



應用 HTML5 及版本控制技術 提昇 Web-based 營建資訊管理系統 使用效率之研究

撰寫人：何岳峰、黃濬彥、謝孟勳

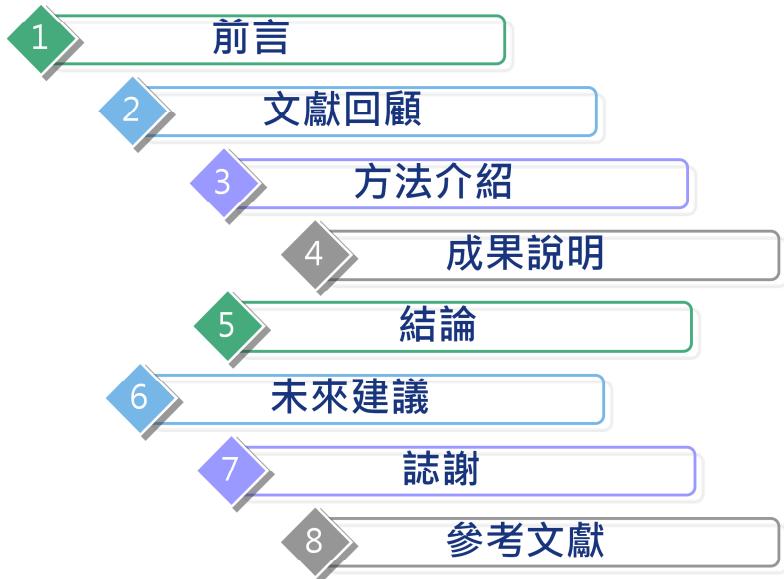
報告人：何岳峰

謹謝「本研究承蒙 2009~2010 年經濟部科技研究發展專案-學研聯合研究計畫提供經費(編號：98-EC-17-A-29-S2-0034)，特此申謝。」

1

各位老師、先進、同學，我現在要為各位報告「應用 html5 及版本控制技術提升 Web-based 營建資訊管理系統使用效率」，我是何岳峰。

簡報大綱



2

我的簡報大綱分成 8 個部份。

前言(1/3)

1. 使用網路技術，能為營建業提高效率及降低成本
[Tam, 1991]
2. 營建業是高度資訊相依的產業之一
[Chassiakos, 2008]
3. 目前這些要求難以達到或需要人員手動整合
[Navon, 2007]
4. 營建工程的施工區屬於戶外或郊區。不利以下資訊的收集
 - 1) 品質文件
 - 2) 施工日誌
 - 3) 監造報表
 - 4) 檢驗停留點相片
 - 5) 及其他工地產出資料

3

網路技術可以為營建業提高效率及降低成本，因為營建業是高度資訊相依的產業之一，然而這此要求難以達到或需要人員手動整合，像是營建工程的施工區常屬於戶外或效區，那麼這些資料的收集就會出現部份的資訊缺口。

前言(2/3)



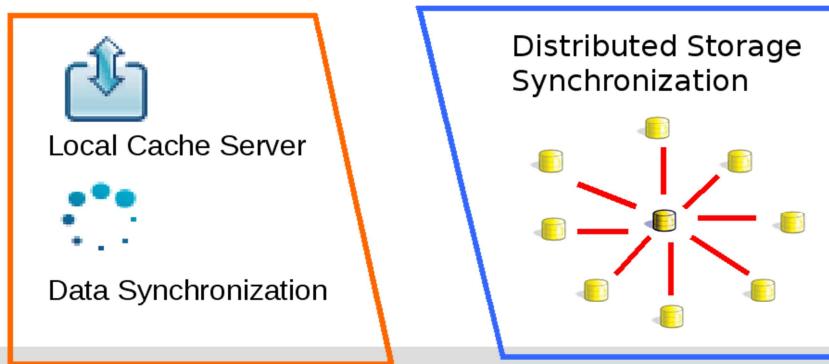
4

本研究希望提出一個機制讓營建資訊管理系統在有網路時，是一個網頁應用系統，在無網路時，是一個桌面應用系統。

前言(3/3)

研究基礎：

- 1) HTML5
- 2) Javascript
- 3) Offline Storage
- 4) Distributed Version Control



5

我們所利用的技術包含 html5, javascript, 離線儲存器及分散式版本控制技術。

文獻回顧 - 營建專案管理系統(1/4)

1. TITS(Total Information Transfer System)
[Deng , 2000]
2. 營建人力的成本控管
[Abudayyeh , 2000]
3. 利用 ADC(automated data collection) 設備透過網路傳遞資料來評估人員生產力或土方工程
[Navon, 2008]
4. 即時資訊整合系統來管理工程進度及控制品質
[Leung , 2008]

6

Deng , abudayyeh , Navon , 林皆有應用資訊技術在營建業上以提升效率。

文獻回顧 - 營建專案管理系統(2/4)

對營建工程相關公司而言，重要的資訊運用技術：

文件管理	工作流管理	人員通訊目錄	工作項目紀錄
進階搜尋	會議及電子白板	線上討論	排程及行事曆
即時拍照	施工相片	檔案格式轉換	列印
自定網站版面	離線存取	傳真/Email/語音服務	金融服務
整合無線網路	專案資訊封存	資訊服務	電子化採購

[Nitithamyong , 2004]

7

而這是 pollaphat 整理出對營建工程相關公司重要的資訊運用技術。

文獻回顧 - 營建專案管理系統(3/4)

工程現地必備的工作：

- 1) 工作項目數量紀錄
- 2) 施工相片功能

而「施工現地的網路環境條件參差不齊」：

- 河道整治、邊坡防治、水土保持等類型，其基地往往座落在鄉野僻壤之間
- Leung 的即時資料整合系統利用 GPRS/3G/Wi-Fi 等網路科技解決傳輸問題 [Leung, 2008]
- 大樓物業的設備管理，離不開 Wi-Fi

8

在營建資訊管理系統中，二個重要且必備的系統是施工項目數量紀錄系統及施工相片紀錄系統。

但是施工現地的網路環境條件參差不齊，像是河道整治、邊坡防治、水土保持等類型，其基地往往座落在鄉野僻壤之間。而林能利用 GPRS/3G/Wi-Fi 等網路科技解決資料傳輸問題，是因為他的範例工程在一般的都市之中。

另外像是大樓物業的設備管理系統，如果它是網頁系統的話，那它一樣離不開 Wi-Fi 訊號，但一般大樓常有 Wi-Fi 死角，像地下室就不容易建置 Wi-Fi 基地台。

文獻回顧 - 營建專案管理系統(4/4)

小結：



營建資訊管理系統對營建業**很重要**

但**離不開網路**對營建業是個大缺點



9

從資訊系統的文獻回顧中，我們可以得到一個小結論，就是營建資訊管理系統對營建業**很重要**，但**離不開網路**對營建業是個大缺點。

文獻回顧-HTML5

WHATWG 提出 [Hickson , 2010]

- 與現行 **HTML4 差別**：
 - ① 內嵌媒體播放
 - ② 平行處理
 - ③ 二維繪圖
 - ④ 離線存取
 - ⑤ 瀏覽器端儲存器
- 市面上的**瀏覽器大多已部份支援**：
 - ① Safari 5+
 - ② Firefox 3.6+
 - ③ Chrome 5+
 - ④ Opera 10+
 - ⑤ IE 9+

10

在 html5 的部份，它與相行 html4 差別有這些，目前市面上的瀏覽器大多已部份支援 html5 功能，而本研究所需的離線存取及瀏覽器端儲存器則是全部都有支援。

文獻回顧-版本控制(1/3)

- 軟體開發與維護中最重要的問題就是**版本控制** [RCS，1985]
- **集中式**的版本控制技術：
 - ① CVS
 - ② Subversion
- **分散式**的版本控制技術：
 - ① Mercurial [Mercurial, 2009]
 - ② Git
 - ③ BZR
 - ④ BitKeeper
- 將版本控制技術**應用在資訊系統**中：
 - 課程教學內容管理 [Clifton，2007]
 - 學生作業管理 [Karen，2005]

11

- 軟體開發與維護中最重要的問題就是**版本控制** [RCS，1985]
- 版本控制技術可分為集中式及分散式
- 將版本控制技術**應用在資訊系統**中：
 - 課程教學內容管理 [Curtis，2010]
 - 學生作業管理 [Karen，2005]

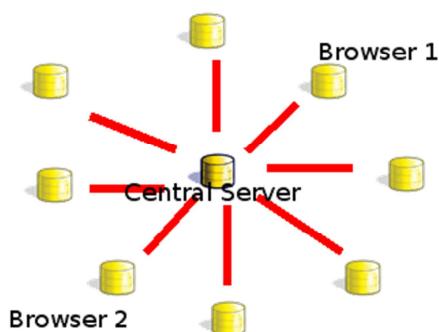
文獻回顧-版本控制(2/3)

- 分散式比集中式要好 [Alwis , 2009] :
 - 權限管理彈性
 - 無網路環境下，大部份功能仍可執行
 - 可作單元交付(Atomic Commit)
 - 支援自動合併
 - 可提供實驗性修改

12

• 分散式比集中式要好的部份有這些

文獻回顧-版本控制(3/3)



- 不同的瀏覽器可從一份中央伺服器下載資料
- 不同的下載時點將導致不同版本的資料
- 使用者在離線階段乃編輯的是瀏覽器上的資料庫(儲存器)
- 分散式版本控制技術中，其儲存單元為檔案
- 分散式儲存器中，其儲存單元為資料庫的單筆紀錄

13

本研究運用分散式版本控制技術的基礎在於，當不同的瀏覽器從中央伺服器下載資料後，只要使用者離線了，那麼使用者的資料就會與中央伺服器上的資料不一致了，有可能是使用者修改本機上的資料，也有可以是其他使用者修改了中央伺服器上的資料。

方法介紹-離線存取(1/2)

網頁原始碼範例：

```
<html manifest="cache.manifest">  
  <head>  
    <title>這是測試</title>  
  </head>  
  <body>...</body>  
</html>
```

14

HTML5的離線存取技術，就是在網頁原始碼中新增一個 manifest 屬性。

方法介紹-離線存取(2/2)

cache.manifest 內容：

```
CACHE MANIFEST  
NETWORK:  
comm.cgi  
CACHE:  
style/default.css  
images/sound-icon.png
```

1. 需以 text/cache-manifest 的 MIME Type 格式輸出。
2. 首行需為 CACHE MANIFEST
3. NETWORK 後所接網址為連線狀態下須重新下載更新。
4. CACHE 後所接網址，則一律下載至瀏覽器快取，除非 manifest 文件內容有變，才更新下載。

15

而這個 cache.manifest 的檔案要照一個特定的格式紀錄要離線使用的檔案有那些。

它需要以 text/cache-manifest 的 MIME Type 格式從網頁伺服器上輸出。

首行需為 CACHE MANIFEST

NETWORK 後所接網址為連線狀態下須重新下載更新。

CACHE 後所接網址，則一律下載至瀏覽器快取，除非 manifest 文件內容有變，才更新下載。

方法介紹-瀏覽器端儲存器

- 可在瀏覽器上儲存資料
- Key-Value 形式
- 利用 Javascript 的 localStorage API 存取
 - getItem 取值
 - setItem 賦值
- javascript 範例：

```
1) alert(localStorage.getItem('foo')); //第一次顯示 undefined  
2)                                         //第二次以後皆顯示 bar  
3) var s = 'bar';  
4) localStorage.setItem('foo', s);  
5) alert(localStorage.getItem('foo')); //永遠顯示 bar
```

16

除了網頁的檔案能先下載到瀏覽器中，使用者要編輯的資料也要放在瀏覽器中，他才有東西可以編輯。

這部份利用 HTML5 的瀏覽器端儲存器功能。它是一種 key-value 的資料形式，然後使用 getItem 函式來取值， setItem 函式來賦值。

這個範例中，第一行是顯示 foo key 的值為何，如果是使用者是第一次執行這個 javascript 程式，則它的值為是 undefined ，等到執行 3~5 行時，我們才會在這瀏覽器中，設定 foo key 的值為 bar ，那麼之後，使用者來到相同的網頁時，它都會顯示 bar 了。

方法介紹-分散式儲存器同步技術(1/5)

1. 使用者在瀏覽過網站後，先將中央伺服器的網頁、
javascript 程式檔、圖檔、數據皆下載並儲存在瀏覽器
內的「儲存器」上
2. 不同的下載時點造成各使用者的瀏覽器內的「儲存器」
其資料不一致
3. 使用者在離線環境下，新增、修改、刪除本機瀏覽器
內的「儲存器」數據
4. 不同版本的「儲存器」資料在連線環境下，同步回中
央伺服器的方法有三：
 - ① 後者為主
 - ② 鎖定後編輯
 - ③ 分散式版本控制

17

一個可離線的網頁系統，使用者的使用流程如下：

首先他先要瀏覽器整個網站，將中央伺服器的整個網頁皆下載並儲存在瀏覽器內的「儲存器」上。

不同的下載時點造成各使用者的瀏覽器內的「儲存器」其資料不一致。而使用者接下來可作離線的動作，並在離線環境下新增、修改、刪除本機瀏覽器內的「儲存器」數據。

之後在回到辦公室，將這些與中央伺服器不同版本的資料，同步回去。同步的方法有三：

方法介紹-分散式儲存器同步技術(2/5)

後者為先：

- 1) 後令壓前令原則
- 2) 實作簡單
- 3) 同現今的網頁應用程式

	A 使用者	B 使用者	C 使用者	D 使用者	
下載時間	7/10	7/11	7/10	7/12	• A 填完了，B 有更新到
上傳時間	7/11	7/12	7/13	7/14	• B 填完的資料卻被 C 洗掉
結果	被 B 更新	被 C 洗掉	被 D 洗掉	有考慮 A, B 的數據，但刪除了 C 的數據	• 還好 D 有更新到 B 的資料，但 D 還是把 C 的資料洗掉。

18

第一種是後者為先，以下是 4 個使用者對同一份網頁表單的下載時間及填寫上傳時間。

所以我們可以看到
A 填完了，B 有更新到
B 填完的資料卻被 C 洗掉
還好 D 有更新到 B 的資料，但 D 還是把 C 的資料洗掉

使用「後者為主」的原則在某些狀況下，會失去部份使用者的填寫資訊。

方法介紹-分散式儲存器同步技術(3/5)

鎖定後編輯：

- 1) 以權限方式鎖定編輯功能
- 2) 實作簡單
- 3) 不利分工合作，資訊系統在權限設定上的實作將非常困難
- 4) 若有人鎖定專案後，卻放了長假，其他人都不能用了

19

另一種是由某人主動鎖定整份表單，除了他能修改外，別人都不行。但這**不利分工合作**，資訊系統在權限設定上的實作將非常困難，且若有有人鎖定專案後，卻放了長假，**其他人都不能用了**。

方法介紹-分散式儲存器同步技術(4/5)

分散式版本控制 [Mercurial , 2009] :

- 1) 從中央伺服器下載版本為 A 版本
- 2) 使用者在離線階段編輯的版本為 C 版本
- 3) 使用者欲同步時，再下載一次中央伺服器數據，此為 B 版本
- 4) 當 A 等於 B ，則直接上傳(push) C 版本
- 5) 當 A 不等於 B ，則使用者要檢視 B 與 C 的差異並手動合併 (merge) 資料

20

比較好的方法是用分散式版本控制技術來同步這些離線資料。

使用者所下載的資料版本我們定為 A 版本，在離線編輯後，我們定義為 C 版本，等到要同步時，我們再一次下載伺服器上的資料，其版本，我們定義為 B 版本。

再來比較 A, B 版本，如果相同，代表使用者的編輯的時間內，沒有其他人作編輯動作，自然以 C 版本上傳。如果 A, B 的比較有所不同，則由使用者檢視 B, C 之間的差異並手動合併資料後再上傳。

方法介紹-分散式儲存器同步技術(5/5)

分散式版本控制的同步程式有以下特點：

1. 資料皆由中央資料庫複製而來，如果複製時點相同，初始資料皆是一致的，而後經使用者修改，才有不一致的現象。
2. 資料本身**不具有**邏輯相關性及整合性。
3. 從使用者的角度來看，他要能明確了解目前所操作的是中央資料庫或是離線儲存器。
4. 本系統的資料整合工作是由**使用者自行決定**。

21

分散式版本控制的同步程式有以下特點：

資料皆由中央資料庫複製而來，如果複製時點相同，初始資料皆是一致的，而後經使用者修改，才有不一致的現象。

資料本身**不具有**邏輯相關性及整合性。

從使用者的角度來看，他要能明確了解目前所操作的是中央資料庫或是離線儲存器。

本系統的資料整合工作是由**使用者自行決定**。

方法介紹-分散式儲存器同步技術(5/5)

分散式版本控制的同步程式有以下特點：

5. 所有的離線儲存器只與中央資料庫有整合關係，離線儲存器之間則無。
6. 採用分散式版本控制技術，可將用戶的本機儲存器資料在網路連線狀態下，同步回中央資料庫，並由使用者決定什麼資料是正確的，合理地解決資料版本不一致的問題。而同步化原則是一次只能同步一個離線式儲存器至中央資料庫，所以要在 merge 及 push 階段，短暫鎖定中央資料庫，待 push 完成後才解鎖。

22

所有的離線儲存器只與中央資料庫有整合關係，離線儲存器之間則無。

並且要在 merge 及 push 階段，短暫鎖定中央資料庫，待 push 完成後才可解鎖。

成果說明 - 以施工日誌系統為例(1/2)

依工程會的『公共工程施工品質管理作業要點』，營造廠商須逐日填寫施工日誌

- 1) 施工數量
- 2) 人機料使用數量
- 3) 勞安衛事務處理紀錄
- 4) 取樣試驗紀錄
- 5) 通知分包商辦理事項
- 6) 其他重要紀錄

23

我們以一個施工日誌系統作應用實例

成果說明 - 以施工日誌系統為例(2/2)

- 營造廠商先在網路環境下，下載工程案的施工項目資訊
 - 1) 施工項目名稱
 - 2) 單位名稱
 - 3) 設計數量
 - 4) 單價
- 在無網路環境下，在瀏覽器中點選要紀錄的日期，針對施工項目填寫數量及備註，以及其他處理紀錄
- 回到辦公室或任一具備網路環境的空間，將離線資料同步回中央資料庫

24

- 所以營造廠商先在網路環境下，下載工程案的施工項目資訊
- 在無網路環境下，在瀏覽器中點選要紀錄的日期，針對施工項目填寫數量及備註，以及其他處理紀錄
- 回到辦公室，將離線資料同步回中央資料庫

成果說明 - 填寫2010-6-1的施工日誌

填寫「2010-06-01」的報表

連線時下載資料

上午天氣： 晴天 ☀ 下午天氣： 晴天 ☀

離線時編輯

一、依施工計畫書執行按圖施工概況(含約定之重要契約項目及完成數量等):

項次	工項名稱	單位	設計數量	單價
1	東內堤釣區	式	1.000	2759092.000
1-1	內釣區護欄	組	43.500	11590.000
1-2	東內堤釣魚平台	式	1.000	1072291.000
1-2-1	既有混凝土打除	m3	48.350	232.000
1-2-2	海側平台護欄g1(TYP.1)	座	4.000	1517.000
1-2-3	海側平台護欄g1(TYP.2)	座	5.000	3159.000
1-2-4	海側平台護欄g2(TYP.1)	座	4.000	2093.000
1-2-5	海側平台護欄g2(TYP.2)	座	5.000	4551.000
1-2-6	S1板	M2	268.400	1252.000
1-2-7	CS1板	M2	57.200	1363.000
1-2-8	G1縱梁(東內堤釣魚平台)	式	1.000	140121.000
1-2-9	地梁TB2(東內堤釣魚平台)	式	1.000	75185.000
1-2-10	F1基礎版	處	9.000	3041.000
1-2-11	梁B1(東內堤釣魚平台)	式	1.000	72400.000
1-2-12	C1柱(東內堤釣魚平台)	式	1.000	53348.000
1-2-13	20mm不銹鋼索	M	537.000	371.000
1-2-14	五金零配件	式	1.000	10202.000
1-2-15	塑鋼門,850*1900mm	樘	1.000	6956.000
1-2-16	橫拉塑鋼窗,1500*1100mm,th=8mm強化玻璃,附紗窗	樘	1.000	9275.000
1-3	東內堤釣魚平台階梯	式	1.000	226230.000
1-3-1	地樁TB3	M	9.400	4352.000
1-3-2	地樁TB1	M	3.900	2307.000

紅色區塊是廠商在連線環境時所下載的。

藍色區塊是廠商在離線環境時所填寫編輯的。

成果說明 - 填寫2010-6-1的施工日誌

- A 版 = 同伺服器原版
- B 版 = 在使用者下載資料後，伺服器上的資料已被其他人更新過
- C 版 = 使用者在離線環境下，所填寫的資料

漁港碼頭面及相關設施維護及釣魚區設置工程 2010-06-01 施工日誌同步表單

編號	施工項目	A. 伺服器原版	B. 其他人版本(<input type="checkbox"/> 完全套用)	C. 您的版本(<input type="checkbox"/> 完全套用)
1.	東內堤釣區/內釣區護欄	0	0	<input checked="" type="radio"/> 20, 0k+300開始
2.	東內堤釣魚平台/既有混凝土打除	48.35	48.35	48.35
3.	東內堤釣魚平台/S1板	0	<input type="radio"/> 10	<input checked="" type="radio"/> 20
4.	東內堤釣魚平台/CS1板	0	20	20
5.	東內堤釣魚平台階梯/地樑TB3	0	<input checked="" type="radio"/> 22	<input type="radio"/> 0

取消同步 確定同步

26

接下來，廠商回到辦公室，可同步資料回中央伺服器，他所上傳的資料會遇到 5 種狀況

第一種是在離線這段時間之內，只有他一個人有編輯過。

第二種是在離線這段時間之內，沒有人有編輯動作。

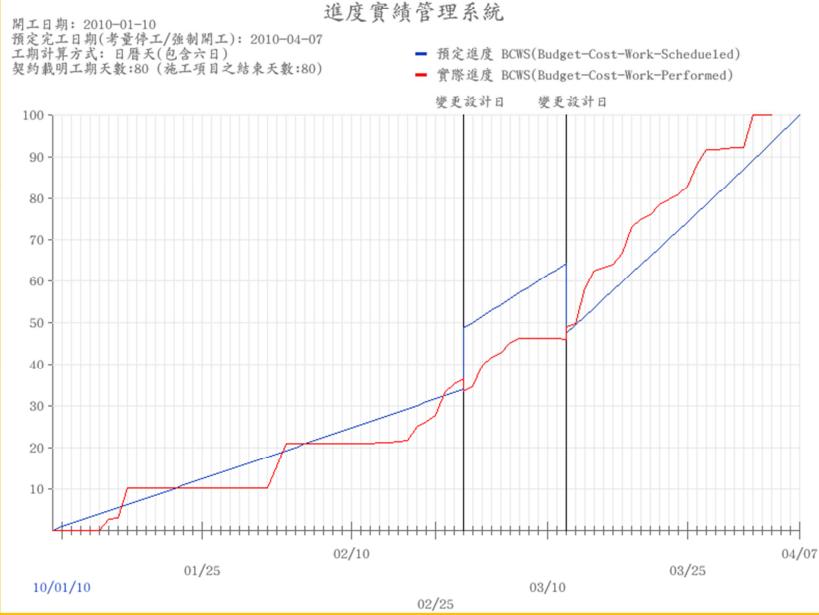
第三種是在離線這段時間之內，有其他人也作了編輯，但數量居然不一樣。

第四種是在離線這段時間之內，有其他人也作了編輯，但數量是一樣的。

第五種則是，自己沒編輯，但其他人有編輯。

在這五種狀況下，廠商在同步時，自行判斷到底要依那一版本為主，設定好後就可以上傳了。

成果說明 - 中央伺服器上的運算結果



27

上傳後，在中央伺服器上，可以提供較豐富的計算，像這張圖是該工程的施工進度，紅色表實際進度，藍色表預定進度，而該廠商有作了兩次的變更設計。

結論(1/2)

1. 使用網頁應用程式架構，使用者無須安裝軟體，系統升級不會造成使用者電腦系統的衝突，所以容易大量佈署，符合營建工程參與人數眾多的特性。
2. 利用 Total Web-based 技術可突破營建郊外工區無法使用網路技術之困境，大量運用在營建專案資訊管理系統中，可扭轉現行工地紙本作業的落伍現象，進而大大提昇工程品管效率，也促使公共工程達到預期之功能與投資報酬。

28

1. 使用網頁應用程式架構容易大量佈署，符合營建工程參與人數眾多的特性。
2. 利用 Total Web-based 技術可突破營建郊外工區無法使用網路技術之困境，可扭轉現行工地紙本作業的落伍現象，進而大大提昇工程品管效率。

結論(2/2)

3. 從資料接受端即達到資訊數位化，**避免多次謄寫**造成的人工錯誤及降低往來溝通成本，**以便利營造廠商、監造單位、主辦機關、主管機關**於無網路之偏遠地區**進行工程品質管控**，且利用網頁應用程式將施工進度計算動作交由伺服器處理，提高正確率及減少計算時間，**有助降低廠商的管理成本**。

29

從資料接受端即達到資訊數位化，**避免多次謄寫**造成的人工錯誤及降低往來溝通成本，**有助降低廠商的管理成本**。

未來建議

- 本研究以施工日誌管理系統為實例來說明分散式儲存器的同步方法，就施工日誌而言，它的資料類型屬於數字與字串，在版本比較上，較為容易。但工地管理上，施工相片的紀錄也很重要，如何將離線編輯相片資料及伺服器原版、其他人版本作比較，這部份須發展其他比對方法，才能有效率地同步相片資訊。

30

本研究以施工日誌管理系統為實例來說明分散式儲存器的同步方法，就施工日誌而言，它的資料類型屬於數字與字串，在版本比較上，較為容易。但工地管理上，施工相片的紀錄也很重要，如何將離線編輯相片作版本比較，這部份須發展其他比對方法，才能有效率地同步相片資訊。

誌謝

本研究承蒙 2009 ~ 2010 年經濟部
科技研究發展專案 -
學研聯合研究計畫提供經費
(編號 : 98-EC-17-A-29-S2-0034)

特此申謝。

參考文獻(1/2)

1. Z.M. Deng, H. Li, C.M. Tam, Q.P. Shen, and P.E.D. Love, "An application of the Internet-based project management system," *Automation in Construction*, vol. 10, Jan. 2001, pp. 239-246.
2. Abudayyeh O.[1], Temel B., Al-Tabtabai H., and Hurley B., "An Intranet-based cost control system," *Advances in Engineering Software*, vol. 32, Jan. 2001, pp. 87-94.
3. R. Navon, Automated productivity control of labor and road construction, in *25th International Symposium on Automation and Robotics in Construction*. Vilnius, Lithuania, 29-32, 2008.
4. A. Chassiakos and S. Sakellaropoulos, "A web-based system for managing construction information," *Advances in Engineering Software*, vol. 39, Nov. 2008, pp. 865-876.
5. N. Šuman and M. Pšunder, "Mobile Computing Changing the Traditional Ways of Organizing the Construction Company," vol. 5, 2008, pp. 42-47.
6. R. Navon, "Research in automated measurement of project performance indicators," *Automation in Construction*, vol. 16, Mar. 2007, pp. 176-188.
7. C.M. Tam, "Use of the Internet to enhance construction communication:Total Information Transfer System," *International Journal of Project Management*, vol. 17, Apr. 1999, pp. 107-111.
8. S. Leung, S. Mak, and B.L. Lee, "Using a real-time integrated communication system to monitor the progress and quality of construction works," *Automation in Construction*, vol. 17, Aug. 2008, pp. 749-757.

參考文獻(2/2)

9. P. Nitithampong and M.J. Skibniewski, "Web-based construction project management systems: how to make them successful?," *Automation in Construction*, vol. 13, Jul. 2004, pp. 491-506.
10. Ian Hickson and Google, "HTML5", Working draft, WHATWG, Jun. 2010.
11. Ian Hickson and Google, "Web Storage", Working draft, W3C, Jun. 2010.
12. W.F. Tichy, "Res — a system for version control," *Software: Practice and Experience*, vol. 15, 1985, pp. 637-654.
13. Karen L. Reid and G.V. Wilson, "Learning by doing: introducing version control as a way to manage student assignments," *SIGCSE Bull.*, vol. 37, 2005, pp. 272-276.
14. B.D. Alwis and J. Sillito, "Why are software projects moving from centralized to decentralized version control systems?," *Proceedings of the 2009 ICSE Workshop on Cooperative and Human Aspects on Software Engineering*, IEEE Computer Society, 2009, pp. 36-39.
15. Clifton, C., L.C. Kaczmarczyk, and M. Mrozek. Subverting the fundamentals sequence: Using version control to enhance course management. *Proceedings of the 38th SIGCSE Technical Symposium on Computer Science Education.*, 2007.
16. B.O'Sullivan. Distributed revision control with Mercurial. <http://hgbook.red-bean.com/>, 2009.



簡報完畢，敬請指教