

# 一個應用於CSCW環境中的協同控制機制之雛形 ~ 研究CSCW之通訊控制

姜國輝

國立政治大學  
資訊管理系副教授

楊博文

國立政治大學  
資訊管理研究所研究生

## 摘要

在CSCW環境中，提供一個共享的環境以支援組織成員溝通(communication)、合作(colaboration)以及協調(coordination)等活動，使得群體的工作能順利進行，並借由電腦系統的輔助來提昇組織的效率和品質。對於一個多使用者的資訊共享環境中，要使得群體合作能順利進行必須要能有效的協調控制成員們的動作，以避免衝突及重覆的動作以增進效率。為達此目的便須要有一個協同控制機制，來協調所有同作的異動不會相互干擾。因此本文提出一個協同控制機制的概念模式，並可依此模式發展一個協同控制協定，實際應用於CSCW環境。而此協定的主要目的是增進CSCW系統的協同程度和確保CSCW環境中資料的一致性。

關鍵字：電腦輔助協力作業，協同控制

## 壹、緒論

在以Internet為基礎的網路系統上，例如TANet，提供許多”網路上的多人合作應用的資源”，但是對於新的CSCW研究領域：“即多個在不同地點的使用者，同時間進行相關工作”，仍缺乏足夠的支援。這些活動例如是電子教室、遠程教學、視訊會議、即時討論群。上述問題的基本困難在於至今仍缺乏適合的協同控制機制[4、7]。本論文中討論一個可應用於網路上CSCW環境的協同控制機制雛形。此一模式屬於政大資管系所於TANet上CSCW相關教學研究計畫之部份工作。

在CSCW的環境中，強調的是多人在一個共享的環境裡共同完成同一件工作。雖然有些群組軟體允許使用者可以在不同時間進入系統(如:E-Mail)，但如果是強調即時的在同一環境來共同工

作，那麼勢必會遇到多人同時存取或異動同一筆資料的情形。這在共同編輯(co-editing)、共同寫作(co-authoring)以及共同設計(co-design)的應用中更加明顯[6,7]。在此情形下，如果對異動並無即時的要求，那麼只須在這群工作者中，選擇一位來負責將各別的異動整合即可。但如果一個要求即時性高的應用中，由於要求整個系統環境必須能即時的處理來自各處的異動，因此必須有一個協同機制來保證這些同作的異動不會相互干擾彼此異動的處理，並且即時地將最新的資訊傳送給群體成員。因此本文便針對這個問題來發展一個協同控制機制模式(Model)，並可依此模式發展一個協同控制協定，實際應用於CSCW環境。而此協定的主要目的是增進CSCW系統的協同程度(degree of concurrency)和確保CSCW環境中資料的一致性(Consistency)。

有關多人同時分享資料的協調機制，傳統的資料庫管理系統技術提供了不少解決方法，Rakesh曾將這些方法概略分為三類：1.凍結法(blocking)，兩段式鎖定法(two-phase locking: 2PL)是最常見的一種方法。2.基本時標法(Basic Timestamp Ordering)。3.樂觀法(Optimistic) [1]。

雖然DBMS技術所提供的資料模式化(data modeling)和交易管理(transaction management)非常適用於企業的資料處理，但卻不適用於CSCW的應用。原因在於在一個CSCW的環境中，要處理的資料可能包括結構化資料物件(data objects)的集合(如：圖形，軟體模組....等)，並且資料彼此間的關係可能更為複雜。雖然新的程式語言提供了更優良的抽象化機制(excellent abstraction facilities)來處理這些類別的資料，但是對於資料的儲存和分享卻僅提供少數的支援。使得在發展一個CSCW環境時，必須另外撰寫適用於此系統的資料管理程序[2]。因此本模式便依據CSCW環境的

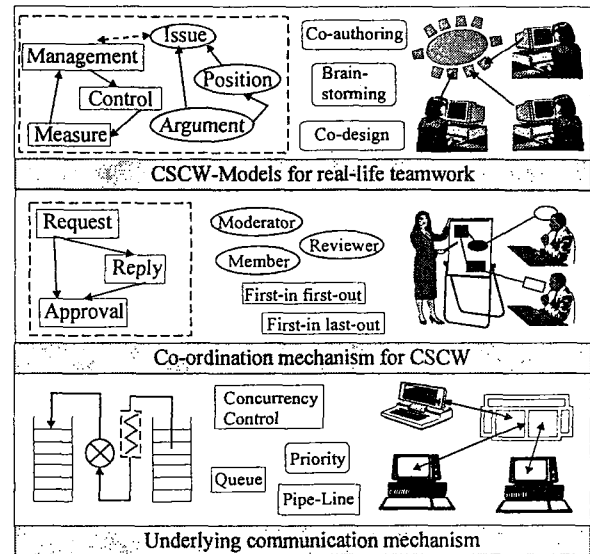
特性，參考適當的資料庫協同控制方法加以修改和擴充，發展成爲一個可以應用於CSCW環境的協同控制協定。

## 貳、研究背景及目的

隨著個人電腦功能的增強，以及電腦網路技術的進步與普及，使得電腦系統的應用逐漸由傳統的主機-終端機以及個人單機的使用，開始朝向開放性分散式系統的應用。過去大部份的軟體系統僅支援使用者和系統之間的互動(例如：編輯一份文件，對資料庫做資料查詢)，即使是針對多使用者應用所設計的系統(如：辦公室資訊系統)也僅提供有限的使用者與使用者之間互動的支援。然而，在現實世界中有很多半結構化(semi-structured)和非結構化(un-structured)的問題，並無法以傳統的分析方式或以結構化的處理程序來解決。而必須透過群體討論的方式，針對問題彼此間進行相互溝通和協調，共同努力來找出解決方案。因此很明顯的，一個能提供群體間互動支援的系統環境，對於需要群體智慧的應用相當重要。爲了要支援這樣的應用，勢必要採用不同以往的軟、硬體架構，來發展一個適用於支援群體合作的電腦環境。因此，在1980年代開始陸續有學者提出電腦輔助協合作業(Computer-Supported Cooperative Work: CSCW)的觀念、理論和架構，作爲發展群組軟體(groupware)的基礎。依Eills對群組軟體的定義[4]:

“群組軟體提供一個資源共享環境的界面，以支援群體成員從事同一工作(或達成同一目標)的電腦軟體系統”

由定義可知群組軟體的目的在於支援組織成員的溝通(communication)、合作(colaboration)以及協調(coordination)等活動，使得群體的工作能順利進行，並借由電腦系統的輔助來提供組織的效率和生產力。爲了達到這個目的，群組軟體在設計上便需要能1.提供成員一個資訊共享的環境，使得合作能順利進行，2.有效的協調控制成員們的動作，避免衝突及重覆的動作以增進效率，3.透過通訊網路使分散各地的群體成員能有效溝通。因此我們建構了一個發展CSCW環境的三階層概念模式[7](如圖一)，目前我們開發CSCW環境之策略便是依此架構來發展，本論文所提出之協同控制機制模式便屬於架構中的最底層。

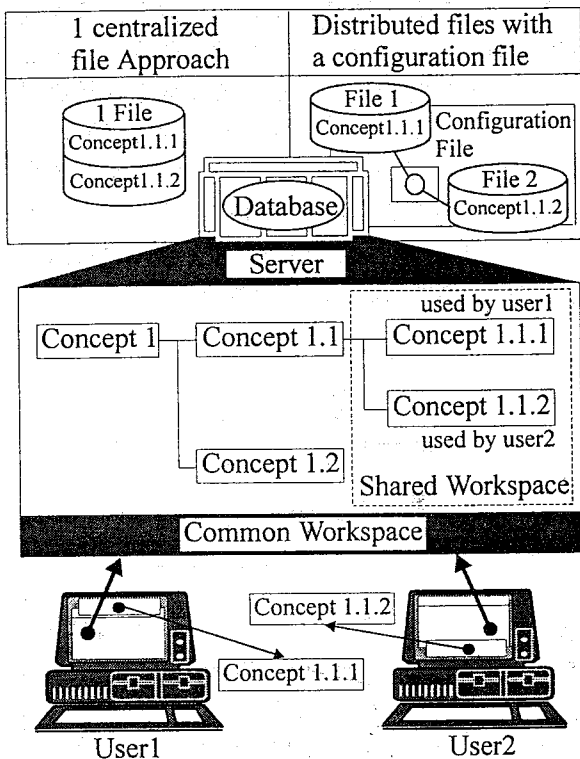


圖一. A three-layer schemata for developing CSCW environment

雖然在多人使用的資料庫或分散式資料庫中，有關協同控制的研究和探討相當多。但針對有關CSCW環境中協同控制問題的探討，似乎較少學者發表相關的研究成果。此外在PC上要透過NFS來達到資訊分享似乎有其限制。如圖二，如果user1和user2在一個分享的環境(Shared Workspace)中共同工作，user1目前正使用Concept1.1.1的資料，而user2則使用Concept 1.1.2的資料。雖然兩者對於目前所使用的資料有相當大的自主權，但如果user1要對Concept 1.1.2的資料做存取，以目前的PC環境無法達到這樣的要求。因此，本研究便希望發展一個適用於CSCW環境中的協同控制機制，並可解決目前在PC環境上有關同作異動的問題。

由於同樣是處理協同控制的問題，許多分散式即時系統和多使用者資料庫(Multiuser Database)所採用的協同控制的概念，可適用於CSCW環境中來解決有關協同控制的問題。但由於兩者的應用不同，因此必須將這些方法做適當的修改和擴充，才能使應用於CSCW環境中。例如：在一個多人使用的資料庫，是支援各人完成自己的工作，因此對於資料的異動並不須主動的通知所有的使用者。然而在CSCW環境中是透過資訊分享的方式，讓群體成員共同解決同一問題或完成同一工

作，因此如果資料有新的異動必須讓所有群體成員能獲得最新的資訊[6]。所以在採用原先針對多使用者資料庫所發展的協同控制方法時，必須做適當的修改和擴充使其能應用於CSCW環境中。因此在本研究的第一階段便是先尋找適合的多使用者資料庫的協同控制模式加以擴充，並驗證其運用在CSCW環境的可行性。在研究的過程中，發現文獻[5]所提出的即時性物件導向資料庫的協同控制機制模式非常適合上述需求，因此便以此模式為基礎做進一步之發展。



圖二. 使用PC NFS來負責資訊分享的困難

### 參、系統一環境架構

由於CSCW環境是支援多人共同分享工作環境來完成同一工作，因此在此環境中必定有許多使用者由各地進入此工作環境。為了符合目前朝向分散式系統環境的趨勢，因此假設此協同控制機制未來是用在主從架構(Client/Server Architecture)的環境中。環境架構採用分散式系統架構的好處是可以將有關使用者介面的處理交由Client端，而和資料處理有關的工作則交由Server端負責，如此可以增加處理的效率。雖然環境架構採用分散式系統架構，但在資料的管理上則採用集中式的管理，亦即所有有關資料的異動均須透過Server端的

協同控制機制來存取資料。採用這樣的方法是針對CSCW應用的特性，在一個支援群體工作的環境中，無論是群體決策、會議系統、共同寫作或共同設計的系統，較少會有須處理大量資料的情形，因此並不須在Local端另外維護一個資料庫。相反的，在CSCW環境中，由於大家是針對同一議題作討論或是對同一文件作處理。因此必須即時的了解當時每個人的意見及想法，或是文件的最新內容。如果採用分散式資料庫的方式，那麼為了維護資料的一致性，必定會造成系統額外的負擔。此外，在CSCW環境中大部份的資料異動其資料量均不會太大，如果採用集中式資料管理方式並不會造成通訊網路太大的負擔。因此在資料儲存和管理方面將採用集中式的方式。

另一方面，在一個即時性的CSCW環境中，異動處理的正確性不只和維護資料的一致性有關，同時也與異動是否能在時限內完成有密切的關係。因此，在此模式中必須使用一個有效的異動排程策略來使得異動皆能在其時限之內完成。所採用的異動排程法則主要是以異動的優先序大小做為排程的依據。關於優先序的指定策略主要有最早死限法(Earliest Deadline: ED)、最少迴轉時間法(Least Slack Time: LST)、以及先到先服務法(First Come First Served: FCFS)。由於這些優先序的指定策略原本主要應用於即時性資料庫系統中，雖然可以達到即時性的要求，但在一個即時性的CSCW環境，其作業特性和即時性資料庫系統仍有不同，因此必須另外擬定符合即時性CSCW環境的優先序指定策略，其修正後的演算法則如下：

$$\text{Prio}(T_i) = W_1 * D + W_2 * Tr + W_3 * Cl$$

Prio(T<sub>i</sub>) : 對於每一個異動T<sub>i</sub>有一個與之對應的優先序Prio

D : deadline

Tr : transaction rank

Cl : class level

W<sub>i</sub> : weighting factor

在此異動的優先序指定策略中，考慮CSCW系統是建置在一個分散式架構以及多工的環境中，因此每一個異動必須儘量讓它在時限內完成。在此方法中對於擁有較早時限的異動給予較高的優先序。除了考慮異動的時限外，尚考慮異動的範圍Tr(異動範圍愈大，優先序愈高)和異動的重要

性CI(例如在co-authoring的應用, 如果更改章節的條目, 應比更改內文更先被處理)。另外此方法利用參數設定與權重的調整, 使得此模式可以用於各種不同的CSCW應用。

在協同控制方面, 此模式採用兩段式鎖定法, 並且允許當異動間有存取的衝突狀況發生時, 擁有較大優先序的異動有權力去插斷(preempt)優先序較小者。採用兩段式鎖定法的原因是因為它較適用於CSCW環境, 而其他協同控制方法並不適合。例如: 基本時標排序法必須隨時維護各個資料項最後一次的讀寫記錄, 這樣會造成系統的額外負擔, 而這問題在一個多人共用且分散式的CSCW環境更加嚴重。而樂觀法其假設前提是不同的異動在執行時發生衝突的機會非常小, 因此若異動作業過度地集中在某些特定的資料時, 則此法並不適用。然而在CSCW環境中由於有多人共同合作, 因此對於有爭議的議題或彼此看法不一致的地方, 其資料的存取和異動將相當頻繁, 故此法亦不適用。

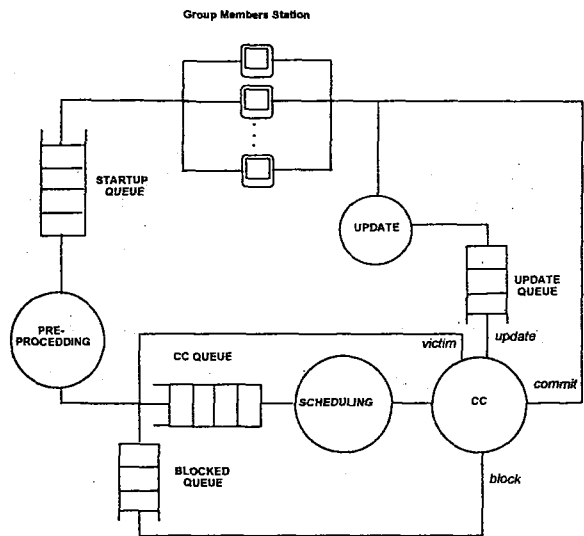
但是採用兩段式鎖定法以及可以插斷的方式並非沒有缺點。首先, 採用鎖定的方式, 對於不相容的異動將被凍結(blocked)。一但有異動被凍結, 系統便需執行死結偵測(deadlock detection)的工作以偵測系統是否有死結的情形存在。若偵測到有死結存在則應立刻予以處理, 以免影響到系統的整體績效。另外, 採可插斷的方式是針對在群體工作中, 有些資料的改變將會直接影響其他資料。因此必須先被處理, 然而採用這種方法將使得在執行中的異動因優先序較小而被插斷造成異動回復(rollback)。使得先前已執行的部份等於虛功而須將來重新再做一次, 造成系統資源的浪費, 因此在這一部份必須進一步的改良。

#### 肆、協同控制機制模式

根據前面所述的原則, 所發展的模式如(圖三)。在本模式中, 異動是由分散於各地的群組成員的工作平台(group membersstation)所發出。當一個新的異動由某一工作平台發出後, 此異動資料即透過網路傳到Server端並且馬上被送到啟動佇列(STARTUP QUEUE)中。在此時, 一些初步的準備工作如查詢佇列(queue)分析、存取權限檢查等步驟在此(PRE-PROCESSING)完成。當這些例行的工作完成後, 異動便進入協同控制佇列

(concurrency control queue: CC QUEUE)中, 並準備發出協同控制要求。

在異動發出協同控制要求之前, 系統中的即時排程程序(SCHEDULING)將先啟動, 此排程程序會審查在協同控制佇列中等待發出協同控制要求的所有異動, 根據此即時排程器所使用的異動執行優先序指定法則去計算每一個異動的執行優先序, 並將所得到的優先序值寫回異動中的相關欄位。在協同控制佇列中的每一個異動便根據其執行優先序依序發出協同控制要求, 擁有最高優先序的異動便得以最早發出協同控制要求。在發出協同控制要求的同時, 系統必須去檢查此異動是否已經錯過(missed)它的時限。若已超過其時限, 則此異動將從協同控制佇列中去掉。將其去掉是因為錯過時限的異動均被列為無效的異動, 一異動既然已經無效便無須再浪費系統的時間去執行, 以免影響到系統中的其它異動。



圖三. 應用於CSCW環境中的協同控制機制模式

		requested mode		
		NULL	R	W
current mode	NULL	Y	Y	Y
	R	Y	Y	N
	W	Y	N	N

表一. 讀寫的鎖定模式之相容矩陣[3]

若異動Ti的協同控制要求所要存取的資料項為另一異動Tj目前所存取, 且兩者的鎖定模式並不相容(見表一)。此時Ti的協同控制要求將會被凍結

(block)，異動 $T_i$ 進入凍結佇列(BLOCK QUEUE)中等待異動 $T_j$ 結束以釋出其所占有的資料項。一旦 $T_j$ 釋出了 $T_i$ 所需的資料項後， $T_i$ 將馬上從凍結佇列中移出並放入協同控制佇列中，等待發出協同控制要求。每當有異動的協同控制要求因被其它的異動鎖定導致送入凍結佇列的情形發生時，系統中的協同控制機制(concurrency controller: CC)便須啓動系統中的死結偵測(deadlock detection)機制以避免系統中發生死結的現象。

若偵測到死結的存在，則選擇死結中優先序最小的異動為犧牲者(victim)以解決死結的情況，並將做為犧牲者的異動重新啓動送入協同控制佇列中。如果一個異動為唯讀(read-only)的型式，則執行其所有的操作後，便將資料直接傳回(commit)至使用者的工作平台。若為一個修改資料的異動，則須先送到修改佇列(Update Queue)中，隨後將其修改後的資料值寫回資料庫中，並將更改結果即時傳給共同工作的群組成員。

## 伍、模式的建置

爲了能將本模式實際應用於CSCW環境中，必須選擇適當的作業平台。由於本模式在協同控制方面，允許當異動間有存取的衝突狀況發生時，擁較大優先序的異動有權力去插斷(preempt)優先序較小者。因此並不適合將其建置在於那些僅提供非插斷式(nonpreemptive)的作業系統(如結合DOS和WINDOWS 3.x)。此外，這樣的作業系統亦不適用於即時的應用，因爲採用非插斷式(nonpreemptive)的排程策略並無法保證使用者可以在一定的時間內得到回應。而且DOS和WINDOWS 3.x對於系統核心(kernel)的保護並不完善，因此時常會發生莫名其妙的當機，這對於一個即時的CSCW應用將是一個嚴重的致命傷。因此在Client端的作業系統平台，將選擇採用插斷式(preemptive)排程策略的作業系統，例如：OS/2, Windows NT, UnixWare, SCO Open Desktop。這些新一代的32位元作業系統，不僅可以在現今32位元以及未來64位元的個人電腦上提供真正的多工作業能力，並且擁有完整的網路功能及提供TCP/IP的支援(但目前的UnixWare僅伺服器版有提供TCP/IP支援，個人版則尚未提供)。另外，這些作業系統對於系統核心的保護較為完善，使用時比較不易產生當機，因此較適合用來建置CSCW

環境，在Server端則採用UNIX或SUN OS的作業系統平台，而其間的通訊協定則採用TCP/IP。採用的原因是TCP/IP已逐漸成爲網際網路的標準，在大部份的系統平台上均有支援，這對一個支援分散在各地群組成員工作的CSCW環境相當重要。

## 陸、未來研究方向

本文先針對適用於CSCW環境的協同控制機制提出一個概念模式(conceptual model)，未來的工作便是將其實際的建置，然後驗證其可行性。然後再考慮採用其他的協同控制方法(例如：不採用鎖定的方法改用其他方法，或資料採用複製的方式)，並和此方法作績效和協同性的比較，找出最適合用於CSCW環境的協同控制機制。接著便朝上圖一所示的三層架構的上二層發展，逐步建立一個完整的CSCW環境。

## 柒、結論

以往大部份電腦的應用侷限於結構化的資料處理，且過去大部份的軟體系統是針對支援使用者和系統之間的互動來設計，對於使用者與使用者之間互動的支援則相當的有限。然而，在現實世界中有很多半結構化(semi-structured)和非結構化(un-structured)的問題，必須透過群體討論的方式，針對問題彼此間進行相互溝通和協調，共同努力來找出解決方案。因此一個能提供群體間互動支援的系統環境，對於現今日益複雜的環境以及許多需要群體智慧的應用更顯得其重要性。

對於一個多使用者的資訊共享環境中，要使得群體合作能順利進行必須要能有效的協調控制成員們的動作，以避免衝突及重覆的動作以增進效率。爲達此目的便須要有一個協同控制機制，來協調所有同作的異動不會相互干擾，以增進CSCW系統的協同程度(degree of concurrency)和確保CSCW環境中資料的一致性。此外，在一個用於群體合作環境中，尚必須要將最新的資訊傳送給群體成員，使得成員間能即時了解彼此的想法，促使群體合作的活動順利進行。

由於同樣是處理協同控制的問題，因此許多即時系統和多使用者資料庫(Multiuser Database)所採用的協同控制的概念和方法，同樣適用於CSCW環境中來解決有關協同控制的問題。但必須做適當的修改和擴充才能應用於CSCW環境中，針對CSCW環境的特性，本文提出一個修正後的協同控制模式使其能運用在CSCW環境中。

本論文所闡述之協同控制模式，其優點在於使用極為便利，可在短期內用於台灣學術網路(TANet)上以支援與CSCW相關工作環境的發展。這一個雛形模式可在相關的技術發展中再繼續作校驗修正(Validation)，並可作為後續類似之協同控制機制發展之參考基準(Benchmark)。

致謝：本文在撰寫期間，非常感謝中研院計算中心林誠謙主任給予許多珍貴的意見，及政大計算機中心劉文卿主任的鼓勵，使得本模式的建立能更完善。

## 捌、參考資料

- [1] Rakesh Agrawal, M. J. Carey and M. Livny: "Concurrency Control Performance Modeling: Alternatives and Implications", ACM Transactions on Database System, Vol.12, No.4, 1987, pp.609-654.
- [2] Greif, I. & Sarin, S.: "Data Sharing in Group Work": Computer-Supported Cooperative Work: A book of Readings, Morgan Kaufmann Publishers, 1988, pp.477-508.
- [3] Chen, Shyuan-Ling: "Granularity of locks in object-Oriented Database System": Institute of Computer Science and Information Engineering, National Chiao-Tung University, Master Thesis, 1989.
- [4] Eils, C. A., Gibbs, S. J. and Rein, G. L.: "GROUPWARE: Some Issues and Experiences", Communications of the ACM, Vol.34, No.1, Jan 1991, pp.39-58.
- [5] 李怡君, "即時性物件導向資料庫的協同控制", 國立交通大學資訊管理研究所, 碩士論文, 民82.
- [6] Chiang, J. K. (姜國輝): "Entwicklung einer ganzheitlichen Qualitaetsplanungssystematik auf der Basis von Hypertext-Techniken und 7-QM-Tools" (開發一個以超文字及新七項品質管理工具為基底之全面品質規畫系統), Annual of DFG Advanced Research Program '93, FQS-Schrift 96-04, Forschungsgemeinschaft Qualitaetsicherung e.V. (FQS), Frankfurt am Main, March 1994, pp.95-104.
- [7] Chiang, J. K. (姜國輝): "Qualitaetsplanungssystematik auf der Basis von Hypertext-Techniken - Ein Beitrag zur rechnerunterstuetzten Teamwork in der ganzheitlichen Qualitaetsplanung" (以超文字為基底之品質規畫系統 - 全面品質管中之電腦輔助群體工作), Verlag Mainz, Wissenschaftsverlag, Aachen, Germany, August 1994, ISBN 3-930085-90-9.