公告本

申請日期	86.10.02
案號	86114376
類別	G06F 12/00

中文說明書修正本(89年3月)
A4
C4

498203

89. 3. 13 （以上各欄由本局填註）

發明 新型 專利說明書

一、發明新型名稱	中文	供一整合發展環境之層級組織式描述其他資料之資料的儲存
	英文	HIERARCHICAL METADATA STORE FOR AN INTEGRATED DEVELOPMENT ENVIRONMENT
二、發明創作人	姓名	1.克里斯多夫 勞倫斯 伯瑞立 2.傑佛瑞 格蘭特 莊斯頓 3.維拉迪米爾 克立尼克 4.大衛 馬丁 勞隆 5.洛克 提 羅伊 6.德克 亞力山大 席爾曼 二世
	國籍	均加拿大
	住、居所	1.加拿大安大略省北約克市康可德區7號411棟 2.加拿大安大略省史卡伯羅市厚塞希克雷斯路34號 3.加拿大安大略省歐夏瓦市松林街567號 4.加拿大安大略省艾托比寇克市瑞剛路47號 5.加拿大安大略省東約克市公園景色大道4號1028棟 6.加拿大安大略省松希爾市金戴爾道56號
三、申請人	姓名（名稱）	美商萬國商業機器公司
	國籍	美國
	住、居所（事務所）	美國紐約州阿蒙市
	代表人姓名	費羅普

（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
加拿大	1997年3月27日	2,201,278	□有 □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

五、發明說明（1）

發明範疇

本發明係關於資料處理之領域，尤其係關於提供整合工具發展環境之領域。

發明背景

傳統工具發展的步驟是建立文字模式原始碼，然後在特定平台上編譯及執行。隨著跨平台工具功能及整合複雜度的增加，更多的人必須加入工具發展團隊。要產生功能完整的應用程式，通常只有使用特殊發展其它工具的應用程式發展工具。今日的發展工具包含了低階工具如文字編輯器、除錯器，和高階工具如介面建構器、資料存取建構器、分散化支援建構器、以及程式碼產生器等。這種環境下建構的應用程式可執行於多樣性的環境，不同的硬體平台，並提供許多分散式及資料存取機構。以現有技術發展這類應用程式必須使用多個不同廠商的工具，每一種解決一部份問題。

一整合發展環境(IDE)，使程式發展工具能分享各自提供的功能，避免不必要或不相容的介面操作係有需要的。目前的發展工具僅符合本要求之一小部份，他們通常僅針對一原始碼層次、單一目的應用程式，並使用一種簡單的檔案系統文件夾(目錄)模式來處理資訊。

當應用程式發展工具緊密整合時，使用交互對照表來分享程式的資訊。然而在一個較大的發展環境中，交互對照表可能會失去與原始碼的同步，而包含了不足的資訊。

況且任何不能緊密整合的工具(通常不同廠商的工具皆如

五、發明說明（2）

此)必須剖析其原始碼或表格，以使程式語言不是一種障礙。剖析之處理係剖析其差異定義，或剖析其內部表格定義。

發明總結

建構一供複雜應用程式使用之整合發展環境的主要問題之一為如何在不同的整合發展環境工具之間有意義地共享資訊。一個"資訊儲存"來達成共享係必須的。該儲存必須支援多工具／多使用者環境，並且提供能擷取語意資訊的彈性架構。

該儲存也必須定義一有關發展應用程式的全面性資訊結構，並能夠提供今日應用程式所需的自由度。這些自由度包含目標、語言、組成模型、分散等。該整合發展環境還必須定義一供所有工具分享應用程式語意的模型，而同時又允許個別工具以其私有資料擴充該共享資訊。該模型也必須"管理"程式規劃"元素"名稱，跨越不同自由度的範圍。

因此，本發明之目的在提供一支援複雜商業應用程式發展以組織描述其它資料之資料機制，該機制定義一處理應用程式發展描述其它資料之資料的階層化(layered)資料模型，並以單一描述其它資料之資料儲存說明不同應用程式自由度。

根據這些目的，本發明為物件導向環境中存取工具資料提供一描述其它資料之資料容器(container)，該容器為一層級組織式架構，組成包括含有子部份與一封裝

五、發明說明（3）

(encapsulated)行為之簡單構造種類、包含目的語言的性質之元件、以及允許分散化分割之組成部份。

本發明亦為物件導向程式規劃環境中供共同存取工具資料提供一描述其它資料之資料儲存，該儲存的組成包括定義工具資料中元素共同行為的單一基礎物件類別、以及繼承自單一基礎物件類別分離抽象物件類別層級組織以定義工具資料的名稱範圍與包容(containment)等部份。

圖示之簡述

圖1為本發明較佳實施例之描述其它資料之資料階層化各部份分類之示意圖；

圖2為本發明較佳實施例之模型階層式層級組織；及

圖3為本發明較佳實施例之整合發展環境元素基礎物件類別C++程式碼。

較佳實施例詳述

本較佳實施例實施為開放資料模型，提供由所有欲共同工作之發展工具或可存取及擴充之程式原始碼或共同中間模式。該資料模型與語言無關，提供交互系統原始碼共享。本模型可驅動程式碼產生器。

本較佳實施例之資料模型提供一支援工具整合架構之描述其它資料之資料儲存，該架構如我們的已申請之申請案"一種提供多工具共同物件關係與環境管理的物件導向機制"(IBM備忘錄第CA997-002號)。

為了處理不同的應用程式，從基本原始程式規劃到基於視覺架構應用程式的軟體元件(component)，本發明之資

五、發明說明（4）

料模型提供共享描述其它資料之資料儲存，其中發展資訊被階層化。每一階層定義該層次所需之共同行為。使用該模型之工具自提供之階層之行為開始，依各自的需求加以擴充。

該資料模型擷取發展應用程式所需之不同"元素"的資訊。資訊包括應用程式邏輯片段如文件夾、檔案及應用程式部份(part)。於最高層次，資料模型定義整合發展環境內需被辨認元素之共同行為，包括描述元素、公佈內容(若合)、及觸發其動作之能力。

與現有應用程式發展資訊大致相同，該模型將資訊儲存於檔案系統(原始檔及目錄／文件夾)，不過增加了一個鑰匙區域：

1.與文件夾包容(contain)不同之命名空間的辨認；及

2.調整應用程式"部份"觀念為不同部份階層。

本發明資料模型定義名稱範圍如一包容關係之"並行"集合，而不需發展者去辨認原始碼之計畫文件夾包容與應用程式部份名稱或強迫名稱關係與文件夾包容相同。這允許一包含C++物件類別A::B::C的部份被正確定義為一連續的巢狀名稱範圍，同時允許該部份可用於數個計畫文件夾(經由包容或連結)。

本發明資料模型支援來自部份之應用程式定義與組合(assembly)。本發明中，部份係指一全域應用程式之自我包含片段，不同於包含該部份原始碼之原始檔案。根據程式規劃環境，一部份可能參考數個檔案(如C++中的.hpp

五、發明說明(5)

和.cpp檔案)，但並非一檔案容器者。本發明資料模型中，部份為一應用程式定義(描述其它資料之資料)獨立可編輯片段，不同於任何可能與該相關或自該部份定義產生之原始檔案。如後面將詳述的，本發明資料模型中，所有的部份是有名稱且有名稱範圍的。一部份(根據實施)可能是其它部份的名稱範圍，比如在C++中的巢狀物件類別。

係如圖1所示，描述其它資料之資料儲存的階層。資料模型中部份1的語意被階層化，其分層(spectrum)從共同用途工具使用的部份到增加中之特殊工具使用的部份。

共同部份2位於分層中的最頂端，其係基礎部而相應目標語言中一結構類型之語意。共同部份包括子部份、封裝行為如方法(method)與繼承。通常依物件導向語言如C++或Java™的類型物件定義或某些結構形式如COBOL而定。

退化的共同部份的情況是本資料模型所提供的各種類型。這些類型為特殊化之共同部份，擷取或封裝以下之概念：基本類型(如整數、字元)屬於原始類型4；記錄屬於結構3；覆疊記憶體屬於集合7；有名稱的基數屬於列舉6；別名部份屬於定義(類型定義)5。

元件部份8是額外提供元件模型的基本名稱部份。若不考慮其最終原始語言實施，元件部份定義其外在行為如一集合，如一組"屬性"(即所謂"性質")、部份產生之事件及可由外觸發之動作(action)。資料模型擷取該資訊作為成分部份，並產生適當的目的元件模型(如Java Beans，IBM® VBE模型等)。在整合發展環境中，這表示高階"建構器"工

五、發明說明(6)

具所使用之支援程式。在圖2中，成分部份被稱為"消耗性建構器"部份。

分割容器9是一種元件部份，提供其部份之內在行為至一允許該部份為分散而重分割之層次。經由該資料模型，該等部份提供其聚合之部份間之互動及該等部份間之互動，該資訊層次被用來擷取應用程式分散式結構，而不考慮目標的分散機制。

根據特定之需求，一工具在資料模型中選擇適當層次部份資源以儲存其工具部份描述其它資料之資料。共享行為繼承自資料模型，而工具依需求自行擴充。通常這些擴充部份是私有的(opaque)性質。例如，該結構允許一資料存取部份，該部份為一包含存取依資料庫資訊之元件部份，供使用者介面(UI)建構器產生使用者介面部份用。該使用者介面建構器為一複合部份，包含資料存取部份和使用者介面控制。該部份可繼續被分割成一使用者部份(使用者介面與伺服器呼叫)、伺服器部份(伺服器邏輯)以及資料庫存取。這些皆藉由編輯資料模型資訊完成。同樣的資料模型資訊可用於產生一含有對伺服器之RMI呼叫的Java bean和對資料庫作JDBC存取，或一能對含有一區域資料庫之伺服器作DCE存取之C++使用者。

本較佳實施例以物件導向程式規劃層級組織之實施，該層級組織藉繼承而促進部份之分享。圖2說明本較佳實施例資料模型階層為每一層提供一物件類別定義的方法，並說明在該層級組織中層次的描述其它資料之資料類別如何為

五、發明說明(7)

工具使用。

最抽象的基本物件類別(整合發展環境元素20)供整合發展環境樹狀結構圖(tree-view)之實施。該圖使用完全虛擬的方法，開始對其元素或節點展開行動並企圖對層級組織作結構上改變，不考慮其內容細節，而顯示模型的內容。運作(特別是結構的改變)的成功全賴各個樹元素來源所提供的語意。

樹的節點之共同行為被定義為該樹狀圖中每一元素將包括名稱、圖案、元素的擁有或連結、以及元素是否可進一步擴充的指示。

擴張樹狀圖中分枝的基礎係列出任何一整合發展環境元素內容的能力。因此，整合發展環境元素基本物件類型20提供一顯露內容之共同行為。本較佳實施例中其描繪整合發展環境元素基本物件類型之C++標頭檔範例如圖3所示。每一個衍生的整合發展環境元素定義其內含結構。不像本模型中其它關係，"包含"關係必須提供以其名稱直接存取該元素以及游標式存取的能力，因為樹狀圖期符樹中節點可能被以拖曳放置或以重序動作之選擇的方式重新排序。元素的相關次序由模型提供。一個例子是視覺建構器工具列，其中每一個項目包含一可為建構器使用之部份。常用的項目應該被放在頂端，以減少捲動。

該模型亦於該抽象層次提供一組方法(method)，允許改變該樹之結構而不必考慮元素的細節。運作藉由衍生的實施完成。以下是兩個構成容器關係的機制：

五、發明說明（8）

1.擁有關係表示對主部份中子元素的聚合。當擁有之元素被父元素放棄時，擁有的元素將被解構；及

2.參照關係表示對樹中其它擁有或聚合之子元素的軟性連結或影子連結。

一元素上之移除動作將沿著擁有關係而非參照關係傳遞。擁有的元素將被解構，連結的元素將被移除連結。注意本資料模型並不強迫單一父連結規則，雖然大部份擁有關係皆屬此類。在擁有關係中，當父元素被解構或當父元素放棄子元素時，子元素亦將被解構。

由於衍生實施可能不允許改變樹節點之內容，必須提供能檢驗運作是否合法的方法。即使合法，也可能會有其它原因致使運作失敗。是否接受一內容動作全視內容之包容者(也就是父物件而非子物件)來決定。

本較佳實施例中，整合發展環境元素基本物件類別20提供額外的共同行為，包括：

1.動作(action)支援：本模型提供對模型物件的觸發編輯動作之機制。所謂"編輯"意指任何工具獨特的動作。每一個工具都被期望能在其具體物件類型起源中提供一編輯方法之實施。

2.程式碼產生支援：模型描述其它資料之資料的每一元素皆能導致原始碼輸出之產生。方法generate()的預設內容僅包含return的動作。如果要產生輸出，必須重寫新的generate()方法。generate()為一前後文相關方法。

3."失效"部份支援：一整合發展環境元素可被工具標示

五、發明說明(9)

為"失效"，表示該元素或其依存關係已改變，其原始碼應該被重新產生。本模型提供管理該支援的方法如圖3所示。

4.關聯檔案運作(operation)：本模型將包含與關聯檔案視為相異。一包含檔案為一結構上相關於其包含者之整合發展環境元素。一例子為包含於計畫群組文件夾中之原始碼檔案。包含檔案以本節稍前所述內容與結構運作處理。關聯檔案為一定義整合發展環境元素關聯性之參照檔案。關聯檔案的例子是使用部份所需之標頭檔，或從一群組或部份產生之原始碼。通常關聯其不包含於與該元素相關的元素中。

下一個抽象階層是包容範圍22和名稱範圍24基本物件類別。如前所述，包容與部份名稱是平行，因而是獨立的關係。

於包容範圍端，基本物件類別22表示一通用群組構造。在預設的形式中，允許能被包容或連結群組、部份、檔案之任意集合，同時其內容亦可經由樹狀圖或工具動作加以"編輯"。這是整合發展環境中的共同"文件夾"40結構。

基本名稱範圍物件類別24定義支援本發明較佳實施例模型之巢狀名稱範圍。它原來的目的是界定名稱部份的範圍。名稱範圍文件夾26與有名稱的物件類別被用來建構反應該部份定義範圍的層級組織。該階層與包容之層級組織平行，允許與部份包容無關之名稱範圍。

名稱範圍文件夾物件類別26與包含範圍物件類別22除皆定義名稱範圍外，其餘方面都相同。名稱範圍文件夾26允

五、發明說明（10）

許除了名稱群組及其起源物件及起源物件之部份外的名稱範圍。這是整合發展環境的共同名稱基本結構。

有範圍的名稱係藉由使用群組與部份結構而組合成的巢狀集合範圍來建構。結構的唯一起點(模型，model)是全域名稱範圍。被建構的群組和部份來源物件對允許建構之項目指名為其父名稱範圍。如果沒有指明名稱範圍，這些物件將被定義為全域性的範圍，其建立在名稱範圍層級組織中模型下。

一般而言，樹節點的子節點視為區域性名稱範圍。在預設的情況下，子元素必須擁有唯一的名稱。由於部份的名稱是全面唯一的(根據名稱範圍機制)，預部之模型之實施，允許建立一與任何非部份結構無關之部份，而有相同的名稱，卻不允許部份被儲存在一個元素容器內若該容器已有一相同名稱之子元素。

再回到名稱範圍層級組織。有名稱的部份物件類別28代表一個共同化部份結構44。在預設狀況中，其內容僅能藉由明確的工具動作編輯。此部份物件類別並非文件夾，雖然其可能被定義為子部份的聚合。

有名稱的部份物件類型模型以結構性類型為基礎，並允許資料模型擷取有關模型定義的屬性性質、繼承、資料成員、子部份聚合、可外部觸發行為或方法等資訊。所有的部份能表現如其他部份的名稱範圍。

如前所述，原始部份30為基本有名稱部份28的一個退化例子。原始部份30為資料模型所提供的不同原始類別如整

五、發明說明（11）

數、字元、浮點數、布林值等。

消耗性建構器部份物件類別32實施元件部份之語意。這些建構器部份可為視覺組成編輯器(圖形組成器)所消耗。其定義包括屬性、事件與動作。於產生時期，消耗性建構器部份要點對應到適合的目標成分模型46。

組成性建構器部份物件類別34是一定義可分割組成部分語意之分割物件類別，而非可分散部份。如果建構器工具被允許利用代理程式存取遠端部份之具體物件(instance)，該部份即為可分散的。於資料模型中這稱為部份性質。如果分散支援可有效地跨越數個部份重整部份之內容，該部份即為可分割的。衍生自本物件類別之具體部份對應至組成部份48的且係由分散支援成為可分割的。

一部份之子部份間的區域聯合在部份定義之內被擷取。他們代表子部份間的交互作用。本模型提供一些當子部份由一部份重新展開至另一部份時，處理部份重新分割之方法。如果重新展開成功，一部份間之分散聯合將被建立。這即是不同部份間子部份的分散部份之跨部份之交互作用。

一最終基本物件類別為檔案參考物件類別36，為一通用性的檔案參考結構。該物件類別為一指向檔案項目50之符號指標。

習於此藝之人士對本發明所述之實施例所作之更改皆包括於本申請案範圍內。

A5
B5

四、中文發明摘要（發明之名稱：供一整合發展環境之層級組織式描述其他資料之資料的儲存）

本發明提供一種整合發展環境之描述其它資料之資料的儲存。該描述其它資料之資料儲存被階層化以定義不同型式應用程式發展工具的共同行為層次。最常用之工具存取描述其它資料之資料的簡單架構層次；較特殊之工具存取包含目的語言性質之元件；更特殊之工具存取可建構分散式應用程式的可組合之可分割部分之描述其它資料之資料。

英文發明摘要（發明之名稱：HIERARCHICAL METADATA STORE FOR AN INTEGRATED DEVELOPMENT ENVIRONMENT）

A metadata repository for use in an integrated development environment is provided. The metadata repository is layered to define levels of common behaviour useful to different types of application development tools. The most general use tools have access to metadata at the level of simple constructed types; more specialised tools have access to components that contain properties of a target language; highly specialised tools have access to composed partitionable part metadata that can be used for constructing distributed applications.

A8
B8
C8
D8

六、申請專利範圍

本發明實施例申請專利範圍包含：

1. 一種電腦系統用以在物件導向程式規劃環境中用以儲存並取回在一供共同存取工具資料之容器中的描述其它資料之資料，具有一層級組織式架構，包含：

用以儲存電腦可讀取資料及電腦程式碼之裝置；

一該電腦程式碼之第一程式碼區段，包含多數個簡單構造種類其包容子部份及封裝行為；

一該電腦程式碼之第二程式碼區段，包含多數個元件包容目的語言之性質；

一該電腦程式碼的第三程式碼區段，包含多數個組成部份其允許分散化分割；以及

用以處理該儲存於儲存裝置中之第一、第二及第三程式碼區段之裝置，其致能多數個電腦實行設計工具以分享一設計之代表，實行及一應用程式之分散，其中該資料在儲存裝置中組織用以由設計工具直接操作。

2. 一種電腦系統用以在物件導向程式規劃環境中用以儲存並取回供共同存取工具資料的描述其它資料之資料，包含：

用以儲存電腦可讀取資料及電腦程式碼之裝置；

一該電腦程式碼之第一程式碼區段，包含一單一基礎物件類別以定義工具資料中元素共同行為；

至少一該電腦程式碼之其他程式碼區段，每一該至少其它程式碼區段包含一分離抽象物件類別層級組織，繼承自該單一基礎物件類別，以定義工具資料的名稱範圍

六、申請專利範圍

與包容；以及

用以處理該第一程式碼區段及該至少一其它程式碼區段之裝置，該程式碼區段儲存於該儲存裝置中，致能多數個電腦實行工具以存取由該程式碼區段所定義之共同工具資料。

3. 如申請專利範圍第2項之電腦系統，該定義名域之抽象物件類別層級組織包含：

一代表共同化部份結構之部份物件類別，該共同化結構適合包含子部份及封裝行為之。

4. 如申請專利範圍第2、3項之電腦系統，該定義名域之抽象物件類別層級組織包含：

一適合實施目的語言之性質的元件類別。

5. 如申請專利範圍第2、3項之電腦系統，該定義名域之抽象物件類別層級組織包含：

一適合包含部份分割語意定義之分割物件類別。

6. 如申請專利範圍第2項之電腦系統，該定義名域、劃分工具資料成為層級組織式架構之抽象物件類別階層包含：

包含子部份與封裝行為之簡單結構類別；

包含目的語言的性質之元件；及

允許分散式分割之組成部份。

7. 一種電腦實行方法用以在物件導向程式規劃環境中儲存供共同存取工具資料之描述其它資料的資料，該方法包含以下步驟：

識別一單一基礎物件類別，定義工具資料中元素共同

六、申請專利範圍

行為；

識別至少一分離抽象物件類別層級組織，繼承自該單一基礎物件類別，以定義工具資料的名稱範圍與內容；以及

代表於物件導向電腦程式碼中該識別基礎物件類別及該至少一物件類別層級組織，致能多數個電腦實行工具以存取如在該電腦程式碼中所代表之共同工具資料。

8. 如申請專利範圍第7項所述之電腦實行方法，其中該抽象物件類別層級組織以定義名稱範圍包含：

一部份物件類別代表共同化部份結構適合包含子部份及封裝行為。

9. 如申請專利範圍第7或8項所述之電腦實行方法，其中該抽象物件類別層級組織以定義名稱範圍包含：

一元件類別適合實行目的語言之性質。

10. 如申請專利範圍第7或8項所述之電腦實行方法，其中該抽象物件類別層級組織以定義名稱範圍包含：

一分割物件類別適合包含部份分割語意定義。

11. 如申請專利範圍第7項所述之電腦實行方法，其中該抽象物件類別層級組織以定義名稱範圍劃分工具資料成為階層架構，包含：

簡單結構類別，其包含子部份與封裝行為；

元件，其包含目的語言的性質；以及

組成部份，允許分散式分割。

12. 一種電腦可讀取媒體包含電腦程式碼用以在物件導向程

六、申請專利範圍

式規劃環境中儲存供共同存取工具資料的描述其它資料之資料，致能多數個電腦實行工具以存取如在該電腦程式碼中所代表之共同工具資料，該程式碼實行下列步驟：

識別一單一基礎物件類別，定義工具資料中元素共同行為；以及

識別至少一分離抽象物件類別層級組織，繼承自該單一基礎物件類別，以定義工具資料的名稱範圍與包容。

13. 一種電腦程式產品包含指令裝置用以實行如申請專利範圍第7至11項中任一項之方法。

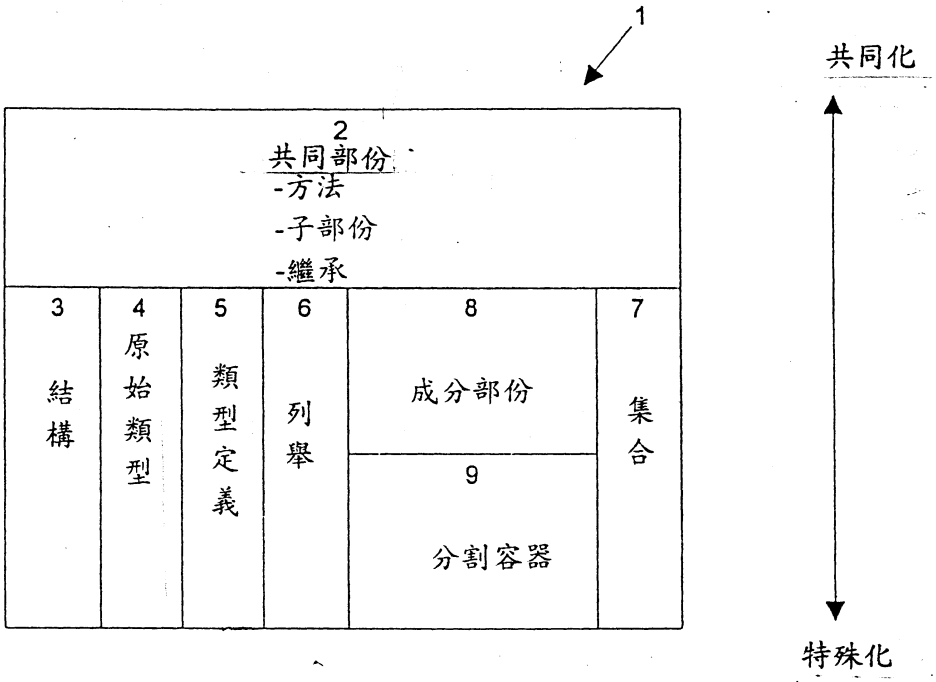

圖 1

整合發展環境模型階層

整合發展環境工具用法

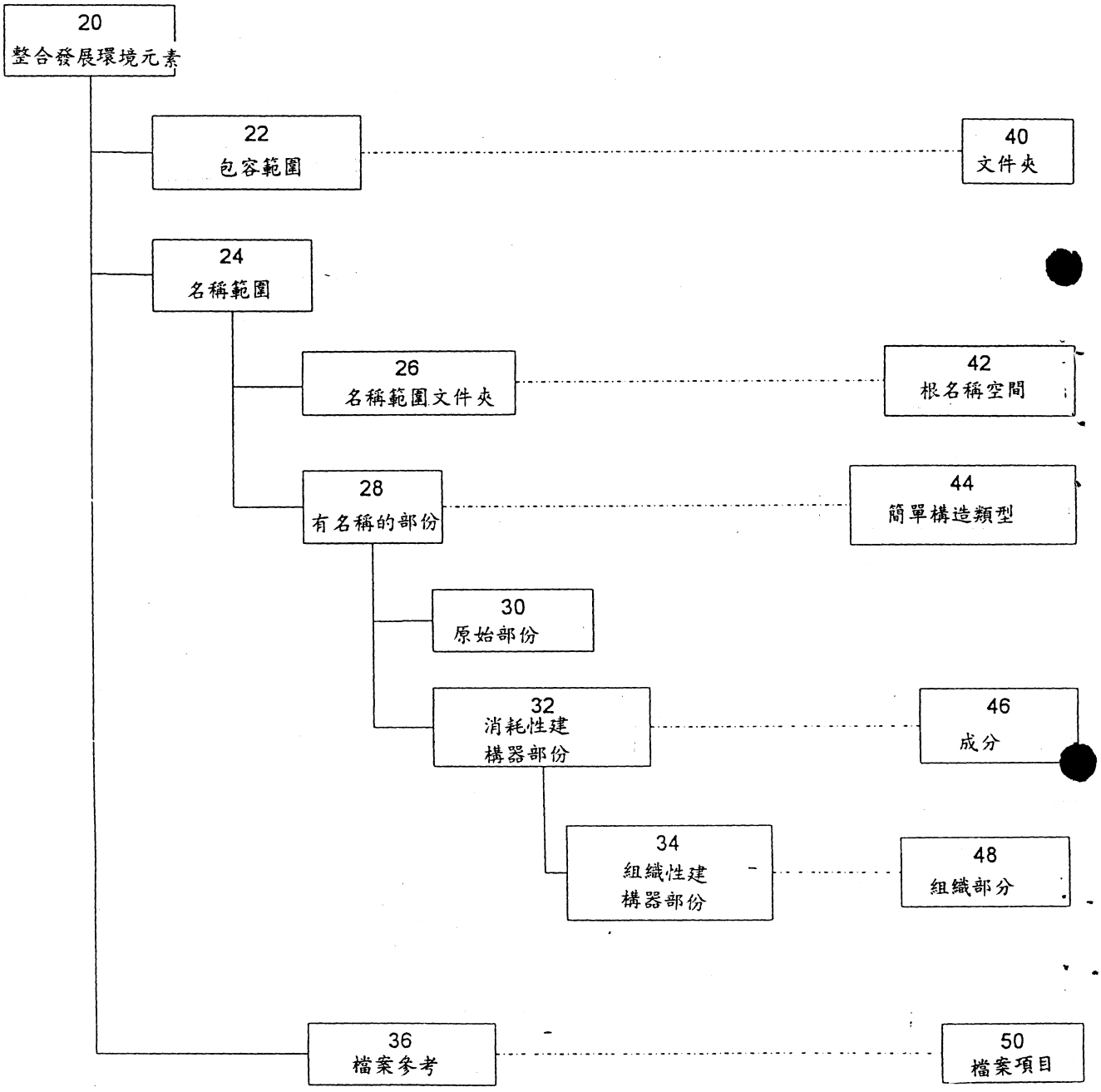

圖 2

```
class DMIDEElement : public DMBase
{

public:

  // Methods to query/manipulate content.

  class ContentCursor : public DMKeyedOrderedCursor
  {
  public:
    virtual ~ContentCursor () {}
    virtual Boolean isLinked () const = 0;
    virtual DMIDEElement const& element () const = 0;
  };
  DMIDEElement::ContentCursor* newCursor ();
  Boolean hasContent () const;
  unsigned int numberOfElements () const;
  DMIDEElement& elementAt ( const DMIDEElement::ContentCursor& curs ) const;
  DMIDEElement& elementWithKey ( const IString& key ) const;
  DMIDEElement* findElementWithKey ( const IString& key ) const;

  // Methods to access/change icon or bitmap.

  DMResource getIcon () const;
  void setIcon ( DMResource const& resource );
  DMResource getBitmap () const;
  void setBitmap ( DMResource const& resource );

  // Methods to allow tools to determine if/how content
  // is manipulated.

  virtual Boolean canOwn ( DMIDEElement& elem ) const = 0;
  virtual Boolean canLink ( DMIDEElement& elem ) const = 0;
  virtual Boolean canRemove ( DMIDEElement& elem ) const;
  virtual Boolean canMove ( DMIDEElement& elem ) const;


  // Methods to manipulate content

  virtual void move ( DMIDEElement& fromParent,
                      DMIDEElement& toParent,
                      Boolean own,
                      Boolean autoLocking = true );
  void ownAsFirst ( DMIDEElement& elem );
  void ownAsNext ( DMIDEElement& elem, DMIDEElement::ContentCursor& curs );
  void linkAsFirst ( DMIDEElement& elem );
  void linkAsNext ( DMIDEElement& elem, DMIDEElement::ContentCursor& curs );
  Boolean remove ( DMIDEElement& elem );
  void remove ( DMIDEElement::ContentCursor& curs );
  Boolean remove ( const IString& key );

  // Methods to query or manipulate associated files.

  Boolean hasFiles () const;
  void associateFile ( DMFile& file );
  DMFile& fileAt ( const ICursor& curs ) const;
  ICursor* createFileCursor () const;

  // Methods to invoke, enable, disable associated
  // toolactions.

  void disableToolAction ( DMToolAction& action ) const;
  void enableToolAction ( DMToolAction& action ) const;
  long perform ( DMToolAction& action, DMSearchCritter const& context );
  virtual Boolean canEdit () const;
  virtual void edit( DMSearchCritter const& context );

  // Method to indicate if tree element is to be shown in the tree

  virtual Boolean isVisible () const;

  // Method to allow and invoke code generation

  virtual void generate( DMSearchCritter const& context );
  virtual Boolean canGenerate () const;

  // Methods to query, manipulate the dirtiness state of the
  // tree element object

  void setDirty ( Boolean dirty = true );
  void resetDirty () { setDirty(false); }
  Boolean isDirty () const;

};
```

圖 3