

國立交通大學
土木工程研究所
碩士論文

應用 IFC 於規範自動審查系統-
RC 柱構件之研究

Application of IFC in automated code checking
system against RC column members



研究生：周承禹

指導教授：洪士林 博士

中華民國 九十七 年 九 月

應用 IFC 於規範自動審查系統- RC 柱構件之研究

學生：周承禹

指導教授：洪士林 博士

國立交通大學土木工程學系

摘要

國際標準組織 ISO 制定了許多的標準涵蓋各個領域，而標準的推行將隱含著技術的領先與市場的競爭，所以標準的制定必須是嚴謹的。另一國際組織 IAI 則推動以工程應用領域為主的資訊標準，在工程應用領域的資訊標準的推動下，解決了建築物生命週期內資料交換與傳遞時所造成的成本浪費問題，IFC(Industrial Foundation Classes)標準格式就是目前工程界常用的標準格式之一，而剛興起的 BIM(Building Information Model)概念也是以 IFC 標準格式為基礎所發展出來的，BIM 概念有別於傳統 CAD 繪圖軟體，是以物件的方式來處理建築構件，將構件的資料以屬性的方式存在物件中。新加坡所使用的電子化法規自動審核系統 (e-PlanCheck)同樣也是以 IFC 檔案格式作為標準格式，上傳工程方案的圖檔，由電腦依照法規來檢核，加快了審核的速度，從原本平均兩個禮拜的審查時間縮短為二個工作天。

本研究希望利用 BIM 的物件導向特性與 IFC 標準資料格式，發展一套線上的規範自動審核系統，以我國的建築規範作為審核的標準，使用 IFCEngine OCX&DLL 程式作為輔助開發的工具，這是一個免費的 IFC 實體檔案存取的應用程式介面，我們將以此為基礎，加以開發與應用，來完成我們的審查系統。但是基於時間的限制，決定以「RC 柱構件」為主題，將 RC 柱構件的相關規範寫入系統，作為審核重點。本研究額外撰寫了一個可設計柱構件之柱筋且符合 IFC 格式的程式，來彌補目前建築 CAD 繪圖軟體對 IFC 鋼筋支援的不足，也讓本研究的規範自動審核系統有鋼筋的 IFC 檔案可以傳入作檢核。

關鍵字：IFC、BIM、IFCReinforcingBar、自動規範審查

Application of IFC in automated code checking system against RC column members

Student: Cheng-Yu Chou

Advisor: Dr. Shih-Lin Hung

**Department of Civil Engineering
College Engineering
National Chiao Tung University**

Abstract

Standards in International Organization for Standard (ISO) cover in several domains. Establishing and following standards implies leading of technology and competition of market. Therefore, creating standards to lead engineers or companies follow is an essential task in industry. International organization IAI also develops information standard about domain of engineering application primarily. Following corresponding standards for design results in cost saving of information exchange during the life cycle of a building. In engineering construction industry, IFC (Industrial Foundation Classes) is one of most frequently applied standard formats. BIM (Building Information Model) is a novel CAD concept, developed with the basis of IFC standard format. In BIM, any member for a building is viewed as an object including conventional drawing geometry data with material, analyzed results, etc. information and stored as attributes in the object.

The objective of this research is to develop an on-line automated code checking system with Taiwan's building regulation by using property of BIM and IFC standard format. Herein, freeware IFCEngine OCX&DLL program is employed as APIs platform (Application Programming Interface) to develop the system. Due to time constraints, this work only focuses on code checking of RC column members for a given building. Therefore, the corresponding building regulations about RC column members are re-coded in a data base. Although the entity of IFCReinforcingBar had been defined and released in IFC, it is rarely applied in CAD system. Therefore, a program was developed for design an IFC-supported RC column member model with corresponding reinforcing bar in order to cover the drawback of the current CAD software. Reveal from study cases, the system demonstrated the feasibility of code checking for column members for a given building with IFC supported format.

Keywords: IFC; BIM; IFCReinforcingBar; automated code checking

誌謝

研究所兩年即將結束，在交大六年的求學生涯也將劃下句點，研究所的學習方式有別於大學時期，不再是一味的接收別人給的，而是要有獨立思考的精神，所以我在這兩年內成長了許多，也學到了不少，其中最要感謝的人是我的指導老師洪士林博士，無論是在修課或是做研究的方向，老師都給了我很大的自由度，使我能夠發揮所長，同時也從旁給我指導與建議，所以本研究能夠順利完成，都要感謝我的老師，在此祝福老師身體健康、青春永駐。

此外，我還要感謝土木所資訊組一哥沈秉廷，在我多次遇到難題而不知道如何是好的時候，總是能夠給我一些意見，讓我可以迎刃而解，同時也是本研究得以完成的關鍵人物。再來要感謝我的室友彥忠、文凱、岳勳，陪我在無聊與苦悶的時候打打三國，舒解壓力，還有戰友智仁、浚昇、聰吉、國維學長。感謝國際事務處的美女姊姊妹妹們，讓我在打工的時候不會感到無聊。感謝同研究室的怡廷、世賢陪我度過熬夜看洋基、趕論文的日子，感謝志銘、彥伶、智中學弟妹們，有了你們實驗室多了歡笑。感謝家族的元旻學長、伯傑學弟、千惠學妹，給了我家族的溫暖，也不忘督促我要順利畢業。最後我要說，有你們真好。

目錄

摘要	i
Abstract	ii
誌謝	iii
目錄	iv
表目錄	vii
圖目錄	viii
第一章 緒論	1
1.1 研究背景	1
1.2 研究動機與目的	3
1.3 研究方法	4
1.4 論文架構	5
第二章 研究相關知識	7
2.1 IAI 與 IFC 簡介	7
2.2 BIM 簡介	9
2.3 CORENET	11
2.3.1 CORENET e-Submission	12
2.3.2 CORENET e-PlanCheck	12
2.3.3 CORENET e-Info	13
2.4 IFC 相關應用研究	13
第三章 系統架構分析	17



3.1	整體系統架構 v.s.實作系統架構.....	17
3.2	IFCEngine OCX & DLL 介紹.....	17
3.3	DLL 說明.....	21
3.4	OCX 說明.....	22
3.5	CGI 說明.....	23
第四章 系統實作與開發.....		25
4.1	檔案上傳系統.....	25
4.2	IFC 格式分析與資料擷取.....	26
4.3	柱筋 IFC 檔產生器.....	27
4.4	IFC Engine DLL&OCX 新增功能.....	29
4.5	梁柱板構件檢核之功能.....	30
4.6	柱筋相關規範與檢核之功能.....	32
4.6.1	檢核一：柱筋之鋼筋比.....	32
4.6.2	檢核二：相鄰箍筋與繫筋之最大間距.....	33
4.6.3	檢核三：橫向鋼筋之間距.....	33
4.6.4	檢核四：彎鉤長度.....	33
4.6.5	檢核五：保護層厚度.....	34
4.6.6	檢核六：柱筋之搭接.....	34
第五章 系統展示.....		35
5.1	網頁架構展示.....	35
5.2	系統環境.....	36
5.3	梁柱板構件檢核功能展示.....	37

5.4	柱筋檢核功能展示	38
5.4.1	設計模型一	39
5.4.2	設計模型二	41
第六章	結論建議與未來展望	43
6.1	結論	43
6.2	建議	44
6.3	未來展望	45
參考文獻	46



表目錄

表 4-2-1	一段 IFC 原始碼.....	48
表 4-2-2	IFCCOLUMN 的定義.....	48
表 4-6-1	縱向鋼筋之鋼筋比規定[18].....	49
表 4-6-2	橫向鋼筋之相鄰箍筋與繫筋之最大間距規定[18]	49
表 4-6-3	橫向鋼筋之間距規定[18].....	49
表 4-6-4	標準彎鉤之規定[18].....	49
表 4-6-5	鋼筋保護層之規定[18].....	50
表 4-6-6	柱筋搭接之規定[18].....	51



圖目錄

圖 2-1-1	IFC 架構圖[5].....	52
圖 2-3-1	CORENET 的目的.....	52
圖 2-3-2	傳統規範審核流程示意圖[17].....	53
圖 2-3-3	自動規範審核流程示意圖[17].....	53
圖 3-1-1	整體系統架構 OVERVIEW	54
圖 3-1-2	整體系統架構與實作部份比較	55
圖 3-2-1	左:拉遠功能(ZOOM OUT) 右:拉近功能(ZOOM IN).....	56
圖 3-2-2	物件旋轉功能(ROTATE).....	56
圖 3-2-3	物件拖移功能(PAN).....	57
圖 3-2-4	RESET FRONT 功能，重置從物體正面觀察	57
圖 3-2-5	RESET SIDE 功能，重置從物體側面觀察.....	58
圖 3-2-6	IFC ENGINE DLL&OCX 的原有功能.....	58
圖 3-4-1	IFC ENGINE DLL&OCX 運作過程.....	59
圖 3-4-2	註冊 OCX 檔.....	59
圖 3-4-3	將 OCX 檔、DLL 檔與 INF 檔壓縮打包成 CAB 檔	60
圖 3-5-1	CGI 運作流程與原理	60
圖 4-1-1	上傳系統流程圖	61
圖 4-1-2	以訊息視窗來警告使用者只能傳入 IFC 格式的檔案.....	61
圖 4-1-3	檔案上傳詳細內部運作流程	62
圖 4-2-1	IFC 檔案的內容.....	63
圖 4-2-2	IFCQUICKBROWSER 軟體開啟 IFC 檔	64
圖 4-2-3	IFCCOLUMN 結構展開之樹狀結構.....	64
圖 4-3-1	原始碼編譯成 CGI 之過程	65
圖 4-3-2	表格資料與 CGI 技術內部參數傳遞過程	65

圖 4-3-3	目前業界所用鋼筋圖與鋼筋表示法 I	65
圖 4-3-4	目前業界所用鋼筋圖與鋼筋表示法 II	66
圖 4-3-5	IFC 鋼筋種類 BARROLE 之列舉型態	66
圖 4-3-6	IFC 鋼筋表面種類 BARSURFACE 之列舉型態	67
圖 4-3-7	柱筋產生器執行時參數輸入的過程	67
圖 4-3-8	命令列下編譯 RBAR.C	68
圖 4-3-9	命令列下編譯 UTIL.C	68
圖 4-3-10	命令列下連結 RBAR.OBJ 與 UTIL.OBJ 成為 CGI 檔	68
圖 4-3-11	IFC 資訊架構圖	69
圖 4-3-12	加入鋼筋的實體 IFCREINFORCINGBAR，並連接	69
圖 4-3-13	鋼筋 IFC 產生器所產生的 IFC 檔	70
圖 4-4-1	新增功能-抽出柱資料	71
圖 4-4-2	新增功能-抽出梁資料	71
圖 4-4-3	新增功能-抽取鋼筋資料	72
圖 4-4-4	新增功能-抽取樓板資料	72
圖 4-4-5	隱藏物件功能(HIDE)	73
圖 4-4-6	顯示所有物件(DISPLAY ALL)	73
圖 4-4-7	被點選的物件變成黃色	74
圖 4-5-1	大梁跨小梁示意圖，左為俯視圖，右為側視圖	74
圖 4-5-2	梁身超過柱俯視圖，右邊兩根梁，梁身都超過柱	75
圖 4-5-3	梁跨柱不在同一直線上之俯視圖	75
圖 4-5-4	因為要做樓梯之故要去掉樓板	76
圖 5-1-1	紅色框選部分為本研究實作的部份	76
圖 5-1-2	系統實作首頁，可上傳所要檢核的 IFC 檔	77
圖 5-1-3	點選瀏覽鈕可選擇要上傳的檔案	77

圖 5-1-4	矩形柱與柱筋設計的表格網頁(RBAR.HTML)	78
圖 5-1-5	使用者設計的柱筋模型可點選下載	78
圖 5-1-6	模型之 3D 展示	79
圖 5-1-7	柱半透明化來觀察柱內部的鋼筋分布	79
圖 5-3-1	以 ARCHICAD 繪製大梁跨小梁模型	80
圖 5-3-2	以紅色來表示錯誤的地方，並在右方顯示警告錯誤的數目 .	80
圖 5-3-3	檢核出大梁跨小梁，在右方顯示對錯誤物件的描述。	81
圖 5-3-4	以 ARCHICAD 繪製梁身超過柱模型	81
圖 5-3-5	以紅色來表示錯誤的地方，並在右方顯示警告錯誤的數目 .	82
圖 5-3-6	檢核出梁身超過柱，在右方顯示對錯誤物件的描述。	82
圖 5-3-7	以 ARCHICAD 繪製兩根梁跨柱不在同一直線上的模型	83
圖 5-3-8	以紅色來表示錯誤的地方	83
圖 5-3-9	檢核出左邊梁與右邊梁不在同一直線上。	84
圖 5-3-10	以 ARCHICAD 繪製樓層樓板中空的模型	84
圖 5-3-11	以紅色來表示錯誤的地方，並在右方顯示警告錯誤的數目	85
圖 5-3-12	檢核出樓板可能會遭受剪力破壞。	85
圖 5-4-1	柱與柱筋的設計參數表單	86
圖 5-4-2	根據實際柱配筋圖，來設計一柱筋模型，如框選的斷面	87
圖 5-4-3	選定柱配筋圖的某一斷面，填入參數，來設計柱筋模型	88
圖 5-4-4	檢核出鋼筋比不符，並在右方顯示警告錯誤的數目。	88
圖 5-4-5	點選紅色的錯誤物件後，右下會有對錯誤物件的描述	89
圖 5-4-6	將視窗深入柱之內部查看鋼筋分布	89
圖 5-4-7	跳出新視窗，顯示鋼筋比之規範	90
圖 5-4-8	跳出新視窗，顯示繫筋與閉合箍筋相鄰各肢中心距之規範 .	90
圖 5-4-9	檢核出橫向鋼筋間距不符，在右方顯示警告錯誤的數目。 .	91

圖 5-4-10	點選紅色的錯誤物件後，右下會有對錯誤物件的描述.....	91
圖 5-4-11	跳出新視窗，顯示橫向鋼筋間距之規範.....	92
圖 5-4-12	檢核出彎鉤延伸長度不符。.....	92
圖 5-4-13	跳出新視窗，顯示標準彎鉤之規範.....	93
圖 5-4-14	檢核出保護層不足，在右方顯示警告錯誤的數目。.....	93
圖 5-4-15	點選紅色的錯誤物件後，右下會有對錯誤物件的描述。...	94
圖 5-4-16	跳出新視窗，鋼筋保護層之規範.....	94
圖 5-4-17	檢核柱筋鋼筋比，沒有錯誤.....	95
圖 5-4-18	檢核相鄰箍繫筋的最大間距，沒有錯誤.....	95
圖 5-4-19	檢核橫向鋼筋間距，沒有錯誤.....	96
圖 5-4-20	檢核鋼筋彎鉤長度，沒有錯誤.....	96
圖 5-4-21	檢核保護層厚度，沒有錯誤.....	97
圖 5-4-22	跳出新視窗，柱筋搭接之規範.....	97



第一章 緒論

1.1 研究背景

國際標準組織ISO(International Organization for Standard)已將開放文件格式ODF(Open Document Format)列為正式的國際標準，而ODF可運用延伸標籤語言XML(Extensible Markup Language)來定義工作的文件格式，例如文字、圖表、繪圖文件等等，透過XML對文件的定義(Schema)，使得文件的資料可以獨立出來不受應用程式的限制，並依資料標準來相互交換使用。這個文件標準已經應用到多種不同的專業領域，聯合國的貿易簡化與電子商務中心UN/CEFACT (United Nations Centre for Trade Facilitation and Electronic Business)正如火如荼的在推動國際資訊交換標準的訂定與推廣，其範圍包括國際貿易、軍事、醫藥、化學、一般商務、物流、營造建築業等等，也遵循ISO程序推動國際認同成為國際標準。

而另一國際組織IAI(International Alliance for Interoperability)則推動以工程應用領域為主的資訊標準，重點著重在工程設計的本體，也就是3D的幾何資訊模型，其資料標準為IFC(Industrial Foundation Classes)格式，目的不僅是要提高工程設計的效率及正確性，也透過幾何的資訊模型，在設計過程中不同的專業領域，如建築設計、水電空調、管線配置、成本估價、營建管理等等，透過此標準格式互相使用工程的資料與訊息，達成資訊共享與再利用。此IFC格式顯然已成工程業界的標準，許多工程應用軟體已開始提供IFC格式資料的輸出輸入介面，例如AutoCAD、MicroStation、ArchiCAD等等。從上面敘述可看出標準的推行將隱含著技術的領先與市場的競爭，可謂兵家必爭。

由這些國際的走向趨勢及全球性的資訊交換組織來看不難發現，在全球

化、國際化的發展下，跨國界、跨企業、跨產業、跨專業的資訊交換需求與日俱增，由國際資訊交換標準組織所制訂的資料格式也顯得格外重要，在這強調跨領域、跨平台、跨國際的資訊交換強力趨勢下，是各領域應努力的方向[1]。

而由IAI提出的建築資訊模型(BIM, Building Information Model)是以3D物件代表建築物各元件的資料模型，在建築物整個生命週期內所有的資料與訊息都儲存在這個模型中，在建築生命週期裡任何領域或任何軟體都可以在這模型裡獲得所需的資料與資訊，而這個資料的標準儲存格式就是IFC。一份來自英國的報告表明，在早期，建築項目的成本大約有30%損失在工程建設的訊息交流上[2]，IFC資料標準的出現解決訊息交換與共享的問題。而目前國際間，各工程專業領域的設計知識亦多以IFC的格式儲存，故IFC已廣泛成為BIM的資料交換格式及建構BIM的基礎。

新加坡推行的CORENET (COstruction and Real Estate NETwork)系統，中譯為建築與房地產網路，在2004已達完善階段，至今已成熟營運，這系統以IFC資料格式為基礎，發展電子化檔案繳交(e-submission)系統與自動化的線上規範審核(e-plan check)系統，並提供線上審查進度查詢，有了這個系統，建築設計公司在呈交建築藍圖時，無須走遍政府所有部門一一辦理申請手續，可以透過電子化的網路上傳繳交系統遞交建築藍圖、結構設計圖與申請各種許可證明。此電子系統每年為新加坡政府、建築公司與業主節省了大筆的金錢與時間，加快了審核的速度，從原本平均兩個禮拜的審查時間縮短為二個工作天[3]，對於各國的土木建築業來說，是一個很值得仿效學習的案例。

1.2 研究動機與目的

基於上述的國際標準發展之背景，本論文研究動機分敘如下：

動機一，在 BIM 概念的推行下，許多的 CAD 繪圖軟體運用 BIM 將圖形資料與文字資訊包含在一個模型中，不再像以前用標注或註解等方式來描述物體，而是將資訊變成有意義的物件屬性記錄在物件中，本研究就是要利用這種模型中搭載圖形資訊的方式，藉由程式來擷取出模型裡的資料，進而對於此模型進行分析。

動機二，在了解新加坡的營建規範電子化自動審查系統後，是否台灣也可以發展一套這種建築營建規範自動審查的系統？不僅節省了人力與成本，縮短審核的時間，更可以使工程有效率的運作下去，有這麼多的好處，我們也應該仿倣之，開發一個合適的自動審核系統。

動機三，資訊的標準化是一種必然的趨勢，本研究選擇越來越受歡迎的 IFC 資料標準格式來當作系統程式讀入的檔案格式，如此一來，多款建築 CAD 繪圖軟體只要輸出成 IFC 檔案格式(.ifc)都可以由系統的程式來讀入進行檢核。

動機四，雖然 IFC 對各建築構件都有完整的定義，包括鋼筋部分，但是目前的 CAD 繪圖軟體都沒有相對應的鋼筋元件，所以也無法使用 IFC 格式來描述鋼筋，這是非常可惜的，所以如果能在現存的 IFC 檔自行加入 IFC 對鋼筋描述的實體，是否可行還是未知的，但是是值得研究的。

綜合以上各點，本研究的目的是要利用 BIM 的特性與 IFC 的標準化，開發一個營建規範自動審查系統，以我國的建築規範為審查準則，將建築 CAD 繪圖軟體設計的 3D 模型圖檔傳入系統，作與規範比對的各項審核，但是由於時間、資源、技術等等的限制，加上結構審查項目繁多，決定先

將重點放「RC 柱構件」的部分，以「RC 柱構件」作為規範審查的對象，將柱構件的相關規範內容寫入系統。並撰寫一個程式用來設計鋼筋構件，最重要的是此程式將以 IFC 對鋼筋的定義之格式來儲存，以彌補目前 CAD 繪圖軟體對 IFC 支援之不足，當然也可藉此程式，設計包含鋼筋的柱構件，傳入系統做柱構件之柱筋的檢核。

1.3 研究方法

想要對建築模型作審核，必定要知道模型中物件的資訊(如座標、尺寸等等)，所以我們要從有搭載圖形屬性資訊的 BIM 模型裡擷取出所需要的資料，要擷取資料就必須了解 IFC 這檔案格式的定義與語法(schema)，若要自己寫程式來解析 IFC 檔案實在是一個龐大的工程，所幸網路上已經有一些 IFC 的研究者開發出許多 IFC 相關的應用程式，其中 TNO 公司所開發的應用程式最能符合本研究的需求，也適合本研究的目的做深入的應用。TNO 公司所提供的應用程式介面(API) —IFC Engine OCX&DLL，這是一個免費的 IFC 實體檔案存取的應用程式介面，而此應用程式介面已經提供了解析(parse)IFC 檔案的函數，所以可以直接使用這些函數就可以擷取 IFC 檔案中的一些屬性或資料，追蹤(trace)它的 C++ source code，了解其函數的意義與功能，再進而改寫或增加所需功能的函數，將規範所規定的內容寫進程式裡，輸入擷取到的資料進行檢核，本研究主要是針對「RC 柱構件」部份進行檢核，由於 ArchiCAD 等繪圖軟體無法畫出支援 IFC 格式的鋼筋，這部份將另外寫一個可以設計鋼筋嵌入柱子裡並符合 IFC 檔案格式的程式，在此暫時稱之為柱筋產生器，此程式由 C 語言撰寫成，並以 CGI 的方式由網頁表格填入鋼筋與柱子的資料，產生一個使用者所要求的 IFC 檔，至於其他檢核梁柱不合理的案例模型，則可以由

ArchiCAD 軟體來繪成，並匯出成 IFC 格式，再由 IFC Engine OCX&DLL 此應用程式讀入，進行一系列的審核。

1.4 論文架構

本論文其餘各章的內容蓋述如下：

第二章:說明與本研究主題相關的知識以及文獻，內容有 IAI 公司所推行的 IFC 資料格式與 BIM 建築資訊模型的介紹。新加坡 CORENET 系統的深入介紹。

第三章:本章節在說明整體系統架構之概念與有實作的系統架構內容，並介紹用於開發系統之程式 IFC Engine OCX&DLL 的基本功能，並說明 OCX 與 DLL 的原理，最後說明系統用到的網頁技術 CGI。

第四章:本章節在說明開發系統的過程，以動態網頁技術 JSP 實作檔案上傳的功能，改寫與新增應用程式介面 IFC Engine OCX&DLL 的功能，再分析 IFC 格式來撰寫柱筋產生器並用 CGI 技術應用於網頁，之後說明梁柱板空間分布不合理的審核演算法，最後說明如何將規範中的柱筋規範寫成程式來檢核。

第五章:本章節先對實作系統做概觀說明，再來用 ArchiCAD 繪製一些實際模型，傳入檢核系統做檢核，並展示結果，最後用系統開發的柱筋產生器產生柱筋模型帶傳入系統做檢核，並展示結果。

第六章:此章節綜合了幾項結論，並對於可改進的部份做了建議，更進一步做出對於未來的期許。



第二章 研究相關知識

本章節介紹與本研究相關的知識及資訊，包括 IAI 的成立目標及發展史，IFC 資料格式的發展與簡介，BIM 的內容說明及優點，最後介紹新加坡所推行的 CORENET System。

2.1 IAI 與 IFC 簡介

IAI 全名為 International Alliance for Interoperability，目標為致力於全世界範圍內建築與設備管理的資訊共享，提供一套統一標準的資訊共享的過程，並對工業基礎類別進行定義，促進整個建築生命週期內訊息的交流，時間的縮短，以及品質的改善。

IAI 的發展起源於 1994 年，美國 12 家公司共同研究不同應用軟體在協同工作的可能性，他們覺得協同工作是可行的，而且這種協同工作能力會帶來可觀的經濟效益。1995 年克服了 kernel 的問題後他們在建築工程系統研討會上展示了他們的研究成果，這成果就是一套簡稱 IFC 的資料交換格式，IFC 格式使得資料格式有了依據的標準，資料的標準化、一致性使不同軟體間可相互交換資料無需再各自建立資料。很多與會的組織對此都表示出了濃厚的興趣，有意加入，希望能共同研究。之後，他們決定對外開放，對建築工程與設備管理領域組織開放，對所有軟體開發商開放，同年 10 月，他們在北美成立 IAI 組織。IAI 成員們很快發現這是全球工業的問題，隨後他們將軟體間協同工作的思想推廣到其他國家，很快的其他國家也相繼成立了 IAI 分部。1996 年在倫敦召開了第一次的 IAI 國際會議。1997 年 IAI 發表了第一板 IFC1.0。

在 IFC1.0 資料標準化的範圍主要包括下列四個領域：建築

(Architecture)、暖氣流通與空調(HVAC)、營建管理(Construction Management)、設施運作與管理(Facilities Operation and Management)，清楚界定IFC的資料標準並非單一應用領域的資料結構，而是跨設計、施工、營建、管理、維護等跨平台也跨專業領域的資訊結構。之後發展之IFC1.5隨即將此一標準資訊結構匯入繪圖商用軟體，從此進入商用工程設計市場，目前IFC資料結構已重新改版為IFC2.0，並有多種延伸板本IFC2x板(IFC2.0Extension)。目前最新板為2x3 Final，本研究所用的IFC板本皆為2x3 Final。IFC各板本所能容納各專業領域的程度以建築之資訊最多，營建管理、物料庫、營運維護、估價發包、水電空調分析的資訊則次之，而以地理資訊系統(GIS)的資訊最少，但目前將GIS納入IFC是IAI近年來推動的重點之一[4]。

IFC (Industry Foundation Classes) 中譯為工業基礎類別，是一種塑模語言，目前主要用於建築、營建工程與設備管理等領域，目的在於讓各種在此領域內的應用軟體可以遵循同一種資料格式，或使領域內不同的 BIM 系統有一個資料交換的標準。

IFC 以物件導向(Object Oriented)為基本概念，參考了 STEP 部分標準，為了實作考量也參考了物件導向語言 C++程式語言的一些特性。以架構而言可以分為四個層次，由上而下分別為：領域層(Domain Layer)、共享交換層(Interoperability Layer)、核心層(Core Layer)、資源層(Resource Layer)，每層分別定義了不同的種類的資料類型(Data Type)與實體(Entity)，如圖 2-1-1 所示[5]。

◆ 領域層(Domain Layer)

這層為定義 AEC/FM(Architecture ,Engineering , Construction and Facilities Management)領域內各專業領域的實體，包含:建築、結構分析、

營建管理、設施管理、機電設備、水電空調、配管工程等等專業領域的實體。

◆ 共享交換層(Interoperability Layer)

這層定義能夠在 AEC/FM 領域內做資訊共享與交換的共同實體，例如：梁、柱、門、窗、空間等等資訊。

◆ 核心層(Core Layer)

這層定義 IFC 基礎的實體，此層的實體定義了許多共同的介面，它們可被資訊交換層或專業領域層的實體參考或繼承。

◆ 資源層(Resource Layer)

這層定義較一般性的實體，類似 STEP 的整合性一般資源，通常它們都是依附或參考在較高層次的實體上。

目前支援 IFC 格式的建築 CAD 繪圖或分析軟體有逐漸增多的趨勢，就目前所知的有 SAP2000、ArchiCAD、AutoCAD、MicroStation 等軟體支援，但是 IFC 檔案於軟體間的傳遞會有遺漏的情形，同一個軟體匯出成 IFC 格式再以 IFC 格式讀入都可能會有少許遺漏的情況，各軟體間也由於支援 IFC 程度的不同，有些元件沒定義或定義不同，造成資料遺漏或錯誤的情況，所以這一部份是 IFC 還不健全的地方，也是未來各 CAD 軟體商需要努力的地方，在本研究中只使用 ArchiCAD 軟體來繪製所需要的模型，不需要在軟體間轉換，減少了資料遺失的情況。

2.2 BIM 簡介

建築資訊模型(BIM, Building Information Model)是在整個建築物生命

週期中各專業領域的設計者提供該領域所貢獻的建築資料與專業知識，把這些資料儲存起來就是一個建築資訊模型，其他領域可從建築資訊模型中獲取所需的資料與欲處理的建物的資訊，達到資料共享與資訊再利用，所以建築資訊模型可以說是一個資料庫，將該建築物生命週期裡的所有資料與訊息存在其中，也是各個專業領域設計者知識與經驗的累積所匯集的知識庫。而建築 CAD 繪圖軟體從 2D 平面圖形的表示轉變為 3D 的模型呈現，這種建築資訊的模型化也是 BIM 的另一種意義。

◆ BIM 概念與傳統 CAD 繪圖軟體的比較

傳統建築 CAD 軟體繪圖用點、線、標注或註解來表示圖形之建築資訊。例如梁之配筋圖，從這些圖中無法直接取得有意義的資訊(例如梁或鋼筋斷面尺寸、長度等等)，必須經由人為的判讀或計算才能得到圖檔裡的資訊，然而 BIM 的世界裡，梁、柱、板、牆、門、窗等等都是一個個獨立物件，它們的屬性就是資訊，可以直接取得，這樣一個一個物件組合起來，改變各物件的屬性來達到想要繪製的結果，這樣就是一個建築資訊的模型。

◆ BIM 的優點

建築資訊模型(BIM)，藉由建立3D的幾何模型也同時建立資訊模型的成果，其所帶來的優點有[6]:

1. 工程設計具體化：可將設計方案具體化、視覺化、3D化，也就是可以以動畫或虛擬現實場景的方式呈現。
2. 避開工程風險：在實際進入施工階段之前可就構件衝突問題或有錯誤的地方事先偵測並作預防性的處理。
3. 物件數量計算：可直接從模型中抽取必要資料並加以統計，將其表格化

的呈現，例如門窗數量的計算統計，可以輸出以excel表格的方式表示。

4. 變更設計時保持一致性與協調性：建築資訊模型能自動地使專案的所有資訊協調一致，建築模型的任何改動都會立刻反映到整個專案產生的檔案中。
5. 提昇設計品質，縮短設計週期，效率提高：設計者更能具體掌握其設計成果的視覺效果、造價成本等，而且由於在使用的同時動態生成專案的有關資訊錯誤和資訊遺失會大幅度減少，因此提高了設計品質，減少了資料往返的次數。
6. 屬性資料的直接使用：有別於傳統CAD軟體，模型的資料與訊息直接以屬性的方式存於BIM中，所以可以很方便的取用，無須經過人為的判讀。
7. 工程資料再利用：可將設計資訊完整的保留在建築資訊模型之中，從早期方案階段的構思到詳圖設計階段的最終變化都得予以保存，方便其在整個工程生命週期中各個階段來使用。

2.3 CORENET

CORENET全名為COstruction and Real Estate NETwork，中譯為建築與房地產網路，主要以資訊科技為主，是新加坡國家發展部所發展的一個計畫，CORENET的目標是發展一個整合建築生命週期的四個基本過程—設計(Design)、實現(Procure)、建造(Build)、維護(Maintain)的資訊科技系統(如圖 2-3-1)，而在設計這階段裡已經發展了三個系統且已成熟的運作，這三個系統分別為CORENET e-Submission，CORENET e-PlanCheck，CORENET e-Info[7]。

2.3.1 CORENET e-Submission

是一個以網路為基礎的上傳繳交系統，可透過這系統上傳計畫方案的圖檔(例如設計圖、工程圖、建築圖等等)與文件給工程審核機構，做各方面的審核，包括計劃審核、建築審核、結構審核、管線空調與防火安全規範等等。

◆ e-Submission的優點:

1. 數位資料取代了紙本資料，減少了印刷與紙張，對於全球減碳節能的今日，可以說是非常重要的。
2. 一天24小時，一個禮拜7天，任何時刻只要你可以上網都可繳交資料，打破以往只能在上班時間繳件的限制。
3. 同一份文件可能要送到多個審核單位(如圖 2-3-2)，所以需要浪費很多人工來收件，e-Submission算是一個虛擬的收件者。
4. 文件在審核單位之間的傳遞也是以人工傳遞，這不僅增加人事成本，也增加人為遺失資料的可能性，e-Submission解決了這些問題(如圖 2-3-3)。

2.3.2 CORENET e-PlanCheck

這系統的目的是提供一系列的資訊科技應用程式，來自動地檢核透過上述e-submission所上傳的建築相關電子圖檔，是否有符合規範的要求，這部份也是本論文所欲探討研究的議題與實作的重點。

◆ e-PlanCheck的優點[8]:

1. 自動檢核取代了人工檢核，傳統人工檢核是人力密集且耗費時間的工

- 作，能以電腦來取代可以說是省時又省工。
2. 審核人員對規範的解釋不一致，當規範語義有模糊不清或模稜兩可時，不同的審核人員可能有不同的審核結果，將規範寫入程式後，系統會以相同的標準來審核，不會有上述的情況發生。
 3. 規範審查是件乏味的工作，以人工來檢核可能會發生人為檢查的失誤，或審查人員被收買、放水或故意刁難的情形，審查自動化後就可避免以上的情形，且可以刺激建築業，使其朝向採用物件導向的電腦輔助設計 (CAD)。

2.3.3 CORENET e-Info

本系統提供一個中央的儲藏庫，儲藏建築物或結構等相關的資料，之前上傳過的資料，可以在任何時間、任何地點透過網際網路來存取。此儲藏庫也儲存新加坡建築規範與新加坡各房屋與建築管理單位的公告，會自動地寄送關於房屋與建築業的新聞消息給它的訂閱者。

2.4 IFC 相關應用與研究

受到 BIM 觀念廣為被採用的影響，IFC 這建築資訊交換標準也受到矚目，成為近年來熱門的檔案標準格式之一，研究 IFC 與 BIM 的相關應用與文獻也就紛紛出爐，本研究參考了幾篇 IFC 相關的研究的內容，得到了一些靈感與方法來幫助研究，使得本研究得以順利進行。

Lee、Chin 與 Kim(2003)發展一個整合的資訊系統稱為 DIMS(Design Information Management System)，著重在於以 IFC 為基礎的繪製資訊之呈現，此系統結合了排程、評估估價與結構工程資訊，整合來自各方資訊

的 DIMS 透過對建築元件階層式的分類可從多個方面呈現計畫的資訊，DIMS 也可計算每一種建築構件的數量，並幫助使用者於 3D 系統之中更清楚辨認出與之相對應的 3D 圖形物件[9]。

Halfawy 與 Froese(2005)使用 IFC 於 AEC 專案資料與進程之整合，AEC 工業整合的趨勢正增加專案系統的實作與使用的需求，標準資訊模型被廣泛地當作是主要的發展整合 AEC 系統的科技，多數重要的模型也都是以 IAI 所發展的 IFC 作為標準之資料格式，這篇文章呈現了 IFC 類別的實作，用來完全地支援 AEC 專案的資料與進程，也討論連結專案設計、排程、花費細項與不同系統間雙向的資料交換[10]。

曾旭東與譚潔(2006)發表對 IFC 標準的建築結構模型的自動生成，建築模型之自動生成為符合多種結構分析與設計軟體的結構模型，並通過實例驗證此研究方法的實用性與可行性，結果此方法為各種設計軟體間訊息的交換與共享提供了一種通用的解決方案，為企業內部實現建築設計集成化提供了技術保證。最後討論了該研究方向中對於 IFC 標準之建築模型的結構偏心、節點連接、荷載處理 等問題以及後續研究方向[11]。

沈秉廷(2007)運用物件導向技術於 IFC 建築資訊，此研究由物件導向的觀點探討 IFC 規格設計與實作方法，介紹 IFC 實體檔案存取應用程式介面並探討其 IFC 資訊存取方式。此研究運用物件導向技術，撰寫程式將 IFC Entity 規格轉換為 .Net Framework 類別(Class)，再以類別庫(Class Library)封裝。此類別庫提供較直觀的 IFC 應用程式開發方式，並能夠運用物件導向於系統開發。最後使用所建置之類別庫開發以外掛(Plug-In)為架構的 IFC BIM 系統，並開發一些外掛(Plug-In)元件進一步示範如何應用類別庫，例如：3D 建築模型瀏覽、梁柱斷面資訊瀏覽、建築物框架建模等等外掛(Plug-In)元件[12]。

Faraj 與 Alshawi(1999)等人發展一以網頁與 IFC 為基礎的協同工作的結構工程運算環境，此研究討論一個給結構協同工作之發展與實作的環境，此環境為英國 Salford 大學中一個為人所熟知的以網頁為基礎的共享專案的環境，簡稱 WISPER (Web-based IFC Shared Project Environment)，這環境有三層的架構，分別為使用者介面、商業邏輯與資料庫，這環境支援 CAD 設計、視覺化的虛擬實境、評估、專案、詳細計畫書、供應商資訊，可說是一個網頁與 IFC 為基礎的分散式電腦整合環境。WISPER 使專案資訊透過 STEP PART 21 來進行檔案交換，透過 IFC 進行資料庫共享，同時很多網頁允許遠端互動，例如存取檔案，這提供了很大的彈性與可攜性，因此使得結構專業人員能夠分開地執行與管理他們的項目[13]。

邱奎寧與王磊(2004)在「IFC 標準的實現方法」中概要地分析了軟體中的 IFC 標準之實現方法與過程，IFC 標準即將作為中國大陸的國家標準，所以將 IFC 標準的應用於開發項目會涉及到的相關問題作探討，IFC 標準實現方法的內容有三個步驟：技術準備、方案設計、一致性測試。技術準備主要要解決三個問題：了解 IFC 的概況、學習 EXPRESS 語言、熟悉 IFC 整體框架。由於大陸地區之建築 CAD 軟體開發商普遍不了解 BIM 意義，對建築資訊的交換與共享也不熟悉，而這兩者正式 IFC 的內容，所以首要課題為瞭解 IFC 的概況。再者，IFC 標準本質為建築物與工程數據的定義，它採用了 EXPRESS 語言來描述與定義，學習 EXPRESS 語言可幫助我們更容易了解 IFC。而 IFC 的總體架構是分層的與模組化的，對於一般的應用開發只需要熟悉即可，不需要全然了解。方案設計的主要內容是設計資訊的共享與交換的方式，一是未知系統只要能夠使別 IFC 格式的文件就能夠與本應用系統共享與交換資訊。另一是透過 SDAI(Standard Data Access Interface)，未知系統透過此介面也能與本應用系統共享與交

換資訊。最後是一致性測試，要對實現 IFC 標準的應用軟體進行測試，無論何時、何地，測試結果都必須一致，並透過紀錄來檢查測試步驟的正確性，一致性測試在資訊共享與交換的“質”與“量”給予保證[14]。



第三章 系統架構分析

本章將對整體理念的系統架構作概觀的介紹，對要開發實作的系統架構進行分析與說明，並做兩者的比較。之後對開發工具用 IFCEngine OCX & DLL 做功能的介紹，並說明 OCX 與 DLL 原理，再對製作網頁用到的 CGI 技術應用做說明。

3.1 整體系統架構 v.s. 實作系統架構

本研究的整體系統架構如圖 3-1-1 所示，系統入口是一個登入(login)系統，為了保護系統的安全以及使用者管制，亦或是分辨使用者的身分(例如建築師、結構技師、業主等等其他相關人員)來開放不同的存取權限或功能項目，而設置這樣一個需要輸入帳號密碼的功能。登入後，可以依使用者權限的不同，有不同的功能選項，例如建築設計師可以上傳建築圖檔進行規範審核、構件數量計算與估價、工程進度與排程等等，其他功能有如查閱最新規範、建材最新單價、業界最新消息、結構分析等等，此完整的系統將會涵蓋很大的範圍。

然而基於時間的限制，本研究實作的系統架構如圖 3-1-2，只佔整體系統架構的一部分，此部分完成了上傳檔案系統與 RC 柱筋的檢核。並額外開發了產生柱筋 IFC 檔的程式，以彌補目前建築 CAD 軟體無法繪製鋼筋 IFC 檔的不足，本研究為整體系統架構的實現跨出第一步。

3.2 IFCEngine OCX & DLL 介紹

IFCEngine OCX&DLL 為 TNO 公司所開發的應用程式，分為 OCX 檔

(.ocx)與 DLL 檔(.dll)兩部分，是一個 IFC 資料存取的應用程式介面(API, Application Program Interface)，目前最新板本為 1.02，但是在實作時 1.02 板本尚未釋出，所以整個實作過程都是用 1.01 板本，1.01 板本可支援 IFC2x2 與 IFC2x3。

TNO 公司免費提供學術研究及非營利用途使用，若要使用在商業用途則必需付費。此應用程式為 Windows 平台下的 DLL 元件，使用 Microsoft Visual C++編譯器建置，使用此 API 需要 C 程式語言的基礎，因為其用到許多指標(pointer)來操作資料，而且 Visual C++與一般 C++尚有些許不同，這部份可以上網 msdn(Microsoft Developer Network)或找相關書籍參考。

IFCEngine DLL 根據 Standard Data Access Interface (SDAI, ISO 10303-22)設計，SDAI 定義為以 STEP 建模之模型的資料存取介面，這些資訊存取的方式相當低階，使用上繁雜，但效率高[15]。

IFCEngine OCX 為開發出來更容易使用 IFCEngine DLL 的 ActiveX 控制項，這 ActiveX 控制項能夠嵌在其他應用程式上，本研究即是將 OCX 嵌在 IE 網際網路瀏覽器應用程式中。而在此 OCX 中有 3D 模型瀏覽的 3D viewer 之功能，所以若要使用 IFCEngine OCX 必須安裝有 DirectX 8 或更新的板本。IFCEngine OCX 只支援部份的 SDAI 功能，但是 IFCEngine OCX 是公開原始碼的，所以使用者能夠任意修改或是新增所需功能[16]。

以下介紹一些有用到的 API:

sdaiOpenModelBN

用於開啟一個檔案，需指定檔案的所屬的 schema，回傳 ModelID。這邊我們用來開啟一個 IFC 檔。

sdaiCloseModel

用於關閉一個檔案，但這個檔將不被儲存。這邊我們用來將已開啟的 IFC 檔關閉。

initializeModelling

用於在程式的 3D viewer 啟動前做初始化的動作，決定 3D viewer 的規模 (scales)，也就是 3D 瀏覽器的大小，並回傳儲存 indices 與 vertices 所需的空間。

sdaiGetEntity

從 Model 中藉由 Entity 名稱取得 Entity 物件，回傳 InstanceID。



sdaiGetEntityExtentBN

從 Model 中藉由 Entity 名稱取得一組 Entity 物件，回傳 aggregate ID，再藉由 engiGetAggrElement 函式去取得 Instance ID

sdaiGetMemberCount

傳回特定聚合 Instance 的成員數目。在程式中有些 Instance 的資料型態是由多個 Instance 所組合形成的聚合型態(aggregate)，我們要知道是由多少個 Instance 所聚合才有辦法抓取聚合 Instance 中的特定成員，此函數輸入 aggregate ID 就可得知是由多個 Instance 所組合。

sdaiGetAttrBN

藉由 Instance ID、屬性名稱及類型取得該屬性的內容。這是一個很重要的函數，我們使用它來擷取 Instance 的屬性。

IFCEngine OCX&DLL 所提供的基本功能，可分成兩部份，第一部份是 3D viewer 展示，第二部分是雙擊(double click)物件就可以顯示該物件的基本屬性。

1. 3D viewer 展示

建築 CAD 繪圖軟體發展至今，幾乎都支援 3D 展示瀏覽的功能，就是 3D viewer，這可以方便我們以立體的角度來看我們從平面設計的物件，以確保畫出來的是我們所想要的，在 IFC Engine OCX&DLL 應用程式介面中也有提供 3D viewer 的功能，此 3D viewer 也提供一般 3D 瀏覽時所提供的功能，有拉近拉遠(zoom 如圖 3-2-1)、旋轉(rotate 如圖 3-2-2)、拖移(pan 如圖 3-2-3)等等功能，並有額外的兩個功能:正視觀察(reset front 如圖 3-2-4)與側視觀察(reset side 如圖 3-2-5)，讓我們方便調整方向以想要的角度來觀察物體。

而將檔案載入 3D 展示瀏覽器的是一個“Load IFC File”的功能，內部的語法是將“fileName”這個變數的值指向 IFC 檔所在的網路位置，例如 fileName='http://www.ifcbrowser.com/test.ifc'，所以如果要讀入新的 IFC 檔案，只需要將新檔案的網路位置傳給 fileName 即可。

2. 基本屬性擷取

IFCEngine OCX&DLL 提供一個基本的功能，就是用滑鼠指標雙擊(double click) 3D viewer 中的物件，這會觸發 Javascript 的事件(event)，

進而呼叫 OCX 中的 `doubleclick()` 這函數，此函數會回傳被滑鼠所點到的物件 ID，有了這 ID 就可當作參數帶入一些函數中求得該物件的一些基本屬性，例如 `sdaiGetStringAttrBN(id,attribution)`，而一般物件基本的屬性值有 `GlobalId`、`Name`、`Description`、`ObjectType` 等等(如圖 3-2-6 所示)。不過這些預設提供的資料對我們來說是不夠的，所以我們必須改寫或自行新增一些函數來擷取想要的資料，以利我們做一些運算與比較。

3.3 DLL 說明

DLL 全名為 `Dynamic Link Library`，中文譯為動態連結函式庫，DLL 也可以說是一種執行檔，但是 DLL 中的程式碼只能被其他執行檔或 DLL 執行，換一個角度看 DLL 扮演提供函數服務的角色，相對於動態連結函式庫，我們寫的程式大多是靜態連結的，因為函式的程式碼一經複製到執行檔就不能做改變，如果要更改，就必須重新編譯(`compile`)及連結(`link`)，靜態連結會在編譯階段就將程式所需要的函式加執行檔中，執行檔所須的記憶體空間自然就會增加；因此隨著系統越來越複雜，所需要的函式越來越多，這就衍生二個問題：

1. 當系統功能越複雜,隨著引用的函式數量增加,執行檔需要更大的記憶體空間才能執行。
2. 不同的執行檔無法共用相同的函式，必須在各自的程式碼中複製同樣的函式，形成浪費。

為了改進靜態連結所產生的缺點，在 UNIX 以及 Linux 作業系統上，就逐漸發展出分享式函式庫(`Shared Library`)，而 Win32 平台則發展出動態連結函式庫(`Dynamic Link Library`)。由於動態連結函式庫的函式的程式

碼是在應用程式載入或是執行階段才與應用程式相連結，也就是說，函式的程式碼與應用程式的程式碼在編譯時彼此是各自獨立成不同的檔案。與靜態連結相比，由於函式被分離成個別的動態連結檔，應用程式所佔的記憶體空間自然就縮減了，也因為獨立的動態連結檔，才可以讓不同的應用程式共用相同的函式，作業系統能更有彈性地調配資源。

3.4 OCX 說明

OCX 是一種 ActiveX 控制項，可由 VC、VB 或 Delphi 等程式語言來開發。ActiveX 控制項是一種可以被執行的檔案，像是 OCX、DLL 或是 EXE 檔案，除了 EXE 檔案外，其它如 OCX 或 DLL 必須要在其它的應用程式下才能被執行，這些應用程式如 VB 的表單、IE 網頁瀏覽器或是 Office 軟體，ActiveX 的概念就如 Netscape 的 PlugIn 的概念一樣，都是程式需要某一項額外的功能時，就把一個可執行的檔案掛入執行，這樣就能擴充程式本身做不到的功能。

ActiveX 控制項的特點是：一般軟體需要用戶端下載然後執行安裝，而 ActiveX 控制項是當用戶瀏覽到特定的網頁時，IE 瀏覽器即可自動下載並提示用戶端是否安裝。但是 ActiveX 控制項安裝前必須經過用戶端的同意及確認。

在本研究中就是將 OCX 嵌在 IE 中，ActiveX 控制項 OCX 需要登錄註冊後才可以使用，在 IFCEngine DLL&OCX 此應用程式介面裡，已經將註冊 OCX 的語法寫成一個批次檔(.bat)，我們只要執行該批次檔就可以將 OCX 檔註冊，其實註冊的語法很簡單，在命令提示位元切換到 OCX 所在資料夾，輸入 regsvr32 filename.ocx 即可。運作方式如圖 3-4-2，IFCEngine DLL&OCX 將 OCX 檔、DLL 檔與 INF 檔壓縮打包成 CAB 檔，

CAB 格式為微軟的封包檔，為一壓縮封包，可包含多個相關的檔案。封包檔格式為最佳的壓縮率。IFCEngine DLL&OCX 中也將打包 CAB 檔的指令寫成一個批次檔，我們只要執行該批次檔就可以將 OCX 檔、DLL 檔與 INF 檔壓縮打包成 CAB 檔(如圖 3-4-3)，十分方便，開啟網頁後，按下載入檔案的按鈕，網頁就會去呼叫 CAB 檔中的 OCX 檔，OCX 算是比較外層的程式，因為我們在原始碼(source code)裡所做的更改只會更改到 OCX 檔，而一些被 OCX 檔使用到的函數已經編譯成 DLL 檔，DLL 檔這部份的原始碼並沒有公開，所以我們無法看到當然也無法修改，我們只能就 IFCEngine DLL&OCX 所提供的功能加以應用，若真的要深入底層只能聯絡作者，付費就能得到原始碼。DLL 檔裡的內容大致是對 IFC 格式的解碼(parse)。IFC Engine DLL&OCX 運作原理如圖 3-4-1。

3.5 CGI 說明



CGI 全名 Common Gateway Interface 一般中譯為共用閘道介面，是介於客戶端(Client)和伺服器端(Server)的一種通訊介面，客戶端可透過此介面將資料傳到伺服器端，撰寫 CGI 的程式語言有很多，其中 C 與 Perl 程式語言較常用來撰寫，而 CGI 的運作原理如圖 3-5-1 所示，客戶端透過網頁中的表格(form)形式填入資料，再由 get 或 post 機制傳遞參數給在伺服器端的 CGI 程式，用 get 機制或用 post 機制並不是由客戶端的使用者決定的，而是由伺服器端的管理者寫在網頁中的，而伺服器端的 CGI 程式就是我們一般的 C 語言程式，只是我們不編譯(compile)成執行檔(.exe)而是編譯成 CGI 檔(.cgi)，這邊我們所使用來架設網頁伺服器的軟體是 Apache http server，這是一個免費且好用的軟體，所以使用率很高，參數傳遞機制我們用 post，原因是 get 傳遞機制會將變數名稱與輸入的值

連接到網址後面，如圖 3-5-1 所示，這不僅沒有隱私且這方法只能傳有限長度的資料，一旦資料量增多，可能會無法傳遞資料，而 post 傳遞機制中輸入的值會被當做是 Standard_input 也就是當成外在輸入，我們可以在程式中去抓取，不過不論是 get 機制或 post 機制，傳遞過來的資料都是以變數名稱 A=值_A &變數名稱 B=值_B &...，所我們必須一個個變數去作分析(parse)的動作，將所需的值濾出來作為程式的 input，而字串分析這麻煩的動作已經有很多熱心的人寫好程式給我們用了，所以我們不用額外費心自己去撰寫，這類程式網路上已經有很多可以下載，而我用的是網路上找到的 util.c 這個 C 程式來幫我做解析。



第四章 系統實作與開發

本章將說明實作系統的過程，先用 JSP 動態網頁技術實作檔案上傳的功能。再改寫 IFC Engine DLL&OCX 的原始碼，以新增一些功能，讓使用起來更便利。再以 IFC 檔案開啟程式 IFCQuickBrowser 來幫助分析 IFC 檔案格式，並依據 IFC 格式撰寫一個產生柱筋 IFC 圖檔的程式，並將該程式以 CGI 的技術應用於網頁。最後利用 IFC Engine DLL&OCX 內的 API 來擷取 IFC 檔案內物件的資料，擷取這些資料與規範比對進行審核。

4.1 檔案上傳系統

檔案上傳系統流程如圖 4-1-1 所示，於上傳頁面(index.htm)運用 javascript 語法來限制只能選擇副檔名為.ifc 的檔案上傳，並以訊息視窗來警告使用者(如圖 4-1-2)，以免系統讀入不符的檔案造成毀損或當機，再來架設資料庫系統，使用的資料庫軟體為 Microsoft Office ACCESS 2003，並以網頁伺服器架設軟體 Tomcat HTTP Server 來搭配動態網頁技術 JSP(Java Server Pages)，JSP 文件主要是以 Java 語言來開發，為提供網頁應用程式發展的語法，負責網頁與網頁間參數的傳遞，還有網頁與資料庫間的溝通，來達成對資料庫中資料的存取或變更等等，這邊使用的 Tomcat HTTP Server 為 Tomcat 4.1，Java SDK 的板本為 6.0，並使用 Advantys 公司所撰寫的檔案上傳的元件 jspSmartUpload，來幫助我們實作上傳的機制，此元件是免費的，使用時需要將它包含(include)進來，語法為<%@ page import="com.jspsmart.upload.*" %>，這樣一來我們就能建立 SmartUpload 類別的物件，以此物件來使用 SmartUpload 類別所提供的上傳相關函數。在此用到的函數有：

initialize(pageContext)	初始化上傳或下載的相關工作
upload();	依照指定檔案進行上傳的動作
save(path)	將上傳的檔案儲存到指定路徑，並回傳成功儲存的檔案數
getFiles()	取得上傳檔案的集合物件(Files)，以獲取更別檔案的資訊
getFileName()	取得上傳檔案的檔案名稱

檔案上傳後，將會存放到指定路徑下的資料夾，本程式的指定路徑為 Apache HTTP Server 中的資料夾，並且在資料庫 ACCESS 裡紀錄上傳檔案的檔名，若順利儲存完成，網頁則顯示上傳成功，並有功能項目的超連結引導使用者往下一個功能，而本研究中只針對了檢核功能做實作。詳細內部運作流程如圖 4-1-3 所示。

進入了檢核頁面，而這頁面就是原本 IFCEngine OCX&DLL 應用的頁面，我們運用 JSP 語法將讀入檔案的路徑(也就是變數“filename”的值)改成網頁伺服器 Apache HTTP Server 的路徑，並自動從資料庫裡擷取剛上傳檔案的檔名附於路徑後面，這樣一來就能將剛上傳的檔案讀入進行 3D 瀏覽與規範檢核了。



4.2 IFC 格式分析與資料擷取

IFC 的內容原始碼，我們用記事本或 WordPad 等文字編輯器開啟 IFC 檔時就會看到這樣一大串密密麻麻的原始碼，圖 4-2-1 所示，乍看之下很凌亂，但是這些碼是有規則有涵義的，每一行的表示方式都是“#Number = EntityName(Attribute1, Attribute2,...);”，當 Entity 要參考另一個 Entity 時就是靠著前面的#Number 來參照，但是這樣照順序排列很難看出 Entity 與 Entity 之間的相互關係，所以用軟體 IFCQuickBrowser 來輔助我們閱讀及解析 IFC 的原始碼，因為在 IFC 格式中，描述物件的方式是以樹狀結構的方式來儲存，IFCQuickBrowser 可以快速的找尋子節點(children

node)和父節點(parent node)，表 4-2-1 是一段 IFC 的原始碼，這段 IFC 原始碼是一段描述柱(column)的原始碼，以 IFCQuickBrowser 開啟 IFC 檔並找到 IFCCOLUMN 點選展開就會如圖 4-2-2 所示，可以看出是以 IFCCOLUMN 為樹根(root)向下擴展的一個樹狀結構(如圖 4-2-3)，這根柱的一些屬性就會記錄在這樹狀結構中。再看看表 4-2-2 對 IFCCOLUMN 的定義，對照表 4-2-1 的#83 內容，屬性 GlobalId 的內容就是字串 '2OCKxkKIP618ocbVxgfG\$\$'，屬性 OwnerHistory 的內容為 #13(IFCOWNERHISTORY)，屬性 Name 的內容為字串'CRE - 001'，屬性 Description 的內容為\$表示為空值(NULL)，屬性 ObjectType 的內容也為 \$ 表示空值，屬性 ObjectPlacement 的內容為 #132(IFCLOCALPLACEMENT)，屬性 Representation 的內容為 #124(IFCPRODUCTDEFINITIONSHAPE)，而屬性 Tag 的內容也為空值，當屬性的內容須參考下一個 Entity 時，我們必須一直參考下去，直到最後有實值的部份，這實值可能為字串(String)、實數(Real)、列舉(Enum) 等等，而 OPTIONAL 的意思則表示為此屬性內容可以不存在也就是空值。

在 IFCEngine 中會給每行的 Entity 一組 id，所以我們只要知道該行 Entity 的 id，就可用函數 sdaiGetStringAttrBN (id,attribution)就可擷取出該行 Entity 的任何屬性質。以此類推，依此方法我們就可以了解整個 IFC 檔的內容與屬性資料的擷取。

4.3 柱筋 IFC 檔產生器

這是一個用 C 語言所撰寫的程式，來產生有鋼筋實體的 IFC 檔，目前業界的鋼筋圖是用 AutoCAD 軟體所繪製，僅用線段及標注來表示鋼筋的資訊，鋼筋圖中僅呈現鋼筋整體之某個截面的情形(如圖 4-3-3)，用標注

標示這斷面有幾根幾號的鋼筋(如圖 4-3-4)，有些還得看兩三張鋼筋圖才能知道整個鋼筋分布的全貌，這種 2D 的鋼筋圖我們很難想像它的 3D 實際形狀是長的怎樣，既使匯出成 IFC 格式，由於圖中的鋼筋不是以 IFC 的語法來描述，我們在 IFC 3D viewer 中看到的也只會是一些線段甚至無法匯出成 IFC 格式，在 BIM 的應用下，是無法以標注或注釋的方式來表示物件的資訊，而是以屬性的方式儲存在物件本身中，目前的 CAD 建築繪圖軟體都沒有使用到 IFC 對鋼筋所定義的格式，有的甚至連畫鋼筋的元件都沒有，但是 IFC 的資訊架構中(如圖 4-3-11)，有預留與鋼筋連接的部份，所以本研究參考 IFC 的規格與 IFC 定義鋼筋的實體格式 IFCReinforcingBar，寫了一個簡易的程式，先拿現有的柱子 IFC 檔案，在該檔案中加入鋼筋的實體 IFCReinforcingBar，並填入鋼筋屬性，再用 IFCRELAGGREGATES 這個 Entity 來連接原來柱子(IFCCOLUMN)與剛設計的鋼筋(IFCREINFORCINGBAR)，使柱與鋼筋產生關係(如圖 4-3-12 所示)。



程式設計成可以讓使用者輸入柱子與柱筋的一些參數，會產生一個矩形柱內嵌柱筋且符合 IFC 規格的圖檔，產生出來的 IFC 檔如圖 4-3-13 所示，我們就可將這個矩形柱內嵌柱筋檔案給規範檢核程式讀入，並做一些柱筋的規範審核，看看有沒有不符合規範的地方。

IFCReinforcingBar 是一個描述鋼筋的 Entity，除了與梁、柱相同的基本屬性外，此 Entity 還有 SteelGrade、NominalDiameter、CrossSectionArea、BarLength、BarRole、BarSurface 這些鋼筋屬性，分別代表鋼筋等級、鋼筋斷面直徑、鋼筋斷面積、鋼筋長度、鋼筋種類、鋼筋表面種類，其中鋼筋種類(BarRole)與鋼筋表面種類(BarSurface)是列舉(Enumeration)型態(如圖 4-3-5 與圖 4-3-6)，我們設計的鋼筋種類只能是列舉型態中有列出的，例如主筋(MAIN)、箍筋(RING)等等，鋼筋表面

種類也是一樣。

鋼筋產生器之原始碼(rbar.c)編譯後以執行檔(rbar.exe)的形式執行時如圖 4-3-7 所示。輸入的矩型柱資料有柱斷面長度、寬度及柱高度，主筋資料有主筋數、主筋號數與主筋長度，繫筋資料有繫筋號數、縱向繫筋數及橫向繫筋數，其他輸入資料有橫向鋼筋間距、彎鉤延伸長度與保護層厚度。這邊我們限制主筋數目只能是偶數根，因為主筋數目若是奇數根就會有偏心的情況，為了避免偏心的產生所以簡化設計，對使用者設計主筋數目做出限制。另外程式只限定設計出矩形柱，且無法畫出柱筋搭接的情況，也是為了簡化設計，方便程式的撰寫，因為將來的建築 CAD 繪圖軟體會有符合 IFC 格式的鋼筋繪圖元件，並且會有更多的功能，所以之後的 IFC 應用開發者就可直接用 CAD 繪圖軟體來繪製鋼筋圖並匯出成 IFC 檔來做應用，目前已有某些 CAD 繪圖分析軟體已經可以設計鋼筋搭接的部份了。

而系統在架構中，為了與網路結合，方便在網頁中執行，使用了 CGI 技術，編譯過程如圖 4-3-1 所示，將鋼筋產生器原始碼(rbar.c)編譯成物件檔(rbar.obj)，另外用來幫助解析(parse)輸入資料的 util.c 也編譯成物件檔(util.obj)，再將這兩者連結(link)成一個 CGI 檔(rbar.cgi)，之後撰寫一個表格(form)網頁在 rbar.html，來接收使用者所設計的參數，將輸入的參數傳到 rbar.cgi 執行，產生所設計的 IFC 檔，參數傳遞過程如圖 4-3-2 所示。編譯環境是於 VC++ 命令列(command line)下編譯，編譯指令為"cl"(如圖 4-3-8、圖 4-3-9、圖 4-3-10 所示)。

4.4 IFC Engine DLL&OCX 新增功能

IFC Engine DLL&OCX 之原始程式提供了幾項基本的功能，讓我們可

以作為參考加以延伸的開發下去，新增或修改原有的功能，以符合本系統所需要的功能，而新增檢核的功能將留在 4.5 與 4.6 做說明。

新增功能方面有擷取新的屬性，梁柱方面我們針對矩形斷面的梁柱擷取了斷面長度與寬度、梁柱長度與兩端斷面的中心座標(如圖 4-4-1、圖 4-4-2)。鋼筋方面我們擷取了鋼筋等級、斷面直徑、斷面積、鋼筋長度、鋼筋種類(主筋、箍筋、繫筋等等)、鋼筋表面形狀(竹節等等)與兩端斷面的圓心座標(如圖 4-4-3)。樓板方面我們擷取樓板面積、樓板厚度及樓板四個角點的座標(如圖 4-4-4)。

此外在 3D 瀏覽的功能方面，我們新增了隱藏物件(hide)的功能(如圖 4-4-5)與顯示全部物件(display)的功能(如圖 4-4-6)，這方便我們將不想看到或有礙於觀察的物件隱藏起來，並且提供恢復的功能，增加操作的靈活性與便利性。

而在物件雙擊(double click)方面，為了使點選的物件方便觀察，我們將被點選的物件塗上亮黃色(如圖 4-4-7)作為記號，以凸顯此物件是正被點選的。

4.5 梁柱板構件檢核之功能

本節說明如何實作對梁柱板做空間分佈的檢核，所用的 3D 模型都是用 ArchiCAD v.11 繪圖軟體所建，並輸出成 IFC 檔案格式，檢核的項目包括：大梁跨小梁、梁身超過柱、兩根梁跨柱不在同一直線上與樓板開口不良造成剪力無法傳遞。檢核後若有錯誤，會將有問題的物件塗成紅色，並在右方顯示錯誤(Warning)的數目及對錯誤物件的描述(Description)。

◆ 大梁跨小梁

大梁跨小梁是一錯誤的情況，樓板承受載重，會將力傳遞給小梁，小梁再傳遞給大梁，大梁再傳給柱子，一旦發生大梁跨小梁的情形，小梁承受的力將無法傳到大梁只好承受下來，這時萬一受力超過負荷，小梁就會發生破壞，整個結構體也會遭受影響。大梁跨小梁的例子如圖 4-5-1。

檢核時，我們先擷取出梁的端點座標，判斷哪些端點是梁與梁的相接之處，在梁相接處的兩個端點座標，再比較兩者的斷面積，大者為大梁，小者為小梁，若小梁的垂直向座標(z 座標)比大梁的垂直向座標來的小，則判斷為設計錯誤，因為發生了大梁跨小梁的情況。

◆ 梁身超過柱

梁身超過柱也是一種不好的設計，如圖 4-5-2 所示，梁身超過柱的話梁會有一部分懸空，梁傳遞受力到柱就會有一點偏心，對梁或對柱都會造成剪力，有機會對梁或柱造成剪力破壞。

檢核時，我們先抽出梁的兩端點座標與柱與梁相接的端點座標，計算出梁兩端點連線的空間直線方程式，再計算柱端點座標到此直線的距離，此距離平方減去一半梁高的平方，將最後的值開根號，此值若大於柱寬的一半則判斷為梁身超過柱。

◆ 梁跨柱不在同一直線上

這是情形是兩支梁接在同一根柱上，但是這兩支梁不在同一直線上，如圖 4-5-3 所示，這是一個有待商榷的情況，所以我們要把這情況挑出來。

檢核時，先判斷任兩根梁是否有接在同一根柱上，抽出梁的兩端點座標後計算兩端點所形成的向量，若兩梁的向量夾角不是 0 度或 180 度則判斷為不在同一直線上。

◆ 樓板開口不良造成剪力無法傳遞

在樓板方面，有時可能會因為管線通道或電梯間等等因素要去掉幾塊樓板，如圖 4-5-4 所示，若有水平向的力可能會造成樓板較少的那一列對剪力的傳遞不良，造成破壞。這是作結構外審時可能會被檢驗的地方，若有這情形會被挑出來要求解釋說明或重新設計。

4.6 柱筋相關規範與檢核之功能

本節說明如何實作對柱筋作檢核，所依據的規範內容均是參考「[土木 401-96]混凝土工程設計規範與解說」一書。主要是針對「RC 柱構件」的規範。程式中也是以紅色作為警示，來表示有錯誤的物件，並在右方顯示警告錯誤的數目(Warning)及對錯誤物件的描述(Description)，若使用者對規範有疑慮，可點選檢核項目右邊的規範超連結，將會跳出新視窗來說明這條檢核所依據的規範內容。

4.6.1 檢核一：柱筋之鋼筋比

檢核時，以 `sdaiGetStringAttrBN (id,BarRole)=".MAIN."` 擷取出任意一主筋之 `id`，再以 `sdaiGetStringAttrBN (id,CrossSectionArea)` 擷取出該主筋的斷面積，再計算主筋的數目是以所有鋼筋 `id` 帶入 `sdaiGetStringAttrBN (id,BarRole)`，如果值是 `".MAIN."` 則判斷為主筋，之後再擷取出柱之斷面的長度與寬度，而鋼筋比就等於鋼筋斷面積乘鋼筋數目再除以柱之斷面積，看是否介於規範的 0.01 與 0.06 之間，這就是鋼筋比的審核。詳細規範內容如表 4-6-1。

4.6.2 檢核二：相鄰箍筋與繫筋之最大間距

檢核時，我們先將箍筋與繫筋分成平行柱寬與平行柱長兩部份，先做平行柱寬的部份，從所有的鋼筋中選出箍筋與繫筋且平行柱寬，並將它們存入一個陣列中，寫兩個迴圈計算出每一根鋼筋與其鄰近的鋼筋之距離，因為相鄰之鋼筋最多兩根最少一根，所以我們選兩者之大者為最大之間距，比較所有鋼筋與相鄰鋼筋之最大間距，傳回最大值，這就是平行柱寬的箍筋或繫筋與相鄰鋼筋之最大間距，同理我們可求出平行柱長的筋與或繫筋與相鄰鋼筋之最大間距，我們再取這兩者之大者作為整個斷面相鄰箍筋與繫筋之最大間距與規範規定的值進行比較。詳細規範內容如表 4-6-2。

4.6.3 檢核三：橫向鋼筋之間距

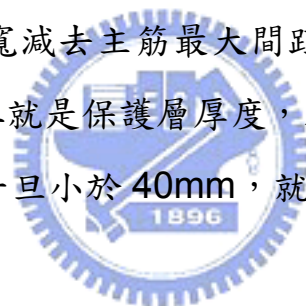
檢核時，我們先從所有鋼筋中擷取出箍筋的部份，這一步我們可用鋼筋屬性“BarRole”來辨認是不是箍筋，`sdaiGetStringAttrBN (id,BarRole)=".RING."`則判斷為箍筋，再計算出所有箍筋的間距，取最大的除以箍筋數減一，值就是相鄰箍筋的間距，也就是我們所要求的橫向鋼筋的間距。而限制條件中構材斷面可從柱的屬性來抽取，比較長度與寬度小者為構材斷面最小尺寸。主筋直徑的取得要先挑出主筋，也就是取鋼筋屬性“BarRole”為 MAIN 的鋼筋，`sdaiGetStringAttrBN (id,BarRole)=".MAIN."`，再抽出主筋直徑值，`sdaiGetStringAttrBN (id,NominalDiameter)`。而 h_x 可由檢核二中求得。比較橫向鋼筋間距與三個限制式，看是否都不超過，以檢核橫向鋼筋是否符合規範。詳細規範內容如表 4-6-3。

4.6.4 檢核四：彎鉤長度

檢核時，我們先從所有鋼筋中挑出繫筋與箍筋，`sdaiGetStringAttrBN (id,BarRole) ="LIGATURE" 或 ".RING."`，再擷取出該鋼筋直徑，`sdaiGetStringAttrBN (id,NominalDiameter)`，當然直徑大於 D25 是一定不符合規範的，之後再依彎鉤的走向向量(extruded vector)判斷出彎鉤的角度，並擷取出彎鉤的長度這就是我們要求的彎鉤延伸長度，再判斷該彎鉤所適用的規範來檢核。詳細規範內容如表 4-6-4。

4.6.5 檢核五：保護層厚度

檢核時，我們要算出保護層的厚度，我們擷取出柱的寬度，再計算出於柱寬方向相距最遠的主筋間距，並抽出主筋直徑與箍筋直徑，這時保護層厚度的計算就等於柱寬減去主筋最大間距再減去兩倍箍筋直徑與一倍主筋直徑，再將值除以二就是保護層厚度，為簡化檢核，我們設定保護層厚度需不小於 40mm，一旦小於 40mm，就視為不符規範。表 4-6-5。



4.6.6 檢核六：柱筋之搭接

鋼筋的搭接在柱筋產生器裡並沒有實作出來，而是以一根鋼筋到底的方式來繪製，由於時間的限制加上目前已經有建築 CAD 軟體可以畫出鋼筋搭接的情形，所以就不著重在鋼筋搭接的部份，只提供搭接部份的規範供參考，並不提供檢核。表 4-6-6。

第五章 系統展示

章節一開始將展示本研究所實作的系統，也就是網頁的架構。並以設計好的案例模型傳入系統程式作檢核，先以建築繪圖軟體 ArchiCAD 建立梁、柱、板之模型，傳入系統程式作檢核。再以開發的柱筋產生器產生矩形柱內嵌鋼筋的模型，傳入系統程式作檢核，並展示檢核的結果。

5.1 網頁架構展示

本系統分四個部份的網頁(如圖 5-1-1)，分別是首頁(index.html)，用來上傳檔案。柱筋設計的表格網頁(rbar.html)，用來填入設計參數。伺服端的 CGI 程式(rbar.cgi)回傳所產生 IFC 檔案。檢核頁面(check.jsp)，對柱筋與梁、柱、板做簡單的審核。

第一部分首頁(index.html)是一個檔案上傳的系統(如圖 5-1-2)，按瀏覽鈕可以選擇所要上傳的檔案(如圖 5-1-3)，由於是針對 IFC 檔案來實作，所以系統限制只能上傳副檔名為.ifc 的檔案，這是利用 JSP 撰寫的動態網頁，這部份實作了類似 e-submission 的上傳檔案功能。

第二部份是鋼筋設計的表格網頁(rbar.html 如圖 5-1-4)，這頁是由首頁超連結(hyperlink)過來的，輸入的柱資料有柱斷面長度、寬度及柱高度，主筋資料有主筋數、主筋號數與主筋長度，繫筋資料有繫筋號數、縱向繫筋數及橫向繫筋數，其他輸入資料有橫向鋼筋間距、彎鉤延伸長度與保護層厚度。並有鋼筋橫斷面圖片及鋼筋標準尺寸表可引導使用者設計出想要的柱筋設計圖。

第三部份是柱筋設計好按送出(submit)，設計資料送至伺服端後，執

行 CGI 程式(rbar.cgi)，回傳所產生 IFC 檔案(rbar.ifc)的超連結於頁面，供使用者下載剛剛設計的柱筋 IFC 檔(如圖 5-1-5)。

第四部份是上傳檔案後所到的檢核頁面(check.jsp)，這頁是本研究的核心所在，使用者可將設計好的柱筋 IFC 檔案上傳至檢核頁面，進行 3D 模型瀏覽展示(如圖 5-1-6)，並透過柱半透明化來觀察柱內部的鋼筋分布(如圖 5-1-7)，雙擊(double click)柱或鋼筋物件來檢視其基本屬性，看看與理想的設計是否一致，並可對設計的柱筋進行規範的檢核。如果我們上傳了有梁、柱或板的圖檔，則可以檢核梁、柱與板之空間分布不合理的情況，這部份實作了部份的類似 CORENET e-plan check 的自動規範審核系統。

5.2 系統環境



本小節將建議使用者最佳的使用環境:

- ◆ 作業系統:WindowsXP
- ◆ 網頁瀏覽器:IE browser
- ◆ 顯示晶片:nvidia
- ◆ 3D 加速軟體:DirectX 8.0 或以上

另外要自訂 IE 網際網路的安全性，從 IE 的工具→網際網路選項→自定等級，將“ActiveX 控制項與外掛程式”全部啟用，否則系統所用的程式控制項會被預設的安全性設定擋住，導致無法執行。

5.3 梁柱板構件檢核功能展示

本節展示以 ArchiCAD v.11 繪圖軟體建置所需要的模型，有大梁跨小梁模型、梁身超過柱模型、兩根梁跨柱不在同一直線上之模型與樓層樓板開口不良造成剪力無法傳遞的模型。再分別上傳至系統，讓系統作檢核，並展示檢核結果。

◆ 大梁跨小梁實例

大梁跨小梁模型如圖 5-3-1 所示，有兩組梁，右邊的梁明顯斷面積比較大，視為大梁，左方的為小梁，但是上方這組接合處是大梁搭在小梁上，而下方這組作為對照，是小梁搭在大梁上。將檔案以 IFC 格式匯出，上傳至檢核系統，選擇檢核梁柱不合理的選項，結果發現一個錯誤且上方那組梁被標示為紅色(如圖 5-3-2 所示)，點選紅色那組梁，錯誤物件描述(Description)的欄位顯示了發生了大梁跨小梁的情況(如圖 5-3-3 所示)。

◆ 梁身超過柱實例

梁身超過柱模型如圖 5-3-4 所示，最左邊與最右邊的梁都有一部份沒有跨在柱上，有點懸空的情況，將檔案以 IFC 格式匯出，上傳至檢核系統，選擇檢核梁柱不合理的選項，結果發現有七個錯誤，左方與右方的梁柱組合都被標示紅色(如圖 5-3-5)，點選任意的紅色物件，錯誤物件描述(Description)的欄位顯示梁身超過柱的情形發生了(如圖 5-3-6)。

◆ 兩根梁跨柱不在同一直線實例

兩根梁跨柱不在同一直線上的模型如圖 5-3-7 所示，上方的梁與下方的梁連接在同一根柱上，但是這兩根梁並不能連成一條直線也就是說不在同一直線上，將檔案以 IFC 格式匯出，上傳至檢核系統如圖所示，選擇檢

核梁柱不合理的選項，發現有兩個錯誤，就是上方的梁與下方的梁，並將這兩根梁以紅色表示有問題(如圖 5-3-8)，點選任意一根，錯誤物件描述(Description)的欄位顯示梁跨柱不在同一直線上的情形發生了(圖 5-3-9)。

◆ 樓板開口不良造成剪力無法傳遞

樓層樓板中空的模型如圖 5-3-10 所示，工程可能基於某些原因會中空樓板，我們故意將設計模型某一行的板挖空，以凸顯出整個樓板若受剪力時，較少樓板的那一行會無法傳遞剪力，產生應力集中的現象，將本檔案以 IFC 格式匯出，上傳至檢核系統，選擇檢核樓板不合理的選項，發現有一個錯誤，該錯誤以紅色表示(如圖 5-3-11 所示)，就是較少樓板的那一行剩下的樓板，也是可能會發生破壞的地方，錯誤物件描述(Description)的欄位也告訴我們也許會發生剪力無法傳遞與破壞的情況(如圖 5-3-12 所示)。



5.4 柱筋檢核功能展示

本節展示以自行開發的柱筋內嵌於矩形柱之 IFC 檔產生程式來設計，依系統檢核項目設計特殊的案例模型，傳入系統程式作檢核，並顯示檢核的結果。

開啟設計頁面(rbar.html)，會看到表格形式的表單，輸入想要設計的柱資料、主筋資料、繫筋資料與其他的資料參數，執行後，就可以將設計的檔案下載下來，上傳至檢核系統，開始作柱筋的規範檢核。

在此設計兩個矩形柱內嵌柱筋的模型，設計模型一是任意設計的錯誤模型，設計模型二是依照設計圖的柱配筋圖來設計的模型，分別上傳系統所做的檢核。

5.4.1 設計模型一

模型設計參數如圖 5-4-1 所示，隨意輸入矩形柱斷面為 1.3x1.3(m)，柱長為 3.14(m)，16 根 11 號的主筋，長度為 6.28(m)，設計主筋長度大於柱長度以方便觀察鋼筋的排列，縱向繫筋 2 根，橫向繫筋 2 根，繫筋使用 7 號鋼筋，橫向鋼筋間距為 1(m)，彎鈎延伸長度為 0.1(m)，保護層厚度為 0.2(m)。

◆ 檢核柱筋鋼筋比

上傳所設計的模式檔案後，進入檢核頁面，點選檢核柱筋鋼筋比的選項，檢核結果顯示一個錯誤，錯誤描述欄位會顯示鋼筋比不符，參照鋼筋比規範如表，鋼筋比的值必須介於 0.01 與 0.06 之間，由設計表單填入的參數實際計算，柱的斷面長度與寬度皆為 1.3(m)，所以斷面積為 1.69(m)，設計的主筋為#11，查表發現#11 鋼筋直徑為 35.8(mm)，柱筋數為 16，經過計算，鋼筋比為 0.000952，並不在規範所定的 0.01 與 0.06 之間，所以這個設計是有問題的。如圖 5-4-4、圖 5-4-5、圖 5-4-6 所示，程式將錯誤的物件塗成紅色做出警告。若使用者對規範有疑慮，可點選檢核項目右邊的規範超連結，將會跳出新視窗來說明這條規範的內容，如圖 5-4-7 所示。

◆ 檢核相鄰箍繫筋最大間距

上傳所設計的模式檔案後，進入檢核頁面，點選檢核相鄰箍繫筋最大間距的選項，檢核結果顯示一個錯誤，錯誤描述欄位會顯示相鄰箍筋與繫筋之最大間距不符，由設計表單填入的參數實際計算也很難計算出相鄰箍筋與繫筋之最大間距是多少，必須透過模型的座標，來計算空間中節點的距離，而目前只知道這距離大於規範所規定的 35 公分，設計者也許要多

加橫向的繫筋數，才能縮短相鄰箍繫筋最大間距，若有疑問可點選檢核項目右邊的規範超連結，將會跳出新視窗來說明這條規範的內容，如圖 5-4-8 所示。

◆ 檢核橫向鋼筋間距

上傳所設計的模式檔案後，進入檢核頁面，點選檢核橫向鋼筋間距的選項，檢核結果顯示一個錯誤，錯誤描述欄位會顯示橫向鋼筋間距不符，由設計表單填入的參數來看，設計的橫向鋼筋間距為 1(m)，規範對橫向鋼筋間距的規定如表 4-6-3 所示，由表中的三項條件選擇最大者作為比較的標準，經過計算是構材斷面最小尺寸的四分之一，而設計的構材斷面最小尺寸為 1.3(m)，所以構材斷面最小尺寸的四分之一為 0.325(m)，設計值大於規範規定，設計不符合規定，需重新設計。如圖 5-4-9、圖 5-4-10 所示，程式將會做出警告。若使用者對規範有疑慮，可點選檢核項目右邊的規範超連結，將會跳出新視窗來說明這條規範的內容，如圖 5-4-11 所示。

◆ 檢核彎鉤長度


上傳所設計的模式檔案後，進入檢核頁面，點選檢核彎鉤長度的選項，檢核結果顯示有七十個錯誤，錯誤描述欄位顯示彎鉤延伸長度不符，根據設計表單填入的參數，設計的彎鉤長度為 0.1m，用的繫筋是#7(D22)，而主筋為#11(D36)，主筋直徑為 35.8mm，依據規範的內容如表所示，D22 的 90°彎轉，其自由端應作至少 12 倍主筋直徑之直線延伸，而 D22 的 135°彎轉，其自由端應作至少 6 倍主筋直徑之直線延伸。經過計算 6 倍的主筋直徑為 0.02148m，大於設計的彎鉤延伸長度 0.1m，所以不論是 90°或是 135°的彎鉤的設計錯了，所以總共有 70 個錯誤。如圖 5-4-12 所示，程式將會做出警告。若使用者對規範有疑慮，可點選檢核項目右邊的規範超

連結，將會跳出新視窗來說明這條規範的內容，如圖 5-4-13 所示。

◆ 檢核保護層厚度

上傳所設計的模型檔案後，進入檢核頁面，點選檢核保護層厚度的選項，檢核結果顯示一個錯誤，錯誤描述欄位會顯示保護層厚度不符，從設計表單的內容可看出設計的保護層厚度是 0.02m，而保護層的規範內容如表所示，我們採用 40mm 為最小保護層厚度，很明顯的設計值小於規範規定，所以是錯誤的。如圖 5-4-14、圖 5-4-15 所示。若使用者對規範有疑慮，可點選檢核項目右邊的規範超連結，將會跳出新視窗來說明這條規範的內容，如圖 5-4-16 所示。

5.4.2 設計模型二



模型設計參數如圖 5-4-3 所示，是依據某一實際案例的柱配筋圖所設計的(如圖 5-4-2)。根據柱配筋圖矩形柱斷面為 130x130(cm)，所以輸入的矩形柱斷面為 1.3x1.3(m)，柱長假設為 0.5(m)，設計的主筋為 36 根 8 號鋼筋，長度假設為 1(m)，縱向繫筋為 4 根，橫向繫筋也為 4 根，使用的是 5 號鋼筋，橫向鋼筋間距在圍束區與非圍束區為皆為 10(cm)也就是 0.1(m)，彎鉤延伸長度自行假設為 0.2(m)，保護層厚度為設計圖規定的 4(cm)，所以輸入 0.04(m)。

柱筋鋼筋比審核結果如圖 5-4-17，相鄰箍繫筋最大間距審核結果如圖 5-4-18，橫向鋼筋間距審核結果如圖 5-4-19，彎鉤長度審核結果如圖 5-4-20，保護層厚度審核結果如圖 5-4-21。由於是根據柱配筋圖的設計值來設計，所以一如預期的沒有任何與規範不符的地方。

小結

IFC 圖檔上傳至審核系統可以選擇要審核的項目，在本系統中，按下所要審核的項目後，錯誤警告欄位會先顯示錯誤物件的數目，並將錯誤的物件塗成紅色，點選紅色物件在錯誤描述欄位會顯示錯誤的原因，若對該項審核有疑慮，可點選該條規範之後所附的規範詳細說明，以方便設計者釐清錯誤並做修改。而檢核柱筋搭接的功能並沒有實作例子，所以只有提供柱筋搭接之規範如圖 5-4-22。



第六章 結論建議與未來展望

本研究為發展一個梁柱板空間分布不合理與柱筋規範之自動審查系統，使用免費的 IFC 資料存取的應用程式介面—IFCEngine OCX&DLL 來幫助開發，並用 C 程式語言撰寫一個產生柱筋 IFC 檔的程式，來補足目前 CAD 繪圖軟體無法繪製鋼筋之 IFC 檔的缺憾。以下將是本研究所歸納出來的結論與建議，並對未來可能的發展加以期望。

6.1 結論

1. 本研究由於時間的限制只實作了整個自動審查系統架構的一小部份，就是檔案的上傳系統與對柱筋規範的自動審核，但是對於其他構件的審查可以依此方法深入的發展下去，只是需要更多的時間以及資源的投入，相信不久的將來，台灣也會有一套對於建築規範的自動審查系統。
2. 目前已經有許多 CAD 繪圖軟體有 3D 展示的功能，可以由立體的角度來觀察物件，不需再由 2D 平面圖來推測，但是建築圖或設計圖一旦構件增多，3D 圖形看起來就會很複雜，想要用肉眼來偵錯就變的十分困難，所以利用 BIM 的特性，擷取出構件的參數資料，透過一些計算與比較，讓電腦程式來自動判斷那些地方是不合理的，這也是本系統的功能之一。
3. 本研究中對於梁、柱、板之空間分布不合理的審核，雖然沒有法規的依據指出這些情況是錯誤的，甚至現存的房屋都可能有這些不合理的情況，但是這些情況都是可以再多加考慮的，是否真的要做這樣的一個設計，本研究對梁、柱、板的審核是提供設計者在設計時

的一個參考。

6.2 建議

1. 對於程式有待改進的建議是，規範的部份可以建成一個資料庫，要審核時再由資料庫裡讀入規範的數據，這樣一來，一旦規範有更新或有新增的規範，只要改變資料庫的內容即可，無須再改變程式裡函數的內容，增加維護與管理的便利性。對於審核的結果，應該要整理成一份審核報告，讓設計者清楚的了解該設計檔的錯誤之處，並註明錯誤的原因，以方便設計者作修改。
2. CAD 繪圖軟體商對於 IFC 資料格式的支援度應再提高，因為資訊的標準化是必然的趨勢，而 IFC 又是越來越多人使用資料標準格式，所以各家軟體商應再加強與 IFC 資料格式的結合，讓未來想開發 IFC 的開發者可以更便利、更快速。
3. IFC Engine OCX&DLL 此 IFC 應用程式介面中的 3D viewer 是由 ActiveX 控制項所提供之功能，作者在開發程式時，就是預定在 IE 網頁瀏覽器上使用，但是必須調整網頁安全性的設定，而另一個網頁瀏覽器 Netscape 安裝一些套件也可以支援 ActiveX 控制項，其他網頁瀏覽器(如 Firefox 等等)目前則無法支援且尚無解決辦法，這是比較可惜的地方，只能希望開發商加快更新該網頁瀏覽器的板本了。
4. IFC 採用了 EXPRESS 語言來描述與定義，學習 EXPRESS 語言可幫助我們更容易了解 IFC。所以建議要作 IFC 相關軟體的開發人員要先熟悉 EXPRESS，才能更靈活且完整的使用 IFC 標準。

6.3 未來展望

1. 相信不久的將來，我們台灣也會有一套適合於台灣建築規範的自動審查系統，那將會是台灣土木建築業的一大進步。
2. 之後的系統可加入人工智慧(AI, Artificial Intelligence)來幫助我們做複雜的檢核，因為對於經驗的法則或不成文的規範，可能需要經過人為的判定，這時我們將匯集眾多領域內的專家，依其專業知識與實際經驗，幫助我們打造一個專家系統(Expert System)，將可使建築規範自動檢核更加完善。
3. IFC 還有很廣泛的應用，例如結合 GIS 資訊做都市房地產的開發、室內溫度的調節，也就是暖氣與空調的配置(HVAC)、房屋管線的配置、業主預算的評估與標準的比較與計算、火災模擬與防火安全等等，這些應用都已經在開發階段，甚至有的已經在日常生活中被應用了，將來必定有更多與 IFC 相關的應用，讓我們拭目以待。



參考文獻

- [1] 李萬利,馬俊強,許英珽,“空間設計資訊整合模型之建置及應用探討”,
中華技術 special report 專題報導 2, 台灣世曦工程顧問股份有限公司/
資訊系統部, 2007.
- [2] 施恭,“基於 IFC 標準的建築工程 4D 施工管理系統”,中國建設信息
p.57–60, 2007/01.
- [3] 曾仁杰,“「IAI 國際標準組織年會」會議之出國報告”,2005.
- [4] IAI, “About IAI”, <http://www.iai-international.org/>, 2008.
- [5] IAI, “IFC2x Technical Guide”,
http://www.iai-international.org/Model/documentation/IFC_2x_Technical_Guide.pdf, 2000.
- [6] 龍庭資訊有限公司,“BIM 觀念與 IFC 支援”,
<http://www.academicd.com/index2.htm/>, 2008.
- [7] Singapore Government Building & Construction Authority,
“CORENET”, <http://www.corenet.gov.sg/>, 2008.
- [8] AECbytes, “CORENET e-PlanCheck: Singapore's Automated Code Checking System”, AECbytes "Building the Future" Article, 2005.
- [9] Keunhyoung Lee, Sangyoon Chin and Jaejun Kim, “A Core System for Design Information Management Using Industry Foundation Classes”, Computer-Aided Civil and Infrastructure Engineering, 2003.
- [10] Mahmoud Halfawy, Thomas Froese, “INTEGRATION OF DATA

AND PROCESSES OF AEC PROJECTS USING THE INDUSTRY FOUNDATION CLASSES”, 6th Construction Specialty Conference, 2005.

[11] 曾旭東,譚潔,“基於參數化智能技術的建築信息模型”,重慶大學大學報(自然科學板),Vol.29 No.6 P.107-110, 2006.

[12] 沈秉廷,“運用物件導向技術於 IFC 建築資訊”,碩士論文,國立交通大學土木工程研究所, 2008.

[13] I. Faraj, M. Alshawi ,et al, “An industry foundation classes Web-based collaborative construction computer environment: WISPER”, Automation in Construction 10 79–99,2000.

[14] 邱奎寧,王磊,“IFC 標準的實現方法”, BUILDING SCIENCE Vol.20 No3, 2004.

[15] TNO Building and Construction, “IFC Engine DLL”,
<http://www.ifcbrowser.com/ifcengineDll.html>, 2008.

[16] TNO Building and Construction, “IFC Engine OCX”,
<http://www.ifcbrowser.com/ifcengineocx.html>, 2008.

[17] Cheng Tai Fatt, “An IT Roadmap for Singapore’s Construction Industry”,http://www.iai.no/2005_buildingSMART_oslo/June%201st/03%20cheng%20tai%20fatt%20Norway%20-%20IAI.pdf,2005.

[18] 中國土木水利工程學會,“混凝土工程設計規範與解說(土木401-96)”,科技圖書, 2007.

[19] 黃軍棠,“結構實驗 3D 資訊處理系統之研究”,碩士論文,國立台灣大學土木工程研究所, 2006.

附表

表 4-2-1 一段 IFC 原始碼

```

#13= IFCOWNERHISTORY(#12,#5,$,.NOCHANGE.,,$,$,1208002255);
#83= IFCCOLUMN('2OCKxkKIP618ocbVxgfG$$',#13,'CRE - 001',
              $,$,#132,#124,$,$,$,$,MAIN.,$);
#100= IFCDIRECTION((1.,0.));
#102= IFCCARTESIANPOINT((-100.,0.));
#104= IFCAXIS2PLACEMENT2D(#102,#100);
#105= IFCRECTANGLEPROFILEDEF(.AREA.,$,#104,200.,200.);
#106= IFCAXIS2PLACEMENT3D(#35,#30,#32);
#107= IFCEXTRUDEDAREASOLID(#105,#106,#32,9926.0076);
#108= IFCSHAPEREPRESENTATION(#40,'Body','SweptSolid',(#107));
#112= IFCCARTESIANPOINT((-100.,0.,-200.));
#114= IFCBOUNDINGBOX(#112,200.,9926.0076,200.);
#115= IFCDIRECTION((6.1230318E-17,1.));
#117= IFCGEOMETRICREPRESENTATIONCONTEXT
      ('Plan','Sketch',3,1.0000000E-5,#37,#115);
#120= IFCSHAPEREPRESENTATION(#117,'IAI','BoundingBox',(#114));
#124= IFCPRODUCTDEFINITIONSHAPE($,$,(#108,#120));
#127= IFCDIRECTION((0.,-1.,0.));
#129= IFCCARTESIANPOINT((2029.1953,14012.94,2800.));
#131= IFCAXIS2PLACEMENT3D(#129,#32,#127);
#132= IFCLOCALPLACEMENT(#72,#131);
    
```

表 4-2-2 IFCCOLUMN 的定義

```

ENTITY IfcColumn;
GlobalId          :IfcGloballyUniqueId;
OwnerHistory      :IfcOwnerHistory;
Name              :OPTIONAL IfcLabel;
Description       :OPTIONAL IfcText;
ObjectType        :OPTIONAL IfcLabel;
ObjectPlacement  :OPTIONAL IfcObjectPlacement;
Representation    :OPTIONAL IfcProductRepresentation;
Tag              :OPTIONAL IfcIdentifier;
    
```

表 4-6-1 縱向鋼筋之鋼筋比規定[18]

規範之 15.5.3.1，縱向鋼筋面積 A_{st} 不得低於 $0.01A_g$ ，亦不得大於 $0.06A_g$ 。

(A_{st} :縱向鋼筋面積， A_g :鋼筋混凝土總斷面積)

表 4-6-2 橫向鋼筋之相鄰箍筋與繫筋之最大間距規定[18]

規範之 15.5.4.3，在構材橫斷面上，繫筋或閉合箍筋相鄰各肢之中心距 h_x 不得超過 35cm。

表 4-6-3 橫向鋼筋之間距規定[18]

規範之 15.5.4.2，橫向鋼筋之間距不得超過

(1)構材斷面最小尺寸的 1/4

(2)6 倍主筋直徑

(3) $S_0 = 10 + \left(\frac{35 - h_x}{3}\right)$ ， s_0 之值不得超過 15cm，亦不必小於 10cm。

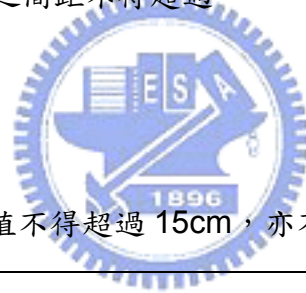


表 4-6-4 標準彎鉤之規定[18]

規範之 13.3.3 標準彎鉤:

90°之彎轉，其自由端應作至少 12 倍主筋直徑之直線延伸。

規範之 13.3.4 標準彎鉤，肋筋或箍筋之標準彎鉤為:

(1)D16 及較小之鋼筋—90°彎轉，其自由端應作至少 6 倍主筋直徑之直線延伸。

(2)D19、D22 及 D25 鋼筋—90°彎轉，其自由端應作至少 12 倍主筋直徑之直線延伸。

(3)D25 及較小之鋼筋—135°彎轉，其自由端應作至少 6 倍主筋直徑之直線延伸。

表 4-6-5 鋼筋保護層之規定[18]

規範之 13.6.1 鋼筋之保護層(單位:mm, d_b 為鋼筋或鋼線之標稱直徑)

狀 況	板、牆、閣 柵及牆板	梁、柱 及基腳	薄殼及 摺板
不受風雨侵襲且不與土壤接觸者：			
鋼線或 $d_b \leq 16\text{mm}$ 鋼筋	20	40	15
$16\text{mm} < d_b \leq 36\text{mm}$ 鋼筋	20	40	20
$D_b > 36\text{mm}$ 鋼筋	40	40	20
受風雨侵襲或與土壤接觸者：			
鋼線或 $d_b \leq 16\text{mm}$ 鋼筋	40	40	40
$D_b < 16\text{mm}$ 鋼筋	50	50	50
澆置於土壤或岩石上或經常與水及 土壤接觸者：	75	75	
與海水或腐蝕性環境接觸者：	100	100	

表 4-6-6 柱筋搭接之規定[18]

規範之 5.18.2 柱筋之搭接:

規範之 5.18.2.1:

在設計載重作用下柱筋只承受壓力時，則其搭接須符合第 5.17.1 及 5.17.2 節之規定，且按其適用狀況須符合下列要求：

(1)橫箍柱在壓力作用下，若其柱筋在搭接長度內被有效面積不小於 $0.0015hs$ 之橫箍筋所圍封，則其受壓搭接長度可乘 0.83 予以折減，惟不小於 30 cm。其中有效面積是指垂直於斷面尺寸 h 方向之箍筋面積和。

(2)螺箍柱在壓力作用下，若其柱筋在搭接長度內被螺箍筋所圍封，則其受壓搭接長度可乘 0.75 予以折減，惟不小於 30 cm。

(h :構材總厚或全深， s :縱向鋼筋、橫向鋼筋、預力鋼線、錨栓之中心距)

規範之 5.18.2.2:

在設計載重作用下柱筋承受拉力時，符合下列三條件者其搭接長度可用表 5.16.2 中之甲級，否則須為乙級：

(1)柱筋拉應力不超過 $0.5f_y$

(2)柱任一斷面之搭接鋼筋面積百分比不大於 50%

(3)柱筋搭接位置至少錯開 l_d (受拉鋼筋之伸展長度)

(f_y :鋼筋之降伏強度， l_d :受拉鋼筋之伸展長度)

規範之 5.18.2.3:

在設計載重下柱筋承受之拉應力超過 $0.5f_y$ 時，柱筋搭接長度應採用

乙及搭接。



附圖

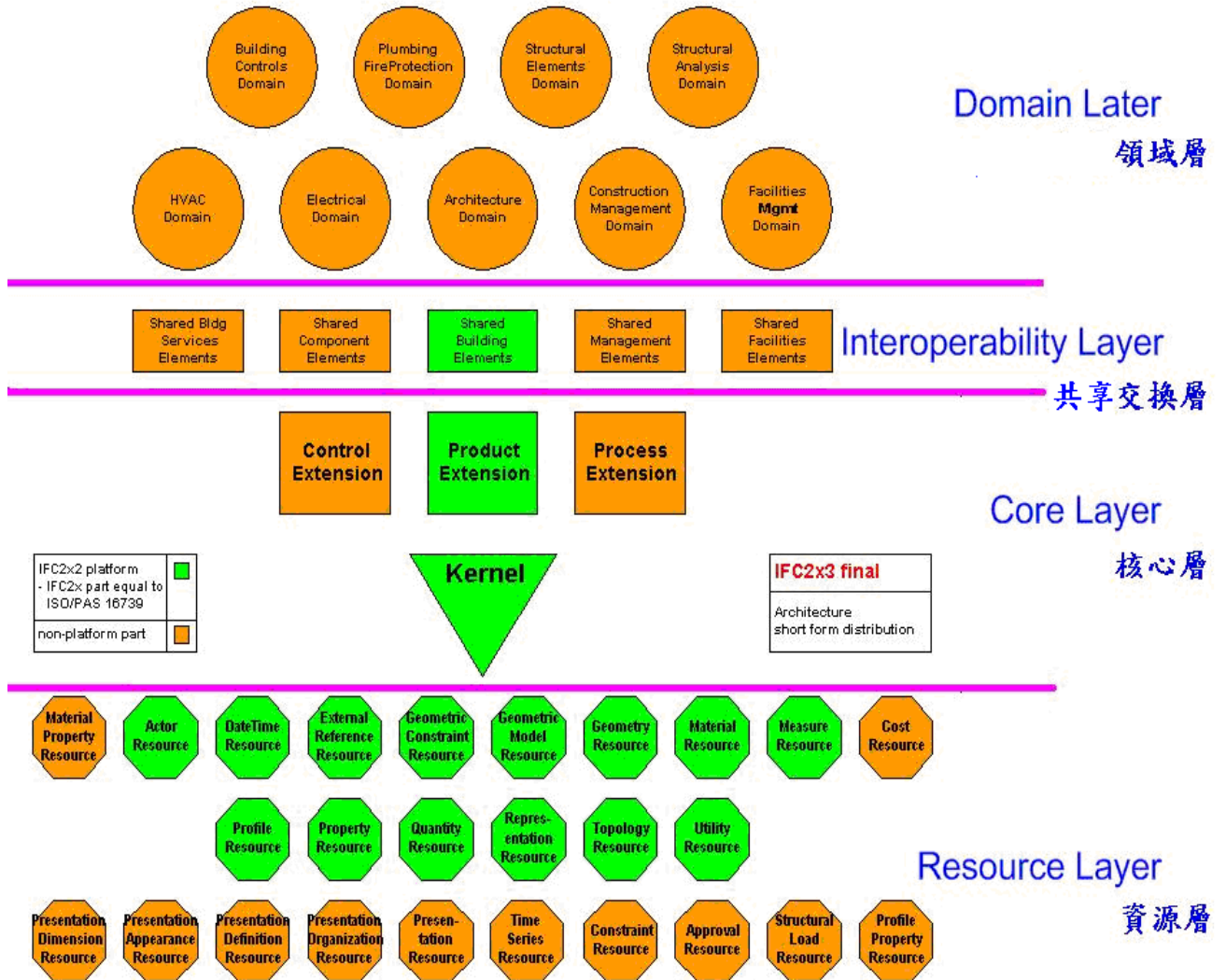


圖 2-1-1 IFC 架構圖[5]



圖 2-3-1 CORENET 的目的:發展 IT 系統來整合建築生命週期中的四個基礎過程-DESIGN, PROCURE, BUILD, MAINTAIN[8]

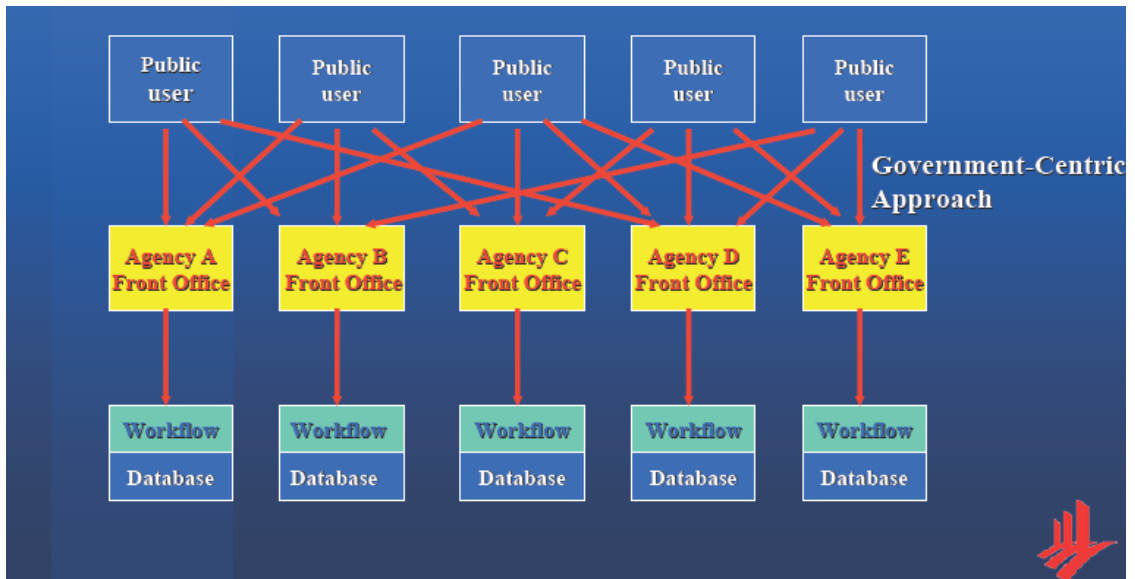


圖 2-3-2 傳統規範審核流程示意圖[17]

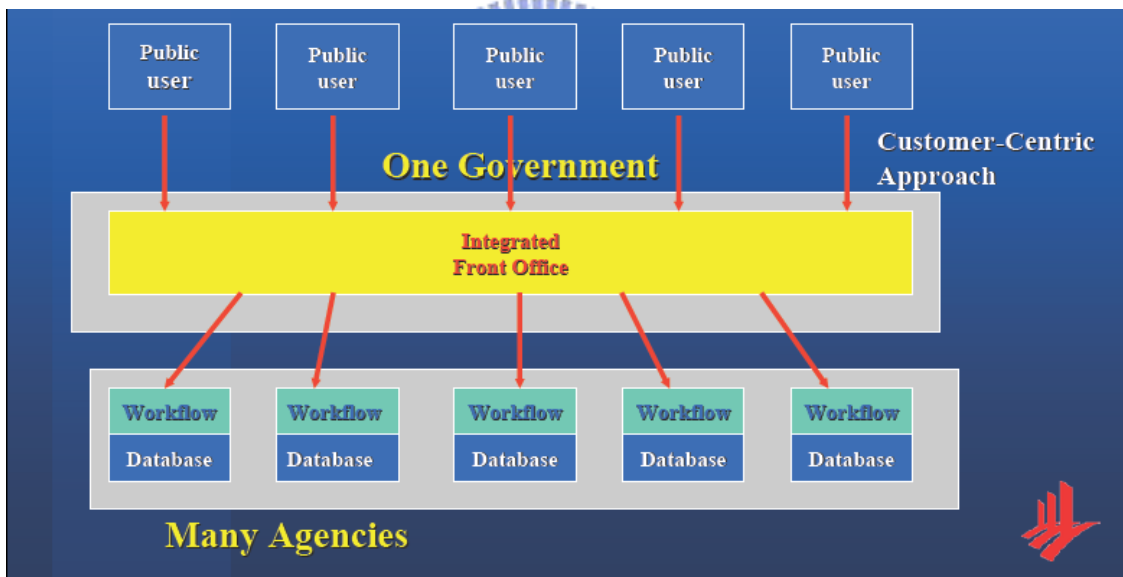


圖 2-3-3 自動規範審核流程示意圖[17]

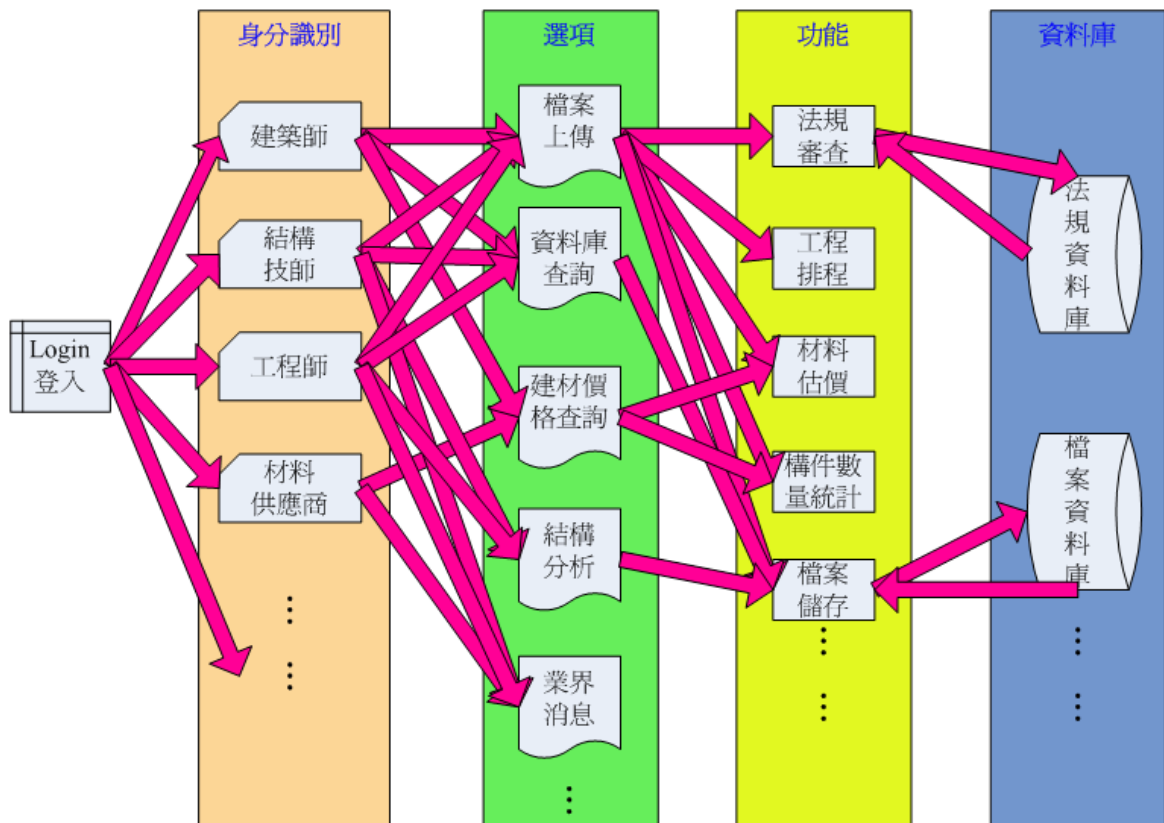


圖 3-1-1 整體系統架構 overview



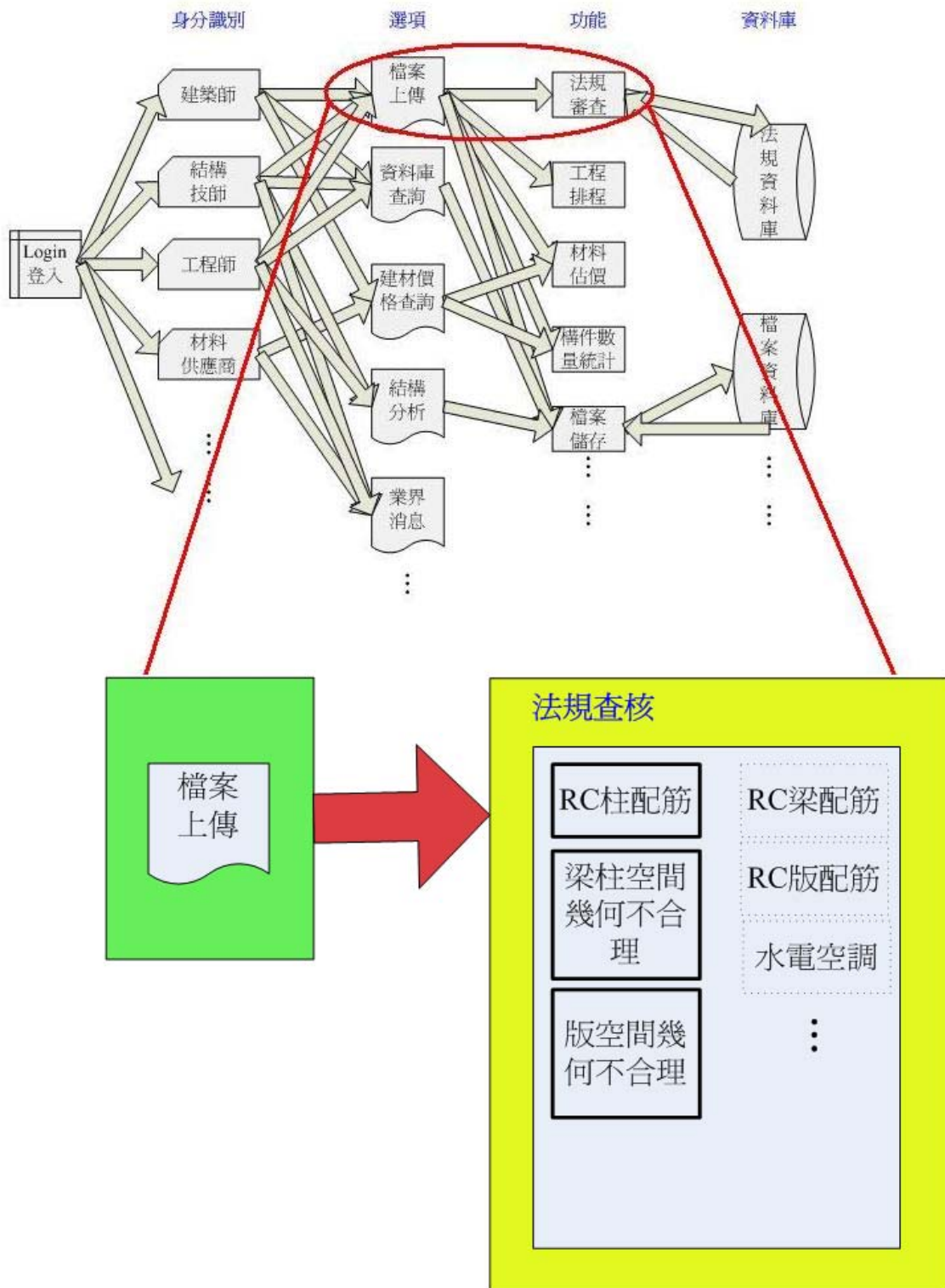


圖 3-1-2 整體系統架構與實作部份比較



圖 3-2-1 左:物件拉遠功能(zoom out) 右:物件拉近功能(zoom in)

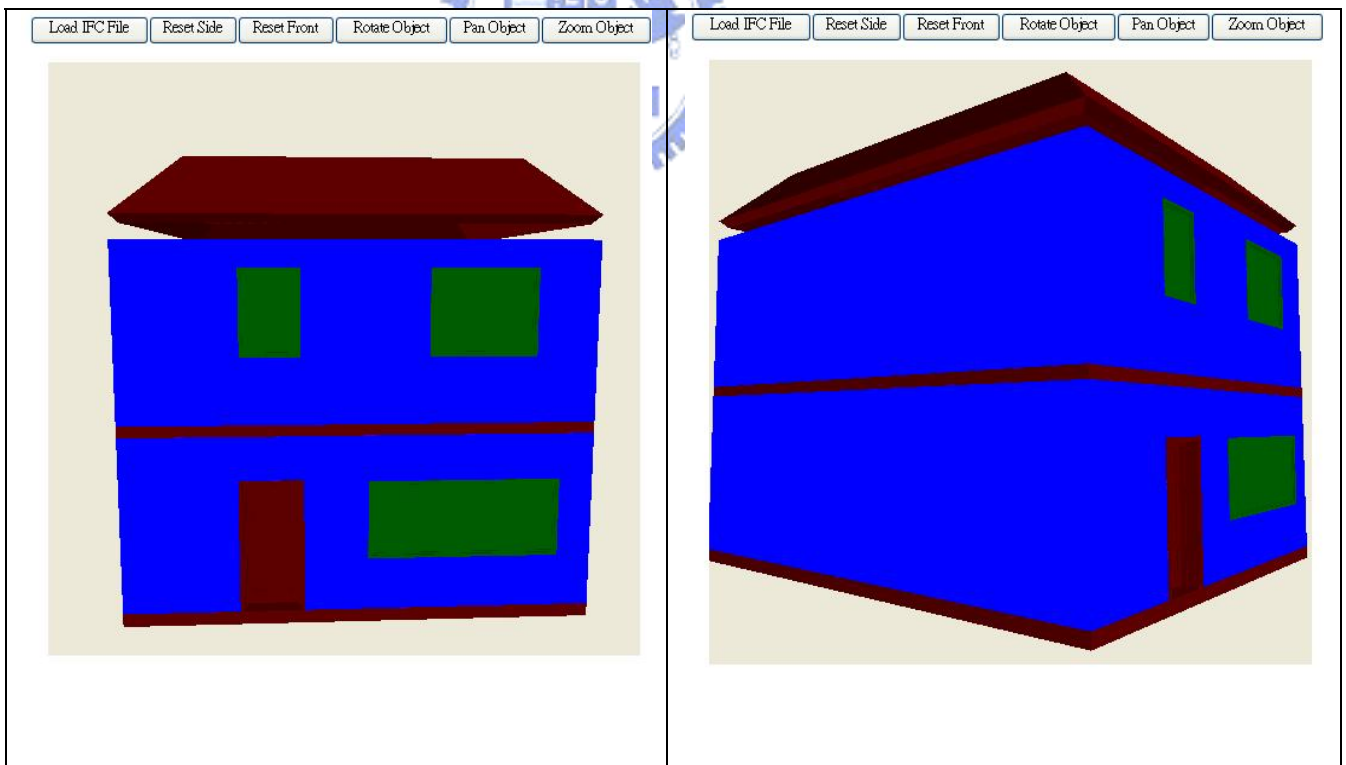


圖 3-2-2 物件旋轉功能(rotate)

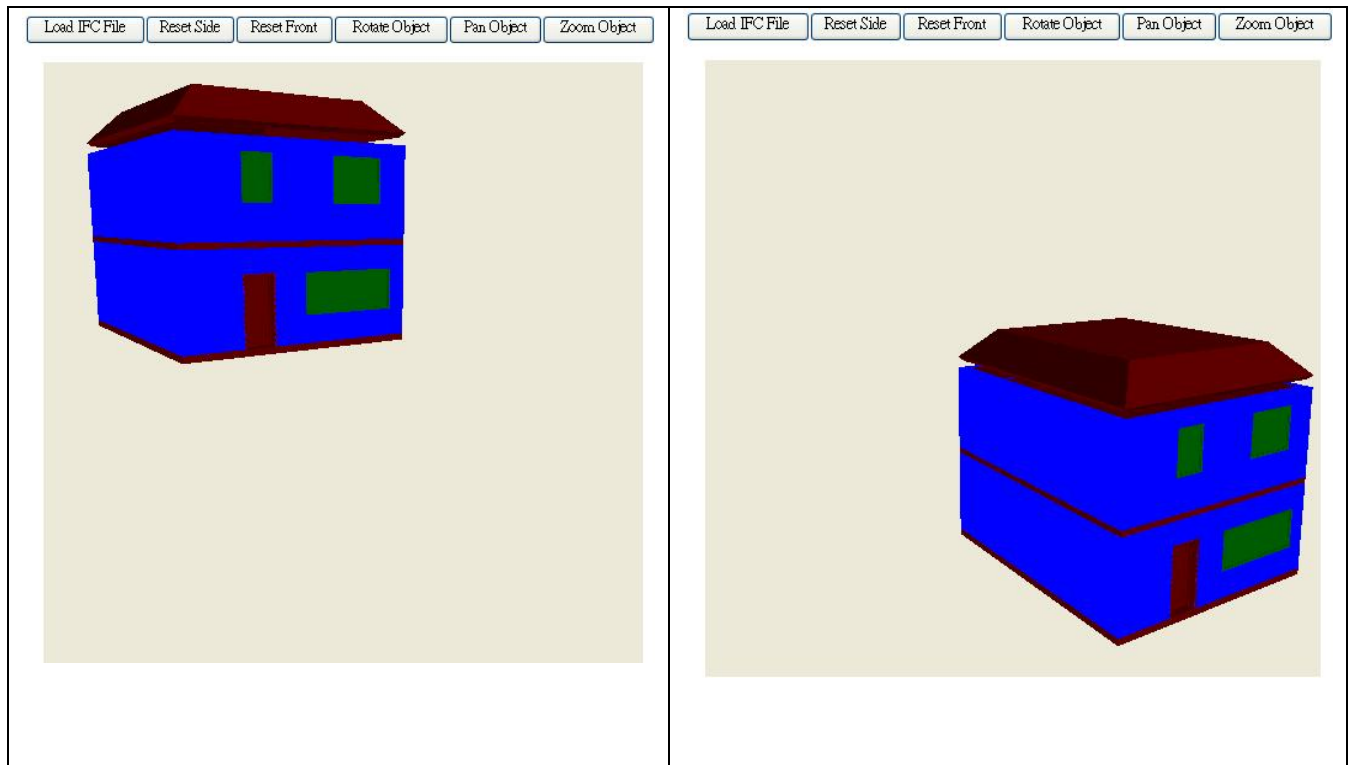


圖 3-2-3 物件拖移功能(pan)

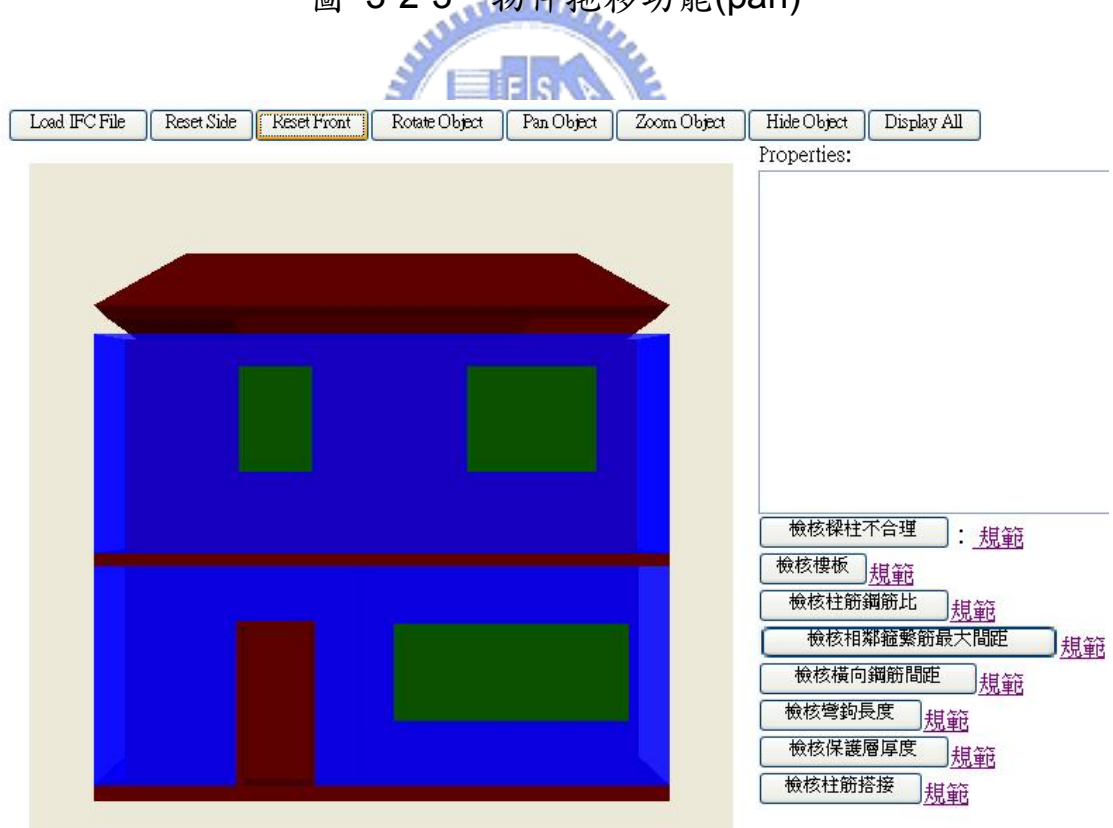


圖 3-2-4 reset front 功能，重置從物體正面觀察

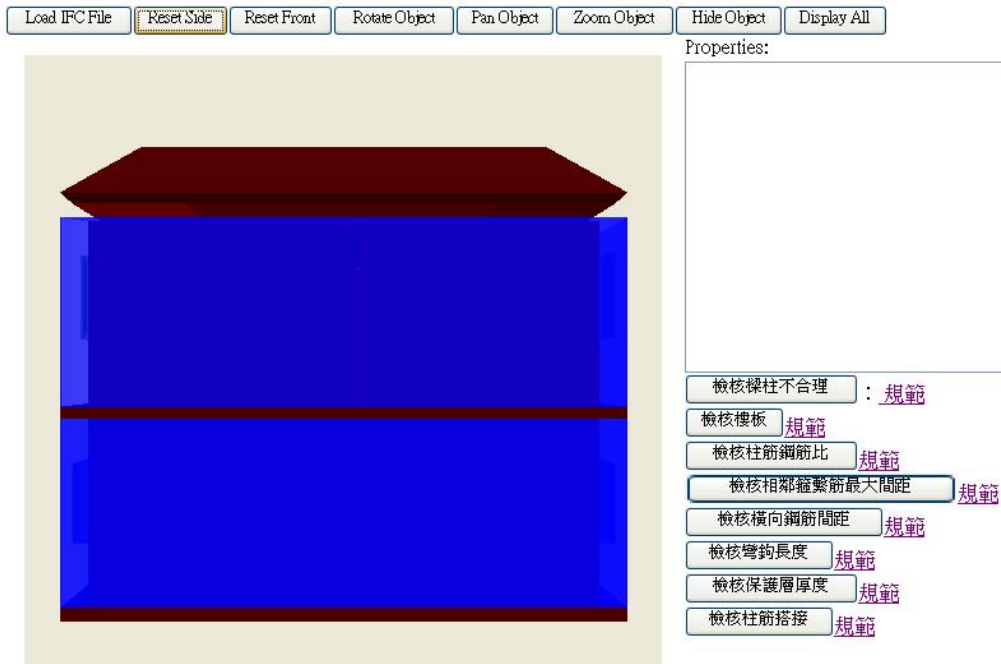


圖 3-2-5 reset side 功能，重置從物體側面觀察

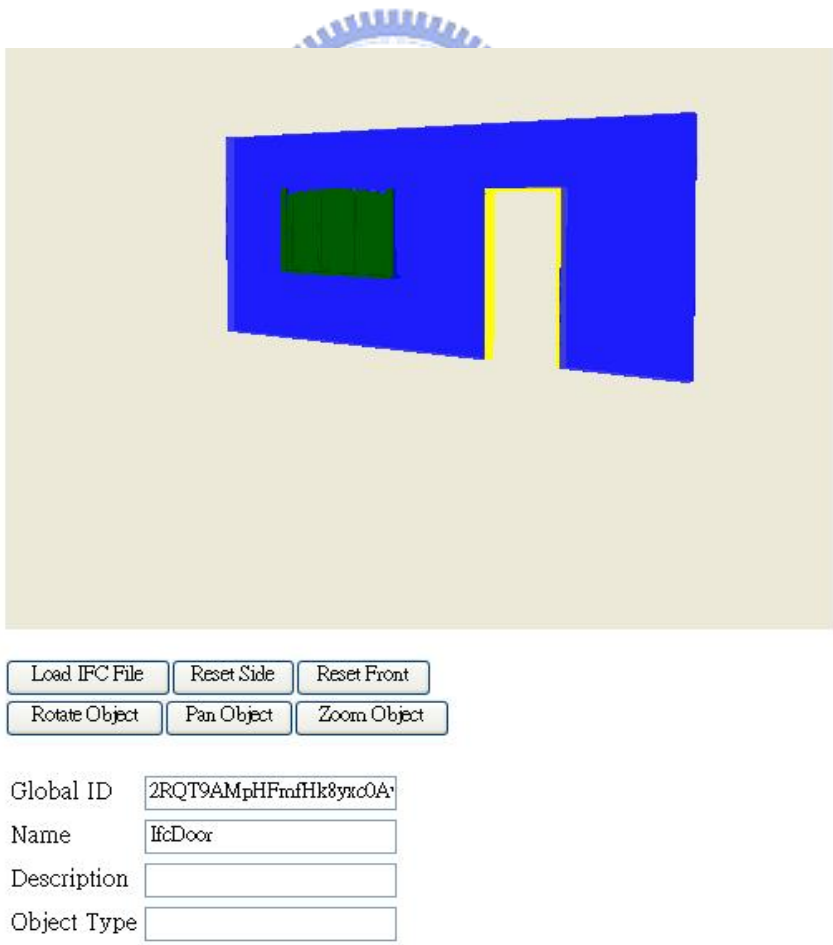


圖 3-2-6 IFC Engine DLL&OCX 的 3D viewer 功能及基本屬性的擷取

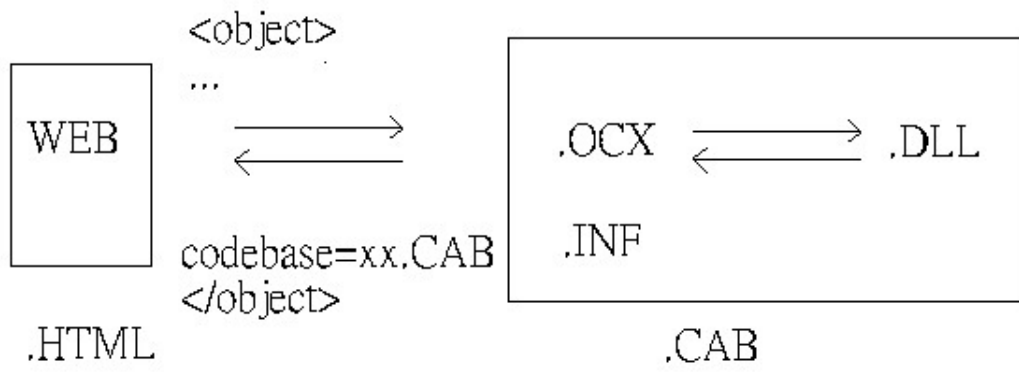


圖 3-4-1 IFC Engine DLL&OCX 運作過程

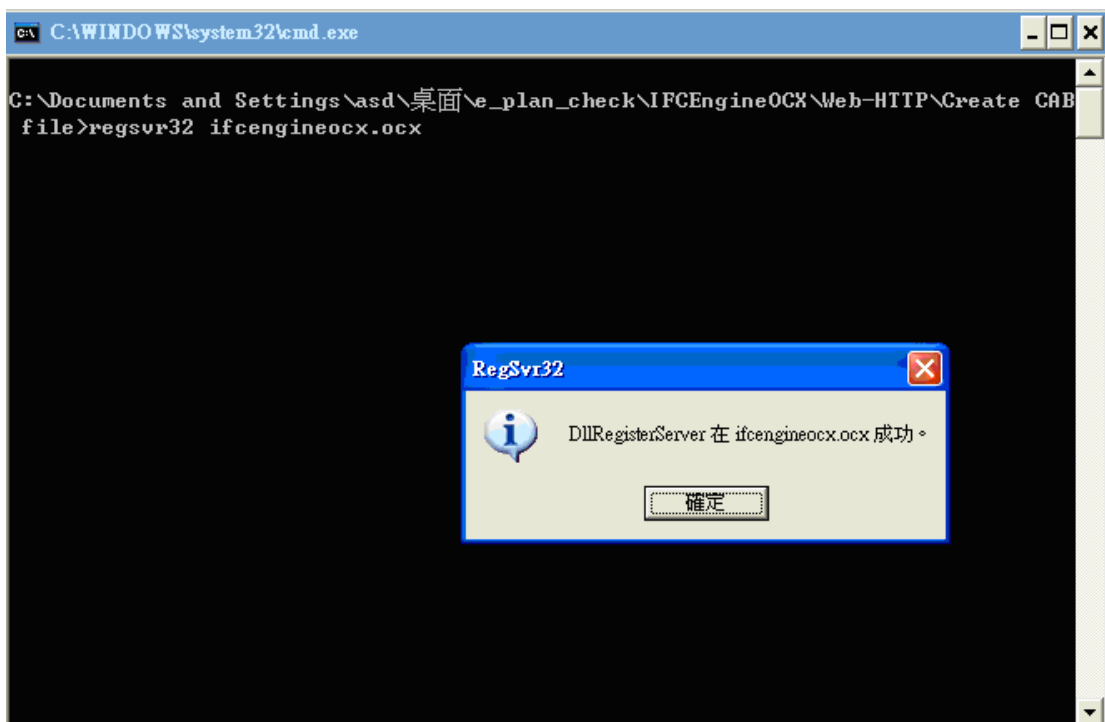


圖 3-4-2 註冊 OCX 檔

```

C:\WINDOWS\system32\cmd.exe

C:\Documents and Settings\asd\桌面\e_plan_check\IFCEngineOCX\Web-HTTP>Create CAB
file>cabarc.exe -s 6144 n ifcengine.cab ifcengineocx.ocx ifcengine.dll ifcengin
eocx.inf

Microsoft (R) Cabinet Tool - Version 1.00.0601 (03/18/97)
Copyright (c) Microsoft Corp 1996-1997. All rights reserved.

Creating new cabinet 'ifcengine.cab' with compression 'MSZIP':
-- adding ifcengineocx.ocx
-- adding ifcengine.dll
-- adding ifcengineocx.inf

Completed successfully

C:\Documents and Settings\asd\桌面\e_plan_check\IFCEngineOCX\Web-HTTP>Create CAB
file>pause
請按任意鍵繼續 . . .

```

圖 3-4-3 將 OCX 檔、DLL 檔與 INF 檔壓縮打包成 CAB 檔

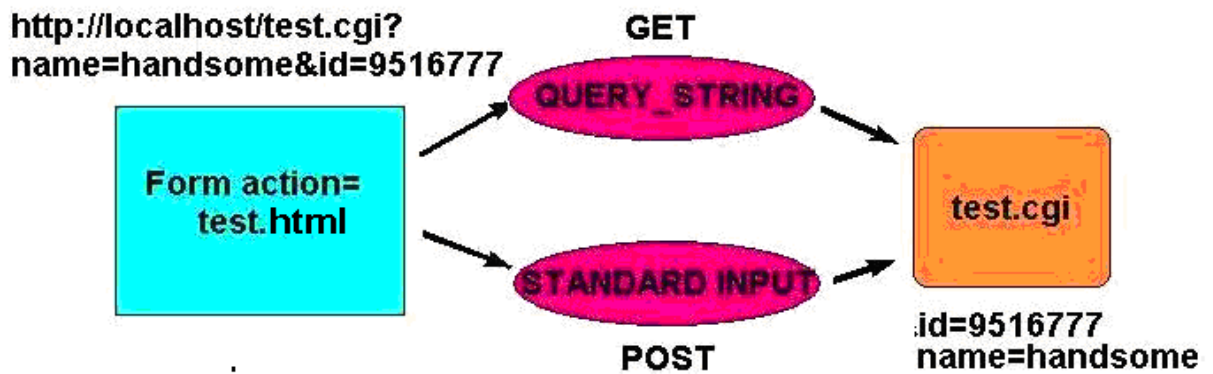


圖 3-5-1 CGI 運作流程與原理

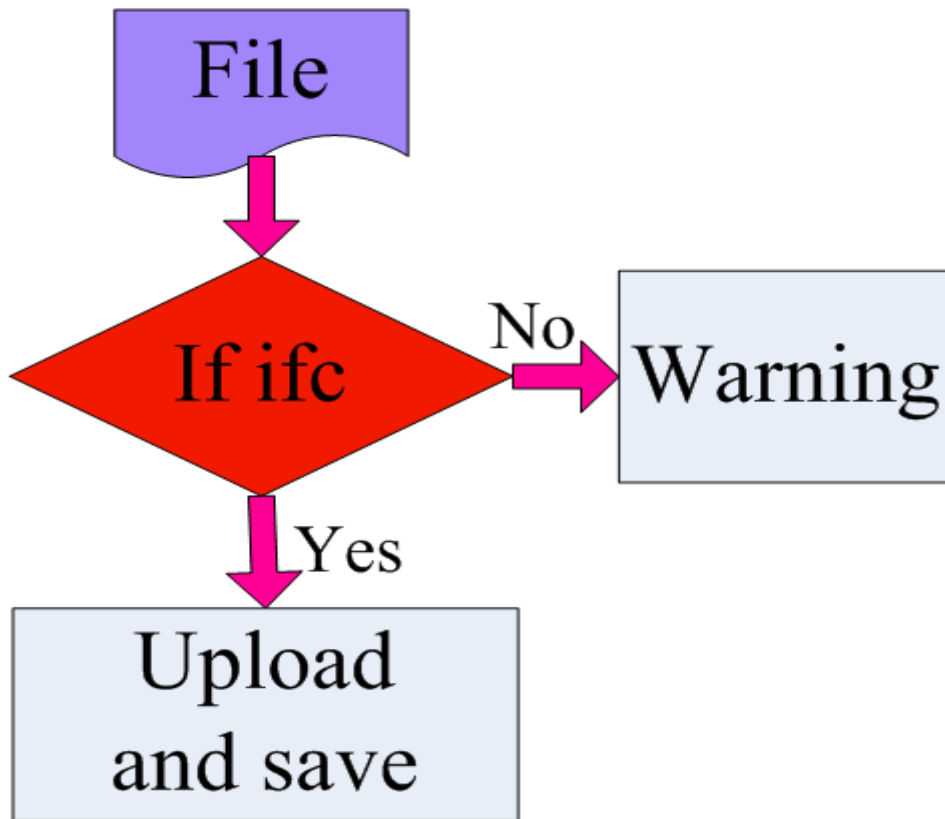


圖 4-1-1 上傳系統流程圖



圖 4-1-2 以訊息視窗來警告使用者只能傳入 IFC 格式的檔案

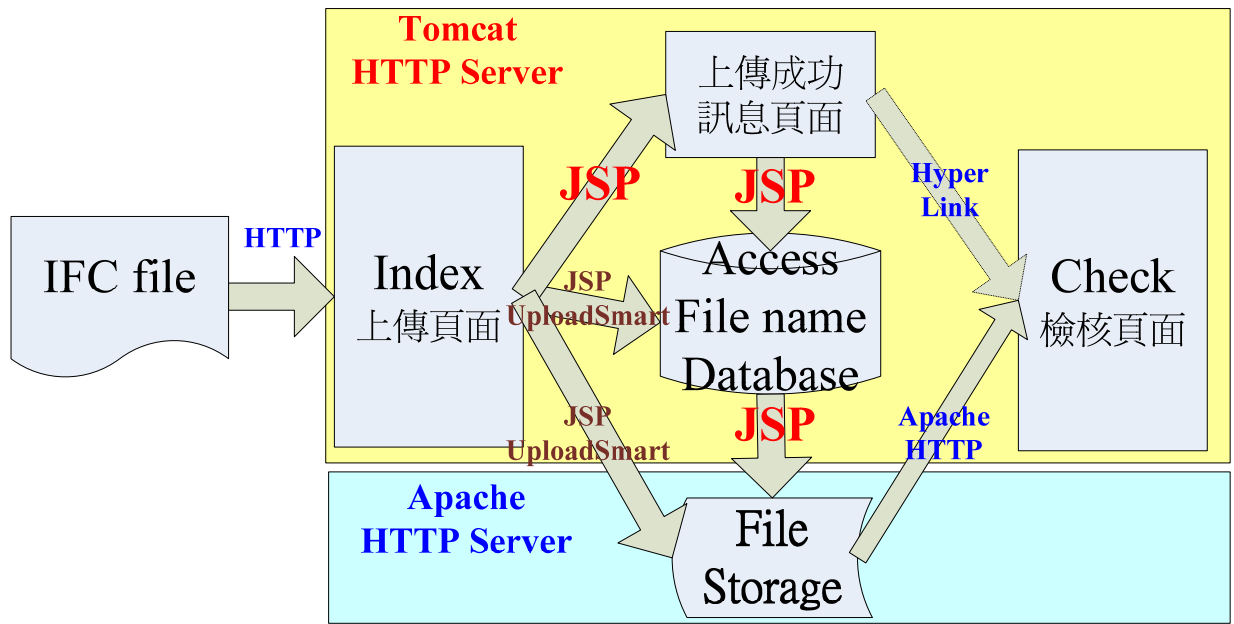


圖 4-1-3 檔案上傳詳細內部運作流程



```

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

ISO-10303-21;
HEADER;
FILE_DESCRIPTION(('ArchiCAD 10.00 Release 1 generated IFC file.','Build Number of the Ifc 2x2 interface: 55361 (31-03-
FILE_NAME('C:\Documents and Settings\lasd\My Documents\ArchiCAD\ArchiCAD\beam.ifc','2008-04-12T20:10:56','Architect'),('Building
FILE_SCHEMA(('IFC2X2_FINAL'));
ENDSEC;

DATA;
#1= IFCORGANIZATION('GS','Graphisoft','Graphisoft',$,$);
#5= IFCAPPLICATION(#1,'10.0','ArchiCAD 10.0','ArchiCAD');
#6= IFCPERSON($,'Undefined',$,$,$,$,$);
#8= IFCORGANIZATION($,'OrganizationName',$,$,$);
#12= IFCPERSONANDORGANIZATION(#6,#8,$);
#13= IFCOWNERHISTORY(#12,#5,$,'NOCHANGE',$,$,$,1208002255);
#14= IFCSIUNIT(*,'LENGTHUNIT',MILLI,METRE);
#15= IFCSIUNIT(*,'AREAUNIT',SQUARE,METRE);
#16= IFCSIUNIT(*,'VOLUMEUNIT',CUBIC,METRE);
#17= IFCSIUNIT(*,'PLANEANGLEUNIT',RADIANT);
#18= IFCMEASUREWITHUNIT(IFCPLANEANGLEMEASURE(0.017453293),#17);
#19= IFCDIMENSIONALEXONENTS(0,0,0,0,0,0);
#20= IFCCONVERSIONBASEDUNIT(#19,'PLANEANGLEUNIT','DEGREE',#18);
#21= IFCSIUNIT(*,'SOLIDANGLEUNIT',STERADIAN);
#22= IFCSIUNIT(*,'MASSUNIT',GRAM);
#23= IFCSIUNIT(*,'TIMEUNIT',SECOND);
#24= IFCSIUNIT(*,'THERMODYNAMICTEMPERATUREUNIT',DEGREE,CELSIUS);
#25= IFCSIUNIT(*,'LUMINOUSINTENSITYUNIT',LUMEN);
#26= IFCUNITASSIGNMENT((#14,#15,#16,#20,#21,#22,#23,#24,#25));
#28= IFCDIRECTION(1,0,0);
#30= IFCDIRECTION(0,1,0);
#32= IFCDIRECTION(0,0,1);
#35= IFCARTESIANPOINT(0,0,0);
#37= IFCAXIS2PLACEMENT3D(#35,#32,#28);
#38= IFCDIRECTION(6.1230318E-17,1);
#40= IFCGEOMETRICREPRESENTATIONCONTEXT('Plan','Design',3,1.0000000E-5,#37,#38);
#42= IFCPROJECT('0E8FsP3i79cwJK6Wug3W$$',#13,'Default Project',$,$,$,(#40,#117),#26);
#49= IFCLOCALPLACEMENT($,#37);
#51= IFCSITE('0BF0jrg0DA2gG0QbpsEdNd',#13,'Default Site',$,$,#49,$,$,ELEMENT,(24,28,0),(54,25,0),$,$,$);
#60= IFCLOCALPLACEMENT(#49,#37);
#62= IFCBUILDING('07pmp3Zuz5cFIU08Q18zEt',#13,'Default Building',$,$,#60,$,$,ELEMENT,$,$,$);
#71= IFCAXIS2PLACEMENT3D(#35,#32,#28);
#72= IFCLOCALPLACEMENT(#60,#71);
#74= IFCBUILDINGSTOREY('0uP6zpp3X8xumJ22gMtmY',#13,'Ground Floor',$,$,#72,$,$,ELEMENT,0);
#83= IFCCOLUMN('20CKxkIP618ocbUxgFG$$',#13,'CRE - 001',$,$,#132,#124,$);
#100= IFCDIRECTION(1,0,0);
#102= IFCARTESIANPOINT((-100,0));
#104= IFCAXIS2PLACEMENT2D(#102,#100);
#105= IFCRECTANGLEPROFILEDEF(.AREA,$,#104,200,200);
#106= IFCAXIS2PLACEMENT3D(#35,#30,#32);
#107= IFCEXTRUDEDAREASOLID(#105,#106,#32,9926.0076);
#108= IFCSHAPE REPRESENTATION(#40,'Body','SweptSolid',(#107));
#112= IFCARTESIANPOINT((-100,0,-200));
#114= IFCBOUNDINGBOX(#112,200,9926.0076,200);
#115= IFCDIRECTION(6.1230318E-17,1);
#117= IFCGEOMETRICREPRESENTATIONCONTEXT('Plan','Sketch',3,1.0000000E-5,#37,#115);
#120= IFCSHAPE REPRESENTATION(#117,'IAI','BoundingBox',(#114));
#124= IFCPRODUCTDEFINITIONSHAPE($,$,#108,#120);
#127= IFCDIRECTION(0,-1,0);
#129= IFCARTESIANPOINT((2029.1953,14012.94,2800));
#131= IFCAXIS2PLACEMENT3D(#129,#32,#127);
#132= IFCLOCALPLACEMENT(#72,#131);
#134= IFCMATERIAL('Structural Concrete');
#136= IFCRELASSOCIATESMATERIAL('0mrN3tLU54Re04UN_fpkFE',#13,$,$,(#83),#134);
#138= IFCPROPERTY SINGLEVALUE('Layername',$,IFCLABEL('Structural - Bearing'),$);
#142= IFCPROPERTY SINGLEVALUE('Red',$,IFCINTEGER(0),$);
#146= IFCPROPERTY SINGLEVALUE('Green',$,IFCINTEGER(191),$);
#150= IFCPROPERTY SINGLEVALUE('Blue',$,IFCINTEGER(191),$);
#154= IFCCOMPLEXPROPERTY('Color',$,'Color',(#142,#146,#150));
#159= IFCPROPERTY SET('jXse0hx55kOI_9gI2piCD',#13,'PSet_Draughting',$,(#138,#154));
#165= IFCREDEFINESBYPROPERTIES('2B3UDhvnncsebc6stVEQhS',#13,$,$,(#83),#159);
#167= IFCRELCONTAINEDINSPATIALSTRUCTURE('0wiVa6v85A8hKS$ikru$Mh',#13,'BuildingStoreyContainer','BuildingStoreyContaine
#169= IFCARTESIANPOINT(0,0,3100);
#171= IFCAXIS2PLACEMENT3D(#169,#32,#28);
#172= IFCLOCALPLACEMENT(#60,#171);
#174= IFCBUILDINGSTOREY('0jh91704nD38$TWFb6GCLU',#13,'',$,$,#172,$,$,ELEMENT,3100);
#183= IFCARTESIANPOINT(0,0,6200);
#185= IFCAXIS2PLACEMENT3D(#183,#32,#28);
#186= IFCLOCALPLACEMENT(#60,#185);
#188= IFCBUILDINGSTOREY('00A0TD5gj2meN8IbYQ4B6m',#13,'',$,$,#186,$,$,ELEMENT,6200);
#197= IFCREL LAGGREGATES('14M158WKLCl9LXJXEZu$gS',#13,'BuildingContainer','BuildingContainer for BuildigStories',#62,(#7
#201= IFCREL LAGGREGATES('1xK6uEj6DDyGKaia1R3RA3',#13,'SiteContainer','SiteContainer For Buildings',#51,(#62));
#203= IFCREL LAGGREGATES('13GFrPDTfBQqENRj8UX0ei',#13,'ProjectContainer','ProjectContainer For Sites',#42,(#51));
ENDSEC;

```

圖 4-2-1 IFC 檔案的內容

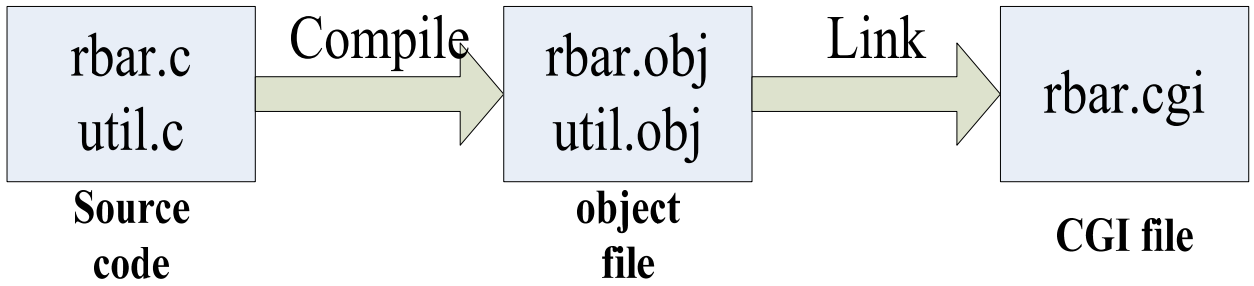


圖 4-3-1 原始碼編譯成 CGI 之過程

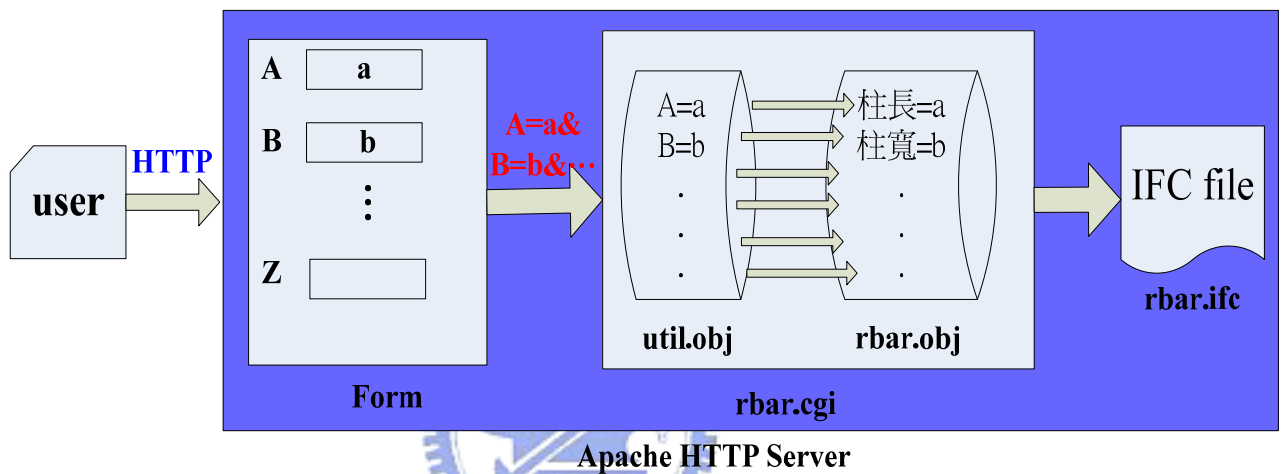


圖 4-3-2 表格資料與 CGI 技術內部參數傳遞過程

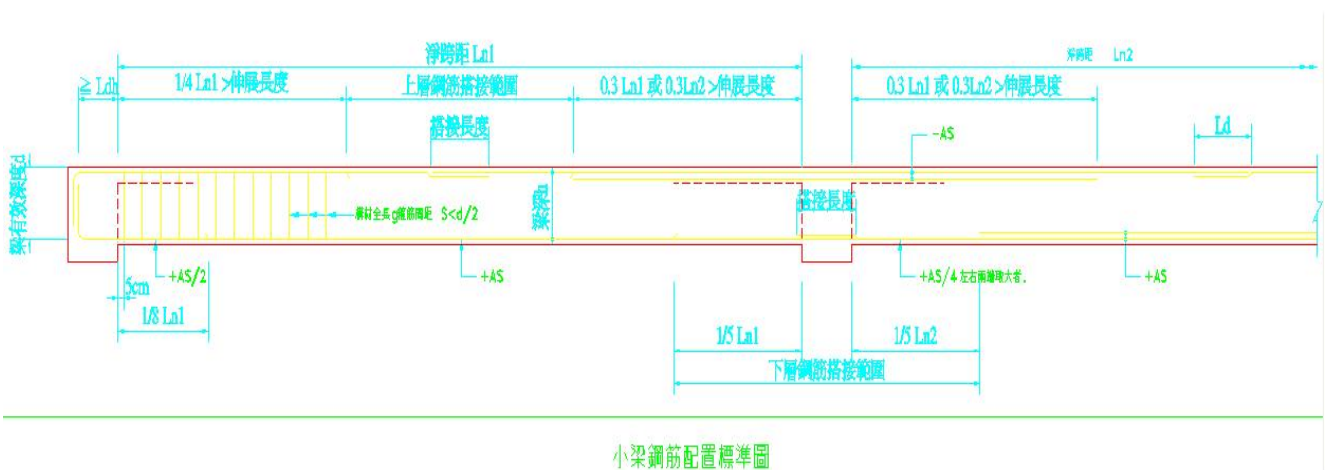


圖 4-3-3 目前業界所用鋼筋圖與鋼筋表示法 I

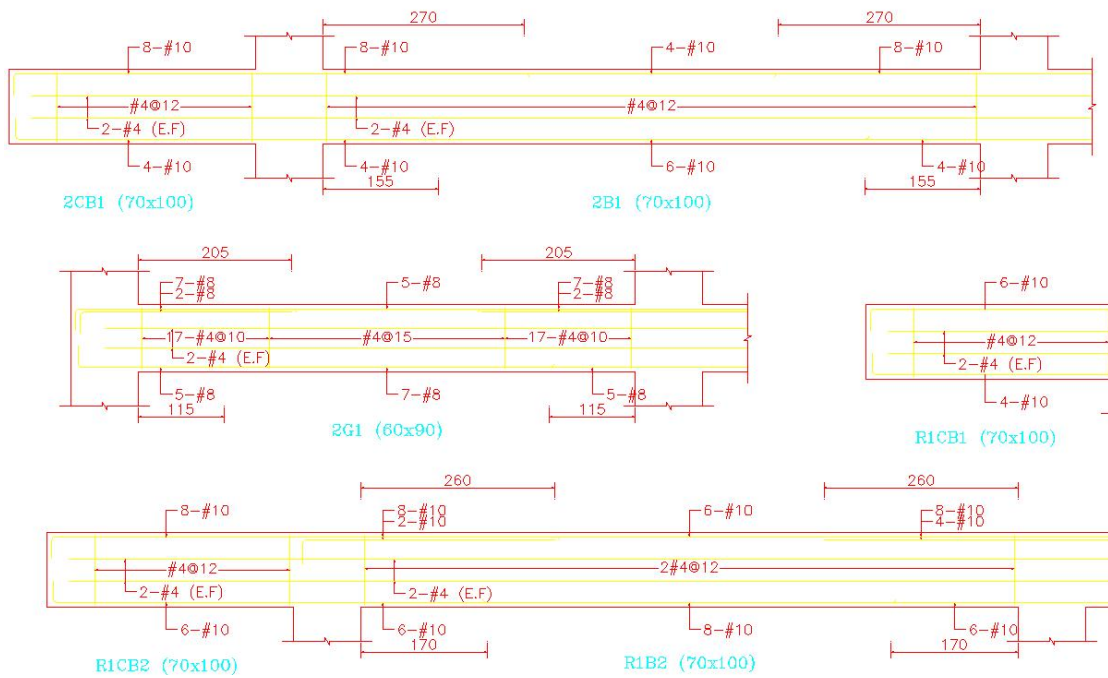


圖 4-3-4 目前業界所用鋼筋圖與鋼筋表示法 II



ENUMERATION

- **MAIN** The reinforcing bar is a main bar.
- **SHEAR** The reinforcing bar is a shear bar.
- **LINK** The reinforcing bar is a link (stirrup).
- **STUD** The reinforcing bar is a stud.
- **PUNCHING** Punching reinforcement.
- **EDGE** Edge reinforcement.
- **RING** Ring reinforcement.
- **ANCHORING** Anchoring reinforcement.
- **USERDEFINED** The type of reinforcement is user defined.
- **NOTDEFINED** The type of reinforcement is not defined.

EXPRESS specification:

```

TYPE IfcReinforcingBarRoleEnum = ENUMERATION OF
  (MAIN,
   SHEAR,
   LIGATURE,
   STUD,
   PUNCHING,
   EDGE,
   RING,
   USERDEFINED,
   NOTDEFINED);
END_TYPE;

```

圖 4-3-5 IFC 鋼筋種類 BarRole 之列舉型態(Enumeration)

ENUMERATION

- **PLAIN** The reinforcing bar surface is plain.
- **TEXTURED** The reinforcing bar surface is textured (ribbed).

EXPRESS specification:

```
TYPE IfcReinforcingBarSurfaceEnum = ENUMERATION OF  
  (PLAIN,  
   TEXTURED);  
END_TYPE;
```

圖 4-3-6 IFC 鋼筋表面種類 BarSurface 之列舉型態(Enumeration)

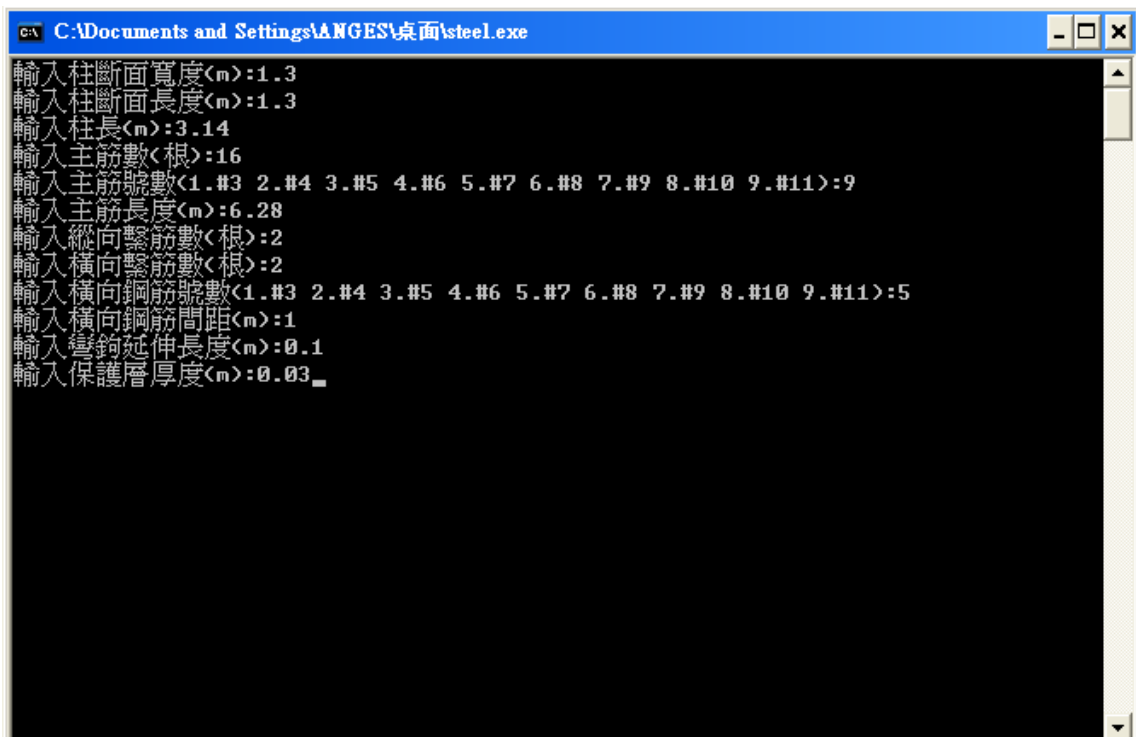


圖 4-3-7 柱筋產生器執行時參數輸入的過程

```
C:\WINDOWS\system32\cmd.exe
C:\src>cl rbar.c
Microsoft (R) 32-bit C/C++ Standard Compiler Version 12.00.8168 for 80x86
Copyright (C) Microsoft Corp 1984-1998. All rights reserved.

rbar.c
rbar.c(46) : warning C4047: '=' : 'char *' differs in levels of indirection from
'int '
rbar.c(49) : warning C4047: '=' : 'char *' differs in levels of indirection from
'int '
Microsoft (R) Incremental Linker Version 6.00.8168
Copyright (C) Microsoft Corp 1992-1998. All rights reserved.

/out:rbar.exe
rbar.obj
rbar.obj : error LNK2001: unresolved external symbol _makeword
rbar.obj : error LNK2001: unresolved external symbol _unescape_url
rbar.obj : error LNK2001: unresolved external symbol _plustospace
rbar.obj : error LNK2001: unresolved external symbol _fnakeword
rbar.exe : fatal error LNK1120: 4 unresolved externals

C:\src>
```

圖 4-3-8 命令列下編譯 rbar.c

```
C:\WINDOWS\system32\cmd.exe
C:\src>cl util.c
Microsoft (R) 32-bit C/C++ Standard Compiler Version 12.00.8168 for 80x86
Copyright (C) Microsoft Corp 1984-1998. All rights reserved.

util.c
Microsoft (R) Incremental Linker Version 6.00.8168
Copyright (C) Microsoft Corp 1992-1998. All rights reserved.

/out:util.exe
util.obj
LIBC.lib(crt0.obj) : error LNK2001: unresolved external symbol _main
util.exe : fatal error LNK1120: 1 unresolved externals

C:\src>
```

圖 4-3-9 命令列下編譯 util.c

```
C:\WINDOWS\system32\cmd.exe
C:\src>cl -o rbar.cgi rbar.obj util.obj
Microsoft (R) 32-bit C/C++ Standard Compiler Version 12.00.8168 for 80x86
Copyright (C) Microsoft Corp 1984-1998. All rights reserved.

Microsoft (R) Incremental Linker Version 6.00.8168
Copyright (C) Microsoft Corp 1992-1998. All rights reserved.

/out:rbar.exe
/out:rbar.cgi
rbar.obj
util.obj

C:\src>
```

圖 4-3-10 命令列下連結 rbar.obj 與 util.obj 成為 cgi 檔

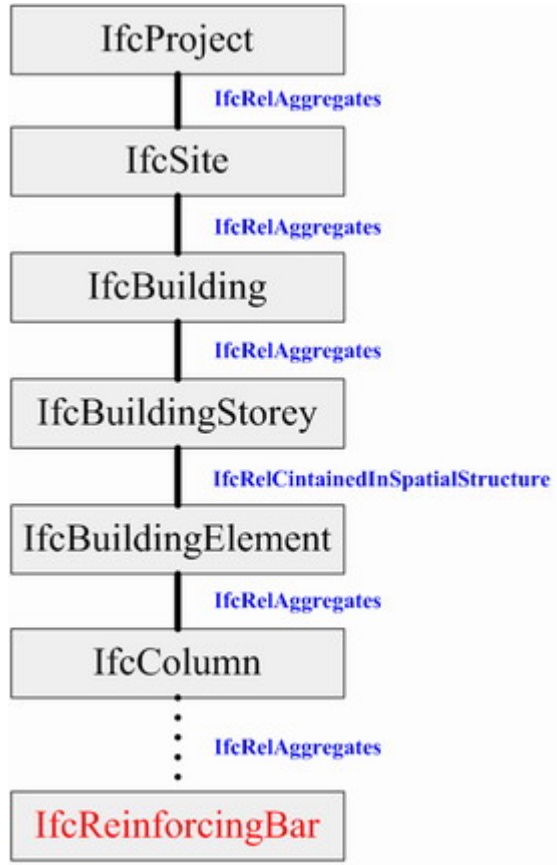


圖 4-3-11 IFC 資訊架構圖

IFC file

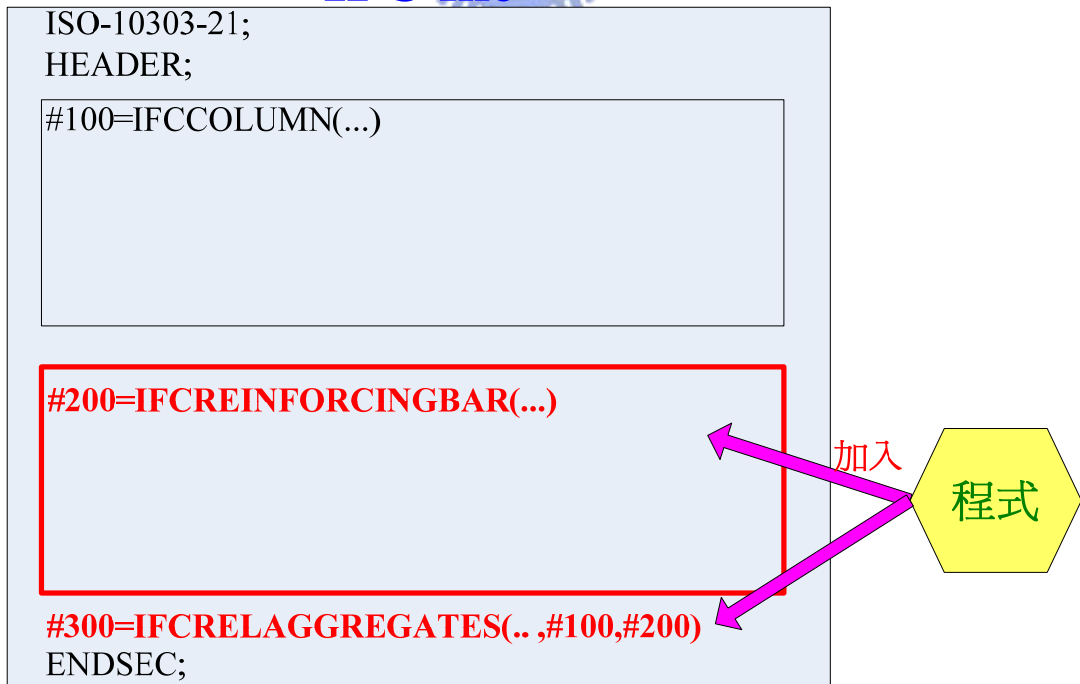


圖 4-3-12 加入鋼筋的實體 IFCReinforcingBar，並連接 IFCColumn

```

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

FILE_SCHEMA (('IFC2X3'));
ENDSEC;

DATA;
#10=IFCCIRCLEPROFILEDEF(.AREA.,$, #27,0.00795);
#11=IFCCIRCLEPROFILEDEF(.AREA.,$, #28,0.00795);
#12=IFCCIRCLEPROFILEDEF(.AREA.,$, #29,0.00795);
#13=IFCCIRCLEPROFILEDEF(.AREA.,$, #30,0.00795);
#14=IFCCIRCLEPROFILEDEF(.AREA.,$, #31,0.00795);
#15=IFCCIRCLEPROFILEDEF(.AREA.,$, #32,0.00795);
#16=IFCCIRCLEPROFILEDEF(.AREA.,$, #33,0.00795);
#17=IFCCIRCLEPROFILEDEF(.AREA.,$, #34,0.00795);
#18=IFCREINFORCINGBAR('710YmXq1t0+aViTepzFpaA', $, '#5', 'Main', $, #75, #55, $, 'Gr50', 0.1, 0.2, 0.03, .MAIN., .PLAIN.);
#19=IFCREINFORCINGBAR('EQpIRC/Yi0SFLr2RohoJQQ', $, '#5', 'Main', $, #76, #56, $, 'Gr50', 0.1, 0.2, 0.03, .MAIN., .PLAIN.);
#20=IFCREINFORCINGBAR('6UfiG2ub5UeW0TTHur83cA', $, '#5', 'Main', $, #77, #57, $, 'Gr50', 0.1, 0.2, 0.03, .MAIN., .PLAIN.);
#21=IFCREINFORCINGBAR('buic5P6VSEyootmWmrUwA', $, '#5', 'Main', $, #78, #58, $, 'Gr50', 0.1, 0.2, 0.03, .MAIN., .PLAIN.);
#22=IFCREINFORCINGBAR('mqx8E6IjiUWvufbqB0augw', $, '#5', 'Main', $, #79, #59, $, 'Gr50', 0.1, 0.2, 0.03, .MAIN., .PLAIN.);
#23=IFCREINFORCINGBAR('op2vaTX9aUOkfFrr40j0aQ', $, '#5', 'Main', $, #80, #60, $, 'Gr50', 0.1, 0.2, 0.03, .MAIN., .PLAIN.);
#24=IFCREINFORCINGBAR('mvdTRrYsGUuwlqCMZuzJMG', $, '#5', 'Main', $, #81, #61, $, 'Gr50', 0.1, 0.2, 0.03, .MAIN., .PLAIN.);
#25=IFCREINFORCINGBAR('BCQfnWwKRGzE4tMLrFttA', $, '#5', 'Main', $, #82, #62, $, 'Gr50', 0.1, 0.2, 0.03, .MAIN., .PLAIN.);
#26=IFCAXIS2PLACEMENT2D(#140, #87);
#27=IFCAXIS2PLACEMENT2D(#142, #91);
#28=IFCAXIS2PLACEMENT2D(#145, #97);
#29=IFCAXIS2PLACEMENT2D(#148, #103);
#30=IFCAXIS2PLACEMENT2D(#151, #109);
#31=IFCAXIS2PLACEMENT2D(#154, #115);
#32=IFCAXIS2PLACEMENT2D(#157, #121);
#33=IFCAXIS2PLACEMENT2D(#160, #127);
#34=IFCAXIS2PLACEMENT2D(#163, #133);
#35=IFCRECTANGLEPROFILEDEF(.AREA., $, #26, 0.2, 0.15);
#36=IFCETRUEDEDAREASOLID(#35, #168, #84, 2.91204395571221);
#37=IFCETRUEDEDAREASOLID(#10, #170, #94, 2.91204395571221);
#38=IFCETRUEDEDAREASOLID(#11, #172, #100, 2.91204395571221);
#39=IFCETRUEDEDAREASOLID(#12, #174, #106, 2.91204395571221);
#40=IFCETRUEDEDAREASOLID(#13, #176, #112, 2.91204395571221);
#41=IFCETRUEDEDAREASOLID(#14, #178, #118, 2.91204395571221);
#42=IFCETRUEDEDAREASOLID(#15, #180, #124, 2.91204395571221);
#43=IFCETRUEDEDAREASOLID(#16, #182, #130, 2.91204395571221);
#44=IFCETRUEDEDAREASOLID(#17, #184, #136, 2.91204395571221);
#45=IFCSHAPEREPRESENTATION(#185, 'Body', 'SweptSolid', (#36));
#46=IFCSHAPEREPRESENTATION($, 'Body', 'SweptSolid', (#37));
#47=IFCSHAPEREPRESENTATION($, 'Body', 'SweptSolid', (#38));
#48=IFCSHAPEREPRESENTATION($, 'Body', 'SweptSolid', (#39));
#49=IFCSHAPEREPRESENTATION($, 'Body', 'SweptSolid', (#40));
#50=IFCSHAPEREPRESENTATION($, 'Body', 'SweptSolid', (#41));
#51=IFCSHAPEREPRESENTATION($, 'Body', 'SweptSolid', (#42));
#52=IFCSHAPEREPRESENTATION($, 'Body', 'SweptSolid', (#43));
#53=IFCSHAPEREPRESENTATION($, 'Body', 'SweptSolid', (#44));
#54=IFCPRODUCTDEFINITIONSHAPE($, $, (#45));
#55=IFCPRODUCTDEFINITIONSHAPE($, $, (#46));
#56=IFCPRODUCTDEFINITIONSHAPE($, $, (#47));
#57=IFCPRODUCTDEFINITIONSHAPE($, $, (#48));
#58=IFCPRODUCTDEFINITIONSHAPE($, $, (#49));
#59=IFCPRODUCTDEFINITIONSHAPE($, $, (#50));
#60=IFCPRODUCTDEFINITIONSHAPE($, $, (#51));
#61=IFCPRODUCTDEFINITIONSHAPE($, $, (#52));
#62=IFCPRODUCTDEFINITIONSHAPE($, $, (#53));
#63=IFCCOLUMN('KqWYn1vKk0qEnPGe0u32hw', #196, 'Col:1', $, $, #74, #54, $);
#64=IFCRELCONTAINEDINSPATIALSTRUCTURE('Nv4mmMq5a0uBFA/KvatXiQ', #196, 'for Building Element', $, (#63), #65);
#65=IFCBUILDINGSTOREY('j7QyRrwHUESHsP4biXT3FA', #196, 'Ground', $, $, #73, $, $, .ELEMENT., 0.);
#66=IFCBUILDING('EJF9TIRc6EArXB6m8BzVzQ', #196, 'Building', $, $, #72, $, $, .ELEMENT., 0., 0., $);
#67=IFCRELAGGREGATES('gBJm+7UM0k2afMb/2b8S9w', #196, 'ProjectContainer for Sites', $, #186, (#83));
#68=IFCRELAGGREGATES('zR8UkzyX0EuCQQWsuU24uA', #196, 'SiteContainer for Buildings', $, #83, (#66));
#69=IFCRELAGGREGATES('nJH5HIN36kSHjhixrJuWUq', #196, 'BuildingContainer for BuildiqStories', $, #66, (#65));
#70=IFCRELAGGREGATES('8bFGwIKwA02e+uENWYu0Ag', $, $, $, #63, (#18, #19, #20, #21, #22, #23, #24, #25));

```

圖 4-3-13 鋼筋 IFC 產生器所產生的 IFC 檔

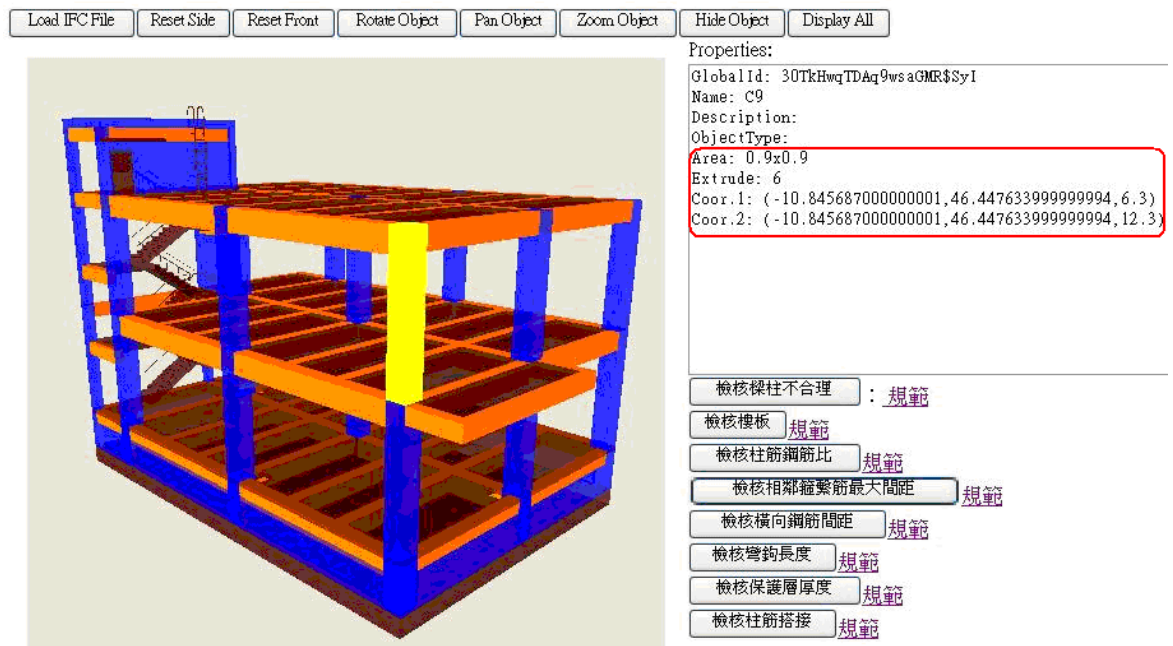


圖 4-4-1 新增功能-抽出柱斷面長度與寬度、柱高與兩端斷面的中心座標

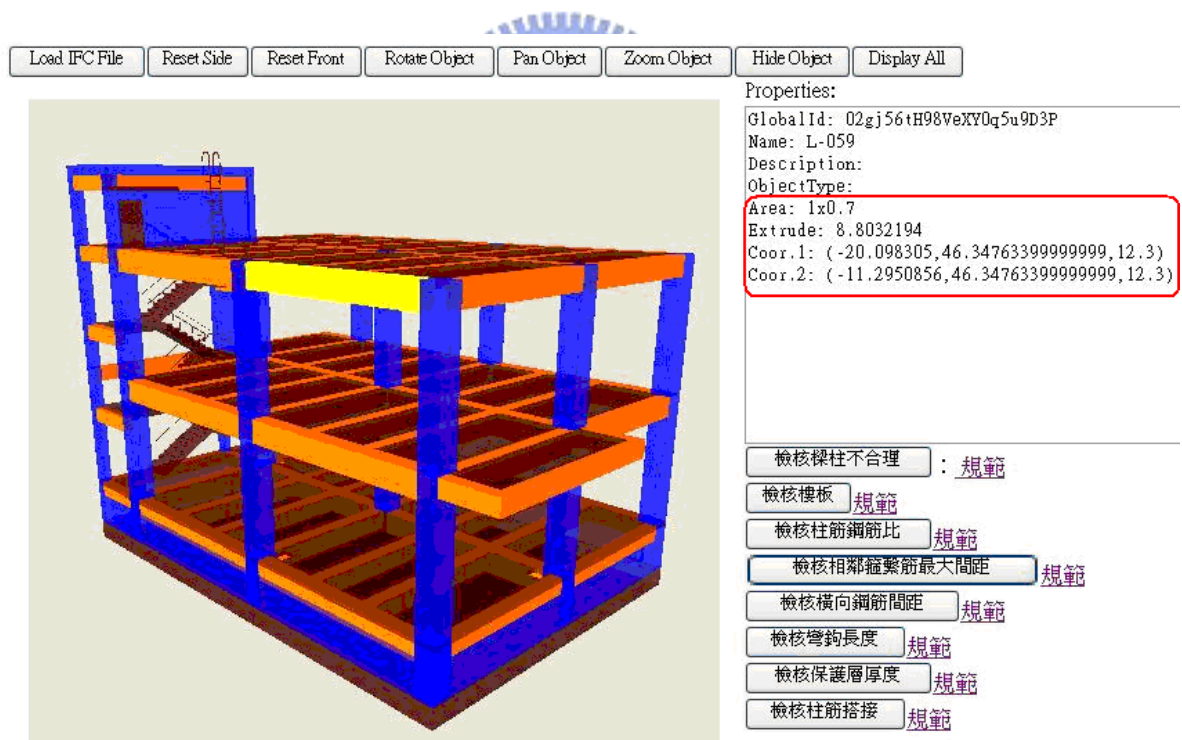


圖 4-4-2 新增功能-抽出梁斷面長度與寬度、梁深與兩端斷面的中心座標

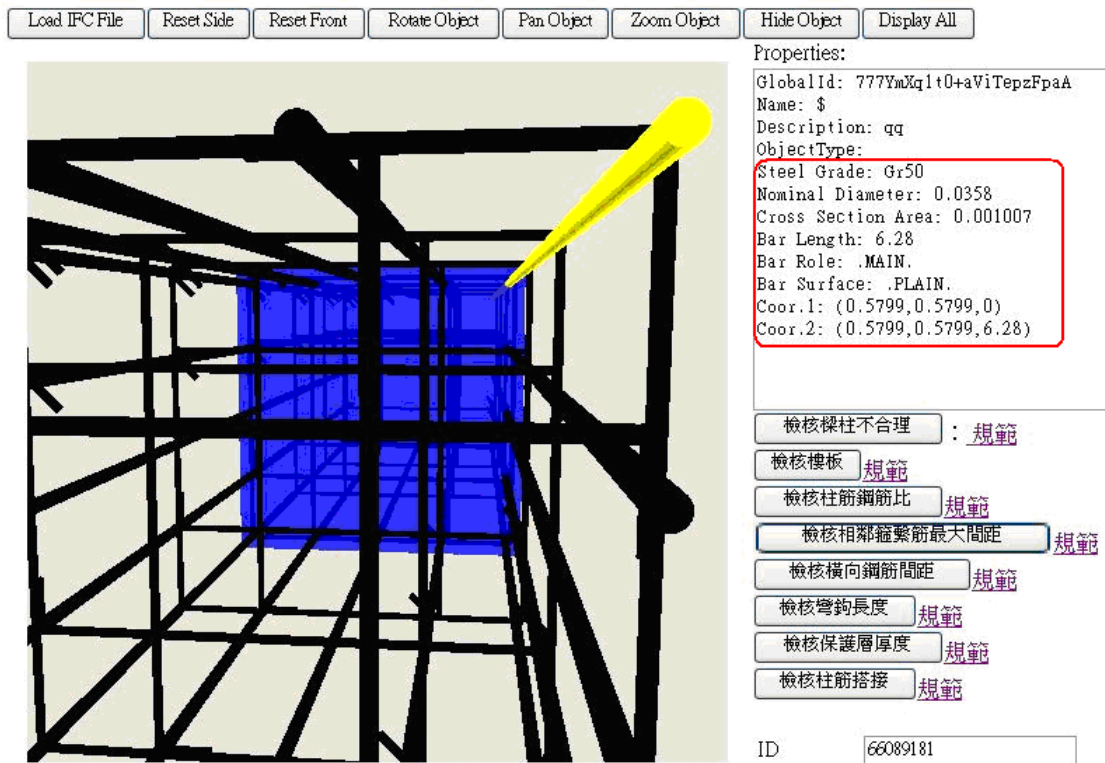


圖 4-4-3 新增功能-抽取鋼筋等級、斷面直徑、斷面積、鋼筋長度、鋼筋種類、鋼筋表面形狀與兩端斷面的圓心座標



圖 4-4-4 新增功能-抽取樓板面積、樓板厚度及樓板四個角點的座標

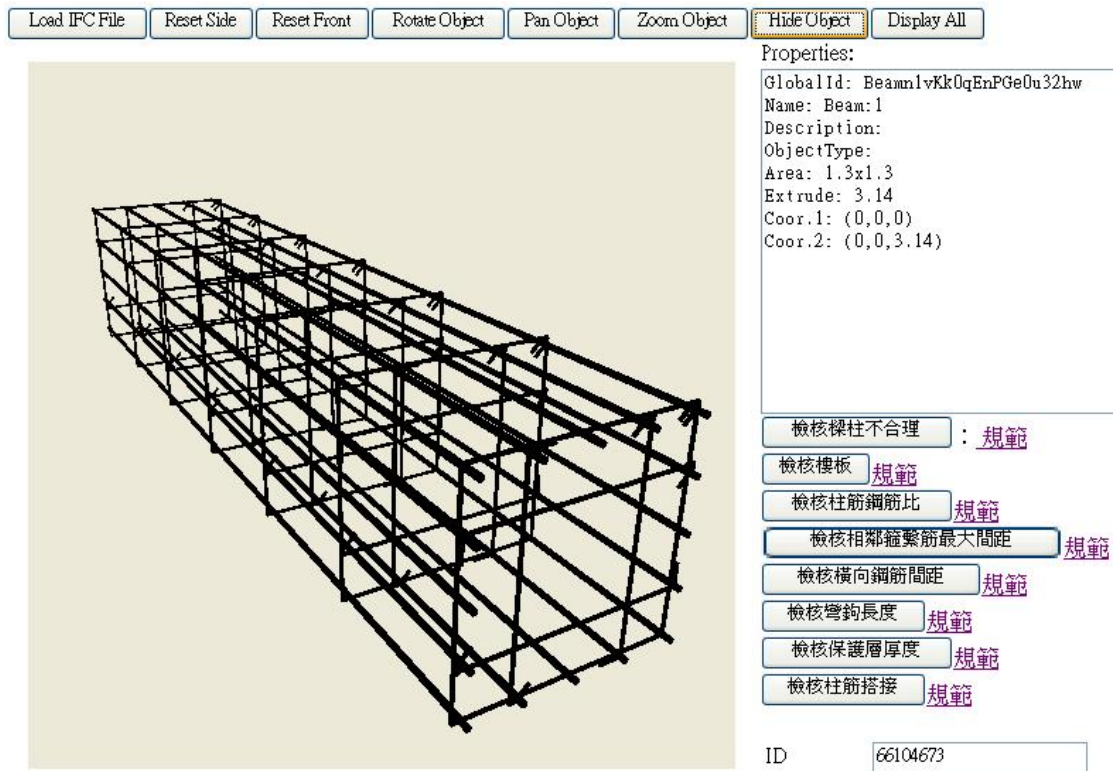


圖 4-4-5 隱藏物件功能(Hide)，本圖隱藏了原有的柱子，僅剩柱筋裸露

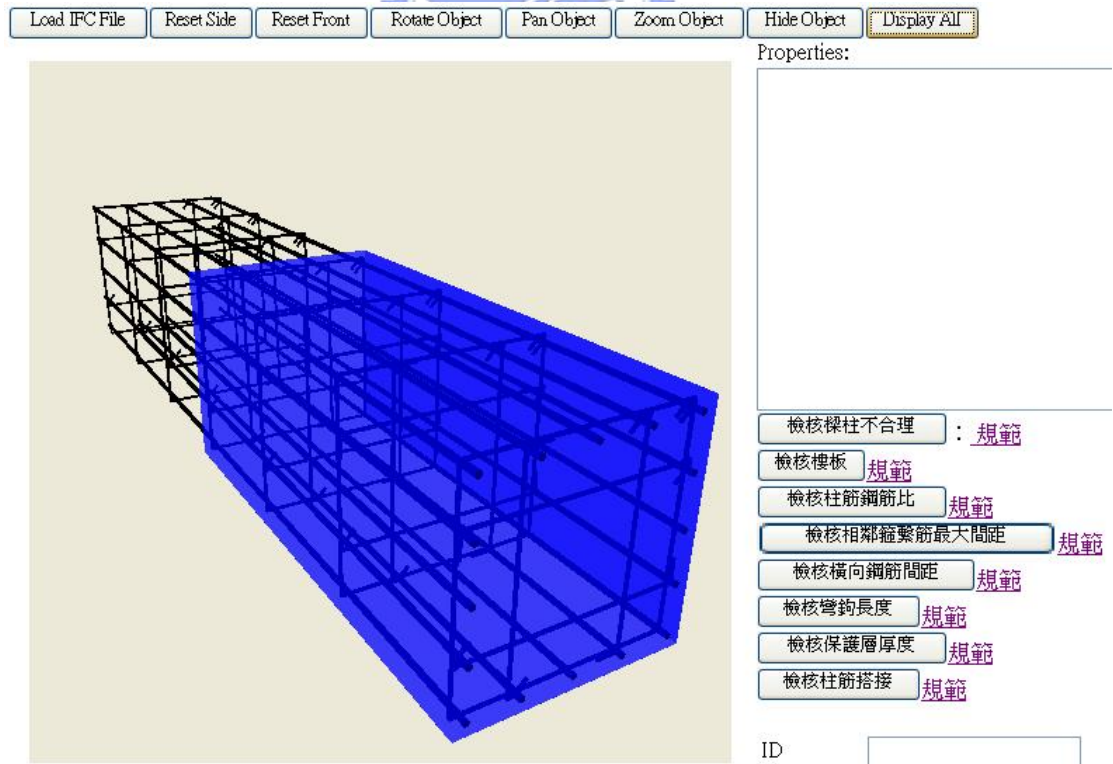


圖 4-4-6 顯示所有物件(Display All)喚醒所有隱藏物件

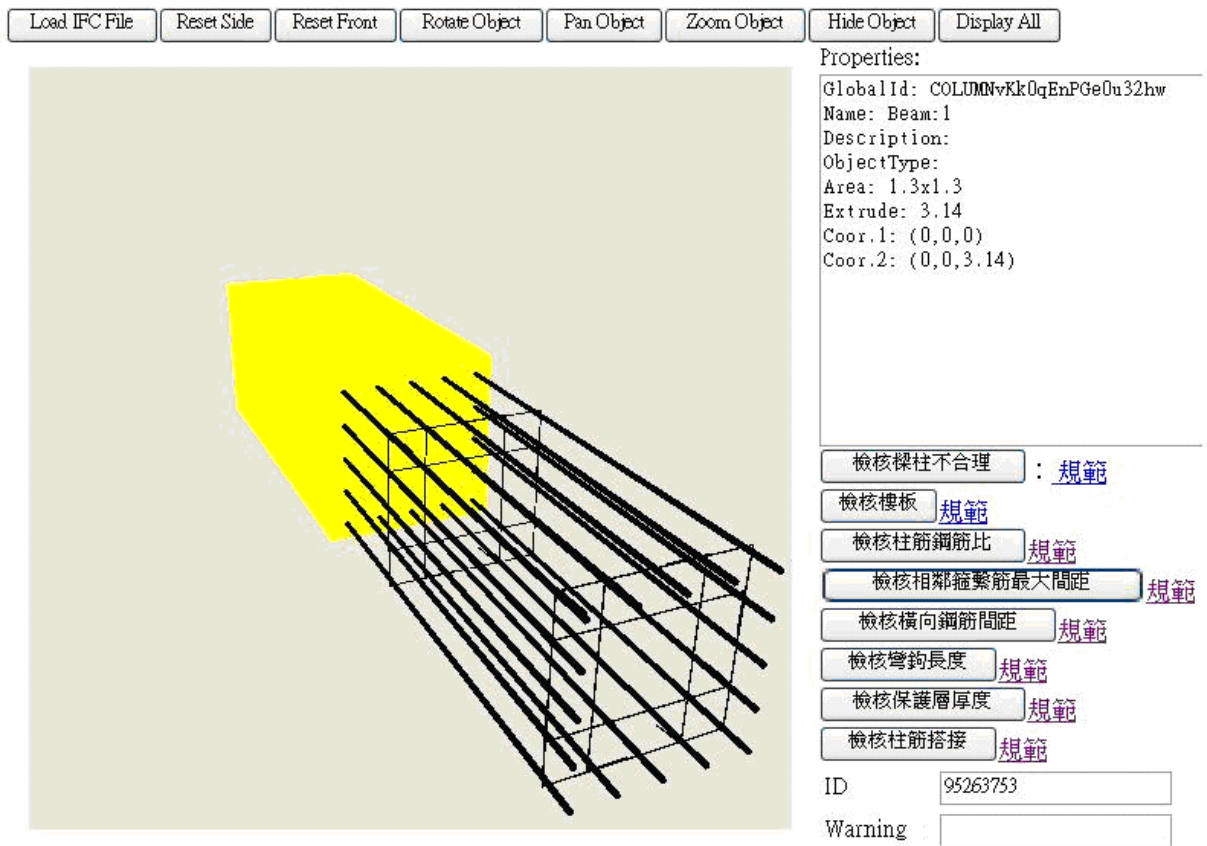


圖 4-4-7 被點選的物件變成黃色

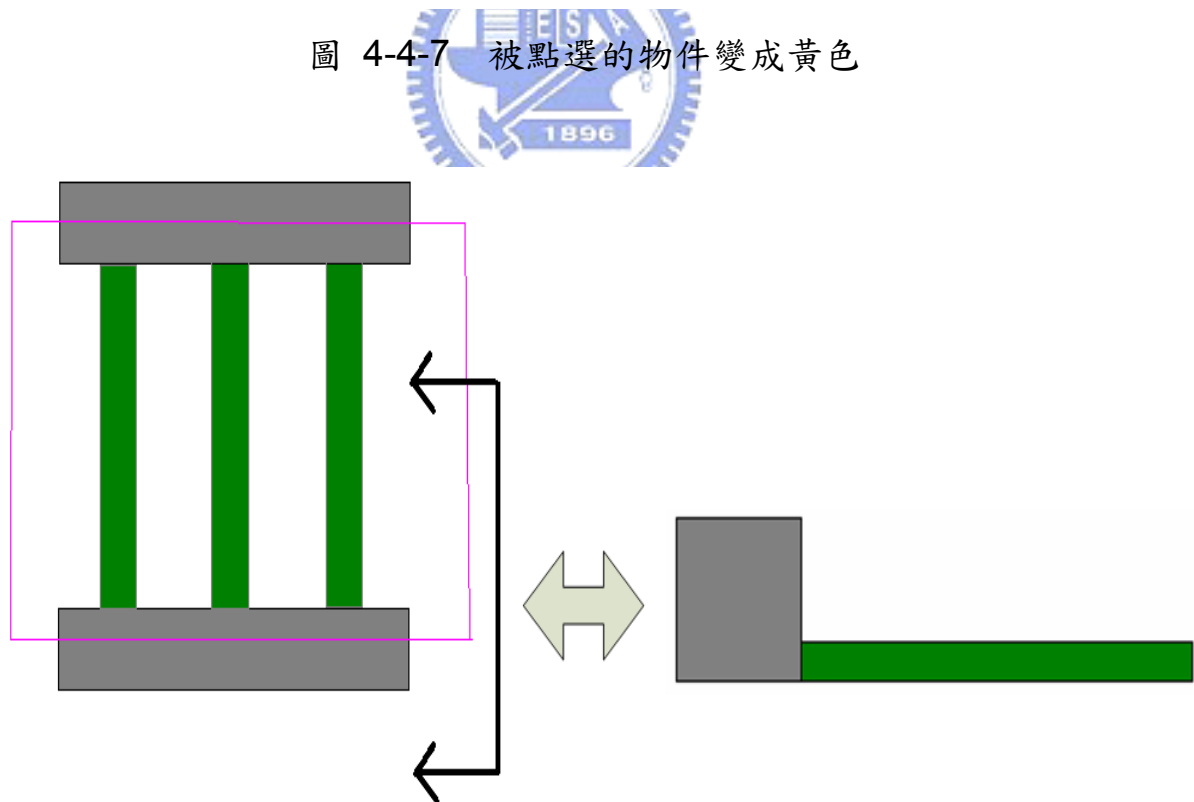


圖 4-5-1 大梁跨小梁示意圖，左為俯視圖，右為側視圖

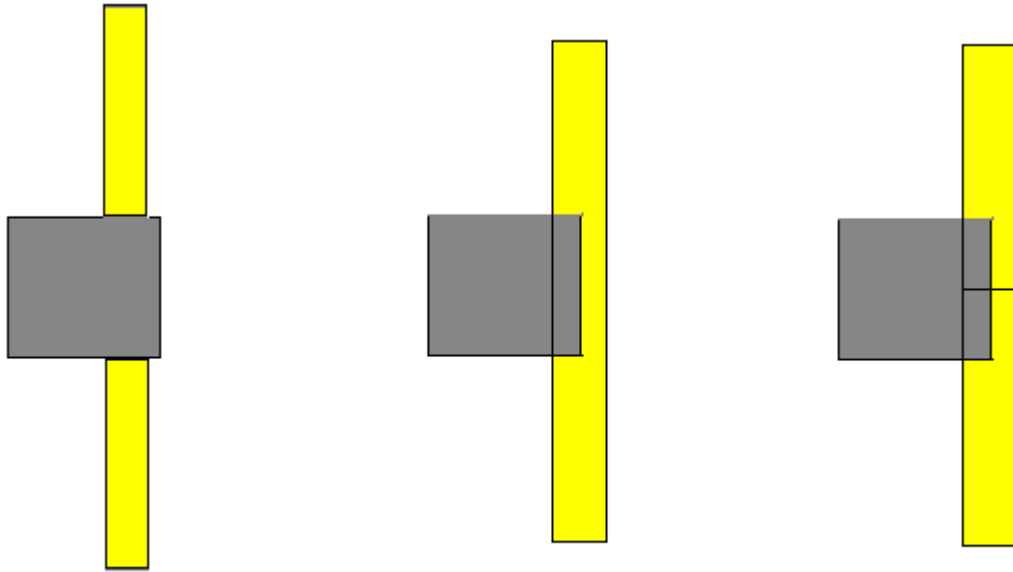


圖 4-5-2 梁身超過柱俯視圖，右邊兩根梁，梁身都超過柱

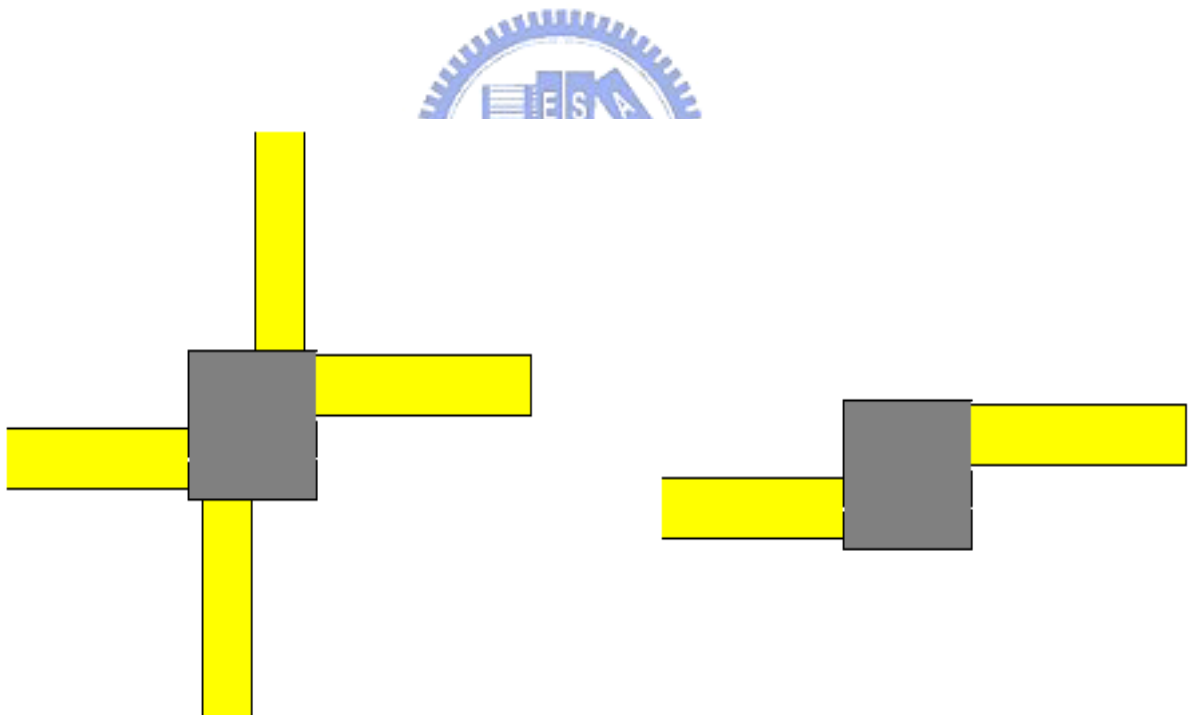


圖 4-5-3 梁跨柱不在同一直線上之俯視圖，右圖為橫向的梁不在同一直線上，左圖橫向與縱向的梁皆不在同一直線上

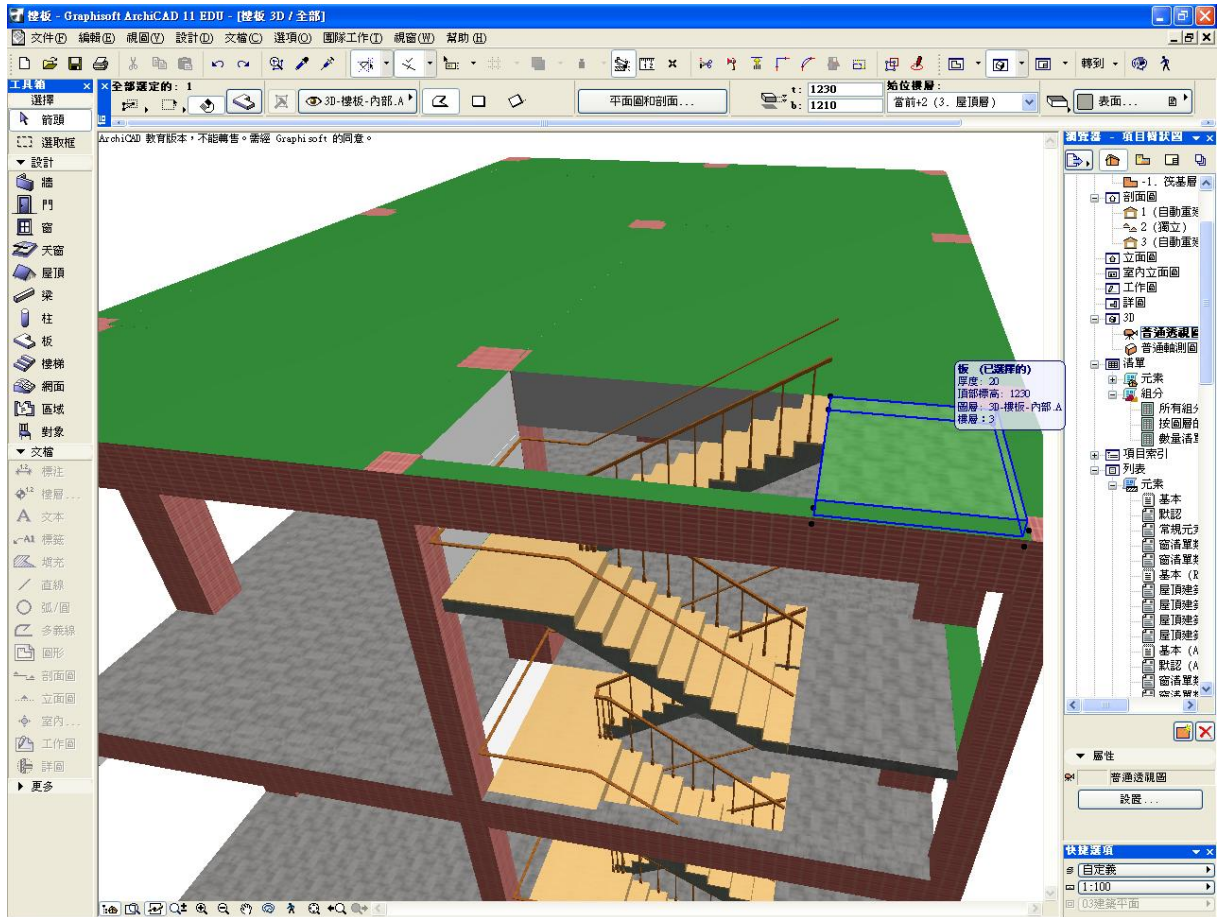


圖 4-5-4 因為要做樓梯之故要去掉樓板

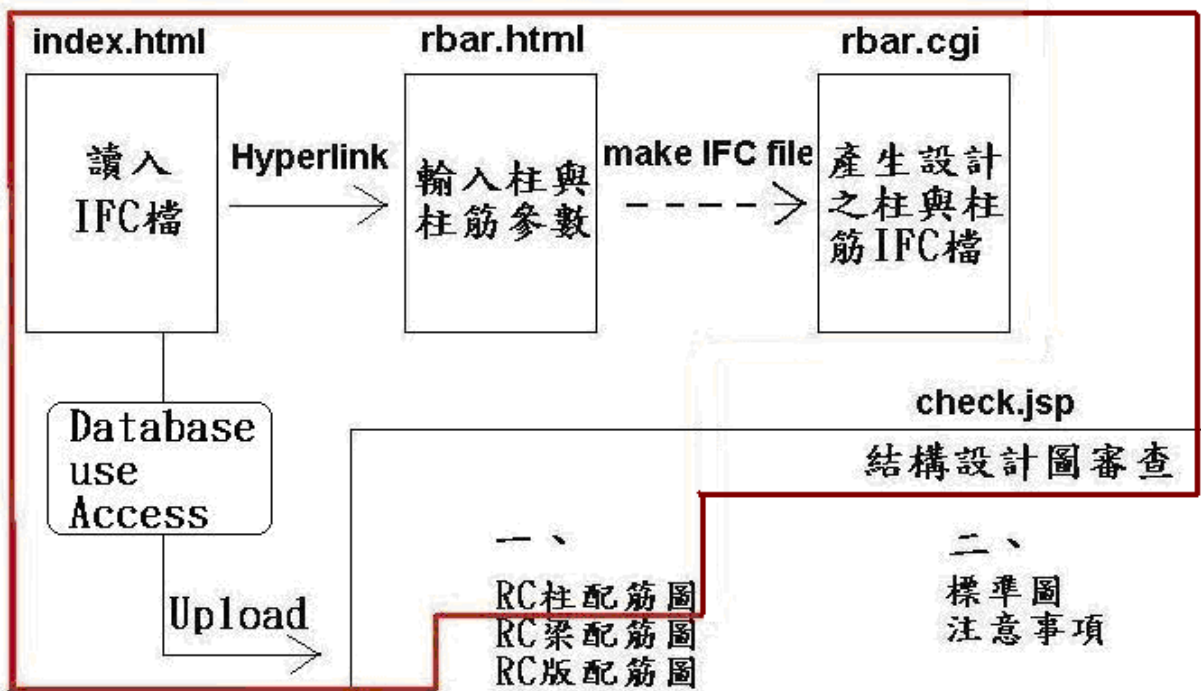


圖 5-1-1 紅色框選部分為本研究實作的部份

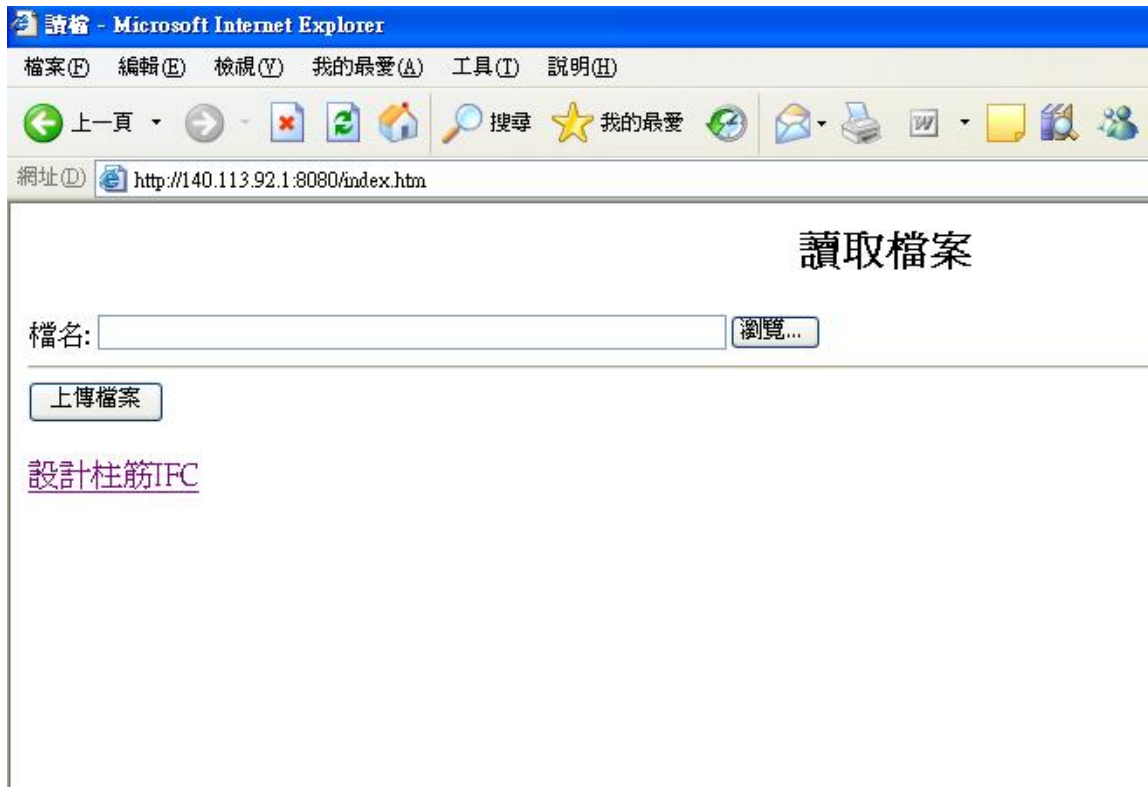


圖 5-1-2 系統實作首頁，可上傳所要檢核的 IFC 檔

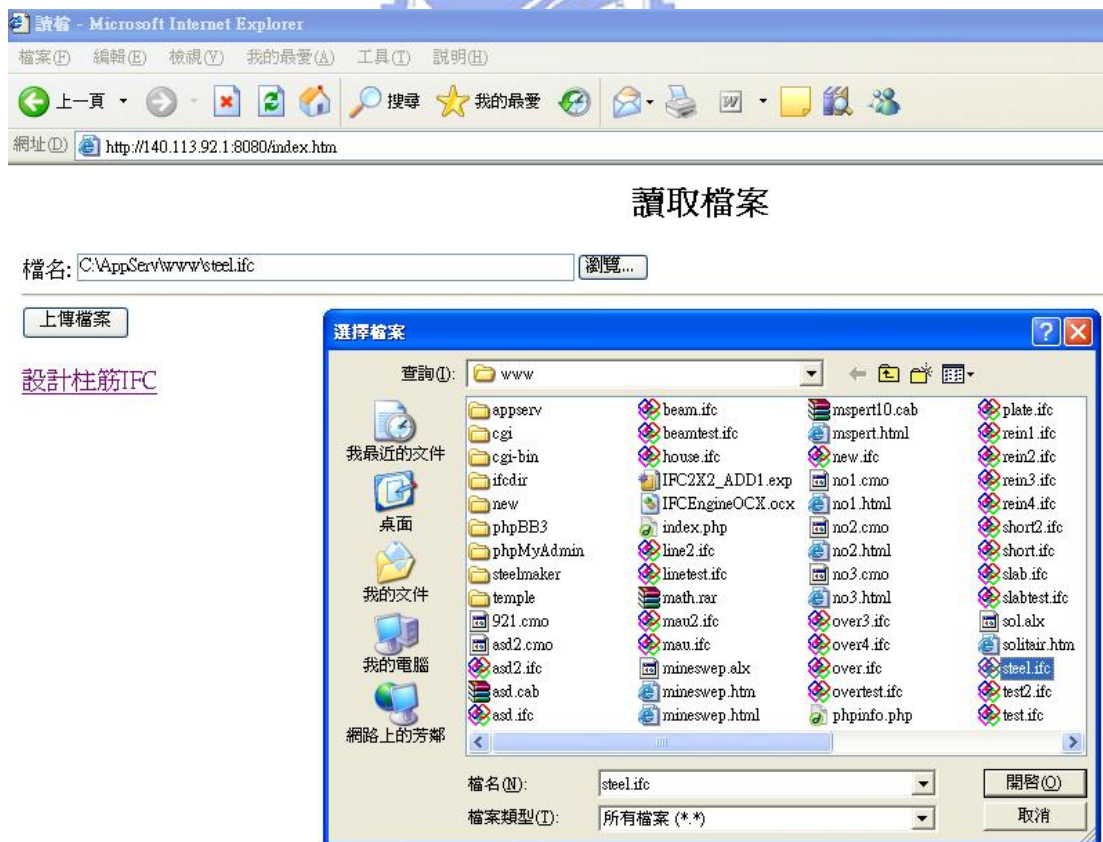


圖 5-1-3 點選瀏覽鈕可選擇要上傳的檔案

index - Windows Internet Explorer
 http://140.113.92.1/barmaker/rbar.html

您目前的安全性設定讓您的電腦暴露在風險中。按這裡變更安全性設定...

鋼筋稱號	D10	D13	D16	D19	D22	D25	D29	D32	D36
	(#3)	(#4)	(#5)	(#6)	(#7)	(#8)	(#9)	(#10)	(#11)
標稱直徑(mm)	9.53	12.7	15.9	19.1	22.2	25.4	28.7	32.2	35.8
重量(kgf/m)	0.56	0.994	1.56	2.25	3.04	3.98	5.08	6.39	7.90

柱資料

輸入柱斷面寬度(m):

輸入柱斷面長度(m):

輸入柱長度(m):

主筋資料

輸入主筋數(根):

輸入主筋鋼筋號數:

輸入主筋長度(m):

繫筋資料

輸入縱向繫筋數(根):

輸入橫向繫筋數(根):

輸入繫筋鋼筋號數:

輸入橫向鋼筋間距(m):

輸入彎鉤延伸長度(m):

輸入保護層厚度(m):

圖 5-1-4 矩形柱與柱筋設計的表格網頁(rbar.html)

http://localhost/STEELMAKER/test.cgi - Microsoft Internet Explorer

檔案(F) 編輯(E) 檢視(V) 我的最愛(A) 工具(T) 說明(H)

← 上一頁 → 搜尋 ★ 我的最愛

網址(D) http://localhost/STEELMAKER/test.cgi

IFC產生成功
[下載檔案](#)

圖 5-1-5 使用者設計的柱筋模型可點選下載

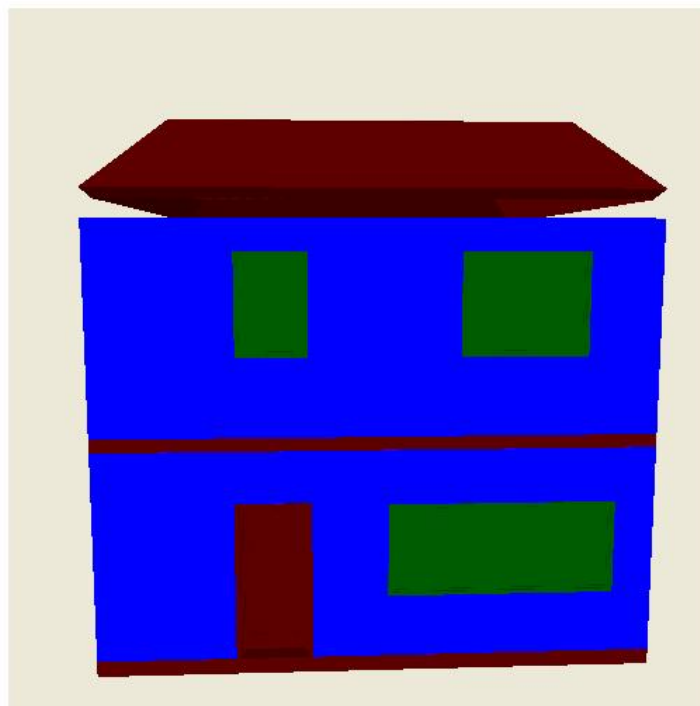
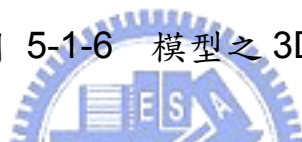
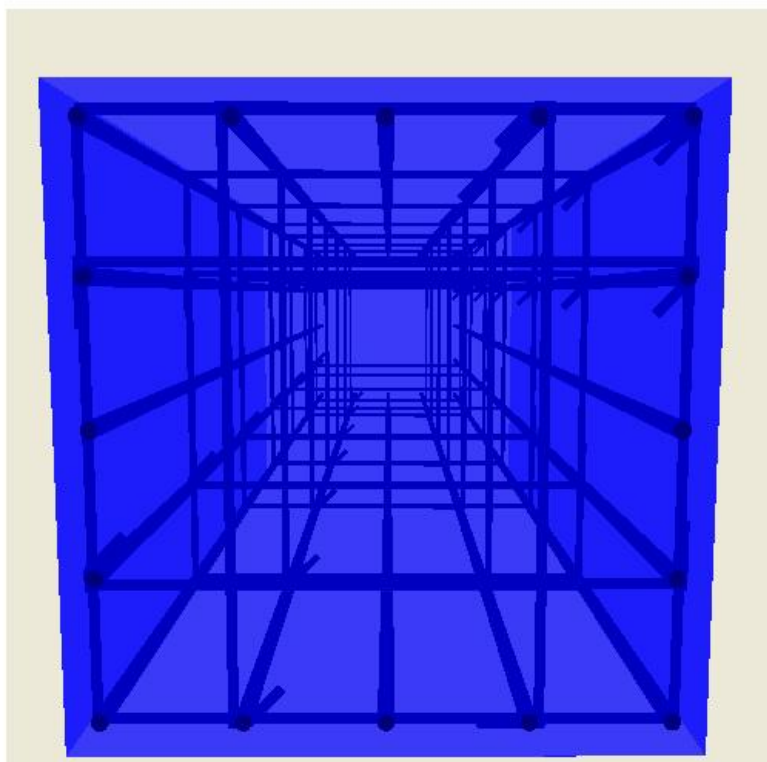


圖 5-1-6 模型之 3D 展示



Properties:



- 檢核樑柱不合理 : 規範
- 檢核樓板 規範
- 檢核柱筋鋼筋比 規範
- 檢核相鄰箍筋最大間距 規範
- 檢核橫向鋼筋間距 規範
- 檢核彎鉤長度 規範
- 檢核保護層厚度 規範

ID

圖 5-1-7 柱半透明化來觀察柱內部的鋼筋分布

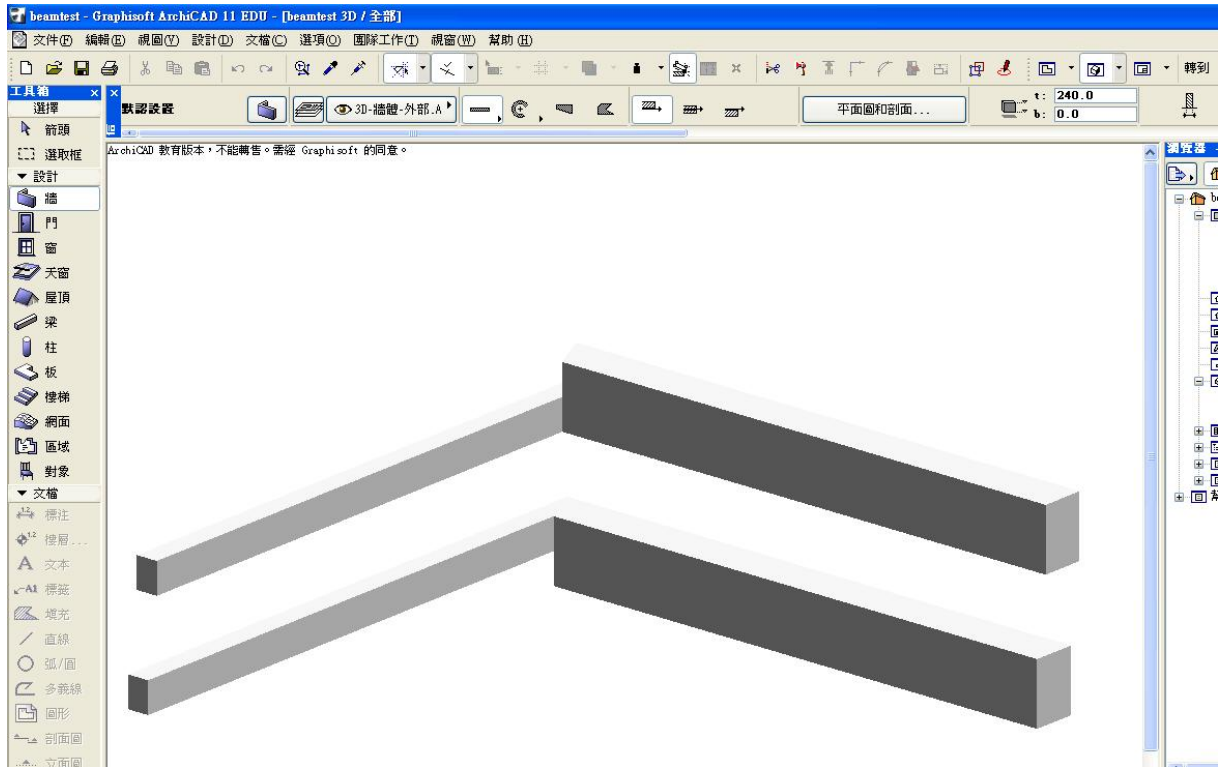


圖 5-3-1 以 ArchiCAD 繪製大梁跨小梁模型

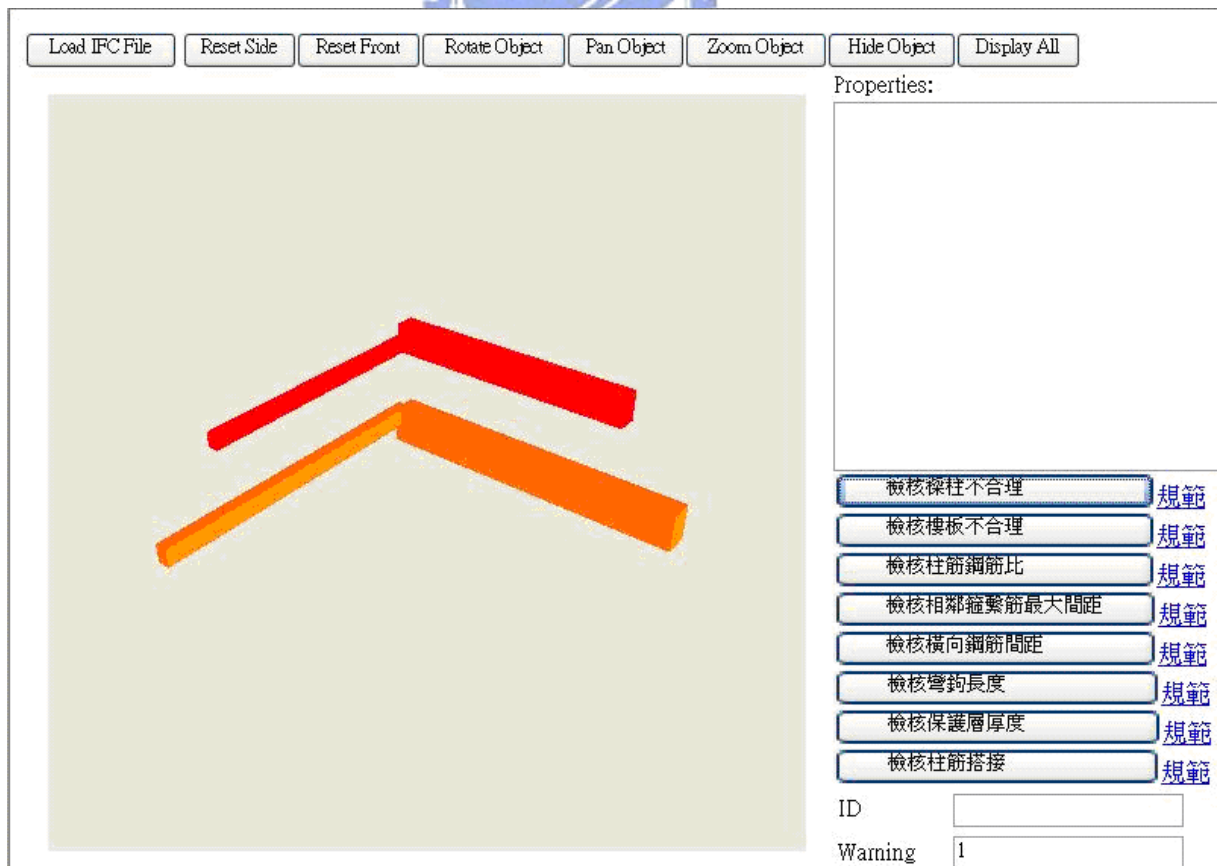


圖 5-3-2 以紅色來表示錯誤的地方，並在右方顯示警告錯誤的數目

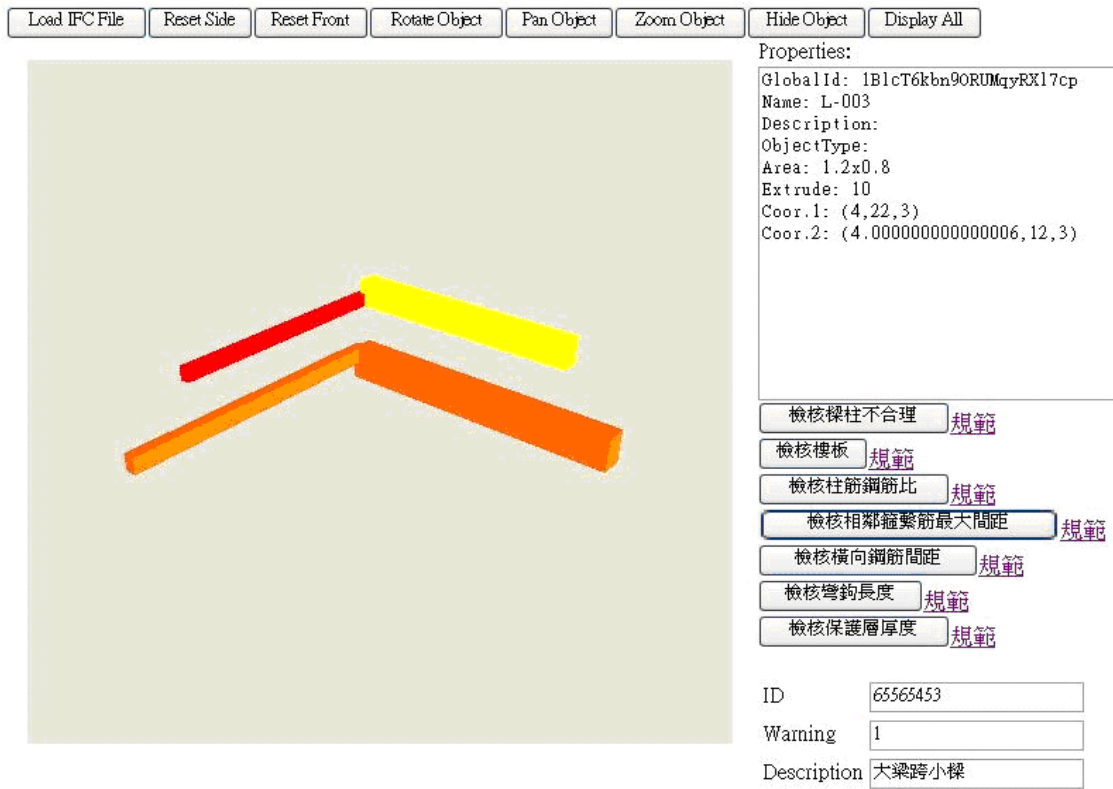


圖 5-3-3 檢核出大梁跨小樑，點選後錯誤物件後，在右方顯示對錯誤物件的描述。

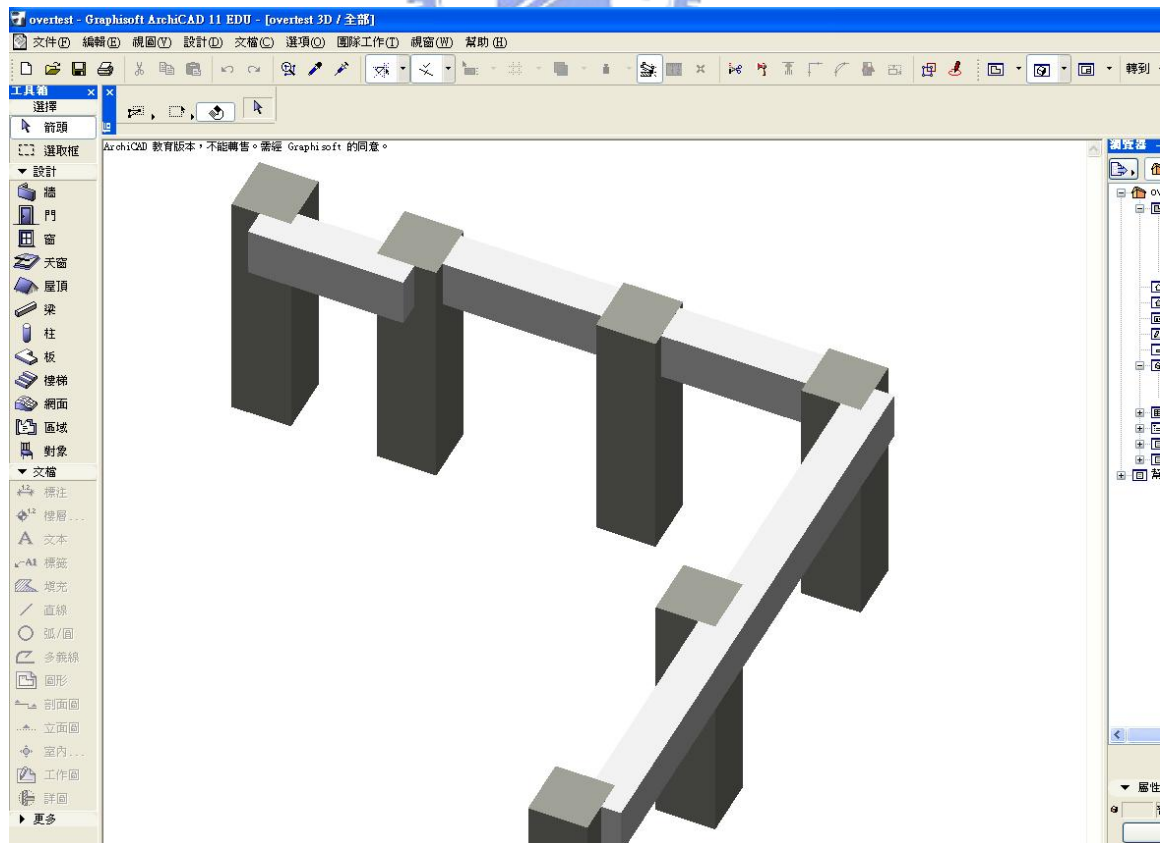


圖 5-3-4 以 ArchiCAD 繪製梁身超過柱模型

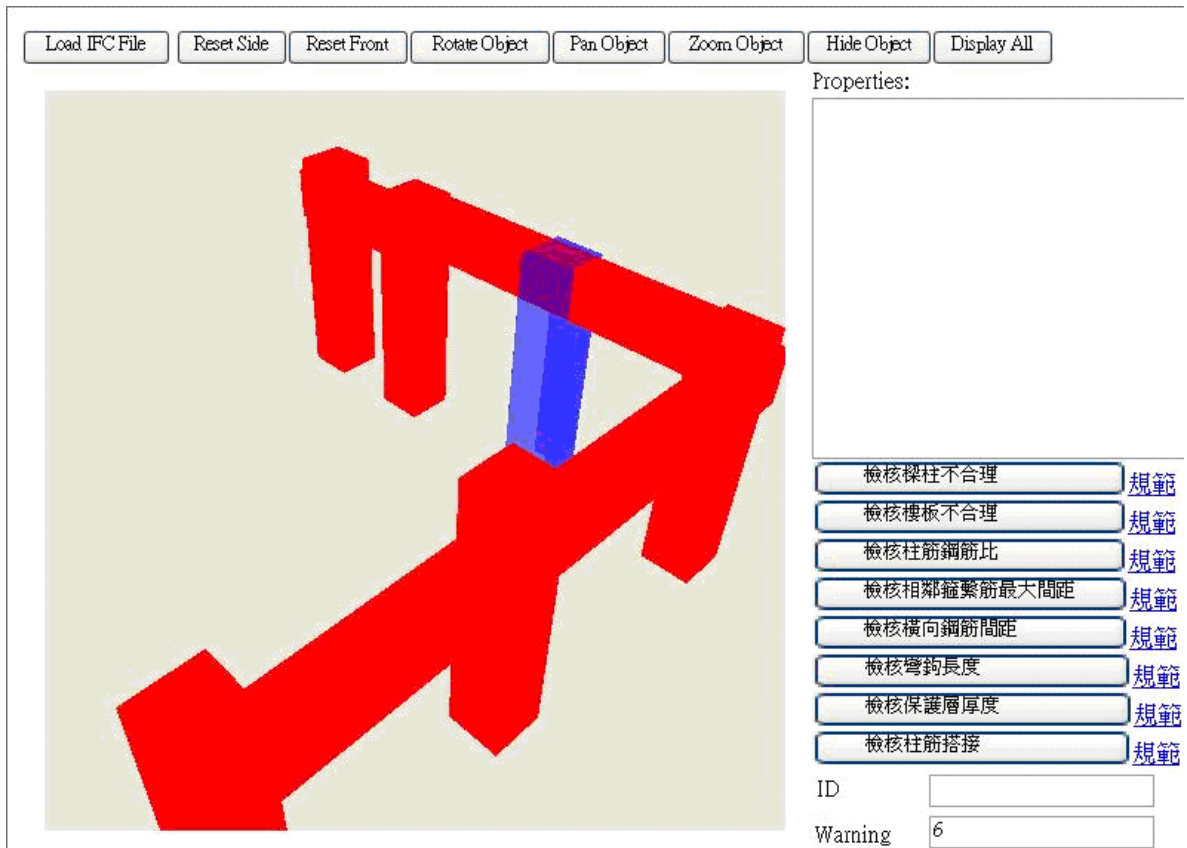


圖 5-3-5 以紅色來表示錯誤的地方，並在右方顯示警告錯誤的數目

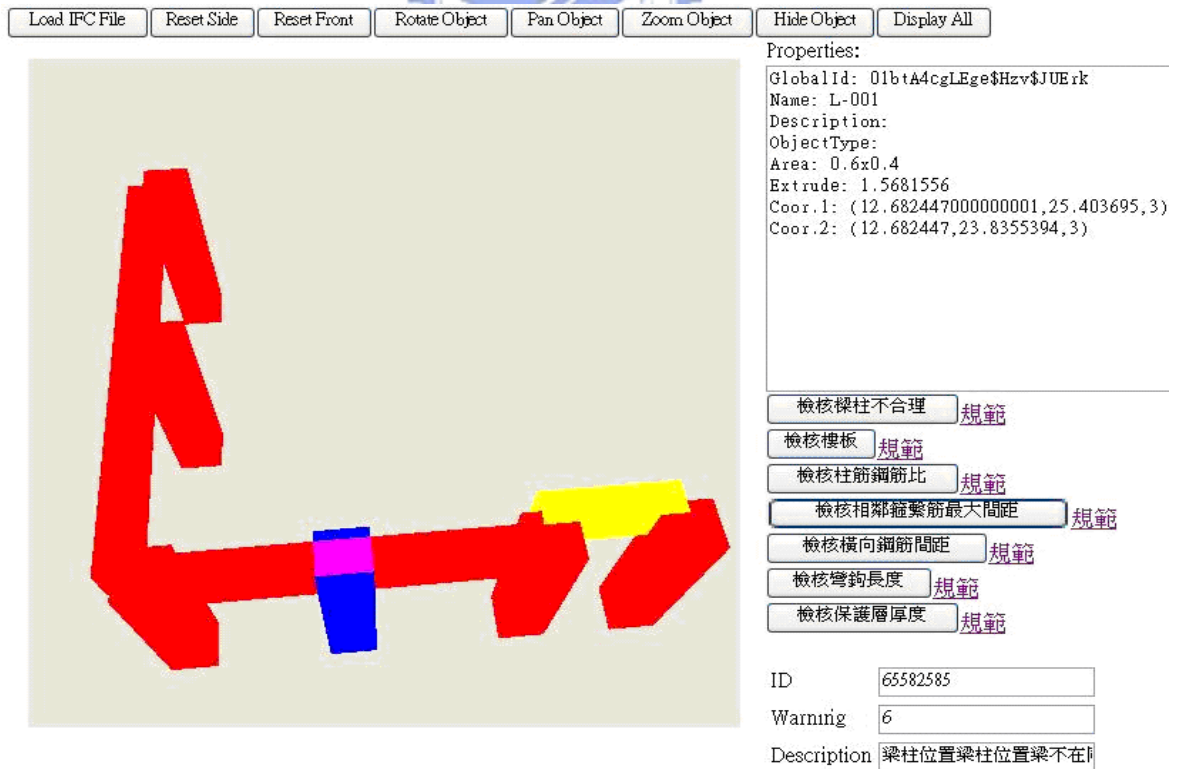


圖 5-3-6 檢核出梁身超過柱，點選後錯誤物件後，在右方顯示對錯誤物件的描述。

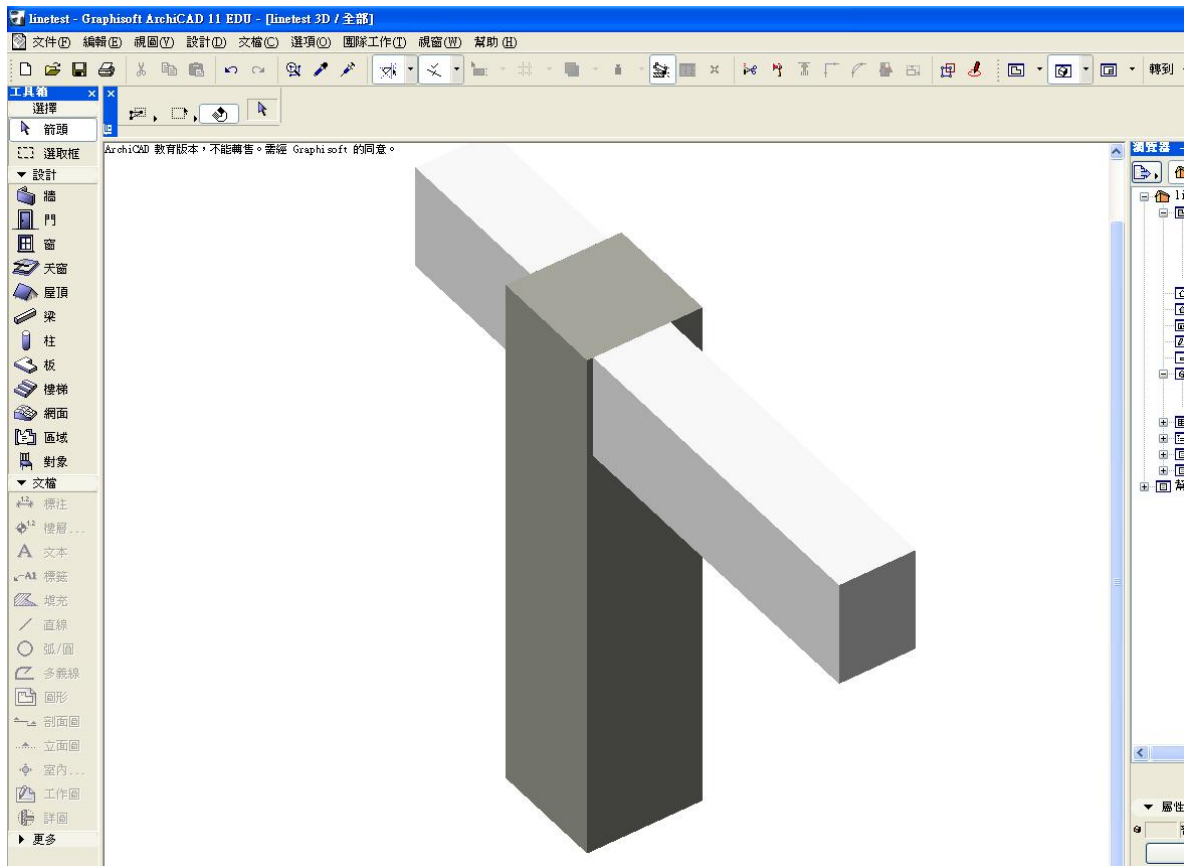


圖 5-3-7 以 ArchiCAD 繪製兩根梁跨柱不在同一直線上的模型

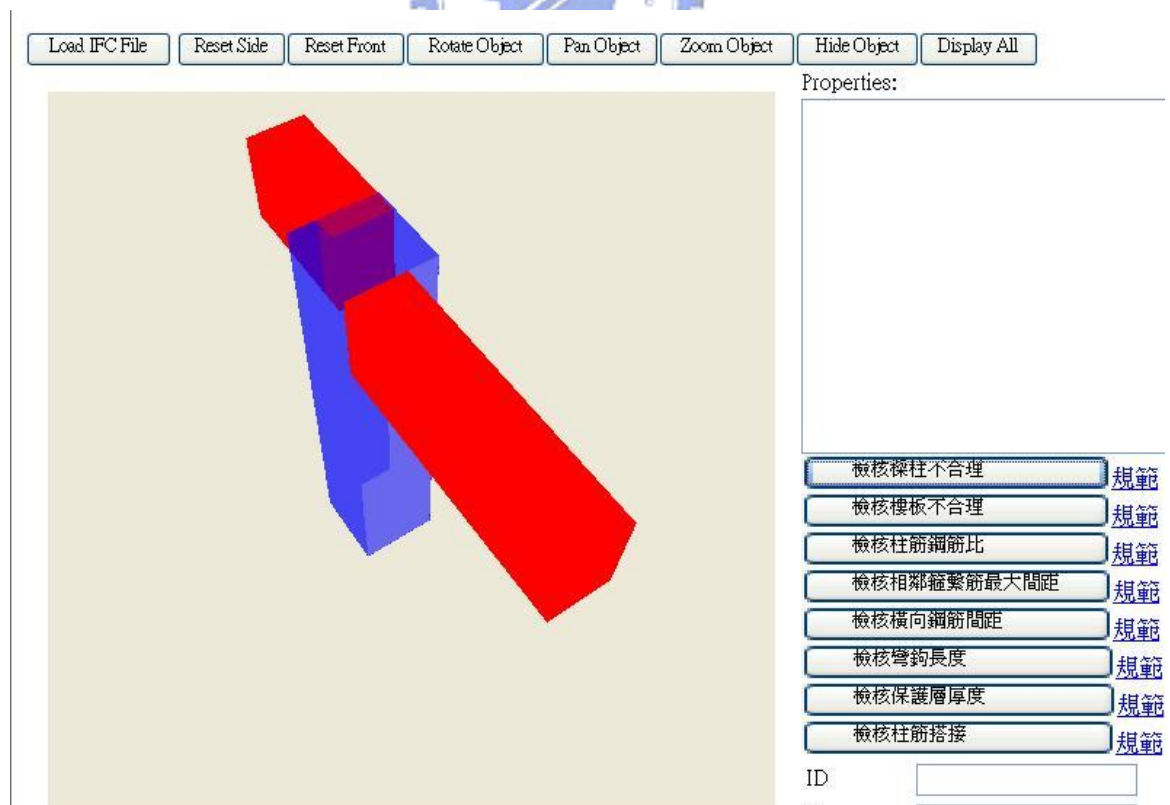


圖 5-3-8 以紅色來表示錯誤的地方

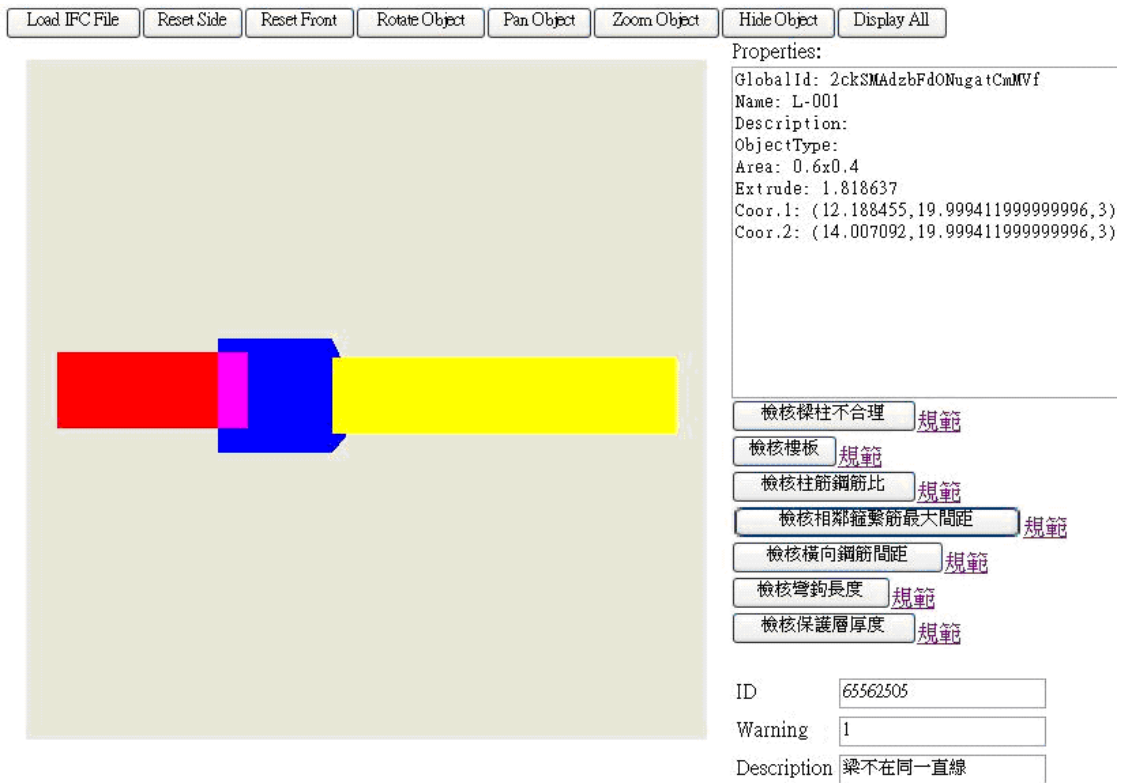


圖 5-3-9 檢核出左邊梁與右邊梁不在同一直線上，點選錯誤物件後，在右方顯示對錯誤物件的描述。

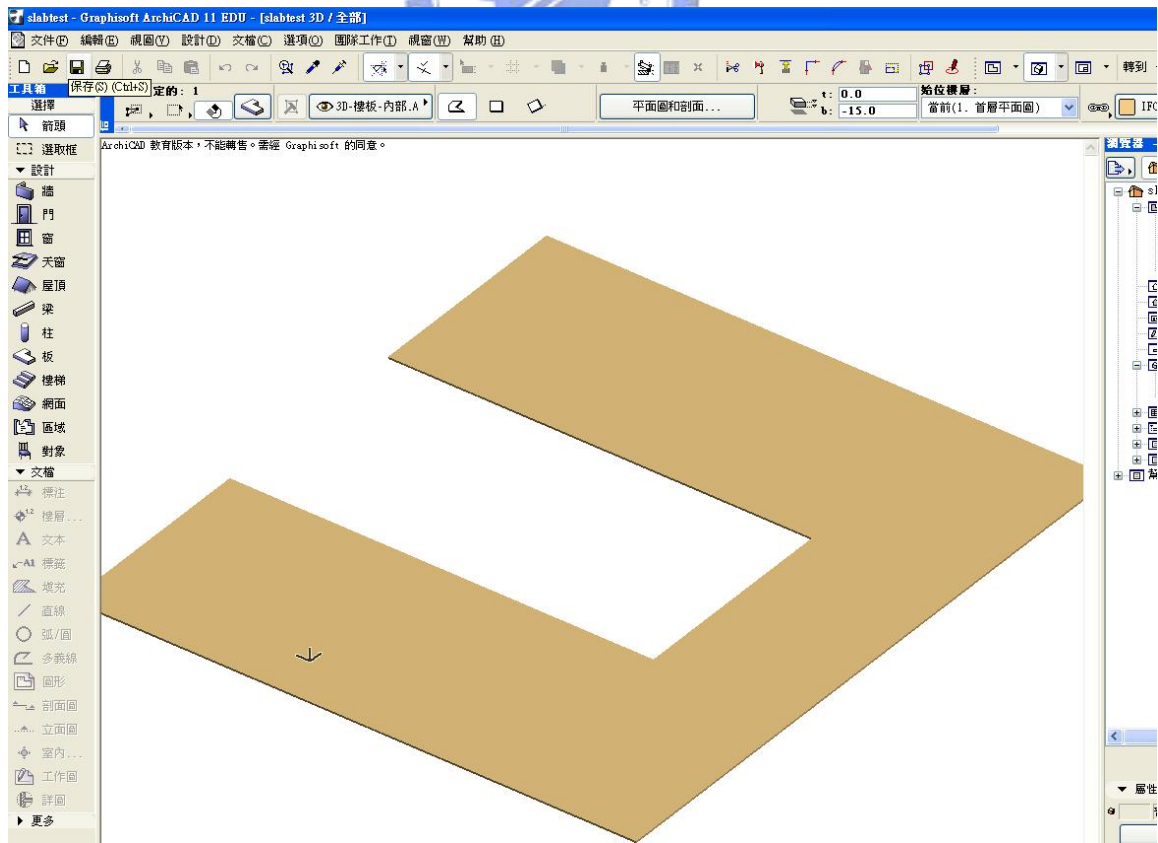


圖 5-3-10 以 ArchiCAD 繪製樓層樓板中空模型

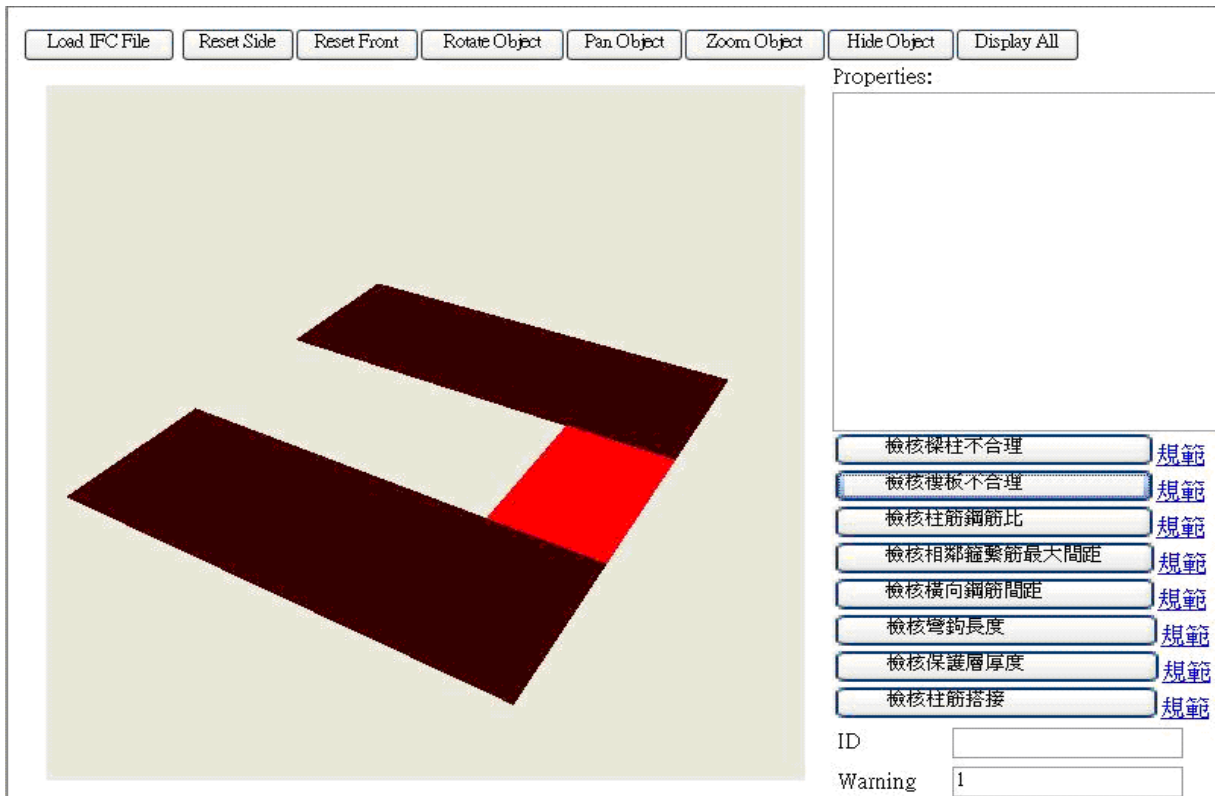


圖 5-3-11 以紅色來表示錯誤的地方，並在右方顯示警告錯誤的數目

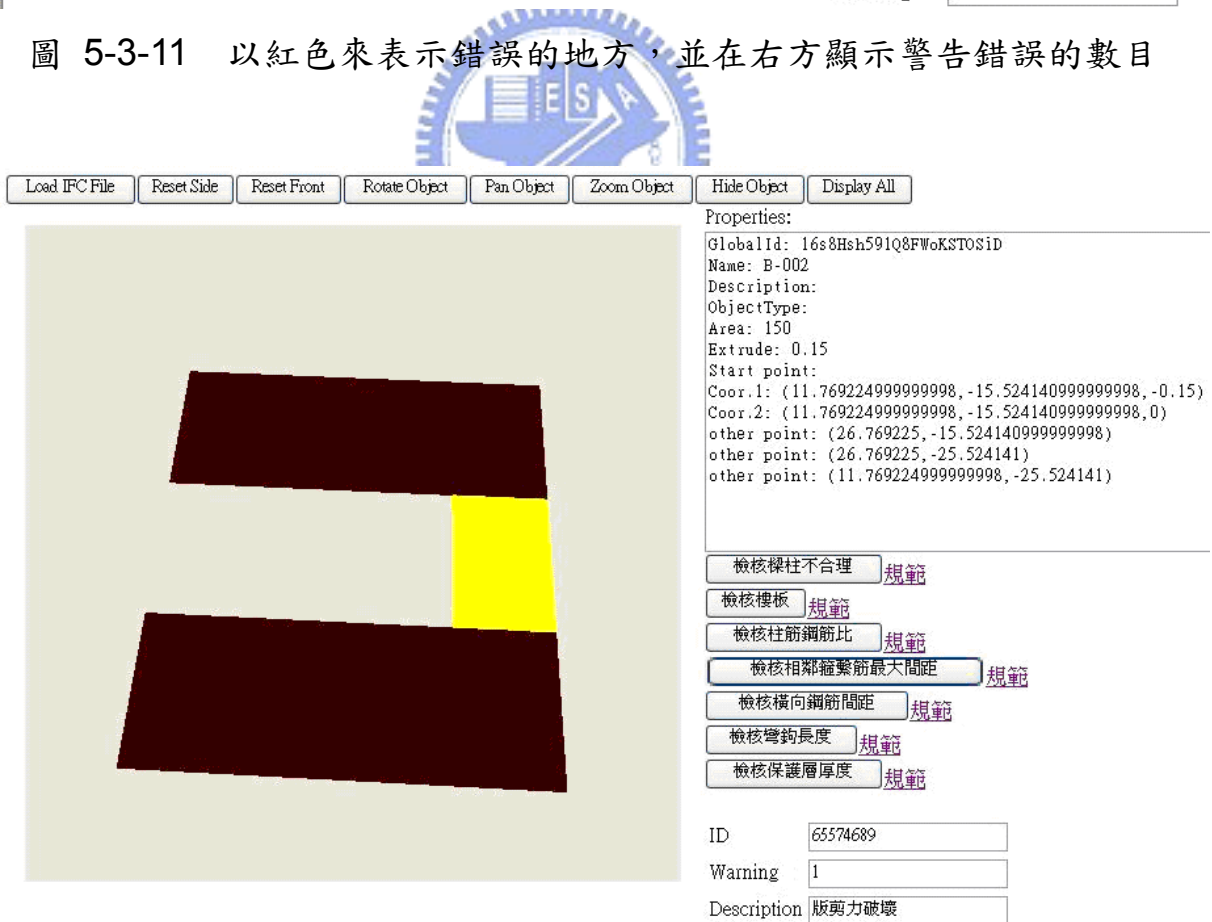


圖 5-3-12 檢核出樓板可能會遭受剪力破壞，點選後錯誤物件後，以紅色來表示錯誤的地方，並在右方顯示對錯誤物件的描述。

index - Windows Internet Explorer
 http://140.113.92.1/barnsker/rbar.html

檔案(E) 編輯(E) 檢視(V) 我的最愛(A) 工具(T) 說明(H)

index

您目前的安全性設定讓您的電腦暴露在風險中。按這裡變更安全性設定...

柱資料

輸入柱斷面寬度(m):

輸入柱斷面長度(m):

輸入柱長度(m):

主筋資料

輸入主筋數(根):

輸入主筋鋼筋號數:

輸入主筋長度(m):

繫筋資料

輸入縱向繫筋數(根):

輸入橫向繫筋數(根):

輸入繫筋鋼筋號數:

輸入橫向鋼筋間距(m):

輸入彎鉤延伸長度(m):

輸入保護層厚度(m):

鋼筋稱號	D10	D13	D16	D19	D22	D25	D29	D32	D36
	(#3)	(#4)	(#5)	(#6)	(#7)	(#8)	(#9)	(#10)	(#11)
標稱直徑(mm)	9.53	12.7	15.9	19.1	22.2	25.4	28.7	32.2	35.8
重量(kgf/m)	0.56	0.994	1.56	2.25	3.04	3.98	5.08	6.39	7.90

完成 國際網路

圖 5-4-1 柱與柱筋的設計參數表單

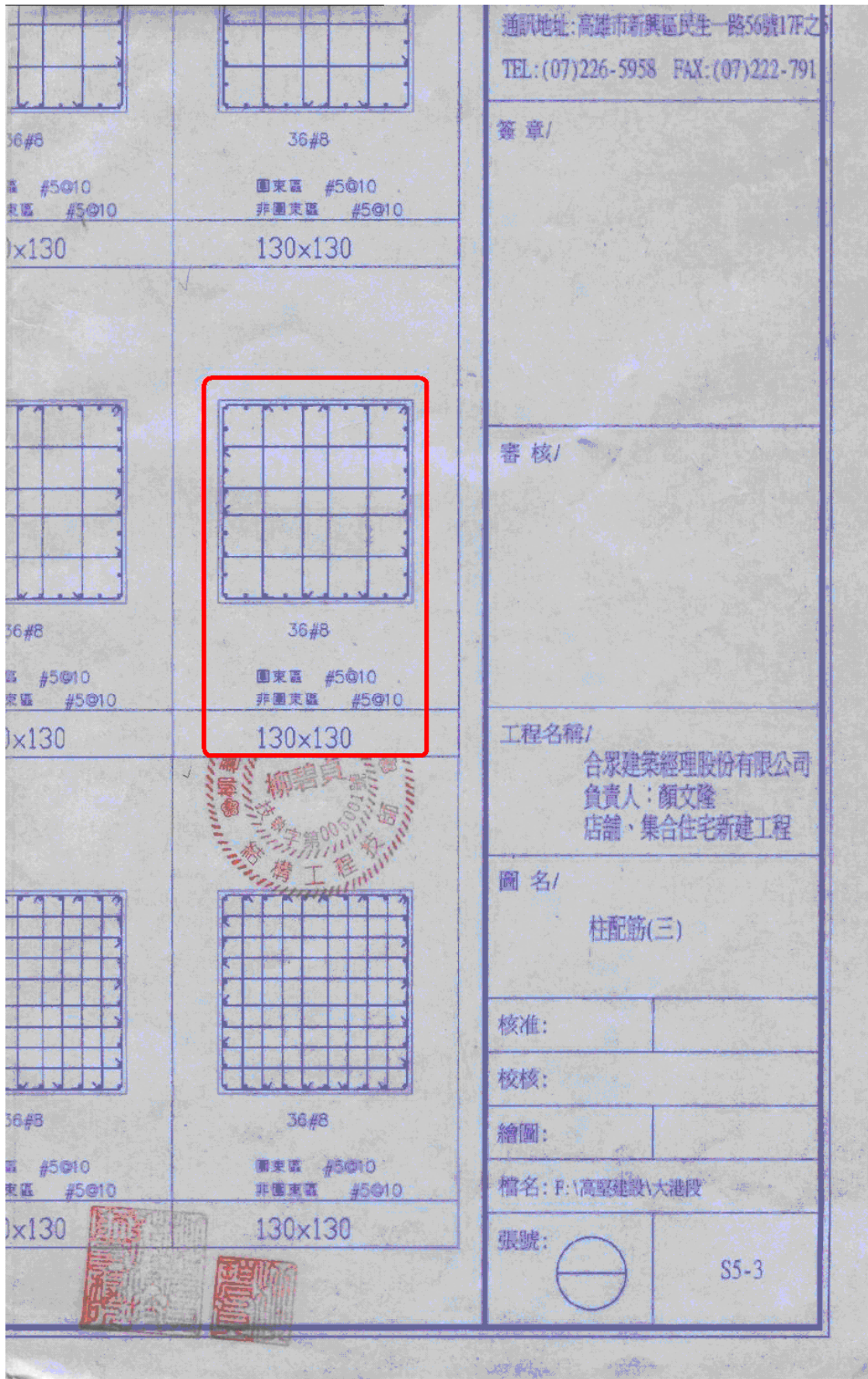


圖 5-4-2 根據實際柱配筋圖，來設計一柱筋模型，如紅色框選的斷面

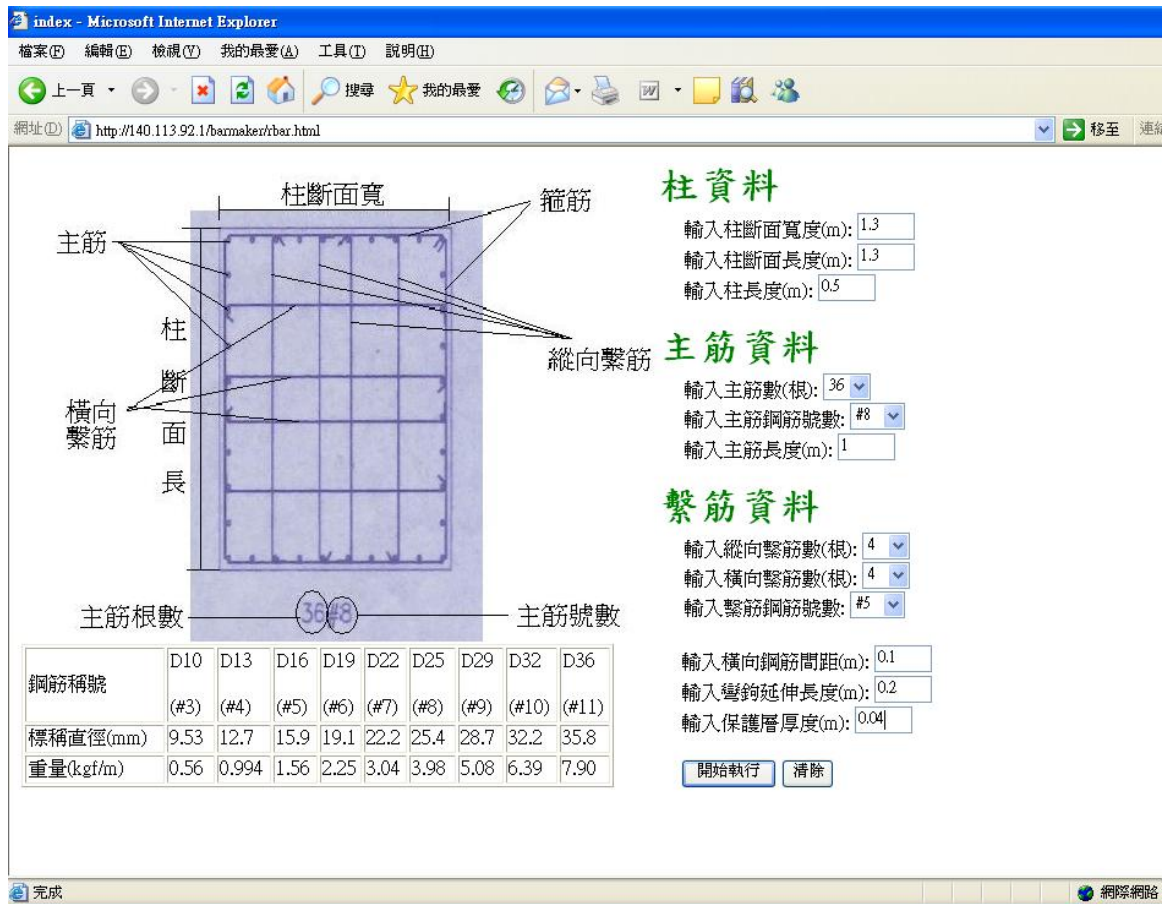


圖 5-4-3 選定柱配筋圖的某一斷面，填入指定參數，來設計柱筋模型

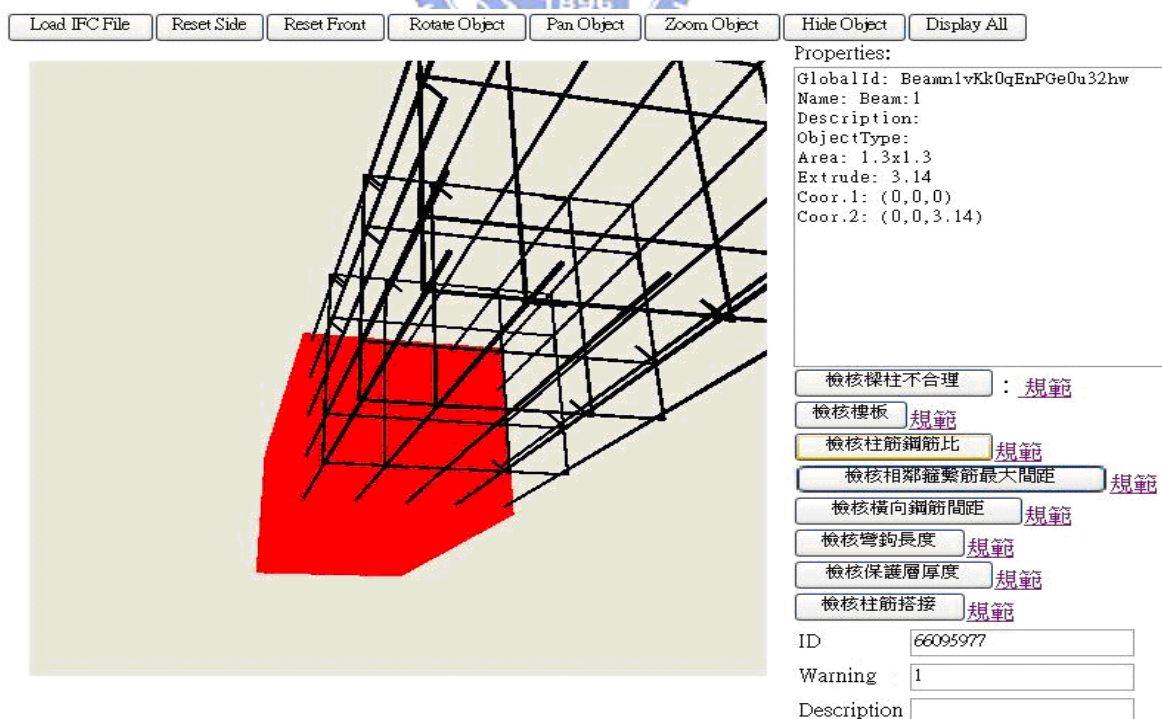


圖 5-4-4 檢核出鋼筋比不符，以紅色來表示錯誤的地方，並在右方顯示警告錯誤的數目。

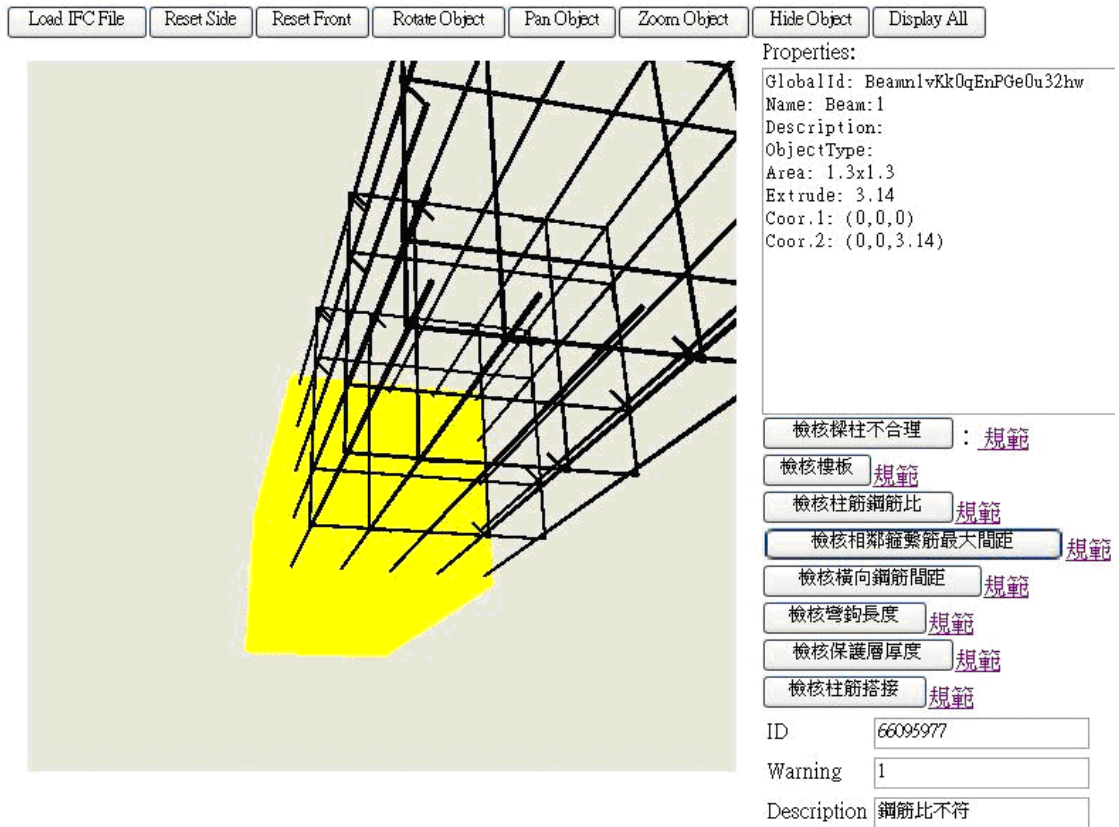


圖 5-4-5 點選紅色的錯誤物件後，右下會有對錯誤物件的描述

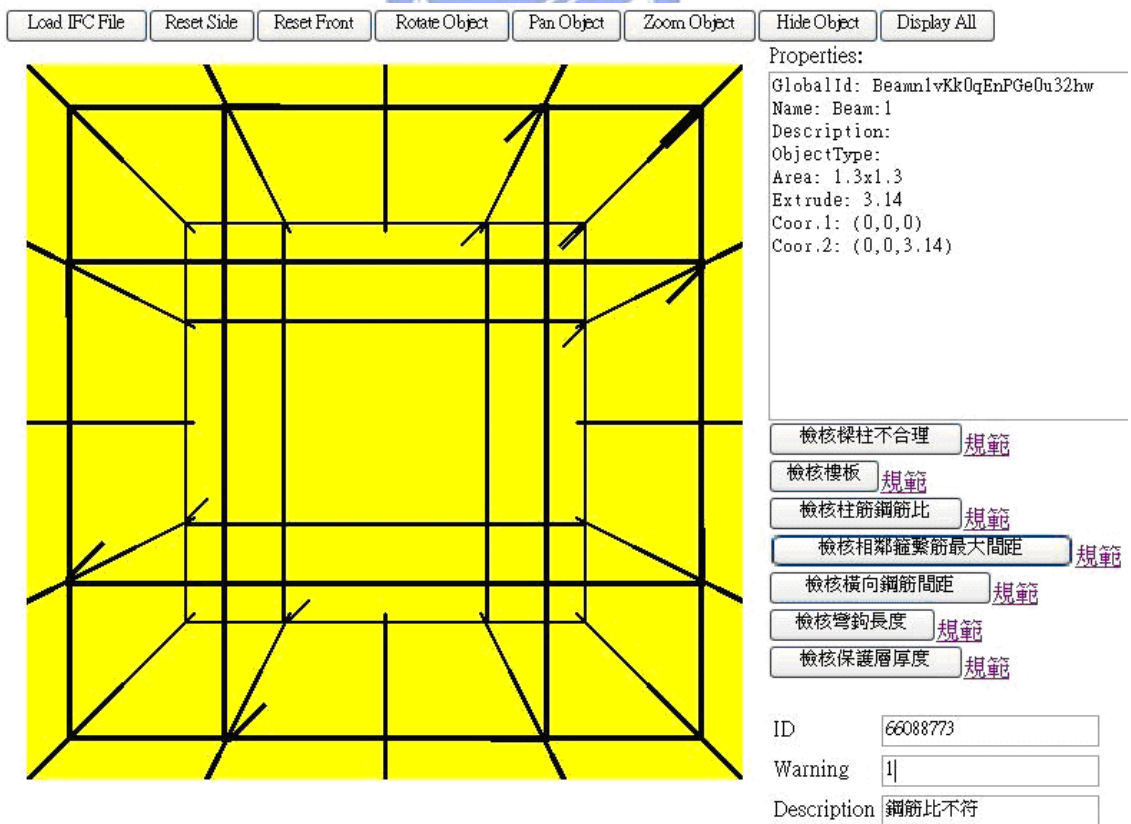


圖 5-4-6 將視窗深入柱之內部查看鋼筋分布

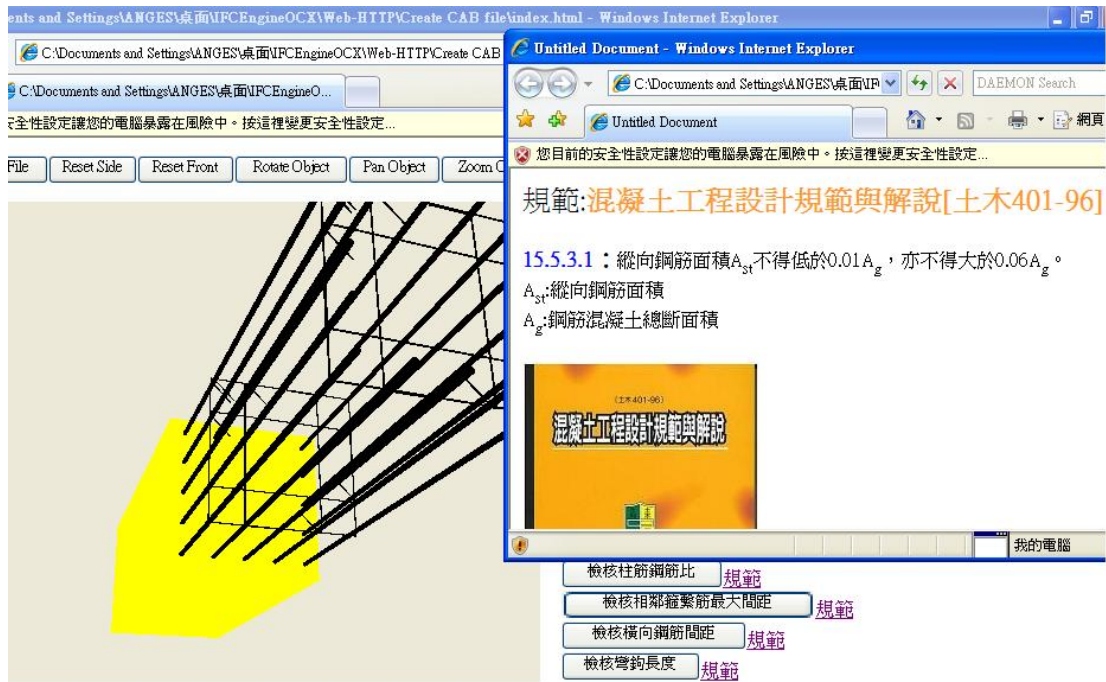


圖 5-4-7 跳出新視窗，顯示鋼筋比之規範

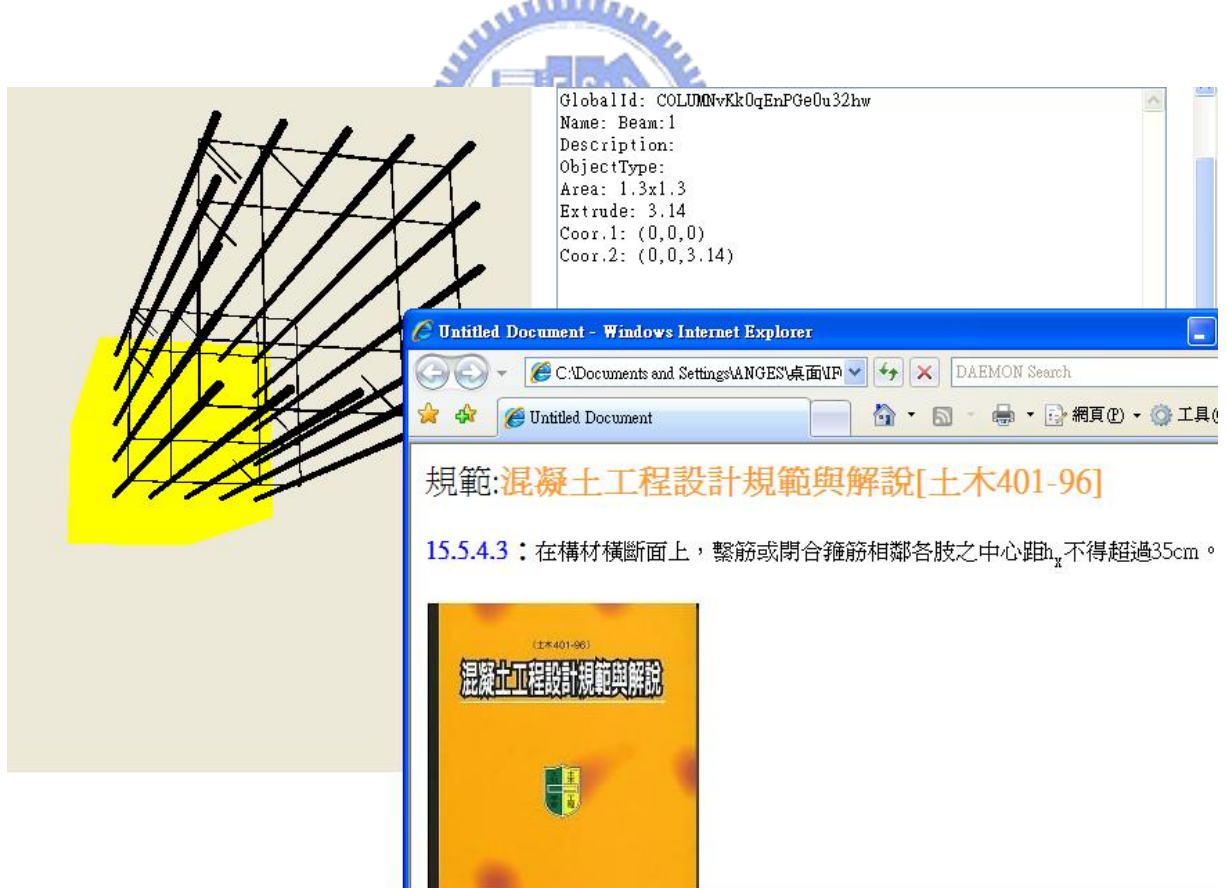


圖 5-4-8 跳出新視窗，顯示構材橫斷面上，繫筋與閉合箍筋相鄰各肢中心距之規範

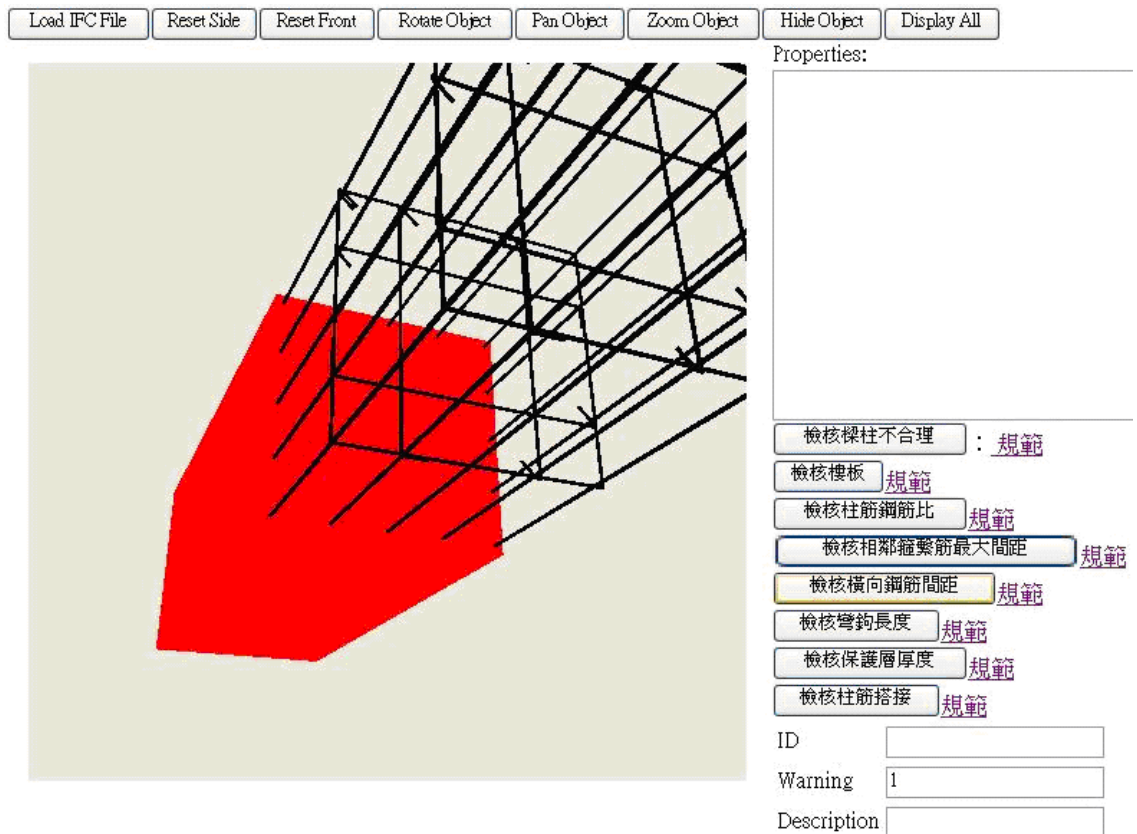


圖 5-4-9 檢核出橫向鋼筋間距不符，以紅色來表示錯誤的地方，並在右方顯示警告錯誤的數目。

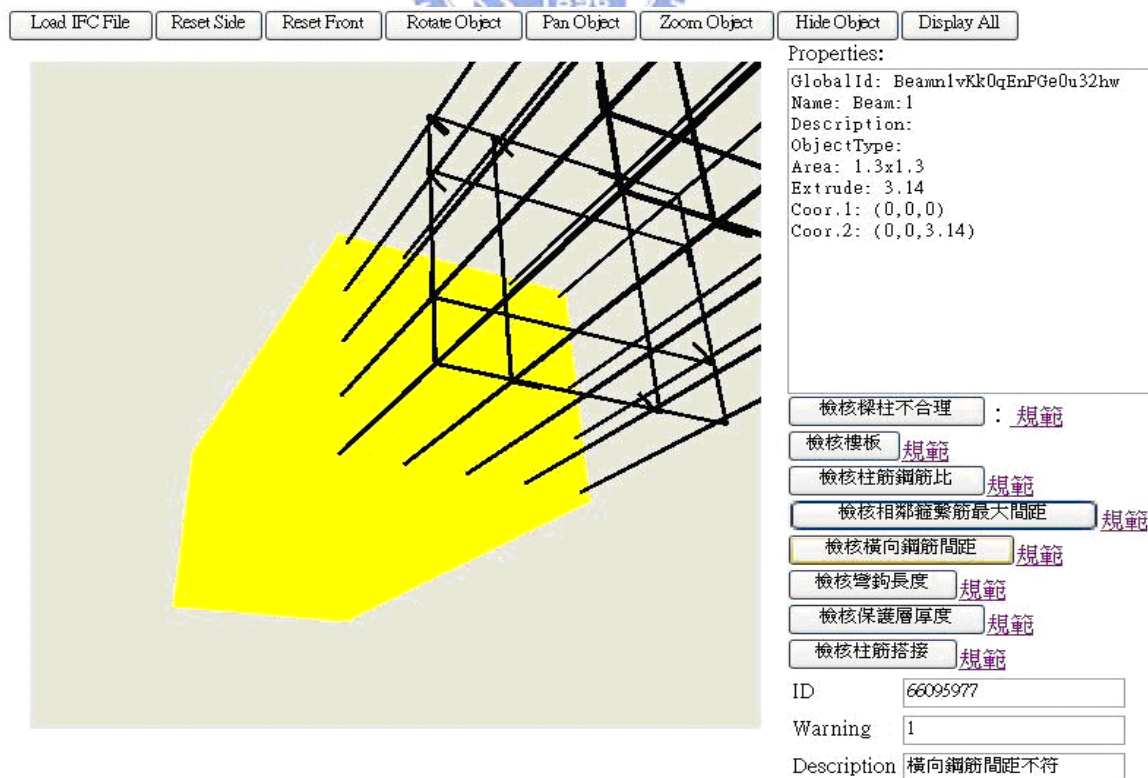


圖 5-4-10 點選紅色的錯誤物件後，右下會有對錯誤物件的描述

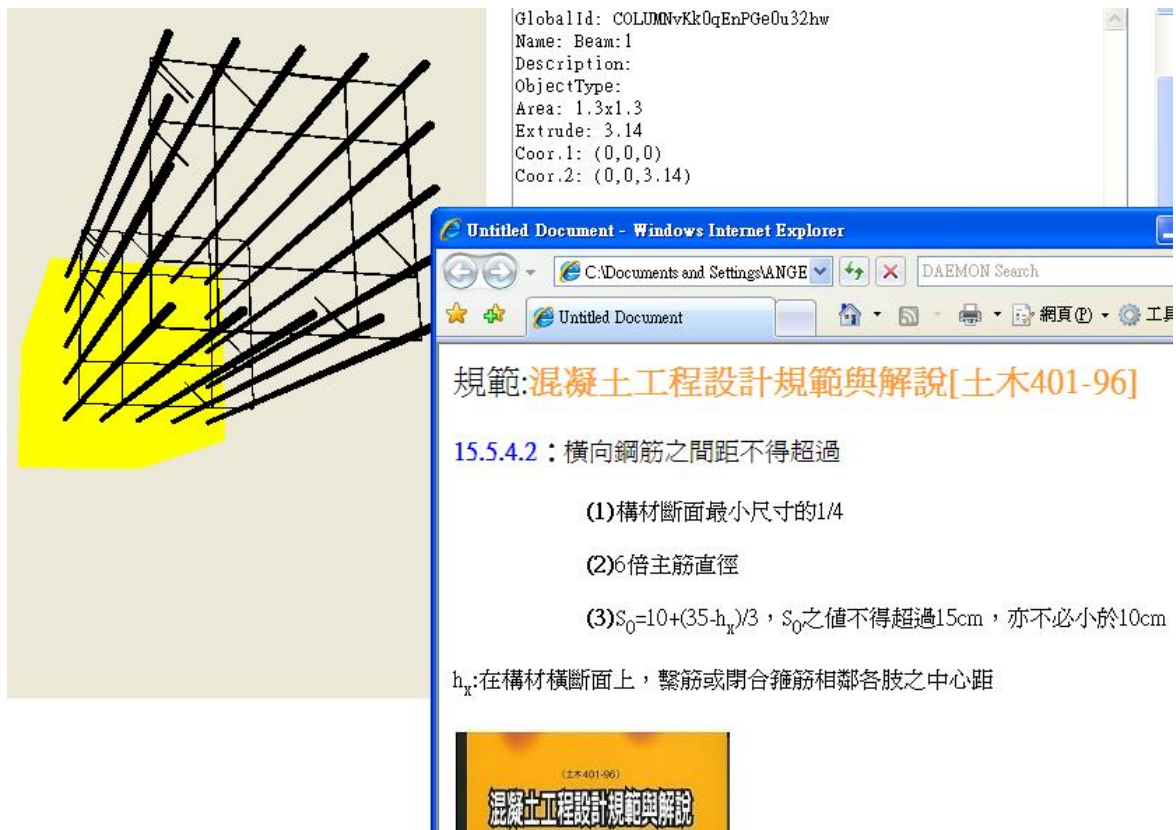


圖 5-4-11 跳出新視窗，顯示橫向鋼筋間距之規範

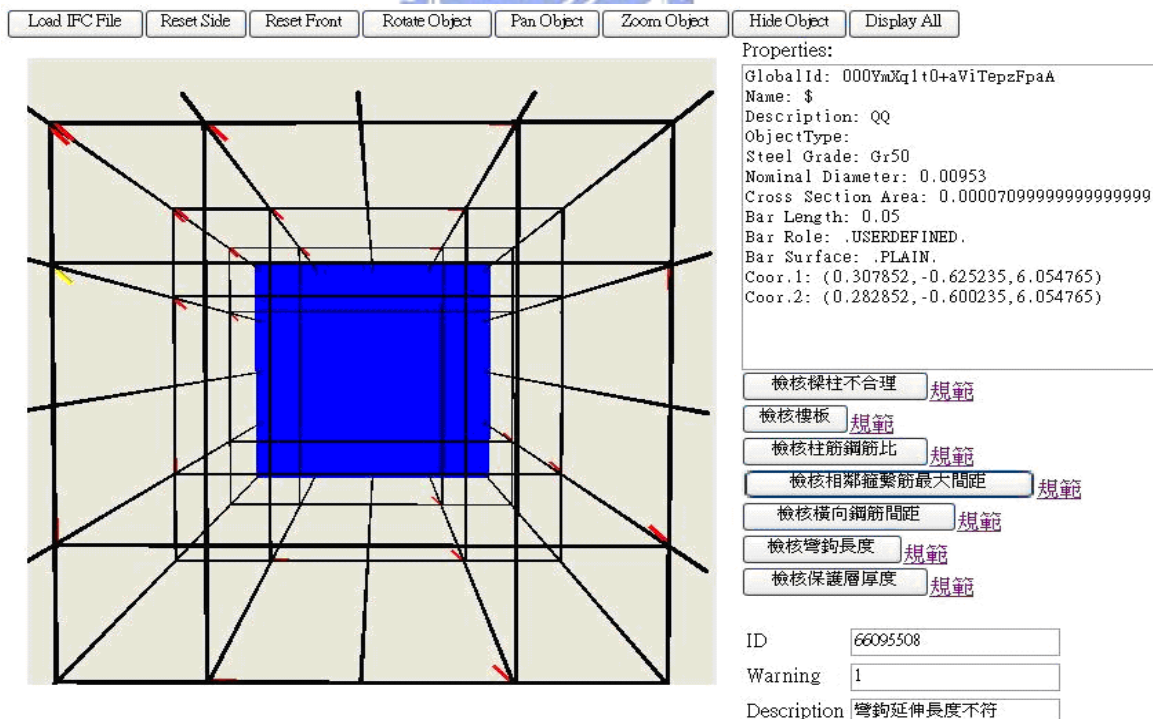


圖 5-4-12 檢核出彎鉤延伸長度不符，以紅色來表示錯誤的地方，發現所有彎鉤都不符規範，在右方有顯示警告錯誤的數目，點選紅色的錯誤物件後，右下會有對錯誤物件的描述。

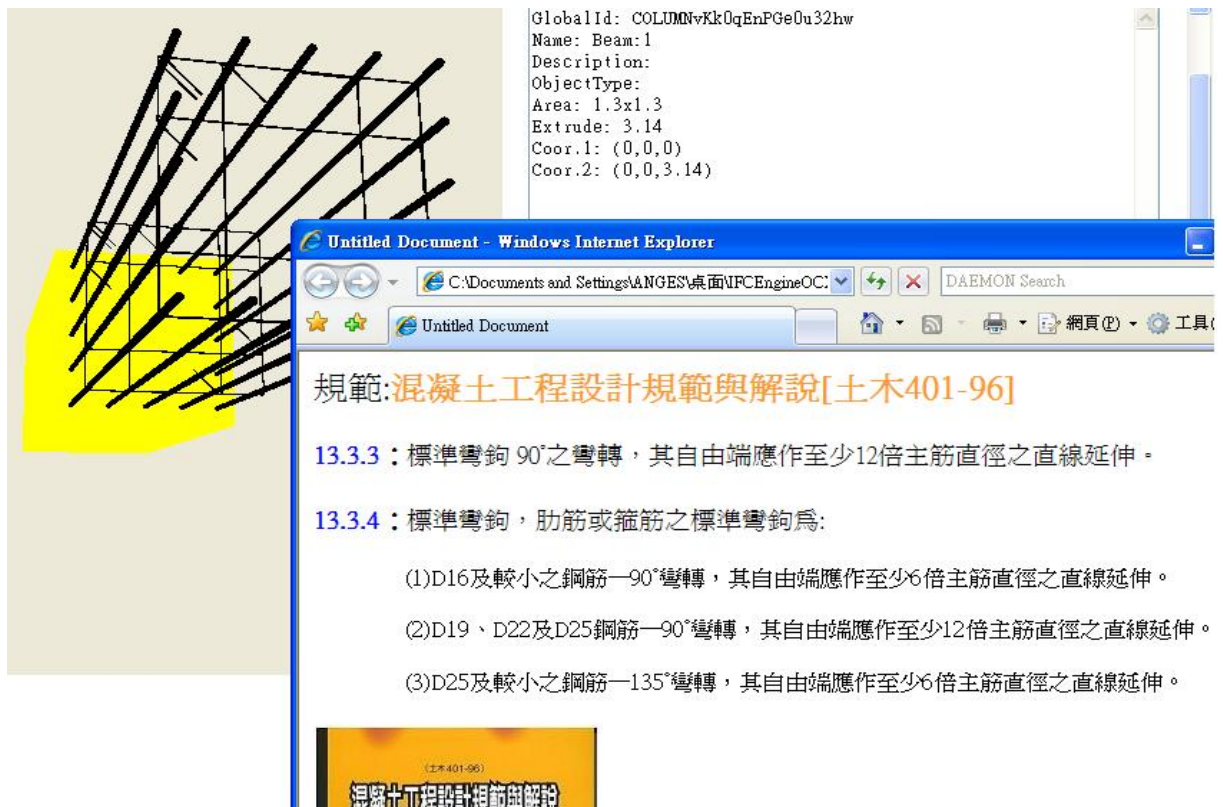


圖 5-4-13 跳出新視窗，顯示標準彎鉤之規範

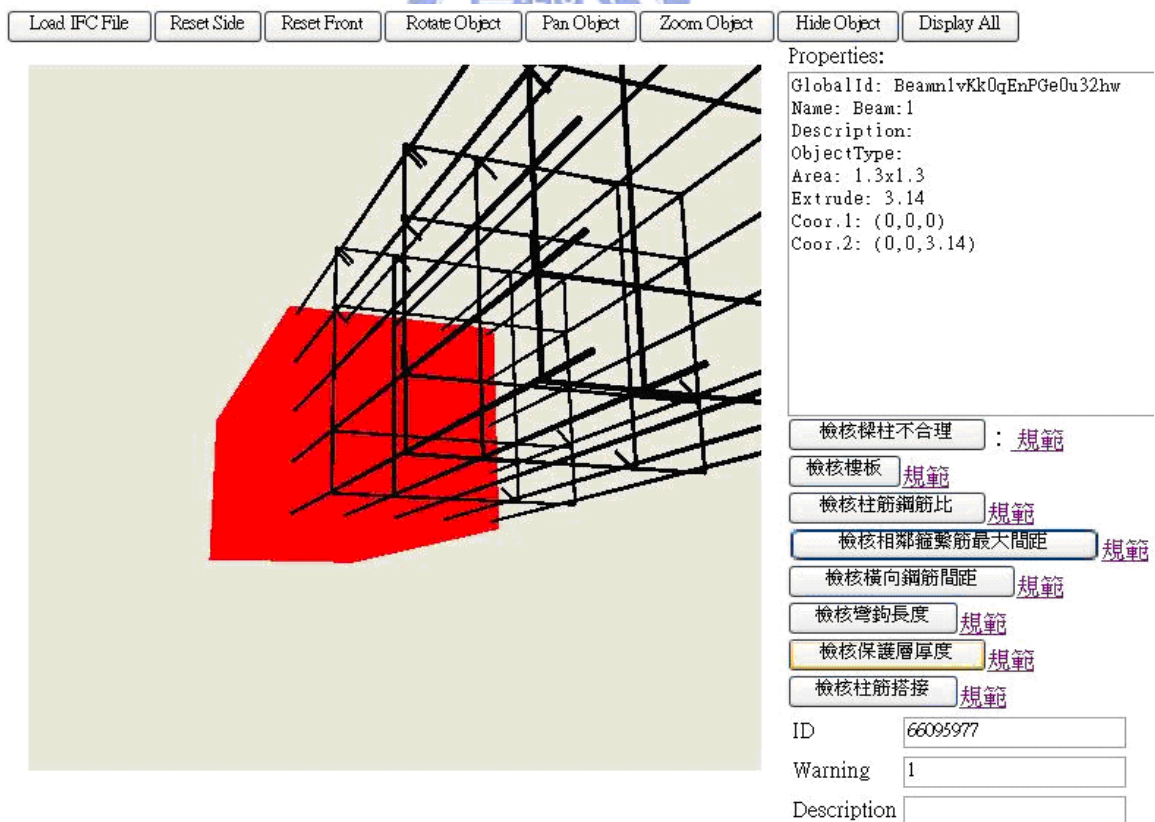


圖 5-4-14 檢核出保護層不足，以紅色來表示錯誤的地方，並在右方顯示警告錯誤的數目。

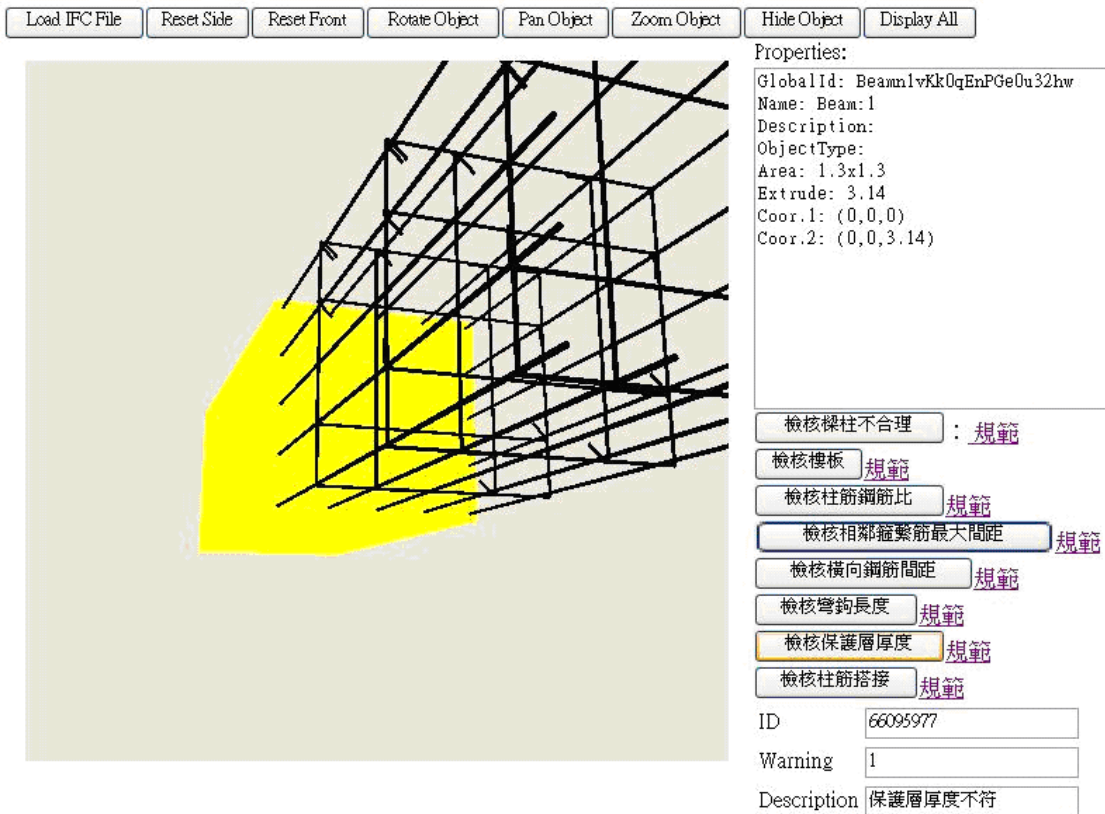


圖 5-4-15 點選紅色的錯誤物件後，右下會有對錯誤物件的描述。

規範: **混凝土工程設計規範與解說[土木401-96]**

15.6.1：鋼筋之保護層，如下:(單位:mm)

狀況	版、牆、欄及牆版	梁、柱及基腳	薄殼及摺版
不受風雨侵襲且不與土壤接觸者：			
鋼線或 $d_b \leq 16\text{mm}$ 鋼筋	20	40	15
$16\text{mm} < d_b \leq 36\text{mm}$ 鋼筋	20	40	20
$d_b > 36\text{mm}$ 鋼筋	40	40	20
受風雨侵襲或與土壤接觸者：			
鋼線或 $d_b \leq 16\text{mm}$ 鋼筋	40	40	40
$d_b < 16\text{mm}$ 鋼筋	50	50	50
澆置於土壤或岩石上或經常與水及土壤接觸者：	75	75	
與海水或腐蝕性環境接觸者：	100	100	

d_b 為鋼筋或鋼線之標稱直徑

混凝土工程設計規範與解說

圖 5-4-16 跳出新視窗，鋼筋保護層之規範

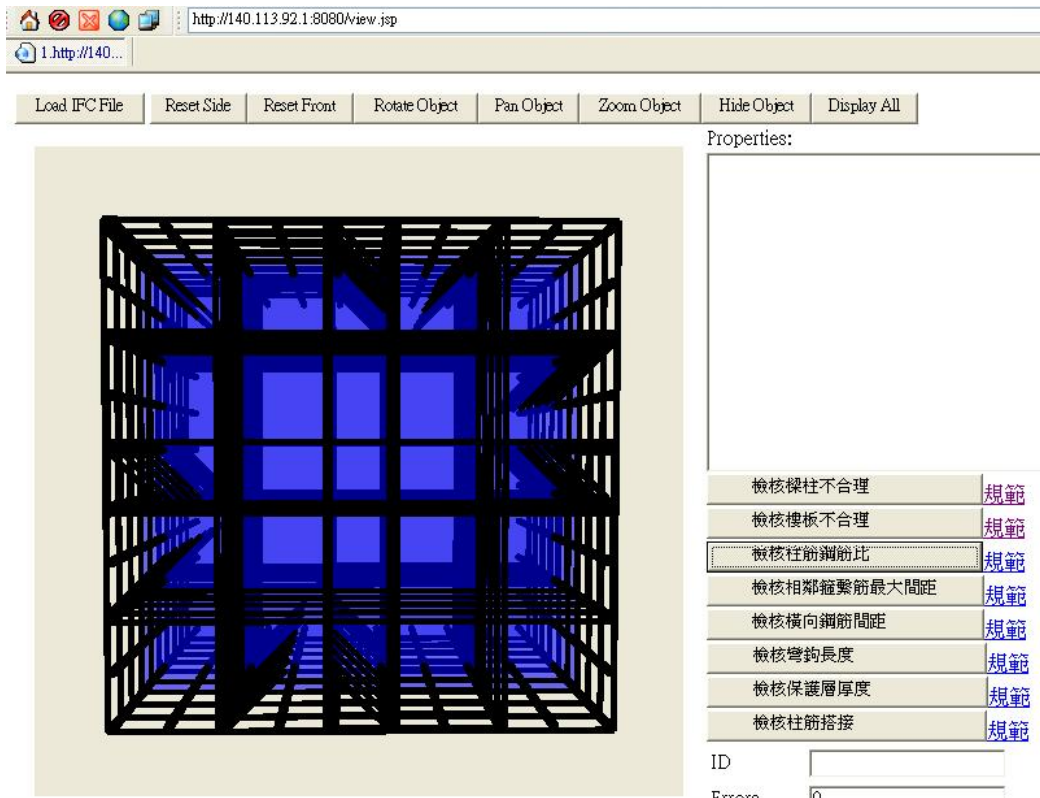


圖 5-4-17 檢核柱筋鋼筋比，沒有錯誤

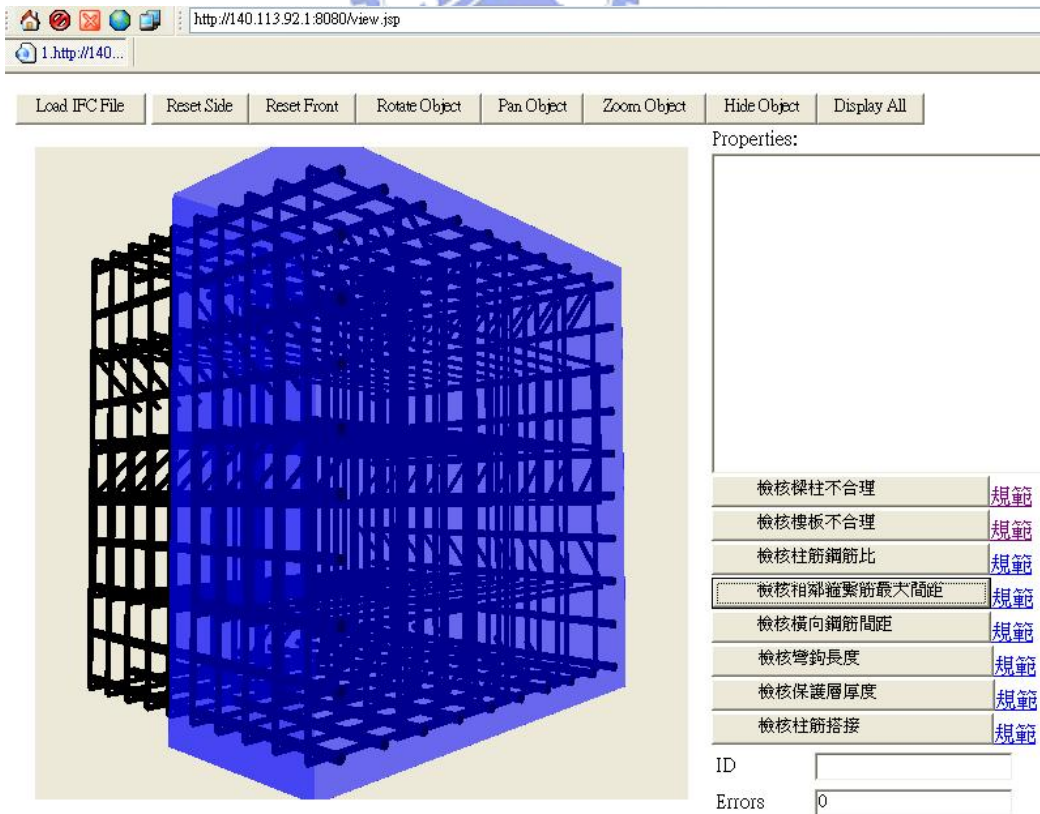


圖 5-4-18 檢核相鄰箍繫筋的最大間距，沒有錯誤

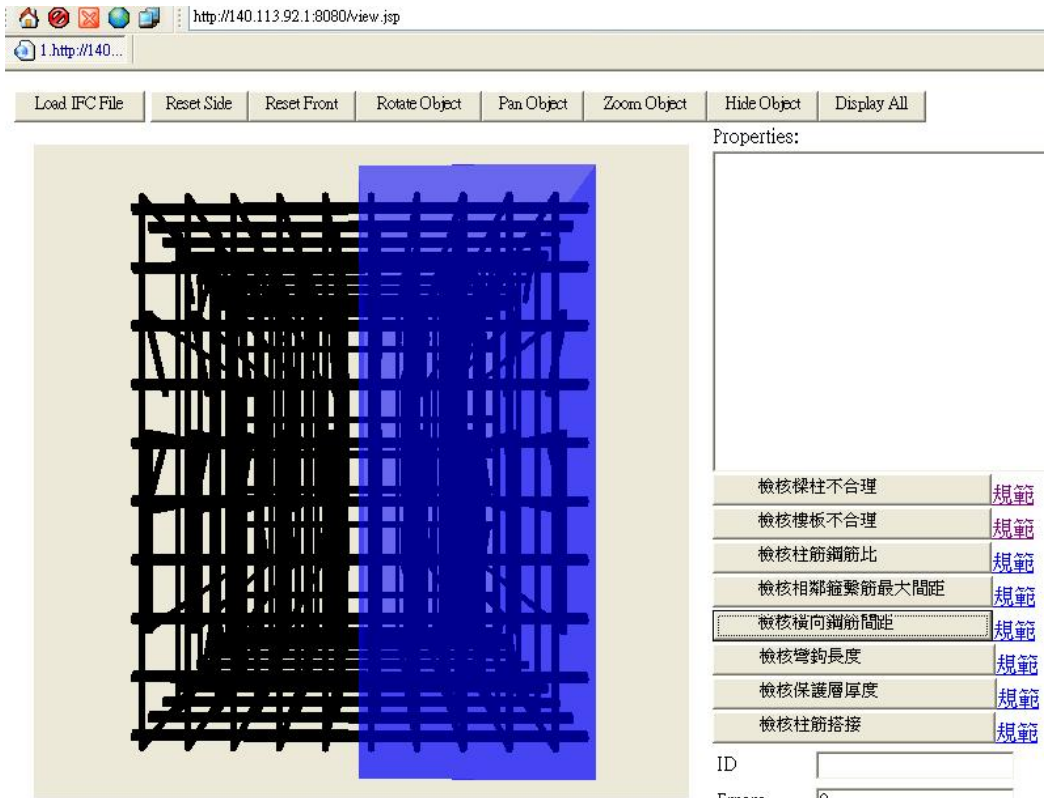


圖 5-4-19 檢核橫向鋼筋間距，沒有錯誤

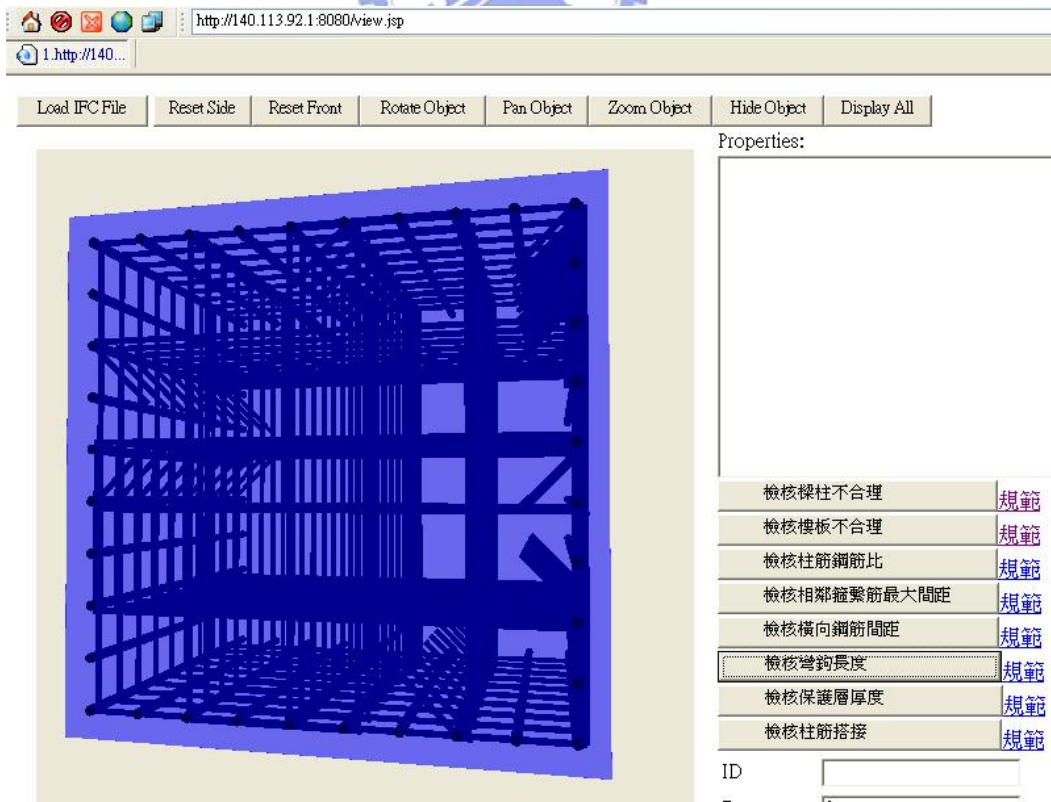


圖 5-4-20 檢核鋼筋彎鉤長度，沒有錯誤

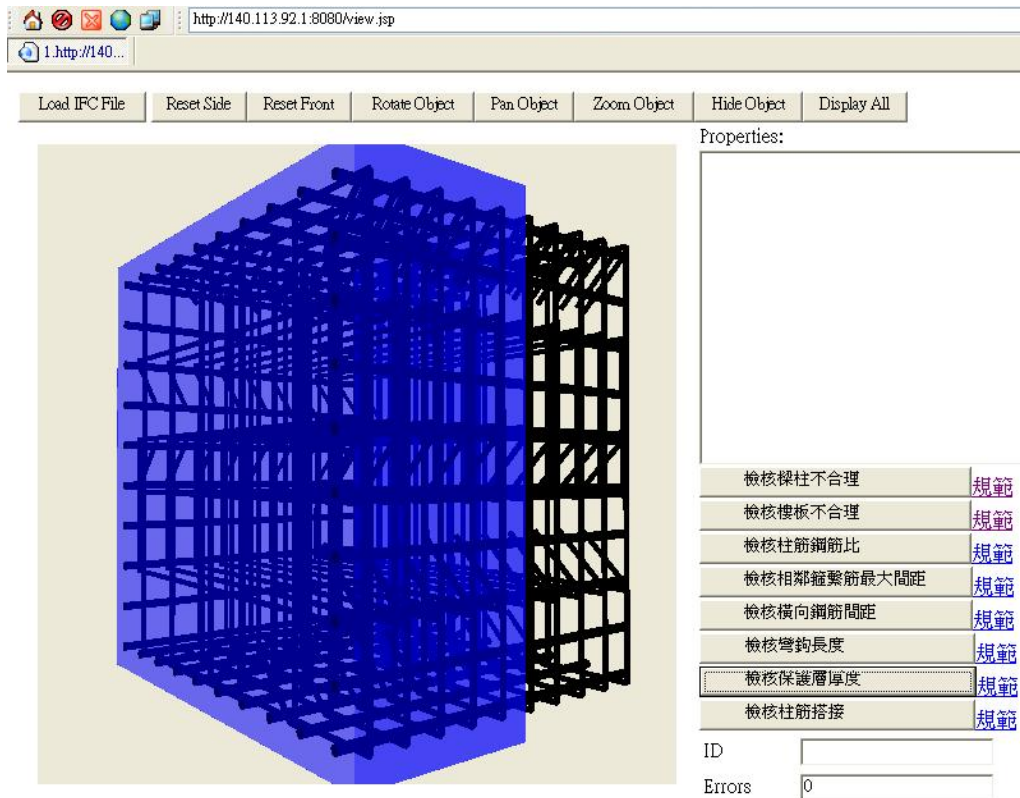


圖 5-4-21 檢核保護層厚度，沒有錯誤



圖 5-4-22 跳出新視窗，柱筋搭接之規範