附件 1: 第 85108517 號專利申請案
中文說明書修正本　　民國 89 年 6 月呈

申請日期	85 年 7 月 13 日
案號	85108517
類別	G06F 9/06.

410312

A4
C4

（以上各欄由本局填註）

發明 新型 專利說明書

410312

一、發明 新型 名稱	中文	資料處理方法和裝置
	英文	Data processing method and device
二、發明 創作 人	姓名	(1) 橫手靖彥
	國籍	(1) 日本
	住、居所	(1) 日本國東京都品川區東五反田三－一四－一三 （株）ソニーコンピュータサイエンス研究所
三、申請人	姓名（名稱）	(1) 蘇妮股份有限公司 ソニー株式会社
	國籍	(1) 日本
	住、居所（事務所）	(1) 日本國東京都品川區北品川六丁目七番三五號
	代表人 姓名	(1) 出井伸之

經濟部智慧財產局員工消費合作社印製

本紙張尺度適用中國國家標準（CNS）A4規格（210×297公釐）

附件 1: 第 85108517 號專利申請案
中文說明書修正本　　民國 89 年 6 月呈

申請日期	85 年 7 月 13 日
案號	85108517
類別	G06F 9/06.

410312

A4
C4

（以上各欄由本局填註）

發明 新型 專利說明書

410312

一、發明 新型 名稱	中文	資料處理方法和裝置
	英文	Data processing method and device
二、發明 創作 人	姓名	(1) 橫手靖彥
	國籍	(1) 日本
	住、居所	(1) 日本國東京都品川區東五反田三－一四－一三 （株）ソニーコンピュータサイエンス研究所
三、申請人	姓名（名稱）	(1) 蘇妮股份有限公司 ソニー株式会社
	國籍	(1) 日本
	住、居所（事務所）	(1) 日本國東京都品川區北品川六丁目七番三五號
	代表人 姓名	(1) 出井伸之

經濟部智慧財產局員工消費合作社印製

本紙張尺度適用中國國家標準（CNS）A4規格（210×297公釐）

（由本局填寫）

承辦人代碼：	A6 B6
大　　類：	
ＩＰＣ分類：	

本案已向：

國（地區）	申請專利，申請日期：	案號：	，□有　□無主張優先權
日本	1995 年 7 月 14 日	7-178625	☑無主張優先權
日本	1996 年 3 月 25 日	8-067881	☑無主張優先權

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（ 1 ）

發明背景

本發明係有關於資料處理方法及裝置，尤其是結構簡單的低成本資料處理方法及裝置。

最近，已廣泛使用個人電腦，這些個人電腦已用於經網路存取預述的伺服器，因此可得到預述資訊。

必需要有應用程式以執行具這些個人電腦的多種處理。使用者購買可在其個人電腦之作業系統（下文稱爲ＯＳ）下操作的應用程式，且直接從記錄媒質或經通訊連結安裝並使用這些應用程式。

在日本特評公開專利案號Ｈｉｅ　１－１５２５３６１２中提出一構想，可決定當經一通訊連結安裝應用程式時，決定一客體是否適用。但是此動作之目的在於決定客體電腦的使用者是否爲一正規使用者，而在程式安裝時，程式及電腦間的相容性並沒有決定。

當使用不同的ＯＳ時，應用程式亦不同，然後各使用者選擇且購買適用本身之ＯＳ的應用程式。而各應用程式可提供者（應用程式設計者）亦需要設計多種應用程式（等於ＯＳ數）以執行基本上爲相同的處理，其包含多種處理而且相當耗時。

當應用程式具相同ＯＳ時也發生相似的問題。即當這些應用程式之一與其他應用程式不同時，兩應用程式必需分開設計，甚至當他們在同一ＯＳ之下操作。此使得工作量及成本高於提供單一的應用程式。

本發明係用於解決這些問題。本發明的目的係提供簡

（請先閱讀背面之注意事項再填寫本頁）
裝 訂 線
經濟部智慧財產局員工消費合作社印製
本紙張尺度適用中國國家標準（CNS）A4規格（210×297公釐）

五、發明說明（ 2 ）

單，成本低且與電腦型式無關的單一應用程式。

發明概述

依據本發明的資料處理方法，當一應用程式將下載至客體，一伺服器核對一客體是否具有一用於應用程式的執行環境，且該執行環境依據核對的結果下載應用程式至客體中。

而且，使用一方法執行應用程式，以解釋及執行轉換至中間碼的應用程式，及一方法，用於動態編譯中間碼，且執行產生的二位元碼。例如，當編譯困難時，可逐漸解釋及執行中間碼。

本發明的資料處理裝置提供核對器當應用程式將下載客體時，用於核對客體是否具有一用於將下載之應用程式的執行環境，且提供一下載器，用於依據核對器的核對結果，將應用程式下載至客體中。

資料處理裝置更包含一告知器，當應用程式將從伺服器中下載時，用於告知執行環境應用程式該下載，且包含一下載器，依據告知器的告知結果從伺服器中下載應用程式。

此資料處理更包含第一執行器，以解釋及執行轉換成中間碼的應用程式，用於動態編譯中間碼且產生二位元碼的二位元碼產生器，及第二執行器，用於執行二位元碼及系統物件。例如當編譯困難時，中間碼可逐漸解釋及執行。

（請先閱讀背面之注意事項再填寫本頁）
裝 訂 線
經濟部智慧財產局員工消費合作社印製
本紙張尺度適用中國國家標準(CNS)A4規格(210×297公釐)

五、發明說明（ 3 ）

圖式簡述

圖１爲本發明資料處理方法所使用之系統結構之視圖；

圖２之視圖顯示應用程式之結構；

圖３之視圖示同時物件之結構；

圖４之視圖用於說明物件從伺服器下載至多個販賣機的客體中；

圖５之視圖說明增加下載；

圖６之視圖說明共同標準；

圖７之視圖說明物件的動態改變；

圖８之視圖說明物件之動態擴充；

圖９之視圖說明客體之最小功能；

圖１０之視圖說明適於應用及動態再結構之客體環境結構；

圖１１之視圖說明一特徵結構的結構；

圖１２之視圖顯示本發明之資料處理裝置所應用之系統結構例；

圖１３之視圖顯示ＭＶＭ及ＭＫ的邏輯結構；

圖１４之視圖結構及說明器之邏輯結構；

圖１５之視圖顯示ＭＶＭ之整個結構；

圖１６之視圖顯示文書及環繞文書之項目的資料結構；

圖１７之視圖顯示變數表項的型式；

（請先閱讀背面之注意事項再填寫本頁）
裝 訂 線
經濟部智慧財產局員工消費合作社印製
本紙張尺度適用中國國家標準(CNS)A4規格(210×297公釐)

五、發明說明（ 4 ）

圖１８之視圖顯示原始物件表列及介面名稱；

圖１９之視圖示原始物件介面；

圖２０之視圖示圖１９中原始物件介面之接續圖；

圖２１之視圖示圖２０中原始物件介面之接續圖；

圖２２之視圖示Ｉ介面結構組；且

圖２３之視圖示ＭＫ介面。

符號說明

１　伺服器

２　客體

３　網路

１１　應用程式

１２　執行環境

１３　應用程式介面

１４　物件

１５　物件

２１　應用程式

２２　執行環境

２３　應用程式介面

３１　核心

３１ａ　ＭＶＭ

３１ｂ　ＭＫ

３２　裝置驅動器

３３　個人物件

（請先閱讀背面之注意事項再填寫本頁）
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（ 5 ）

34 分類庫

35 應用程式

41 動作文書

42 訊息框

43 執行機構

44 原始物件

51 文書結構

52 說明器結構

61 物件

62 分類

63 I方法結構

較佳實施例之詳細說明

第一實施例

圖1示系統結構之一例，該結構爲本發明之資料處理方法所使用者。該系統包含一伺服器1（資料處理裝置），一客體2（資料處理裝置）及一網路3。

在此實施例中，伺服器1含兩應用程式11－1及11－2。一應用程式11－1含一執行環境12－1，用於定義應用程式11－1及應用程式介面（下文稱爲"API"）13－1包含一介於應用程式11－1及執行環境12－1間的介面。

應用程式11－1包含多個物件14－1，且執行環境12－1包含多個物件15－1。

（請先閱讀背面之注意事項再填寫本頁）
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（ 6 ）

應用程式１１－２亦包含一執行環境１２－２，其定義用於此應用程式１１－２的環境，且一ＡＰＩ１３－２，其功能作爲應用程式１１－２及執行環境１２－２間的介面。

而且，應用程式１１－２亦包含多個物件１４－２，且執行環境１２－２包含多個物件１５－２。

客體２亦包含兩應用程式１２－１及１２－２。應用程式１２－１含執行環境２２－１，用於定義使用在此應用程式１２－１的環境，及一ＡＰＩ２３－１作爲應用程式２１－２及執行環境２２－１間的介面。應用程式２１－１包含多個物件２４－１且執行環境２２－１包含多個物件２５－１。

應用程式２１－２亦包含一執行環境２２－２，形成用於應用程式２１－２及ＡＰＩ２３－２的環境。應用程式２１－２包含多個物件２４－２，且執行環境２２－２包含多個物件２５－２。

上述諸物件均定義成同時處理之物件，其與其他物件同時處理。多個ＡＰＩ存在於伺服器及客體２之間，因爲一ＡＰＩ設定可視爲執行環境。

如圖１，２所示，應用程式１１包含多個聚集之物件１４。而且，經由應用程式１１由架構物件１４處理成同時物件而增加執行速度。因爲物件１４爲一取代單元，含操作上錯誤的物件或執行有問題的物件等可由設有錯誤的物件加以取代。物件１４的問題因此可解決，而不需要對

（請先閱讀背面之注意事項再填寫本頁）
裝 訂 線
經濟部智慧財產局員工消費合作社印製
本紙張尺度適用中國國家標準(CNS)A4規格(210×297公釐)

五、發明說明（ 7 ）

應用程式１１完全再塑形。而且，由該物件１４作爲一部份，且結合物件１４與作爲已存在應用程式之一部份的標物而很容易地製作一新的應用程式。

此應用程式爲單一服務單元，且例如可爲一應用程式，且只用於顯示來自伺服器１的影像資料；用於應用一ＶＣＲ重取影像資料；選擇一使用一表單的服務；郵購；與郵購連結的家用計帳簿；稅款計算等。

用於共同特徵的操作可由共用物件間的應用程式得到。例如，可共用輸入資料至家用計帳應用程式之編輯器，及用於在郵購中輸入資料的編輯器。

其次，可給定現用物件的說明，該現用物件之配置可參見圖３。爲現用物件的物件１４包含一向外部大家公開的方法進入表１４Ａ，一方法碼１４Ｂ，一記憶體區１４Ｃ，用於保留物件狀態，及一簡化的執行紋（thread）用於執行該方法。在同時物件處存在的執行正文（亦稱爲執行紋）。因此同時物件接收一訊息，且在此訊息處理期間，不處理到達訊息，直到現在正處理的訊息已完全執行。

在物件內只提供一執行紋者具有下列優點。

（１）不需要考慮多個動作間的同步。即，不再需要執行此處理以決定對共用資料存取之序列，當共用資料存在時，該共用資料使用一如號誌（semaphore）的同步指令。另言之，向物件遷移之訊息的序列決定存取序列。

（２）結果，在得到同步時所產生的錯誤而導致程式誤差不再發生，且可改進物件的再使用性。

（請先閱讀背面之注意事項再填寫本頁）
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（ 8 ）

（ 3 ）在大部份的例子中，由提供如使用此方法的裝置驅動器可防止同步誤差。

（ 4 ）因爲更換裝置驅動器時所產生的同步誤差可防止，所以裝置驅動器可安全更換。

（ 5 ）用於裝置驅動器之部份（而非用於實際控制硬體的部份）可獨立提供，而與ＯＳ無關，由於裝置驅動器之程式發展上所需要的時間量可共用，所以程式發展周期可更換。

（ 6 ）與物件間執行裝置驅動器有關的說明可從應用程式之說明中除去。例如，通常需要將執行紋執行控制程式併入應用程式內，例如使用多thread方式（當使用多個thread時）。然後，當thread程式環境改變時，需要重寫應用程式。但是，如果在本發明中只有一thread，則不需要在應用程式中說明此部份。因此甚至當執行控制方法改變，也不需要重寫應用程式。輸入此同時物件（在此執行環境中）最適當的執行控制方法爲由使用物件之對應擴展系統所提供。

（ 7 ）因此當說明一應用程式時，不需要考需平行處理。如果處理同時物件，即系統執行平行處理，此後硬體位在最適當的自動執行控制之下，此係因爲物件爲平行處理單元。在程序化期間，多個處理及thread數之產生需要加以設計。如果此項設計的製作並沒有考慮硬體性能，則應用程式變爲有助於特定的硬體。

應用此一系統，如需要可將物件下載。圖 4 中示一系

（請先閱讀背面之注意事項再填寫本頁）

裝 訂 線

經濟部智慧財產局員工消費合作社印製

本紙張尺度適用中國國家標準(CNS)A4規格(210×297公釐)

五、發明說明（ 9 ）

統之例，此系統用於從伺服器１中下載物件至多個販賣機的客體２。爲對應販賣機所使用的Client API １３（１３－１，１３－２）由執行環境１２（１２－１，１２－２）所實現。

當一物件下載由客體２時（２－１，２－２），其決定與伺服器１之執行環境１２相同的應用程式２２（２２－１，２２－２）存在客體２上。如果同一執行環境不存在，則下載物件。如果不，在與伺服器１執行環境１２相同之執行環境２２架構後執行下載動作。

例如，在圖４中，圖伺服器１之應用程式１１－１的物件１４－１下載，如同客體２－１之應用程式２１－１的物件２４－１時，一對應伺服器之執行環境的物件１５－１A之物件２５－１A需要在客體２－１的執行環境２２－１中。然後，比如說，執行環境１２－１之物件１５－１B（核對機構）咨訊執行環境２２－１之物件２５－１B（告知機構）之特徵結構（將於下文中說明）。然後，執行環境１２－１的物件１５－１C（下載機構）及執行環境２２－１的物件２５－１C（下載機構）依據此回應下載執行環境１２－１的物件１５－１A及１５－１B如同執行環境２２－１的物件２５－１A至２５－１B一般。

在相關的方法中，需要對將下載之物件考慮客體ＡＰＩ。比如說，當客體爲一ＵＮＩＸ系統時，同一ＵＮＩＸ系統可使用在系統上，需要提供一物件，其與同類橫向發

（請先閱讀背面之注意事項再填寫本頁）
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（10）

展環境所架構。如果伺服器及客體均配備同一執行環境通常客體裝置必需配備昂貴的計算來源。例如，當與提供助益（dedicated）執行環境之例子比較，需要提供更多的記憶體，且需要提供更速ＣＰＵ（中央處理單元）以保證充分的執行速度，但此將導致成本增加。

因此之故，依據本發明的系統，此問題的解決方法爲當下載應用程式同時，對應用程式下載執行環境。即當在客體中沒有備製不需要的來源時，在客體中架構現在需要的執行環境２２。例如，如果客體２不需要３Ｄ繪圖，則不需要製備一用於此目的資料庫。

而且，當客體爲一使用ＶＯＤ（Vedeo On Demand）的重新播放電影影像時，用於與使用者互動的服務（當觀看電影時，不需要伺服器）可從客體中除去，而計算的對應量可宣告予其他工作。然後，這些來源作爲來自伺服器１之影像資料的預先取出影像。然後，當需要時，可從伺服器下載互動服務。

下列項中視物件狀況應用於本發明的系統下載。

（１）全部應用程式。

（２）裝置驅動器群（如ＭＰＥＧ８１３），ＡＴＭ驅動器，影像控制驅動器等）用於控制爲客體所提供的硬體來源。

（３）物件群（如ＶＣＲ命令管理，串列管理，即時計算器，記憶體管理，視窗管理，下載控制，通信準則管理，執行管理等），其提供用於應用程式的系統伺服器。

（請先閱讀背面之注意事項再填寫本頁）
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（ 11）

用於應用程式之最適當執行環境可由結合這些應用程式及物件群而在物件上架構。

伺服器可爲經網路記憶體遷移影像資料及應用程式的裝置或遷移資訊至客體２的裝置。客體２爲一裝置，用於處理來自伺服器１的資訊，且通常不需要客體與該網路連結。因此，可對各應用程式備製最適應的執行環境，此係因執行環境提供予各應用程式。

在相關技術中，當架構伺服器時，必需先預估應用程式的特徵。例如，當應用程式需要處理影像資料時，需要該系統服務配備ＶＣＲ等使用者介面，以處理即時規畫及影像資料。而且，應用程式使用３Ｄ繪圖，提供用於此目的一資料庫，且傾向於擴充系統。典型例爲ＵＮＩＤ（註冊商標）及視窗（註冊商標），其中每次型式改進時，系統所需要的記憶體量跟著增加。應用本發明的系統，只提供一部份的功能以執行應用程式，且相關系統的問題也已解決。

由架構此應用程式１１爲多個物件的組合，此提供這些物件作爲同時物件，使用作爲一單元的各物件，可能進行同時物件，且同時，當執行應用程式時，可下載物件。此時，如圖５所示，在增加的應用程式執行中，由下載所需要的物件，使用者可感覺到下載單一應用所需的時間已降低。

例如，如圖５所示，如客體２之應用程式２１的物件２４－１－１至２４－１－１１，當必需要下載伺服器１

（請先閱讀背面之注意事項再填寫本頁）
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（12）

之應用程式11的物件14-1-1至14-1-11時（而非任何下載物件），首先需要在應用程式21之執行中所需要的物件14-1-1至14-1-3先下載，如物件24-1-1至24-1-13一般。

如果這些物件存在，可動作應用程式21，且開始處理。當執行此處理時，然後，在第二至第四步驟中，如應用程式21的物件24-1-4至24-1-11，依序下載其餘物件14-1-4至14-1-11。執行此第二至第四下載，係依何物件需要先下載而執行之。

當下載應用程式21的物件24-1-1至24-1-3時，用於應用程式21的處理開始。然後，使用者感覺是否所有物件用於下載。因此使用者需要知道下載三個物件所需要的時間，其短於下載11個物件所需要的時間。另言之，實際上，使用者不知道下載8個物件所需要的時間，因此如同該時間已移除一般。

此亦可用於客體2中圖4中所說明的執行環境22之架構中（在圖10中亦將再予說明）。在此例中，對使用者來說，對執行環境下載物件所需要的時間似乎少於第一次下載只用於執行來自包含執行環境22之物件的應用程式所需要的物件之時間，此方法亦可用於啓動系統。

此次，增加下載意謂著以物件爲單位下載應用程式或執行環境，或下載包含所需要之應用程式或執行環境的部份，而非一次下載。當在傳統個人電腦通訊中，下載應用程式時，壓縮的應用程式一次下載。因此應用程式不能使

（請先閱讀背面之注意事項再填寫本頁）
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（ 13）

用，直到完成下載爲止。而且，例如應用系統自動載入，當所有系統已讀入記憶體時，則開始啓動系統。在UNIX disc-less工作站之例，在全部物件已從伺服器讀入記憶體時，系統才開始，因此在讀取完成前，系統無法使用。但是當使用增加下載（incremental downloading）時則情況就不同了。

此方法很成功地使用到伺服器1中，且如下文說明一Set Top Box（下文稱爲”ＳＴＢ”）作爲客體2。首先，在系統開始前毋需等待，如時下所用的個人電腦一般，因爲在打開電源後，不久即可使用ＳＴＢ。因爲ＳＴＢ完全適於家中電子裝置，不需要系統開始後，令使用者等待。

當打開ＳＴＢ時，ＳＴＢ下載且開始執行所需物件。使用者所需要等待的時間，正爲開始下載這些物件所需要的時間。下載物件所需要的典型時間爲幾微秒至幾十微秒。因此由適當的使用者介面，可使此時間顯得不具意義。然後，但如用於處理系統之動作處理，所需要的物件與伺服器動作同時下載。

當同時執行多個應用程式時，因爲如用於伺服器1的多個計算來源並不在客體2中製備，亦產生一些限制。例如，當由搜尋應用程式以觀賞電影時選擇一ＶＯＤ服務，一當開始看電影時，由搜尋應用程式所佔的來源（記憶體）可作爲電影應用程式。然後，當再需要搜尋的應用程式時，可再次下載這些來源（管理記憶體的物件）。

（請先閱讀背面之注意事項再填寫本頁）
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（ 14）

”需要時的時間”爲當對一物件傳送訊息時所需要的時間。即當最先下載的物件送一訊息至另一物件時，接收該訊息的物件已下載。然後，送下一訊息的物件可事先由使用交互相關的物件及參考關係下載。經由與應用程式之執行同時執行此動作，然後由訊息通訊在下載時所產生延遲可變的較小。所以此項處理增加了下載的效率。

執行環境１２，２２亦爲物件１５，２５的組合，因此如同用於應用程式１１，２１之物件１４，２４的操作亦可能進行。因此，如果物件１５，２５，可備製特用於已應用程式１１，２１中控制下載序列的共用物件。依此方式，使用增加下載，上列物件所使用的物件下載序列可指定爲適於特定應用，且使用者需要等待的時間可達到最小。

用於從伺服器至客體２下載的功能，核對物件之執行環境相容性的附功能，及用於架構執行環境的功能爲實現此系統所需要者。此視爲在此提供的所有裝置（客體２）必需要有的基本功能。在此說明書中，此稱功能稱爲共同標準。ＯＳ等執行環境之ＡＰＩ可應用此共同標準自由擴充。然後，此最小應用與此擴充可允許ＡＰＩ從此編輯各種類型之應用。

例如，應用圖６中的系統，其獨立ＡＰＥ的ＯＳ可在各伺服器１－１及１－２，和客體２－１，２－２中操作。即用於應用程式２１－１的ＡＰＩ２３－１（ＡＰＩ＃１）在客體２－１中架構，因此對應至執行環境

（請先閱讀背面之注意事項再填寫本頁）

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（ 15）

２２－１。而且，用於執行環境之應用程式２１－２的ＡＰＩ２３－２（ＡＰＩ＃３）在客體２－２中架構。

因此之故，對應這些ＡＰＩ的ＡＰＩ已在將程式下載至客體２－１，２－２的伺服器中製備。在此實施例中，使用於應用程式１２－１的應用程式１１－１的架構使用於應用程式１１－１的 ＡＰＩ１３－１。此ＡＰＩ－１被視爲對應客體２－１之ＡＰＩ２３－１（ＡＰＩ＃１）的 ＡＰＩ。

同樣地，ＡＰＩ１３－３（ＡＰＩ＃３）對應應用程式１１－３，其在執行環境１２－３中形成。此ＡＰＩ１３－３對應客體２－２的ＡＰＩ２３－２（ＡＰＩ＃３）。

提供物件１５－１Ａ至１５－１Ｃ，１５－３Ａ至１５－３Ｃ，２５－１Ａ至２５－１Ｃ及２５－２Ａ至２５－２Ａ作爲在伺服器１－１，１－２及客體２－１及２－２中對應此共同標準的物件。結果，可從伺服器１－１，１－２中依據共同標準陣列適當地從伺服器１－１，１－２中下載。

爲了使進入單一ＯＳ的客體之ＯＳ標準化已在此領域中加以定向。無論如何，由於如上所定義的共同標準，且因只在伺服器側提供對應各客體ＡＰＩ的ＡＰＩ，因此不再需要決定另一項標準。

由於不需要訂定一ＯＳ標準，實現包含應用程式之系統服務的物件可與ＯＳ獨立架構。即用於某執行環境的軟

（請先閱讀背面之注意事項再填寫本頁）
裝
訂
線
經濟部智慧財產局員工消費合作社印製
本紙張尺度適用中國國家標準（CNS）A4 規格（210×297 公釐）

五、發明說明（ 16）

體寫入可對另一使用遷移之分開的執行環境自動再架構。在傳統系統中不提供此項功能。例如，對ＵＮＩＸ的軟體寫入如果沒有再寫入的話則無法於視窗工的作台上操作。爲了使用應用位階軟體實現此功能，需要有可消除此軟體相關的軟體。但是，使用此方法，使得與物件無關，而用於實現包含裝置驅動器的軟體時間成爲可能。

經由此種方式下載物件，如圖７所示當需要時可改變客體２的物件，即可從客體中除去現存的物件，且從伺服器中下載新物件。

例如，應用圖７的實施例，客體２之應用程式２１的物件２４變得不需要，因此加以除去。然後，從伺服器１向客體２下載新且需要的物件１４Ａ。

依此方式，下列事項變得有可能實現。

（１）可更新軟體。如，當在硬體控制的軟體中發現錯誤，可移除此物件，且可以新物件更換。對於非電腦專家之一般消費者可使用配備此電腦的家用電子裝置，且用於更新用於安裝器在某些電腦中使用的軟體並不適用。所以本發明之軟體更新相當有效。

（２）生產循環可延長。如，電視接收機的模型每年都在改變，但一般消費者無法每年買新的電視接收機。但是，應用本發明的系統，可提供使用者最新的軟體功能，而不必改變電視接收機，而由於軟體功能擴充導致接收機模式改變也變得沒有必要。此亦可用於ＳＴＢ。

（３）當使用介面需要時，可隨著改變。例如，當使

（請先閱讀背面之注意事項再填寫本頁）
裝
訂
線
經濟部智慧財產局員工消費合作社印製
本紙張尺度適用中國國家標準(CNS)A4規格(210×297公釐)

五、發明說明（17）

用者開始使用該裝置時，可提供使用者親切表單，由此，然後當使用者尚未習於此裝置時，可改變至更直接操作的使用者介面。但是，兩程序不需要帶至客體側。而且，當時符合使用技術的使用介面可帶至客體側。依此方式，可有效地限制客體來源。

而且，如需要，如圖8所示，一下載物件可擴充客體2的物件，在圖8的實施例中，對客體2之應用程式21－1的物件24－B產生接收新時間的執行環境22－2。然後，從執行環境22－1中遷移需要的物件25－1A，25－B，因此在執行環境22－2中改變物件25－2A，25－2B。其他必需要的物件25－1C及25－1D亦向執行環境22－2遷移。

對於應用程式21－2，遷移應用程式21－1的物件21－1B以變成物件24－2B。

依此方式，例如，在客體產生一新的執行環境，其用於即時規畫中所需要的擴充，且必需要的物件遷移至新環境。因此物件可接收即時規畫時間，而不需要進行任何改變。

因此之故可得到下列結果。

（1）可處理新功能，不需要對應用程式物件加入任何改變。因此應用程式壽命延長，且改進其再用性。在相關的方法中，執行環境的改變意謂著應用程式的再寫入，因爲用於執行環境的獨立碼包含在應用程式中。

（2）考慮包含在設備中的應用程式，除非裝置的模

（請先閱讀背面之注意事項再填寫本頁）
裝
訂
線
經濟部智慧財產局員工消費合作社印製
本紙張尺度適用中國國家標準(CNS)A4規格(210×297公釐)

五、發明說明（ 18）

型並沒有很大的改變，否則對包含使用者介面的高功能控制軟體爲冀望再使用的部份，或者只由使用現存碼擴充軟體功能，以縮短發展周期。但是，如果該部份包含與執行環境相關的介面，包含再使用的工作變得複雜。應用至今所使用的方法，並沒有任何方法可使用該部份，但應用本發明的方法，此自動執行此工作，或使工作量達到最小。

可應用下列式應用本發明的系統。

（１）當應用程式之可信度低時，由下載記憶體保護功能，而防止整個系統停止。

（２）由服務提供者所提供的各項服務，可在各ＳＴＢ販賣機中改變。例如，提供電影公司Ａ之影影的服務提供者可依據那一影像爲公司Ｂ之ＳＴＢ或公司客體之ＳＴＢ所接收，而同時物件將傳輸的影像特徵。

（３）系統的處理方法可依據送至爲使用者所請求的ＳＴＢ 或影像品質之視聽壓縮方法而加以改變。例如，調整影像品質所使用的方法不同於ＭＥＰＧ資料及ＴＰＥＧ資料，因此需要改變系統的處理方法。但應用本發明的方法，依據數據格式選擇所需要的處理方法。

因此對客體２不需要事先提供不同的功能，因爲用於系統的大部份功能可從伺服器１中下載，且因此客體只需要提供所需要的最小功能即可。圖９示本發明系統之客體２所處理的最小功能。在客體２中形成用於裝置驅動器的執行環境２２－１，用於系統物件的執行環境２２－２及用於執行環境的執行環境２２－３作爲最小功能。

（請先閱讀背面之注意事項再填寫本頁）
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（19）

需要事先存在的裝置驅動器爲物件２１－Ａ，其作爲用於處理單元的輸入驅動器，作爲管理時間之時間驅動器的物件２４－１Ｂ，及作爲控制顯示之幕驅動器的裝置２４－１Ｃ。系統物件爲作爲管理時間之輸入處理器的物件２４－２Ａ，作爲管理動作之自動載入陣列的物件２４－２Ｂ，作爲管理記憶體之記憶體管理器的物件２４－２Ｃ。可從伺服器１中下載更高的功能裝置驅動器或相同的物件。

圖１０示適於從伺服器中送出之應用程式（視訊，遊戲，購物等）的客體環境之動態結構。在圖１０中，用於購物的執行環境２２－４架構在客體２－２上，因此可從伺服器１中下載購物應用程式１１－２。而且，當客體２－２從購物應用程式改變應用程式至電影應用程式時，在客體２－２上架構用於電影應用程式２１－３的執行環境２２－３，且從伺服器１中下載電影應用程式１１－１。

例如，可考慮下列處理。

（１）當使用者選擇電影時

此時，從伺服器１－２中下載搜尋應用１１－３，如客體２－１之執行環境２２－１，使選擇所需要的電影，且用於視窗管理，且用於管理來自使用者等的輸入被下載，作爲物件２５－１，在需要的環境中用於搜尋應用１１－３。

（請先閱讀背面之注意事項再填寫本頁）
裝
訂
線
經濟部智慧財產局員工消費合作社印製
本紙張尺度適用中國國家標準(CNS)A4規格(210×297公釐)

五、發明說明（ 20）

（２）當使用者喜觀電影時

此時，用於視訊列管理，預讀緩衝管理的資料及ＶＣＲ功能的物件１５－１從伺服器１－１的執行環境１２－１下載，至如客體２－２的執行環境２２－３，如物件２５－３一般。

在本發明的系統中，導入圖１１的特徵結構，以由下載呈現客體的執行環境。當從１向客體２下載物件時，核對此特徵結構，且在物件下載處，架構所需要的執行環境。

首先，在第一相（妥協相中），對於伺服器１之共用物件空間及客體２之共用物件空間的物件遷移的可能性，伺服器１與客體２妥協。在第二相中（遷移相），然後，實際傳送物件。

物件遷移爲共同位準之程序，其使用物件內部資訊，且如果資訊使用與此物件有關的物件傳送計算來源。由稱爲說明器的共位準物件表示物件的內部資訊。實際上，說明器保留管理物件的共用物件名稱。一典型的說明器硬體用於該物件之管理記憶體磁區的共用物件名稱，執行說明或三個物件之控制的共用物件之名稱（規劃器）及管理對物件給定名稱的共用物件之名稱等。

在第一相（妥協相）核對遷移物件的可能性。在某些共用物件空間中，不需要物件遷移。例如，如果在共用物件空間中遷移物件（裝置驅動器），以管理一裝置驅動器，如果在客體２中實際上不存在硬體，則此遷移變得毫無

（請先閱讀背面之注意事項再填寫本頁）

裝 訂 線

經濟部智慧財產局員工消費合作社印製

本紙張尺度適用中國國家標準(CNS)A4規格(210×297公釐)

五、發明說明（ 21）

意義。而且用於虛擬記憶體管理結構的物件之共用物件管理記憶體磁區無法管理執行環境區段，甚至在物件遷移處虛擬記憶體結構不存在處，執行遷移亦無法達成。因此備製下列方法，以用於遷移陣列。

Feature * Descreiptor：：Can Speak（Feature * P Feature）

在此方法中，特徵結構通過伺服器1作爲變數。結果客體2送一特徵結構回到伺服器1中，指示在客體2中有可能接受。然後，伺服器1核對從客體2回來的特徵結構，因此可知在那一類中包含下列共用物件空間的相容性所給定者。

相容性分類爲：完全相容，部份相容及不相容。

完全相容指甚至在遷移之後物件仍可完全執行。部份相容指在遷移後對物件執行需加上某些限制。不相容指在遷移後，物件無法持續執行。

在不相容下無法執行物件遷移。部份相容時，由使用者決定是否要遷移。實際上，一例外的情況送回使用者，且使用由例外處理規程決定。在完全相容或部份相容的例子中當物件遷移可執行的時候，依據先前送回來的特徵結構之內容執行遷移。

在妥協相之前，由下一操作在說明的共用物件空間處產生一空說明器。

（請先閱讀背面之注意事項再填寫本頁）

裝 訂 線

經濟部智慧財產局員工消費合作社印製

本紙張尺度適用中國國家標準（CNS）A4規格（210×297公釐）

五、發明說明（ 22）

Descriptor： Descriptor（ ）

在前一Can Speak方法可向前推進至此說明器。此時，基於特徵結構的資訊，執行：產生所需要的共用物件，產生參考項，且列出所需要的資訊。

在第二相之共用物件的處理中爲對應傳送說明器之共用物件的遷移或傳送。此處共用物件的遷移爲使此共用物件進入客體2的共用物件空間中，即從說明器中得到參考。而且在客體說明的共用物件空間處呈現在共用物件內傳送資料的共用物件機構之傳輸，該資料作爲用於共用物件之訊息（其參考說明器）。

與共用物件之遷移及傳輸有關的操作使用由妥協相所得到的特徵結構在此第二相（遷移相）中執行。

在遷移相中，共用物件的實際轉換及傳送係應用下列方法動作。

Descriptor & Descriptor：：Operator＝（Descriptor& So urce）

說明器分類一摘要類，其定爲與共用物件之遷移及傳送有關的共同陣列。使用該來源參考說明器之內容與此說明器相抗衡。

實際上的程序可定義爲說明器分類的次分類。

（請先閱讀背面之注意事項再填寫本頁）
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（ 23）

下一方法爲與共用物件之傳送有關的方法。此陣列主要爲一遷移器所使用（遷移器爲一包含在共用物件空間內的共用物件，其用於執行物件遷移）。

Canonical Contxst & Contex t：：as Canonical 1 ）

轉換一與機器相關的文書結構成與機器不相關的形式。當特徵結構指示不可能執行文書之直接轉換時，則執行此陣列。

. Context & Context：：operat or
= （Context & source）
. Context * Context. . oper ator
= （Canonical Context & So urce）

使用者來源所參考的相容啓動爲此所參考的現行相容。

轉換一機器相關區段結構成爲機器不相關形式。當特徵結構指示不可能執行相容之直接轉換時，執行此陣列。

Segment & Segment：：operato r
= （Segment & source）
Segment & Segment：：opera tor
= （Canonical Segment & So urce）

（請先閱讀背面之注意事項再填寫本頁）
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（ 24）

現正爲此所參考的區段使用者來源所參考之文書啓動，且抗衡記憶體中必需要的區域。

圖１爲特徵結構的配置。如圖１１所示，在項目中說明物件說明及環境說明的指標。

在由物件說明之指標指示結構處，說明由環境說明的指標及此物件的來源要求所指示結構之同一結構的指標，即物件名稱。

而且，由暫存器說明之指標所指示的結構處，說明環境名稱，說明硬體的來源資訊，環境的來源要求，及構成執行環境的共用物件表列。

特徵結構之內容的特定例如下。

（１）與物件有關的資訊。

即時可能性

所需要的處理器量

（２）與共用物件有關的資訊

硬體共用物件

＊處理器型式

＊資料格式

區段共用物件

＊尺寸

＊可擴充性，可壓縮性

＊管理原則

文書共用物件

（請先閱讀背面之注意事項再填寫本頁）
裝 訂 線
經濟部智慧財產局員工消費合作社印製
本紙張尺度適用中國國家標準(CNS)A4規格(210×297公釐)

五、發明說明（ 25）

＊暫存器資訊

＊暫時變數之數目

＊處理器狀況

＊郵差物件

＊訊息線索長度

＊變動處理器訊息數

＊所需要的外部郵差

＊訊息傳送方法

外部郵差共用物件

＊訊息線索長度

＊變動處理器訊息數

＊陣列

規劃之共用物件

＊物件狀態

＊規劃陣列

與管理相關的共用物件

＊所擁有的外部名稱數

如上所述，當各客體具不同ＯＳ時，從此特徵結構及伺服器中決定各客體之物件Ｓ，且然後，下載對應這些ＯＳ的物件。

圖１２示本發明之資料處理系統所使用的系統結構之例。此系統之核心３１包含Micro Virtual Machine（MVM）３１a（第一執行機構）及Micro Kernal（MK）３１b（第二執行機構）。MVM３１a說明且往後將說明的中間

（請先閱讀背面之注意事項再填寫本頁）
裝
訂
線
經濟部智慧財產局員工消費合作社印製
本紙張尺度適用中國國家標準(CNS)A4規格(210×297公釐)

五、發明說明（ 26）

碼（I碼），且如需要可使用MK３１b的功能呼叫個人物件（系統物件）。

例如，在圖１２中核心３１之外的部份可使用上述方法從伺服器１中下載。

如需要，此系統的動態將Ｉ碼編譯爲本地碼（二位元碼，機器碼）。已在本地碼中編譯的物件只可在此例中執行，使用ＭＫ３１ｂ的功能呼叫個人物件３３（二位元碼產生機構），且提供服務予應用３５。

包含圖１２中所示核心的ＭＶＭ３１ａ及ＭＫ３１ｂ爲裝置驅動器物件（裝置驅動器）３２及個人物件（個人組件物件＿３３所包含，其又復爲一分類系統（分類庫）３４所包含，該分類系統備用於執行應用程式，此應用程式又爲應用程式３５所圍繞。

個人物件層３３允許此系統提供不同的ＯＳ或虛擬機器。例如，經由使用用於ＢＡＳＩＣ程式的個人物件編譯ＢＡＳＩＣ程式所得到的中間碼以執行ＢＡＳＩＣ程式。

在本發明的系統中，編譯程式成Ｉ碼（中間碼）以管理物件方法，因此得到高度可攜性。雖然在被解釋及執行的推測上不設計Ｉ碼（當解釋程式時執行程式），但如需要的話，設計以編譯成本地碼。但是，當由於不同限制當Ｉ碼的編譯變得相當困難時，ＭＶＭ解釋且執行Ｉ碼。因此可省略伴隨虛擬機器執行之處理速度的即時處理性或無效率的耗損。

Ｉ碼包含兩高度適用的指令組（ＯＰ＿Ｍ，ＯＰ＿Ｒ

（請先閱讀背面之注意事項再填寫本頁）
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（ 27）

），在圖２２中將再予說明，因此可給予高度的交互可操作性。這些指令組的結構意義與ＭＫ３１ｂ的介面具強烈相關性。即指令組受圖１２所示本發明系統結構的強烈影響。因此本發明系統具高度可攜性及互相操作性，而與本地碼的假設無關。

其次，說明用於ＭＶＭ３１ａ及ＭＫ３１ｂ的方法。首先，訂定爲 ＭＶＩ３１ａ及Ｉ碼格式所假設的資料結構。

圖１３示圖１２中ＭＶＭ３１ａ及ＭＫ３１ｂ的邏輯結構。 ＭＶＭ３１ａ及 ＭＫ３１ｂ的邏輯結構基本上相同，兩者均勻動作文書４１，訊息框４２，及執行機構４３。但是ＭＶＭ３１ａ支援使用Ｉ碼執行動作，ＭＫ３１ｂ支援使用本地碼的執行動作。

在圖１３中，動作文書４１指向現正執行的文書（在後面的圖１４中說明），訊息框４２指向用於包含核心３１之ＭＶＭ３１ａ及 ＭＫ３１ｂ之訊息的主要部份。

在ＭＫ３１ｂ的例子中，訊息主要部份之存在主要視配置系統而定，看此系統分配於作爲儲積框的記憶體，或分配予一堆疊（ｈｅａｐ），或分配予多個ＣＰＵ的暫存器而定。另一方面，在ＭＶＭ３１ａ的例子中，訊息框指向在指令碼之後的操作域。除了這些暫存器外，也需要內部暫存器，此視配置而定，但這些暫存器與本方法無關。

在圖１３中，執行機構４３執行Ｉ碼及本地碼。而且原始物件４４包含在ＭＶＭ３１的執行機構４３中，而這

（請先閱讀背面之注意事項再填寫本頁）
裝
訂
線
經濟部智慧財產局員工消費合作社印製
本紙張尺度適用中國國家標準(CNS)A4規格(210×297公釐)

五、發明說明（ 28）

些原始物件４４被編碼，以處理原始物件，以將在圖１８中再次說明。

圖１４示文書及說明器的邏輯結構。示程式之執行狀態的文書結構５１包含物件，分類，方法及共同等領域，而在域方法中提供ｉｃｏｄｅ及ｍｉｎｆ的其他領域。此文書結構５１保留ＭＶＭ３１ａ的結構，其對應ＣＰＵ暫存器，且與程式等之間的記憶體管理及通訊系統完全沒有關聯性。

文書結構５１爲一文書原始物件，如將於圖１６中說明者，各領域應用預述資訊互相連結。而且文書結構５１與核心３１的配置且高度關連性（核心３１指ＭＶＭ３１ａ及ＭＫ３１ｂ），圖１４中示不相關的部份。

在文書結構５１中重要的領域爲共同領域（共用－ｆｉｅｌｄ），此領域指向說明器結構５２。用於說明器結構５２的項目包含三群，＃ｔａｇ，文書及選擇，而個人物件３３的ＡＰＩ由此文書結構５２決定。此處＃ｔａｇ表示ＡＰＩ名稱，ＡＰＩ位址由文書及選擇器表示。

圖１５示ＭＶＭ３１ａ的整個結構。基本上，所有必需要的資訊可參考來自文書結構５１的連結。即文書結構５１連結物件６１。此物件６１包含至一對應物件６１的分類之連結（分類指標）及一瞬間區（物件相關領域）。那種資訊儲存在瞬間區，或者此種資訊之佈局爲何與物件

（請先閱讀背面之注意事項再填寫本頁）

裝 訂 線

經濟部智慧財產局員工消費合作社印製

本紙張尺度適用中國國家標準(CNS)A4規格(210×297公釐)

五、發明說明（ 29）

的配置有關。

操作2主要用於保留該方法。操作2包含名稱（分類名稱），與配置（分類相關領域）有關的部份及至I方法結構63的連結表（方法表）。I方法結構63爲方塊原始物件且包含一表頭（Header），一I碼及一變動表（變數表）。而且magic#爲MVM31a的管理數（ID）。基本上，I碼指令操作域與經此變數表項的目標物件有關。

在圖15中的灰色部份（文書51，I方法63）爲與MVM31a相關的部位，且當MVM31a轉譯且執行I碼時爲所需要的結構。

圖16示與文書結構51連結的外結構。文書結構51的物件領域指向物件61，且分類域指向操作2的分類相關域，此方法域中的icode指向I方法63的I碼。此I碼對應程式計數，且指向被執行的程式。而且，方法域的 vtable指向I方法63的變數表。

暫時域指向暫時儲存資料區域，且共同域指向上述說明的說明器52。而且，物件61的分類指標指向操作2，且操作2的方法表指向I方法63。

變數表項目爲包含型式及值的一群，該值與型式有關。

MVM31a處理如圖17所示的型式。當型式爲T_PRIMITIVE時，值域稱爲原始物件。圖18中所示的碼P_CLASS，P_BODY碼群，如碼

（請先閱讀背面之注意事項再填寫本頁）
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（ 30）

P＿INTEGER，immediate介面，碼P＿STRING，至大量儲存之位址介面儲存在值域中。圖18示件有介面名稱（列在P＿CLASS列中）的原始物件表列。圖19至圖21示圖18中的原始物件介面之例，其使用介面定義語言（IDL）方法加以說明。此IDL方法可參見" COBRAV2.0, July 1995, P3－1至P3－36"。

在圖17中，該物件的ID當型式爲T＿POINTER時儲存在值域中。此ID爲系統內唯一的值，其位置彼此不相關。然後，對應該ID的物件原始爲個人物件33所指定。

圖22示爲MVM31所說明及執行的指令組。在圖12的結構中，指令組OP＿M爲從外進入內側的指令，且指令組OP＿R爲從內側至外側的指令。

即，指令組OP＿M執行爲第一操作域的操作。此操作爲個物件33或原始物件所處理。而且，編譯器產生I碼，以更有效率地執行應用35，作爲由原始物件所處理之處理數。依此方式，整數的數學操作可爲圖18的前導原始物件所處理。

其次，圖23中指定的MK31b介面示微核（MK）31b之介面。如圖13所示MVM31a及MK31b具相同的邏輯結構。此處 MK31b處理圖22中的指令組OP＿M及OP＿R，作爲系統呼叫。

依上述方式架構的系統可應用於如圖4之客體2（2

（請先閱讀背面之注意事項再填寫本頁）
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（31）

－１，２－２）。最適應執行前述應用的執行環境，隨後可由該來自伺服器１之任意部份（而非圖１２中的核心３１）下載而在客體上架構。

如上所述，下列可視爲將下載之物件

（１）所有的應用程式

（２）裝置驅動器群（如ＭＰＥＧ驅動器，ＡＴＭ驅動器，影像控制驅動器等）用於控制爲客體所提供的來源。

（３）物件群（個人物件）對應用程式提供系統服務（如ＶＣＲ 系統管理，串列系統管理，即時規畫器，記憶體管理，視窗管理，下載控制，通訊陣列管理，執行管理等），對應用程式提供系統服務。

然後，對應用程式最適當的執行環境可由結合這些應用程式及物件群而在客體上架構。

例如，當希望執行一ＢＡＳＩＣ程式時，用於ＢＡＳＩＣ的個人物件３３及ＢＡＳＩＣ程式（應用）３５從伺服器１下載。此系統可提供用於此個人物件３３的ＢＡＳＩＣ虛擬機器。然後，下載的 ＢＡＳＩＣ程式暫時由編譯器編譯成中間碼，且由 ＢＡＳＩＣ虛擬機器執行。在應用上述方式編譯成本地碼之後，交替執行ＢＡＳＩＣ程式。

在上述的實施例中，從伺服器向客體執行預述物件的下載操作。但是，本發明亦可用於物件從預述客體向預述伺服器下載，或下載在伺服器間或客體之間的物件。

（請先閱讀背面之注意事項再填寫本頁）
裝 訂 線
經濟部智慧財產局員工消費合作社印製
本紙張尺度適用中國國家標準(CNS)A4規格(210×297公釐)

五、發明說明（ 32）

而且，在述實施例中，I 碼包含兩指令，但本發明並不只限用在此一方面。

依據本發明的資料處理方法及資料處理裝置，記憶體需核對客體是否具有將下載之應用程式的執行環境，且然後，應用程式依據此核對下載客體。因此，可簡化此客體的結構，且可降低成本。所以提供便宜的應用程式。

而且，應用本發明的資料處理裝置，需要告知伺服器將下載之應用程式的執行環境爲何，然後，基於此項告知，從伺服器中下載應用程式。此提供具簡化結構的低成本裝置，且有可能提供低成本的應用程式。

應用本發明的資料處理及資料處理方法解釋及執行轉換成中間碼的應用程式，或動態編譯中間碼，且產生二位元碼。因此當動態編譯不易執行時，可逐漸解釋及執行中間碼。而且可由予結構簡單的中間碼而架構可攜式應用。

（請先閱讀背面之注意事項再填寫本頁）
裝 訂 線
經濟部智慧財產局員工消費合作社印製
本紙張尺度適用中國國家標準(CNS)A4 規格(210 × 297 公釐)

四、中文發明摘要（發明之名稱：　　）

資料處理方法和裝置

一種用於資料處理系統的資料處理方法一種用於資料處理系統的資料處理方法，該資料處理系統包含：一伺服器，此伺服器包含：一由多個物件(object)架構的應用程式；一從多個物件架構的執行環境，用於指定該應用程式之操作；及一應用程式介面，用於指定該應用程式及該執行環境間的介面；及一客體，下載來自該伺服器之該應用程式，且該資料處理方法，包含下列步驟：

當該應用程式將下載至該客體時，使該伺服器執行一核對操作，決定該客體是否適用於該將下載之應用程式的執行環境；及

依據該核對之結果使該伺服器將該應用程式下載至至客體中。

英文發明摘要（發明之名稱：**Data processing method and device**　　）

A data processing method for a data processing system having: a server comprising an application program constructed from a plurality of objects, an execution environment constructed from a plurality of objects for specifying operations of the application program and an application program interface for specifying an interface between the application program and the execution environment; and a client for being downloaded with the application program from the server, the data processing method comprising the steps of:

having the server execute a check whether or not the client has the execution environment for the application environment to be downloaded when the application program is to be downloaded to the client; and

having the server download the application program to the client in accordance with the results of the check.

（請先閱讀背面之注意事項再填寫本頁各欄）

裝

訂

線

經濟部中央標準局員工消費合作社印製

六、申請專利範圍

1．一種用於資料處理系統的資料處理方法，該資料處理系統包含：

一伺服器，此伺服器包含：

一由多個物件構式的應用程式；

一由多個物件構成的執行環境，用於指定該應用程式之操作；和

一應用程式介面，用於指定該應用程式及該執行環境間的介面；和

一客體，下載來自該伺服器之應用程式，和

該資料處理方法，包含下列步驟：

當該應用程式將下載至該客體時，使該伺服器執行一核對操作，決定該客體是否適用於該將下載之應用程式的執行環境；及

依據該核對之結果使該伺服器將該應用程式下載至至客體中。

2．如申請專利範圍第1項之資料處理方法，其中當該客體不具有用於該將被下載之應用程式的執行環境時，一物件從該伺服器下載該客體中，在該客體中架構與該伺服器之執行環境相同的執行環境，然後，從該伺服器向該客體下載該應用程式。

3．如申請專利範圍第1項之資料處理方法，其中該應用程式的物件與該下載同時處理。

4．如申請專利範圍第1項之資料處理方法，其中該伺服器增加下載用於該執行環境的物件，該執行環境爲執

（請先閱讀背面之注意事項再填寫本頁）

裝 訂 線

經濟部智慧財產局員工消費合作社印製

本紙張尺度適用中國國家標準（CNS）A4規格（210×297公釐）

六、申請專利範圍

行該應用程式所需要者。

5. 如申請專利範圍第2項之資料處理方法，其中用於該客體之執行環境至少包含：

一下載功能，用於從該伺服器下載該客體物件；

一核對功能，用於核對該執行環境的相容性；及

一結構功能，用於架構該執行環境，

及其他需要的下載功能。

6. 如申請專利範圍第2項之資料處理方法，其中該客體包含至少：

一用於該裝置驅動器的執行環境；

一用於系統物件的執行環境；及

一用於執行環境的執行環境，且如需要的話，下載其他的執行環境。

7. 如申請專利範圍第1項之資料處理方法，其中該伺服器及該客體給定且接收用於該物件及該執行環境並包含一說明器的特徵結構，以核對該客體之執行環境的相容性。

8. 一種用於應用程式下載至客體的資料處理裝置，其配置從多個物件架構的應用程式，一從多個用於該應用程式之指定操作之物件架構的執行環境，及一應用程式介面，用於指定該應用程式及該執行環境間的介面，

該資料處理裝置包含：

核對機構，當該應用程式將下載至客體時，核對該客體是否具有用於將下載之應用程式的執行環境；及

（請先閱讀背面之注意事項再填寫本頁）
裝 訂 線
經濟部智慧財產局員工消費合作社印製
本紙張尺度適用中國國家標準（CNS）A4規格（210×297公釐）

六、申請專利範圍

下載裝置，依據該核對機構核對的結果，將應用程式下載至該客體中。

9. 一種用於應用程式下載至伺服器的資料處理裝置，其配置從多個物件架構的應用程式，一從多個用於該應用程式之指定操作之物件架構的執行環境，及一應用程式介面，用於指定該應用程式及該執行環境間的介面，

該資料處理裝置包含：

告知機構，當該應用程式將從該伺服器下載時，告知該執行環境該應用程式將下載；及

下載機構，依據該告知機構告知的結果，將應用程式下載至該伺服器中。

10. 一種資料處理裝置，包含以多個物件構成的應用程式，及以對該應用程式提供一執行環境之多個物件構成的系統物件，

該資料處理裝置包含：

第一執行機構，用於編釋及執行該轉換成中間介面的應用程式；

二位元介面產生機構，用於動態編譯該中間碼，且產生二位元碼；及

第二執行機構，用於執行該二位元碼及該系統物件。

11. 如申請專利範圍第10項之資料處理裝置，其中該第一執行機構經該第二執行機構執行該系統物件。

12. 如申請專利範圍第10項之資料處理裝置，更包含

（請先閱讀背面之注意事項再填寫本頁）
裝 訂 線
經濟部智慧財產局員工消費合作社印製
本紙張尺度適用中國國家標準（CNS）A4規格（210×297公釐）

六、申請專利範圍

顯示該應用程式之執行狀態的第一結構，及

第二結構，用於決定該執行環境之應用程式介面，該執行環境爲該系統物件所提供，

基於該第一結構及該第二結構，該第一執行機構及該第二執行機構控制該應用程式之執行。

１３．如申請專利範圍第１０項之資料處理裝置，其中該第一裝置驅動器機構具一第一指令，其由該物件之預述項所發出，用於執行爲一預述操作域所指定的操作，且一第二指令，在操作執行之後，用於將控制回復予發出該指令的物件。

１４．一種資料處理方法，用於一資料處理裝置，該資料處理裝置包含以多個物件構成的應用程式，和以對該應用程式提供一執行環境之多個物件所構成之一系統物件，

該資料處理方法包含下列步驟：

使用用於編釋及執行轉換爲中間碼的應用程式之方法，和用於動態編譯該中間碼或執行產生的二位元碼之方法以執行該應用程式。

（請先閱讀背面之注意事項再填寫本頁）

裝 訂 線

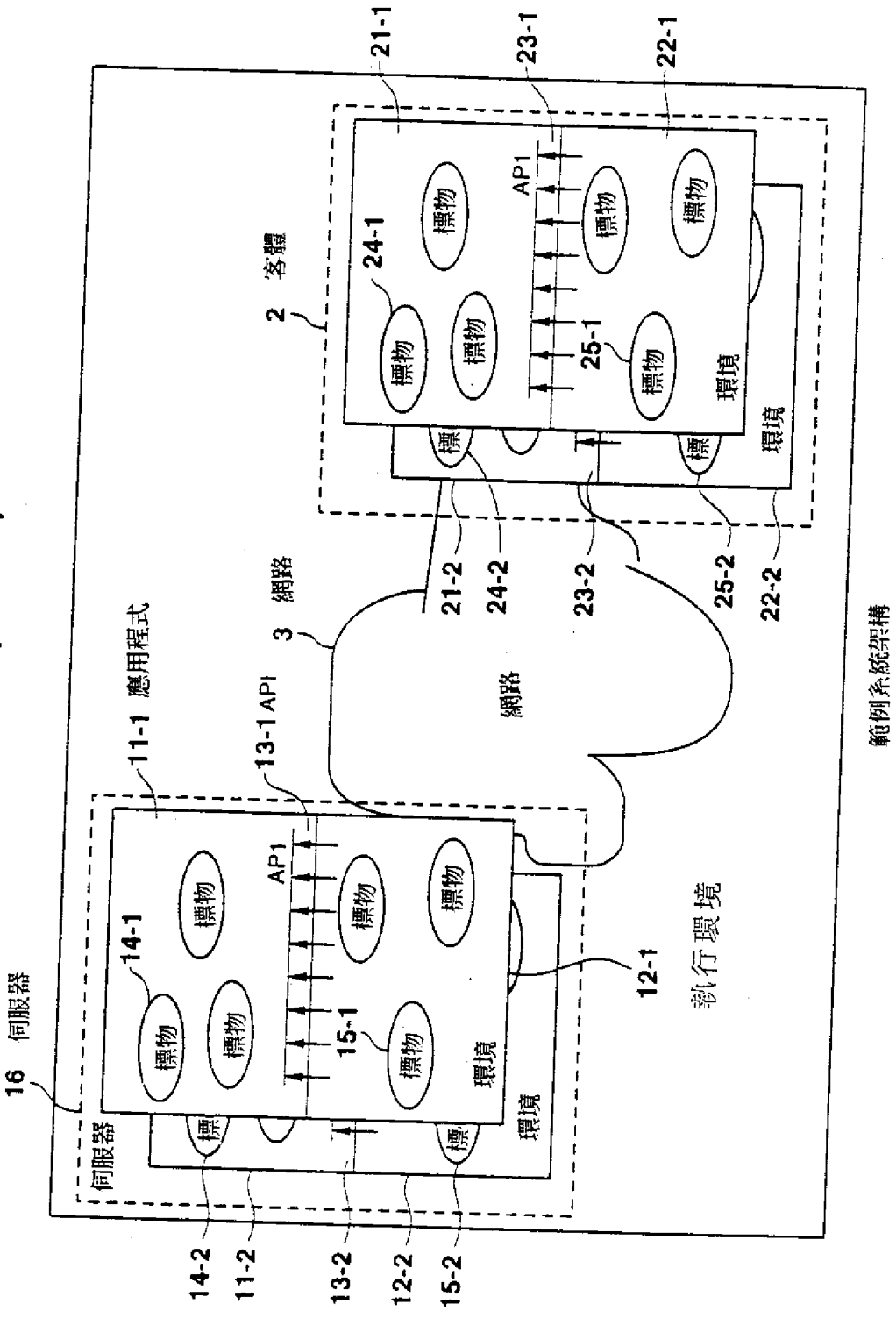
經濟部智慧財產局員工消費合作社印製

本紙張尺度適用中國國家標準（CNS）A4規格（210×297公釐）

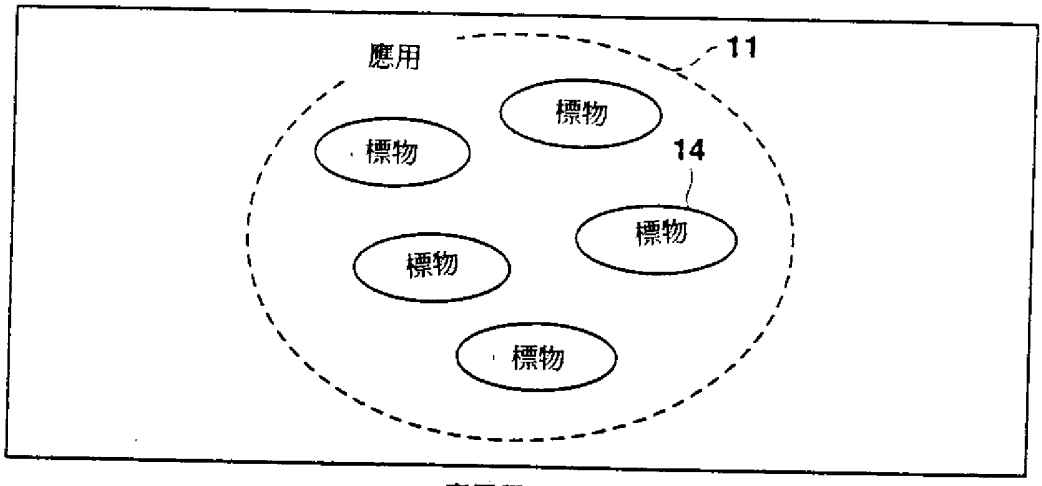
410312

725802

第 1 圖

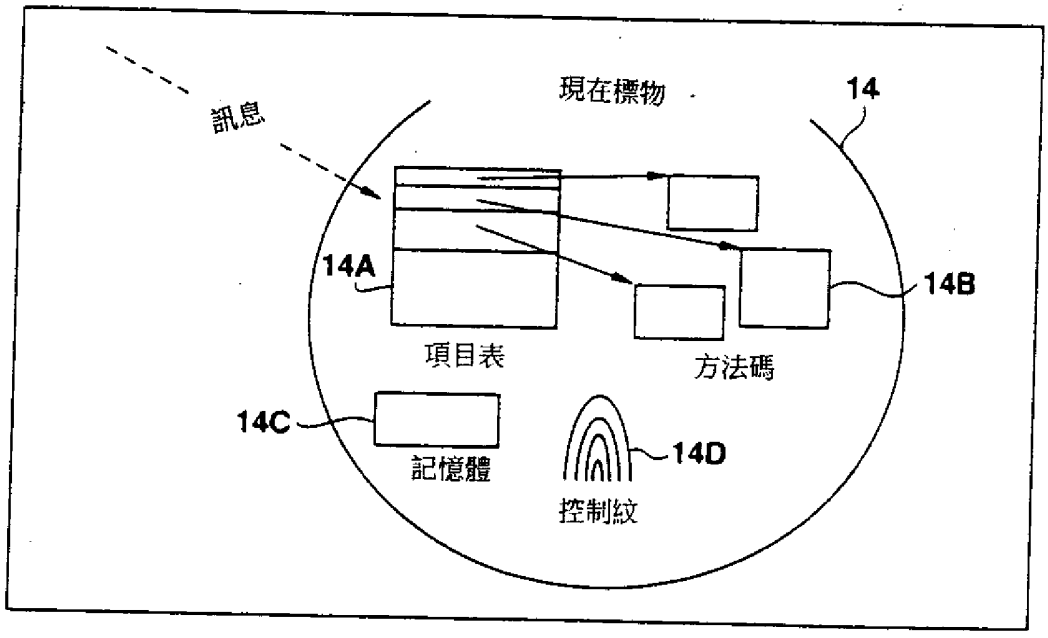


第 2 圖



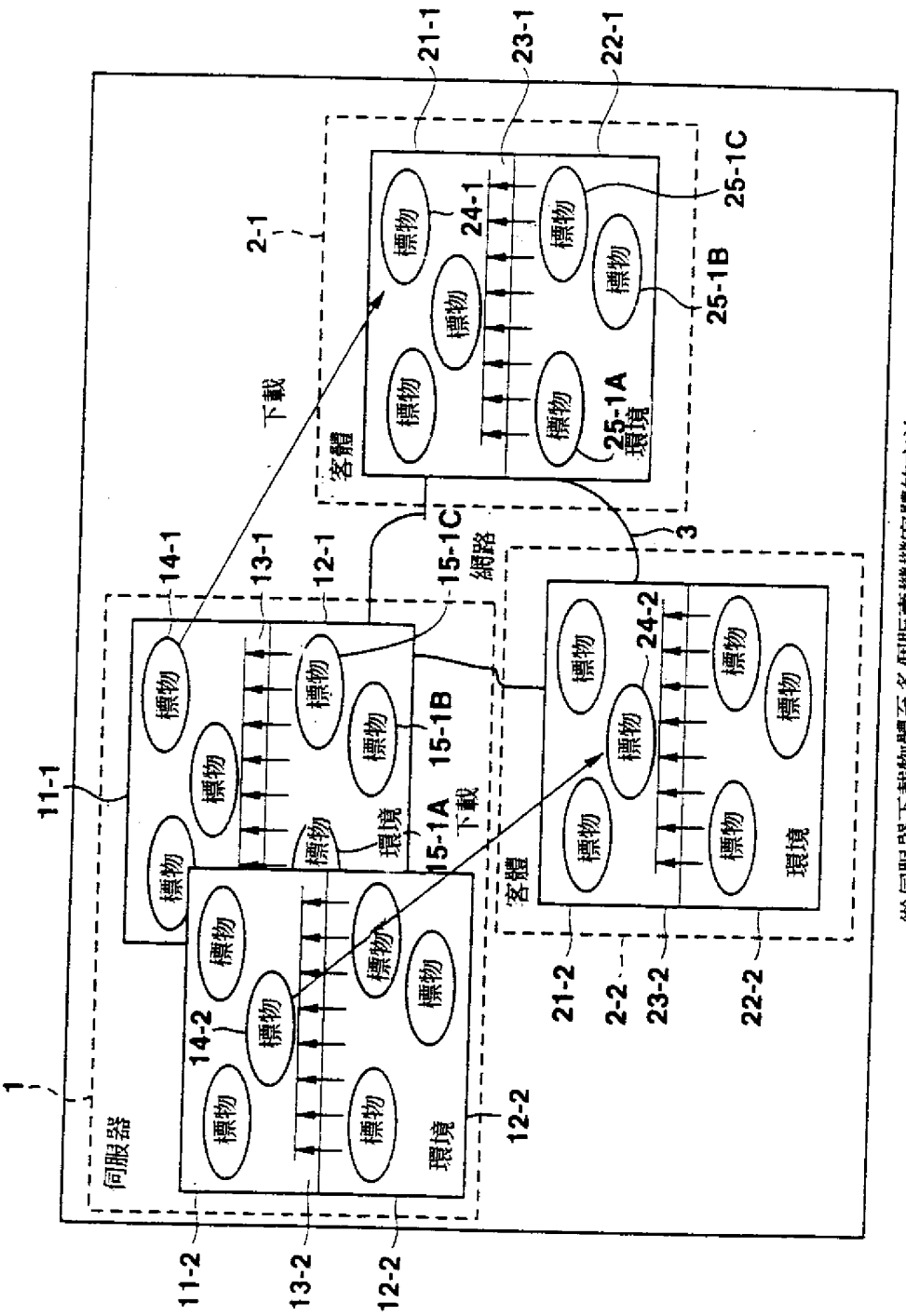
應用程式配置

第 3 圖



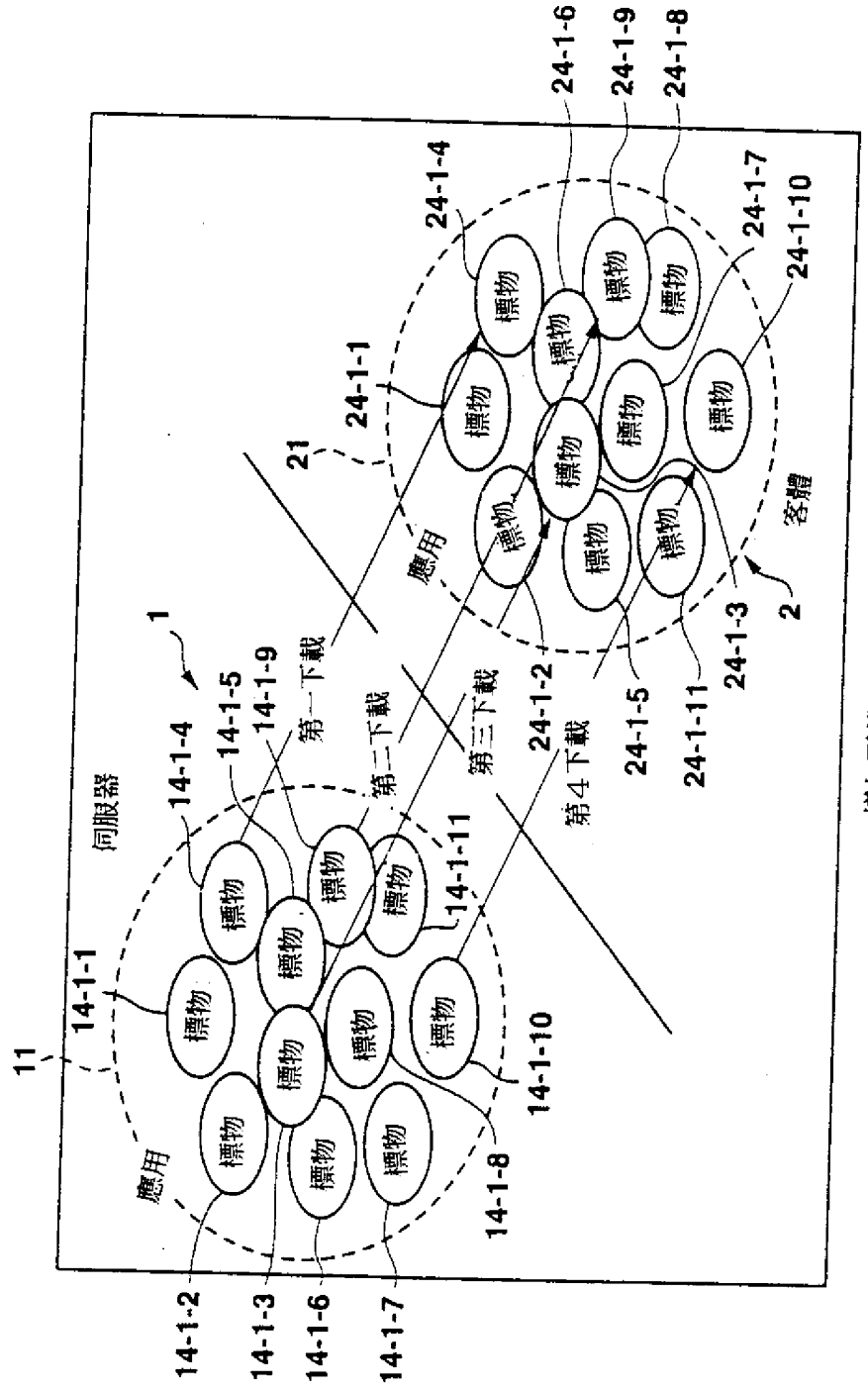
現在標物結構

第 4 圖

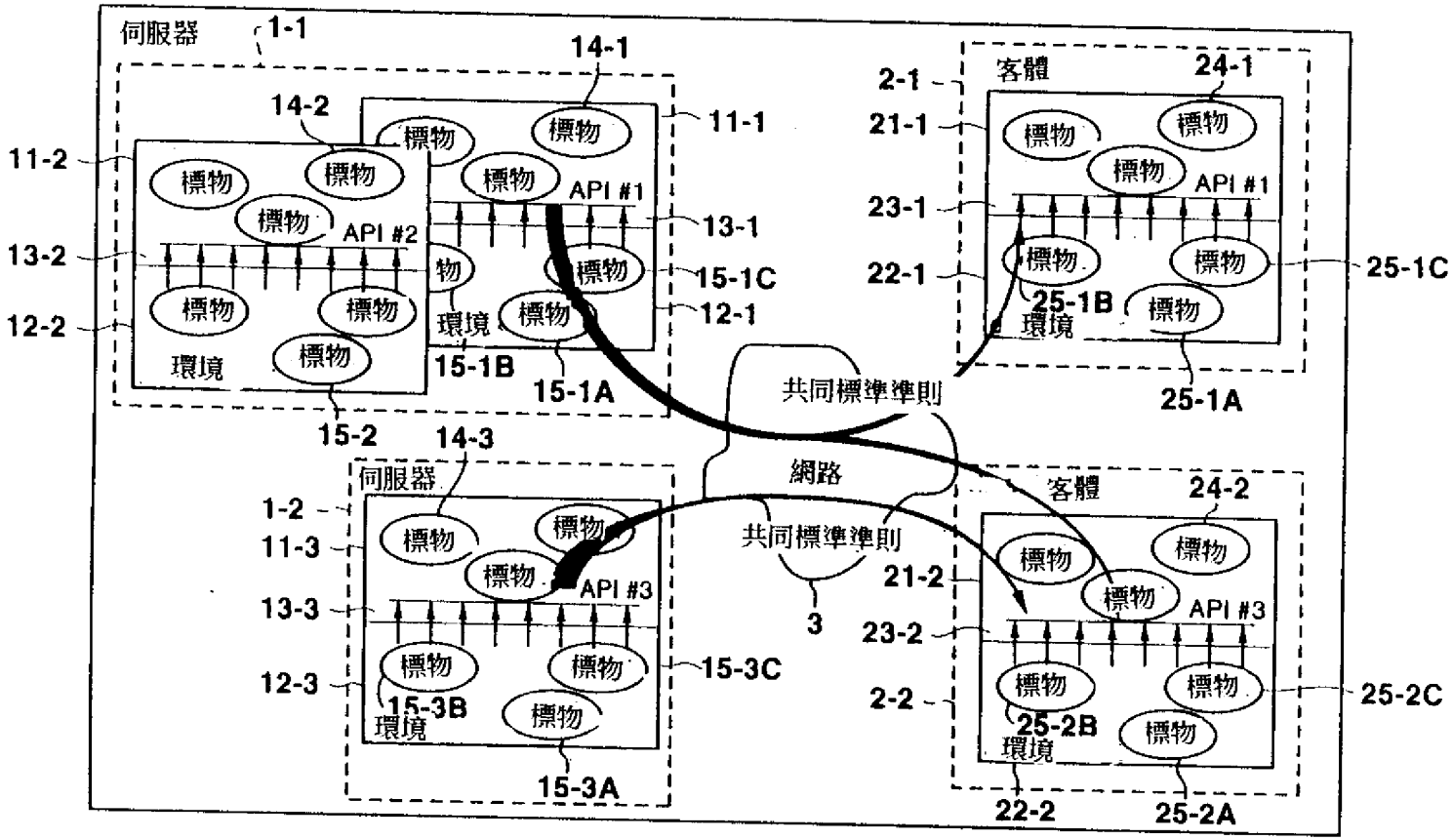


從伺服器下載物體至多個販賣機機客體的方法

第 5 圖

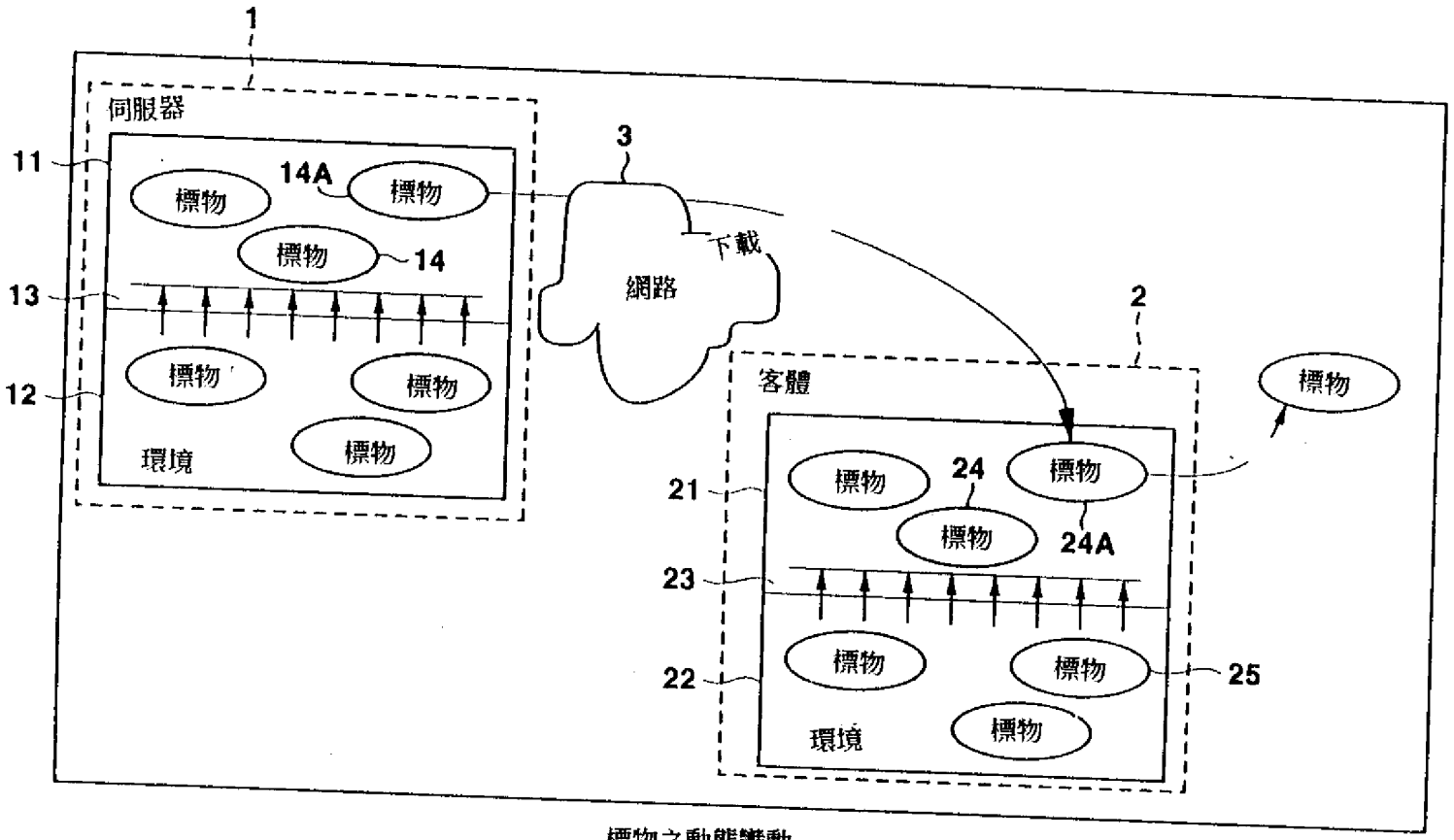

11
1
同服器
14-1-1
14-1-4
14-1-5
14-1-9
第一下載
第二下載
第三下載
第4下載
14-1-11
14-1-10
14-1-8
14-1-2
14-1-3
14-1-6
14-1-7
應用
標物
21
24-1-1
24-1-4
24-1-6
24-1-9
24-1-8
24-1-7
24-1-10
24-1-2
24-1-5
24-1-11
24-1-3
2
容體
增加下載

第 6 圖



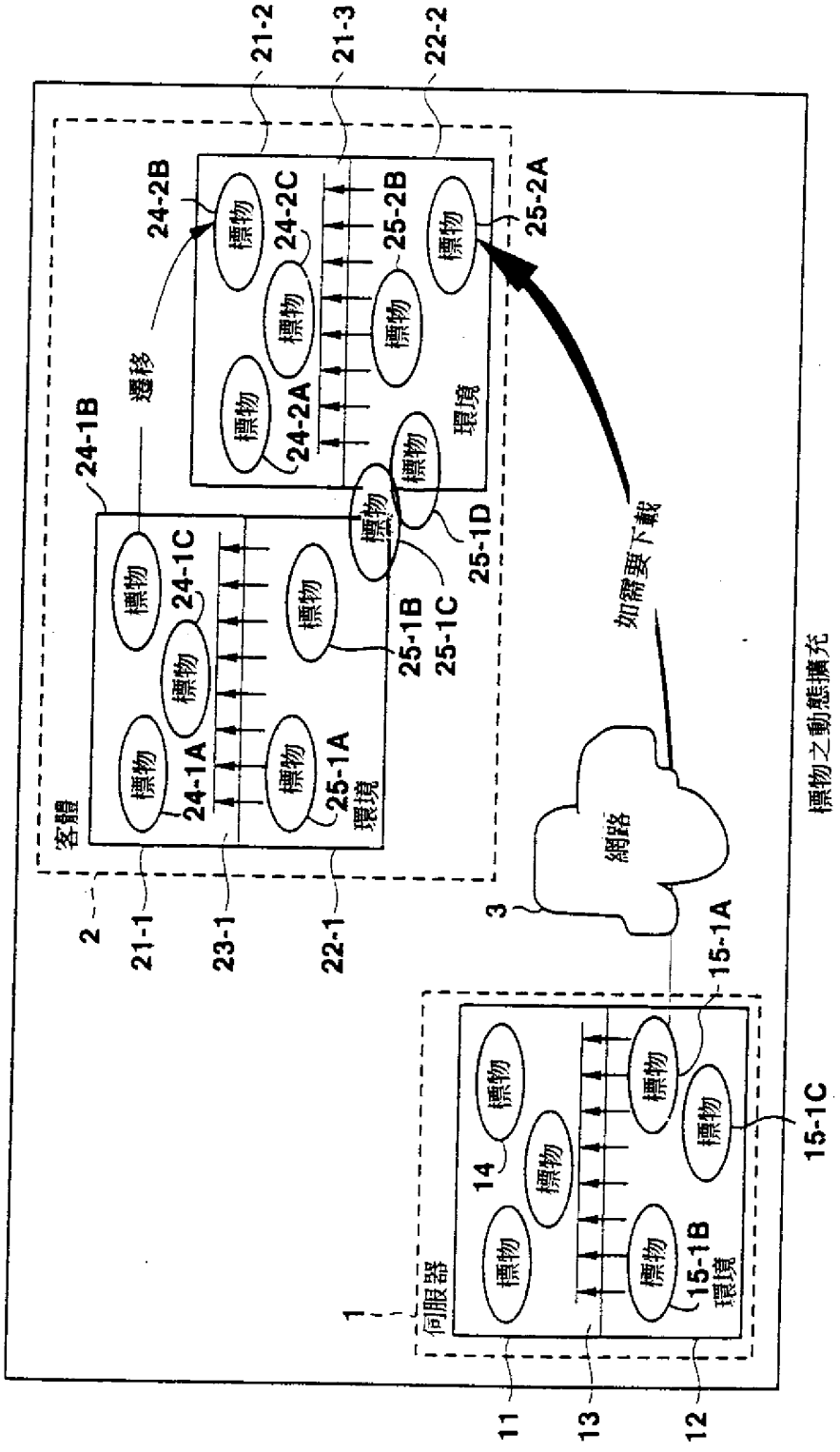
最小共同標準之標準化及擴充

第 7 圖


1
伺服器
11
14A
標物
標物
標物
14
13
標物
標物
12
標物
環境
3
下載
網路
2
客體
標物
21
標物
24
標物
標物
24A
23
標物
標物
22
25
標物
環境

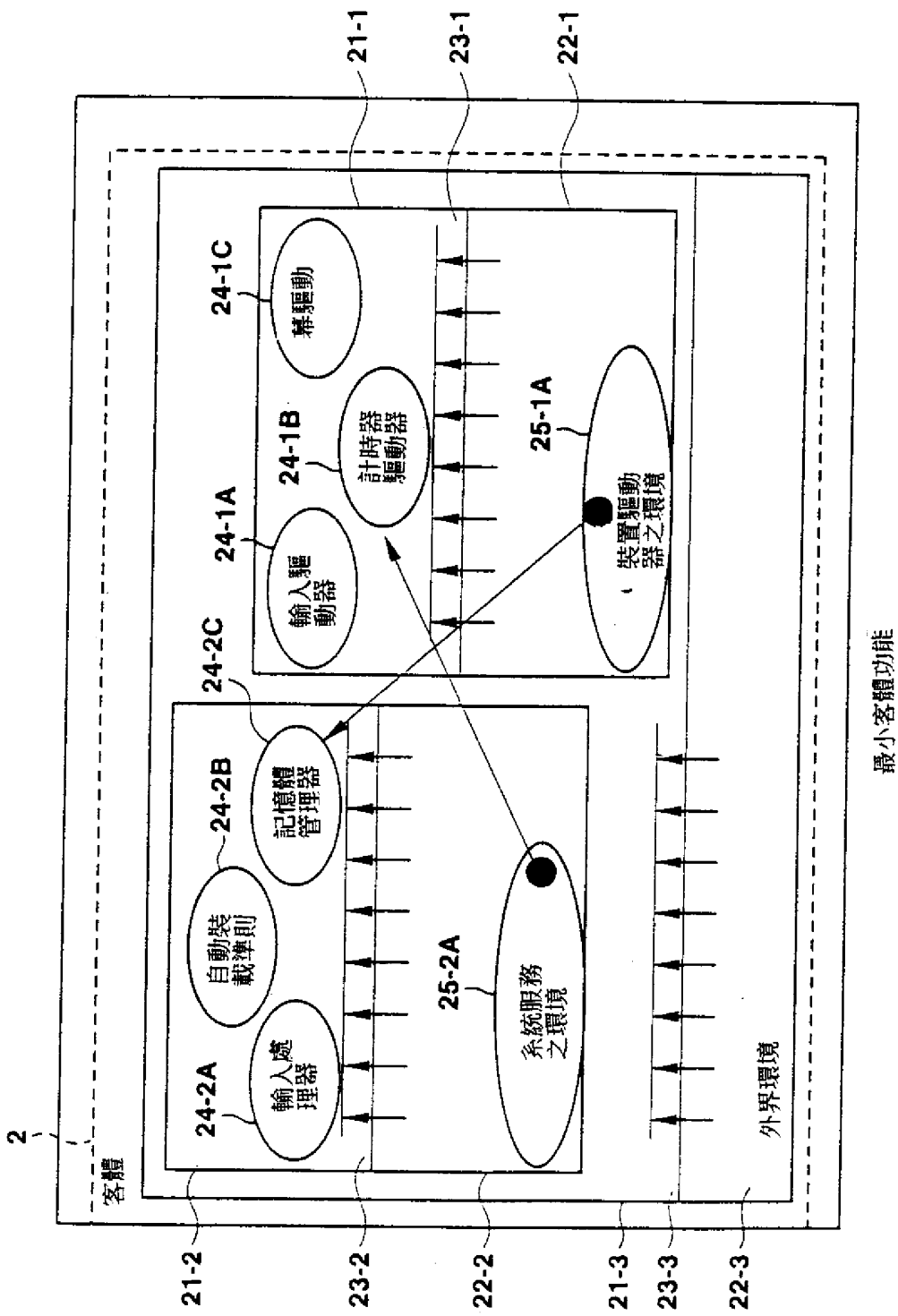
標物之動態變動

第 8 圖

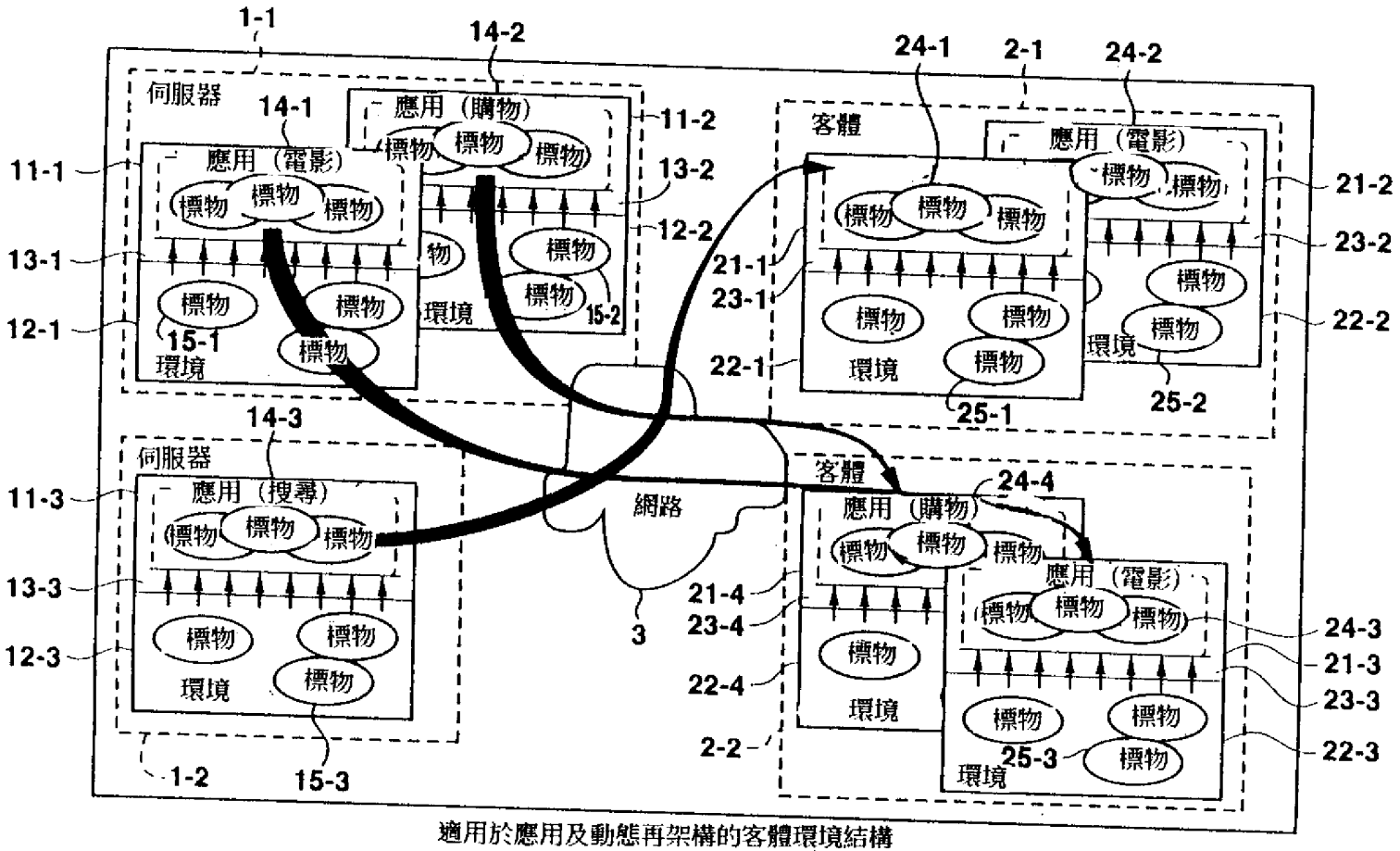

客體
伺服器
環境
標物
遷移
網路
如需要下載
標物之動態擴充
1
2
3
11
12
13
14
15-1A
15-1B
15-1C
21-1
21-2
21-3
22-1
22-2
23-1
24-1A
24-1B
24-1C
24-2A
24-2B
24-2C
25-1A
25-1B
25-1C
25-1D
25-2A
25-2B

410312

第 9 圖

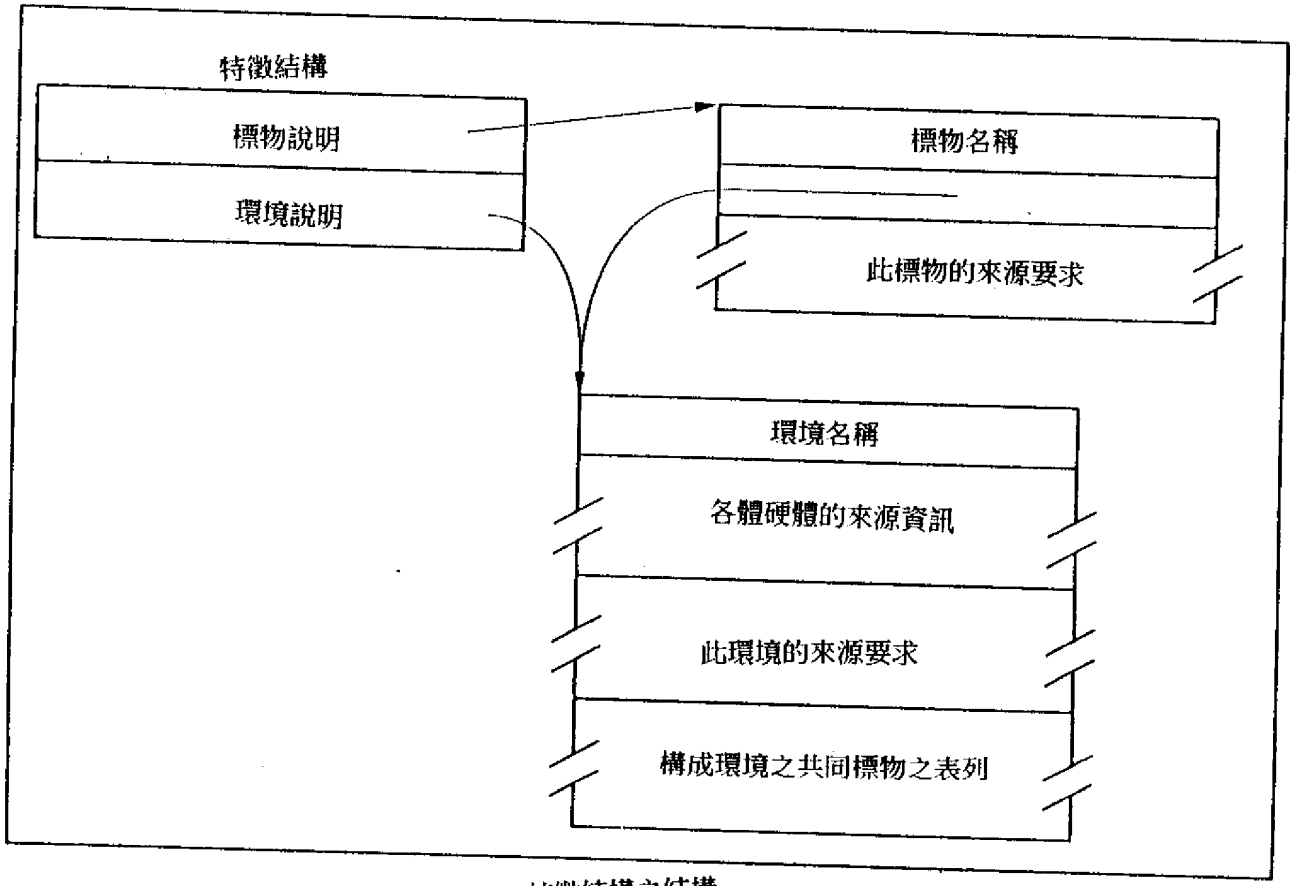

2
客體
21-1
23-1
22-1
24-1C
幕驅動
24-1B
計時器驅動器
24-1A
輸入驅動器
25-1A
裝置驅動器之環境
24-2C
記憶體管理器
24-2B
自動裝載準則
24-2A
輸入處理器
25-2A
系統服務之環境
21-2
23-2
22-2
21-3
23-3
22-3
外界環境
最小客體功能

第 10 圖



適用於應用及動態再架構的客體環境結構

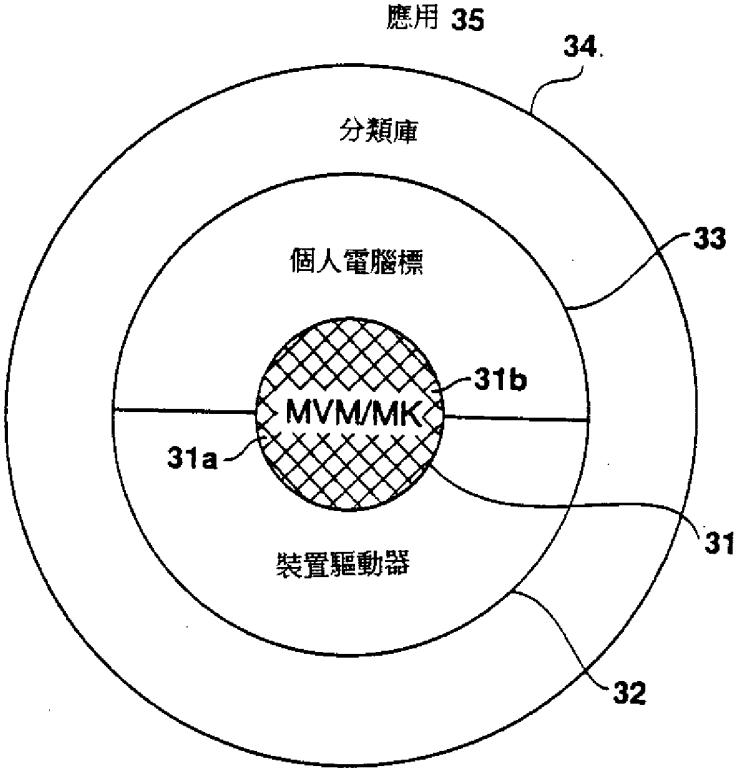
第 11 圖



特徵結構之結構

410312

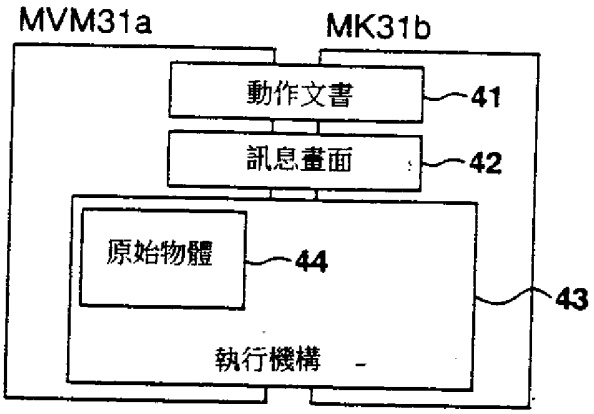
第12圖



MK： 微核
MVM： 微虛擬機器

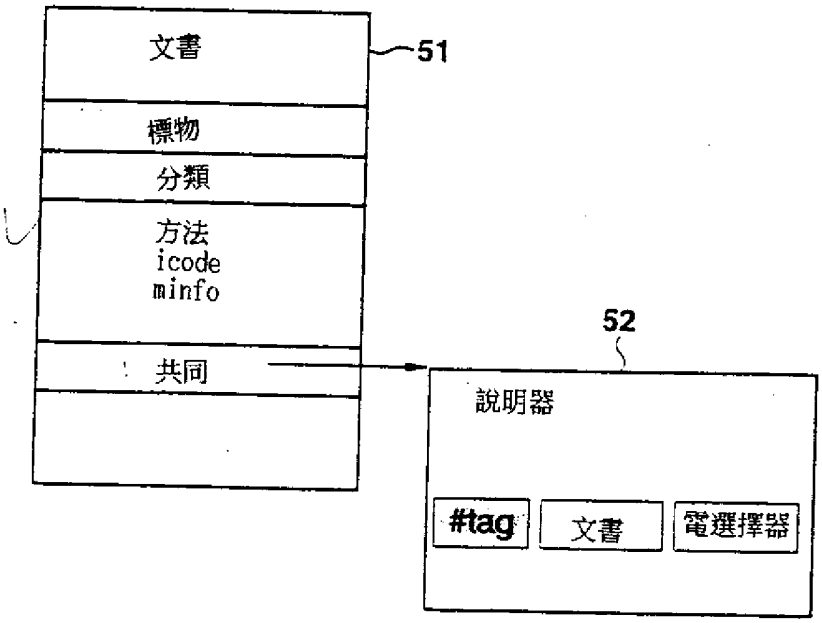
系統功能之結構

第13圖



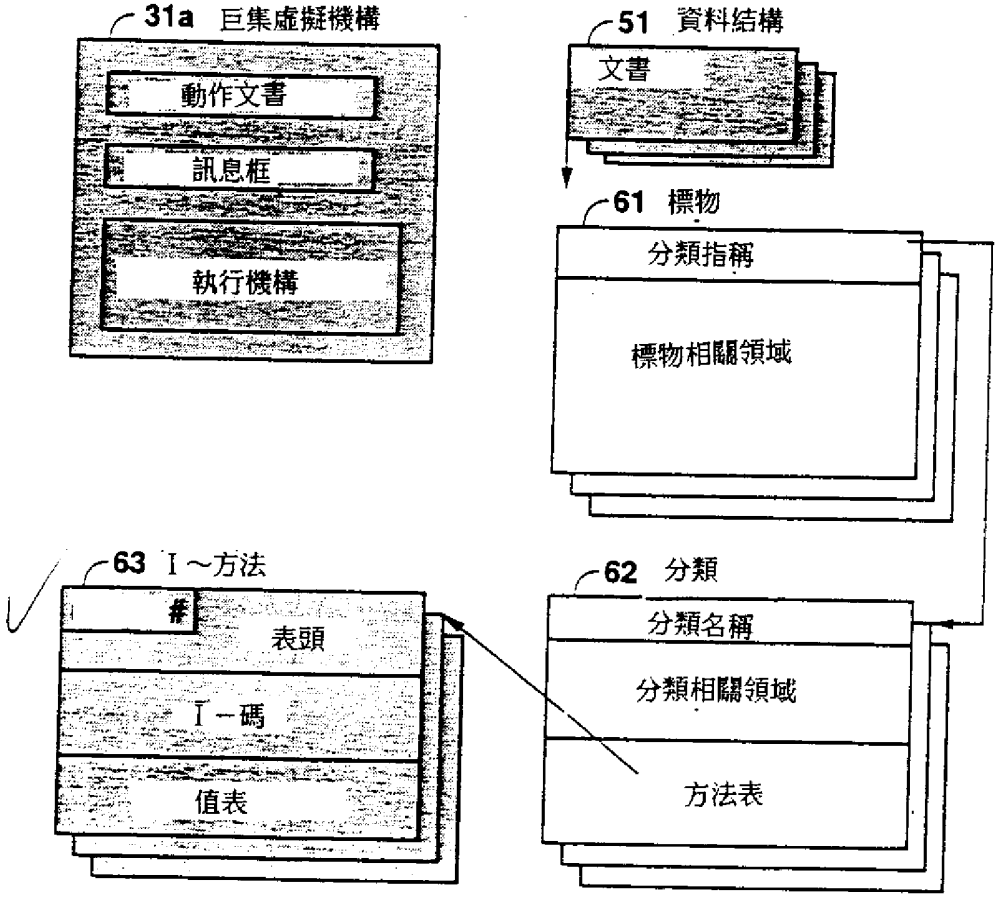
MVM及MK的邏輯結構

第14圖



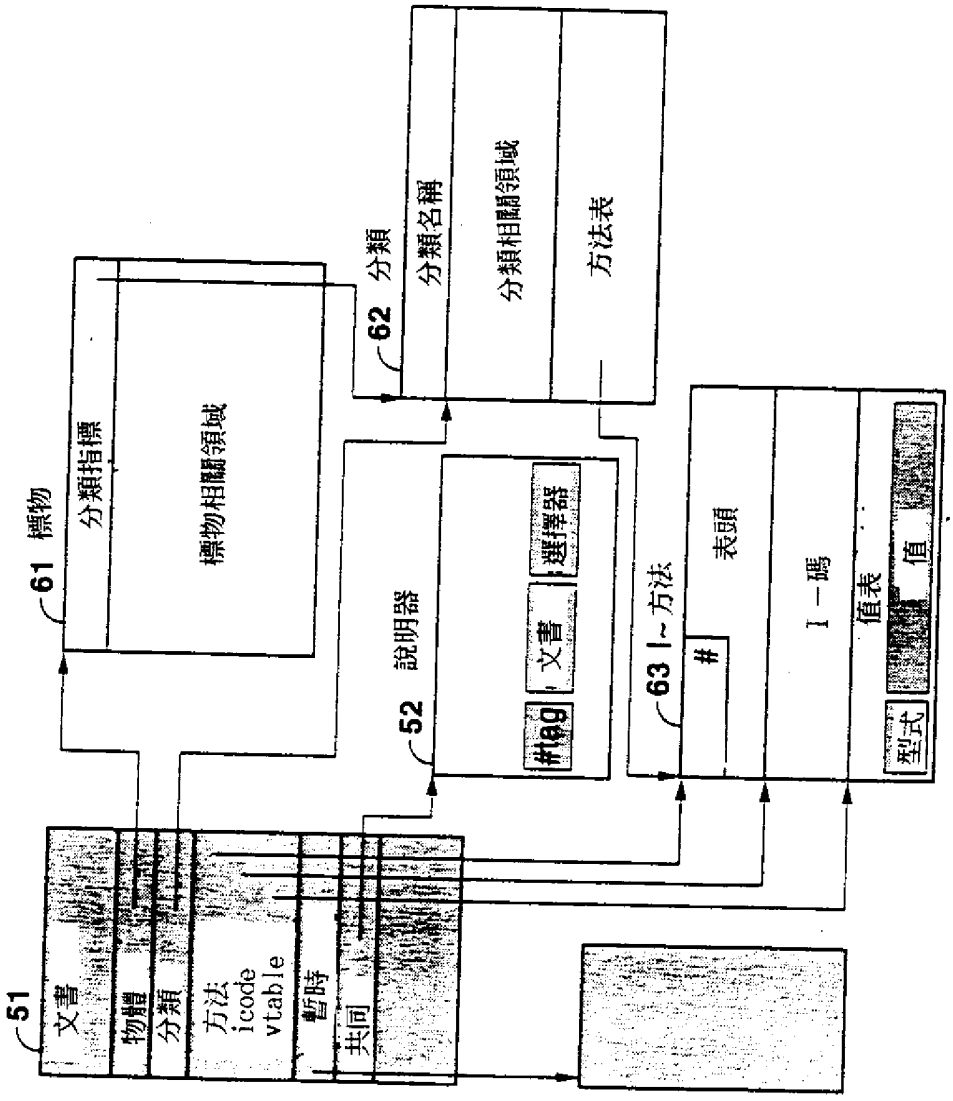
文書及說明器的邏輯結構

第15圖



MVM的整個結構

第16圖



文書及周圍資料結構

第17圖

型式	說明
T_ INVALID	無效項目
T_ PRIMITIVE	該值域表示原始標物，其與MVM上的標物相關
T_ POINTTER	該值域包含一至一標物的指標

變動表示之型式

第 18 圖

分類	P_ CLASS	P_BODY	說明
整數	P_ INTEGER	瞬時	整數
浮點	P_ FLOAT	瞬時	IEEE 754 32位元浮點數
多浮點	P_ DFLOAT	瞬時	IEEE 754 64位元浮點數
轉換器	P_ BOOLEAN	瞬時	真或假
串列	P_ STRING	對大量資料存取	單碼文字序列
陳列	P_ ARRAY	對大量資料存取	同型數據序列
文書	P_ CONTEXT	CName	標動執行表示
跳躍	P_ JUMP	I-碼索引	至重執行之目標位址
指標	P_ POINTER	對記憶體存取	位址
指定	P_ ASSIGN	n/a	變數項之複製
郵差	P_ MAILER	n/a	方法更新
未定義	P_ UNDEF	n/a	未定義之原始標物

原始標物

第19圖

19-1

```
struct Message {
  long length ;
  any body [ ];
};
exception outrange {long position} ;
exception overflow {} ;
exception undef {int primitive; int selector} ;
exception zerodiv {} ;

Interface Integer {
  Integer operator+(in Integer value)
    raise(overflow);
  Integer operator-(in Integer value)
    raise(overflow);
  Integer operator*(in Integer value);
  Integer operator/(in Integer value)
    raise(zerodiv);
  Integer remainder(in Integer value);
  Integer and(in Integer value);
  Integer or(in Integer value);
  Integer xor(in Integer value);
  Integer not(void);
  Integer operator=(in Float value);
  Integer operator=(in DoubleFloat value);
  Boolean opetator<(in Integer value);
  Boolean opetator<=(in Integer value);
  Boolean opetator==(in Integer value);
};
interface Float {
  Float operator+(in Float value)
    raise (overflow, underflow);
  Float operator-(in Float value)
    raise (overflow, underflow);
  Float operator*(in Float value)
    raise (overflow, underflow);
  Float operator/(in Float value)
    raise (overflow, underflow, zerodiv);
  Float operator=(in Integer value);
  Float operator=(in DoubleFloat value);
    raise (overflow);
  Boolean opetator<(in Float value);
  Boolean opetator<=(in Float value);
  Boolean opetator==(in Float value);
```

primitive object interface

第 20 圖

19-2

```
interface DoubleFloat{
        DoubleFloat operator+(in DoubleFloat value)
            raise(overflow, underflow);
        DoubleFloat operator-(in DoubleFloat value)
            raise(overflow, underflow);
        DoubleFloat operator*(in DoubleFloat value)
            raise(overflow, underflow);
        DoubleFloat operator/(in DoubleFloat value)
            raise(overflow, underflow, zerodiv);
        DoubleFloat operator=(in Float value)
            raise( );
        Boolean operator<(in DoubleFloat value)
            raise( );
        Boolean operator<=(in DoubleFloat value)
            raise( );
        Boolean operator==(in DoubleFloat value)
            raise( );
};
interface Boolean {
        Boolean not (void)
};
interface String {
        String operator+(in String string);
        String substring(in Integer position, in Integer length);
            raise(outrange);
        Integer length(void);
};
interface Array {
        void put (in Integer index, in any value);
            raise(outrange);
        any get (in Integer index)
            raise(outrange);
        Integer length (void);
};
interface Context {
  ...accessers...
};
interface Jump {
        void eval (void)
            raise(outrange);
        void evalT (Boolean cond)
            raise(outrange);
        void evalF (Boolean condition)
            raise(outrange);
};
```

primitive object interface (continued)

第21圖

```
19-3
interface Pointer {
        void put (in Address pos, in byte value)
              exception outrange;
        byte get (in Address pos)
              exception outrange;
        void put (in Address pos, in word value)
              exception outrange;
        word get (in Address pos)
              exception outrange;
        void put (in Address pos, in long value)
              exception outrange;
        long get (in Address pos)
              exception outrange;
        void put (in Address pos, in longlong value)
              exception outrange;
        longlong get (in Address pos)
              exception outrange;
};
interface Assing{
        void operator=(in int position1, in int position2);
interface Mailer{
        void call(in ID target, Integer select, in Message msg);
        void reply( );
};
interface undef
};
```

primitive object interface (continued)

第22圖

名稱	說明
OP_M	由第一變數所指定的操作。此操作由個人或原始標物所處理(操作域:操作,操作域數)
OP_R	當由OP-M回復請求操作時,當控制轉回標物 I-碼指令組

第23圖

```
interface MetaCore{
    mcError M (in long method, in Message msg);
    mcError R (in CName ctxt, in long method, in Message msg);
```

巨核介面