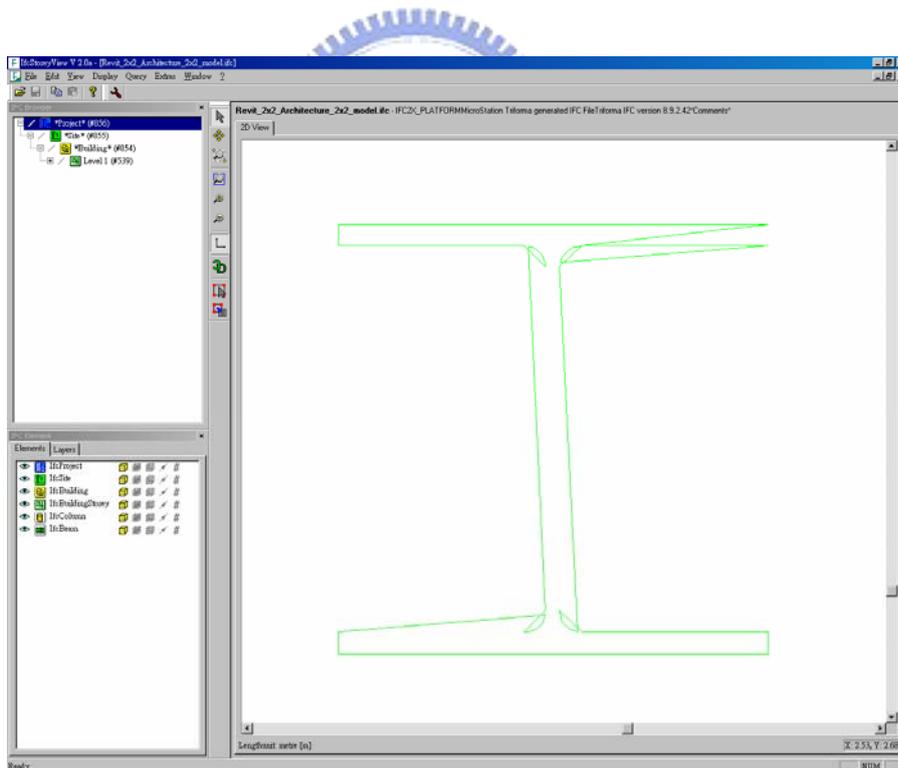


圖 4.20 Revit_Bentley_model 以 IfcStoreyView 檢視 I 斷面放大平面圖

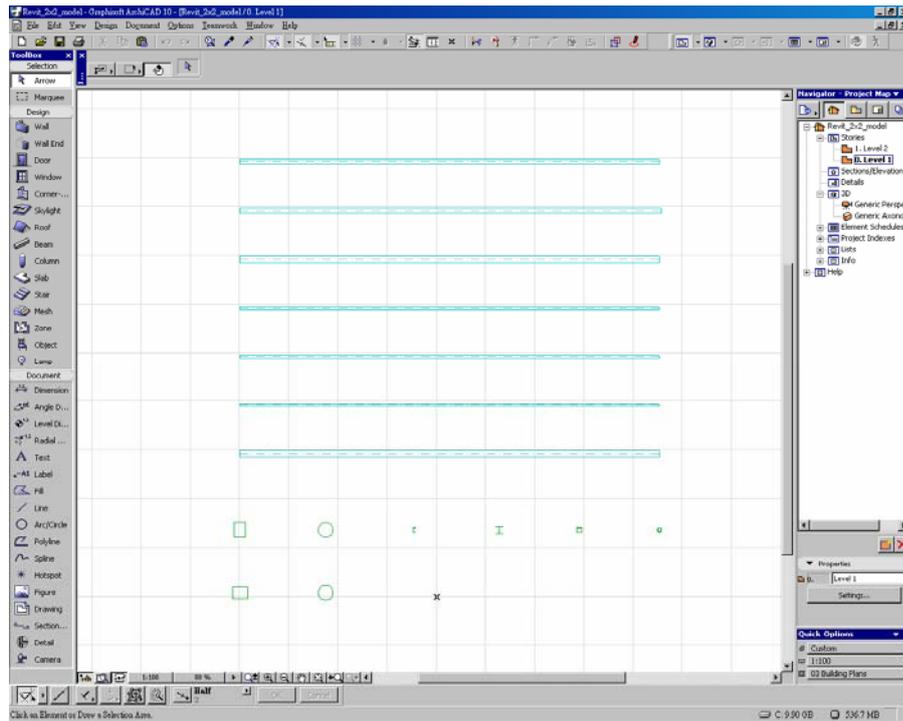


圖 4.21 Revit 建模匯出 IFC 再匯入 ArchiCAD 平面圖

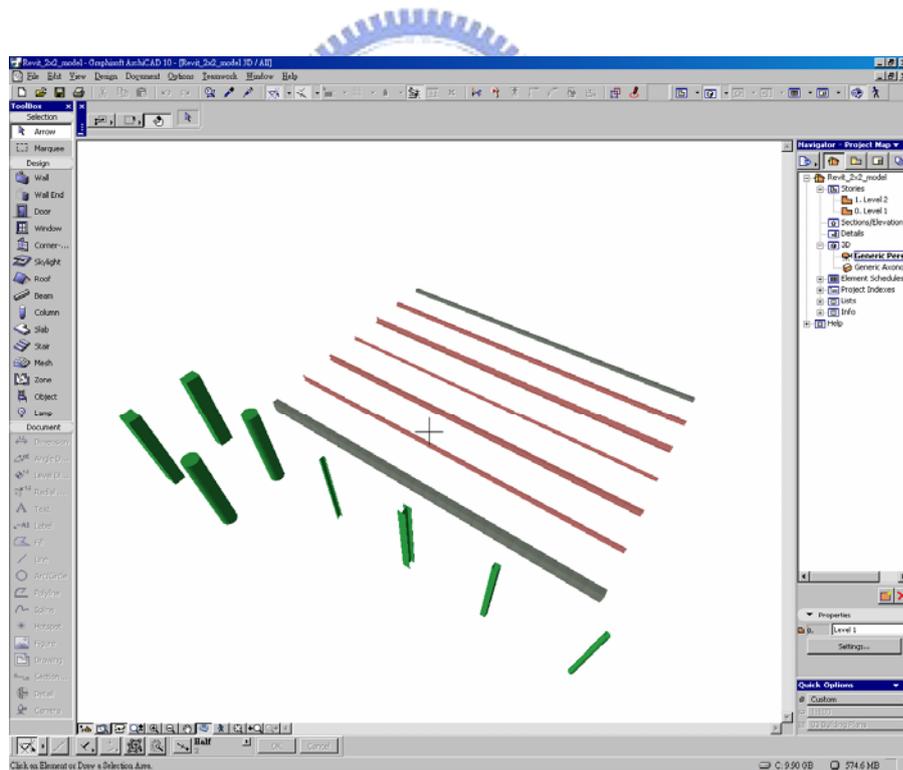


圖 4.22 Revit 建模匯出 IFC 再匯入 ArchiCAD 3D 視圖

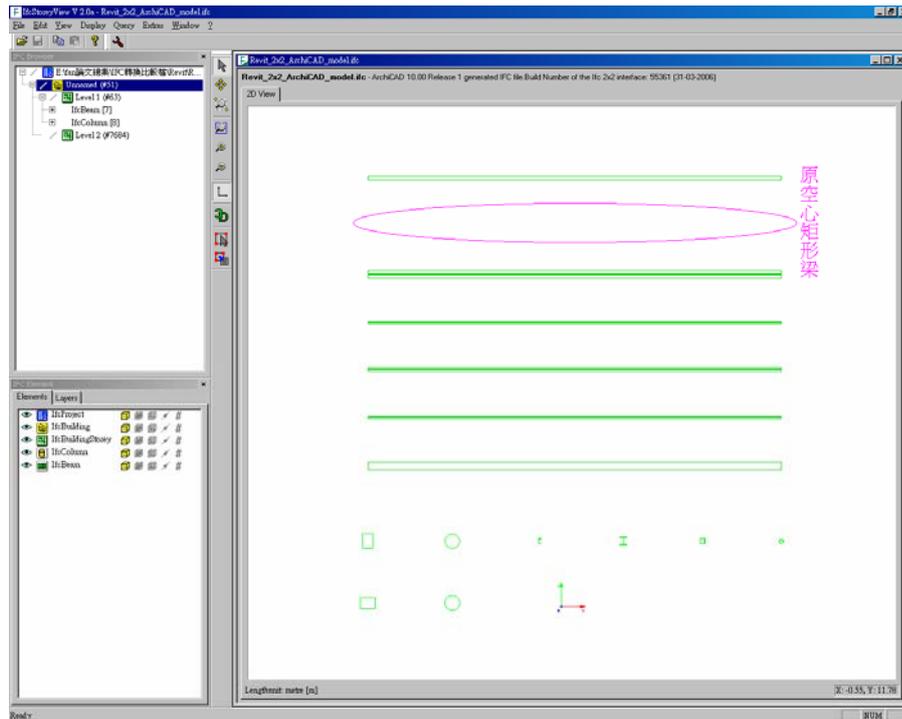


圖 4.23 Revit_ArchiCAD_model 以 IfcStoreyView 檢視平面圖

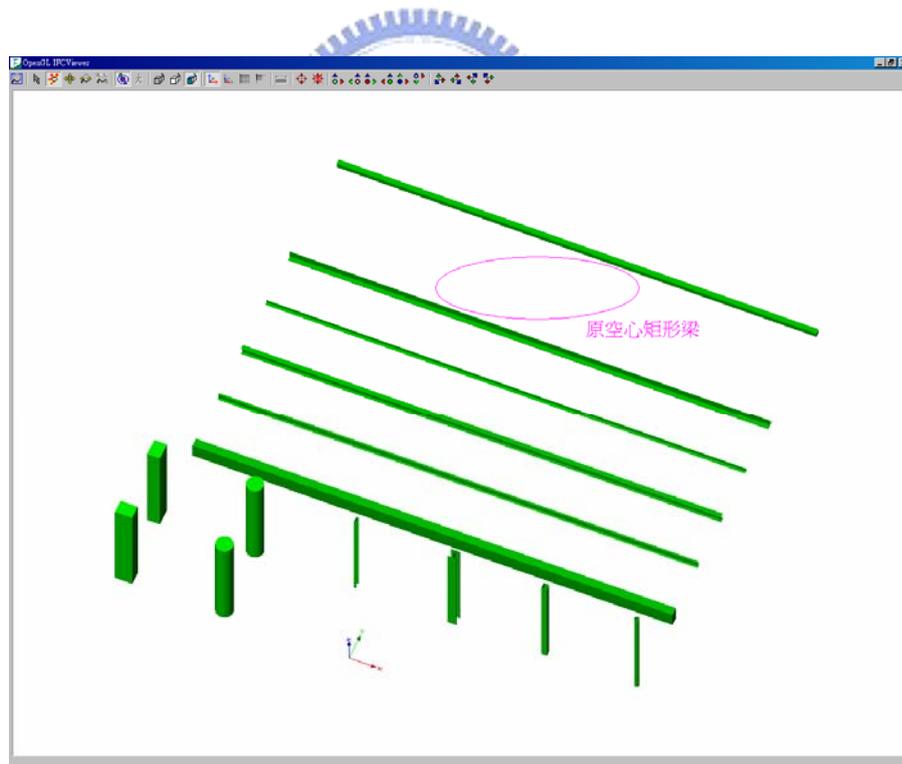


圖 4.24 Revit_ArchiCAD_model 以 IfcStoreyView 檢視 3D 視圖

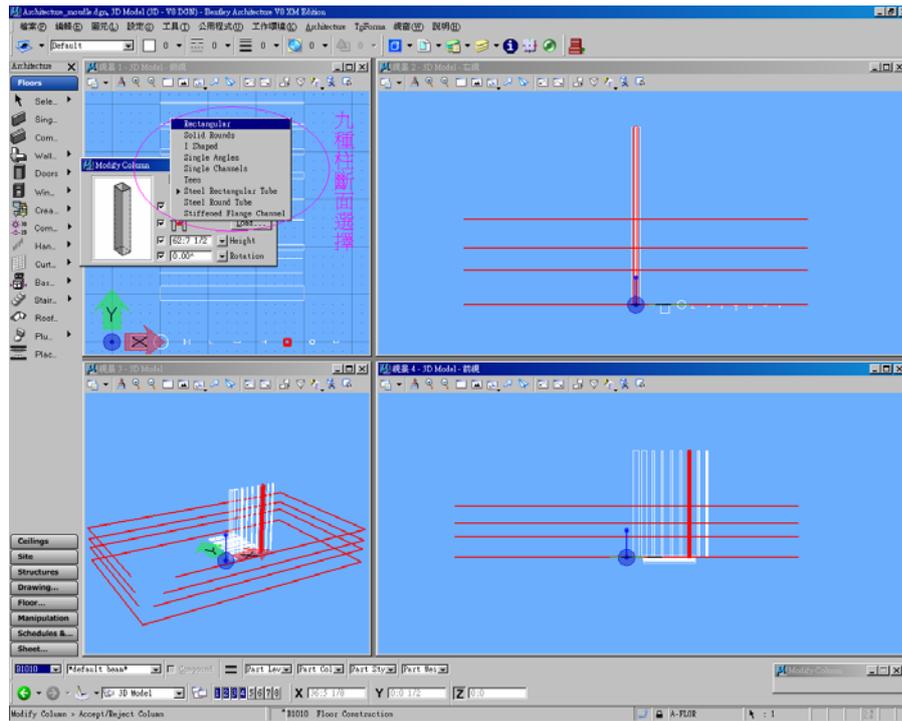


圖 4.25 Bentley 柱斷面形狀選擇圖

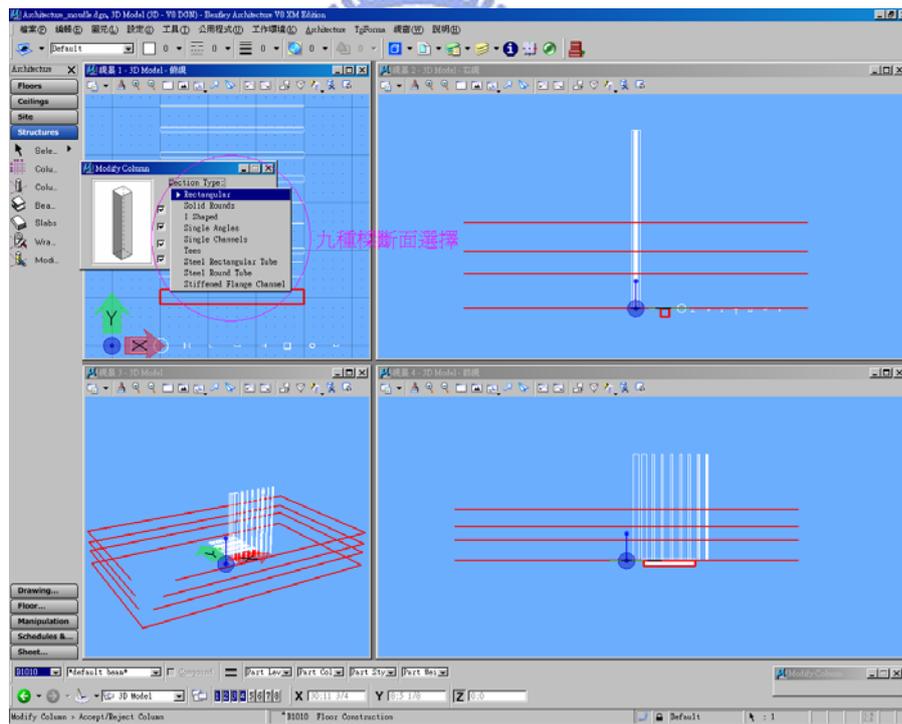


圖 4.26 Bentley 梁斷面形狀選擇圖

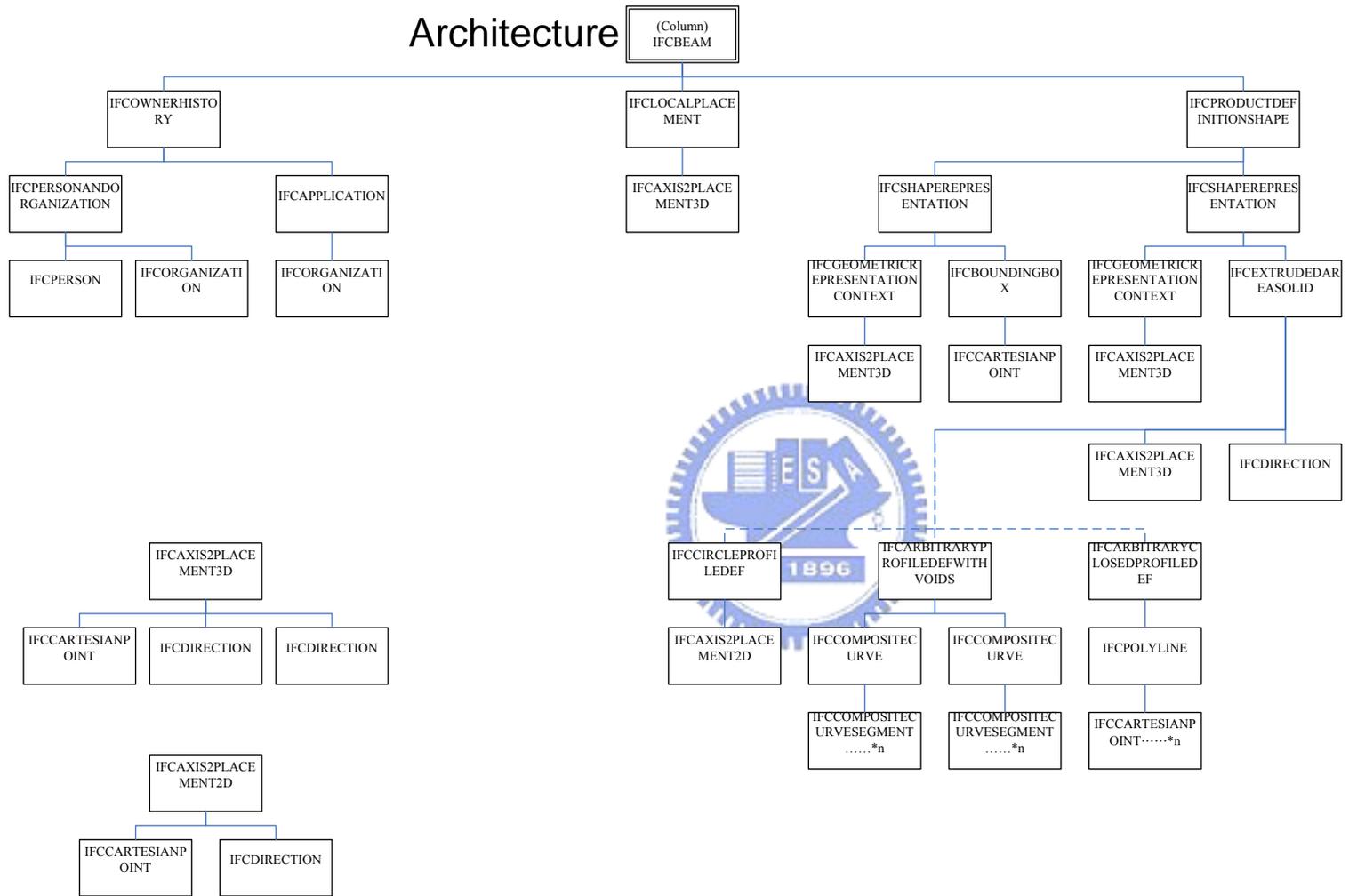


圖 4.27 Bentley Architecture 柱樹狀圖

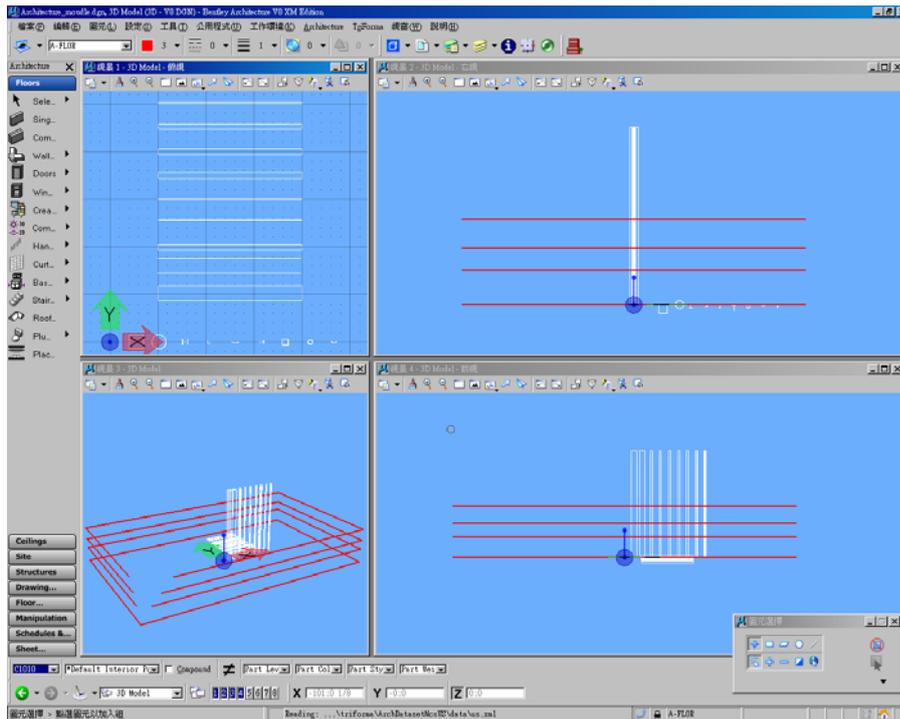


圖 4.29 Bentley 建模四視圖

```

Architecture_2x2_moudle.ifc - 記事本
檔案(F) 編輯(E) 格式(O) 檢視(V) 說明(H)

ISO-10303-21;
HEADER;
/* Generated by software containing ST-Developer
 * from STEP Tools, Inc. (www.steptools.com)
 */

FILE_DESCRIPTION(
/* description */ ('IFC2X_PLATFORM',
'MicroStation Triforma generated IFC File',
'Triforma IFC version 8.9.2.42', '*Comments*'),
/* implementation_level */ ('2;1');

FILE_NAME(
/* name */ 'Architecture_2x2_moudle',
/* time_stamp */ '2007-07-22T01:05:57+08:00',
/* author */ ('*Author*'),
/* organization */ ('*Organization*'),
/* preprocessor_version */ 'ST-DEVELOPER v8',
/* originating_system */ '*WinNT*',
/* authorisation */ '*Administrator*');

FILE_SCHEMA (('IFC2X2_FINAL'));
ENDSEC;

DATA;
#10=IFCAPPLICATION(#11,'Outdated','TriForma','TriForma IFC V1.0');
#11=IFCORGANIZATION($,'Bentley Systems Inc',$,$,$);
#12=IFCPERSON('*Id*','*SurName*','*FirstName*','*MiddleNames*'),('*Titles*'),('*SuffixTitles*),$,$);
#13=IFCPERSONANDORGANIZATION(#12,#11,$);
#14=IFCOWNERHISTORY(#13,#10,.READWRITE,.MODIFIED.,0,$,$,1185037557);
#15=IFCCIRCLE(#24,0.16256);
#16=IFCCIRCLE(#25,0.14986);
#17=IFCCIRCLE(#27,0.16256);
#18=IFCCIRCLE(#28,0.14986);
#19=IFCARBITRARYPROFILEDEFWITHVOIDS(.AREA.,$, #212, (#213));
#20=IFCARBITRARYPROFILEDEFWITHVOIDS(.AREA.,$, #215, (#216));
#21=IFCARBITRARYPROFILEDEFWITHVOIDS(.AREA.,$, #219, (#220));
#22=IFCARBITRARYPROFILEDEFWITHVOIDS(.AREA.,$, #217, (#218));
#23=IFCAXIS2PLACEMENT2D(#369, #575);
#24=IFCAXIS2PLACEMENT2D(#457, #611);
#25=IFCAXIS2PLACEMENT2D(#458, #612);
#26=IFCAXIS2PLACEMENT2D(#484, #628);
#27=IFCAXIS2PLACEMENT2D(#542, #654);
#28=IFCAXIS2PLACEMENT2D(#543, #655);
#29=IFCCIRCLEPROFILEDEF (.AREA., $, #23, 0.4572);
#30=IFCCIRCLEPROFILEDEF (.AREA., $, #26, 0.4572);
#31=IFCRELCONTAINEDINSPATIALSTRUCTURE ('1xxQdHgUH0B819dQ4UL6yR', #14, $, $, (#312,
#313, #314, #315, #316, #317, #318, #319, #320, #321, #322, #323, #324, #325, #326, #327,
#328, #329, #330));
#32=IFCCOMPLEXPROPERTY ('Color', $, 'Color', (#51, #52, #53));
#33=IFCCOMPLEXPROPERTY ('Color', $, 'Color', (#55, #56, #57));
#34=IFCCOMPLEXPROPERTY ('Color', $, 'Color', (#59, #60, #61));
#35=IFCCOMPLEXPROPERTY ('Color', $, 'Color', (#63, #64, #65));

```

圖 4.30 Bentley 建模匯出 IFC 檔以 Notepad 檢視

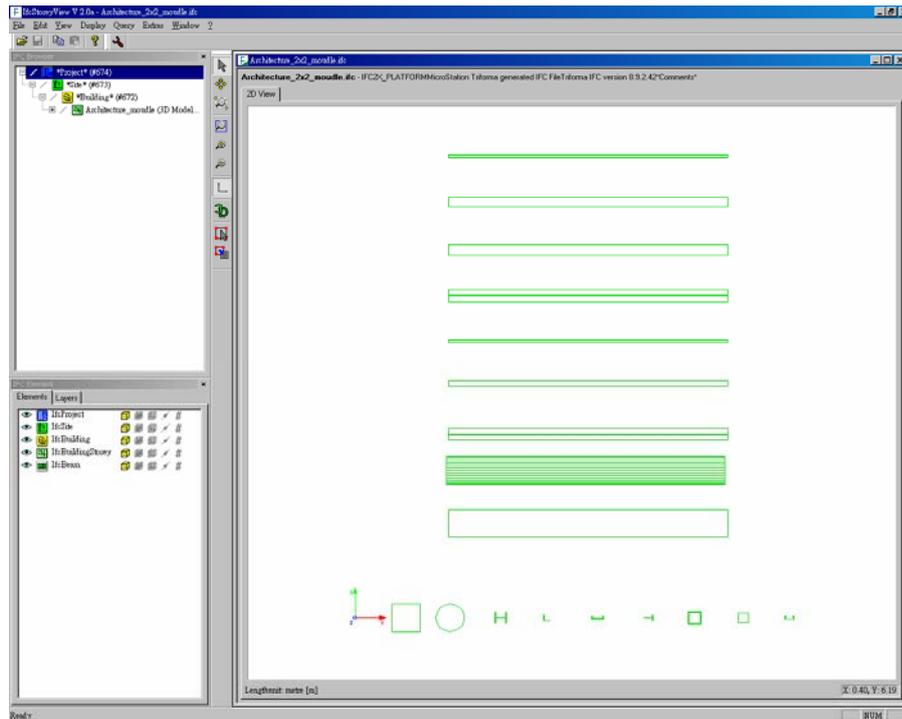


圖 4.31 Bentley 建模匯出 IFC 檔以 IfcStoreyView 檢視平面圖

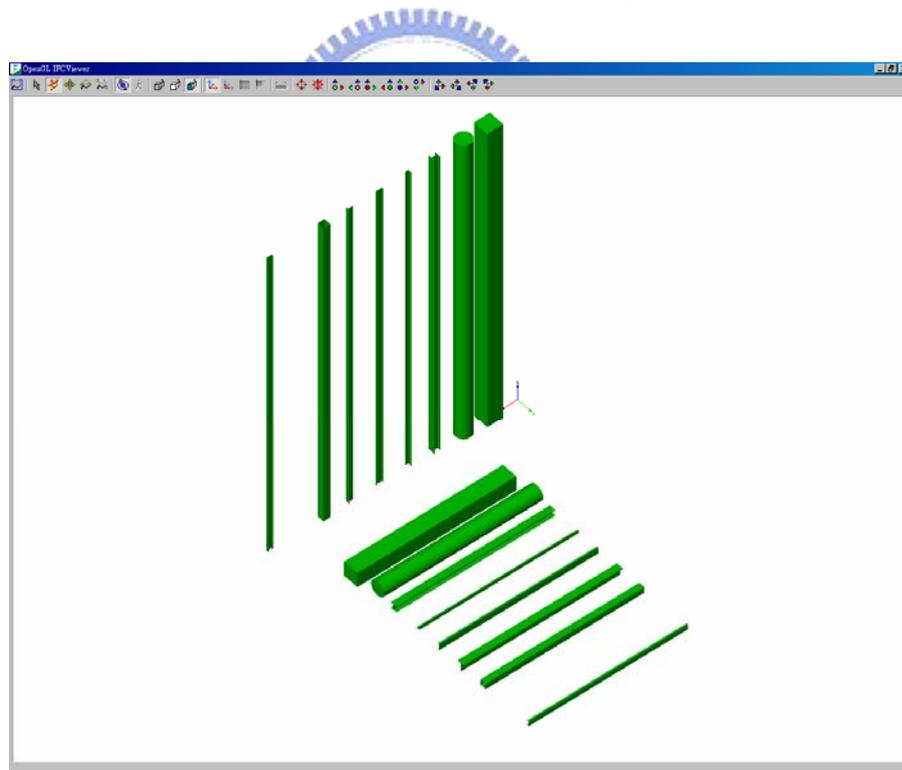


圖 4.32 Bentley 建模匯出 IFC 檔以 IfcStoreyView 檢視立體圖

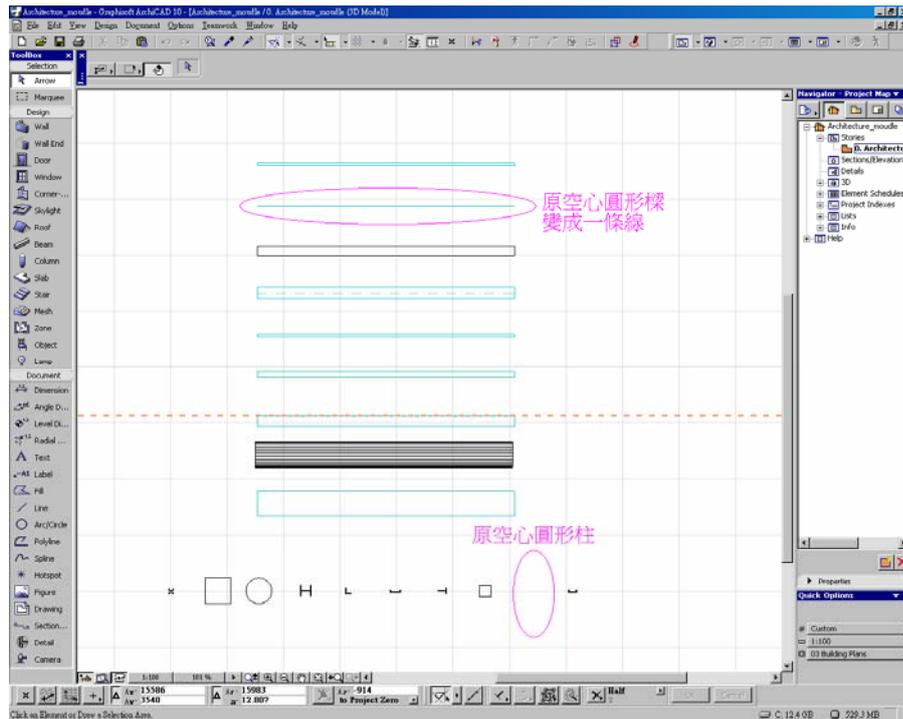


圖 4.33 Bentley 建模匯出 IFC 再匯入 ArchiCAD 平面圖

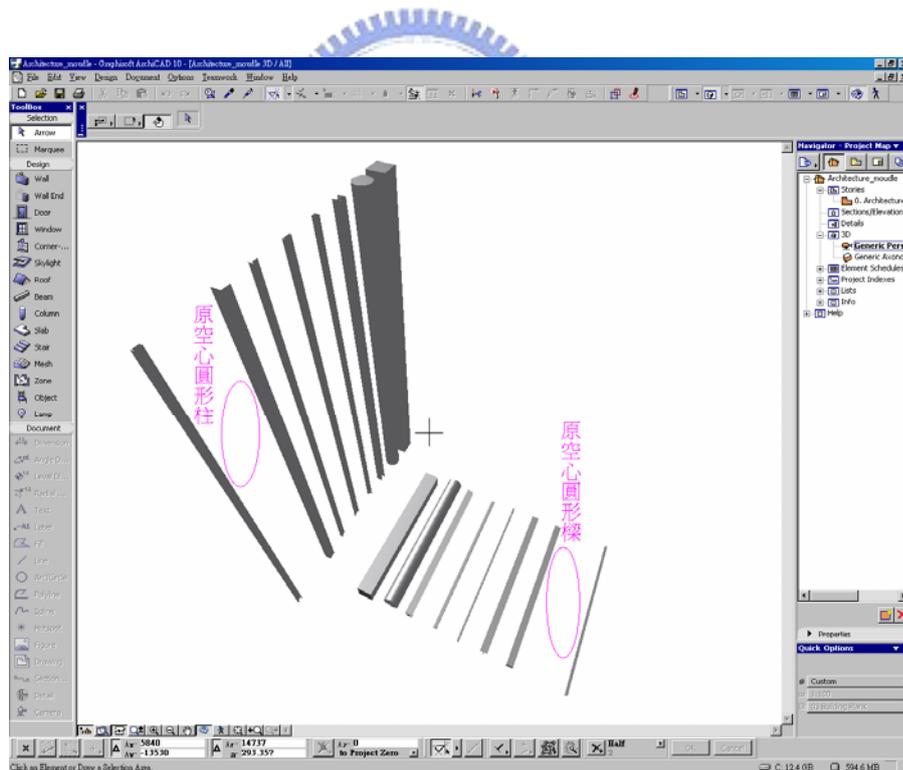


圖 4.34 Bentley 建模匯出 IFC 再匯入 ArchiCAD_3D 視圖

```
#1606= IFCBEAM('0Fw99CITf1Huo0HVz3QFU',#13,B1010--*default beam*',0, Architecture_moudle.dgn, 3D Model:3233',
...#13= IFCOWNERHISTORY(#12,#5,$,NOCHANGE,,$,$,1183672318);
...#1671= IFCLOCALPLACEMENT(#74,#1670);
...#1663= IFCPRODUCTDEFINITIONSHAPE($,$,(#1652,#1659));
...#1652= IFCSHAPE REPRESENTATION(#40,'Body','SweptSolid',(#1651));
...#1659= IFCSHAPE REPRESENTATION(#122,IAI,'BoundingBox',(#1658));
```

圖 4.35 Bentley→IFC→ArchiCAD→IFC 空心圓形梁

```
#1537= IFCBEAM('1xC7$2k6X8$edDrvt77jGH',#13,B1010--*default beam*',0, Architecture_moudle.dgn, 3D Model:3213',
...#13= IFCOWNERHISTORY(#12,#5,$,NOCHANGE,,$,$,1183672318);
...#1574= IFCLOCALPLACEMENT(#74,#1573);
...#1565= IFCPRODUCTDEFINITIONSHAPE($,$,(#1561));
...#1561= IFCSHAPE REPRESENTATION(#40,'Body','SweptSolid',(#1560));
```

圖 4.36 Bentley→IFC→ArchiCAD→IFC 其他梁柱

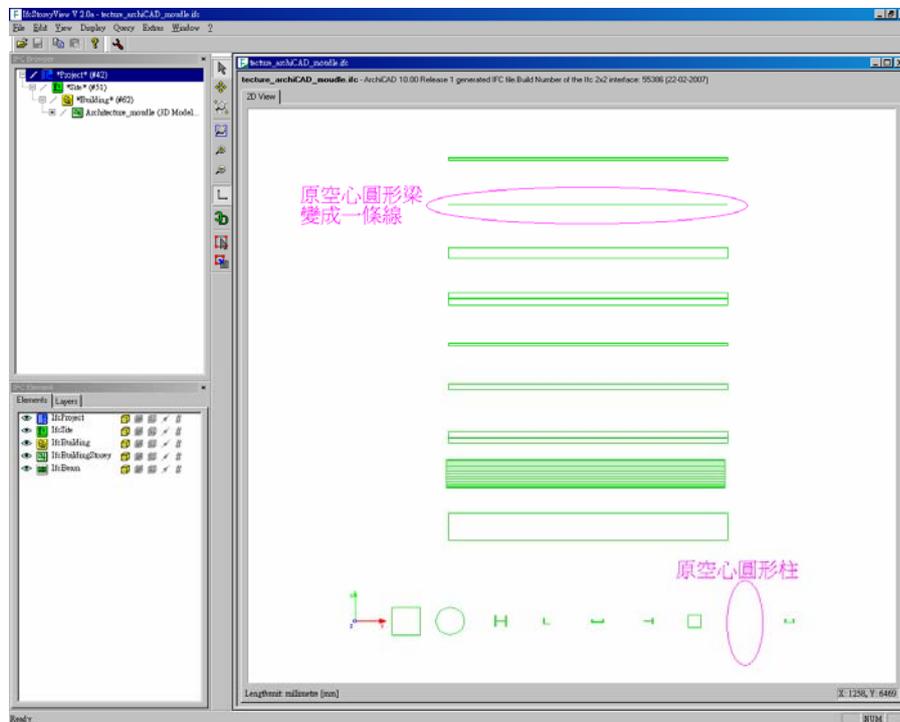


圖 4.37 Bentley_ArchiCAD_model 以 IfcStoreyView 檢視平面圖

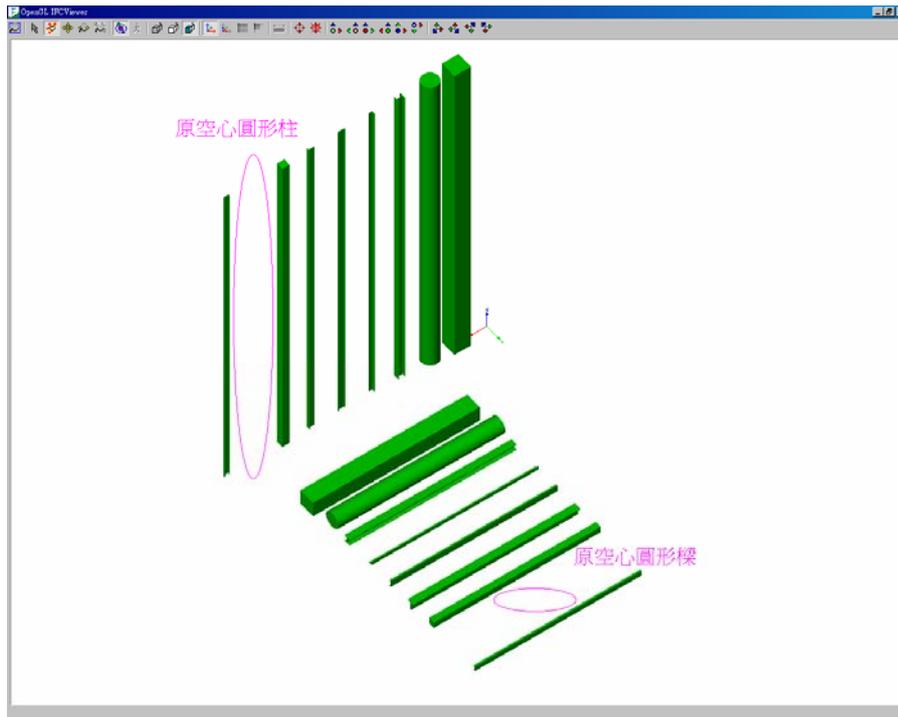


圖 4.38 Bentley_ArchiCAD_model 以 IfcStoreyView 檢視 3D 視圖

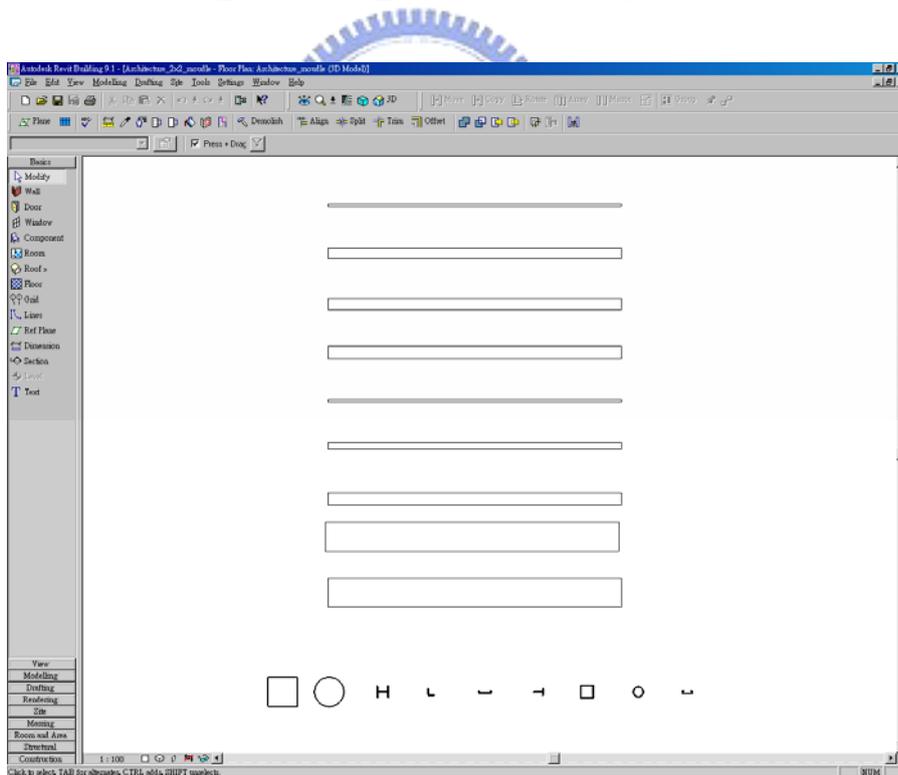


圖 4.39 Bentley 建模匯出 IFC 再匯入 Revit 平面圖

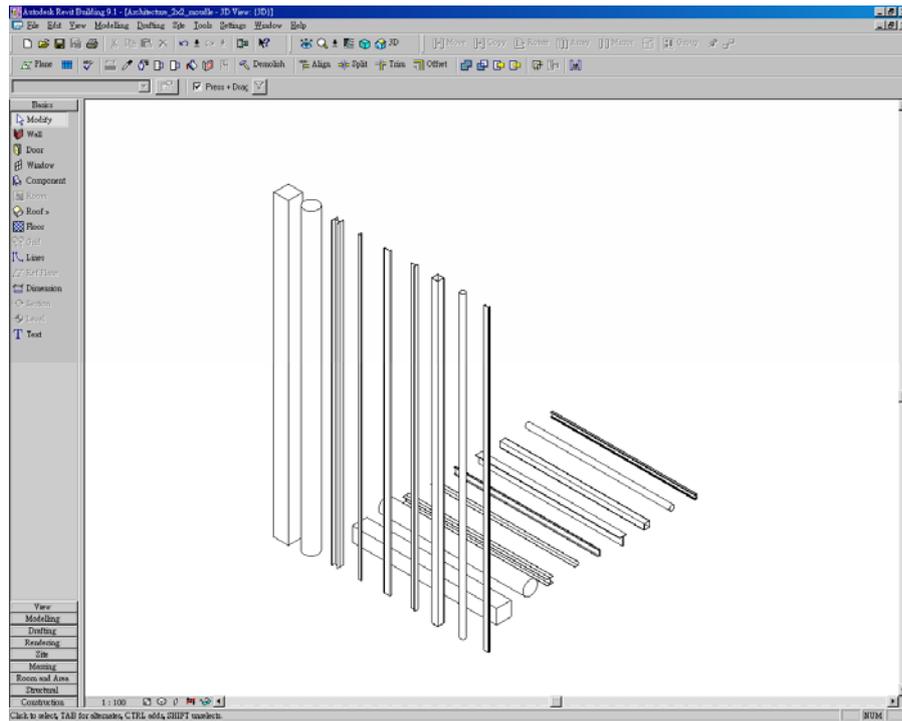


圖 4.40 Bentley 建模匯出 IFC 再匯入 Revit_3D 視圖

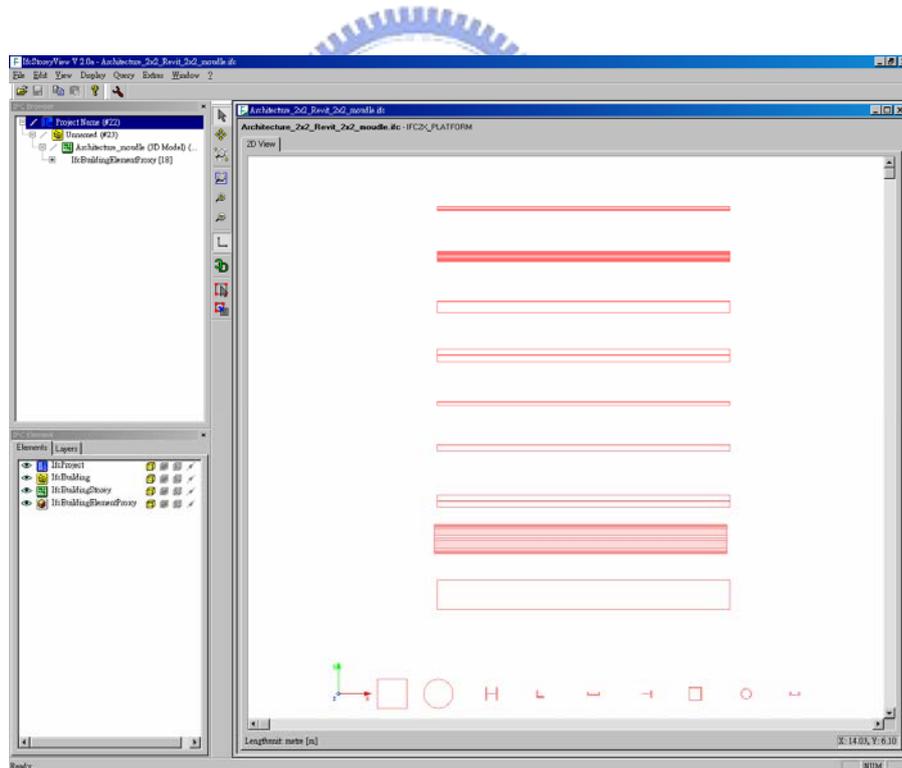


圖 4.41 Bentley_Revit_model 以 IfcStoreyView 檢視平面圖

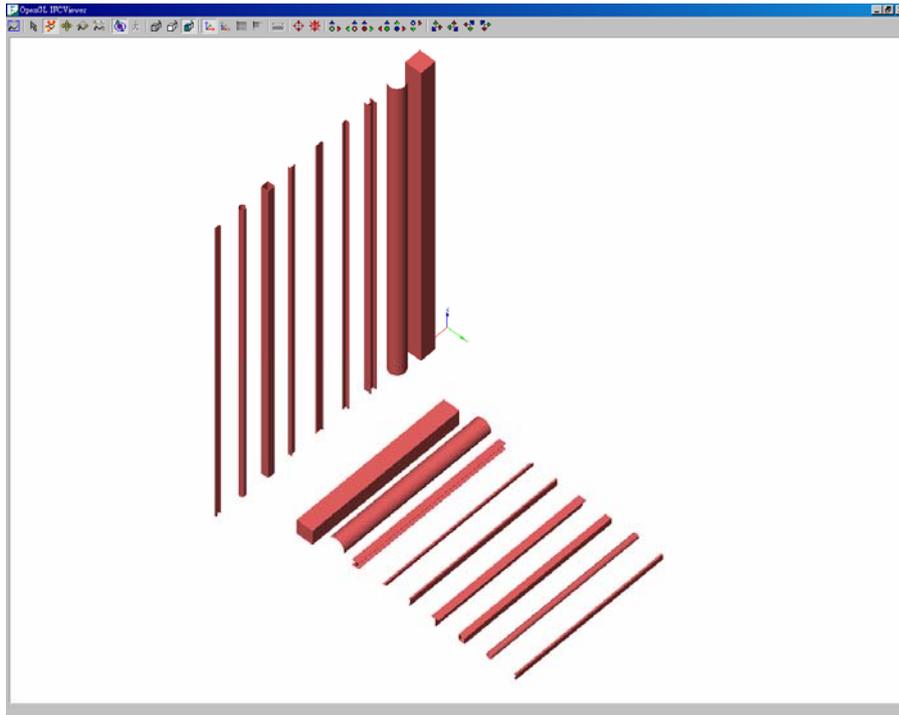


圖 4.42 Bentley_Revit_model 以 IfcStoreyView 檢視 3D 視圖

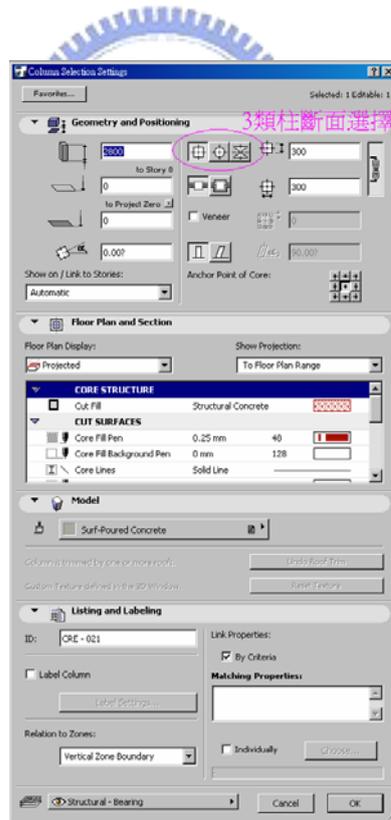


圖 4.43 ArchiCAD 柱断面形狀選擇圖

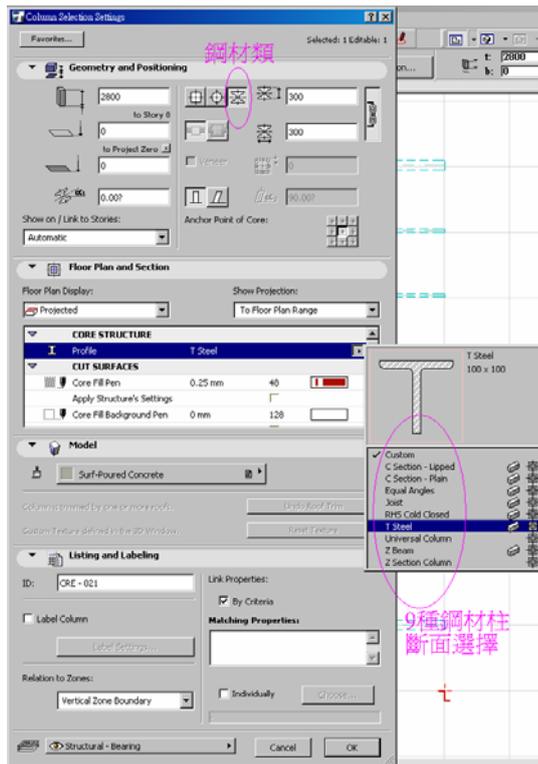


圖 4.44 ArchiCAD 柱之鋼材斷面形狀選擇圖

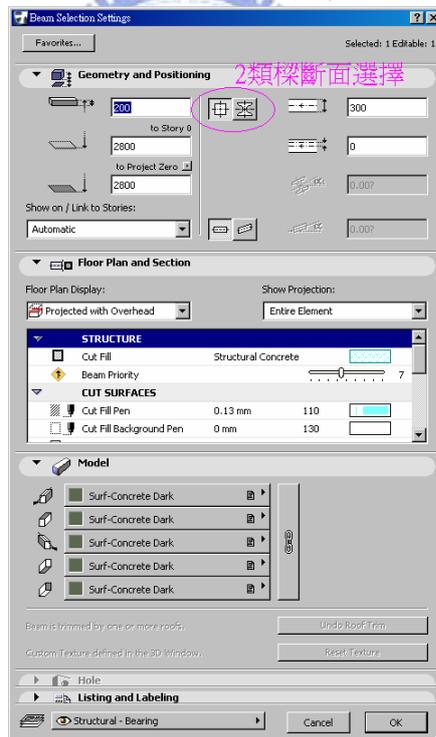


圖 4.45 ArchiCAD 梁斷面形狀選擇圖

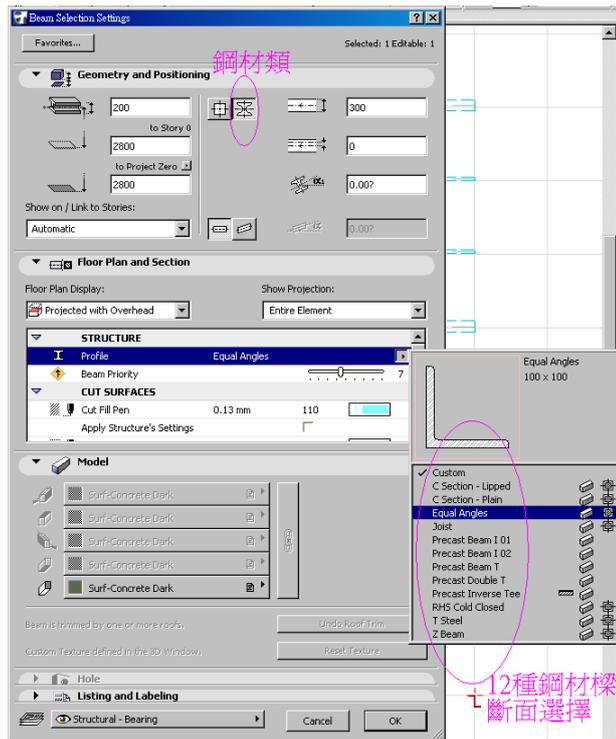


圖 4.46 ArchiCAD 梁之鋼材斷面形狀選擇圖



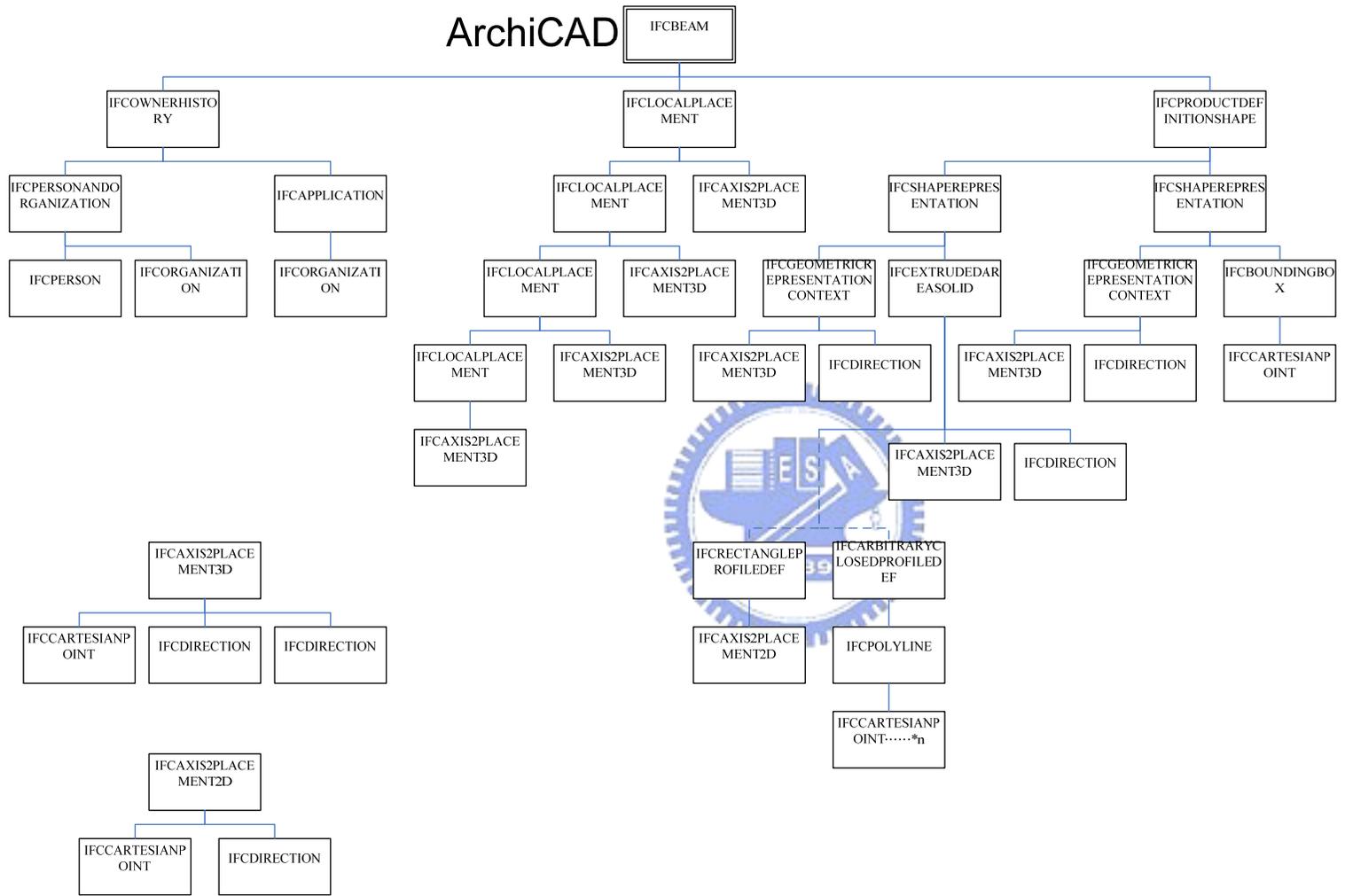


圖 4.48 Graphisoft ArchiCAD 梁樹狀圖

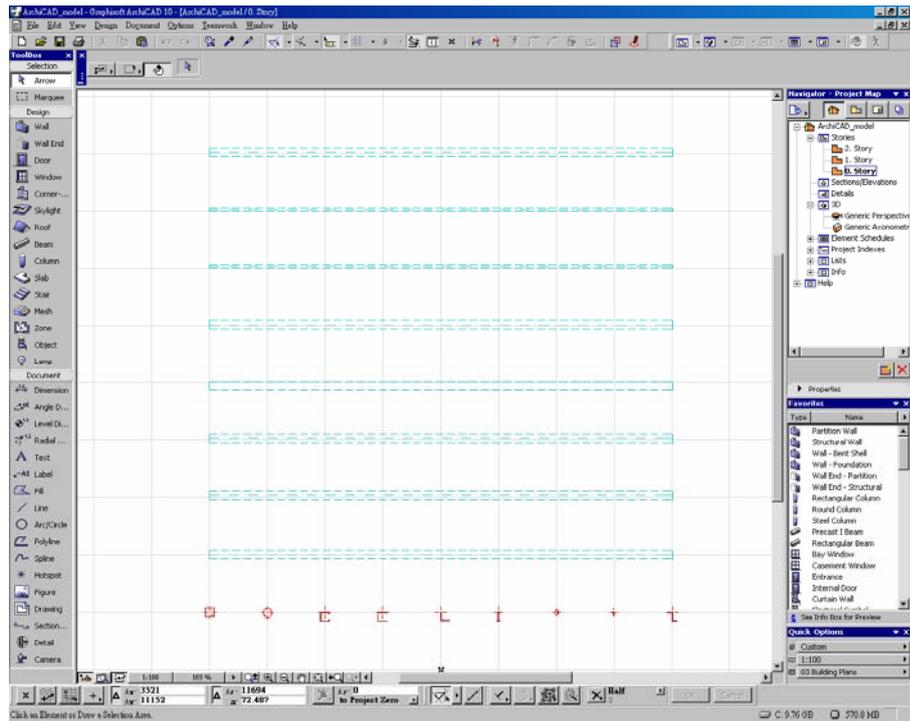


圖 4.49 ArchiCAD 建模平面圖

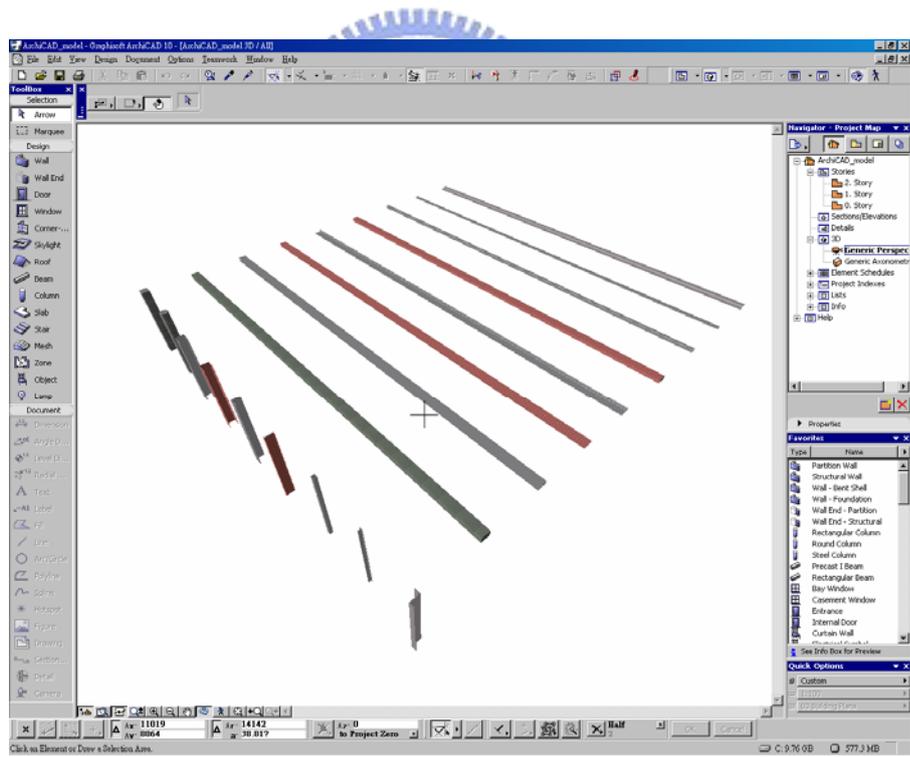


圖 4.50 ArchiCAD 建模 3D 視圖

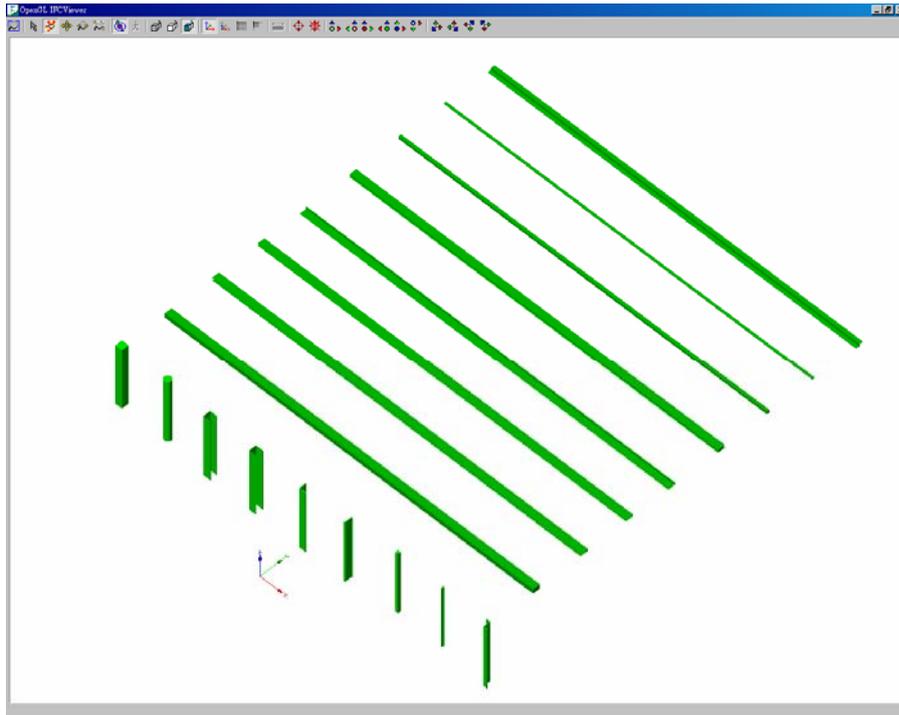


圖 4.53 ArchiCAD 建模匯出 IFC 檔以 IfcStoreyView 檢視立體圖

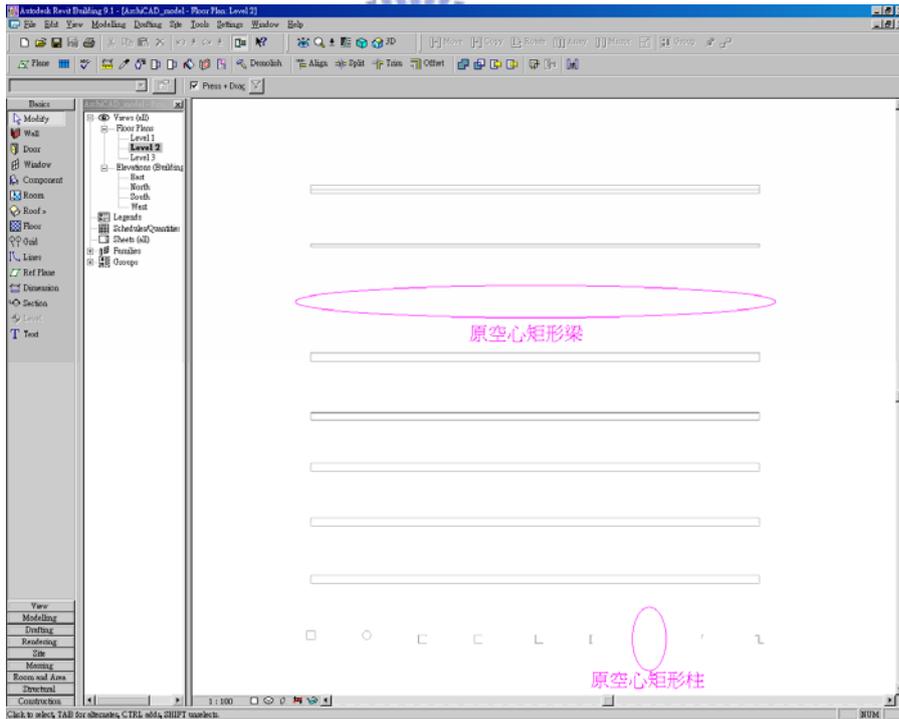


圖 4.54 ArchiCAD 建模匯出 IFC 再匯入 Revit 平面圖

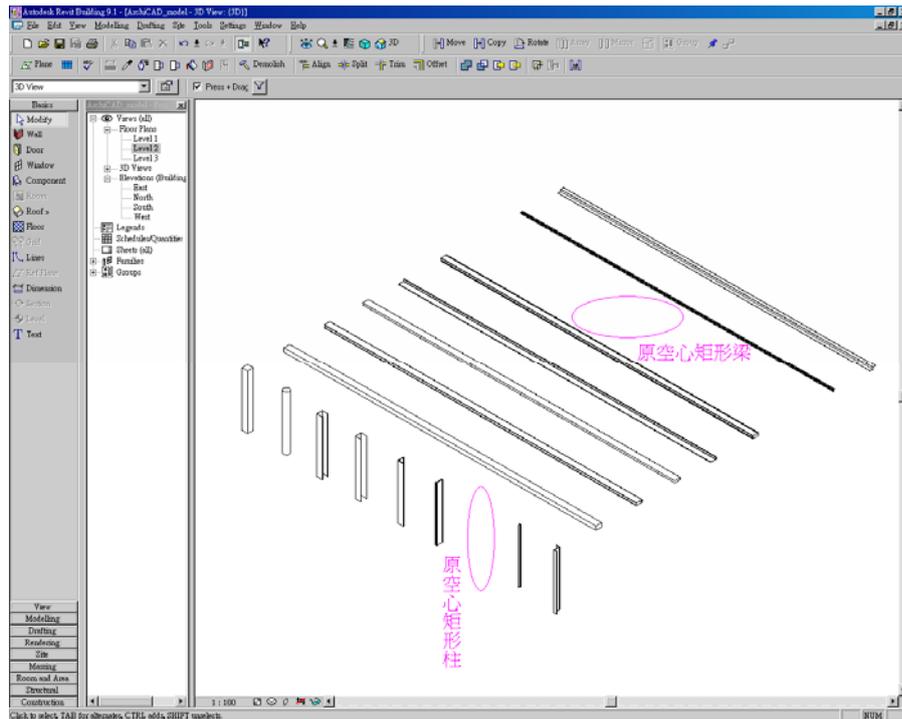


圖 4.55 ArchiCAD 建模匯出 IFC 再匯入 Revit_3D 視圖

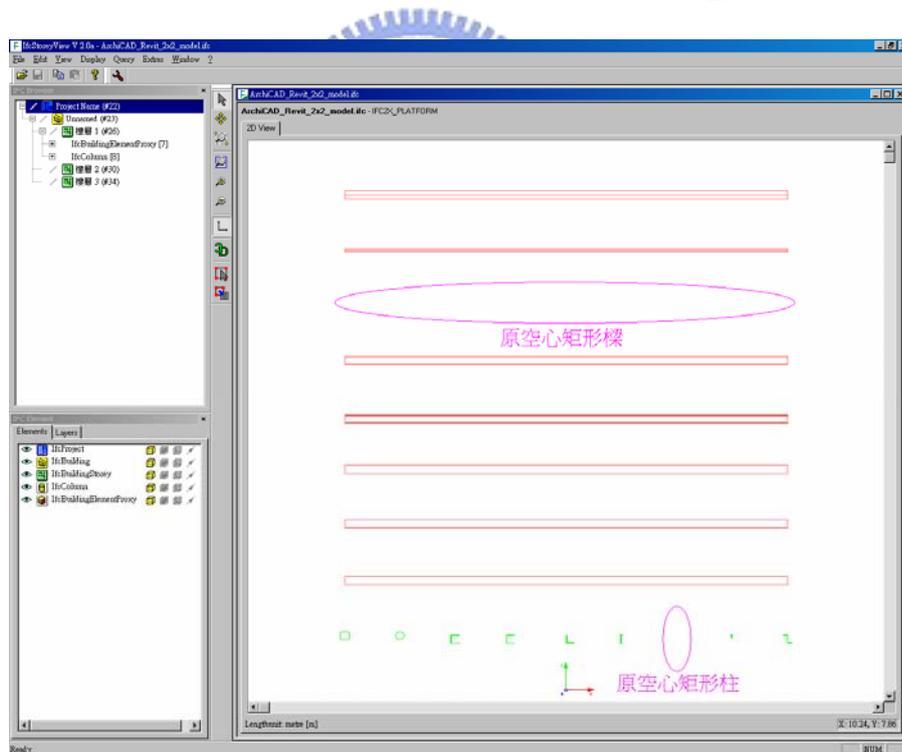


圖 4.56 ArchiCAD_Revit_model 以 IfcStoreyView 檢視平面圖

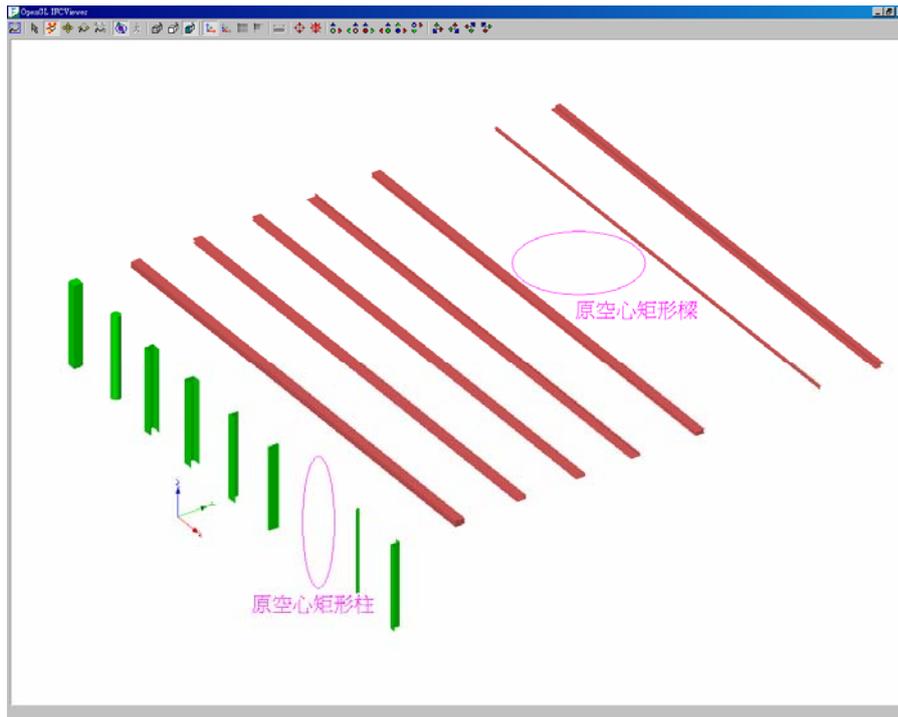


圖 4.57 ArchiCAD_Revit_model 以 IfcStoreyView 檢視 3D 視圖

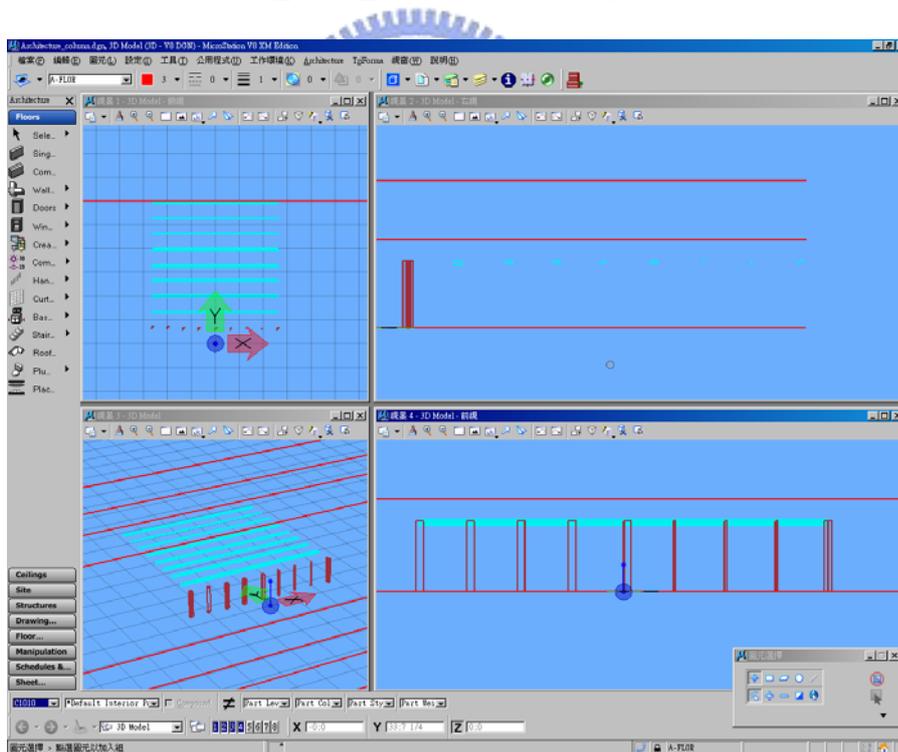


圖 4.58 ArchiCAD 建模匯出 IFC 再匯入 Bentley 四視圖

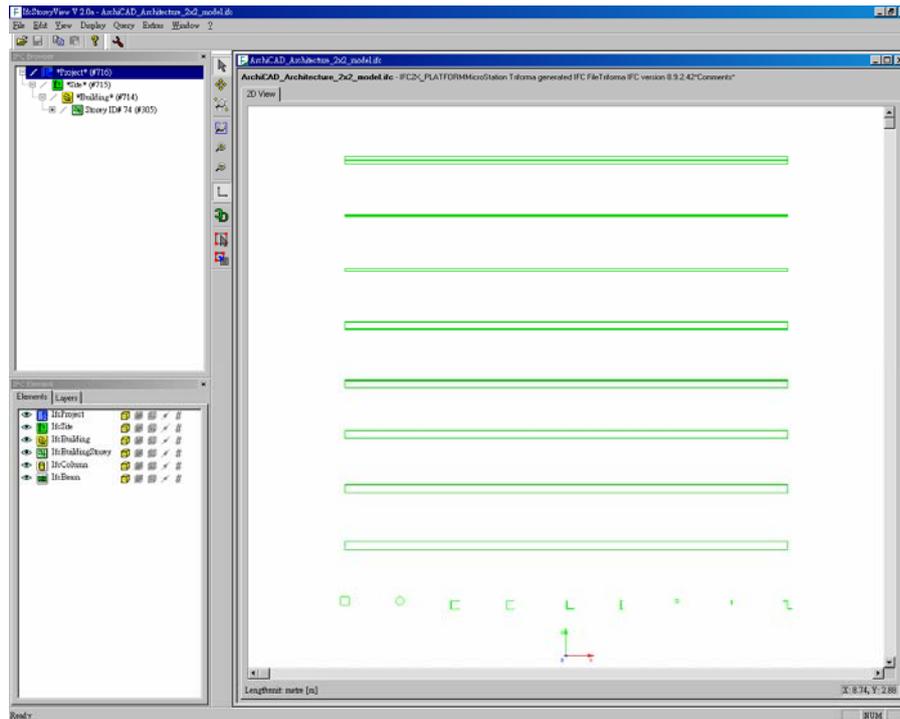


圖 4.59 ArchiCAD_Bentley_model 以 IfcStoreyView 檢視平面圖

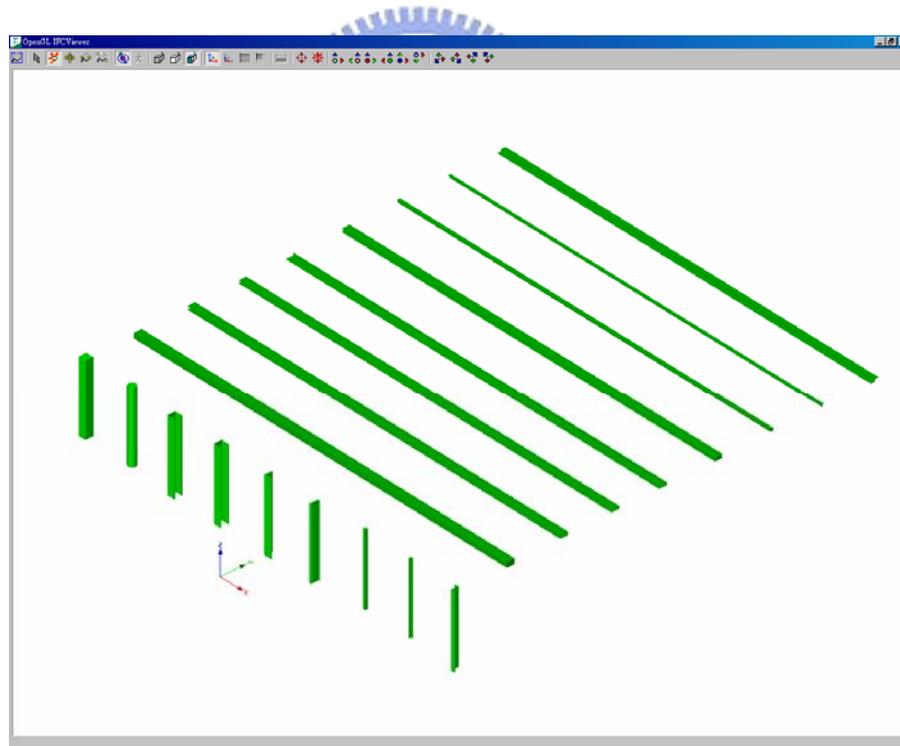


圖 4.60 ArchiCAD_Bentley_model 以 IfcStoreyView 檢視 3D 視圖

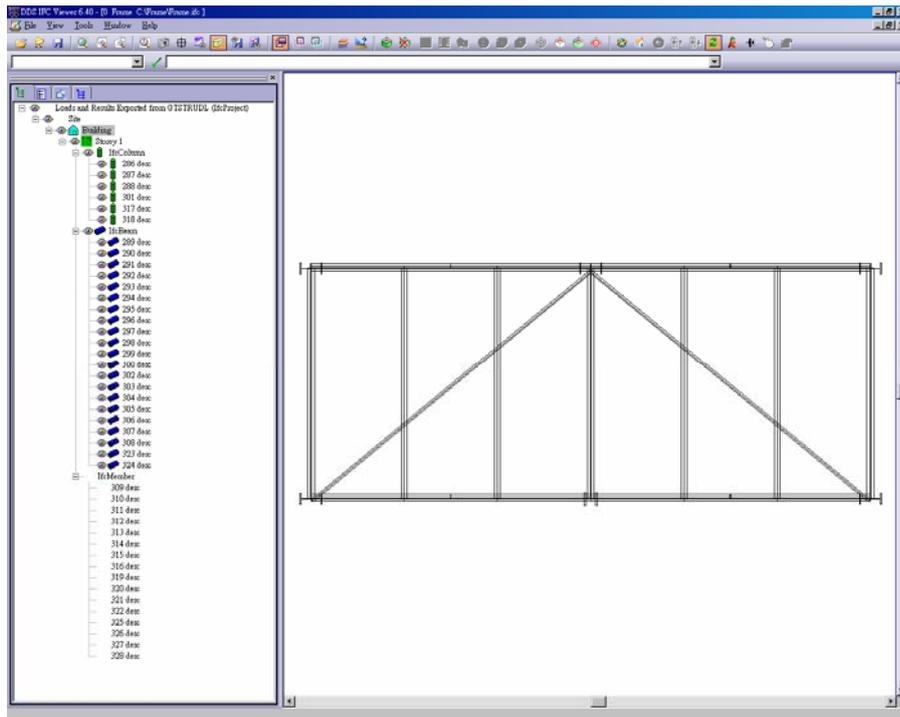


圖 4.61 Frame 平面圖

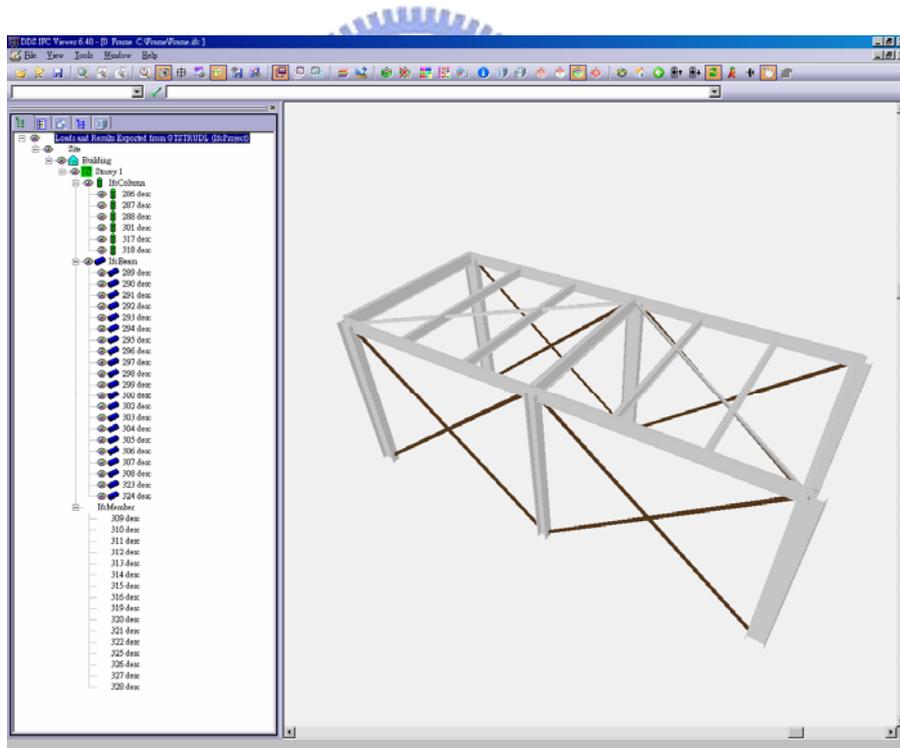


圖 4.62 Frame_3D 視圖

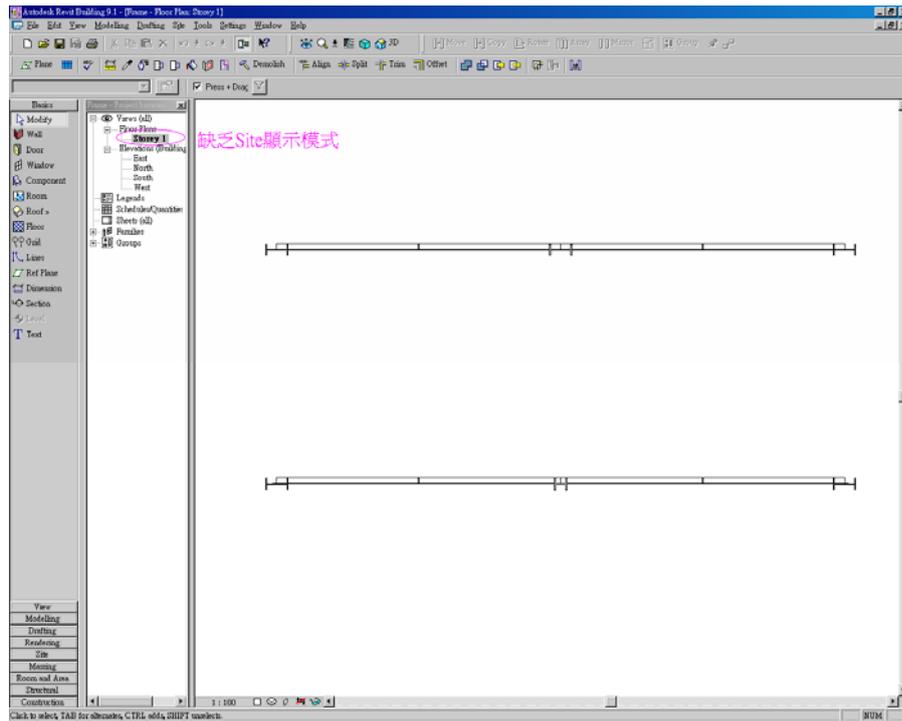


圖 4.63 Frame 匯入 Revit 平面圖

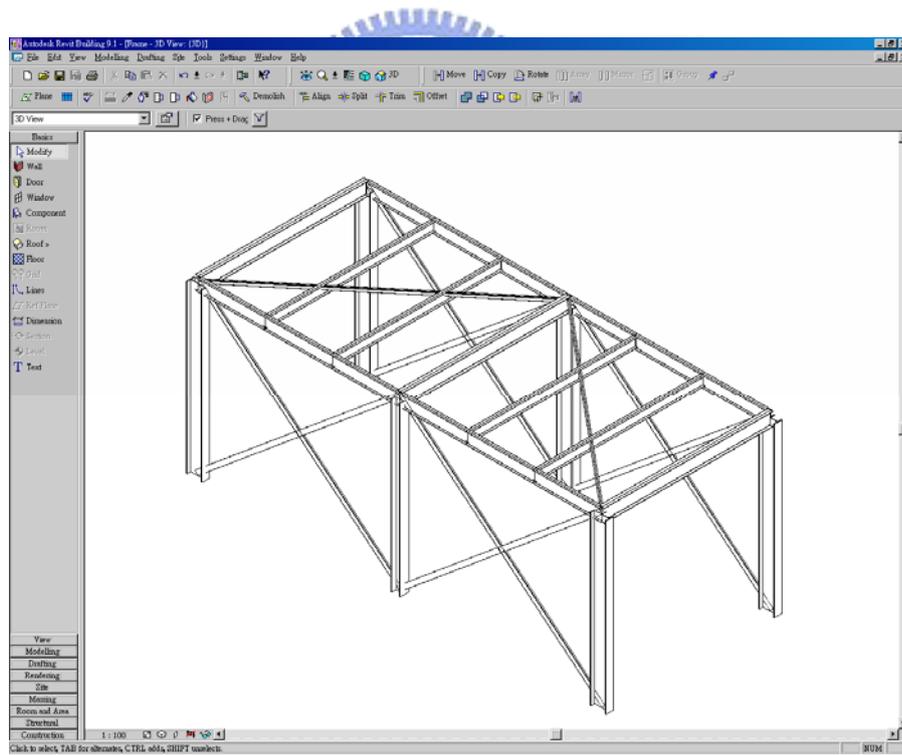


圖 4.64 Frame 匯入 Revit_3D 視圖

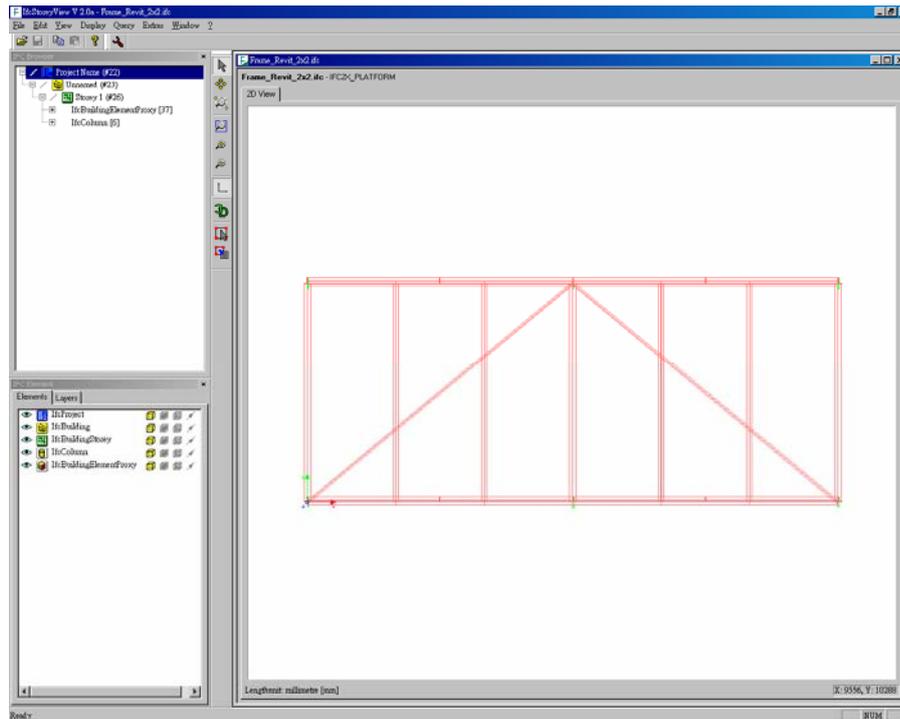


圖 4.65 Frame_Revit 以 IfcStoreyView 平面圖

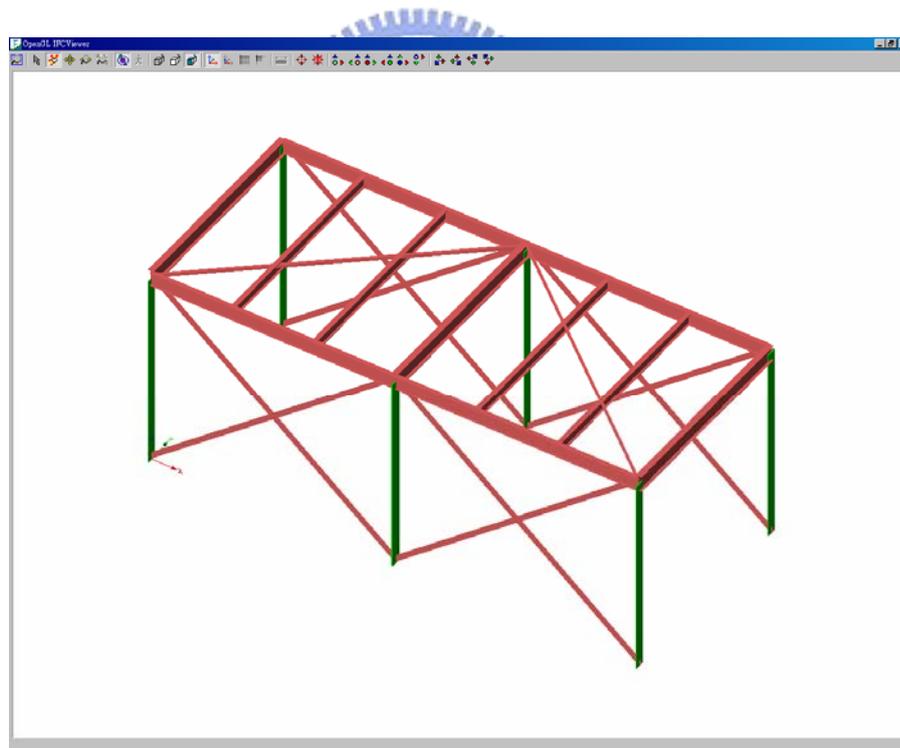


圖 4.66 Frame_Revit 以 IfcStoreyView_3D 視圖

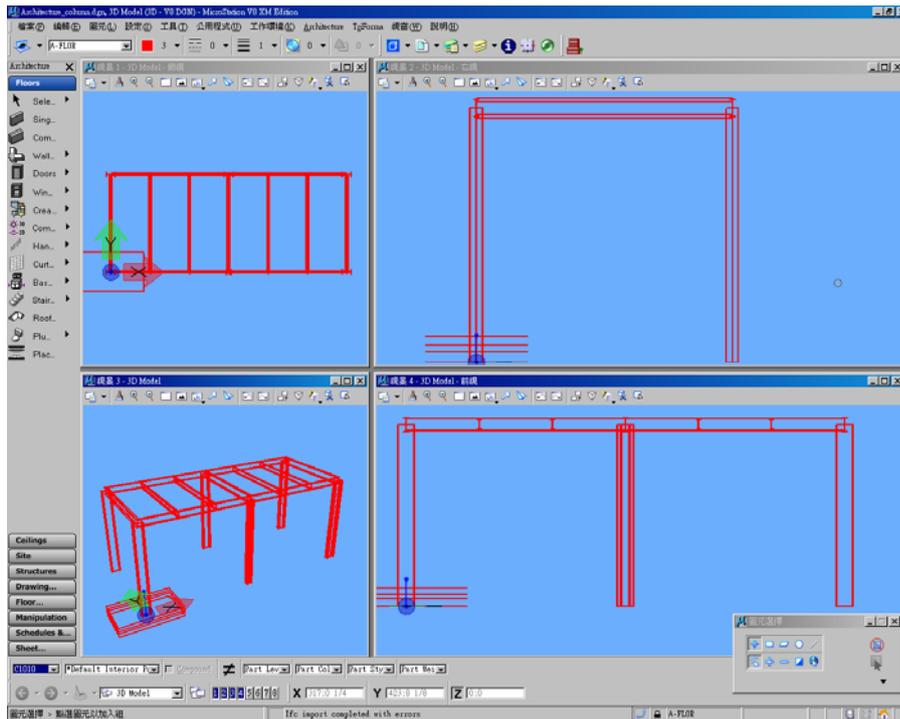


圖 4.67 Frame 匯入 Bentley 四視圖

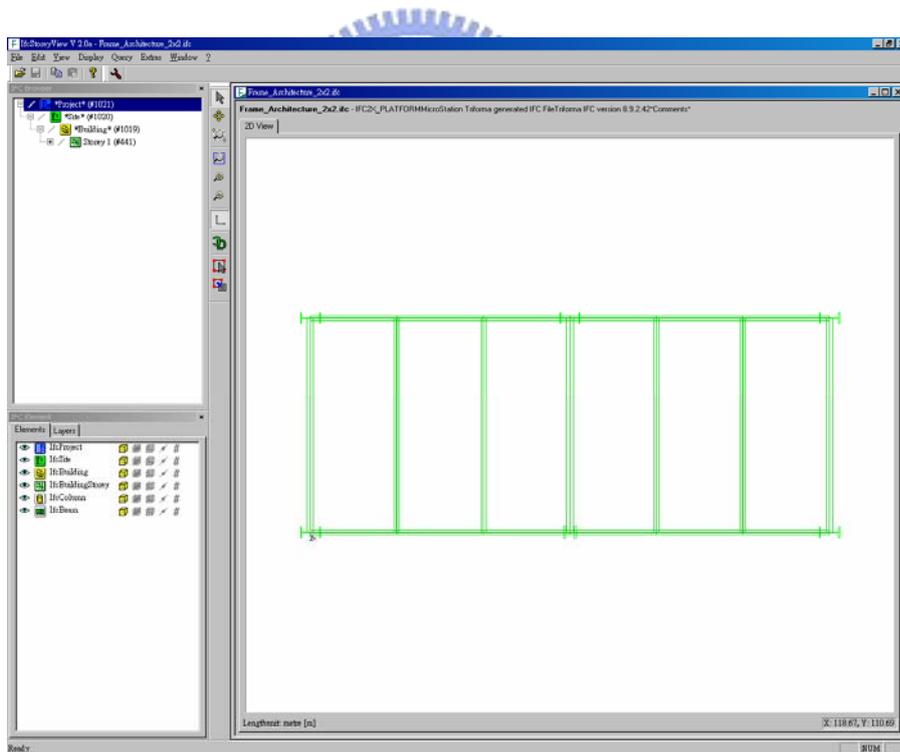


圖 4.68 Frame_Bentley 以 IfcStoreyView 平面圖

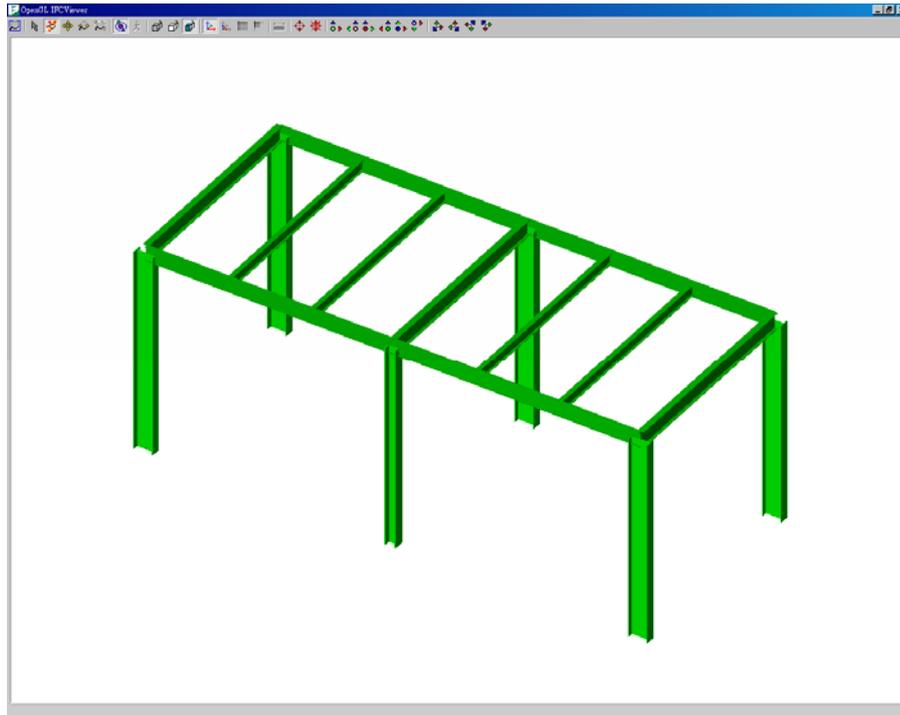


圖 4.69 Frame_Bentley 以 IfcStoreyView_3D 視圖

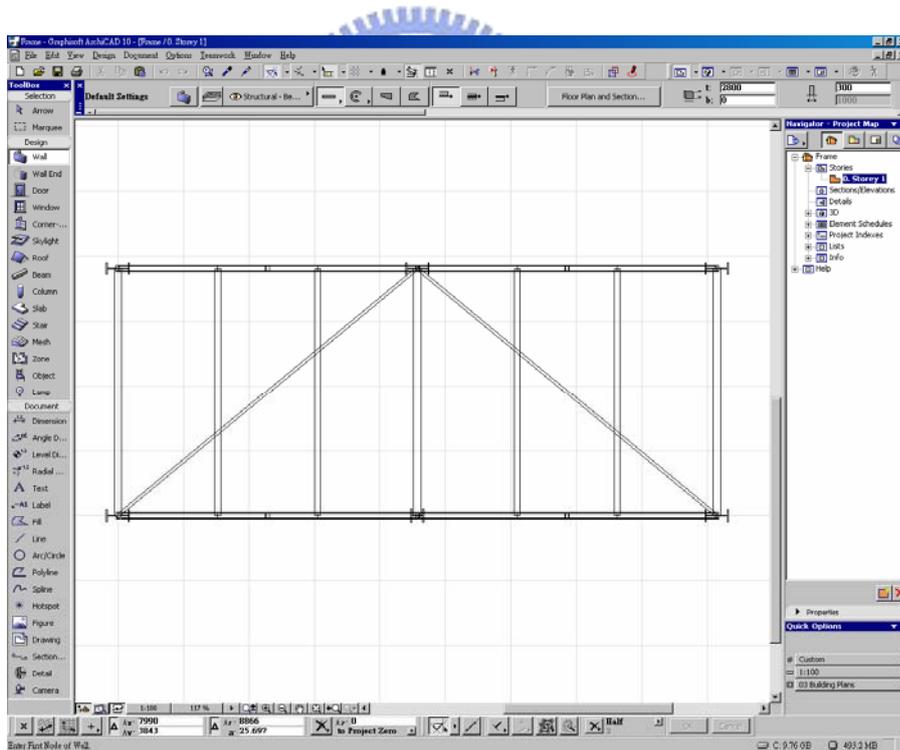


圖 4.70 Frame 匯入 ArchiCAD 平面圖

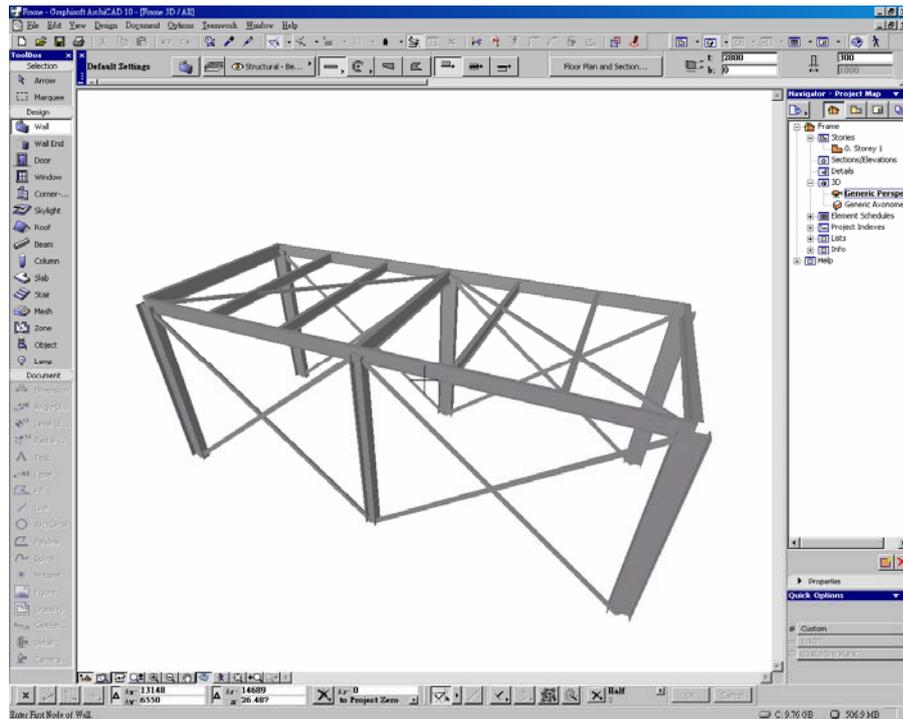


圖 4.71 Frame 匯入 ArchiCAD_3D 視圖

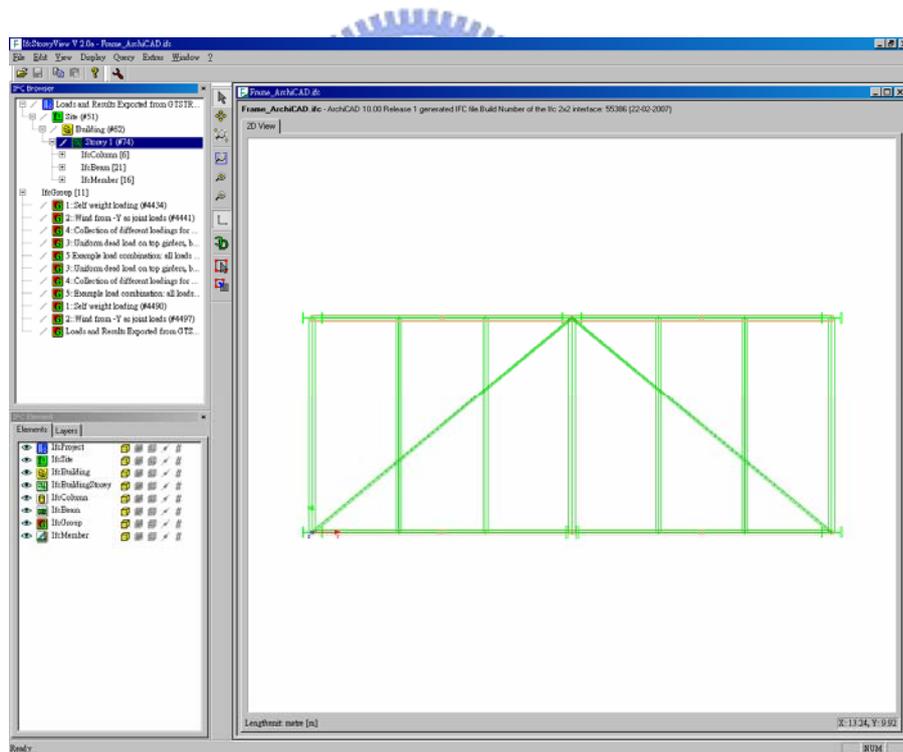


圖 4.72 Frame_ArchiCAD 以 IfcStoreyView 平面圖

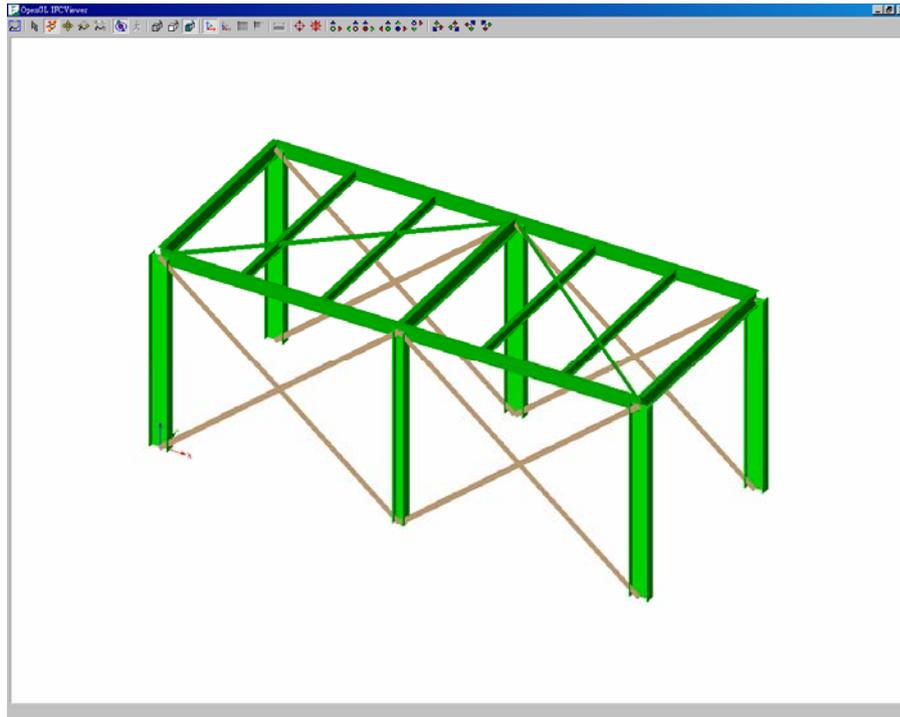


圖 4.73 Frame_ArchiCAD 以 IfcStoreyView_3D 視圖

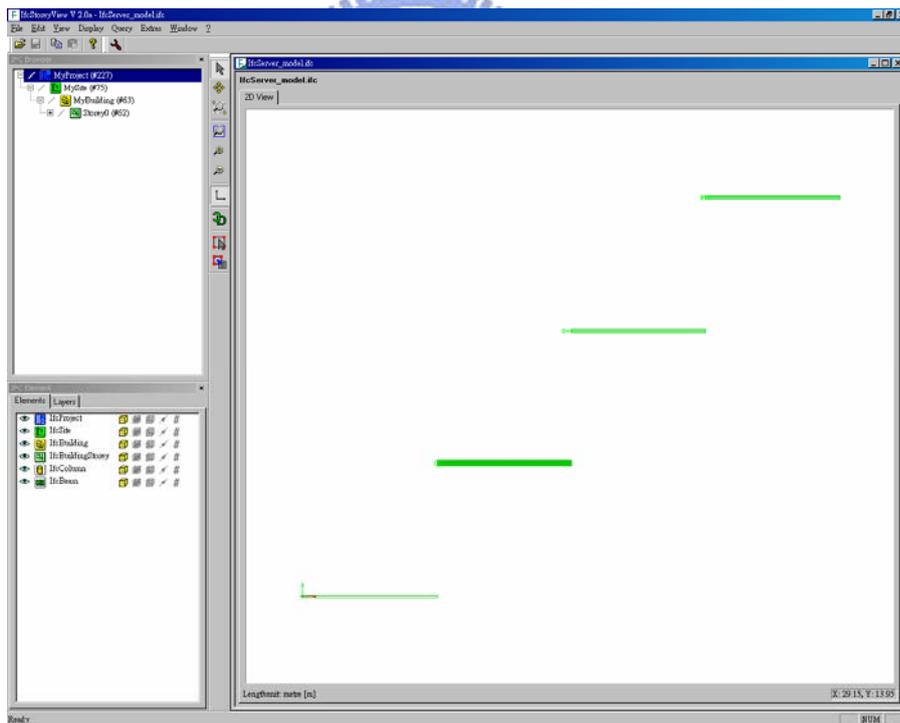


圖 4.74 IFCServer_model 平面圖

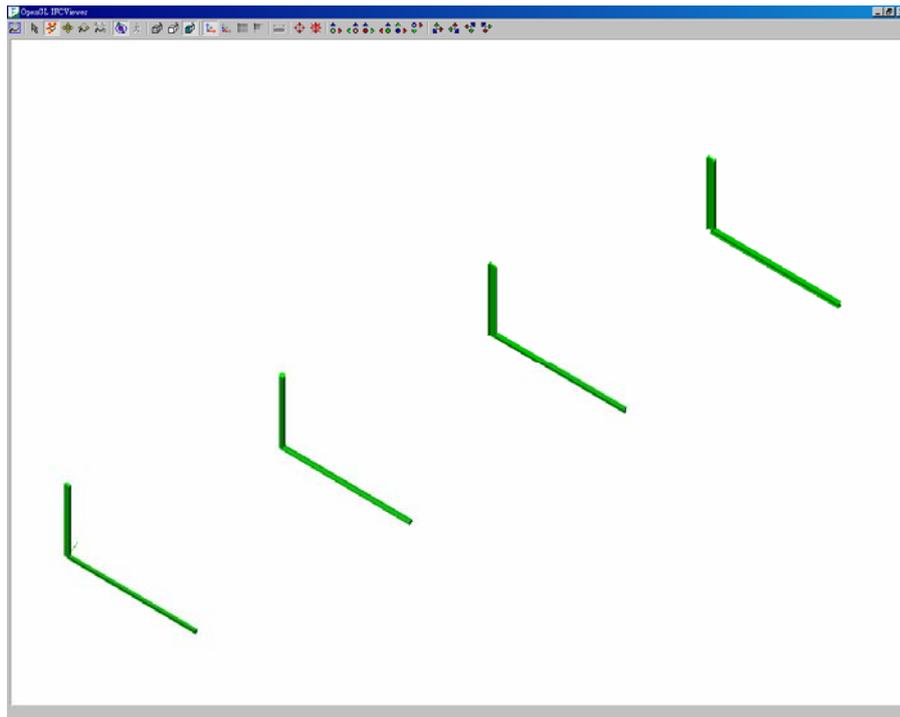


圖 4.75 IFCServer_model_3D 視圖

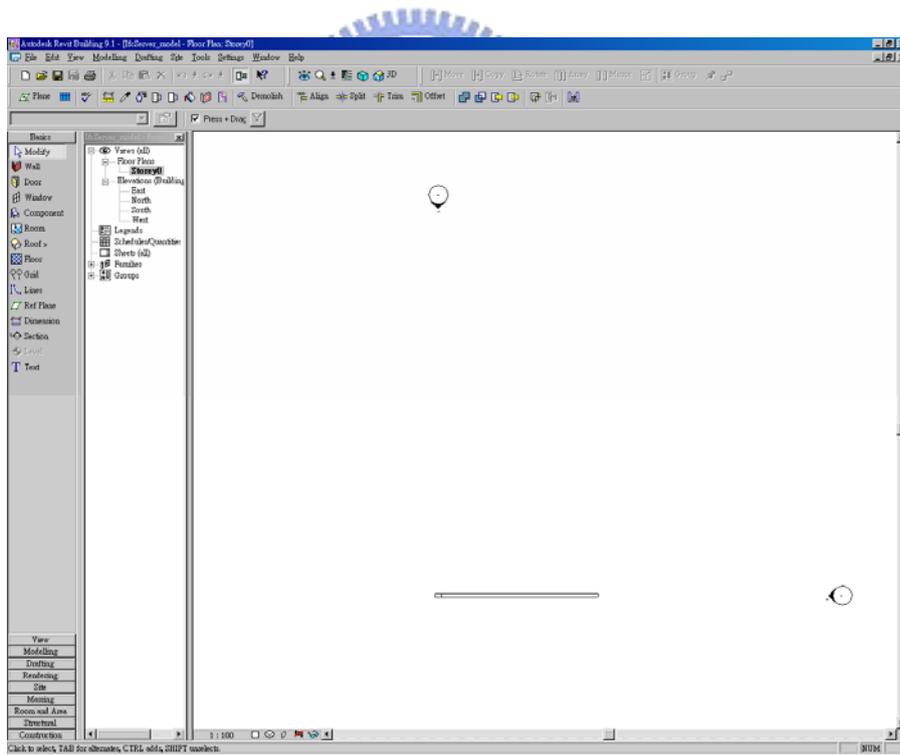


圖 4.76 IFCServer_model 匯入 Revit 平面圖

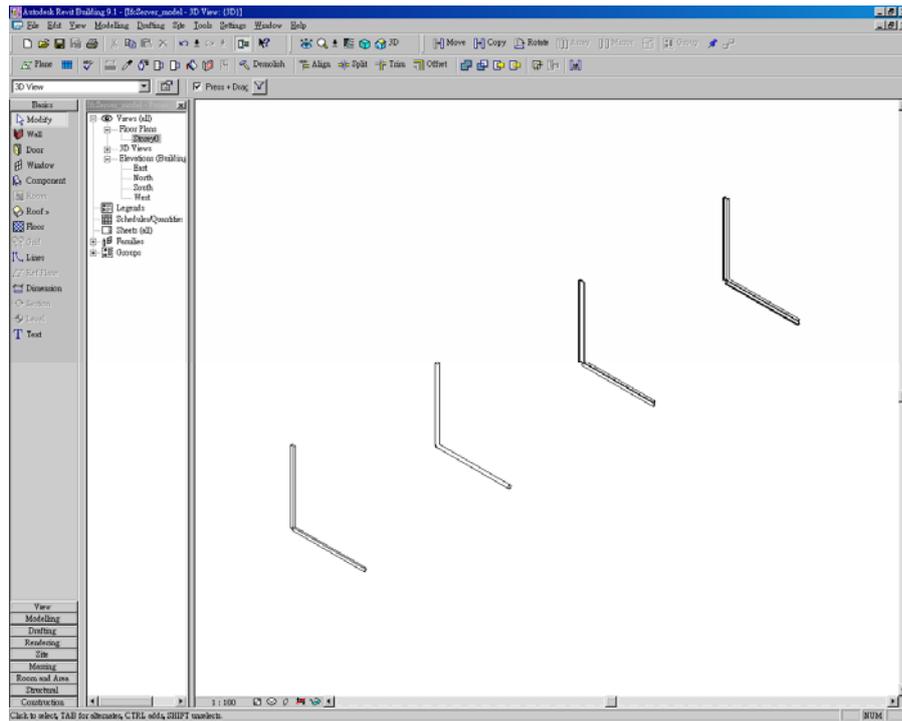


圖 4.77 IFCServer_model 匯入 Revit_3D 視圖

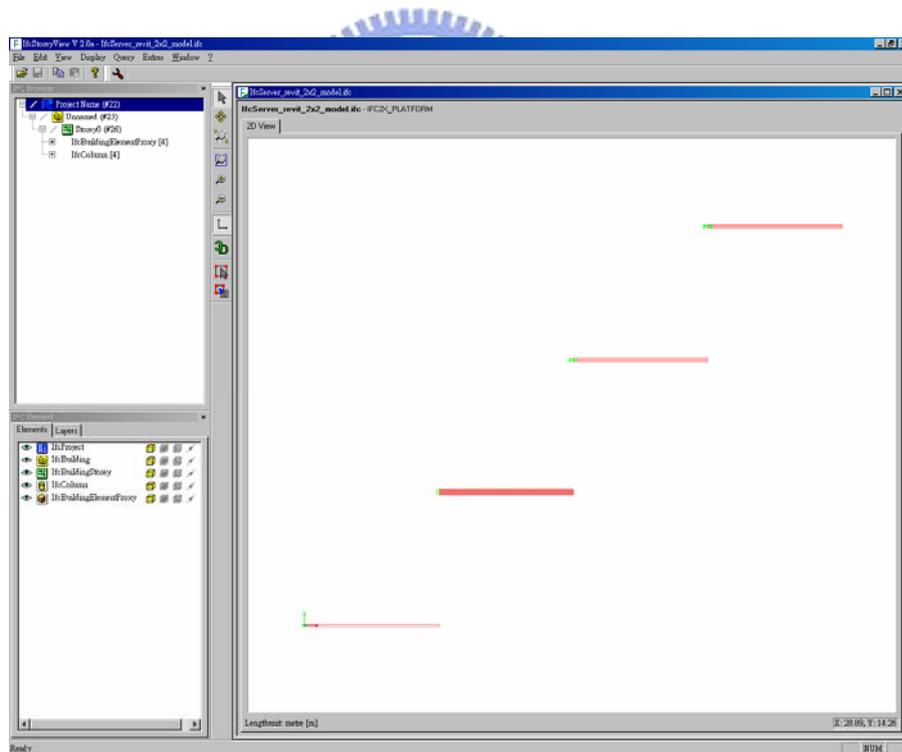


圖 4.78 IFCServer_Revit_model 以 IfcStoreyView 平面圖

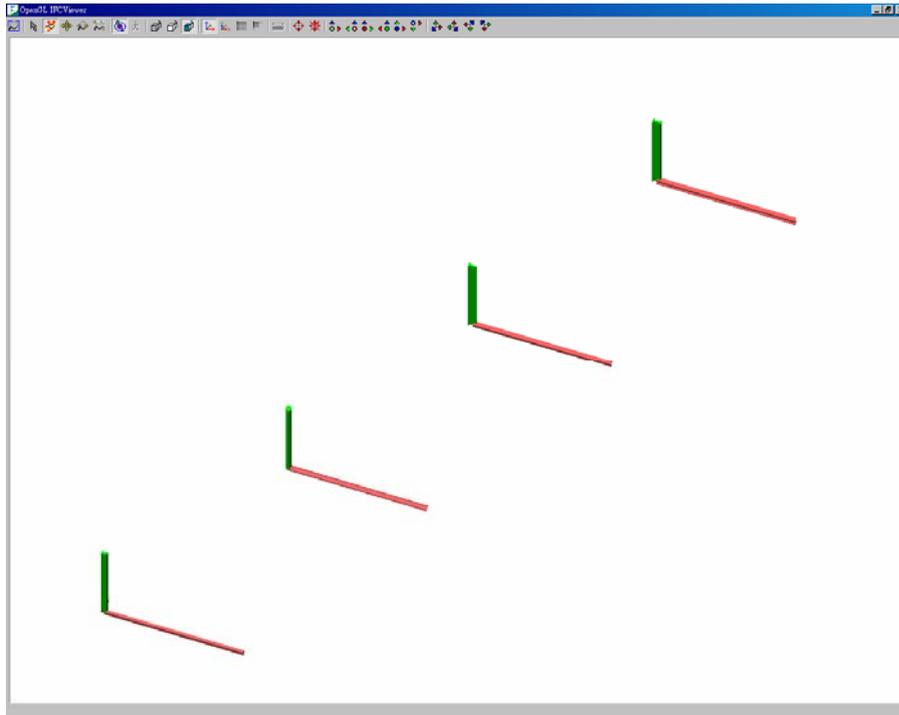


圖 4.79 IFCServer_Revit_model 以 IfcStoreyView_3D 視圖

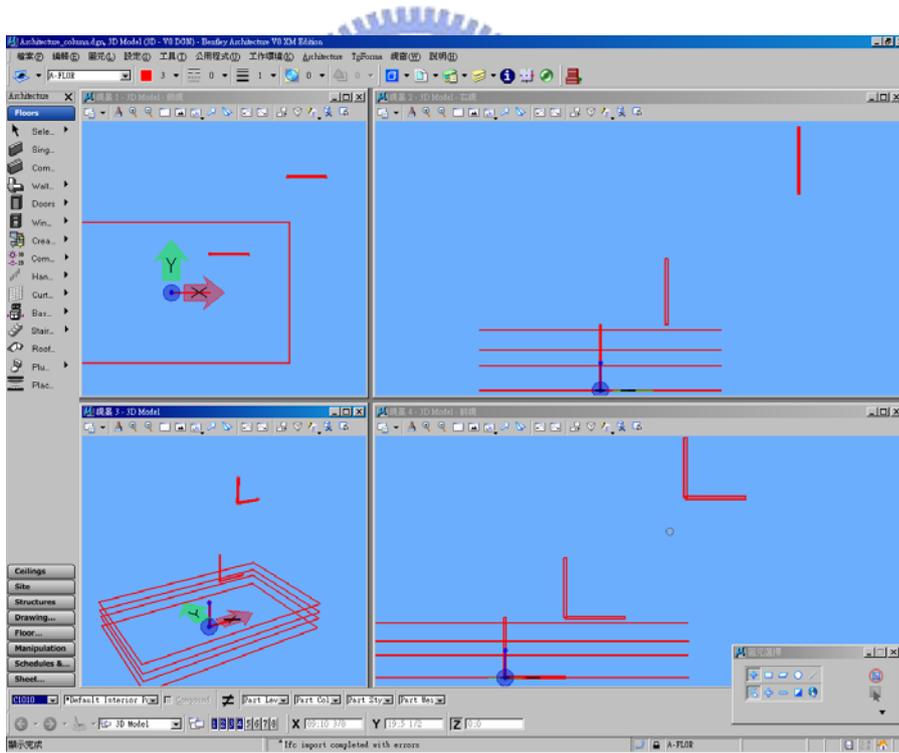


圖 4.80 IFCServer_model 匯入 Bentley 四視圖

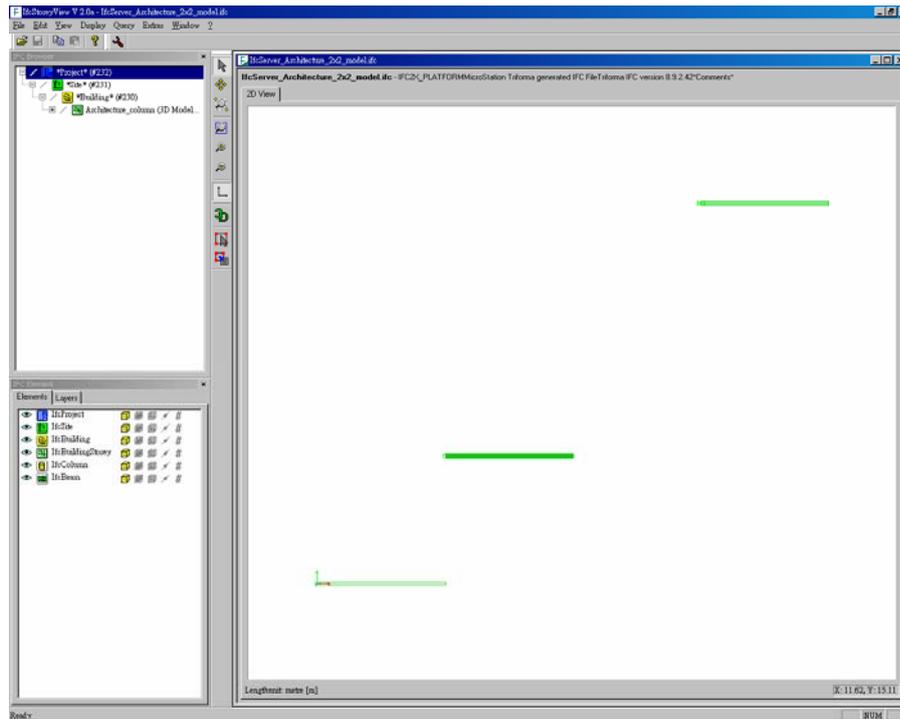


圖 4.81 IFCServer_Bentley_model 以 IfcStoreyView 平面圖

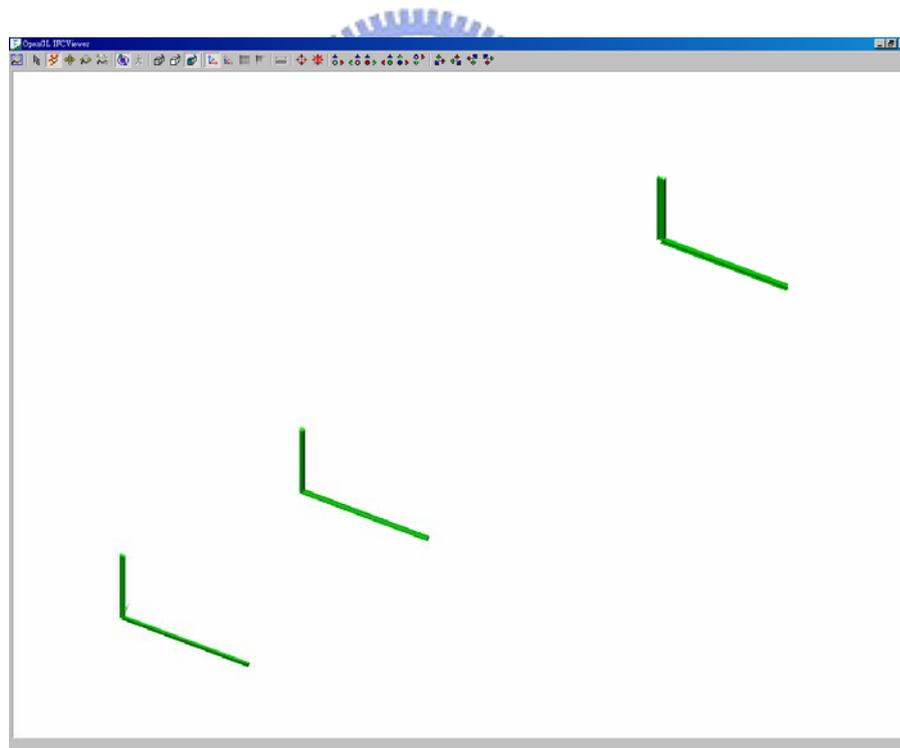


圖 4.82 IFCServer_Bentley_model 以 IfcStoreyView_3D 視圖

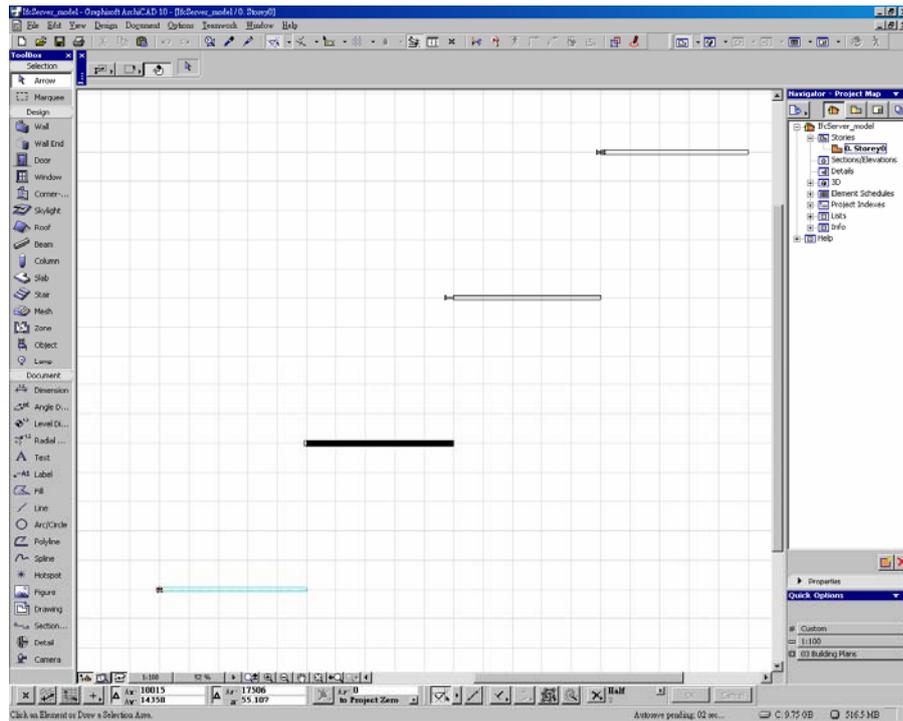


圖 4.83 IFCSrver_model 匯入 ArchiCAD 平面圖

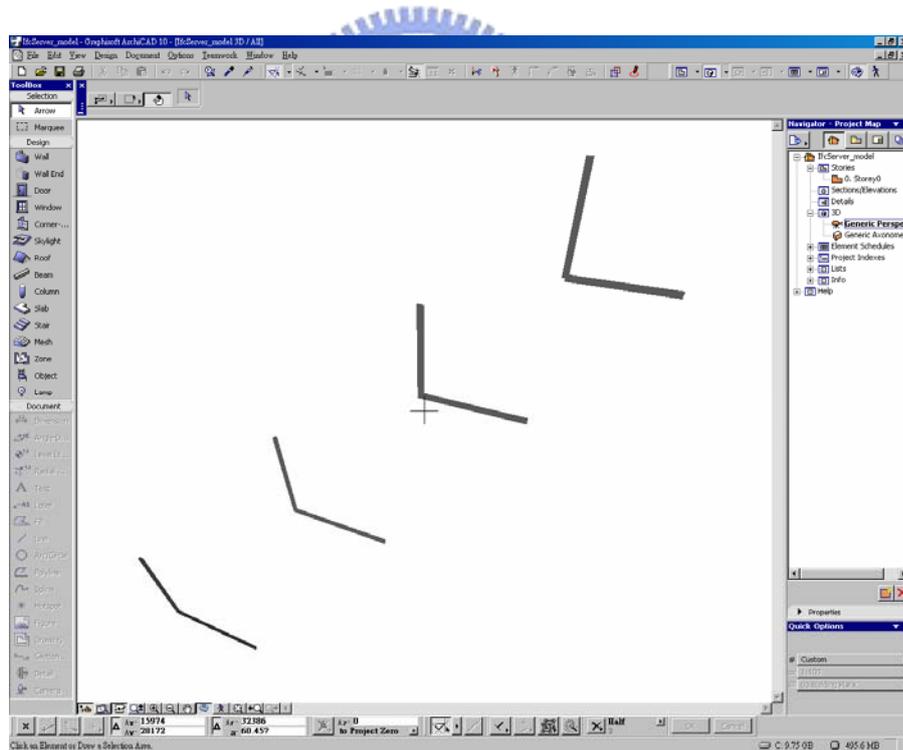


圖 4.84 IFCSrver_model 匯入 ArchiCAD_3D 視圖

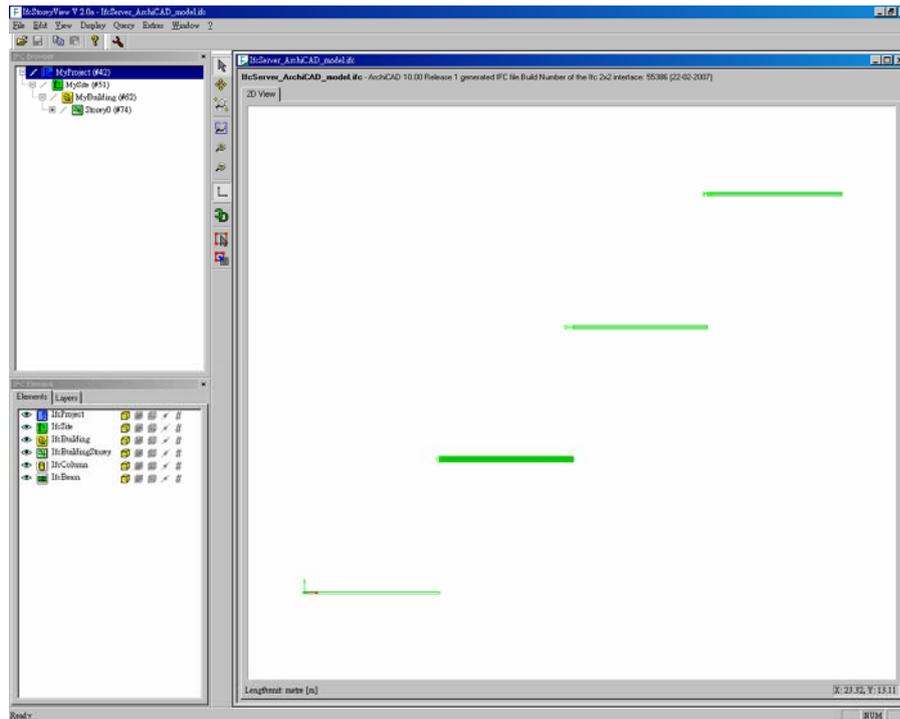


圖 4.85 IfcServer_ArchiCAD_model 以 IfcStoreyView 平面圖

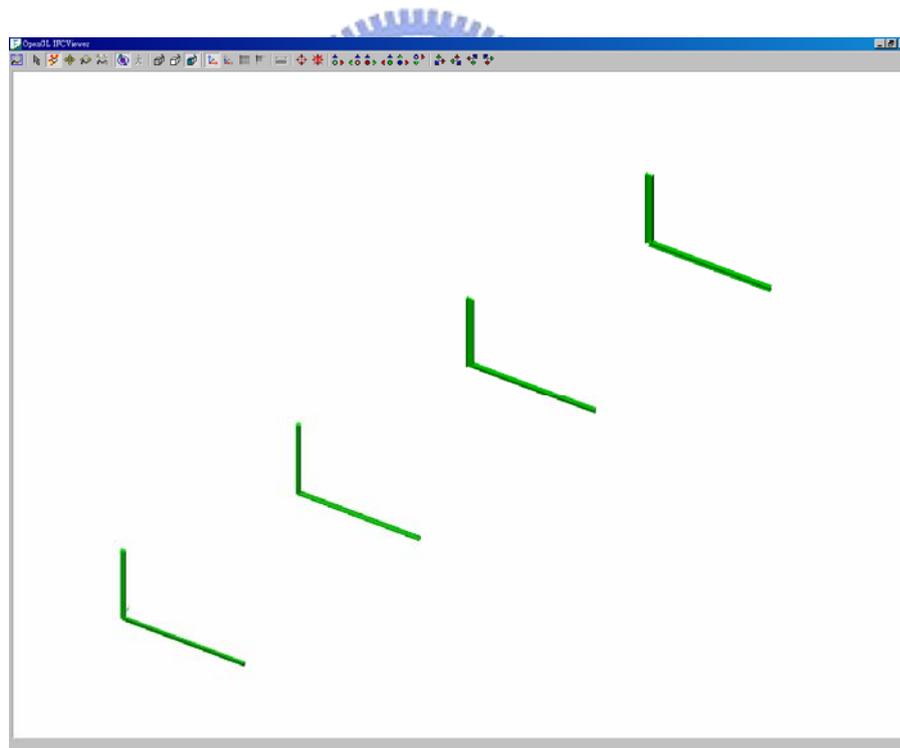


圖 4.86 IfcServer_ArchiCAD_model 以 IfcStoreyView_3D 視圖

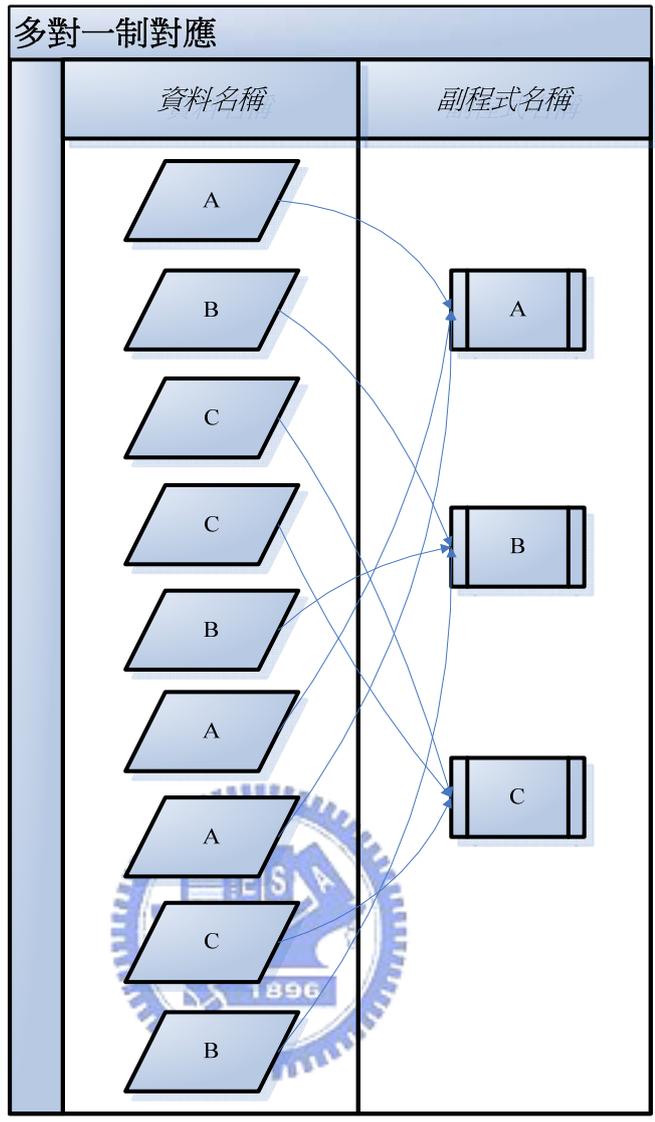


圖 5.1 多對一制對應圖

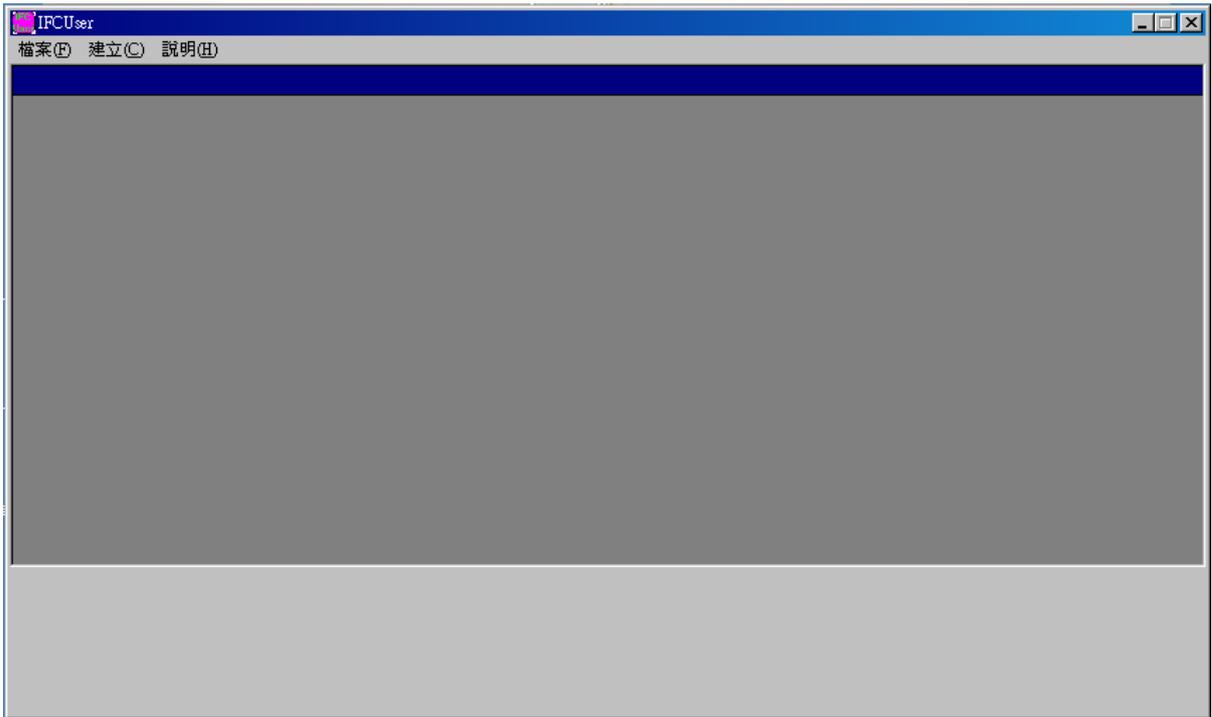


圖 5.3 IFCUser 起始畫面

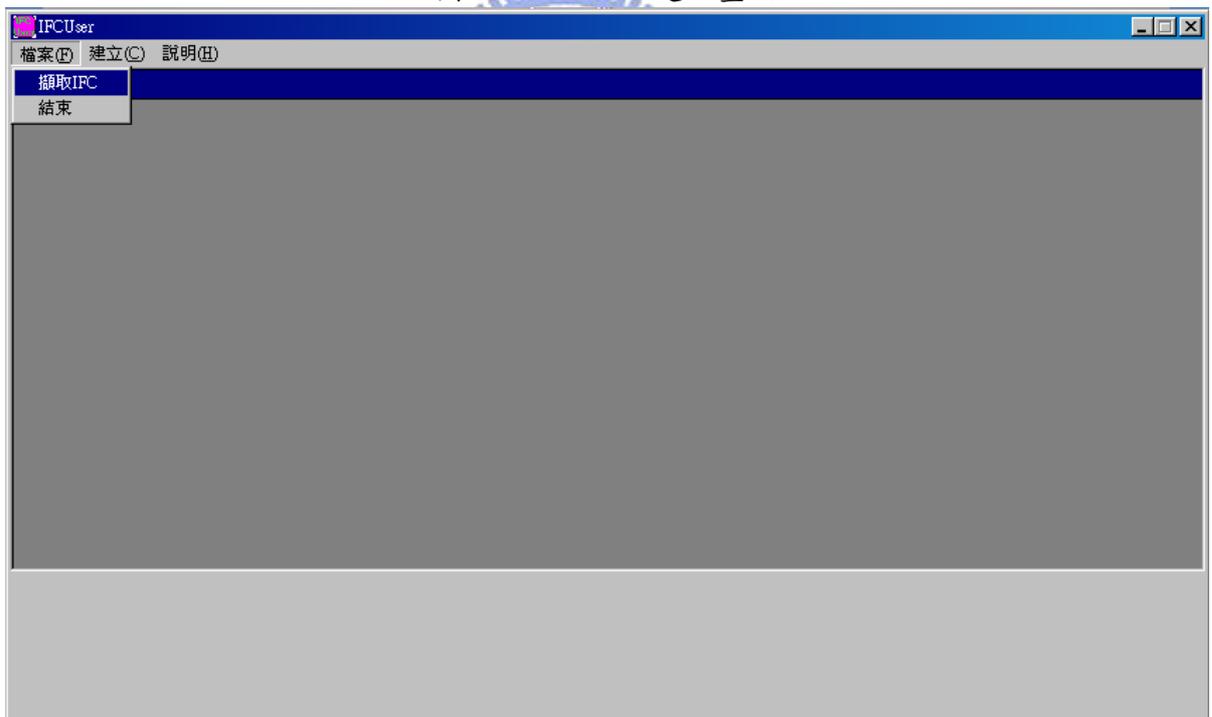


圖 5.4 擷取 IFC 檔案

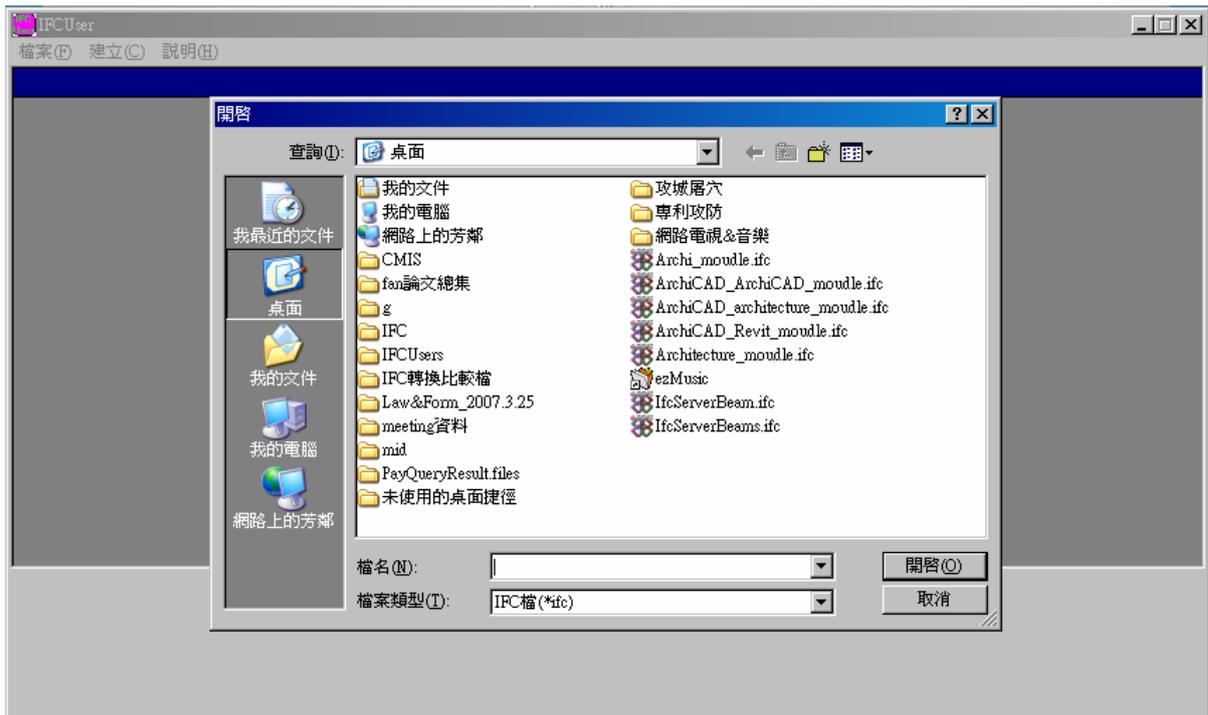


圖 5.5 選擇載入 IFC 檔案

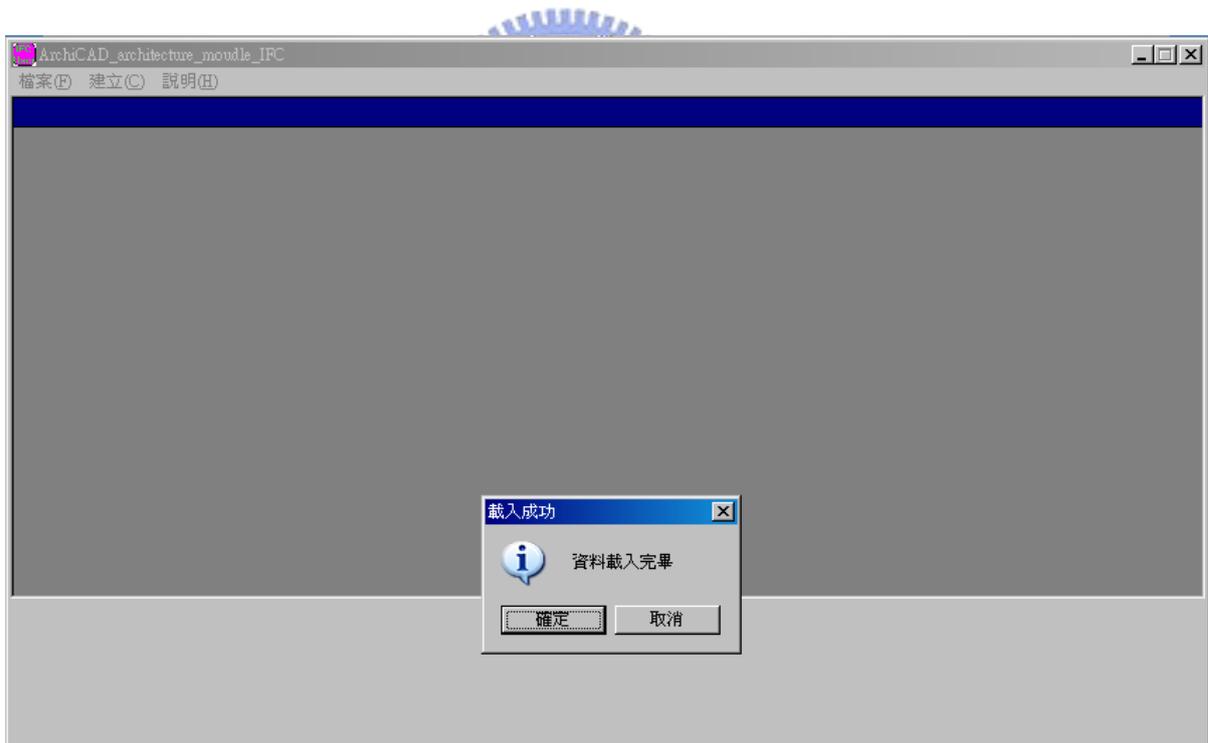


圖 5.6 載入 IFC 檔案成功

ArchCAD_architecture_moudle_IFC

檔案(F) 建立(C) 說明(H)

NewDataSet

物件	形狀	GlobalID	Length	Area	Ix	Iy	XDim
column	Rectangle_Shape	U2EoC2eDb3fAu9ka3Ys5X7'	2.8	.12000	90000E-03	.16000E-02	0.4
column	Circle_Shape	UTwaEiWRD67Q0synlMpSsh'	2.8	.70686E-01	39761E-03	.39761E-03	
column	Rectangle_Shape	'3vz9wgLp5D7gcwQa0vKc\$2'	2.8	.30000	90000E-02	.62500E-02	0.5
column	Rectangle_Shape	UnECDvb_XAIOFn775uNMK'	2.8	.15750	.26578E-02	.16078E-02	0.35
beam	Rectangle_Shape	UFZXMsmh91xAfdSUOKTrgu'	19.7	40000E-01	13333E-03	.13333E-03	0.2
beam	Rectangle_Shape	2DPt\$ml17f28xVSArSv40av'	19.7	40000E-01	13333E-03	.13333E-03	0.2
beam	Rectangle_Shape	UMpLKXrn4zBFaSXcBaCle'	19.7	60000E-01	.20000E-03	.45000E-03	0.3
beam	Rectangle_Shape	UvlePgyEfBAekOGnXHQotg'	19.7	60000E-01	45000E-03	.20000E-03	0.2
*							

圖 5.7 擷取完成

Revit_model_IFC

檔案(F) 建立(C) 說明(H)

NewDataSet

物件	形狀	GlobalID	Length	Area	Ix	Iy	XDim
column	Rectangle_Shape	Uk0adKhX91Og24dJe3FJbH'	4000.	28975E+06	89847E+10	54479E+10	474.9999999
column	Circle_Shape	Uk0adKhX91Og24dJe3FJbE'	3999.99999999967	.29225E+06	.67966E+10	.67966E+10	
column	Rectangle_Shape	Uk0adKhX91Og24dJe3FJCa'	3999.99999999966	27000E+06	45562E+10	81000E+10	600.
column	Circle_Shape	Uk0adKhX91Og24dJe3FJCe'	3999.99999999966	28274E+06	63617E+10	.63617E+10	
column	Arbitrary_Closed_S	2yVdZlB3nEdfmQ4VXrmx0'	3999.99999999966	0.0000	0.0000	0.0000	
column	Arbitrary_Closed_S	2yVdZlB3nEdfmQ4VXrmx07'	3999.99999999966	0.0000	0.0000	0.0000	
column	ArbitraryProfileDef	'3kuERBR4X2\$RiqUTTvs2Q1'	3999.99999999966	0.0000	0.0000	0.0000	
column	ArbitraryProfileDef	'3kuERBR4X2\$RiqUTTvs2Q4'	3999.99999999966	0.0000	0.0000	0.0000	
beam	Rectangle_Shape	'3PxN2kkmb4SxCHgPHrw9Tq'	17056.6918996284	18000E+06	.13500E+10	.54000E+10	599.9999999
beam	Arbitrary_Closed_S	'3PxN2kkmb4SxCHgPHrw9RR'	17056.6918996284	0.0000	0.0000	0.0000	
beam	Arbitrary_Closed_S	'3PxN2kkmb4SxCHgPHrw9R3'	17056.6918996284	0.0000	0.0000	0.0000	
beam	Arbitrary_Closed_S	'3PxN2kkmb4SxCHgPHrw9Rg'	17056.6918996284	0.0000	0.0000	0.0000	
beam	Arbitrary_Closed_S	'3PxN2kkmb4SxCHgPHrw9Qg'	17056.6918996284	0.0000	0.0000	0.0000	
beam	ArbitraryProfileDef	'3PxN2kkmb4SxCHgPHrw9PC'	17131.6918996284	0.0000	0.0000	0.0000	
beam	ArbitraryProfileDef	'3PxN2kkmb4SxCHgPHrwA8V'	17056.6918996284	0.0000	0.0000	0.0000	

圖 5.8 擷取 AutoDesk Revit 模型

物件	形狀	GlobalID	Length	Area	Ix	Iy	XDim
beam	Arbitrary_Closed_S	'37ikSTzL9le933pCr0HCc'	19.088125	0.0000	0.0000	0.0000	
beam	Circle_Shape	'2pfKXL\$Q95XO3aVGD70cIV'	19.088125	65669	34317E-01	34317E-01	
beam	Arbitrary_Closed_S	'1YXUs_WPK9mA12BGNZahBf'	19.088125	0.0000	0.0000	0.0000	
beam	Arbitrary_Closed_S	'3Q25lrX51F1wQGoeEBh1 WHI'	19.088125	0.0000	0.0000	0.0000	
beam	Arbitrary_Closed_S	'3UU0fMnlh8_wLJiVHu_xH'	19.088125	0.0000	0.0000	0.0000	
beam	Arbitrary_Closed_S	'3gE0X2Vsr1fB1KOSUX613p'	19.088125	0.0000	0.0000	0.0000	
beam	Arbitrary_Closed_S	'0HTxwgJXr800ebH0eXEy7'	19.088125	0.0000	0.0000	0.0000	
beam	ArbitraryProfileDef	'0Trf\$uyHBI9IsrbmYBLE'	19.088125	0.0000	0.0000	0.0000	
beam	ArbitraryProfileDef	'298DM4LAX8UQHs4 WVno35'	19.088125	0.0000	0.0000	0.0000	
beam	Arbitrary_Closed_S	'0dCkW6C_rEkfXevaDijKXq'	9.130964	0.0000	0.0000	0.0000	
beam	Arbitrary_Closed_S	'3BIB19Q9B699olmtkEMbr'	9.130964	0.0000	0.0000	0.0000	
beam	Circle_Shape	'02ald6eEvFBAvrl1UDyS7y'	9.130964	65669	34317E-01	34317E-01	
beam	Arbitrary_Closed_S	'2TprPUp511NBBzyo\$fgsLB'	9.130964	0.0000	0.0000	0.0000	
beam	Arbitrary_Closed_S	'2IFueBL14nflh8mSIKPRX'	9.130964	0.0000	0.0000	0.0000	
beam	Arbitrary_Closed_S	'2GMtQbm5P4LbBk3ZsEyCQd'	9.130964	0.0000	0.0000	0.0000	
beam	ArbitraryProfileDef	'1NEzip6iLF\$RohM4CesgeK'	9.130964	0.0000	0.0000	0.0000	

圖 5.9 擷取 Bentley Architecture 模型

物件	形狀	GlobalID	Length	Area	Ix	Iy	XDim
column	Circle_Shape	'3thvBthYX3mRWX0k1J9ocp'	2.8	.70686E-01	39761E-03	39761E-03	
column	Rectangle_Shape	'0\$5GVpfRf2aPfa2wWMkC5f'	2.8	90000E-01	67500E-03	67500E-03	0.3
column	Arbitrary_Closed_S	'3R9tF2Gh57Sv38DkLctdK'	2.8	0.0000	0.0000	0.0000	
column	Arbitrary_Closed_S	'1LOU4TJYLFezrgP1EipOpW'	2.8	0.0000	0.0000	0.0000	
column	Arbitrary_Closed_S	'3OzYeuMaHC9vAphzTqL TiT'	2.8	0.0000	0.0000	0.0000	
column	Arbitrary_Closed_S	'1Cjknuy112EfrKHUchNHf'	2.8	0.0000	0.0000	0.0000	
column	Arbitrary_Closed_S	'2CU95TE0fFokC\$AV2Q8I3'	2.8	0.0000	0.0000	0.0000	
column	Arbitrary_Closed_S	'3ygeCZwIXAixMpOfiUi4MN'	2.8	0.0000	0.0000	0.0000	
column	Arbitrary_Closed_S	'3_ugSBHtFceXlQnWq3DHt'	2.8	0.0000	0.0000	0.0000	
beam	Rectangle_Shape	'30Q8gSsd18_Q\$TqjZRprqi'	16.0207	.60000E-01	45000E-03	20000E-03	0.2
beam	Arbitrary_Closed_S	'1UPWLHTOD4y90A3aOs4iup'	16.0207	0.0000	0.0000	0.0000	
beam	Arbitrary_Closed_S	'0lpmfZZMbdIgiJNysa16qT'	16.0207	0.0000	0.0000	0.0000	
beam	Arbitrary_Closed_S	'3H8aPBH919\$XNRjBcCzA_5'	16.0207	0.0000	0.0000	0.0000	
beam	Arbitrary_Closed_S	'15nJWJMUH8WRoXZ8LYOdttM'	16.0207	0.0000	0.0000	0.0000	
beam	Arbitrary_Closed_S	'3WeGU__15DUB0iFaxAVrNj'	16.0207	0.0000	0.0000	0.0000	
beam	Arbitrary_Closed_S	'1o4vDODDrERPzFX00VcPPI'	16.0207	0.0000	0.0000	0.0000	

圖 5.10 擷取 Graphisoft ArchiCAD 模型

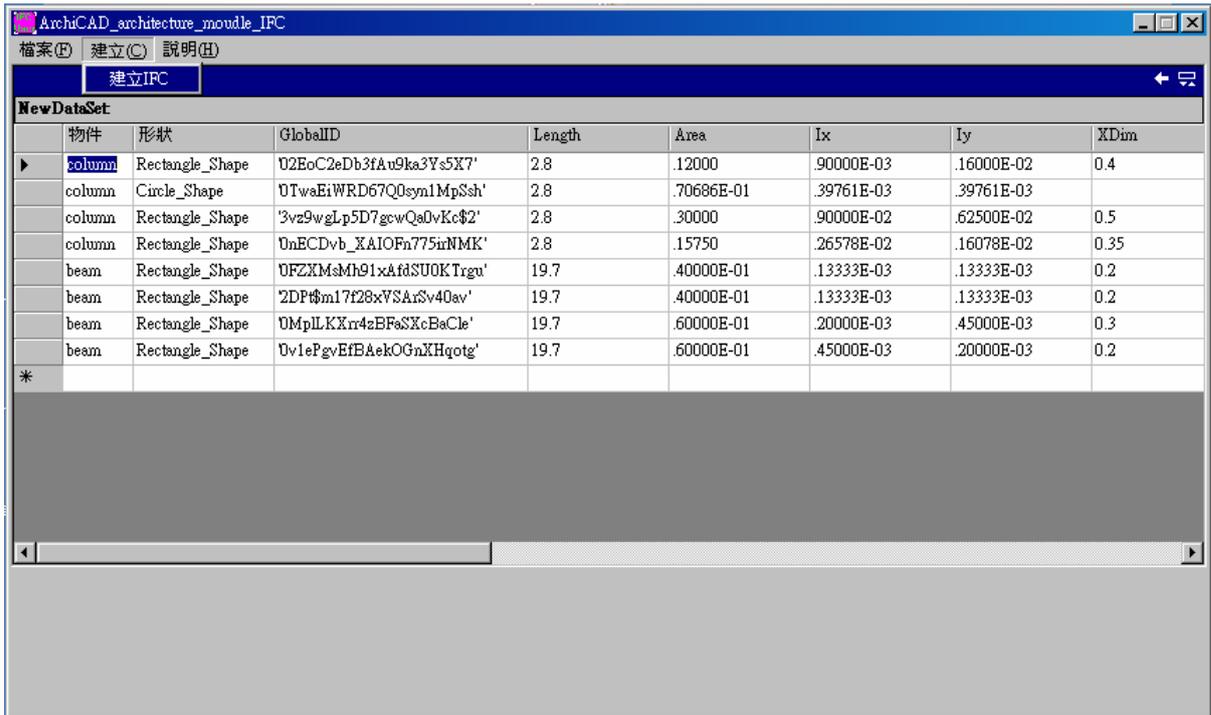


圖 5.11 建立

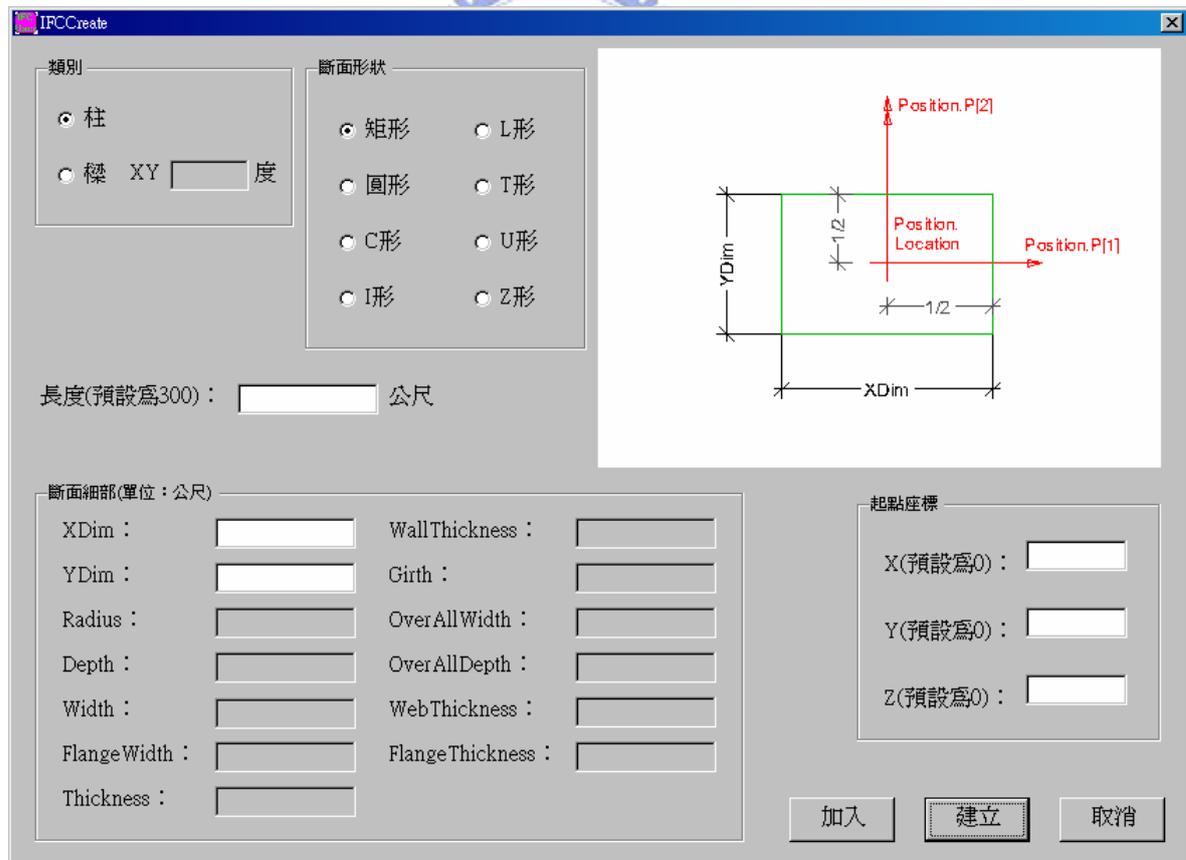


圖 5.12 建立對話盒

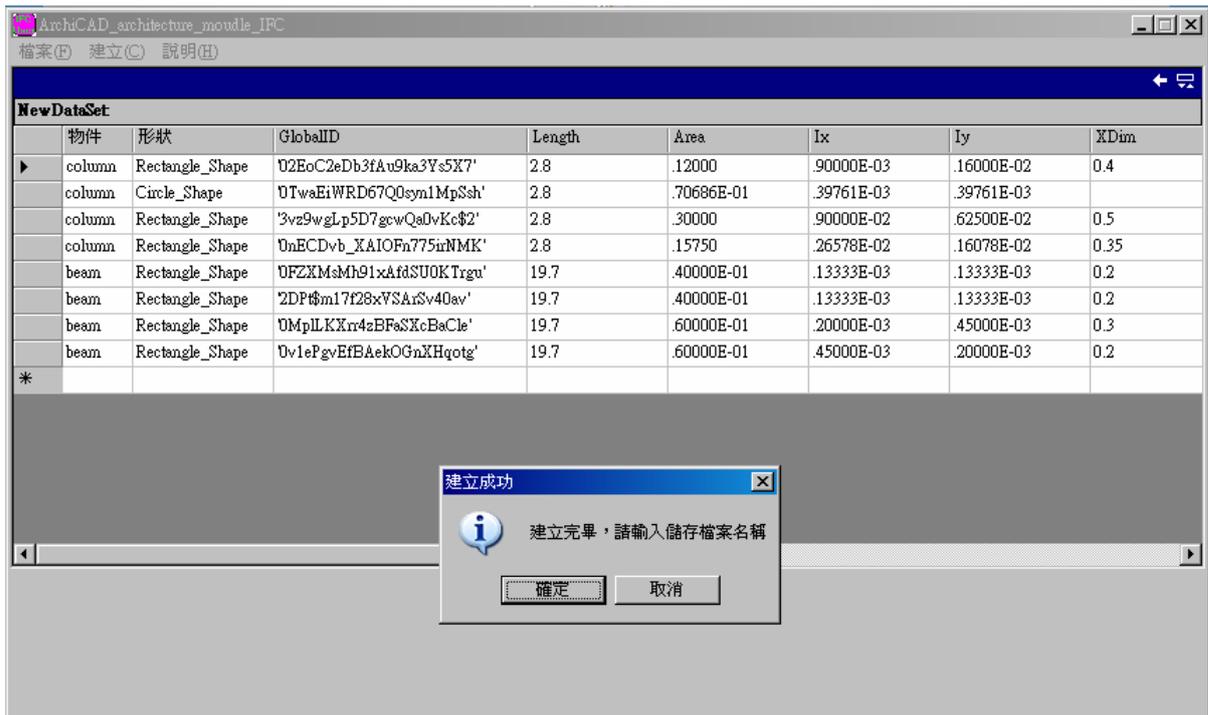


圖 5.13 建立成功

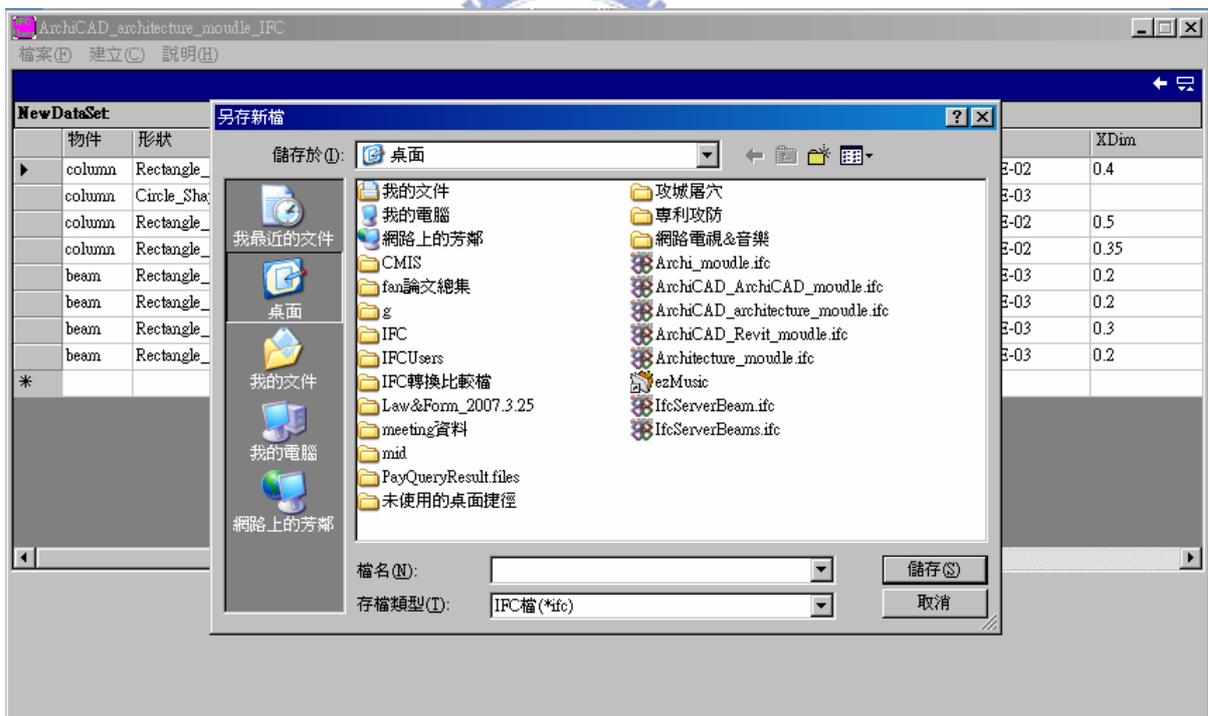


圖 5.14 選擇另存新檔

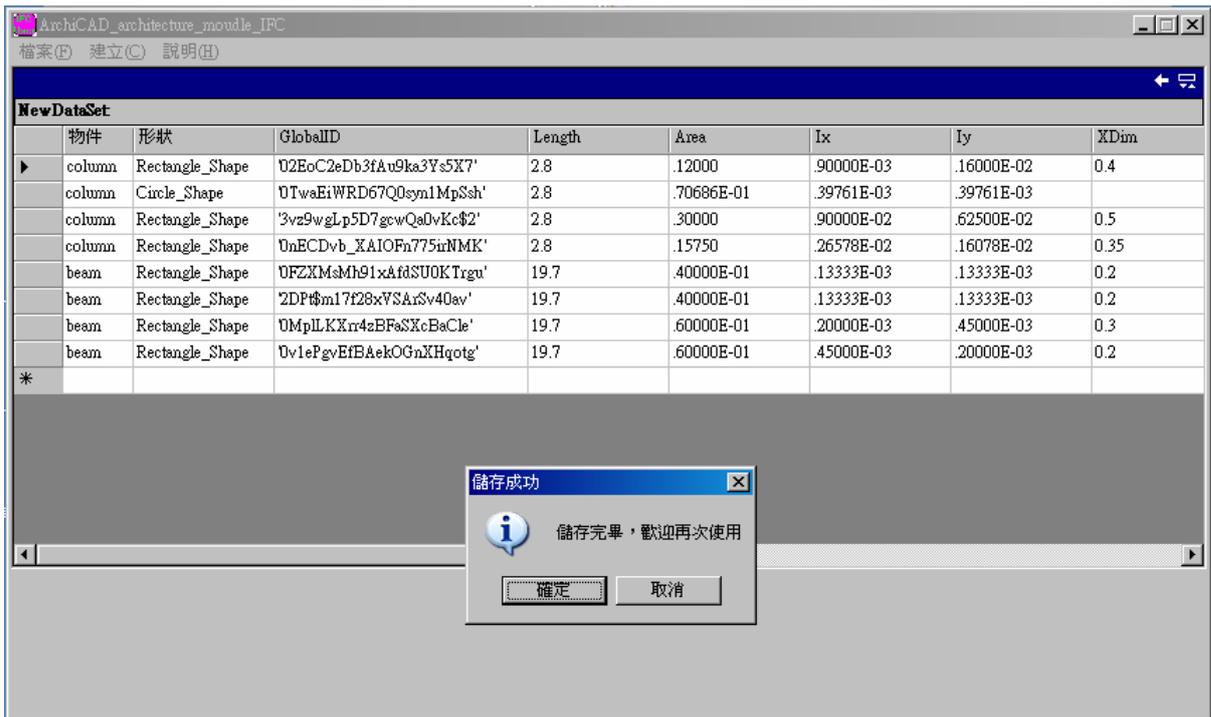


圖 5.15 儲存成功

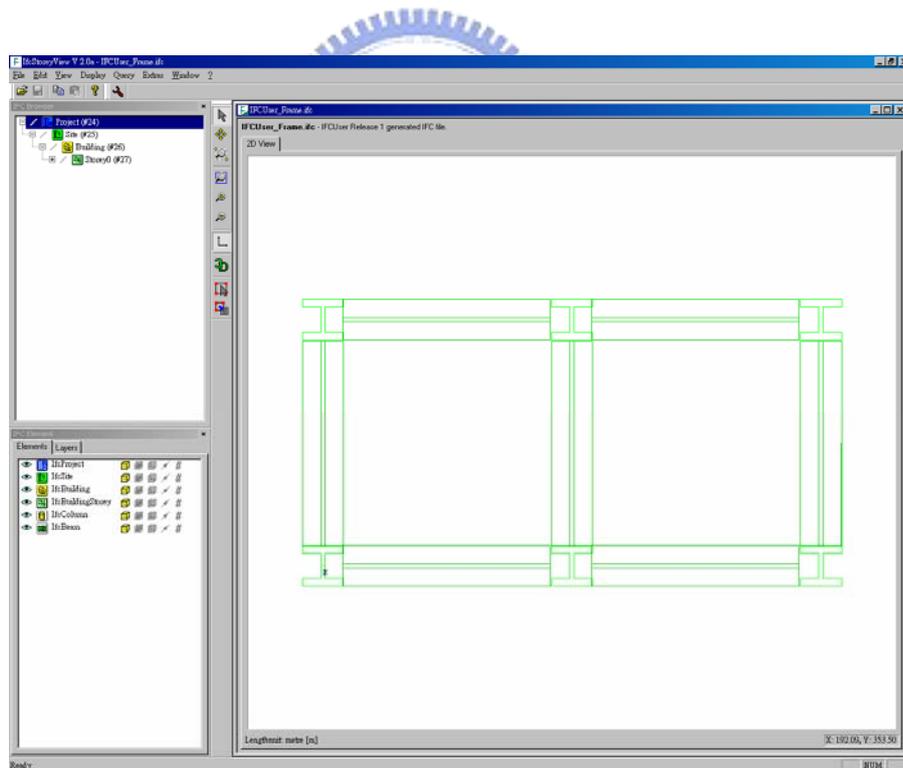


圖 5.16 IFCUser_Frame 以 IfcStoreyView 檢視平面圖

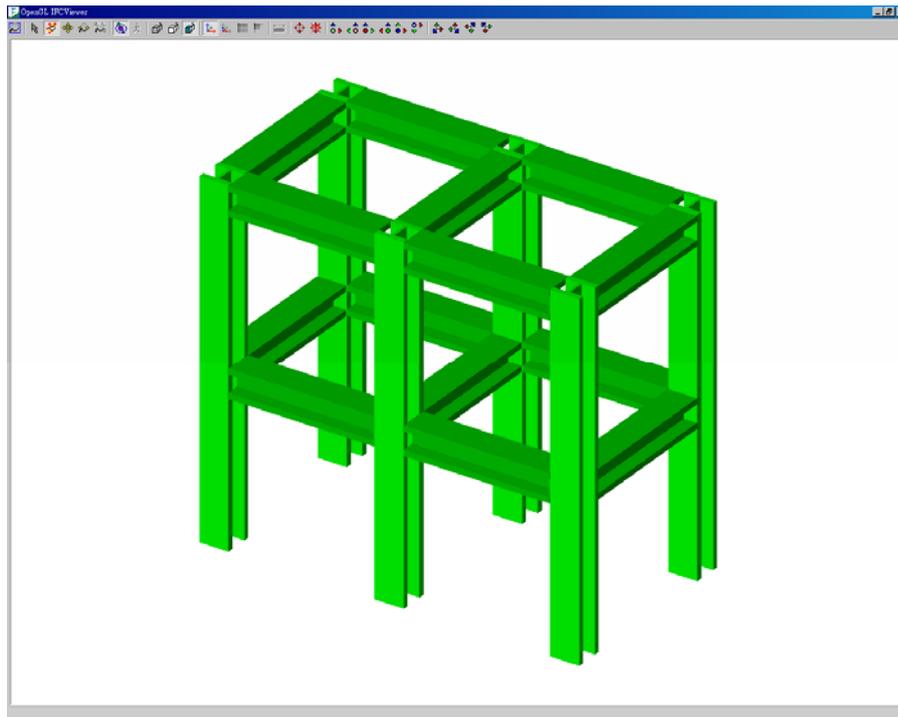


圖 5.17 IFCUser_Frame 以 IfcStoreyView 檢視立體圖

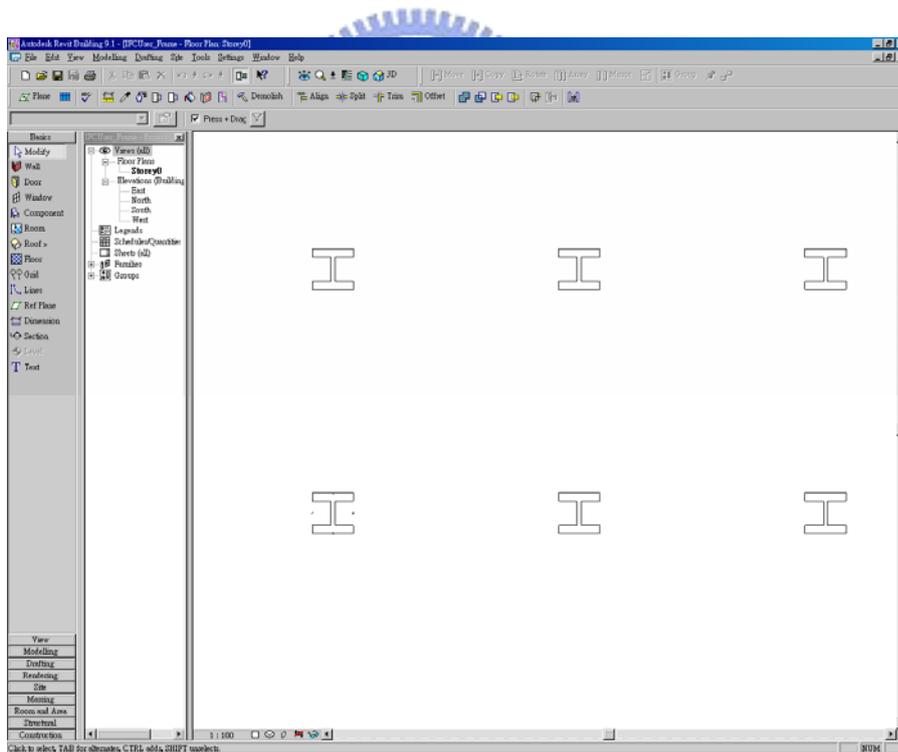


圖 5.18 IFCUser_Frame_Revit 平面圖

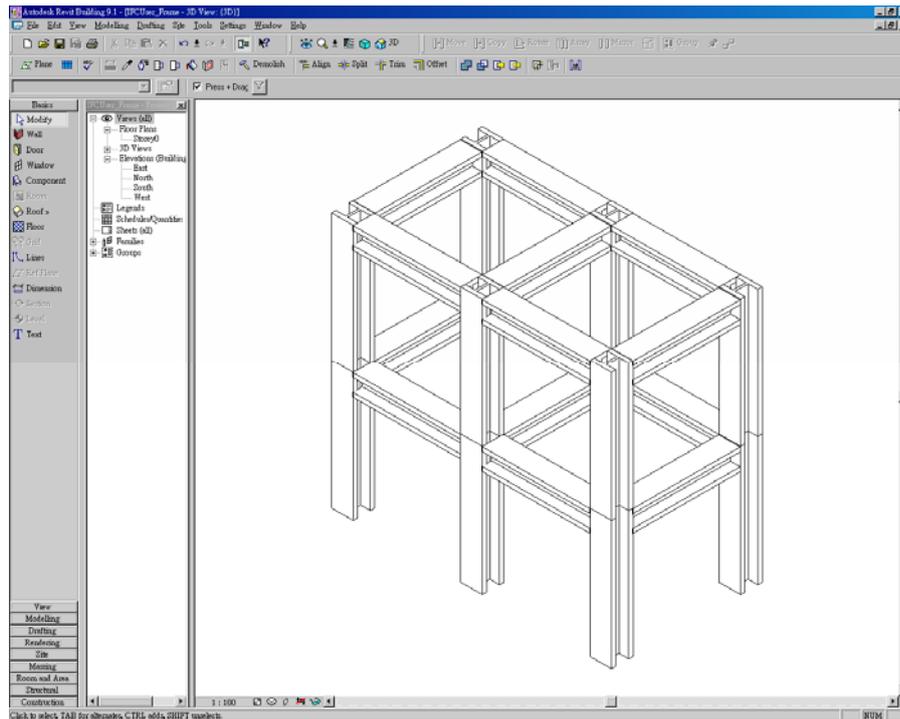


圖 5.19 IFCUser_Frame_Revit 立體圖

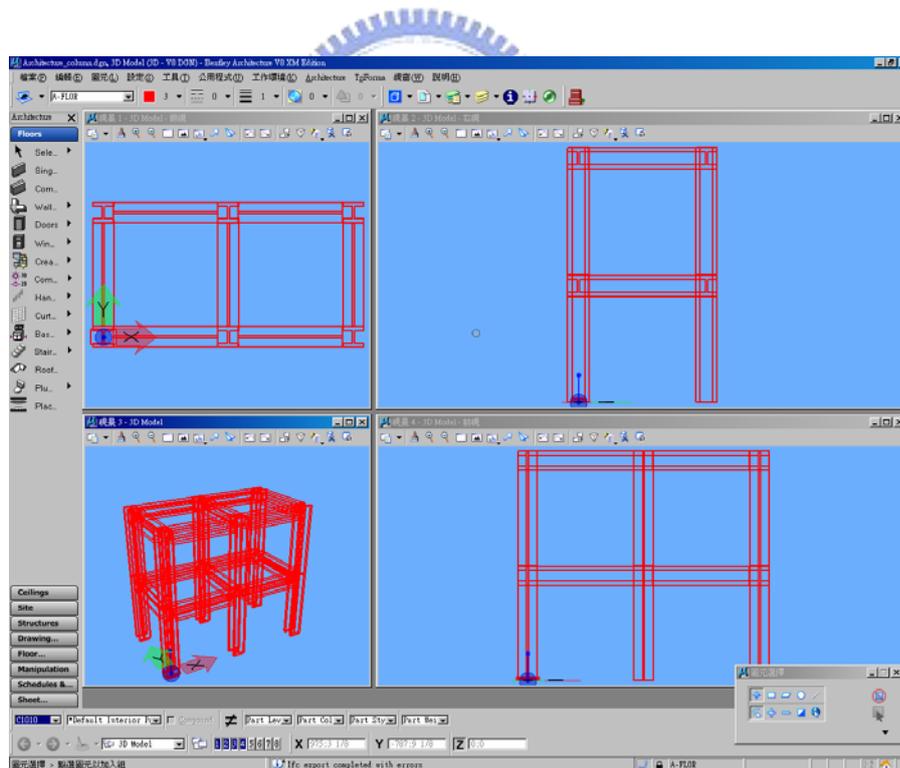


圖 5.20 IFCUser_Frame_Bentley 四視圖

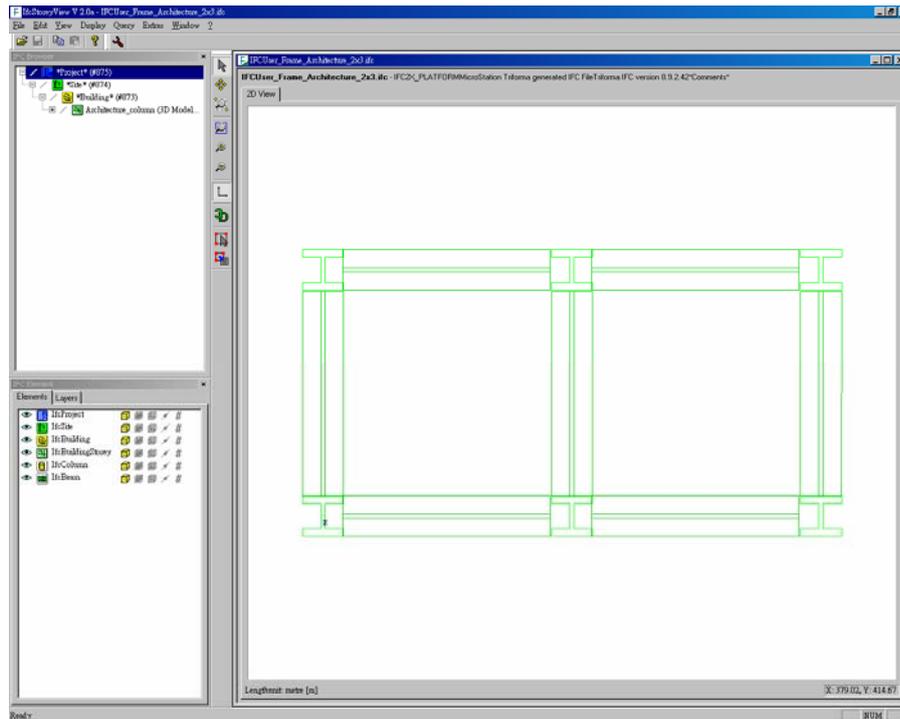


圖 5.21 IfcUser_Frame_Bentley 以 IfcStoreyView 檢視平面圖

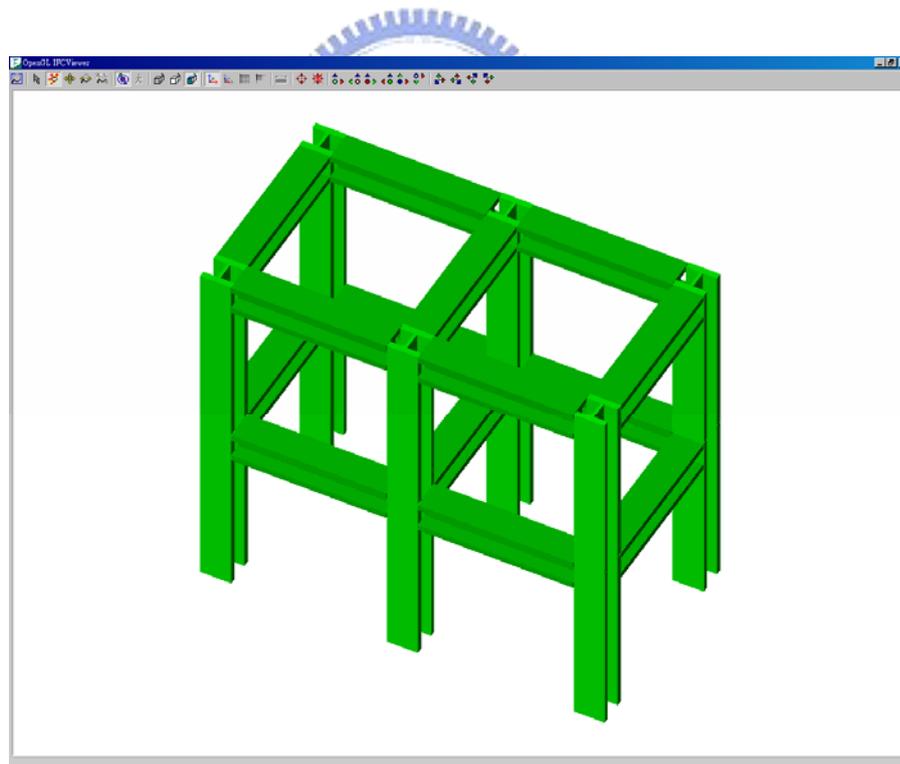


圖 5.22 IfcUser_Frame_Bentley 以 IfcStoreyView 檢視立體圖

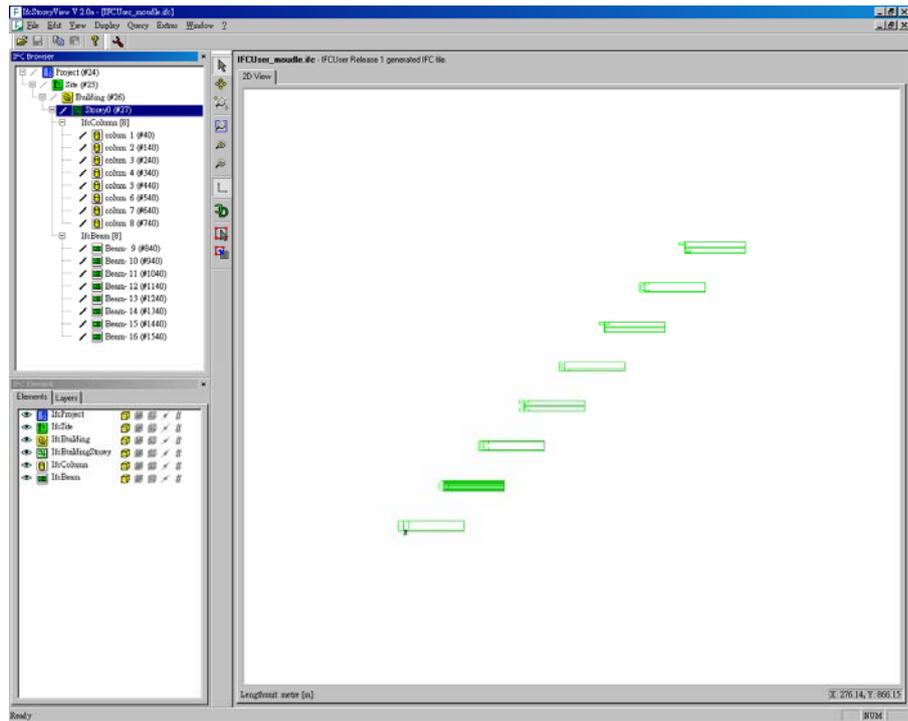


圖 5.23 IFCUser_model 平面圖

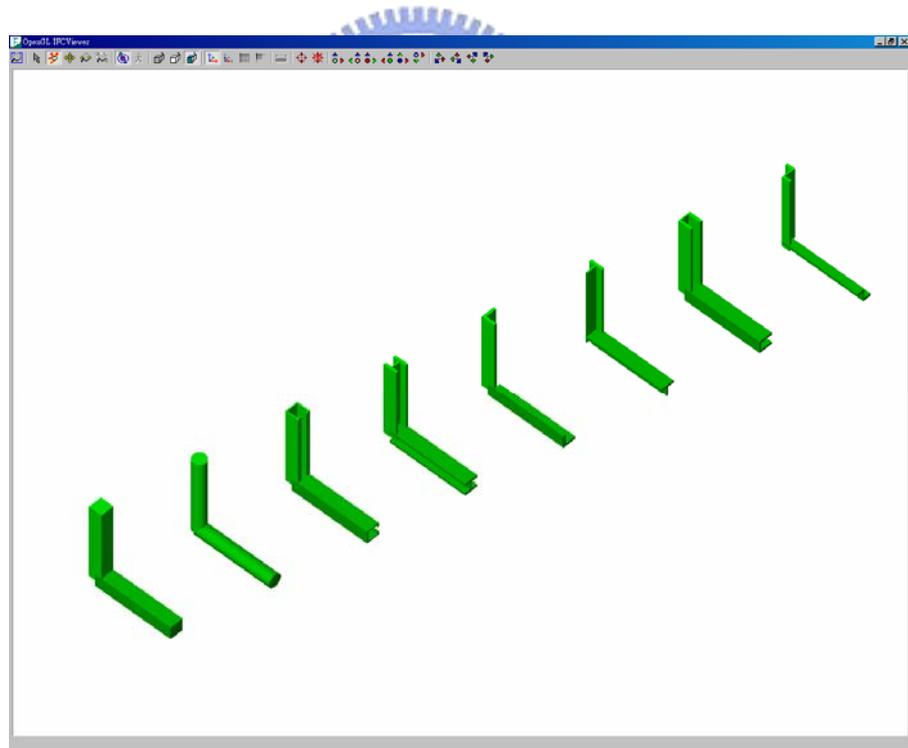


圖 5.24 IFCUser_model_3D 視圖

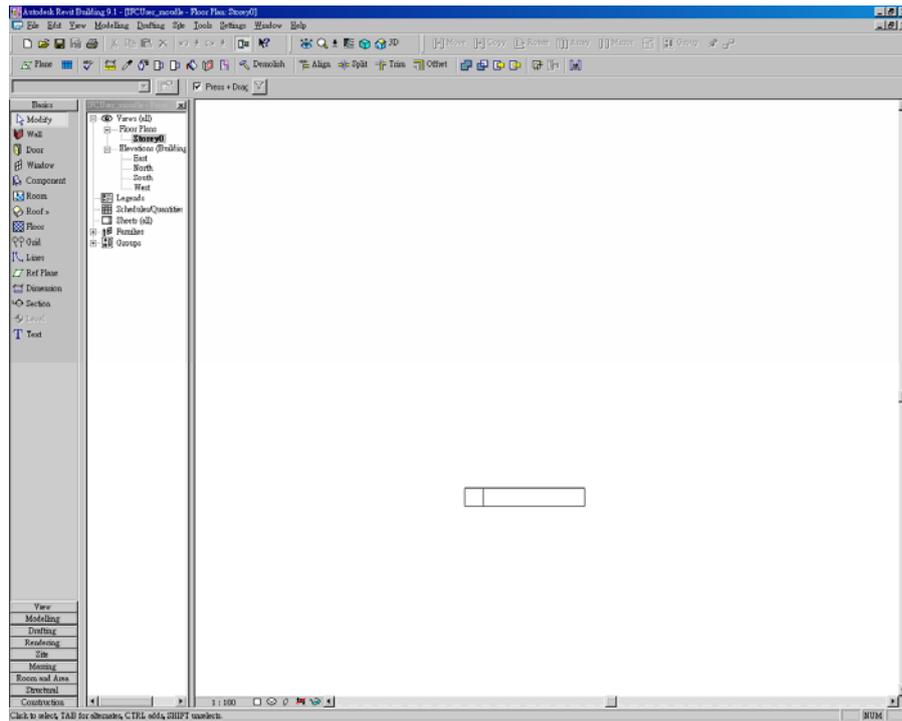


圖 5.25 IFCUser_model 匯入 Revit 平面圖

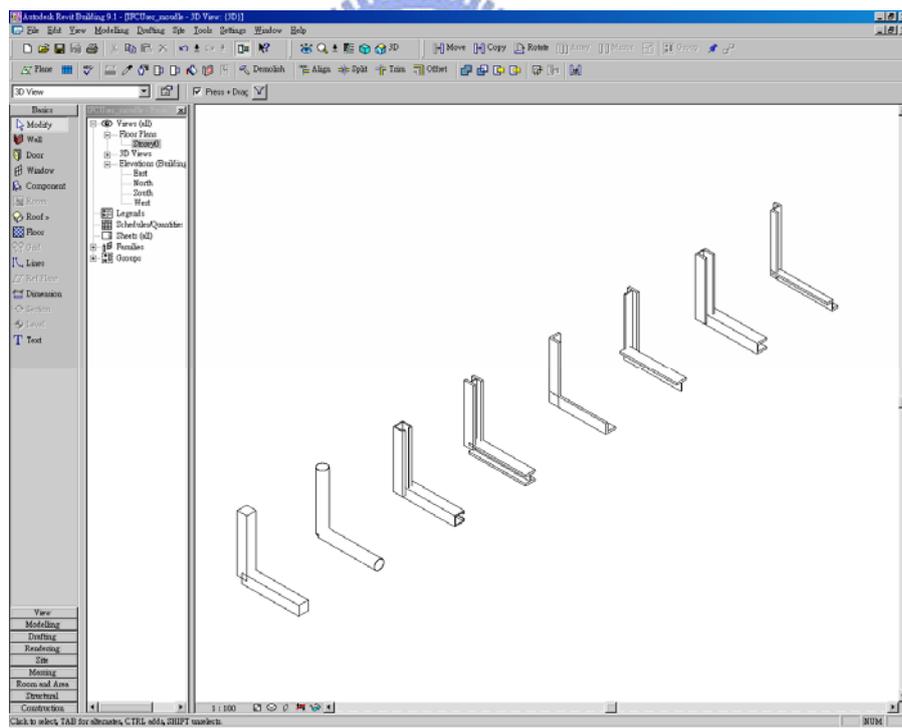


圖 5.26 IFCUser_model 匯入 Revit_3D 視圖

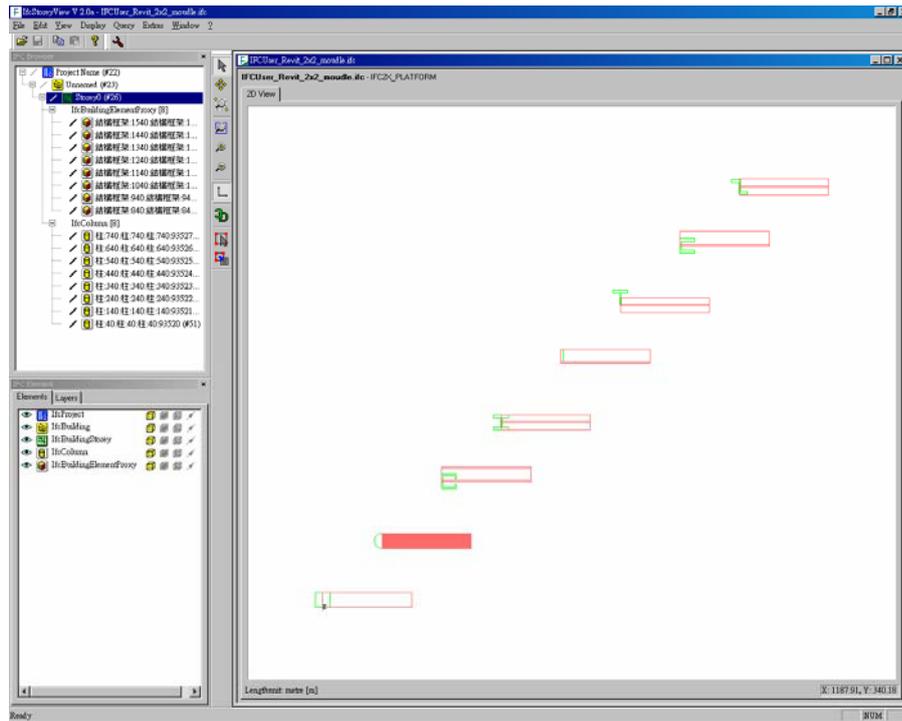


圖 5.27 IFCUser_Revit_model 以 IfoStoreyView 平面圖

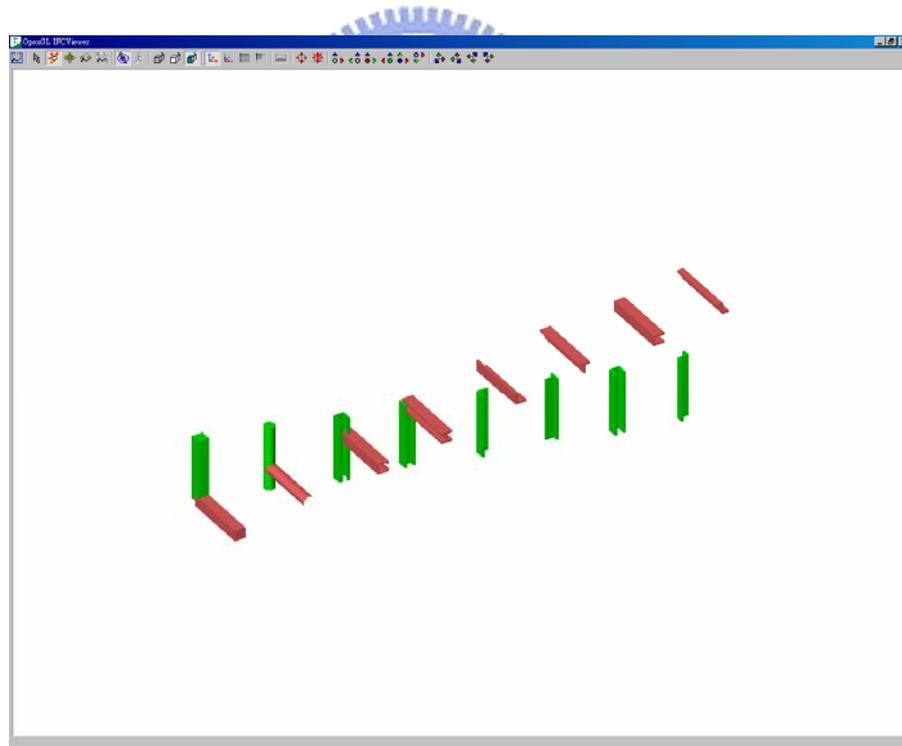


圖 5.28 IFCUser_Revit_model 以 IfoStoreyView_3D 視圖

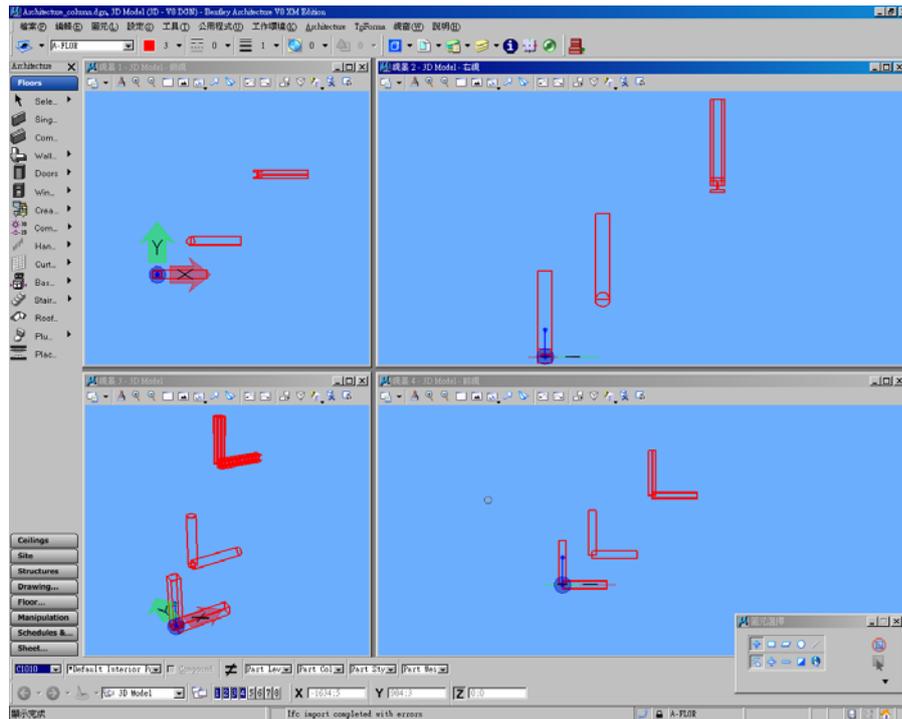


圖 5.29 IFCUser_model 匯入 Bentley 四視圖

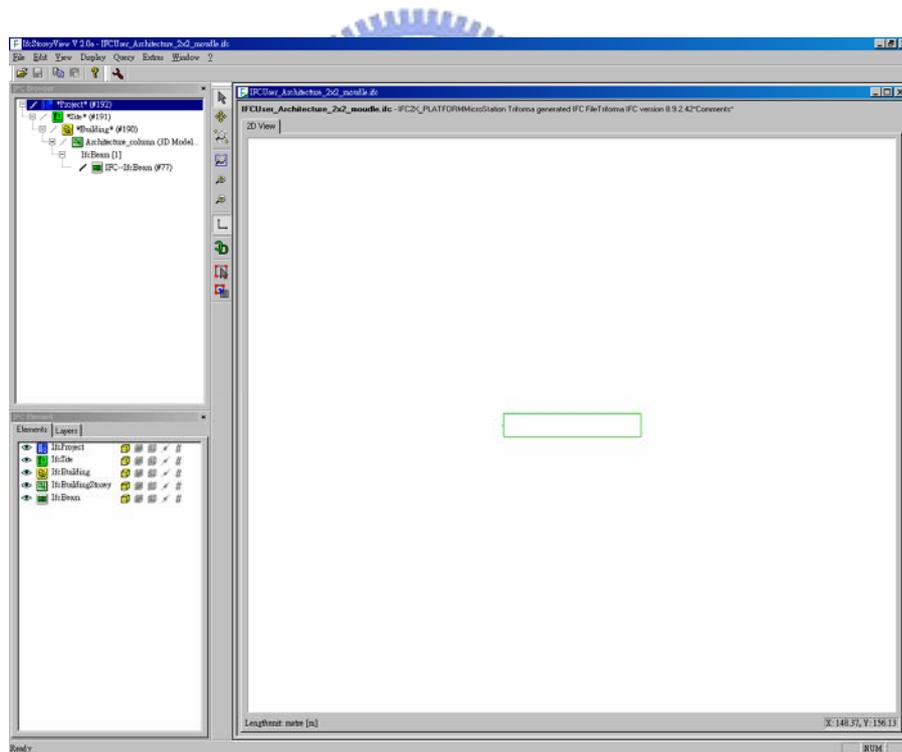


圖 5.30 IFCUser_Bentley_model 以 IfcStoreyView 平面圖

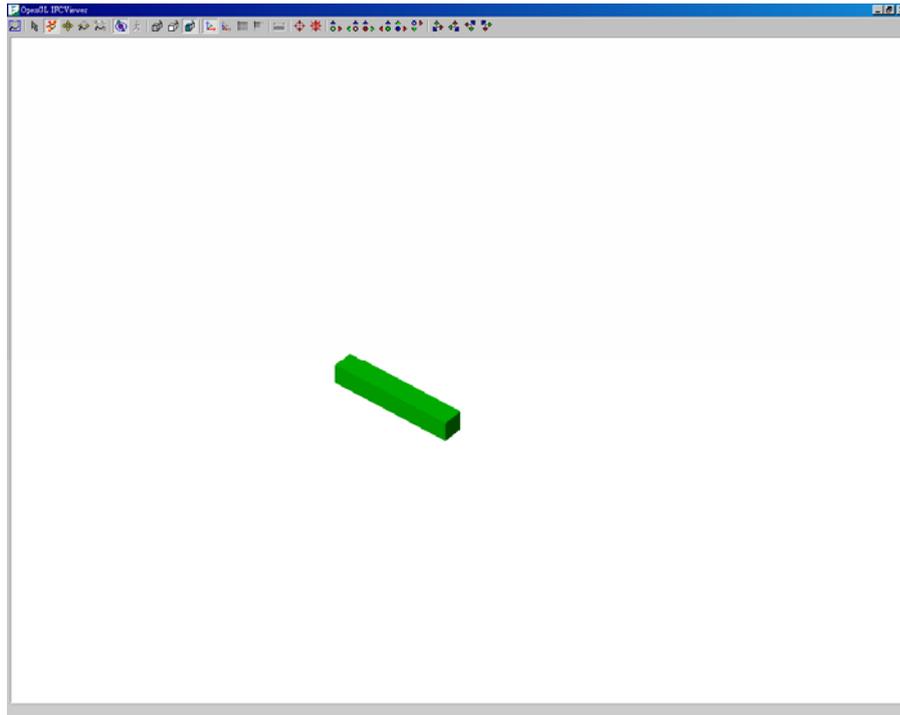


圖 5.31 IFCUser_Bentley_model 以 IfcStoreyView_3D 視圖

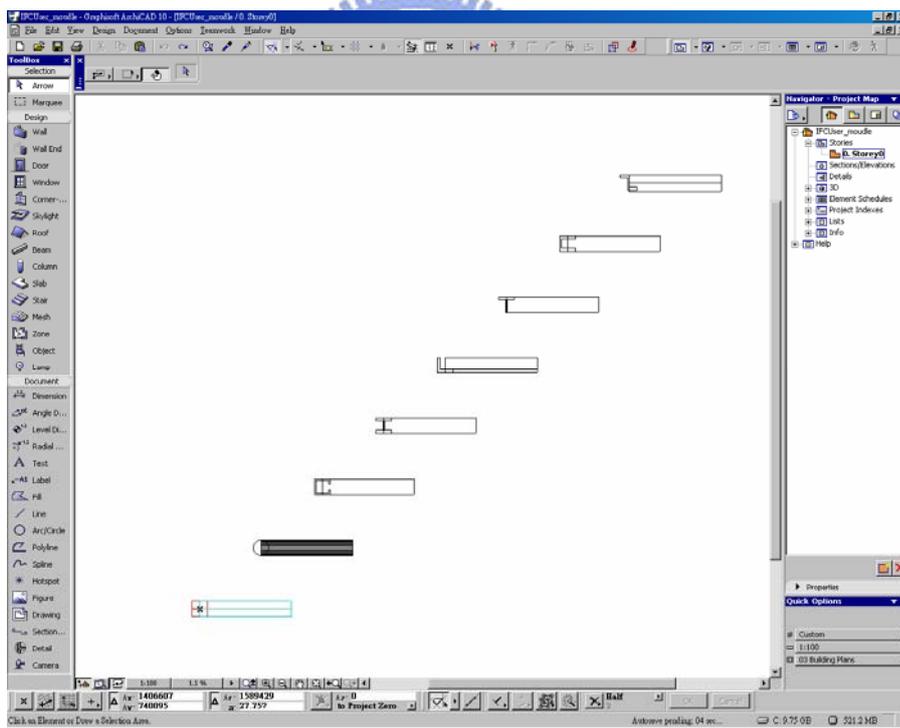


圖 5.32 IFCUser_model 匯入 ArchiCAD 平面圖

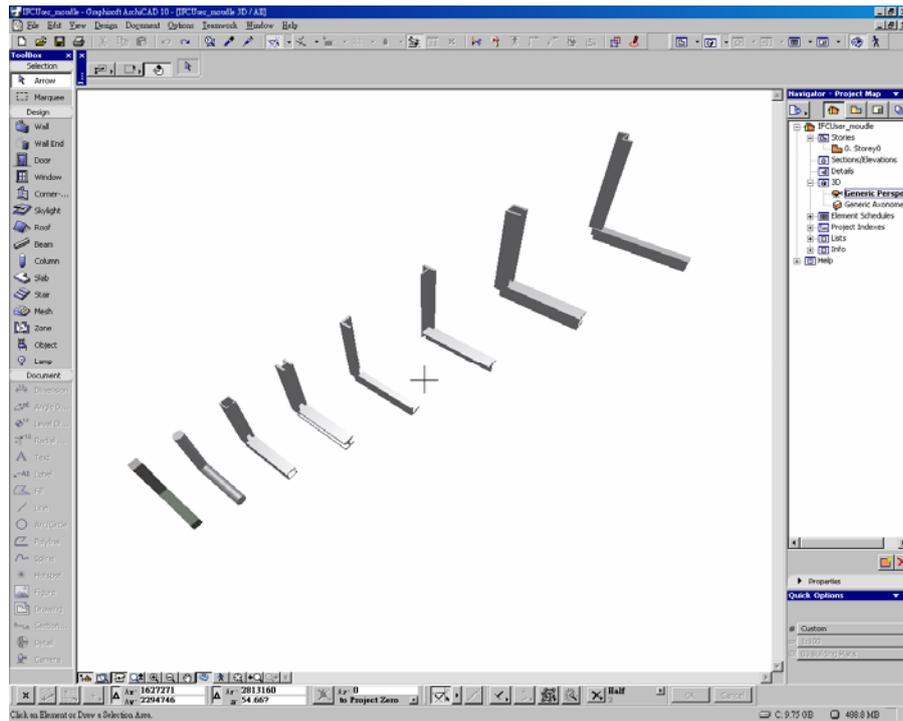


圖 5.33 IFCUser_model 匯入 ArchiCAD_3D 視圖

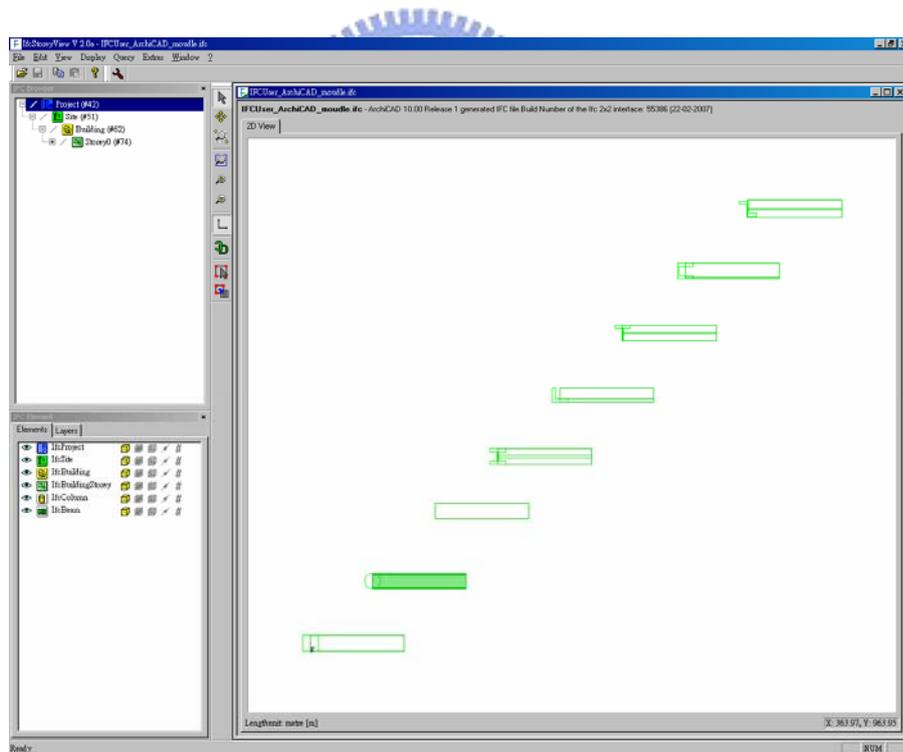


圖 5.34 IFCUser_ArchiCAD_model 以 IfcStoreyView 平面圖

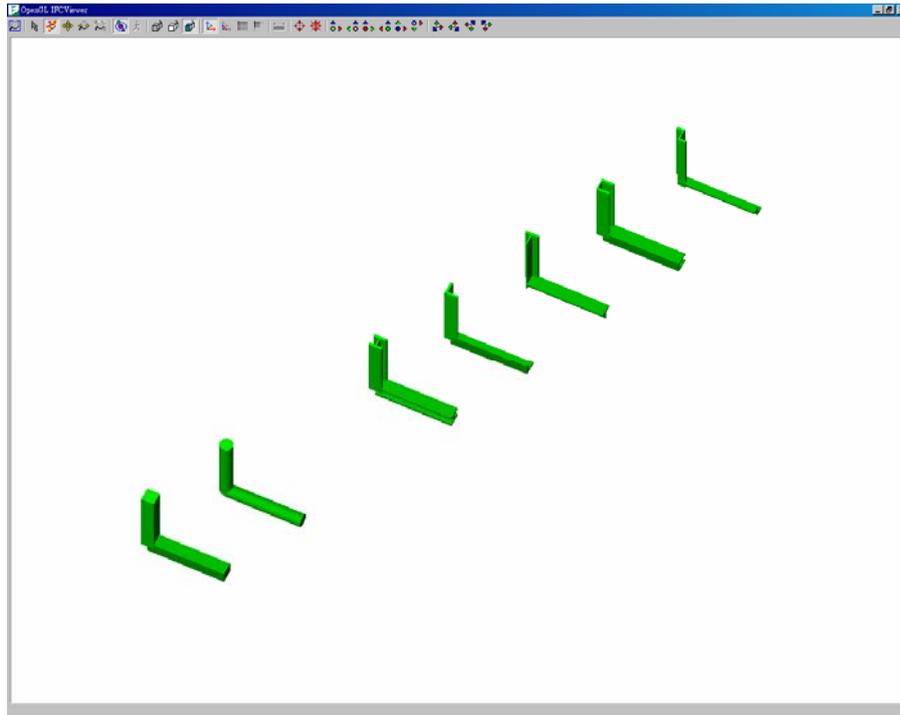


圖 5.35 IFCUser_ArchiCAD_model 以 IfcStoreyView_3D 視圖

