

公告本

A4
C4

527545

申請日期	90.11.27
案 號	90120351
類 別	G06F 15/16 H04L 1/00

(以上各欄由本局填註)

發 明 專 利 說 明 書

一、發明 名稱	中 文	建構多個軟體及硬體之方法及系統
	英 文	A METHOD AND SYSTEM FOR SOFTWARE AND HARDWARE MULTIPLICITY
二、發明 創作人	姓 名	葛瑞格里 L. 金格瑞奇 Gregory L. Gingerich
	國 籍	美國
	住、居所	美國維吉尼亞州赫登林彎圓環 12939 號 12939 Wood Crescent Circle, Herndon, VA 20171, USA
三、申請人	姓 名 (名稱)	葛瑞格里 L. 金格瑞奇 Gregory L. Gingerich
	國 籍	美國
	住、居所 (事務所)	美國維吉尼亞州赫登林彎圓環 12939 號 12939 Wood Crescent Circle, Herndon, VA 20171, USA
	代 表 人 姓 名	

(由本局填寫)

承辦人代碼：
大類：
IPC分類：

A6
B6

本案已向：

國(地區) 申請專利，申請日期： 案號： 有 無主張優先權

本案已向美國申請專利；申請日：2000年11月28日 案號：09/722,321號

有關微生物已寄存於： 寄存日期： 寄存號碼：

(請先閱讀背面之注意事項再填寫本頁各欄)

裝

訂

線

經濟部智慧財產局員工消費合作社印製

五、發明說明()

發明領域：

本發明廣泛地揭露一種用來在處理器之網路上部署及執行軟體物件的方法與系統。本發明所揭露的技術均具體實施於電腦平台系統、產生電腦平台之方法、以及相關的電腦程式產品中。

發明背景：

傳統上大型軟體系統經常超過單一處理器或機器所能提供之計算能力的極限。因此，產生能夠在跨越許多處理器或機器上執行的軟體系統是件非常令人嚮往的事。傳統系統的重心大多在於環繞於硬體和作業系統程度的解決手段。然而，由於網際網路的廣泛使用及可利用性，加上到處存在的網路連線，其抽象的程度已向上提昇至以遠距離物件(Remote Objects)為主的系統之研究與發展。

儘管已有若干種遠距離物件的安裝方式，然而一般的安裝方式係需要例如：編組物件(Marshalling Object)；方法呼叫(Method Invocations)；以及位元組流(Byte Stream)中的資料屬性，藉以傳送至網路的各角落。然後，代表遠距離物件的位元組流便由其他機器除去它的編組，而“遙遠地”執行程式。一般安裝這種遠距離物件的技術包括：DCE、COBRA、XML、COM/DCOM、DOM、SOAP和RMI。這些技術可使網路上的機器以遠距離執行位於其他機器的程式碼。這依序開啟了產生可擴充系統的各種可能性。

實務上，前述系統之機器的數目(和由此而來的擴充

五、發明說明()

性)常常受限於在一打(12個)之內。基本上，這是由於在軟體階層(Software Layers)間的依附性、網路組件的“同步化”(Synchronization)、以及需要將系統負荷有效率地分散至這些機器。更複雜的是，如何適當地設定(Configure)數目龐大的系統，以及在所有的機器部署正確的軟體版本，均有其困難度。使用這種抽象層(遠距離物件)的習知系統，更加承受了缺乏開放、彈性和一致的自動化的缺點，其中此自動化係用以控制系統的資源、物件以及重複資訊(Redundancy)。

結果是，由於軟體和硬體的不當設定或部署，因而大多數習知安裝遠距離物件的系統均承受所造成的中斷現象。而額外的中斷現象(常被視為正常現象)則是發生在升級或更換軟體和硬體元件的時候。還有常利用下班(晚上/週末)這些被認為較不重要的時候，來進行系統的升級。然而，隨著計算越來越普遍，加上網際網路的重要性日益提升，對於顧客和供應商而言，無法再容忍這些中斷現象，如今期待系統的可使用度是二十四小時乘以七天。

發明目的及概述：

為要克服上述之習知技術的問題，本發明的目的在於提供一種解決辦法，藉以使系統隨時保持高度的可使用度，並在系統升級和維護時，不需暫停系統存取。

為要達到所揭露的目標，本發明提供一種方法，藉以在網路上產生可擴充並容忍誤差的計算平台，而此計算平

五、發明說明()

台至少包括複數個節點，且此方法至少包括：初始化時初始合併步驟，其中在此合併步驟進行時，這些複數個節點會被視為一個集合，而這些複數個節點中的每一個節點都加入此集合，且於合併時，至少一第一物件路由器(Router)會被指定；在這些複數個節點之間遷移物件；以及在跨越這些複數個節點之間複製物件。

較佳的是，於合併時，每一個節點加入集合所使用的程序至少包括：要是能發現鄰近的物件路由器，此節點就能找到此物件路由器，否則此節點會成為物件路由器；以及此節點向物件路由器報告可利用的資源。

較佳的是，於合併時，每一個節點加入集合所使用的程序至少包括：給此節點指定網路位址；以及更新此節點的組件。

較佳的是，可利用的資源至少包括記憶體和至少一個處理器。

仍是較佳的是，可利用的資源至少包括網路頻寬。

仍是較佳的是，可利用的資源至少包括驅動埠儲存空間。

仍是較佳的是，此更新步驟係利用儲存於程式碼版本伺服器之資料來進行。

仍是較佳的是，可以確定的是在任何時候物件路由器有至少一個鏡像物件存在。

較佳的是，物件路由器管理資源並平均分配負荷於複數個節點。

五、發明說明()

較佳的是，物件路由器在計算平台中維持所有節點中至少一次集合的位置之知識。

較佳的是，物件路由器重新設置或複製複數個節點中的物件。

較佳的是，進行此遷移物件的步驟所使用的程序至少包括：鎖定(Locking)物件；在複數個節點中之第一節點上轉換物件為位元組流；在網路中轉移(Transferring)位元組流；在複數個節點中之第二節點上，將位元組流轉換回物件；更新物件在物件路由器中的位置；以及將物件解開鎖定(Unlocking)。

較佳的是，此複製物件的步驟所使用的程序至少包括：鎖定物件於第一節點上；形成一個複製的物件；將複製的物件設於第二節點上；以及將物件解開鎖定。

仍是較佳的是，此方法更至少包括：訂閱(Subscribing)與物件有關的事件(Events)。

仍是較佳的是，此複製係藉由自物件產生位元組流和將位元組流轉換回複製物件來進行。

較佳的是，此計算平台容許演進(Evolution)步驟，其中此演進步驟係提供系統軟硬體的過渡時期，以可能完成無瑕縫之升級。

另一方面，本發明揭露執行演進步驟的方法，藉由執行此演進步驟使軟體無瑕縫的過渡，此方法至少包括：評估欲安裝或升級的軟體或硬體；在編譯的時候於計算平台中產生所有物件之相依階層樹狀結構(Dependency

(請先閱讀背面之注意事項再填寫本頁)

裝
訂

五、發明說明()

Hierarchy Tree)；於物件中至少一個類別(Class)的新舊版本之間執行物件類別的差異比較；針對有更改的每一個狀態變數(State Variable)產生一個轉換函數；選取一個物件，此物件為所選定物件類別之物件實體(Instance)；鎖定選取的物件；產生一個尚未初始化的物件實體，此物件係屬於所選取之物件的新版本；藉由應用適當的轉換函數於舊物件版本，以初始化新物件類別實體所有的變數；將所選取的物件解開鎖定；以及對物件類別的每一個物件實體重覆相關的步驟。

仍是較佳的是，此方法至少包括對計算平台中的每一個物件類別重覆相關的步驟。

仍是較佳的是，此方法至少包括部署客戶端(Client-Side)軟體。

仍是較佳的是，此方法至少包括：如果不需要物件的多個版本，銷毀物件的舊版本。

較佳的是，物件路由器維持每一個此物件的多個物件實體。

較佳的是，當來自第二節點的代理物件(Proxy Object)無法接觸到第一節點時，會從這些節點中偵測到此第一節點的故障。

較佳的是，當複數個節點中之一節點沒有回應物件路由器的中心請求(Heartbeat Request)時，會偵測到此節點的故障。

較佳的是，當複數個節點之一節點沒有向物件路由器

(請先閱讀背面之注意事項再填寫本頁)

裝

訂

五、發明說明()

報告狀態或統計數值時，會偵測到此節點的故障。

較佳的是，當複數個節點之一節點故障時，此節點會被稱為故障節點，而此至少一個物件路由器係使用程序來回應，且此程序至少包括：通知除此至少一物件路由器以外的其他物件路由器；於故障節點選取一物件；在此故障節點之外的其他節點製造複製物件，而此複製物件係相當於此選取物件；於計算平台中設置額外的計算資源；針對此複製物件產生新的複製本(Replicate)，並將這些新的複製本設置於新的複製本原來所在的節點以外之其他節點；以及將這些新的複製本之新位置通知此些其他物件路由器。

仍是較佳的是，此方法至少包括將離線(Off-Line)的故障節點通知系統管理員。

仍是較佳的是，此方法至少包括對於故障節點中所有物件重覆相關的步驟。

較佳的是，當此至少一個物件路由器中一物件路由器故障時，此故障的物件路由器就稱為故障物件路由器，而此計算平台係使用程序來回應，且此程序至少包括：利用此故障物件路由器以外之其他物件路由器的服務；複製此故障物件路由器，以產生一複製物件路由器；將複製物件路由器設於至少一個有功能的節點上；以及將複製物件路由器之新位置通知節點中至少一個次集合。

仍是較佳的是，此方法至少包括將複製物件路由器之新位置通知其他物件路由器之至少一個次集合。

五、發明說明()

較佳的是，此物件路由器察覺到其他物件路由器之至少一個次集合。

再一方面，本發明揭露至少包括複數個節點的網路計算平台，此網路計算平台至少包括：至少一個物件路由器，其中此物件路由器能夠管理資源和平均分配負荷於複數個節點，且此物件路由器在計算平台中維持所有節點中至少一個次集合位置的知識。

較佳的是，物件路由器將物件重新設置在複數個節點中。

較佳的是，物件路由器維持每一個物件的多個物件實體，而這些物件實體係部署在網路計算平台中。

較佳的是，這些節點之每一個節點至少包括：將複數個節點視為一個集合的物件合併器(Object Melder)，這物件合併器係用來執行合併，藉以產生此些複數個節點與此至少一個物件路由器。

較佳的是，這平台更至少包括物件版本管理器(Object Versioner)，係用來維持網路計算平台中所有物件之多個版本。

較佳的是，這平台更至少包括物件複製器(Object Replicator)，此物件複製器在與此物件所在之節點不同之節點，藉由複製物件以維持此物件之複製本。

較佳的是，這平台更至少包括物件遷移器(Object Migrator)，藉以將一個物件自一個節點遷移至其他節點。

較佳的是，這平台更至少包括資源監視器(Resource

(請先閱讀背面之注意事項再填寫本頁)

裝

訂

五、發明說明()

Monitor)。

較佳的是，這平台更至少包括資源控制器(Resource Controller)。

較佳的是，這平台更至少包括應用物件(Application Object)。

較佳的是，此些節點之每一個節點至少包括：類別裝載器(Class Loader)負責裝載物件的類別，其中類別裝載器容許相同類別的多個版本同時存在。

較佳的是，這平台更至少包括演進模組(Evolution Module)，其中演進模組容許網路計算平台執行演進步驟，而此演進步驟提供系統軟硬體的過渡時期，以可能完成無瑕縫之升級。

較佳的是，當來自第二節點的代理物件無法接觸到第一節點時，係藉由網路計算平台從此些節點中偵測到此第一節點的故障。

較佳的是，當一節點無法回應物件路由器的中心請求時，係藉由網路計算平台從這些節點中偵測到此節點的故障。

較佳的是，當一節點無法向物件路由器的報告狀態或統計數值時，係藉由網路計算平台從這些節點中偵測到此節點的故障。

較佳的是，當複數個節點之一節點故障時，此節點會被稱為故障節點，而此至少一個物件路由器能通知此至少一個物件路由器以外的其他物件路由器。

五、發明說明()

較佳的是，當複數個節點之一節點故障時，此節點會被稱為故障節點，而此至少一個物件路由器能通知其他節點。

較佳的是，當複數個節點之一節點故障時，此節點會被稱為故障節點，而此至少一個物件路由器能製造複製物件，這複製物件係相當於位在此故障節點以外之其他節點中之物件的至少一個次集合。

較佳的是，當複數個節點之一個節點故障時，此節點會被稱為故障節點，而此至少一個物件路由器能於網路計算平台中設置額外的計算資源，並針對此複製物件產生新的複製本，且將這些新的複製本設於這些新的複製本原來所在的節點以外之其他節點中。

仍是較佳的是，這平台能將這些新的複製本之新位置通知這些其他物件路由器。

仍是較佳的是，這平台能將離線的故障節點通知系統管理員。

較佳的是，當此至少一個物件路由器之一物件路由器故障時，這平台能利用此故障物件路由器以外之其他物件路由器的服務。

較佳的是，當此至少一個物件路由器之一物件路由器故障時，這平台能複製故障物件路由器以產生一複製物件路由器，並將複製物件路由器設置於至少一個有功能的節點。

仍是較佳的是，這平台更能將複製物件路由器之新位

(請先閱讀背面之注意事項再填寫本頁)

裝
訂

五、發明說明()

置通知位於節點中至少一個次集合。

較佳的是，此物件路由器察覺到其他物件路由器之至少一個次集合。

再一方面，本發明揭露針對至少包括複數個節點的網路計算平台之物件路由器，其中此物件路由器能夠管理資源和平均分配負荷於複數個節點，且此物件路由器維持在計算平台中所有節點之至少一個次集合之位置的知識。

較佳的是，物件路由器在複數個節點中重新設置物件。

較佳的是，物件路由器維持每一個物件的多個物件實體，而每一個物件係部署在平台中。

較佳的是，當複數個節點之一節點故障時，物件路由器能通知此物件路由器以外的其他物件路由器。

較佳的是，當複數個節點之一節點故障時，而物件路由器能製造複製物件，這複製物件係相當於物件的至少一個次集合，其中物件係位在此故障節點以外之其他節點中。

較佳的是，當複數個節點之一節點故障時，物件路由器能於計算平台中設置額外的計算資源，並針對複製物件產生新的複製本，且將這些新的複製本設於這些新的複製本原來所在的節點以外之其他節點中。

較佳的是，此物件路由器察覺到其他物件路由器之至少一個次集合。

本發明亦包括電腦程式產品，這電腦程式產品包括有

(請先閱讀背面之注意事項再填寫本頁)

裝

訂

五、發明說明()

電腦可讀式媒介工具(Medium)，而這媒介工具至少包括可使電腦實施本發明所揭露之方法的指令(Instruction)。

圖式簡單說明：

本發明以上所列舉之目標與優點，將於往後之說明文字中輔以下列實施例之圖形作詳盡的闡述以突顯本發明。其中：

第 1(a)圖為繪示本發明所揭露之技術得以實施的一般性網路平台；

第 1(b)圖為繪示用來實施本發明所揭露之技術的平台；

第 2 圖為繪示使用本發明所揭露之平台的節點之架構；

第 3(a)圖至第 3(c)圖為繪示三種複製步驟的實施例；

第 4 圖為繪示一物件版本樹狀圖；

第 5(a)圖至第 5(b)圖為繪示嵌入物件改版(Embedded Object Versioning)管理的實施例；

第 6 圖為繪示物件轉換的實施例；

第 7 圖為繪示類別裝載器的連鎖結構的實施例；

第 8 圖為繪示裝載多個物件版本時之類別裝載器的階層結構的實施例；以及

第 9 圖為繪示系統使用本發明所揭露之技術的位置之實施例。

五、發明說明()

圖號對照說明：

- | | |
|--------------------|---------------|
| 1.1 網際網路 | 1.2 第一開道節點 |
| 1.3 處理節點 | |
| 1.4 第二開道節點/舊資料源 | |
| 1.5 多磁碟陣列 | 1.6 管理設備 |
| 2.1 開機實體程序/保全模組裝載器 | |
| 2.2 物件路由器模組 | 2.3 基本物件服務 |
| 7.1 初步類別裝載器 | 7.2 開機時間類別裝載器 |
| 7.3 一般物件服務 | 7.4 應用物件 |

發明詳細說明：

本說明書提供本發明所揭露之內容和技術的較佳實施例，藉以當軟體和硬體元件在自動設定或升級時，使軟體物件跨越許多機器能夠有效率的分佈和“尋徑”。

在此所描述的較佳實施例產生可高度擴充並容忍誤差的計算平台，而這計算平台可如多磁碟陣列(RAID；Redundant Array of Inexpensive Disks)為主之驅動埠陣列般容易維修。此處亦可以多電腦陣列(Redundant Array of Inexpensive Computers™；RAIC™)為參考。

使用本發明所揭露之技術後，本發明之較佳實施例具體呈現能允許動態的軟硬體重新設定(Reconfiguration)和成長之特性，而這特性係藉由導入下列若干獨特的特徵於平台中來達成：

- ◆ 合併：高度自動化執行機器/節點的設定。

五、發明說明()

- ◆ 自動安裝：自動裝載軟體並分配至各節點。
- ◆ 物件尋徑：物件在各節點間自動尋徑，藉以增加可擴充性和可靠度。
- ◆ 自動恢復(Recovery)：當節點或組件發生故障時，計算平台本能上可自我修復。
- ◆ 演進：藉由軟體改版和相依性的自動管理，計算平台可以無瑕縫地升級，而且對終端使用者的服務不會產生可注意到的中斷。

本發明之較佳實施例的運算動作可分為一些混合的程序。值得注意的是，如本發明所申請的專利範圍，這些混合的程序或步驟可分開或組合起來使用。本發明所揭露之技術範圍並不侷限於任何特別的組合。

第 1(a)圖與第 1(b)圖所示係為一典型的網路計算平台。複數個第一閘道節點(Gateway Nodes)1.2 係連接於實際網路 1.1。複數個處理節點 1.3 係連接於此些複數個第一閘道節點 1.2。複數個第二閘道節點 1.4 係連接於處理節點 1.3。舊資料源(Legacy Data Source)1.4、RAID 1.5 和管理設備 1.6 係連接於複數個第二閘道節點 1.4。第 1(b)圖為繪示用來實施本發明所揭露之技術的平台，其中物件路由器標示為 OR；物件標示為 OJ；以及複製器標示為“r”。

第 2 圖為繪示使用本發明所揭露之平台的節點之架構。當在 2.1 進行系統初始化/開機時，會發生所謂的合併的訊號處理。這合併的程序係由合併器來進行。在合併的

五、發明說明()

程序進行時，每一個節點會發現網路上出現的其他節點與組件，並告知網路其系統資源，這系統資源包括有處理能力、記憶體、儲存空間和網路性能。合併步驟亦包括指定物件路由器 2.2 來負責傳送物件存取的請求訊息至/從每一個節點。物件路由器監視所有節點的資源使用，並據此來分配請求訊息。物件路由器亦維持每一個物件(位於不同的節點上)的多個物件實體，藉以避免在網路中發生故障的單點(Single Points)。值得注意的是，於本發明所揭露之系統中，物件路由器可用來取代較早的遠距離物件查詢服務(包括但不侷限於如由物件管理集團—OMG[Object Management Group]所製作之 COS[Common Object Services] Naming 物件)。

為要處理網路中個別節點的停工或故障，在 2.3 進行所謂的遷移和複製的程序，可提供誤差容忍並且在各節點間有能力移動活躍的(執行中)物件。由物件遷移器來進行遷移，而由物件複製器來進行複製。在硬體升級時也會用到這些程序，此時個別節點是處於停工狀態而不能被使用。當個別節點無法使用時，顯然地系統層級和眾節點的大體系統仍保持供使用者高度使用的狀態。

為要容納軟體的更改，稱為演進的程序係利用演進之性能，並結合轉換函數與複雜的物件改版技術。如此可在保持系統於可供最終使用者使用之狀態的同時，實施軟體組件之無瑕縫的過渡階段。

以上所簡略描述之平台的組件與程序將於以下做更

五、發明說明()

進一步的討論。

第 1 節 合併

合併是個別節點加入較大之節點“集團”的程序。

在合併時，使用一種如網際網路通訊協定(Internet Protocol ; IP)之動態主機配址協定(Dynamic Host Configuration Protocol ; DHCP)針對每一個節點來指定 IP 或其他網路位址。

然後，包括模組/類別裝載器之核心組件的開機實體(Bootstrap)會出現。值得注意的是，在開機時，每一個節點的核心組件可由儲存在“程式碼版本伺服器(Code Version Server)”(如可進行資料交換之超文件傳送協定[Hyper Text Transfer Protocol ; HTTP]、檔案傳送協定[File Transfer Protocol ; FTP]或其他網路通訊協定伺服器)的資料來加以更新。

下一步是決定物件路由器的位置。在節點開機後，節點會試圖使用多重播送數據包(Multicast Packet)、廣播數據包(Broadcast Packet)或“熟知”(已預先設定)的查詢位址(如在網際網路通訊協定之區域名稱伺服器中)，以搜尋附近的物件路由器節點。如沒找到物件路由器的位置，節點便裝入“路由器模組”而成為物件路由器之節點。在本發明之較佳實施例中，確定任何時候都有至少一個物件路由器之“鏡像物件”存在。

然後，進行資源報告的步驟。在進行資源報告時，節點向其物件路由器報告其所能取得之軟硬體之資源。所報

五、發明說明()

告之統計數值包括可利用之記憶體、處理單位、磁碟機儲存空間、網路頻寬、已安裝之軟體以及軟體版本。

第 2 節 物件路由器

物件路由器記錄可利用之資源。物件路由器追蹤其管理之節點內的資源之利用。物件路由器負責平均分配負荷至其所管理之節點上。

物件路由器亦進行遠距離物件的追蹤。物件路由器維持其所管理的節點中之物件位置的知識。必要時，物件路由器負責重新安排物件至各節點。值得注意的是，位置可以包括但不侷限於網址、埠(Port)的號碼、領域名稱系統(Domain Name System; DNS)、一致資源定址器(Uniform Resource Locator; URL)、或在遠距離物件系統中特別指明的遠距離物件之參考資料。

第 3 節 資源監視/控制

本發明之較佳實施例係針對資源監視與控制。資源監視係由資源監視器模組來進行。資源控制係由節點資源控制器來進行。物件路由器利用這兩個模組來監視與控制個別節點的狀態。

在運用爪哇(JAVA)語言的實施中，這些模組可以爪哇原始介面(JAVA Native Interface; JNI)來實施。在這個例子中，建立了用來存取作業系統核心結構(Kernel Structure)的標準化介面，其中此核心結構係例如程序識別代號、執行緒(Threads)、記憶體使用率、網路活動、磁碟活動和相關的統計數值。然後這些模組可以訂製之 JNI 包裝物件

五、發明說明()

(Wrapper Object)來實施。

在這以爪哇語言為基礎的應用中，JNI 包裝物件亦提供了一個機制，來控制程序的優先次序、執行緒、作業系統設定和系統重新開機。如此之包裝物件使這些運算在各種作業系統中一致化。

第 4 節 遷移

遷移是將物件自一節點移至其他節點的程序，係由物件遷移器來進行遷移。

首先，鎖定物件，藉以避免在遷移物件時產生不一致的狀態。

然後，使物件連續化(Serialization)。在這個步驟中，將物件轉換成位元組流，藉以在網路中傳送。

然後，重新設置物件。在這個步驟中，在可利用的節點上將位元組流轉換回物件。

之後，將物件解開鎖定。更新這物件在物件路由器中的位置並使用這物件。

第 5 節 複製

複製是在多個節點中複製物件的程序，係由物件複製器來進行複製。第 3(a)圖至第(c)圖為繪示三種物件複製的實施例。

第 5-1 節 模式 1

第 3(a)圖為繪示所謂的“模式 1”複製之例子，物件的客戶可藉此開始運算。客戶請求的訊息會來自位於其他節點、伺服器或軟體應用的物件。將這運算(或方法呼求)

(請先閱讀背面之注意事項再填寫本頁)

裝

訂

五、發明說明()

送至原來節點中之區域性的代理物件，(從物件路由器中獲得這些遠距離複製物件的位置之後)，代理物件會將此項請求送到位於節點 A 中之第一(或原來的)遠距離物件。節點 A 立刻將同樣已編組之運算轉交給位於節點 B 中之複製物件，然後在節點 A 中之原來的物件繼續執行此運算，並依請求修改其狀態。節點 B 中之複製物件同時執行同樣的運算，並將任何結果組(返回參數)送回到原來節點的代理物件。視情況而定，在兩個遠距離節點之結果組、物件的內部狀態變數或綜合前述二者中，計算檢查總和值(Checksum Value)。檢查總和的步驟係用來偵測錯誤並改正所發生的誤差。

第 3(a)圖所示之複製模式的優點為將節點間的網路交通減至最小，並減少原來節點發生網路阻塞的可能性。這種複製模式也“保證”(至某種程度)所有的複製本與每一個運算動作均保持“同步”。不過由於節點 A 至節點 B 尚需要額外的網路“跳躍”(hop)，因此潛在的信息會增加這種方式可能的負面效果。

值得注意的是，這種方式可藉由在節點 A 和節點 B 的連鎖結構中加入額外的節點而輕易的加以擴充，且每加入一個額外的節點/複製本只有些微的潛在損失。

第 5-2 節 模式 2

第 3(b)圖所繪示為第二個例子，即所謂的“模式 2”複製的例子，物件的客戶藉此來開始運算(這客戶請求的訊息可來自位於其他節點、伺服器或軟體應用的物件)。

五、發明說明()

將這運算(或方法呼求)送至原來節點中之區域性的代理物件，(從物件路由器中獲得這些遠距離複製物件的位置之後)，代理物件會將此項請求同時寄送至位於節點 A 和節點 B 中之二個遠距離物件。在節點 A 和節點 B 上的遠距離複製物件以平行的方式執行運算，而且“原來”遠距離物件會負責將回應送至位於原來節點中之代理物件。

視情況而定，可將結果組、內部狀態變數或二者的檢查總和值，送至區域性的代理物件，或者在遠距離物件間當作是“交互檢查”(Cross-Check)。檢查總和值係用來偵測錯誤並確保一致性。

值得注意的是，這種方式可藉由加入與代理物件通訊之額外的節點而輕易的加以擴充。然而，可以預期會增加網路頻寬的利用。這種方式減少預期的信息潛在因素，但增加整體網路的利用。

第 5-3 節 模式 3

如第 3(c)圖所示，第三種技術牽涉到直接與原來遠距離物件連接的代理物件，原來遠距離物件的狀態變化係與一或多個次級遠距離物件互為鏡像物件，其中次級遠距離物件已“訂閱”反映原來物件內之狀態變化的事件。再者，加入檢查總和值的信息可藉以改善系統的堅實度。

模式 1 至 3 的三種技術全都利用區域性的代理物件，以將應用程式設計師與複製程序的複雜度隔開。這代理物件當作遠距離物件的區域代理者，藉由代理物件和物件路由器一起工作來處理這些程式碼，以減少應用程式師撰寫

五、發明說明()

網路、檢查總和值和錯誤恢復的程式碼之需求。

在實務上，可利用一或多個這些技術來維持遠距離物件間的複製狀態，而混合的方式亦可行。

在所有的模式中，下列的運算是共同的：

物件鎖定：鎖定物件來避免狀態不一致性發生。

物件翻製(Cloned)：當物件經過連續化後便被複製。

物件重新設置：將複製的物件遠距離地設置在次級節點上。

代理物件/事件之訂閱：複製物件的運算。

物件解開鎖定：解開物件的鎖定使其可供使用。

第 6 節 演進：

在較佳實施例的平台中，演進是較獨特與複雜的程序之一。演進係依賴其他特徵的組合來完成，這些特徵包括物件遷移、先進的改版、專門的類別裝載器，以及轉換函數。

第 6-1 節 編譯/前-後處理器：

這裡是指在編譯前/中、或編譯後執行的程序。在原始碼或物件碼中執行額外的步驟，以插入演進時(在軟體部署以前)所需額外的改版資訊。這個步驟通常是由軟體開發者、軟體系統整合者或經銷商來執行。以下敘述幾種方式：

a. 編譯前

在對原始碼執行編譯器之前，可執行前處理器來對原始碼做語法分析，並在儲存版本資訊的每一個類別中

五、發明說明()

插入額外的(原始)程式碼，其中每個類別儲存了資訊，此資訊係反映出每個類別使用的每個外部類別(External Class)所需之版本。舉例來說，類別”A”依賴類別”B”、”C”與”D”。在編譯前，可於類別”A”插入程式碼，而此程式碼指出所需之類別”B”、”C”與”D”之版本，還有類別”A”之版本(很可能是類別定義之計算的拼湊資料)。這種程序的型式有時稱為“原始碼插入”(Source Code Insertion)。

b. 編譯中

在編譯程序中，專門的原始碼編譯器可建構成自動插入改版資訊，類似於第 5(a)圖所示。對進行編譯中的每一個物件而言，編譯器的”外部符號表”含有與正進行編譯中的類別相關之每一個外部類別或方法的參考資料。擴充後編譯器的設計可於外部符號表中加入版本資訊，並且為每一個類別將這資訊儲存於所產生的目的碼中。這是語言編譯器的一種擴充或修訂版。

c. 編譯後

在編譯並插入含有改版資訊之位元組碼後，有可能可以讀/處理爪哇物件位元組碼(JAVA Object Bytecode)(或其他產生的物件碼)。一般係以第 5(a)圖的“嵌入物件改版”圖來表示。一般係藉由軟體開發者或獨立的軟體經銷商(Independent Software Vendor; ISV)來執行用於這項軟體工具的程序。這項方法最大的個別優勢在於第三者軟體的原始碼無法取得時，可以將這個方法應用在此第三者軟體。軟體沒有可用的原始碼時，前述之方法 1 與 2 是行不

五、發明說明()

通的。這裡指的就是“物件碼插入”(Object Code Insertion)。

值得注意的是爪哇物件位元組碼大體上等同於由C/C++軟體編譯所產生的組合/機器碼(Assembly/Machine Code)。類似的方法論可以應用在非爪哇為主的系統中。

另一種直接在物件類別中儲存改版資訊的方法(此為目前顯示較有利的方法)，係將這項資訊儲存於外部，作為儲存在這(些)節點中查詢用的表格，或在一些供替換的資料伺服器或節點上使這項資訊可供利用。在類別裝載時，每處節點的類別裝載器就可取出這項資訊，而不需直接成為物件的一部分。

關於所有方面：

位於每個節點中之專門的類別裝載器，係依靠“擴充的改版”資訊，在每個類別於裝載相依樹狀結構所需之各物件之適合的版本可利用這項資訊。當節點的類別裝載器裝載類別“A”之第“1”版，就知道必需要取得/裝載類別“B”之第“2”版以及類別“C”之第“3”版。

還有，雖然在現今非爪哇為主的類別裝載器並不常見，不過在任何以物件為主的語言，理論上可以產生這樣的架構(即C++的類別裝載器或Oracle™資料庫類別裝載器)。編譯過的類別之版本數，會隨著外部類別的版本數儲存起來。

雖然版本數是多變的[也可以利用例如來自設定管理產品像是ClearCase、SourceSafe、並行版本系統

五、發明說明()

(Concurrent Version System ; CVS)、或 Continuous 的標籤]，但利用每個類別計算的拼湊值，是產生版本數的較佳方法(因為具有獨特性)。這項程序可用於原始碼儲存/設定管理工具之連接--像是上述所指之產品。

第 6-2 節 版本階層：

第 5(a)圖所顯示的是與物件/類別相關之版本資訊的實施例，其所繪示的是物件類別版本之間的關係。在本圖中所說明的版本數是多變的。每個物件類別會指定一個“版本序號”(VersionID)，不過也可以將版本表格在外部儲存到類別中。

版本階層係以下列任何一種方式來形成：

- a) 執行原始碼前處理器(Pre-Processor)；
- b) 執行“修改過”之編譯器(Compiler)；
- c) 執行物件碼後處理器(Post-Processor)。

版本階層正常情況係“暗示儲存”在物件內之運作時間/部署時間(Runtime/Deployment Time)中：

意即：物件類別“A”知道它是依據“B”與“C”；

物件類別“B”知道它是依據類別“D”；

物件類別“C”知道它是依據類別“E”。

這暗示形成這樣的階層：

A → B → D

A → C → E

或是，以另一種方式繪示：

五、發明說明()

A

B C

D E

這種樹狀結構可無限擴充--其中樹狀結構之“終端”節點(理想上)為作業系統及/或硬體組件識別器。(在硬體階層中缺乏可用的資訊時，部署程序會盡“最大努力”的方法以部署並企圖尋求接近的可用方式。)

舉例來說，物件類別“C”係依據物件類別“E”，而物件類別“E”係依據爪哇之爪哇執行環境(Java Runtime Environment; JRE)1.2，而爪哇 JRE1.2 係依據 Solaris 8、OS 增訂版 482、200MB 驅動埠儲存器以及 100MHz SPARC() 晶片版本“E”或更新版，其中 OS 增訂版 482 係依 128MB 隨機存取記憶體(Random Access Memory; RAM)。

在傳統的系統中，編譯器的外面通常並不利用物件相依性，而物件相依性也不會擴充到超過語言的邊界。在編譯之後，這樣的資訊一般就不用了。本發明揭露的系統透過物件的部署與運作時間，來儲存並利用物件相依性與改版資訊。這能加強軟體物件之追蹤與轉換。第 4 圖顯示的是物件類別版本的樹狀圖。第 4 圖中的“v”代表版本數。第 4 圖中的物件類別名稱是多變的，而此第 4 圖也可以說明電腦語言的版本、作業系統的版本、以及硬體版本(即“爪哇”、“Linux”或“英特爾 Pentium” [Intel Pentium] 可以取代“高度[Altitude]”)。為了形成版本樹狀階層，係依據下列的資訊來源：

五、發明說明()

- a) 確保在物件與組件之間能加以部署方法-識別的相容性之編譯器(或類似編譯器者)的演算法。
- b) 程式設計師具體指明、或設定管理工具具體指明版本相依性(CVS、Continuous、ClearCase、SourceSafe等)。
- c) 由第三者軟體經銷商具體指明的資訊：
即：Oracle、Sun、國際商務機器公司(International Business Machines；IBM)、及其他經銷商具體指明可以部署它們的軟體之平台，而且通常會以“發佈須知”(Release Notes)之文件來提供這項資訊。
- d) 資訊系統製造者(或其他外來者)透過其他手段集合並增加至相容設定的儲存器中。

第 6-3 節 版本查詢用表格

第 5(b)圖提供如何將版本表格與物件類別分開儲存的說明。這項資訊可以儲存在一或多個節點中，或者儲存在外部伺服器或資料庫中。查詢用表格(Lookup Table)存在每個物件中，係由物件所需之所有物件的表列，及每個物件裝載適用之版本所組成的。“版本查詢用表格”應能從階層及前處理器的小部分所提供的資訊明顯看出。查詢用表格是專門的類別及其所依據的類別版本之間的一組關係。查詢用表格存在於每個物件中，係由物件所需之所有物件的表列，及每個物件所需之適用版本所組成的。查詢用表格是專門的類別及其所依據的類別版本之間的一組關係。這可以下列其中一種方式儲存：

五、發明說明()

a)在每個被部署的物件中；

b)在一或多個節點中。

物件裝載器係利用這些關係，來確定物件裝載在節點中的相依性。

第 6-4 節 自訂的物件裝載器

專門的物件裝載器係用於容許同時裝載多個物件版本。傳統的系統(及其物件裝載器)在任何時間只能容許裝載一個物件類別之單一版本。

第 6-5 節 轉換函數 $f(x)$ /程序：

第 6 圖所繪示為用於類別“員工”之物件時，物件/類別轉換函數之映像狀態。值得注意的是複數個轉換函數可以結合為單一轉換函數，或者再細分為更多函數。在此第 6 圖中，V1 代表版本 1 而 V2 代表版本 2。

如第 6 圖所示，每個物件實體係以下列步驟來執行：

- 1)鎖定位物件
- 2)建立版本 2(v2)之代理物件/實體
- 3)應用轉換函數
- 4)訂閱變更事件
- 5)解開鎖定位物件

針對每個物件的新版本，建立一系列的轉換函數。轉換函數的目的是要將物件從先前版本轉移到目前的版本。對於許多共同的轉換函數，可存在前定義(Pre-Defined)的函數。對於比較困難的轉換函數，則需要有程式設計師的介入。

五、發明說明()

第 6-6 節 演進的較佳實施例

普遍的演進程序之較佳實施例在此描述。

1) 評估要安裝或升級軟體

- a. 在編譯時，建立所有物件之版本階層樹狀圖。
- b. 對於每個物件的類別，在舊的類別版本與新建立的版本之間，執行物件類別“差異性”的比較。
- c. 針對已經改變的每個狀態變數或方法識別，建立轉換函數。
- d. 在部署時，對於每個即將升級/安裝的物件類別，進行：
 - i. 圈出每個類別的所有物件實體：
 1. 鎖定物件以避免狀態改變。
 2. 建立所需之新物件類別版本、對應之代理物件與複製本之未起始化的實體。
 3. 把轉換函數應用在先前物件版本中，以起始化新物件類別實體的變數。
 - 4 訂閱要變更事件。
 5. 解開鎖定物件。
- e. 在完成之後，會有原來系統的“鏡像物件”複本，其中包含所有物件的新版本。此刻，新的“客戶端”軟體就部署好了。

(請先閱讀背面之注意事項再填寫本頁)

裝

訂

五、發明說明()

f. 在新的“客戶端”軟體完成之後，物件的先前版本就會被處理掉。另一種方式，可以保留多個版本，不過值得注意的是所使用資源的成本。

較佳實施例於平台內的物件中加上複數個擴充部分(Extensions)，以促進遷移及複製所需的機制。此處描述於較佳實施例中的物件支援下列“基本類別”(Base Class)擴充部分：

- ◆ 物件鎖定：物件支援分散式的讀及寫之鎖定的機制。
- ◆ 改變事件：當物件中發生狀態改變，就會產生改變事件。可至各節點遠距離訂閱這些變更事件。
- ◆ 物件的改版：不論是藉由介面計算的拼湊值，或者是開發者或設定管理工具所指定的數值，將物件指定一個版本序號。此外，物件知道所使用的物件版本序號。
- ◆ 可連續的物件：遠距離存取的物件可以轉換為位元組流以透過網路傳輸。
- ◆ 物件擴充部分：藉由物件參考值，物件之運作時間可以加以擴充。

第 6-7 節 硬體的演進

硬體的演進係藉由與軟體稍微不同(雖然相關)的方式來處理：

在升級或維護硬體的一部分時：

五、發明說明()

- a) 升級、替換、或維護節點的“狀態”，以使節點的“狀態”變成反映出“下降”、“離線”、或“更新中”的狀態，並在物件路由器中管理此狀態。
- b) 在節點上要進行更新的物件係“尋徑”到替代的節點。(與故障情形相似)。
- c) 可透過任何必需之程序進行硬體的替換、修改、或升級(意即開啟外殼替換記憶體、驅動埠等)。
- d) 復原硬體並使其回復連線。
- e) 硬體接受“合併”步驟以(再)加入節點之“集合”。

下列各段係敘述硬體演進是如何與軟體演進相關：

於已知的節點上更新軟體，也需要硬體的更新，這在“相依階層”中是可能的(物件路由器已預先詳細了解每個節點的硬體性能)。

例如，應用軟體以最新版爪哇為基礎，節點會需要額外的記憶體或新的音效卡(Audio Card)來支援最新版爪哇。揭露的系統之相依樹狀圖會比“軟體物件版本”來得深入，並可擴充至語言版本、運作時間圖書館版本、以及作業系統規定(Operating System Requirement)。作業系統特徵係依序來依靠硬體的可利用性。這意味在部署時間中，軟體的一部分會通知安裝器(Installer)，需要用來支援軟體之硬體的是無法利用的。接著有必要為了支援軟體的轉變而使硬體升級(如上述所指)。

第 7 節 類別裝載器改變：

五、發明說明()

類別裝載器係在某些物件導向(Object-Oriented)語言中用來裝載物件類別至記憶體中之機制。在這樣的系統中用來改變類別裝載之行為，其常見的技術有時指的是“類別裝載器連鎖結構”(Class Loader Chaining)。利用這項技術，可將改變的行為提供給裝載於系統中各種類別裡。第7圖所繪示為類別裝載器連鎖結構的例子。例如，“初步類別裝載器”(Primordial Class Loader)7.1 能用於裝載語言特定的物件，以及允許這些物件具有最大的權限以存取主機系統的檔案、記憶體、及網路。與初步類別裝載器“連鎖”的子類別裝載器(Sub-Class Loader)其表現則有所不同。例如，“開機時間類別裝載器”7.2 能夠增加儲存於遠距離伺服器的裝載類別之性能。與應用物件7.4相關的類別裝載器可在那些物件中設定保全限制，以禁止它們存取區域機器的檔案。標準類別裝載器“連鎖”的特殊限制，係無法提供處理單一物件類別之多個版本的機制。

此處所敘述的系統藉由創造類別裝載器階層結構(Class Loader Hierarchy)取代連鎖結構，解決了這個問題。第8圖為繪示裝載多個物件版本時之類別裝載器的階層結構的例子。利用類別裝載器的階層結構容許揭露之系統中的主機節點同時裝載同一物件類別的多個版本。

此項技術在結合其他已揭露的技術後，可在軟體版本之間容許無瑕縫的過渡。利用傳統的裝載器技術，使用者必須終止軟體程序並重新開機，以裝載已儲存於記憶體中之類別的新版本。

五、發明說明()

第 7-1 節 實施例：

使用者或系統開始軟體的程序，並裝載類別“卡車”(Truck)、版本 1：

```

類別卡車 {
    靜態字串版本序號 = 1 ;
    公用空的啟動引擎 () {
    }
    公用空的移動快速器 () {
    }
}

```

之後，類別“卡車”的定義就轉變了，而且產生版本 2，即如下所示：

```

類別卡車 {
    靜態字串版本序號 = 2 ;
    公用空的暖光插頭 () {
    }
    公用空的啟動引擎 () {
    }
    公用空的加速器 () {
    }
}

```

正常情況，為了要裝載卡車的版本 2，會需要重新開機的程序。此外，裝載“卡車”的版本 2 會將類別“卡車”的版本 1 從記憶體中移出，並且中斷依賴版本 1 中的移動

(請先閱讀背面之注意事項再填寫本頁)

裝
訂

五、發明說明()

快速器()之方法使用的任何軟體。利用專門的類別裝載器之階層結構設計，版本 1 與版本 2 可以共同存在於程序中，而不必終止任何對客戶或其他依賴版本 1 或版本 2 之物件的服務。

第 8 節 JVM 之擴充：

下列數個擴充提供給爪哇虛擬機器(Java Virtual Machine；JVM)：

- ◆ 擴充的程序控制：應用程式介面(Application Program Interface；API)係提供追蹤程序資源利用、開始、改變優先順序、以及停止程序。
- ◆ 擴充的執行緒控制：API 係提供針對執行系統執行緒獲得較大的控制。
- ◆ 在 VM 中的虛擬程序：藉由執行緒/執行緒集團與專門的類別裝載器機制之結合，在 VM 中產生假程序。
- ◆ 專門支援改版的類別裝載器：在相同 VM 中同時裝載相同物件的多個版本時提供支援。

值得注意的並不是所有上述擴充的部分是絕對必須的，而是這些擴充部分提供更佳的系統運算。

第 9 節 具有多個物件路由器實體之物件路由器的故障

本揭露技術提供的是將故障的物件路由器恢復過來。假設有兩三個節點是作為物件路由器，則每個物件路由器與至少一個次級鏡像物件成對。在系統中每個物件路由器察覺(透過階層結構直接或間接的察覺)其他物件路由

五、發明說明()

器。則故障會以類似於本節所描述的方式偵測出來。關於故障時，可執行下列步驟：

- 1) 節點可利用次級物件路由器的服務(此時次級的物件路由器就變成主要的)。
- 2) 在至少一個次級物件路由器“翻製”(利用類似於前述第 5 節所述之物件複製的技術)並活化“複製”的物件路由器。
- 3) 將新的第二位置通知物件路由器所管理的所有節點。
- 4) 將新的第二位置通知其他的物件路由器。

除了上述步驟之外，可視情況進行下列步驟：

1. 將故障通知系統管理員。
2. 將故障通知其他系統。
3. 要求故障的物件路由器重新啟動成“恢復”模式。

第 10 節 只有一個物件路由器實體之物件路由器的故障

關於故障時(以第 9 節之所述偵測出來的)，可進行下列步驟：

- 1) 剩下的節點“選出”一個新的節點作為物件路由器(如果只有一個節點，此節點就同時執行應用處理以及路由器的功能)。
- 2) 選出作為物件路由器的節點啟動物件路由器的模組，並且從資料庫、平面檔案(Flat File)組、或其他資料伺服器中讀出先前路由器的最近已知的狀態。[倘若可能的話]

五、發明說明()

3)將可利用之新物件路由器廣為通知各節點。

4)新啟動的物件路由器要求各節點，它會直接管理下列：

a)包括系統容量的目前狀態，以及目前“裝載/利用”的統計資料。

b)存在於受管理的節點上目前物件的表列。

除了上述步驟之外，可視情況進行下列步驟：

將故障通知系統管理員。

將故障通知其他系統。

要求故障的物件路由器重新啟動成“恢復”模式。

第 11 節 節點故障

當揭露的系統中一節點發生故障，可以透過下列數種方法但不侷限於此，來偵測出這項故障：

1)來自另一個節點的“代理”物件不能抵達這個節點。這個故障報告至物件路由器。

2)節點無法回應來自物件路由器的“中心”請求。

3)節點無法在一定時間內或無法根據請求向物件路由器報告狀態或統計資料。

在偵測到任何節點發生故障，物件路由器係藉由下列方式回應：

1)將節點故障通知其他物件路由器及所管理的節點。

2)對於在故障的節點上的每個物件：

- 複製的物件(位於其他節點)變成活的/主要的。
- 物件路由器在系統中設置額外的計算來源。
- 物件路由器設置剩下的物件“複製本”並且在次

五、發明說明()

級節點上製作至少一個額外的“翻製本”/複製本。

- 將複製本之新位置通知其他物件路由器與節點。

3)將節點已故障並目前“離線”的情形通知系統管理員。

4)視情況而定，將節點已故障的情形通知其他系統。

5)如果可能的話，將系統節點重新啟動為“恢復”模式--試圖執行自行診斷及修復。(執行安裝的軟體、硬體以及其他相關的診斷工作的重新安裝/檢查總合)。

假定系統中每個物件隨時會正常地保有至少一個複製本。在較高利用率的設定中，物件在多個節點會有多個複製本，並會增加誤差忍受程度。“標準”的設定(具有至少一個複製本)在一小段時間變化中可忍受任何一個節點的故障。“高度重複”(High-Redundant)的設定(具有多個複製本)同時可忍受多個節點故障。

第 12 節 加密(Encryption)

在揭露的系統中，在節點間傳送所有的訊息與網路交通會利用技術進行加密，這些技術包括但不限於：公開與私密金鑰(Key)加密、數位憑證、數位簽名、保全訊息摘要、以及共享私密金鑰加密的資料。

此外，所利用的加密演算法與方法係包括但不限於：資料加密標準(DES)、三重 DES、區塊加解密器(BlowFish)、RSA 公開金鑰、SSL、TLS、PKCS、IMAP、LDAP、Kerberos、S/MIME、RC4、Diffie-Hellman、以及 DSA 公開金鑰。

五、發明說明()

有各種鑑定方案可以利用。這牽涉到種種數位簽名與生物統計的方法，係包括但不侷限於：PGP、LDAP、Kerberos、指紋掃瞄器、手印掃瞄器、視網膜掃瞄器、聲紋辨認、密碼、單次(One-Time)密碼、硬體為主的鑰產生器、電腦卡、植入元件、或前述之組合。這樣的鑑定方案可應用在系統、使用者、運輸工具、物件、或二位元的層級。

類似的加密與鑑定技術也可以應用在所有的訊息、檢查總合、以及在系統之組件間傳遞的網路交通，係包括但不侷限於合併器、物件路由器、遷移器、複製器、程式碼版本伺服器、類別裝載器、節點控制器、以及節點資源監視器。

再者，在揭露的系統中，傳遞給使用者之應用程式、最終使用者(End-User)、以及外部系統的訊息及網路交通，也會以類似的方式加密並鑑定。

熟悉加密技術之人員可以瞭解，其他硬體及軟體技術也可以輕易應用在所揭露的系統中。

第 13 節 最終使用者的好處：

最終使用者將揭露的技術與教示具體實施於計算平台，會經驗到種種的好處。第 9 圖為繪示系統將揭露之技術具體實施的位置之實施例。

為了揭露的目的，客戶系統可以是機器、電腦、元件、或者針對計算服務之最終使用者表現有使用者介面的軟體。

五、發明說明()

客戶-伺服器系統具有客戶系統，客戶系統與利用揭露的教示與技術的伺服器系統產生互動，而主機消費者服務會經驗到這些好處。由於硬體/軟體升級或故障是在伺服器系統內發生，所以客戶系統不會發生性能下降或故障。

同樣地，伺服器-伺服器系統具有伺服器系統“A”，伺服器系統“A”與利用揭露的教示與技術的伺服器系統“B”產生互動，而主機消費者服務會經驗到這些好處。由於硬體/軟體升級或機器故障是在伺服器系統“B”內發生，所以伺服器系統“A”不會發生性能下降或故障。

第 14 節 最終使用者的經驗：

具體實施揭露的教示與技術之較佳實施例，可以彈性部署在任何需要成長、穩定性以及最佳資源管理的系統中。從最終使用者的觀點而言，系統符合與例如水、電力、以及電話所提供的基本實用品相同標準的可靠性。

當軟體應用使用者(或“網路玩家”)遇到像是“找不到伺服器…”、“主機沒有回應…”、“系統無法使用…”或是包括伺服器沒有回應的暫停，他們會感到氣惱而且也許在放棄之前會再嘗試第二次。今日以網路運作之系統的消費者假定已經建構更好的技術，不過目前的系統還沒準備好提供這些必要的服務。不解決長期隱藏在提供使用者所依賴的中樞服務(Backbone Services)之計算環境中的問題，這些使用者所冀望的進展反而會轉向更痛苦的經驗。

每天我們看電視、聽廣播、讀報紙、以及講電話。這

五、發明說明()

些是我們視為理所當然之“基本”的活動中的一些，而且我們也要求一週 7 天、一天 24 小時都能使用。除了傳統的必需品(例如電話和電力)之外，我們逐漸朝向不斷發展資訊科技的使用方式，為更佳的生产力與娛樂帶來新的服務。像是透過網路之隨選影音(On-Demand Video And Audio)的服務很快就廣為接受，而且也期待這些服務能連續並確實的呈現出來。個人透過網路觀賞大碗公(SuperBowl)廣播節目，會期待事件能夠完美的傳送(包括商業廣告)。將揭露之技術具體實施的系統，透過多層重複資訊(Multi-Layer Redundancies)來保證完美的傳送。當越來越多的人透過網路訂閱所廣播的事件(估計約 1,000 至約 1,000,000)，只要不中斷服務，這樣的系統就能不斷擴大。對於現有的與新的使用者而言，他們就好像是享受這項服務之第一也是唯一的使用者。當新的特徵例如 3D 廣播節目成為可行的，依賴這項系統的使用者不用體驗目前服務的降級或必需等待系統擴充，可以選擇在系統中加入這項新的特徵。在主要的公司中，曾發生過服務中斷(Outage)與流通(Flux)等比較熟悉的例子，包括 Ebay、Hotmail、雅虎、以及美國線上(American On Line; AOL)，這些公司也已盡力提供快速成長與變化。利用揭露之技術的系統，為最終使用者提供真正 24 小時乘以 365 天的環境。在今天，針對計畫性的軟體及硬體維護之停工期，大多數生產系統都有“時間表”。即使硬體與軟體升級，具體實施本揭露技術與教示的系統對於使用者仍然是可利

五、發明說明()

用的。即使系統不知道軟體或硬體在進行升級的事實，最終使用者仍可以繼續與系統互動。這提供了史無前例的可利用性。具體實施本揭露技術的系統一項重要的特徵，是使用者在有關機器故障、替換及升級時，不會經歷停工期間。

第 15 節 消費者的好處

利用本揭露技術之系統的消費者是本系統的直接買主，消費者利用此系統可提供確實的服務給最終使用者。消費者很可能包括全球前 500 名公司、電子商務/網路公司、系統整合者、以及應用服務供應者(Application Service Providers; ASPs)。

消費者可經驗下述這些好處：

- 1) 可靠性、擴充性、有效性：此系統由於具有高度重複性/容忍誤差性，可輕易擴充以配合系統所增加的需求。
- 2) 便於使用：此系統簡化大量機器的管理。一旦系統中的節點裝載了必要的組件，系統就由它自己與物件路由器來接管。整個系統就可以由個別統一的介面來管理。
- 3) 擴增容量更新：系統容許以“隨插即用”(Plugging In)(合併)加入額外的節點來輕易升級的能力。
- 4) 故障恢復：系統可“四處搜尋”(Route Around)節點並自行修復(具備已知有效的硬體資源)。
- 5) 無瑕縫的軟硬體升級：系統於軟硬體升級時容許對現有的使用者保留 100%的可用性。

第 16 節 演進程序

五、發明說明()

此處是以最終使用者的觀點來描述演進程序。積極利用以演進為架構之應用的使用者，在系統演進時不會注意到系統性能中有任何的改變。在完成升級時，使用者會看見可以利用的新特徵/功能性。

第 16-1 節 實施例 1

隨選視訊系統的使用者正在選擇並觀賞新網站：FreeVid.com 上免費的影片。

FreeVid.com 了解目前他們處於營運虧損的狀況，而且針對他們的服務必須開始收費。只要使用者繼續選擇並觀賞影片，系統會演進而新的軟體也會部署為使用者需要輸入信用卡號。在系統完成升級，會呈現出螢幕並告知使用者壞消息，即提供使用者可免費觀賞 3 部影片的保證，並要求使用者於後續的服務需要付費。

第 16-2 節 實施例 2

付款系統的使用者藉由登入(Log In)並利用其服務來檢查他們的帳目。只要使用者保持連線，系統進行升級亦可接受信用卡付款。

第 17 節 從消費者的觀點

消費者購買具體實施揭露技術的系統，可執行此處所描述的任務。在消費者準備好升級他們的系統時，他們登入系統管理操縱台(Console)。從系統的操縱台中，他們可以選擇一個軟體模組或一組模組來進行升級/替換。接著他們按下“啟動”或“開始”的按鈕來開始升級。進度表(Progress Meter)會顯示升級進行的情況，在完成時新的特

五、發明說明()

徵組就可以使用了。然後管理器會“關閉”或拋棄不再需要的先前軟體模組。

第 18 節 從開發者的觀點

消費者建立軟體以部署在具體實施揭露技術的系統中，可執行此處所描述的任務。開發者會建構軟體模組，而此軟體模組遵守針對揭露系統之 APIs。他們會編譯他們的軟體。在編譯後，會由前處理器、語言編譯器或“後處理器”來分析類別/物件檔案以及插入改版資訊。於是在軟體的版本之間會呈現物件類別程序的“差異”。可能的話，系統會產生自動的轉換函數。在其他例子中，開發者必須從前定義中選擇或是產生自訂的“轉換”函數，使物件從一個版本遷移至下一個版本。最後，將修改後的類別/物件檔案與轉換函數，包裝成用來傳遞及部署的檔案格式。

第 19 節 電腦程式產品

除了系統及方法之外，電腦程式產品也包括在已揭露之教示的範疇中。這些電腦程式產品至少包括在電腦可讀式媒介工具上的指令，用來使電腦執行此處所揭露的方法。指令包括但不限於原始碼、物件碼以及執行程序(Executables)。執行指令的電腦包括但不限於微型電腦、微電腦、以及大型電腦(Mainframes)。電腦可讀式媒介工具包括但不限於軟碟、隨機存取記憶體、唯讀記憶體、硬碟、磁帶、卡匣(Cartridge)、光碟(CDs)、數位視訊影碟(DVDs)、以及網路下載的。

五、發明說明()

請留意的是可於選擇的網路系統中以任何方式實施所揭露的系統組件與程序。儘管以軟體實施為較佳，不過也可以藉由硬體或硬體/軟體之組合來實施。例如，可使用如 ASICs()之完全的硬體來實施組件或程序的次集合。同樣地，在軟體實施時，關於電腦語言的選擇或等級並沒有限制。例如，可以使用像爪哇、C++等之高階語言，或者使用包括組合及機器語言之低階語言。要不然，可以使用電腦語言的組合。雖然在此處所描述的較佳實施例中，有一些是以爪哇為主的實施來加以揭露，但這並不表示限於此。語言與實施的選擇係依據許多的因素，包括使用者的需求以及資源的可利用性。

同樣地，不應將相容性的問題視為限制。例如，請注意具體實施揭露教示與技術的系統可裝載在現有的網路系統上。這可能會引起一些相容性的問題。不過，這些相容性的問題不須脫離揭露教示之範圍，就可藉由熟練的工作者來加以克服。

如熟悉此技術之人員所瞭解的，以上所揭露的教示與技術僅為本發明之某些較佳實施例而已，並非用以限定本發明之申請專利範圍；凡其它未脫離本發明所揭示之精神下所完成之等效改變或修飾，均應包含在下述之申請專利範圍內。

四、中文發明摘要(發明之名稱：)

建構多個軟體及硬體之方法及系統

本發明揭露在網路上產生可擴充並容忍誤差之計算平台的方法，而此平台至少包括複數個節點。此方法至少包括：初始化時初始合併步驟，其中在此合併步驟進行時，這些複數個節點會被視為一個集合，而這些複數個節點中的每一個節點都加入此集合，且於合併時，會指定至少一個第一物件路由器(Router)；在這些複數個節點之間遷移物件；以及在跨越這些複數個節點之間複製物件。本發明亦揭露與前述之方法相類似的系統，以及用來實施前述之方法的電腦程式產品。

英文發明摘要(發明之名稱：)

A method and system for software and hardware multiplicity

Methods for creating a scalable, fault tolerant computing platform on a network, said platform comprising a plurality of nodes are disclosed. The method comprises initiating melding during initialization, wherein during said melding said plurality of nodes are treated as a collection and each node from said plurality of nodes joins said collection, and during melding at least a first object router is assigned; migrating objects between said plurality of nodes; and replicating objects across said plurality of nodes. Systems similar to the disclosed methods are also disclosed. Computer program products implementing the methods are also disclosed.

六、申請專利範圍

1. 一種產生可擴充並容忍誤差之計算平台的方法，該計算平台至少包括複數個節點，而該方法至少包括：

a) 初始化時初始一合併步驟，其中在該合併步驟進行時，該些節點會被視為一集合(Collection)，而該些節點中的每一節點都加入該集合，且於該合併步驟時，至少一第一物件路由器(Router)會被指定；

b) 在該些節點之間遷移(Migrating)複數個物件；以及

c) 在跨越該些節點之間複製(Replicating)該些物件。

2. 如申請專利範圍第 1 項所述之方法，其中在該合併步驟時，該每一節點都加入該集合係使用一程序至少包括：

(a)(i) 要是能發現鄰近的一物件路由器，該節點就能找到該物件路由器，否則該節點會成為該物件路由器；以及

(a)(ii) 該節點向該物件路由器報告複數個可利用的資源。

3. 如申請專利範圍第 1 項所述之方法，其中在該合併步驟時，該每一節點都加入該集合係使用一程序更至少包括：

(a)(iii) 給該節點指定一網路位址；以及

(a)(iv) 更新該節點的所有組件。

(請先閱讀背面之注意事項再填寫本頁)

裝
訂

六、申請專利範圍

- 4.如申請專利範圍第2項所述之方法，其中該些可利用的資源至少包括一記憶體以及至少一處理器。
- 5.如申請專利範圍第2項所述之方法，其中該些可利用的資源至少包括一網路頻寬。
- 6.如申請專利範圍第2項所述之方法，其中該些可利用的資源至少包括一驅動埠儲存空間(Drive Storage Space)。
- 7.如申請專利範圍第2項所述之方法，其中該更新步驟係利用儲存於一程式碼版本伺服器之資料來進行。
- 8.如申請專利範圍第2項所述之方法，其中係確定在任何時候該物件路由器有至少一鏡像物件(Mirror)存在。
- 9.如申請專利範圍第1項所述之方法，其中該物件路由器管理複數個資源並平均分配負荷於該些節點。
- 10.如申請專利範圍第1項所述之方法，其中該物件路由器在該計算平台中維持所有該些節點中至少一次集合(Subset)的位置之知識。
- 11.如申請專利範圍第1項所述之方法，其中該物件路由器

六、申請專利範圍

重新設置或複製該些節點中之該些物件。

12. 如申請專利範圍第 1 項所述之方法，其中該遷移步驟中一物件使用之一程序至少包括：

(b)(i) 鎖定 (Locking) 該物件；

(b)(ii) 在該些節點中之一第一節點上，將該物件轉換為一位元組流 (Byte Stream)；

(b)(iii) 在該網路中轉移 (Transferring) 該位元組流；

(b)(iv) 在該些節點中之一第二節點上，將該位元組流轉換回該物件；

(b)(v) 更新該物件在該物件路由器中之一位置；以及

(b)(vi) 將該物件解開鎖定 (Unlocking)。

13. 如申請專利範圍第 1 項所述之方法，其中該複製步驟中一物件使用之一程序至少包括：

(c)(i) 鎖定該物件於一第一節點上；

(c)(ii) 形成一複製的物件；

(c)(iii) 將該複製的物件設於一第二節點上；以及

(c)(iv) 將該物件解開鎖定。

14. 如申請專利範圍第 13 項所述之方法，更至少包括：

(c)(v) 訂閱 (Subscribing) 與該物件有關之複數個事件 (Events)。

(請先閱讀背面之注意事項再填寫本頁)

裝

訂

六、申請專利範圍

- 15.如申請專利範圍第13項所述之方法，其中該複製步驟係藉由自該物件產生一位元組流，並且將該位元組流轉換回該複製的物件來進行。
- 16.如申請專利範圍第1項所述之方法，其中該計算平台容許一演進(Evolution)步驟，而該演進步驟係提供一系統軟硬體之過渡時期，以可能完成無瑕縫之升級。
- 17.一種執行演進步驟的方法，藉由執行該演進步驟使軟體無瑕縫的過渡，至少包括：
- (a)評估欲安裝或升級之一軟體或一硬體；
 - (b)在編譯的時候於一計算平台中產生所有複數個物件之一相依階層樹狀結構(Dependency Hierarchy Tree)；
 - (c)於該些物件中至少一類別(Class)的一舊版本與一新版本之間，執行一物件類別的差異比較；
 - (d)針對有更改的每一狀態變數(State Variable)產生一轉換函數；
 - (e)選取一物件，被選取之該物件為一選取物件類別之一物件實體(Instance)；
 - (f)鎖定被選取之該物件；
 - (g)產生一未初始化物件實體，該未初始化物件實體係為被選取之該物件之該新版本；
 - (h)藉由應用適當之複數個轉換函數於該選取物件之

(請先閱讀背面之注意事項再填寫本頁)

裝

訂

六、申請專利範圍

該舊版本，以初始該選取物件類別之一新物件實體之複數個變數；

(i)將該選取物件解開鎖定；以及

(j)對該物件類別之每一該物件實體重覆該(e)步驟至該(i)步驟。

18.如申請專利範圍第17項所述之方法，其中更包括：

(l)對該計算平台中的每一該物件類別重覆該(e)步驟至該(j)步驟。

19.如申請專利範圍第17項所述之方法，其中更包括：

(m)部署一客戶端(Client-Side)軟體。

20.如申請專利範圍第17項所述之方法，其中更包括：

(n)如果不需要該些物件之多個版本，銷毀該些物件之該些舊版本。

21.如申請專利範圍第1項所述之方法，其中該物件路由器維持每一該些物件之多個物件實體。

22.如申請專利範圍第1項所述之方法，其中當來自一第二節點之一代理物件(Proxy Object)無法接觸到一第一節點時，會從該些節點中偵測到該第一節點之故障。

六、申請專利範圍

23. 如申請專利範圍第 1 項所述之方法，其中當該些節點中之一節點沒有回應該物件路由器的中心請求 (Heartbeat Request) 時，會偵測到該節點的故障。
24. 如申請專利範圍第 1 項所述之方法，其中當該些節點中之一節點沒有向該物件路由器報告狀態或統計數值時，會偵測到該節點的故障。
25. 如申請專利範圍第 1 項所述之方法，其中當該些節點之一節點故障時，該節點會被稱為一故障節點，而該至少一物件路由器係使用一程序來回應，且該程序至少包括：
- (i) 通知除了該至少一物件路由器以外之複數個其他物件路由器；
 - (ii) 於該故障節點選取一物件；
 - (iii) 在該故障節點之外的其他該些節點製造複數個複製物件，而該些複製物件係相當於被選取之該物件；
 - (iv) 於該計算平台中設置複數個額外的計算資源；
 - (v) 針對在該 (iii) 步驟中該些複製物件產生複數個新的複製本，並將該些新的複製本設置於該些新的複製本原來所在的節點以外之該些其他節點；以及
 - (vi) 將該些新的複製本之一新位置通知該些其他物件路由器。

六、申請專利範圍

26.如申請專利範圍第 25 項所述之方法，更至少包括：

(vii)將離線(Off-Line)之該故障節點通知一系統管理員。

27.如申請專利範圍第 25 項所述之方法，更至少包括對於該故障節點中所有該些物件重覆該(ii)步驟至該(vi)步驟。

28.如申請專利範圍第 1 項所述之方法，其中當該至少一物件路由器中一物件路由器故障時，故障的該物件路由器就稱為一故障物件路由器，而該計算平台係使用一程序來回應，且該程序至少包括：

(i)利用該故障物件路由器以外之複數個其他物件路由器的服務；

(ii)複製該故障物件路由器，以產生一複製物件路由器；

(iii)將該複製物件路由器設於至少一有功能的節點上；以及

(iv)將該複製物件路由器之一新位置通知該些節點之至少一次集合。

29.如申請專利範圍第 28 項所述之方法，更包括：

(v)將該複製物件路由器之該新位置通知該些其他物件路由器之至少一次集合。

六、申請專利範圍

30.如申請專利範圍第1項所述之方法，其中該物件路由器察覺到複數個其他物件路由器之至少一次集合。

31.一種至少包括複數個節點之網路計算平台，該網路計算平台至少包括：

至少一物件路由器，

其中該至少一物件路由器能夠管理複數個資源，並且平均分配負荷於該些節點，以及

其中該至少一物件路由器在該計算平台中維持所有該些節點中至少一次集合的位置之知識。

32.如申請專利範圍第31項所述之平台，其中該物件路由器將複數個物件重新設置在該些節點中。

33.如申請專利範圍第31項所述之平台，其中該物件路由器維持每一物件的複數個物件實體，而該些物件實體係部署在該網路計算平台中。

34.如申請專利範圍第31項所述之平台，其中每一該些節點至少包括：

將該些節點視為一集合之一物件合併器(Object Melder)，該物件合併器係用來執行合併，藉以產生該些節點與該至少一物件路由器。

(請先閱讀背面之注意事項再填寫本頁)

裝

訂

六、申請專利範圍

- 35.如申請專利範圍第34項所述之平台，更至少包括一物件版本管理器(Object Versioner)，該物件版本管理器係用來維持該網路計算平台中所有該些物件之複數個版本。
- 36.如申請專利範圍第34項所述之平台，更至少包括一物件複製器(Object Replicator)，該物件複製器在與一物件所在之一節點不同之另一節點，藉由複製該物件以維持該物件之一複製本。
- 37.如申請專利範圍第34項所述之平台，更至少包括一物件遷移器(Object Migrator)，藉以將一物件自一節點遷移至其他節點。
- 38.如申請專利範圍第34項所述之平台，更至少包括一資源監視器(Resource Monitor)。
- 39.如申請專利範圍第34項所述之平台，更至少包括一資源控制器(Resource Controller)。
- 40.如申請專利範圍第34項所述之平台，更至少包括一應用物件(Application Object)。
- 41.如申請專利範圍第34項所述之平台，其中每一該些節

(請先閱讀背面之注意事項再填寫本頁)

裝
訂

六、申請專利範圍

點至少包括：

一類別裝載器(Class Loader)負責裝載複數個物件之複數個類別，其中該類別裝載器容許一相同類別之複數個版本同時存在。

42.如申請專利範圍第31項所述之平台，更至少包括：

一演進模組(Evolution Module)，其中該演進模組容許該網路計算平台執行一演進步驟，而該演進步驟係提供一系統軟硬體之過渡時期，以可能完成無瑕縫之升級。

43.如申請專利範圍第31項所述之平台，其中當來自一第二節點之一代理物件無法接觸到一第一節點時，係藉由該網路計算平台從該些節點中偵測到該第一節點的故障。

44.如申請專利範圍第31項所述之平台，其中當一節點無法回應該物件路由器的中心請求時，係藉由該網路計算平台從該些節點中偵測到該節點的故障。

45.如申請專利範圍第31項所述之平台，其中當一節點無法向該物件路由器報告狀態或統計數值時，係藉由該網路計算平台從該些節點中偵測到該節點的故障。

六、申請專利範圍

46.如申請專利範圍第31項所述之平台，其中當該些節點之一節點故障時，該節點會被稱為一故障節點，而該至少一物件路由器能通知該至少一個物件路由器以外的複數個其他物件路由器。

47.如申請專利範圍第31項所述之平台，其中當該些節點之一節點故障時，該節點會被稱為一故障節點，而該至少一物件路由器能通知複數個其他節點。

48.如申請專利範圍第31項所述之平台，其中當該些節點之一節點故障時，該節點會被稱為一故障節點，而該至少一個物件路由器能製造一複製物件，該複製物件係相當於位在該故障節點以外之該些個節點中之複數個物件的至少一次集合。

49.如申請專利範圍第31項所述之平台，其中當該些節點之一節點故障時，該節點會被稱為一故障節點，而該至少一物件路由器能將複數個額外的計算資源設置於該網路計算平台中，並針對複數個複製物件產生複數個新的複製本，並將該些新的複製本設於該些新的複製本原來所在的節點以外之其他該些節點中。

50.如申請專利範圍第49項所述之平台，其中該網路計算平台能將該些新的複製本之一新位置通知複數個其他

(請先閱讀背面之注意事項再填寫本頁)

裝
訂

六、申請專利範圍

物件路由器。

51.如申請專利範圍第 49 項所述之平台，其中該網路計算平台能將離線的該故障節點通知一系統管理員。

52.如申請專利範圍第 31 項所述之平台，其中當該至少一物件路由器之一物件路由器故障時，該網路計算平台能利用該故障物件路由器以外之複數個物件路由器的服務。

53.如申請專利範圍第 31 項所述之平台，其中當該至少一物件路由器之一物件路由器故障時，該網路計算平台能複製該故障物件路由器以產生一複製物件路由器，並將該複製物件路由器設置於至少一有功能的節點上。

54.如申請專利範圍第 53 項所述之平台，其中該網路計算平台更能將該複製物件路由器之一新位置通知位於這些節點中至少一次集合。

55.如申請專利範圍第 31 項所述之平台，其中該物件路由器察覺到複數個其他物件路由器之至少一次集合。

56.一種針對至少包括複數個節點的網路計算平台之物件路由器，其中該物件路由器能夠管理複數個資源以及平

(請先閱讀背面之注意事項再填寫本頁)

裝

訂

六、申請專利範圍

均分配負荷於複數個節點，且該物件路由器維持在該計算平台中所有該些節點之至少一次集合的位置之知識。

57.如申請專利範圍第 56 項所述之物件路由器，其中該物件路由器在該些節點中重新設置複數個物件。

58.如申請專利範圍第 56 項所述之物件路由器，其中該物件路由器維持每一物件的複數個物件實體，而該每一物件係部署在該網路計算平台中。

59.如申請專利範圍第 56 項所述之物件路由器，其中當該些節點之一節點故障時，該物件路由器能通知該物件路由器以外的複數個其他物件路由器。

60.如申請專利範圍第 56 項所述之物件路由器，其中當該些節點之一節點故障時，該物件路由器能製造一複製物件，而該複製物件係相當於複數個物件之至少一次集合，其中該些物件係位在該故障節點以外之其他該些節點中。

61.如申請專利範圍第 56 項所述之物件路由器，其中當該些節點之一節點故障時，該物件路由器能於該網路計算平台中設置複數個額外的計算資源，並針對複數個複製

(請先閱讀背面之注意事項再填寫本頁)

裝

訂

六、申請專利範圍

物件產生複數個新的複製本，並將該些新的複製本設於該些新的複製本所在的節點以外之其他該些節點中。

62. 如申請專利範圍第 56 項所述之物件路由器，其中該物件路由器察覺到複數個其他物件路由器之至少一次集合。

63. 一種電腦程式產品，該電腦程式產品包括一電腦可讀式媒介工具，而該電腦可讀式媒介工具至少包括可使電腦實施產生可擴充並容忍誤差之一計算平台之方法的複數個指令，其中該計算平台係位於一網路上並至少包括複數個節點，而該些指令至少包括用於下列之指令：

a) 初始化時初始一合併步驟，其中在該合併步驟進行時，該些節點會被視為一集合，而該些節點中的每一節點都加入該集合，且於該合併步驟時，至少一物件路由器會被指定；

b) 在該些節點之間遷移複數個物件；以及

c) 在跨越該些節點之間複製該些物件。

64. 如申請專利範圍第 63 項所述之電腦程式產品，其中在該合併步驟時，該每一節點都加入該集合係使用一程序至少包括：

(a)(i) 要是能發現鄰近的一物件路由器，該節點就能找到該物件路由器，否則該節點會成為該物件路由器；

(請先閱讀背面之注意事項再填寫本頁)

裝
訂

六、申請專利範圍

以及

(a)(ii)該節點向該物件路由器報告複數個可利用的資源。

65.如申請專利範圍第 63 項所述之電腦程式產品，其中在該合併步驟時，該每一節點都加入該集合係使用一程序更至少包括：

(a)(iii)給該節點指定一網路位址；以及

(a)(iv)更新該節點的所有組件。

66.如申請專利範圍第 64 項所述之電腦程式產品，其中這些可利用的資源至少包括一記憶體、至少一處理器以及驅動埠儲存空間。

67.如申請專利範圍第 63 項所述之電腦程式產品，其中這些可利用的資源至少包括一網路頻寬。

68.如申請專利範圍第 64 項所述之電腦程式產品，其中該更新步驟係利用儲存於一程式碼版本伺服器之資料來進行。

69.如申請專利範圍第 64 項所述之電腦程式產品，其中係確定在任何時候該物件路由器有至少一鏡像物件存在。

六、申請專利範圍

70.如申請專利範圍第 63 項所述之電腦程式產品，其中該物件路由器管理複數個資源並平均分配負荷於該些節點。

71.如申請專利範圍第 63 項所述之電腦程式產品，其中該物件路由器在該計算平台中維持所有該些節點中至少一個次集合的位置之知識。

72.如申請專利範圍第 63 項所述之電腦程式產品，其中該物件路由器重新設置該些節點中之複數個物件。

73.如申請專利範圍第 63 項所述之電腦程式產品，其中該遷移步驟中一物件使用之一程序至少包括：

(b)(i)鎖定該物件；

(b)(ii)在該些節點中之一第一節點上，將該物件轉換為一位元組流；

(b)(iii)在該網路中轉移該位元組流；

(b)(iv)在該些節點中之一第二節點上，將該位元組流轉換回該物件；

(b)(v)更新該物件在該物件路由器中之一位置；以及

(b)(vi)將該物件解開鎖定。

74.如申請專利範圍第 63 項所述之電腦程式產品，其中該複製步驟中一物件使用之一程序至少包括：

(請先閱讀背面之注意事項再填寫本頁)

裝
訂

六、申請專利範圍

- (c)(i) 鎖定該物件於一第一節點上；
- (c)(ii) 形成一複製的物件；
- (c)(iii) 將該複製的物件設於一第二節點上；以及
- (c)(iv) 訂閱與該物件有關之複數個事件。
- (c)(v) 將該物件解開鎖定。

75. 如申請專利範圍第 74 項所述之電腦程式產品，其中該複製步驟係藉由自該物件產生一位元組流，並且將該位元組流轉換回該複製的物件來進行。

76. 如申請專利範圍第 63 項所述之電腦程式產品，其中該計算平台容許一演進步驟，而該演進步驟係提供一系統軟硬體之過渡時期，以可能完成無瑕縫之升級。

77. 一種電腦程式產品，該電腦程式產品包括一電腦可讀式媒介工具，而該電腦可讀式媒介工具至少包括複數個指令，其中該些指令能使一電腦執行一演進步驟，並藉由執行該演進步驟使一軟體能有一無瑕縫之過渡，而該些指令至少包括用於下列之指令：

- (a) 評估欲安裝或升級之一軟體或一硬體；
- (b) 在編譯的時候於一計算平台中產生所有複數個物件之一相依階層樹狀結構；
- (c) 於該些物件中至少一類別的一舊版本與一新版本之間，執行一物件類別的差異比較；

六、申請專利範圍

(d)針對有更改的每一狀態變數產生一轉換函數；

(e)選取一物件，被選取之該物件為一選取物件類別之一物件實體；

(f)鎖定被選取之該物件；

(g)產生一未初始化物件實體，該未初始化物件實體係為被選取之該物件之該新版本；

(h)藉由應用適當之複數個轉換函數於被選取之該物件之該舊版本，以初始該選取物件類別之一新物件實體之複數個變數；

(i)將被選取之該物件解開鎖定；以及

(j)對該物件類別的每一該物件實體重覆該(e)步驟至該(i)步驟。

78.如申請專利範圍第77項所述之電腦程式產品，更至少包括：

(l)對該計算平台中的每一該物件類別重覆該(d)步驟至該(j)步驟。

79.如申請專利範圍第77項所述之電腦程式產品，更至少包括：

(m)部署一客戶端軟體。

80.如申請專利範圍第77項所述之電腦程式產品，更至少包括：

六、申請專利範圍

(n) 如果不需要該些物件之多個版本，銷毀該些物件之該些舊版本。

81. 如申請專利範圍第 63 項所述之電腦程式產品，其中該物件路由器維持每一該些物件之多個物件實體。

82. 如申請專利範圍第 63 項所述之電腦程式產品，其中當來自該第二節點之一代理物件無法接觸到一第一節點時，會從該些節點中偵測到該第一節點之故障。

83. 如申請專利範圍第 63 項所述之電腦程式產品，其中當該些節點中之一節點沒有回應該物件路由器的中心請求時，會偵測到該節點的故障。

84. 如申請專利範圍第 63 項所述之電腦程式產品，其中當該些節點中之一節點沒有向該物件路由器報告狀態或統計數值時，會偵測到該節點的故障。

85. 如申請專利範圍第 63 項所述之電腦程式產品，其中當該些節點之一節點故障時，該節點會被稱為一故障節點，而該至少一個物件路由器係使用一程序來回應，且該程序至少包括：

(i) 通知除了該至少一個物件路由器以外之複數個其他物件路由器；

(請先閱讀背面之注意事項再填寫本頁)

裝
訂

六、申請專利範圍

(ii)於該故障節點選取一物件；

(iii)在該故障節點之外的其他該些節點製造複數個複製物件，而該些複製物件係相當於被選取之該物件；

(iv)於該計算平台中設置複數個額外的計算資源；

(v)在該(iii)步驟中針對該些複製物件產生複數個新的複製本，並將該些新的複製本設於該複製物件所在的節點以外之該些其他節點；以及

(vi)將該些新的複製本之一新位置通知該些其他物件路由器。

86.如申請專利範圍第85項所述之電腦程式產品，更至少包括：

(vii)將離線之該故障節點通知一系統管理員。

87.如申請專利範圍第85項所述之電腦程式產品，更至少包括對於該故障節點中所有該些物件重覆該(ii)步驟至該(vi)步驟。

88.如申請專利範圍第63項所述之電腦程式產品，其中當該至少一物件路由器中一物件路由器故障時，故障的該物件路由器就稱為一故障物件路由器，而該計算平台係使用一程序來回應，且該程序至少包括：

i)利用該故障物件路由器以外之複數個其他物件路由器的服務；

六、申請專利範圍

ii)複製該故障物件路由器，以產生一複製物件路由器；

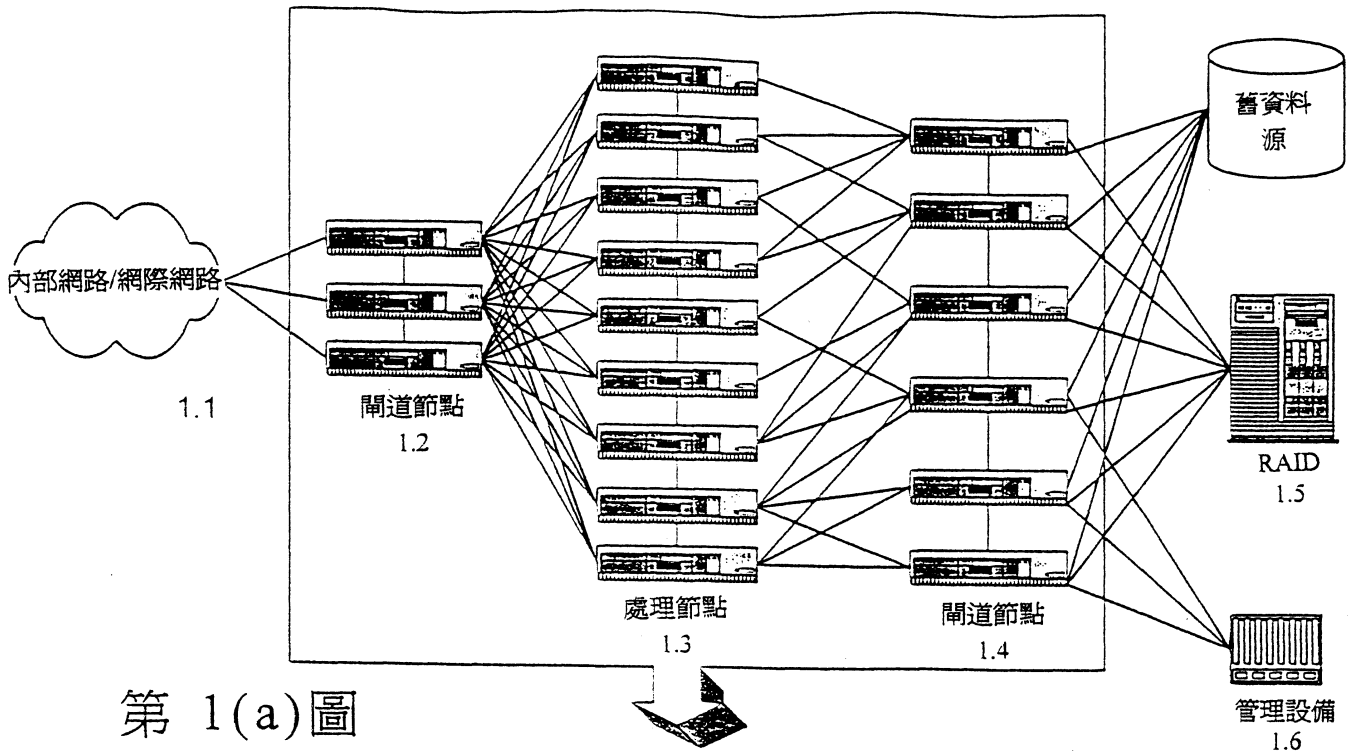
iii)將該複製物件路由器設於至少一有功能的節點上；以及

iv)將該複製物件路由器之一新位置通知該些節點之至少一次集合。

89.如申請專利範圍第63項所述之電腦程式產品，其中該物件路由器察覺到複數個其他物件路由器之至少一個次集合。

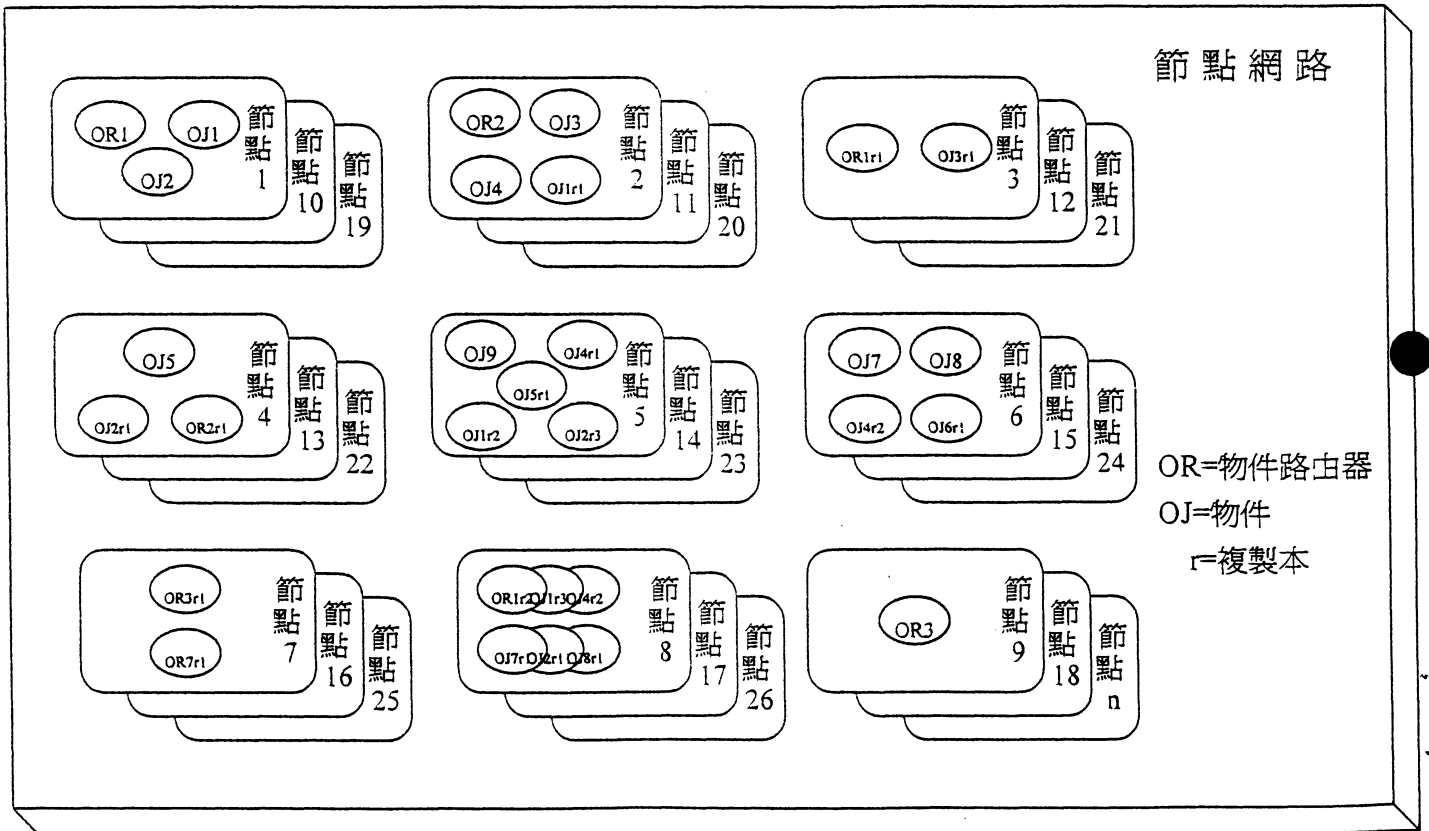
(請先閱讀背面之注意事項再填寫本頁)

裝
訂



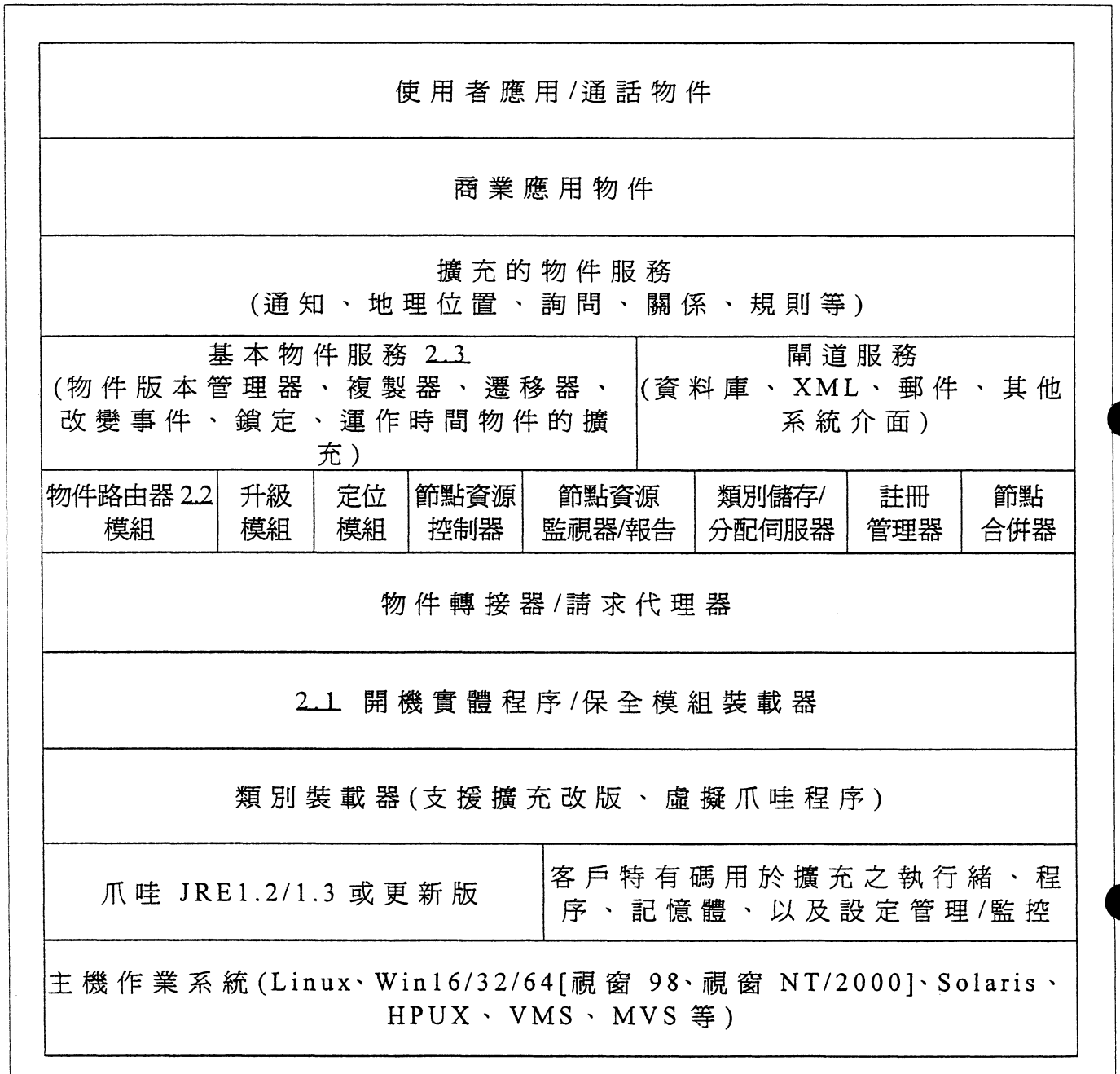
第 1(a)圖

概念圖

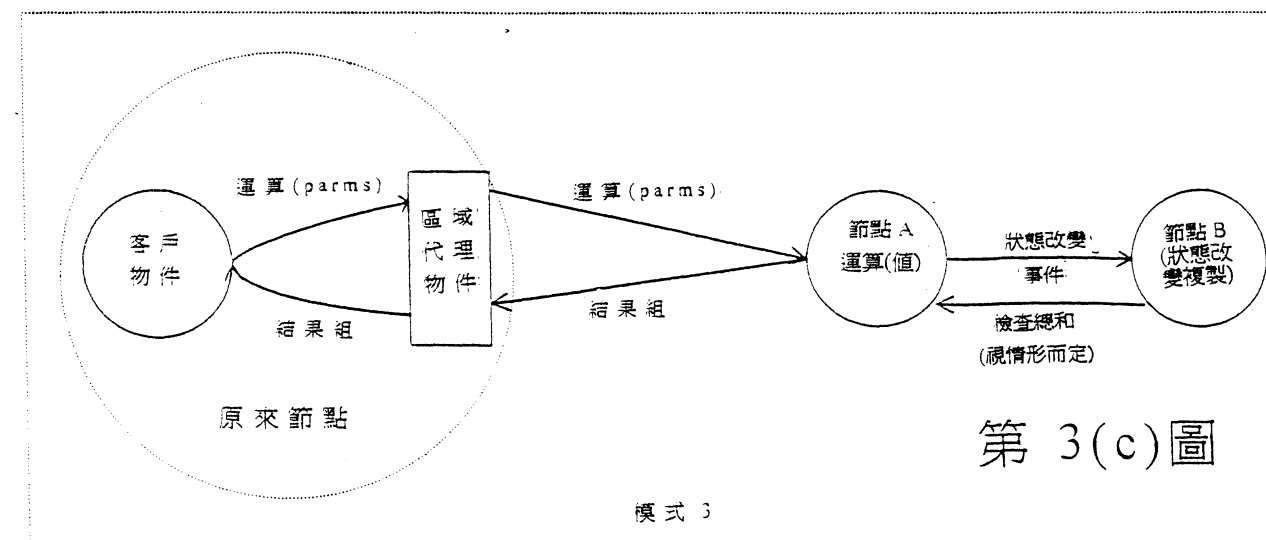
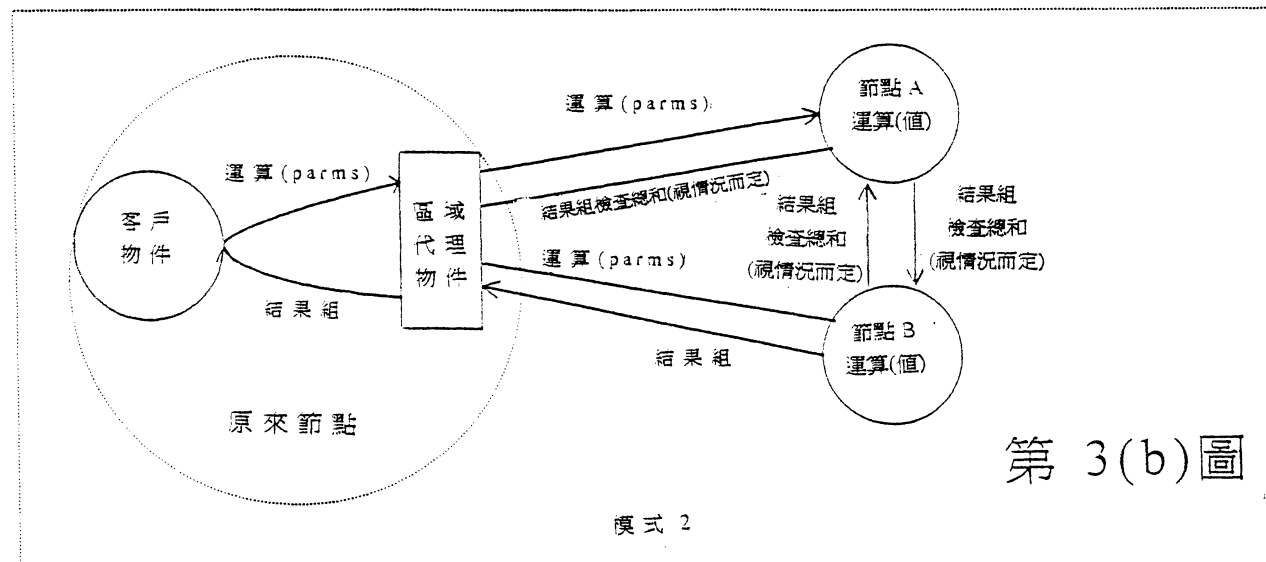
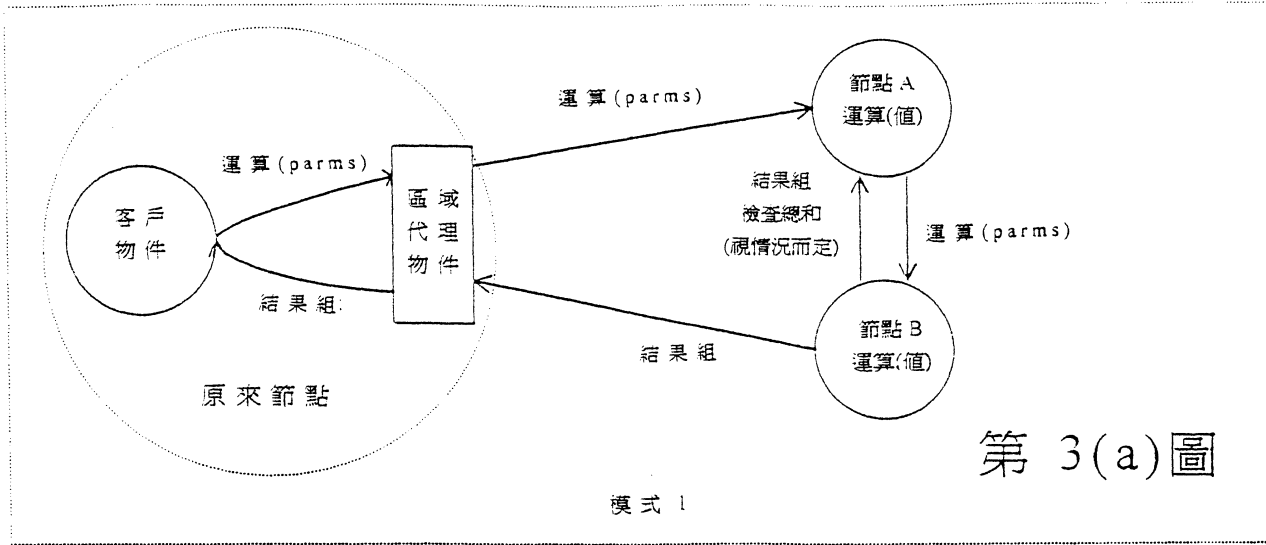


第 1(b)圖

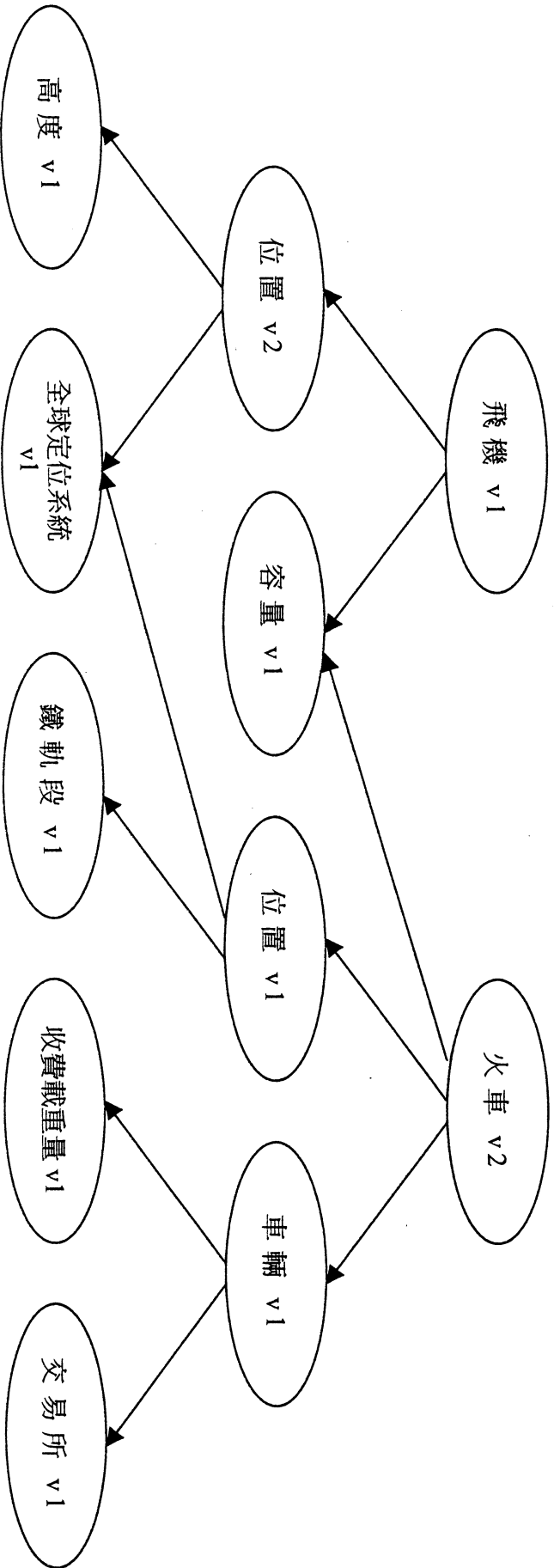
節點架構圖



第 2 圖

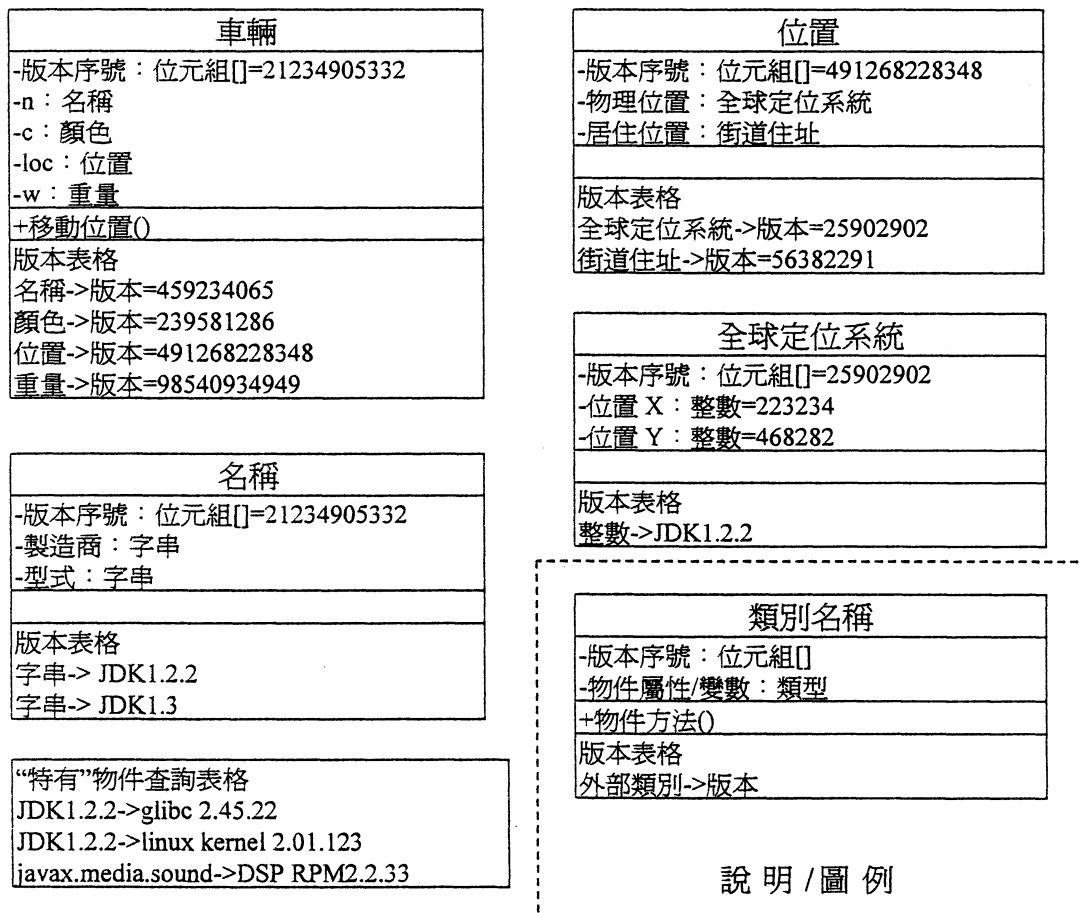


物件類別版本樹狀圖

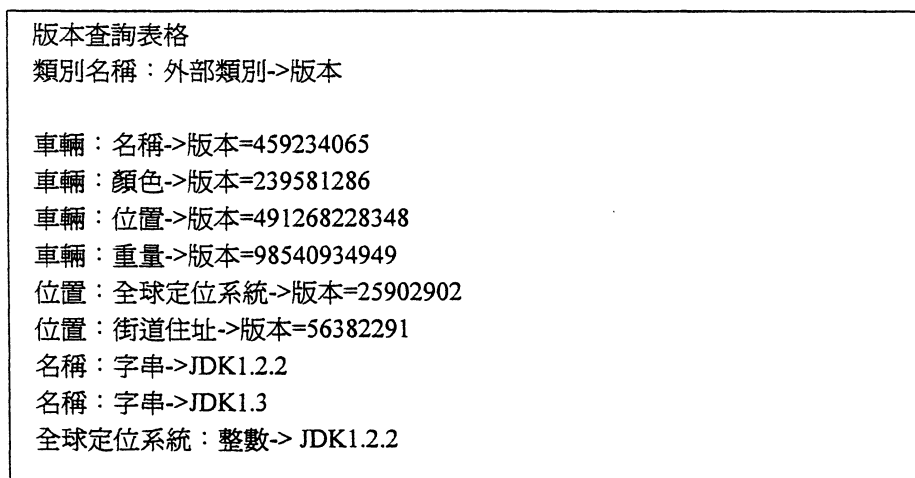


第 4 圖

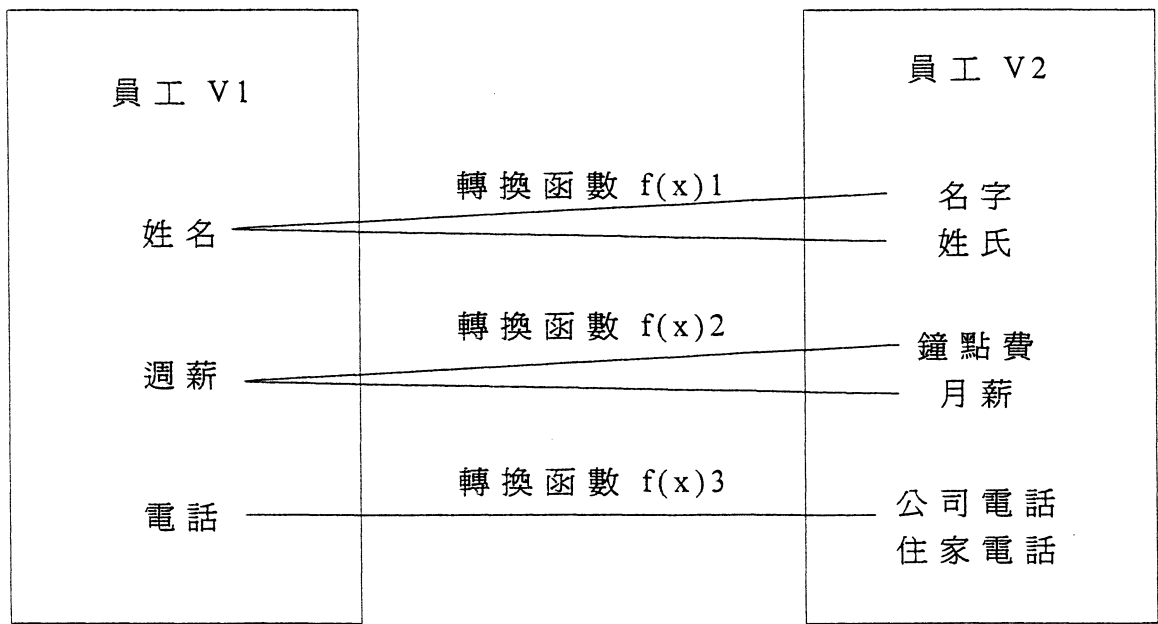
嵌入物件分版管理



第 5(a)圖

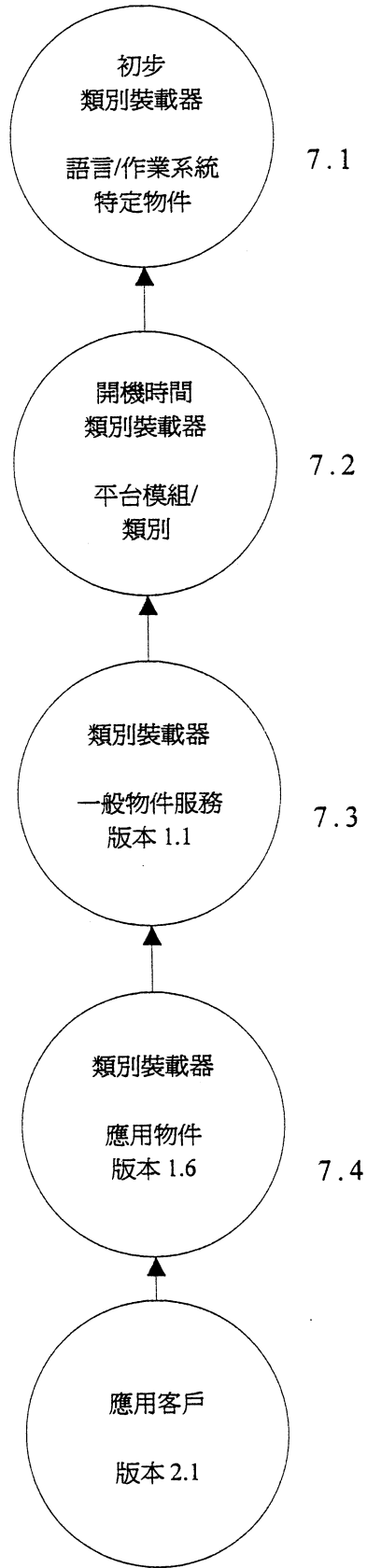


第 5(b)圖



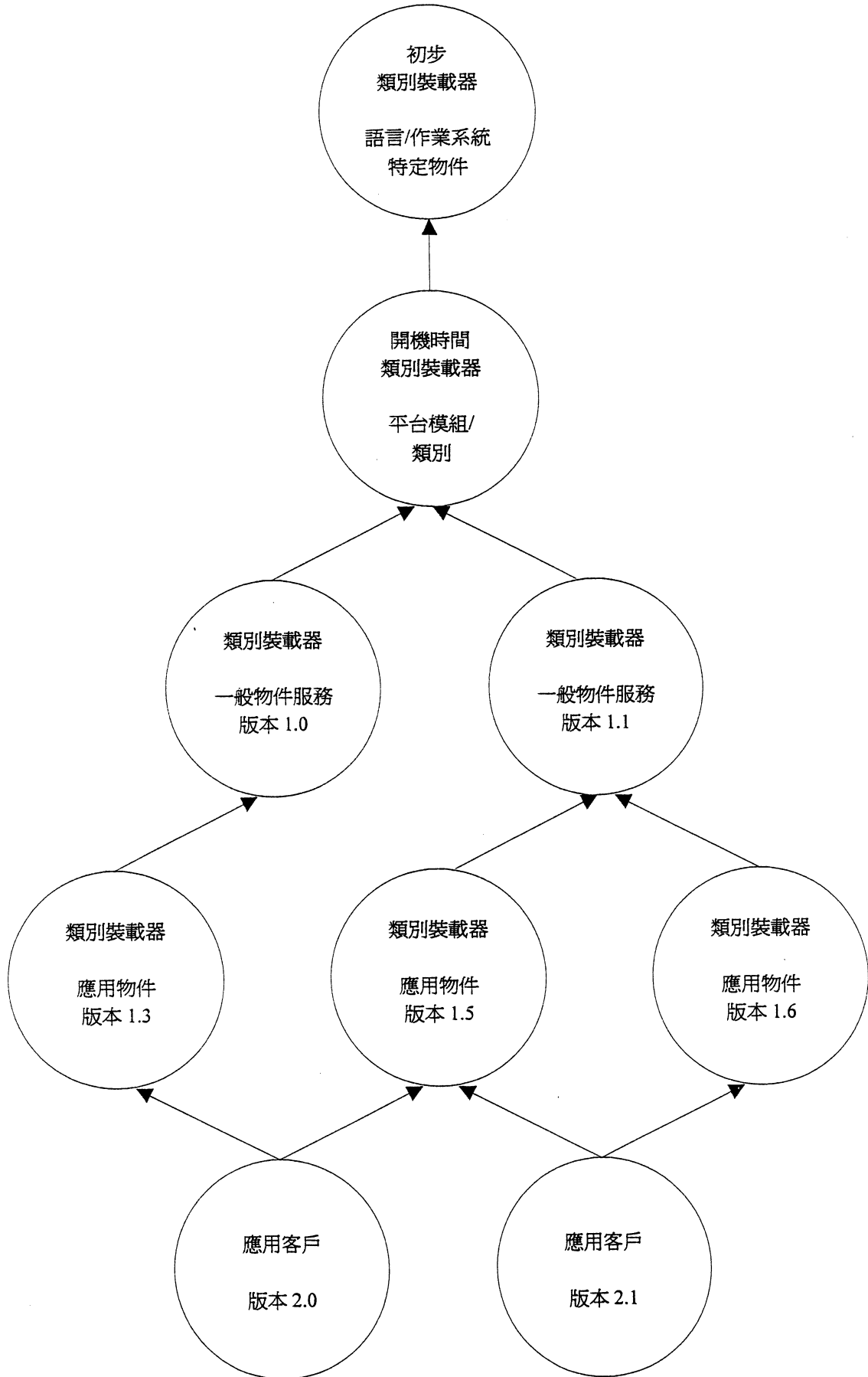
第 6 圖

類別裝載器的連鎖結構

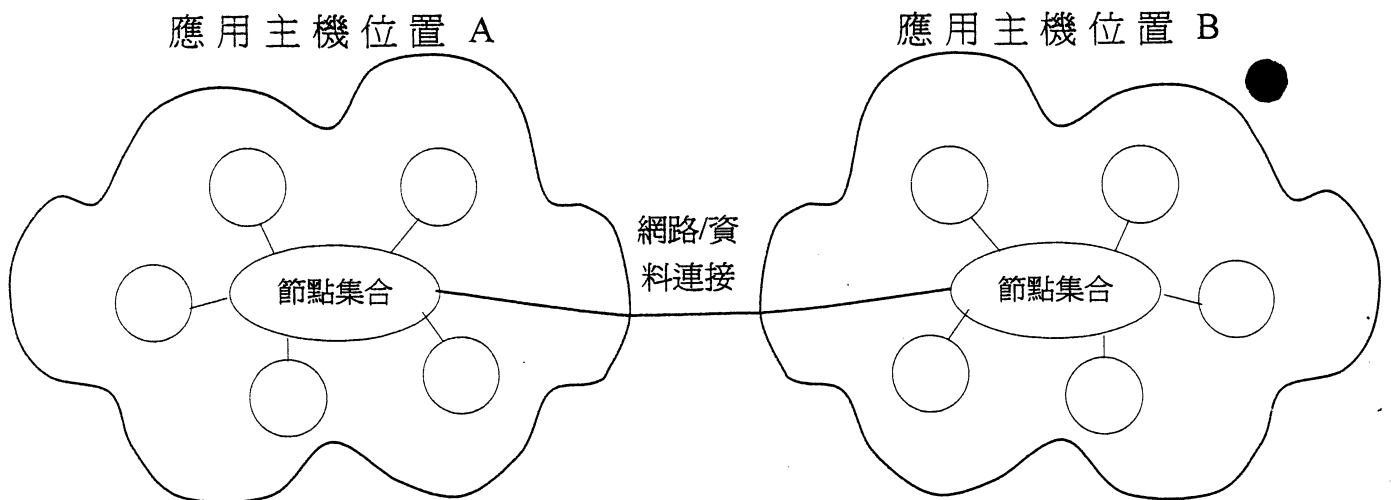
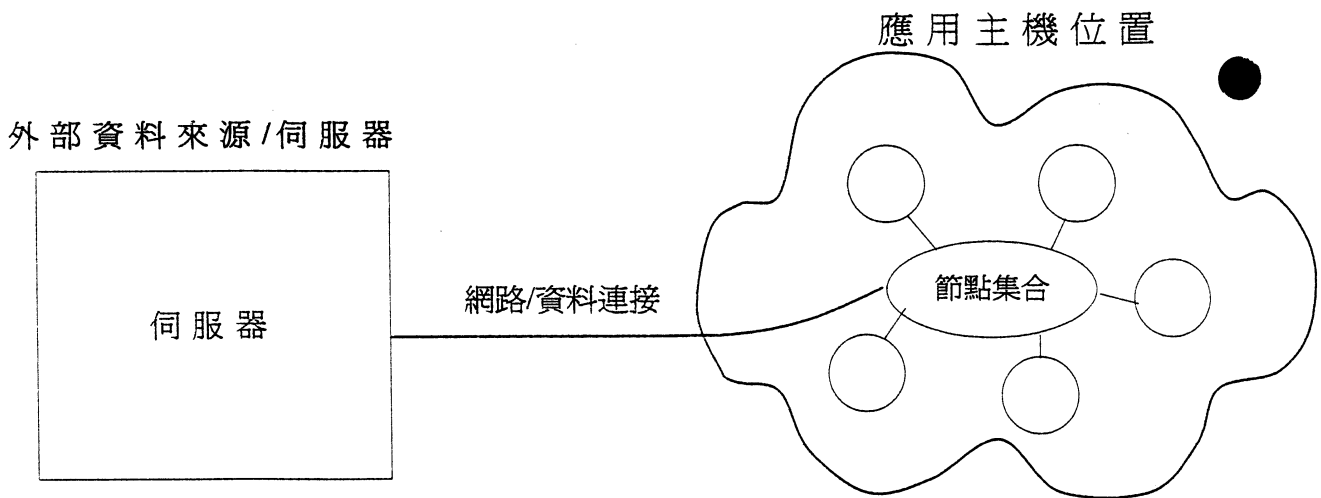
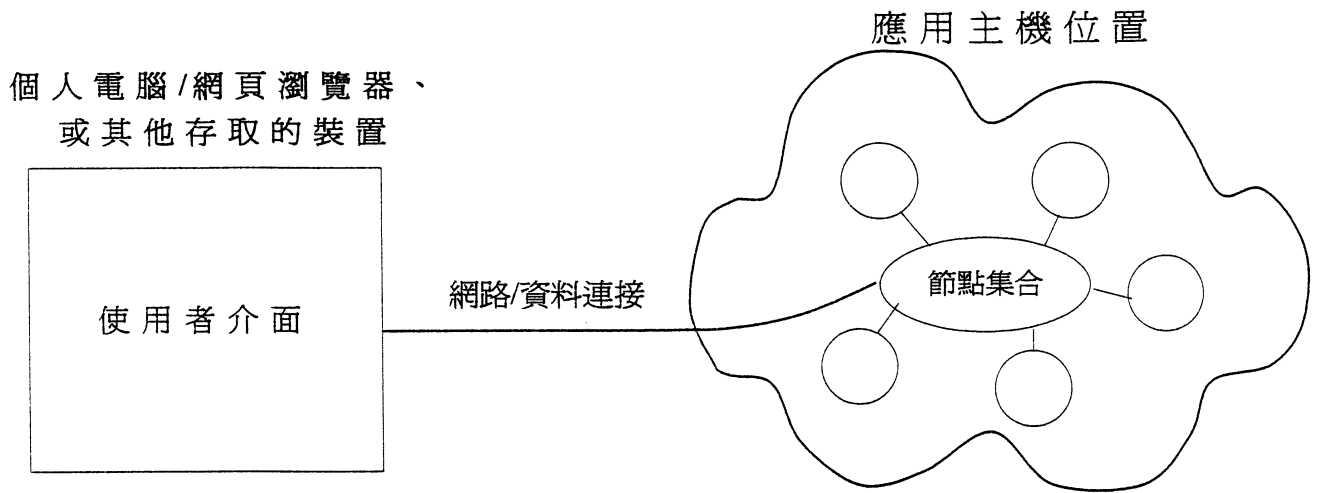


第 7 圖

類別裝載器的階層結構



第 8 圖



第 9 圖