發明專利說明書 200421121

（本說明書格式、順序及粗體字，請勿任意更動，※記號部分請勿填寫）

※ 申請案號： 92128010

※ 申請日期： 92.10.1 ※IPC 分類： G06F 17/22

壹、發明名稱：（中文/英文）

供高效網目確認之XML綱目項之標註自動機編碼

ANNOTATED AUTOMATION ENCODING OF XML SCHEMA FOR HIGH PERFORMANCE SCHEMA VALIDATION

貳、申 請 人：（共 1 人）

姓名或名稱：（中文/英文）

美商萬國商業機器公司

INTERNATIONAL BUSINESS MACHINES CORPORATION

代表人：（中文/英文）

傑拉德　羅森賽

GERALD ROSENTHAL

住居所或營業所地址：（中文/英文）

美國紐約州阿蒙市新果園路

NEW ORCHARD ROAD, ARMONK, NEW YORK 10504, U.S.A

國　籍：（中文/英文）

美國　U.S.A.

參、發明人：(共 4 人)

姓　名：(中文/英文)

1. 傅友青
   YOU-CHIN FUH
2. 王寧
   NING WANG
3. 王雲
   YUN WANG
4. 章谷根
   GUOGEN ZHANG

住居所地址：(中文/英文)

1. 美國加州聖約瑟市安妮皇后大道1140號
   1140 QUEEN ANNE DRIVE, SAN JOSE, CALIFORNIA 95129, U.S.A
2. 美國加州艾維市維拉諾區5623號
   5623 VERANO PLACE, IRVINE, CALIFORNIA 92612, U.S.A
3. 美國加州莎拉托葛市穆琳路20758號
   20758 MAUREEN WAY, SARATOGA, CALIFORNIA 95070, U.S.A
4. 美國加州聖約瑟市卡摩大道5614號4室
   5614 CALMOR AVE #4, SAN JOSE, CALIFORNIA 95123, U.S.A

國　籍：(中文/英文)

1. 3. 均美國　U.S.A.
2. 4. 均中國　REOPLE'S REPUBLIC OF CHINA

肆、聲明事項：

☐ 本案係符合專利法第二十條第一項☐第一款但書或☐第二款但書規定之期間，其日期為： 年 月 日。

☑ 本案申請前已向下列國家（地區）申請專利：

1. 美國；2002年10月15日；60/418,673
2. 美國；2003年04月16日；10/418,658
3.
4.
5.

☑ 主張國際優先權(專利法第二十四條)：

【格式請依：受理國家（地區）；申請日；申請案號數 順序註記】

1. 美國；2002年10月15日；60/418,673
2. 美國；2003年04月16日；10/418,658
3.
4.
5.

☐ 主張國內優先權(專利法第二十五條之一)：

【格式請依：申請日；申請案號數 順序註記】

1.
2.

☐ 主張專利法第二十六條微生物：

☐ 國內微生物 【格式請依：寄存機構；日期；號碼 順序註記】

☐ 國外微生物 【格式請依：寄存國名；機構；日期；號碼 順序註記】

☐ 熟習該項技術者易於獲得，不須寄存。

玖、發明說明：

【發明所屬之技術領域】

本申請案主張35 USC 119(e)下的臨時專利申請案第60/418,673號(申請日期2002年10月15日)之權利。

本發明係關於可擴充標示語言(XML)綱目，更特定言之，係關於依據一XML綱目確認XML文件。

【先前技術】

可擴充標示語言(XML)綱目為技術中所熟知。其允許使用者定義結構複雜的XML資料目標，以便於將XML格式廣泛用於資料儲存與資料庫中的處理中。提供各XML文件時，必須根據其XML綱目定義對其進行確認，以確保其用於資料儲存與處理之前，與定義相符。

傳統上，主要採用兩種XML綱目確認方法。第一種為通用綱目確認剖析器。在該方法中，剖析器接收XML綱目定義與XML文件作為輸入，將該XML文件剖析為一樹格式，將該XML綱目定義剖析為一綱目樹格式，然後根據XML綱目樹，遍曆XML文件樹進行核對。對於許多不同的XML綱目，可使用相同的通用綱目確認剖析器。儘管該方法因其可對許多不同的XML綱目進行確認而較為靈活，但該方法通常效能較差。

第二種方法係根據一特定的XML綱目定義，產生一XML綱目確認剖析器代碼。將該XML綱目確認剖析器代碼編譯為可執行代碼，用以根據該特定的XML綱目定義，確認一XML文件。儘管該方法比第一種方法速度快，但是，當存

在大量的XML綱目定義時，就要有大量的確認剖析器，造成空間負擔。而且，由於各XML綱目確認剖析器只能根據一特定的XML綱目進行確認，因而也不夠靈活。

因此，就需要一種改良的XML綱目確認方法與系統。改良的方法與系統應提供高效，但不會引起空間負擔增加。本發明可滿足此種需要。

【發明內容】

本發明提供一種用於可擴充標示語言(Extensible Markup Language；XML)綱目確認的方法與系統，其包括：將一XML文件載入一運行時間確認引擎中，其中該運行時間確認引擎包括一XML綱目確認剖析器；將一XML綱目定義的標註自動機編碼(annotated automaton encoding；AAE)載入該XML綱目確認剖析器；及由該XML綱目確認剖析器利用該標註自動機編碼，根據該XML綱目定義，確認該XML文件。每一XML綱目定義係一次性編譯成該AAE格式，而不是每次確認一XML文件時編譯，從而明顯節省了時間。該運行時間確認引擎的代碼係固定的，不會隨該XML綱目定義而變化，而是對於每一XML綱目定義不同，從而使空間負擔最小。該確認程序具有靈活性，而且不影響效能。

【實施方式】

本發明提供一種改良的XML綱目確認方法與系統。以下展示的說明可使熟悉技術人士利用與使用本發明，且其係在專利申請案及其需求的背景下提供。對於熟悉技術人士

而言，顯然可對較佳具體實施例進行各種修改，且本文的基本原理可應用於其他具體實施例。因此，本發明無意限制於本文所示的該等具體實施例，而是符合與本文說明的該等原理及特徵相一致的最廣泛範疇。

簡介

依據本發明的改良方法與系統，將XML綱目定義編譯為一標註自動機編碼(AAE)格式，儲存在碟片或資料庫中。各XML綱目定義係一次性編譯為AAE格式。確認一XML文件時，將該XML文件及AAE格式的適當XML綱目定義載入一運行時間確認引擎中。運行時間確認引擎包括一基本XML剖析器及一運行時間綱目確認剖析器。運行時間確認引擎的代碼係固定的，不會隨XML綱目定義而變化。當運行時間綱目確認剖析器根據AAE格式的XML綱目定義，對一XML文件執行一高階確認時，該基本XML剖析器執行一低階確認。運行時間確認引擎的輸出為確認通過或確認失敗。

為了更明確地說明本發明的特徵，請結合下面的內容參考圖1至17。

圖1說明依據本發明，一XML綱目確認系統的一項較佳具體實施例。該系統包括兩組邏輯，一XML綱目編譯102與一運行時間確認引擎104。

該XML綱目編譯102將XML綱目定義106取作輸入，並提供標註自動機編碼(AAE)格式108的各XML綱目作為輸出。AAE格式108的XML綱目定義儲存在一碟片或一資料庫

110中，便於稍後取回。AAE格式包括一剖析表格式，其通常可從一自左至右向前查看(Look Ahead Left to Right；LALR(1))剖析器產生器中獲得，但還帶有附加標註。標註係元素節點的屬性與元素節點的資料類型約束，及為一特定的XML綱目定義所產生的XML綱目樹的屬性。下面參考圖2至9進一步說明XML綱目編譯102與AAE格式。

運行時間確認引擎104將一XML文件112及其對應的AAE格式的XML綱目定義108取作輸入，並提供一確認通過或確認失敗作為輸出。以下將參考圖10至14進一步說明運行時間確認引擎104。

XML綱目編譯

圖2詳細說明依據本發明的XML綱目編譯102。XML綱目編譯102包括一XML綱目編譯器前端202與一XML綱目編譯器後端204。圖3為一流程圖，說明依據本發明XML綱目編譯過程的一項較佳具體實施例。

參考圖2與3，藉由步驟302，XML綱目編譯器前端202接收一XML綱目定義106作為輸入。然後，藉由步驟304，XML綱目編譯器前端202產生一元素結構階層，並以標註樹的形式表示。樹的節點係元素節點，邊緣係節點間的結構關係。標註係元素節點的屬性，及元素節點與屬性的資料類型約束。下面進一步說明該標註樹。

接著，藉由步驟306，XML綱目編譯器後端204對標註樹進行編碼，並產生一標註自動機編碼。然後，藉由步驟308，使標註自動機編碼序列化，並儲存在一碟片或一資料庫

中。因而，標註自動機編碼係AAE格式108的XML綱目定義。

圖4至9說明依據本發明，由XML綱目編譯102實施的方法之範例。圖4說明一範例XML綱目定義400。XML綱目定義400包括一預設的XML名稱空間402、一名稱空間宣告404及一註冊的目標名稱空間405。為安全起見，可註冊XML綱目，並在已註冊的XML綱目參考404所引用的替代位置進行存取。XML綱目定義400也包括用於文件目的的標註406。在本範例中，標註406係定義為包括一名稱為「文件」的元素408，其屬性410為「xml:lang='en'」。

XML綱目定義400進一步包括總體元素宣告，如*personnel*(全體人員)412與*notes*(備註)426。*personnel*總體元素412宣告為一複雜類型，且具有名稱為*employee*(員工)414的子元素序列。*employee*子元素414宣告為一複雜類型，且具有語義約束416。*employee*子元素414本身宣告為具有子元素418，各具有其自己的語法約束420。在本範例中，此等子元素係各為簡單類型(字串)之*lastname*(姓)、*firstname*(名)及*notes*(備註)。*employee*子元素414也宣告為具有若干屬性422，各定義為一簡單類型424。在本範例中，屬性422包括*serno*的類型*integer*(整數)、*userid*的類型*USERID_TYPE*及*department*(部門)的類型*string*(字串)。*integer*與*string*類型係預先定義。*USERID_TYPE*類型則不是。其定義係在428的XML綱目定義中提出。

藉由步驟302，XML綱目編譯器前端202將該XML綱目定

義400接收為輸入。然後，其為該XML綱目定義400產生一元素結構階層，並以標註樹500的形式來表示，如圖5所示。實線連接樹500中的元素節點。虛線將元素節點連接至屬性標註。

例如，實線將*personnel*節點502連接至*employee*節點506，又將*employee*節點506連接至*lastname*節點508、*firstname*節點510及*notes*節點512。樹500說明*notes*與*personnel*為總體元素，因而其處於節點的第一階層。*Employee*為*personnel*的子元素，因而其節點由一實線連接至*personnel*，處於第二階層。*Lastname*、*firstname*與*notes*為*employee*的子元素，因此，其節點508至512由實線連接至*employee*節點506，處於第三階層。最靠近*lastnam e*508、*firstname* 510及*notes* 512節點的虛線橢圓「字串」514至518係資料類型標註節點，表示其個自對應節點的資料類型。

樹500進一步說明*employee*節點506定義為具有屬性，表示為*serno*方塊520、*userid*方塊522及*department*方塊524。最靠近方塊的虛線橢圓*USERID*_TYPE 526、Integer 528與String 530為其對應類型的標註節點。

一旦為XML綱目定義400產生了標註樹500，藉由步驟306，XML綱目編譯器後端204即對標註樹500進行編碼，並產生標註自動機編碼。在對標註樹500進行編碼的過程中，XML綱目編譯器後端204首先以上下文無關文法(context-free grammar；CFG)，表示標註樹500(即實線所指定的樹結構)的元素約束，如圖6所示。然後，其藉由附加

一起始標籤符記作為各元素的前置，及附加一結束標籤符記作為各元素的後置，來擴充該CFG，如圖7所示。例如，對於*personnel*元素，附加一起始標籤符記*PS*與一結束標籤符記*PE*。對於*employee*元素，附加一起始標籤符記*ES*與一結束標籤符記*EE*。類似的，對元素*lastname*、*firstname*及*notes*附加起始與結束標籤符記。然後，使用一LALR(1)剖析器產生器演算法，產生一剖析表或狀態變遷表。再用另一演算法，構建由圖5所示的樹之標註組成的標註記錄。圖8中說明範例XML綱目定義400的最終狀態變遷圖的標註部分。藉由產生此類標註自動機編碼，對一XML資料目標的確認涉及檢查起始或結束標籤序列是否與圖7所示的文法相符。

參考圖8，範例XML綱目定義400包括一自動機編碼表801、一包括總體元素*notes* 802與*personnel* 803的總體元素映像。圖9說明範例XML綱目定義400的自動機編碼表801之狀態變遷圖。為簡明起見，圖9僅說明定義400的SHIFT邊緣，而省略了REDUCE邊緣。

再看圖8，總體元素映像包括*notes*元素名稱802與*personnel*元素名稱803。其依次又分別包括*notes* 802元素與*personnel* 803元素的標註記錄843與809之指標。*personnel*元素標註記錄809包括其元素內容及引數804(<*anyTypeID, null*>)的掃描器ID、其起始標籤符記805(*PS*)、其結束標籤符記806 (*PE*)、一屬性列表807及一候選子元素映像808。*personnel*元素標註記錄809的屬性列表

807為空。其候選子元素映像808包括其局部元素映像的指標。該局部元素映像810包括*employee*元素名稱，其又包括*employee*元素標註記錄816的指標。

*employee*元素標註記錄816包括其元素內容及引數811(<*anyTypeID, null*>)的掃描器ID、其起始標籤符記812 (*ES*)、其結束標籤符記813 (*EE*)、一屬性列表814及一候選子元素映像815。依據XML網目定義400，屬性列表814包括三個屬性：*serno*、*userid*與*department*。因而，屬性列表814包括：*serno*屬性名稱817與其值及引數818 (<*integerTypeID, null*>)的掃描器ID的指標、*userid*屬性名稱819與其值及引數820 (<*anySimpleTypeID ,[a-zA-Z][1][1-9a-zA-Z]*>)的掃描器ID的指標，以及*department*屬性名稱821與其值及引數822 (<*stringTypeID, null*>)的掃描器ID的指標。

*employee*元素標註記錄816的候選子元素映像815包括其局部元素映像的指標，該局部元素映像包括*employee*元素的子元素之標註記錄的指標。依據XML網目定義400，*employee*元素的子元素包括*lastname*、*firstname*及*notes*元素。每一個又分別包括*lastname*元素標註記錄829的指標、*firstname*元素標註記錄837的指標以及*notes*元素標註記錄843的指標。

*lastname*元素標註記錄829包括其內容及引數824 (<*stringTypeID,null*>)的掃描器ID、其起始標籤符記825 (*LS*)、其結束標籤符記826 (*LE*)、一屬性列表827及一候選子元素映像828。依據XML網目定義400，*lastname*元素既

無屬性也無子元素，因而，其屬性列表827及其候選子元素映像828為空。

*Firstname*元素標註記錄837包括其內容及引數832(<*stringTypeID, null*>)的掃描器ID、其起始標籤符記833(*FS*)、其結束標籤符記834(*FE*)、一屬性列表835及一候選子元素映像836。依據XML綱目定義400，*firstname*元素既無屬性也無子元素，因而，其屬性列表835及其候選子元素映像836為空。

*Notes*元素標註記錄843包括其內容及引數838(<*stringTypeID, null*>)的掃描器ID、其起始標籤符記839(*NS*)、其結束標籤符記840(*NE*)、一屬性列表841及一候選子元素映像842。依據XML綱目定義400，*notes*元素既無屬性也無子元素，因而，其屬性列表841及其候選子元素映像842為空。

標註自動機編碼800係用於依據XML綱目定義400來確認XML文件，下面將進一步說明。

運行時間確認引擎

圖10詳細說明依據本發明的運行時間確認引擎104。運行時間確認引擎104包括一XML掃描器集用場1002、一基本XML剖析器1004及一運行時間綱目確認剖析器1006。XML掃描器集用場1002包括一基本掃描器1017及其他用於特定簡單資料類型的掃描器1018至1020。基本掃描器1017能夠掃描所有的詞素符記，但其效能有限。而用於特定資料類型的掃描器1018至1020提供的效能較大。例如，

用於整數類型的掃描器1019與用於字串類型的掃描器1020可以為XML掃描器集用場1002中的一部分。在該項較佳具體實施例中，一AnySimpleType(任意簡單類型)掃描器1018用作任意簡單資料類型的基本掃描器。各掃描器1017至1020具有一唯一掃描器ID(0, 1, ... n, n+ l)。該等掃描器由基本XML剖析器1004呼叫，以使一輸入XML文件112具有符記。要呼叫哪個掃描器，由運行時間網目確認剖析器1006依據AAE格式108中的XML網目來決定。

基本XML剖析器1004接收來自掃描器的符記，並檢查基本的XML文法。若該符記為一元素符記，即一起始標籤名稱或一結束標籤名稱，或若該符記為一屬性符記，則將其作為一詞素(lexeme)轉送至運行時間網目確認剖析器1006。運行時間網目確認剖析器1006根據AAE格式中的XML網目定義來確認元素與屬性詞素。因而，基本XML剖析器1004執行了一低階確認，而運行時間網目確認剖析器1006依據CFG的網目結構，執行了一高階確認(見圖6)。

運行時間網目確認剖析器1006包括一XML網目載入模組1010，其從碟片或資料庫110中，載入適當的AAE格式108中的XML網目。運行時間網目確認剖析器1006也包括一XML網目確認模組1012，其執行實際的高階確認程序。該模組1012包括一元素確認模組1014，用於確認元素詞素，及一用於確認屬性詞素之屬性確認模組1016。

圖11至14為流程圖，說明依據本發明，由運行時間確認引擎執行的運行時間確認程序的一項較佳具體實施例。參

考圖11，首先，藉由步驟1102，XML綱目載入模組1010載入AAE格式108中的XML綱目，並將基本掃描器1017設定為現行掃描器。同樣，實體管理者1008將獲得由XML文件實例所引用的XML外部實體以進行確認。藉由步驟1104，載入XML文件112。現行掃描器使XML文件112具有符記。然後，藉由步驟1106，基本XML剖析器1004呼叫現行掃描器，以獲取一符記。藉由步驟1108，基本XML剖析器1004檢查是否成功返回了一符記。若符記掃描不成功，則藉由步驟1110，該確認以「無效」返回，且程序結束。若符記掃描成功，則藉由步驟1112，基本XML剖析器1004決定符記為一元素或屬性符記，然後藉由步驟1114，將該符記作為一詞素，輸入XML綱目確認模組1012中。詞素可為三種類型之一：一起始標籤名稱、一屬性名稱或一結束標籤名稱。藉由步驟1118、1120或1122，對各類詞素以不同的方式進行處理。若該符記不是一元素或屬性符記，則藉由步驟1124，決定該符記是否為檔案結束(end of file；EOF)符記，即XML文件112結束。若否定，則程序返回至步驟1106，對下一符記重複上述步驟。若肯定，則藉由步驟1125，決定基本XML剖析器1004與元素確認模組1014是否皆處於「接受」模式，即，是否已經完成了剖析與確認。若肯定，則對XML文件112的確認成功，藉由步驟1126，返回一「有效」。若否定，則確認失敗，並藉由步驟1127，返回一「無效」。

圖12為一流程圖，說明依據本發明，對一起始標籤名稱

的處理。首先，藉由步驟1202，基於先前的標註記錄及起始標籤名稱，XML網目確認模組1012找出現行的標註記錄。現行的標註記錄為映像項所指向的標註記錄，該映像項對應於先前的標註記錄的候選子元素映像中的現行標籤名稱。藉由步驟1206，將該現行標註記錄推至堆疊上。然後，藉由步驟1208，XML網目確認模組1012從現行的標註記錄中，獲取對應於起始標籤名稱的符記，並將該符記輸入元素確認模組1014中。元素確認模組1014依據AAE的剖析表部分(如801)，執行LR剖析。若藉由步驟1210，確認不成功，則確認失敗，並藉由步驟1212，返回一「無效」。若藉由步驟1210，確認成功，然後，若藉由步驟1214，現行標註記錄的屬性列表為空，且元素內容為簡單類型，則藉由步驟1216，將現行的掃描器ID設定為用於簡單類型的掃描器。隨後，以步驟1106繼續本程序，如圖11所示。

圖13為一流程圖，說明依據本發明，對一屬性名稱的處理。首先，XML網目確認模組1012獲取現行的標註記錄，並藉由步驟1301，將現行的標註記錄與屬性名稱傳送至屬性確認模組1016。藉由步驟1302，屬性確認模組1016搜尋現行標註記錄中的屬性列表，以獲取屬性名稱。若藉由步驟1304，在現行的標註記錄中未找到屬性名稱，則XML文件112的結構與XML網目定義108不相符，對XML文件112的確認失敗，並藉由步驟1310，返回一「無效」。若藉由步驟1304，找到屬性名稱，則藉由步驟1312，XML網目確認模組1012將現行的掃描器ID設定為該屬性值的簡單類

型。隨後，以步驟1124繼續本程序，如圖11所示。

圖14為一流程圖，說明依據本發明，對一結束標籤名稱的處理。首先，藉由步驟1402，將現行標註記錄從堆疊中移除。依據AAE格式，標註記錄的結束標籤係在起始標籤與任何屬性之後進行處理。由於現行的標註記錄係依據上述步驟1202至1204進行設定，故結束標籤名稱的現行標註記錄應為起始標籤名稱的標註記錄。由於結束標籤名稱為待處理的一標註記錄的最後符記，所以其標註記錄隨其處理過程從堆疊中移除。接著，藉由步驟1404，XML綱目確認模組1012從現行的標註記錄中，獲取結束標籤名稱的符記，並將其輸入元素確認模組1014中。若藉由步驟1406，確認不成功，則確認失敗，並藉由步驟1410，返回一「無效」。若藉由步驟1406，確認成功，則藉由步驟1408，XML綱目確認模組1012決定現行標註記錄的所有屬性是否均已確認，或屬性列表是否為空。若否定，則XML文件112的結構與XML綱目定義108不相符，確認失敗，並藉由步驟1410，返回一「無效」。若肯定，則程序返回至圖11中的步驟1106，對下一符記重複上述步驟。

在較佳具體實施例中，為了檢查唯一性約束430，如對於XML綱目定義400中的*employee*元素，可用一雜湊表或陣列來追蹤以前是否遇到一employee(員工)serno。

在較佳具體實施例中，為了檢查參考完整性，可在掃描XML文件112時，收集一定義列表。然後，可根據定義列表檢查參考。在參考出現在定義之前的情形中，必須要掃

描整個XML文件112，並且在檢查參考完整性之前記住參考。

範例確認

例如，假定XML文件112的XML綱目定義為範例XML綱目定義400(圖4)。圖16說明將確認的一範例XML文件1600。藉由步驟1102，XML綱目載入模組1012載入AAE格式中的XML綱目，並將基本掃描器1017設定為現行掃描器。同樣，藉由步驟1104，載入XML文件1600。藉由步驟1106，基本XML剖析器1004呼叫現行的掃描器，以獲取第一符記，其為*personnel*起始標籤名稱1602。藉由步驟1108，符記掃描成功，再藉由步驟1112，基本XML剖析器1004決定其為一元素符記，然後藉由步驟1114，將其作為一詞素輸入XML綱目確認模組1012中。之後，藉由步驟1116，XML綱目確認模組1012決定該詞素類型。由於該詞素為一起始標籤名稱，故執行處理步驟1118，如圖12所示。

Personnel起始標籤名稱

依據XML綱目定義400之圖800，*personnel*起始標籤名稱803先前的標註記錄指向*personnel*元素標註記錄809。參考圖12，因而，藉由步驟1202，*personnel*元素標註記錄809係現行的標註記錄，並藉由步驟1206，將其推至堆疊上。堆疊的內容變為圖15A中所示。然後，藉由步驟1208，XML綱目確認模組1012從*personnel*元素標註記錄809中，獲取*personnel*起始標籤符記*(PS)*805，並將該符記805輸入元素確認模組1014中。元素確認模組1014係一使用一剖析表進

行狀態變遷的剖析器，如圖9所示。藉由步驟1210，確認成功。藉由步驟1214，*personnel*元素標註記錄809的屬性列表807為空(零)，且其內容為複雜類型(anyTypeID)。因而，確認程序返回至圖11中的步驟1106，對下一符記重複上述步驟。

Employee起始標籤名稱

藉由步驟1106，基本XML剖析器1004呼叫現行的掃描器，以獲取下一符記：*employee*起始標籤名稱1604。藉由步驟1108，符記掃描成功，再藉由步驟1112，基本XML剖析器1004決定其為一元素符記，然後藉由步驟1114，將其作為一詞素輸入XML網目確認模組1012中。之後，藉由步驟1116，XML網目確認模組1012決定該詞素類型。由於該詞素為一起始標籤名稱，故執行處理步驟1118，如圖12所示。

參考圖12，personnel元素標註記錄809的候選子元素映像808，即先前的標註記錄，指向employee起始標籤名稱810，其藉由步驟1202，又指向employee元素標註記錄816，即現行的標註記錄，再藉由步驟1206，將其推至堆疊上。現在堆疊的內容變為圖15B中所示。藉由步驟1208，XML網目確認模組1012從employee元素標註記錄816中，獲取employee起始標籤符記(ES)812，並將該符記812輸入元素確認模組1014中。藉由步驟1210，確認成功。由於藉由步驟1214，employee元素標註記錄816的屬性列表814不為空，所以程序返回至圖11中的步驟1106，並對下一符記重複上

述步驟。

Serno屬性名稱

藉由步驟1106，基本XML剖析器呼叫現行掃描器，以獲取下一符記：屬性名稱*serno* 1606的符記。藉由步驟1108，符記掃描成功，再藉由步驟1112，基本XML剖析器1004決定其為一屬性符記，然後藉由步驟1114，將其作為一詞素輸入XML網目確認模組1012中。之後，藉由步驟1116，XML網目確認模組1012決定該詞素類型。由於該詞素為一屬性名稱，故執行處理步驟1120，如圖13所示。

參考圖13，藉由步驟1301，XML網目確認模組1012將現行的標註記錄與*serno*屬性名稱，傳送至屬性確認模組1016。現行的標註記錄為employee元素標註記錄816。藉由步驟1302，屬性確認模組1016搜尋屬性列表814，以獲取*serno*屬性名稱817。藉由步驟1304，在其中找到*serno*屬性名稱817。確認成功，再藉由步驟1312，XML網目確認模組1012將現行的掃描器ID設定為integerTypeID(整數類型ID)。然後，程序返回至圖11中的步驟1106，對下一符記重複上述步驟。IntegerType掃描器1019掃描整數「12345」，作為*serno*屬性名稱的屬性值。現行的掃描器重新設定為基本掃描器。

Userid屬性名稱

藉由步驟1106，基本XML剖析器呼叫現行的掃描器，以獲取下一符記：屬性名稱*userid* 1608的符記。藉由步驟1108，符記掃描成功，再藉由步驟1112，基本XML剖析器

1004決定其為一屬性符記，然後藉由步驟1114，將其作為一詞素輸入XML網目確認模組1012中。之後，藉由步驟1116，XML網目確認模組1012決定該詞素類型。由於該詞素為一屬性名稱，故執行處理步驟1120，如圖13所示。

參考圖13，藉由步驟1301，XML網目確認模組1012將現行的標註記錄與*userid*屬性名稱819，傳送至屬性確認模組106。現行的標註記錄為employee元素標註記錄816。藉由步驟1302，屬性確認模組1016搜尋屬性列表814，以獲取*userid*屬性名稱819。藉由步驟1304，在其中找到*userid*屬性名稱819。確認成功，再藉由步驟1312，XML網目確認模組1012將現行的掃描器ID設定為anySimpleTypeID(任何簡單類型ID)。然後，程序返回至圖11中的步驟1106，對下一符記重複上述步驟。AnySimpleType掃描器1018掃描「A1bc78D」，作為*userid*屬性名稱的屬性值。現行的掃描器重新設定為基本掃描器。

Department屬性名稱

藉由步驟1106，基本XML剖析器呼叫現行的掃描器，以獲取下一符記：屬性名稱*department* 1610的符記。藉由步驟1108，符記掃描成功，再藉由步驟1112，基本XML剖析器1004決定其為一屬性符記，然後藉由步驟1114，將其作為一詞素輸入XML網目確認模組1012中。之後，藉由步驟1116，XML網目確認模組1012決定該詞素類型。由於該詞素為一屬性名稱，故執行處理步驟1120，如圖13所示。

參考圖13，藉由步驟1301，XML網目確認模組1012將現

行的標註記錄與*department*屬性名稱821，傳送至屬性確認模組106。現行的標註記錄為employee元素標註記錄816。藉由步驟1302，屬性確認模組1016搜尋屬性列表814，以獲取*department*屬性名稱821。藉由步驟1304，在其中找到*department*屬性名稱821。藉由步驟1308，確認成功，再藉由步驟1312，XML網目確認模組1012將現行的掃描器ID設定為stringTypeID(字串類型ID)。然後，程序返回至圖11中的步驟1106，對下一符記重複上述步驟。StringType掃描器1020掃描「sales(銷售)」，作為*department*屬性名稱的屬性值。現行的掃描器重新設定為基本掃描器。

Lastname起始標籤名稱

藉由步驟1106，基本XML剖析器呼叫現行的掃描器，以獲取下一符記：*lastname*起始標籤名稱1612的符記。藉由步驟1108，符記掃描成功，再藉由步驟1112，基本XML剖析器1004決定其為一元素符記，然後藉由步驟1114，將其作為一詞素輸入XML網目確認模組1012中。之後，藉由步驟1116，XML網目確認模組1012決定該詞素類型。由於該詞素為一起始標籤名稱，故執行處理步驟1118，如圖12所示。

參考圖12，employee元素標註記錄816的候選子元素映像815，即先前的標註記錄，指向*lastname*起始標籤名稱823，其藉由步驟1202，又指向*lastname*元素標註記錄829，即現行的標註記錄，再藉由步驟1206，將其推至堆疊上。現在堆疊的內容變為圖15C中所示。藉由步驟1208，XML網目

確認模組1012從lastname元素標註記錄829中，獲取*lastname*起始標籤符記(*LS*)825，並將該符記825輸入元素確認模組1014中。藉由步驟1210，確認成功。由於藉由步驟1214，lastname元素標註記錄829的屬性列表814為空，且lastname元素的內容為簡單類型(stringTypeID)，故藉由步驟1216，將現行的掃描器ID改為stringTypeID。然後，程序返回至圖11中的步驟1106，對下一符記重複上述步驟。隨後，呼叫StringType掃描器1020，掃描「White」作為*lastname*元素的內容。現行的掃描器重新設定為基本掃描器。

Lastname結束標籤名稱

藉由步驟1106，基本XML剖析器呼叫現行的掃描器，以獲取下一符記：*lastname*結束標籤名稱1614的符記。藉由步驟1108，符記掃描成功，再藉由步驟1112，基本XML剖析器1004決定其為一元素符記，然後藉由步驟1114，將其作為一詞素輸入XML網目確認模組1012中。之後，藉由步驟1116，XML網目確認模組1012決定該詞素類型。由於該詞素為一結束標籤名稱，故執行處理步驟1122，如圖14所示。

參考圖14，藉由步驟1402，從堆疊中移除lastname元素標註記錄829，即先前的標註記錄。現在堆疊的內容變為圖15B中所示。藉由步驟1404，XML網目確認模組1012從lastname元素標註記錄829中，獲取*lastname*結束標籤符記(*LE*)826，並將該符記826輸入元素確認模組1014中。藉由

步驟1406，確認成功。由於藉由步驟1408，lastname元素標註記錄829的屬性列表827為空，所以程序返回至圖11中的步驟1106，並對下一符記重複上述步驟。

Firstname起始標籤名稱

藉由步驟1106，基本XML剖析器呼叫現行的掃描器，以獲取下一符記：*firstname*起始標籤名稱1616的符記。藉由步驟1108，符記掃描成功，再藉由步驟1112，基本XML剖析器1004決定其為一元素符記，然後藉由步驟1114，將其作為一詞素輸入XML網目確認模組1012中。之後，藉由步驟1116，XML網目確認模組1012決定該詞素類型。由於該詞素為一起始標籤名稱，故執行處理步驟1118，如圖12所示。

參考圖12，employee元素標註記錄816的候選子元素映像815，即先前的標註記錄，指向firstname起始標籤名稱830，其藉由步驟1202，又指向firstname元素標註記錄837，即現行的標註記錄，再藉由步驟1206，將其推至堆疊上。現在堆疊的內容變為圖15D中所示。藉由步驟1208，XML網目確認模組1012從*firstname*元素標註記錄837中，獲取*firstname*起始標籤符記(*FS*)833，並將該符記833輸入元素確認模組1014中。藉由步驟1210，確認成功。由於藉由步驟1214，firstname元素標註記錄837的屬性列表835為空，且*firstname*元素的內容為簡單類型(stringTypeID)，故藉由步驟1216，將現行的掃描器ID改為stringTypeID。然後，程序返回至圖11中的步驟1106，對下一符記重複上述步驟。

隨後，呼叫StringType掃描器1020，掃描「John」作為*firstname*元素的內容。現行的掃描器重新設定為基本掃描器。

Firstname結束標籤名稱

藉由步驟1106，基本XML剖析器呼叫現行的掃描器，以獲取下一符記：*firstname*結束標籤名稱1618的符記。藉由步驟1108，符記掃描成功，再藉由步驟1112，基本XML剖析器1004決定其為一元素符記，然後藉由步驟1114，將其作為一詞素輸入XML網目確認模組1012中。之後，藉由步驟1116，XML網目確認模組1012決定該詞素類型。由於該詞素為一結束標籤名稱，故執行處理步驟1122，如圖14所示。

參考圖14，藉由步驟1402，從堆疊中移除firstname元素標註記錄837，即先前的標註記錄。現在堆疊的內容變為圖15B中所示。藉由步驟1404，XML網目確認模組1012從firstname元素標註記錄837中，獲取*firstname*結束標籤符記(*SE*)834，並將該符記834輸入元素確認模組1014中。藉由步驟1406，確認成功。由於藉由步驟1408，firstname元素標註記錄837的屬性列表835為空，所以程序返回至圖11中的步驟1106，並對下一符記重複上述步驟。

Notes起始標籤名稱

藉由步驟1106，基本XML剖析器呼叫現行的掃描器，以獲取下一符記：*notes*起始標籤名稱1620的符記。藉由步驟1108，符記掃描成功，再藉由步驟1112，基本XML剖析器

1004決定其為一元素符記，然後藉由步驟1114，將其作為一詞素輸入XML綱目確認模組1012中。之後，藉由步驟1116，XML綱目確認模組1012決定該詞素類型。由於該詞素為一起始標籤名稱，故執行處理步驟1118，如圖12所示。

參考圖12，employee元素標註記錄816的候選子元素映像815，即先前的標註記錄，指向*notes*起始標籤名稱838，其藉由步驟1202，又指向notes元素標註記錄843，即現行的標註記錄，再藉由步驟1206，將其推至堆疊上。現在堆疊的內容變為圖15E中所示。藉由步驟1208，XML綱目確認模組1012從notes元素標註記錄843中，獲取*notes*起始標籤符記(*NS*)839，並將該符記839輸入元素確認模組1014中。藉由步驟1210，確認成功。由於藉由步驟1214，notes元素標註記錄843的屬性列表835為空，且*notes*元素的內容為簡單類型(stringTypeID)，故藉由步驟1216，將現行的掃描器ID改為stringTypeID。然後，程序返回至圖11中的步驟1106，對下一符記重複上述步驟。StringType掃描器1020掃描「regular(正常)」，作為*notes*元素的內容。現行的掃描器重新設定為基本掃描器。

Notes結束標籤名稱

藉由步驟1106，基本XML剖析器呼叫現行的掃描器，以獲取下一符記：*notes*結束標籤名稱1622的符記。藉由步驟1108，符記掃描成功，再藉由步驟1112，基本XML剖析器1004決定其為一元素符記，然後藉由步驟1114，將其作為一詞素輸入XML綱目確認模組1012中。之後，藉由步驟

1116，XML網目確認模組1012決定該詞素類型。由於該詞素為一結束標籤名稱，故執行處理步驟1122，如圖14所示。

參考圖14，藉由步驟1402，從堆疊中移除notes元素標註記錄843，即先前的標註記錄。現在堆疊的內容變為圖15B中所示。藉由步驟1404，XML網目確認模組1012從notes元素標註記錄843中，獲取*notes*結束標籤符記(*NE*)840，並將該符記840輸入元素確認模組1014中。藉由步驟1406，確認成功。由於藉由步驟1408，notes元素標註記錄843的屬性列表841為空，所以程序返回至圖11中的步驟1106，並對下一符記重複上述步驟。

Employee結束標籤名稱

藉由步驟1106，基本XML剖析器呼叫現行的掃描器，以獲取下一符記：*employee*結束標籤名稱1624的符記。藉由步驟1108，符記掃描成功，再藉由步驟1112，基本XML剖析器1004決定其為一元素符記，然後藉由步驟1114，將其作為一詞素輸入XML網目確認模組1012中。之後，藉由步驟1116，XML網目確認模組1012決定該詞素類型。由於該詞素為一結束標籤名稱，故執行處理步驟1122，如圖14所示。

參考圖14，藉由步驟1402，從堆疊中移除employee元素標註記錄816，即先前的標註記錄。現在堆疊的內容變為圖15A中所示。藉由步驟1404，XML網目確認模組1012從employee元素標註記錄816中，獲取*employee*結束標籤符記(*EE*)813，並將該符記813輸入元素確認模組1014中。藉由

步驟1406，確認成功。由於藉由步驟1408，employee元素標註記錄816的所有屬性817、819及821均已確認，所以程序返回至圖11中的步驟1106，並對下一符記重複上述步驟。

對於範例XML文件1600中*personnel*元素下其餘的*employee*元素，可執行類似的確認程序。

Personnel結束標籤名稱

藉由步驟1106，基本XML剖析器呼叫現行的掃描器，以獲取下一符記：*personne*結束標籤名稱1626的符記。藉由步驟1108，符記掃描成功，再藉由步驟1112，基本XML剖析器1004決定其為一元素符記，然後藉由步驟1114，將其作為一詞素輸入XML網目確認模組1012中。之後，藉由步驟1116，XML網目確認模組1012決定該詞素類型。由於該詞素為一結束標籤名稱，故執行處理步驟1122，如圖14所示。

參考圖14，藉由步驟1402，從堆疊中移除personnel元素標註記錄809，即先前的標註記錄。藉由步驟1404，XML網目確認模組1012從personnel元素標註記錄809中，獲取*personnel*結束標籤符記(*PE*)806，並將該符記806輸入元素確認模組1014中。藉由步驟1406，確認成功。由於藉由步驟1408，personnel元素標註記錄809的屬性列表807為空，所以程序返回至圖11中的步驟1106，並對下一符記重複上述步驟。然後，返回一EOF符記。由於藉由步驟1125，基本XML剖析器1004與元素確認模組1014皆處於「接受」模式，故程序完成了對XML文件1600的確認。因此，藉由步

驟1126，返回一「有效」。

先進特徵

儘管以上已用簡單的XML綱目定義說明瞭較佳具體實施例，但本發明也可支援一些先進特徵。例如，可支援一導出類型元素，或一抽象類型及/或抽象元素與替代群組的宣告。

類型推導允許用基礎類型定義新類型。宣告具有導出類型的元素類型時，該宣告僅引用可抽象的基礎類型。若一元素宣告為一具有導出類型的類型，則必須用*xsi:type*屬性來決定在一XML例中，使用抽象類型的哪種導出類型。換言之，不能僅用元素的名稱來決定元素的資料類型。但是，編譯XML綱目定義時，就已知道哪個元素由抽象類型宣告。因而，由抽象類型欄將該資訊編碼成總體與局部元素映像。若運行時間確認引擎104發現在運行時間，用一抽象類型宣告一元素，則其將延緩對該元素屬性的確認，直至找到*xsi:type*。

例如，圖17說明一XML綱目定義的範例導出類型定義之一部分。在該範例中，定義了*employee*元素的三個類型，而非一單一類型：employeeType、regularEmployee與tempEmployee。對employee元素的宣告稱為基礎類型「employeeType」。

依據本發明，為支援XML綱目確認中的導出類型，XML綱目編譯102使用CFG中*employee*元素的一OR(「｜」)：

employee: employeeType

| regularEmployee

| tempEmployee

RegularEmployee(正式員工)與tempEmployee(臨時員工)的BNF皆擴展為包括其含有的所有元素。例如，regularEmployee將擴展為包括所有四個元素：lastname、firstname、notes及salary(薪金)。依據XML綱目建議對XML綱目的限制，不應為任何對於將為合法的XML綱目之最終CFGs所需的預見(lookahead)。其意味著，最終的CFG應在一SLR或LALR(1)剖析器產生器的能力範圍內，否則XML綱目定義會出錯。其餘的綱目編譯將保持不變。

對於運行時間確認引擎104，可以兩種方式完成對employee元素的確認。在第一種方式中，使用一回溯機制。當運行時間確認引擎104遇到可為任何導出類型的元素時，其掃描起始標籤，以獲取*xsi:type*屬性。基於此等屬性的值，其為運行時間綱目確認剖析器1006產生對應的詞素。然後，其重新掃描標籤名稱之後的起始標籤。例如，若遇到下列起始標籤：

<employee xsi:type = “regularEmployee”>

則將一regularEmployee詞素傳送至高階確認剖析器。

在第二種方式中，基本XML剖析器1004將整個起始標籤視為整體，確認剖析器1006確認其內容。基本XML剖析器1004已知*xsi:type*屬性的值，並為運行時間綱目確認剖析器1006產生恰當的符記。然後，運行時間綱目確認剖析器1006確認整個起始標籤。

對於抽象元素及元素替代群組，可由XML綱目編譯102，在綱目編譯中應用類似的機制。然而，與抽象類型不同，可從元素名稱得知元素的類型。因而，無需掃描來獲取*xsi:type*屬性。

儘管在XML綱目的背景下對本發明予以了說明，但熟悉技術人士應明白，對於其他利用綱目的標籤及/或標示語言，也可使用本確認程序，而不致背離本發明的精神與範疇。

結論

以上已揭示了一種改良的XML綱目確認方法與系統。該方法與系統將XML綱目定義編譯為一標註自動機編碼(AAE)格式，儲存在碟片或資料庫中。各XML綱目定義係一次性編譯為AAE格式。確認一XML文件時，將該XML文件及AAE格式之適當XML綱目定義載入運行時間確認引擎中。該運行時間確認引擎包括一基本XML剖析器與一運行時間綱目確認剖析器。運行時間確認引擎的代碼係固定的，不會隨XML綱目定義而變化。當運行時間綱目確認剖析器根據AAE格式的XML綱目定義，對XML文件執行一高階確認時，基本XML剖析器執行一低階確認。運行時間確認引擎的輸出為一確認通過或確認失敗。

由於AAE格式中的XML綱目定義係一次性編譯並儲存，而不是每次確認一XML文件時進行編譯，從而，明顯節省了執行確認程序的時間。此外，因運行時間確認引擎代碼係固定的，而不會隨各XML綱目定義而不同，故即使存在

大量的XML綱目定義，也可使空間負擔最小。而且，該確認程序具有靈活性，對效能沒有影響。

儘管依據所示的具體實施例，對本發明進行了說明，但熟悉技術人士會明白，可對該等具體實施例進行更改，而且此等更改也在本發明的精神與範疇內。因此，熟悉技術人士可做許多修改，而不會脫離隨附的申請專利範圍之精神與範疇。

【圖式簡單說明】

圖1說明依據本發明，一XML綱目確認系統的一項較佳具體實施例。

圖2詳細說明依據本發明的XML綱目編譯。

圖3為一流程圖，說明依據本發明XML綱目編譯程序的一項較佳具體實施例。

圖4說明一範例XML綱目定義。

圖5說明依據本發明，以一標註樹形式表示的範例XML綱目定義的一元素結構階層。

圖6說明依據本發明，一上下文無關文法的範例XML綱目定義的結構。

圖7說明依據本發明，用起始標籤符記與結束標籤符記，對圖6中的上下文無關文法的擴充。

圖8說明依據本發明，對範例XML綱目定義的標註自動機編碼的一標註部分。

圖9說明依據本發明，範例XML綱目定義的自動機編碼表之狀態變遷圖。

圖10為詳細說明依據本發明的運行時間確認引擎。

圖11為一流程圖，說明依據本發明，由運行時間確認引擎執行的運行時間確認程序的一項較佳具體實施例。

圖12為一流程圖，說明依據本發明，對一起始標籤名稱的處理。

圖13為一流程圖，說明依據本發明，對一屬性名稱的處理。

圖14為一流程圖，說明依據本發明，對一結束標籤名稱的處理。

圖15A至15E說明依據本發明，在運行時間確認引擎執行運行時間確認程序期間，堆疊中的內容。

圖16說明將確認的一範例XML文件。

圖17說明一XML綱目定義的導出類型之一範例。

【圖式代表符號說明】

102	XML綱目編譯
104	運行時間確認引擎
106	XML綱目定義
108	標註自動機編碼(AAE)格式
110	資料庫
112	XML文件
202	XML綱目編譯器前端
204	XML綱目編譯器後端
302	步驟
304	步驟

306 步驟

308 步驟

400 範例XML綱目定義

402 預設的XML名稱空間

404 名稱空間宣告

405 註冊的目標名稱空間

406 標註

408 元素

410 屬性

412 總體元素宣告(*personnel*，全體人員)

414 子元素序列(*employee*，員工)

416 語義約束

418 子元素

420 語法約束

422 屬性

424 簡單類型

426 總體元素宣告(*notes*，備註)

430 唯一性約束

500 標註樹

502 節點

504 節點

506 節點

508 節點

510 節點

512	節點
514	虛線橢圓「字串」(資料類型標註節點)
516	虛線橢圓「字串」(資料類型標註節點)
518	虛線橢圓「字串」(資料類型標註節點)
520	serno方塊
522	userid方塊
524	department方塊
526	虛線橢圓(USERID_TYPE)
528	虛線橢圓(Integer)
530	虛線橢圓(String)
800	標註自動機編碼
801	自動機編碼表
802	總體元素(*notes*)
803	總體元素映像(personnel)
804	元素內容及引數
805	起始標籤符記
806	結束標籤符記
807	屬性列表
808	候選子元素映像
809	標註記錄
810	局部元素映像
811	元素內容及引數
812	起始標籤符記
813	結束標籤符記

814 屬性列表

815 候選子元素映像

816 元素標註記錄(employee)

817 屬性名稱(*serno*)

818 值及引數

819 屬性名稱(*userid*)

820 值及引數

821 屬性名稱(department)

822 值及引數

823 起始標籤名稱(*lastname*)

824 內容及引數

825 起始標籤符記

826 結束標籤符記

827 屬性列表

828 候選子元素映像

829 元素標註記錄(*lastname*)

830 起始標籤名稱(firstname)

832 內容及引數

833 起始標籤符記

834 結束標籤符記

835 屬性列表

836 候選子元素映像

837 元素標註記錄(firstname)

838 內容及引數

839	起始標籤符記
840	結束標籤符記
841	屬性列表
842	候選子元素映像
843	元素標註記錄(*notes*)
1002	XML 掃描器集用場
1004	基本XML剖析器
1006	運行時間網目確認剖析器
1008	實體管理者
1010	XML 網目載入模組
1012	XML 網目確認模組
1014	元素確認模組
1016	屬性確認模組
1017	基本掃描器
1018	掃描器
1019	掃描器
1020	掃描器
1102	步驟
1104	步驟
1106	步驟
1108	步驟
1110	步驟
1112	步驟
1114	步驟

1116 步驟

1118 步驟

1120 步驟

1122 步驟

1124 步驟

1125 步驟

1126 步驟

1127 步驟

1202 步驟

1206 步驟

1208 步驟

1210 步驟

1212 步驟

1214 步驟

1216 步驟

1301 步驟

1302 步驟

1304 步驟

1310 步驟

1312 步驟

1402 步驟

1404 步驟

1406 步驟

1408 步驟

1410	步驟
1600	範例XML文件
1602	起始標籤名稱(*personnel*)
1604	起始標籤名稱(*employee*)
1606	屬性名稱(*serno*)
1608	屬性名稱(*userid*)
1610	屬性名稱(department)
1612	起始標籤名稱(*lastname*)
1614	結束標籤名稱(*lastname*)
1616	起始標籤名稱(*firstname*)
1618	結束標籤名稱(*firstname*)
1620	起始標籤名稱(*notes*)
1622	結束標籤名稱(*notes*)
1624	結束標籤名稱(*employee*)
1626	結束標籤名稱(*personne*)

伍、中文發明摘要：

本發明揭示一種用於可擴充標示語言(Extensible Markup Language；XML)網目確認的方法與系統，其包括：將一XML文件載入一運行時間確認引擎中，其中該運行時間確認引擎包括一XML網目確認剖析器；將一XML網目定義的一標註自動機編碼(annotated automaton encoding；AAE)載入該XML網目確認剖析器；及由該XML網目確認剖析器利用該標註自動機編碼，根據該XML網目定義，確認該XML文件。每一XML網目定義係一次性編譯成該AAE格式，而不是每次確認一XML文件時編譯，從而明顯節省了時間。該運行時間確認引擎的代碼係固定的，不會隨該XML網目定義而變化，而是對於各XML網目定義不同，從而使空間負擔最小化。該確認程序具有靈活性，而且不影響效能。

陸、英文發明摘要：

A method and system for Extensible Markup Language (XML) schema validation, includes: loading an XML document into a runtime validation engine, where the runtime validation engine includes an XML schema validation parser; loading an annotated automaton encoding (AAE) for an XML schema definition into the XML schema validation parser; and validating the XML document against the XML schema definition by the XML schema validation parser utilizing the annotated automaton encoding. Each XML schema definition is compiled once into the AAE format, rather than being compiled each time an XML document is validated, and thus significant time is saved. The code for the runtime validation engine is fixed and does not vary depending on the XML schema definition, rather than varying for each XML schema definition, and thus space overhead is minimized. Flexibility in the validation process is provided without compromising performance.

拾、申請專利範圍：

1. 一種用於標示語言網目確認的方法，其包括以下步驟：

(a) 將一標示語言文件載入一運行時間確認引擎中，其中該運行時間確認引擎包括一標示語言網目確認剖析器；

(b) 將對於一標示語言網目定義的一標註自動機編碼載入該標示語言網目確認剖析器中；及

(c) 藉由該標示語言網目確認剖析器利用該標註自動機編碼，根據該標示語言網目定義，確認該標示語言文件。

2. 如申請專利範圍第1項之方法，其中該標示語言包括一可擴充標示語言(XML)。

3. 如申請專利範圍第1項之方法，其中該標註自動機編碼包括至少一元素節點，其中一或多個屬性可與該元素節點相關聯，且其中一或多個資料類型約束可與該元素節點或該屬性相關聯。

4. 如申請專利範圍第1項之方法，其中該標註自動機編碼包括該至少一元素節點的至少一元素標註記錄，其中該至少一元素標註記錄包括由以下所組成的一群組中的一或多個：

一元素內容與引數的一掃描器ID；

一起始標籤符記；

一結束標籤符記；

一屬性列表；及

一候選子元素映像，其能夠包括一子元素名稱的一指標。

5. 如申請專利範圍第1項之方法，其中在該載入步驟(a)之前，包括：

(a1) 接收一可擴充標示語言(XML)綱目定義；

(a2) 為該XML綱目定義產生一元素結構階層，並以一標註樹的形式表示該階層；

(a3) 對該標註樹進行編碼，並產生該標註自動機編碼；

(a4) 使該標註自動機編碼序列化；及

(a5) 儲存該序列化的標註自動機編碼。

6. 如申請專利範圍第1項之方法，其中該確認步驟(c)包括：

(c1) 獲得一可擴充標示語言(XML)文件的至少一符記；

(c2) 藉由一基本XML剖析器，對該至少一符記執行一低階確認；及

(c3) 若該符記為一元素符記或一屬性符記，則藉由一XML綱目確認剖析器，對該至少一符記執行一高階確認。

7. 如申請專利範圍第6項之方法，其中該確認步驟(c)進一步包括：

(c4) 若由該基本XML剖析器與該XML綱目確認剖析器進行的該等確認皆成功，則輸出一確認通過；及

(c5) 若由該基本XML剖析器或該XML綱目確認剖析器進行的該確認不成功，則輸出一確認失敗。

8. 如申請專利範圍第6項之方法，其中該元素符記包括由以

下所組成的一群組中的一或多個：

一起始標籤名稱；及

一結束標籤名稱。

9. 如申請專利範圍第6項之方法，其中該屬性符記包括一屬性名稱。

10. 如申請專利範圍第6項之方法，其中，若該元素符記為一起始標籤名稱，則該執行步驟(c3)包括：

(c3i) 根據一先前的標註記錄及該起始標籤名稱，找出一現行的標註記錄；

(c3ii) 將該現行標註記錄推至一堆疊上；

(c3iii) 從該現行標註記錄中，獲取該起始標籤名稱的一起始標籤符記；

(c3iv) 將該起始標籤符記輸入該XML綱目確認剖析器的一元素確認模組中；及

(c3v) 決定對該起始標籤符記的一確認是否成功。

11. 如申請專利範圍第6項之方法，其中，若該屬性符記為一屬性名稱，則該執行步驟(c3)包括：

(c3i) 將一現行標註記錄及該屬性名稱，傳送至該XML綱目確認剖析器的一屬性確認模組；

(c3ii) 搜尋該屬性名稱的該現行標註記錄之一屬性列表，其中，若在該現行標註記錄中未找到該屬性名稱，則對該XML文件的確認失敗；

(c3iii) 若在該現行標註記錄中找到該屬性名稱，則獲取該屬性符記；及

(c3iv) 決定對該屬性符記的該確認是否成功。

12. 如申請專利範圍第6項之方法，其中，若該元素符記為一結束標籤名稱，則該執行步驟(c3)包括：

(c3i) 從一堆疊中移除一現行標註記錄；

(c3ii) 從該現行標註記錄中，獲取一結束標籤符記；

(c3iii) 將該結束標籤符記輸入該XML網目確認剖析器的一元素確認模組中；

(c3iv) 決定對該結束標籤符記的該確認是否成功；及

(c3v) 決定該現行標註記錄的所有屬性是否皆已確認，或該現行標註記錄的該屬性列表是否為空，其中，若並非該現行標註記錄的所有該等屬性皆已確認，且該現行標註記錄的該屬性列表不為空，則對該結束標籤符記的確認不成功。

13. 一種系統，其包括：

一標示語言網目編譯，用於為至少一標示語言網目定義，產生至少一標註自動機編碼；及

一運行時間確認引擎，其包括一運行時間網目確認剖析器，其中該運行時間網目確認剖析器接收一標示語言文件及該至少一標註自動機編碼作為輸入，其中該運行時間網目確認剖析器利用該至少一標註自動機編碼，根據該至少一標示語言網目定義，確認該標示語言文件。

14. 如申請專利範圍第13項之系統，其中該標示語言包括一可擴充標示語言(XML)。

15. 如申請專利範圍第13項之系統，其中該標註自動機編碼

包括至少一元素節點，其中一或多個屬性可與該元素節點相關聯，且其中一或多個資料類型約束可與該元素節點或該屬性相關聯。

16. 如申請專利範圍第13項之系統，其中該標註自動機編碼包括該至少一元素節點的至少一元素標註記錄，其中該至少一元素標註記錄包括由以下所組成的一群組中的一或多個：

一元素內容與引數的一掃描器ID；

一起始標籤符記；

一結束標籤符記；

一屬性列表；及

一候選子元素映像，其能夠包括一子元素名稱的一指標。

17. 如申請專利範圍第13項之系統，其中該標示語言綱目編譯包括：

一可擴充標示語言(XML)綱目編譯器前端；及

一XML綱目編譯器後端。

18. 如申請專利範圍第17項之系統，其中該XML綱目編譯器前端接收該至少一XML綱目定義，為該XML綱目定義產生一元素結構階層，並以一標註樹的形式表示該階層。

19. 如申請專利範圍第18項之系統，其中該XML綱目編譯器後端對該標註樹進行編碼，從該編碼的標註樹中產生該至少一標註自動機編碼，並使該至少一標註自動機編碼序列化。

20. 如申請專利範圍第13項之系統，其進一步包括：

一儲存媒體，用於儲存該至少一標註自動機編碼。

21. 如申請專利範圍第13項之系統，其中該運行時間確認引擎進一步包括：

一基本可擴充標示語言(XML)剖析器，其中該基本XML剖析器對一XML文件執行一低階確認，其中該運行時間網目確認剖析器對該XML文件執行一高階確認。

22. 如申請專利範圍第13項之系統，其中該運行時間網目確認剖析器包括：

一可擴充標示語言(XML)網目載入模組，用於載入該至少一標註自動機編碼；及

一XML網目確認模組，其包括：

一元素確認模組，用於確認元素符記；及

一屬性確認模組，用於確認屬性符記。

23. 如申請專利範圍第22項之系統，其中該元素符記包括由以下所組成的一群組中的一或多個：

一起始標籤名稱；及

一結束標籤名稱。

24. 如申請專利範圍第22項之系統，其中該屬性符記包括一屬性名稱。

25. 如申請專利範圍第13項之系統，其中該運行時間確認引擎進一步包括一可擴充標示語言(XML)掃描器集用場，其中該XML掃描器集用場包括一基本掃描器與至少一類型特定掃描器。

26. 一種具有用於一標示語言綱目確認之程式指令的電腦可讀取媒體，其包括的該等指令用於：

(a) 將一標示語言文件載入一運行時間確認引擎中，其中該運行時間確認引擎包括一標示語言綱目確認剖析器；

(b) 將一標示語言綱目定義的一標註自動機編碼，載入該標示語言綱目確認剖析器中；及

(c) 藉由該標示語言綱目確認剖析器利用該標註自動機編碼，根據該標示語言綱目定義，確認該標示語言文件。

27. 如申請專利範圍第26項之媒體，其中該標示語言包括一可擴充標示語言(XML)。

28. 如申請專利範圍第26項之媒體，其中該標註自動機編碼包括至少一元素節點，其中一或多個屬性可與該元素節點相關聯，且其中一或多個資料類型約束可與該元素節點或該屬性相關聯。

29. 如申請專利範圍第26項之媒體，其中該標註自動機編碼包括該至少一元素節點的至少一元素標註記錄，其中該至少一元素標註記錄包括由以下所組成的一群組中的一或多個：

一元素內容與引數的一掃描器ID；

一起始標籤符記；

一結束標籤符記；

一屬性列表；及

一候選子元素映像，其能夠包括一子元素名稱的一指

標。

30. 如申請專利範圍第26項之媒體，其中該載入指令(a)之前包括的該等指令用於：

(a1) 接收一可擴充標示語言(XML)綱目定義；

(a2) 為該XML綱目定義產生一元素結構階層，並以一標註樹的形式表示該階層；

(a3) 對該標註樹進行編碼，並產生該標註自動機編碼；

(a4) 使該標註自動機編碼序列化；及

(a5) 儲存該序列化的標註自動機編碼。

31. 如申請專利範圍第26項之媒體，其中該確認指令(c)包括的指令用於：

(c1) 獲得一可擴充標示語言(XML)文件的至少一符記；

(c2) 藉由一基本XML剖析器，對該至少一符記執行一低階確認；及

(c3) 若該符記為一元素符記或一屬性符記，則藉由一XML綱目確認剖析器，對該至少一符記執行一高階確認。

32. 如申請專利範圍第31項之媒體，其中該等確認指令(c)進一步包括的指令用於：

(c4) 若由該基本XML剖析器與該XML綱目確認剖析器進行的該等確認皆成功，則輸出一確認通過；及

(c5) 若由該基本XML剖析器或該XML綱目確認剖析器進行的該確認不成功，則輸出一確認失敗。

33. 如申請專利範圍第31項之媒體，其中該元素符記包括由

以下所組成的一群組中的一或多個：

一起始標籤名稱；及

一結束標籤名稱。

34. 如申請專利範圍第31項之媒體，其中該屬性符記包括一屬性名稱。

35. 如申請專利範圍第31項之媒體，其中，若該元素符記為一起始標籤名稱，則該執行指令(c3)包括的指令用於：

(c3i) 基於一先前的標註記錄及該起始標籤名稱，找出一現行標註記錄；

(c3ii) 將該現行標註記錄推至一堆疊上；

(c3iii) 從該現行標註記錄中，獲取該起始標籤名稱的一起始標籤符記；

(c3iv) 將該起始標籤符記輸入該XML綱目確認剖析器的一元素確認模組中；及

(c3v) 決定對該起始標籤符記的一確認是否成功。

36. 如申請專利範圍第31項之媒體，其中，若該屬性符記為一屬性名稱，則該執行指令(c3)包括的指令用於：

(c3i) 將一現行標註記錄及該屬性名稱，傳送至該XML綱目確認剖析器的一屬性確認模組；

(c3ii) 搜尋用於該屬性名稱的該現行標註記錄之一屬性列表，其中，若在該現行標註記錄中未找到該屬性符記，則對該XML文件的該確認失敗；

(c3iii) 若在該現行標註記錄中找到該屬性名稱，則獲取該屬性符記；及

(c3iv) 決定對該屬性符記的該確認是否成功。

37.如申請專利範圍第31項之媒體，其中，若該元素符記為一結束標籤名稱，則該執行指令(c3)包括的指令用於：

(c3i) 從一堆疊中移除一現行標註記錄；

(c3ii) 從該現行標註記錄中，獲取一結束標籤符記；

(c3iii) 將該結束標籤符記輸入該XML網目確認剖析器的一元素確認模組中；

(c3iv) 決定對該結束標籤符記的該確認是否成功；及

(c3v) 決定該現行標註記錄的所有屬性是否皆已確認，或該現行標註記錄的該屬性列表是否為空，其中，若並非該現行標註記錄的所有該等屬性皆已確認，且該現行標註記錄的該屬性列表不為空，則對該結束標籤符記的該確認不成功。

拾壹、圖式：

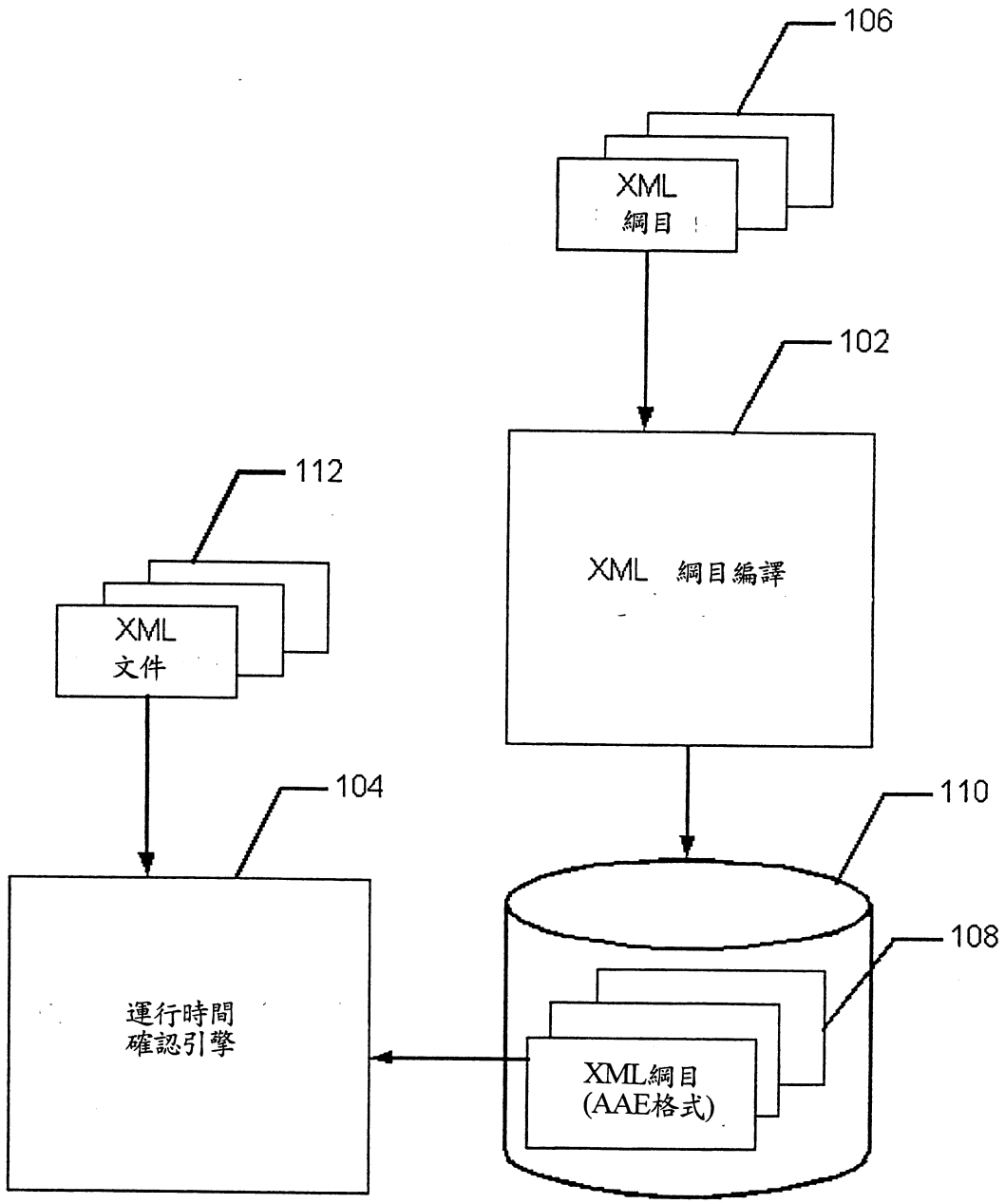

106
XML
網目
102
XML 網目編譯
112
XML
文件
104
110
108
運行時間
確認引擎
XML網目
(AAE格式)

圖1

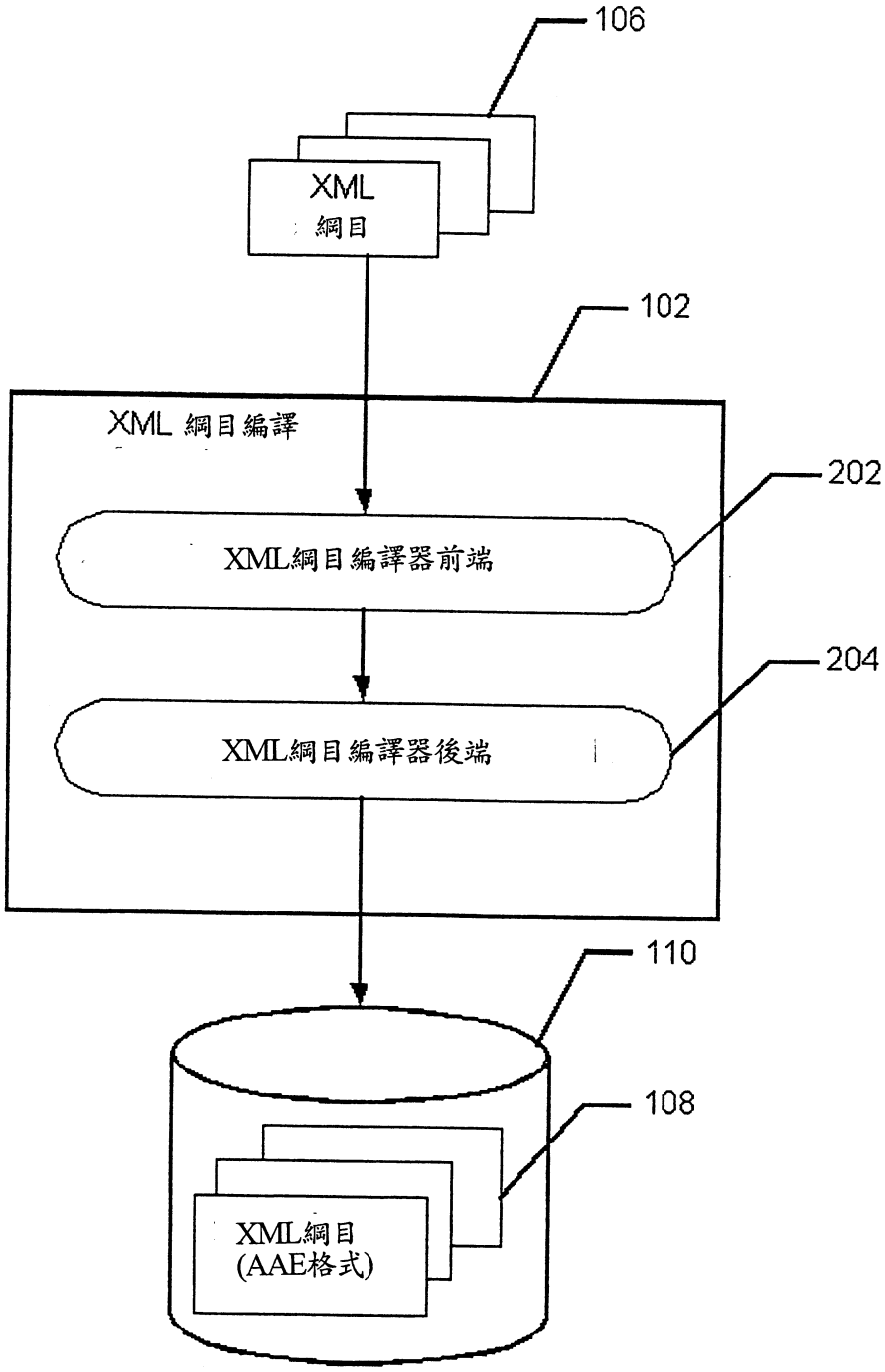

106
XML
網目
102
XML 網目編譯
202
XML網目編譯器前端
204
XML網目編譯器後端
110
108
XML網目
(AAE格式)

圖2

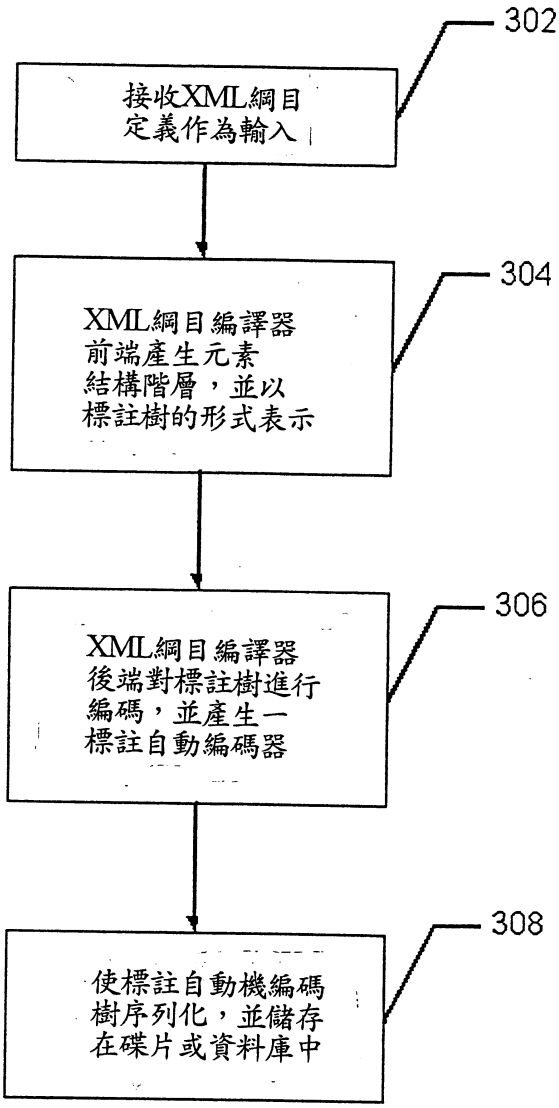

302
接收XML綱目
定義作為輸入
304
XML綱目編譯器
前端產生元素
結構階層，並以
標註樹的形式表示
306
XML綱目編譯器
後端對標註樹進行
編碼，並產生一
標註自動編碼器
308
使標註自動機編碼
樹序列化，並儲存
在碟片或資料庫中

圖3

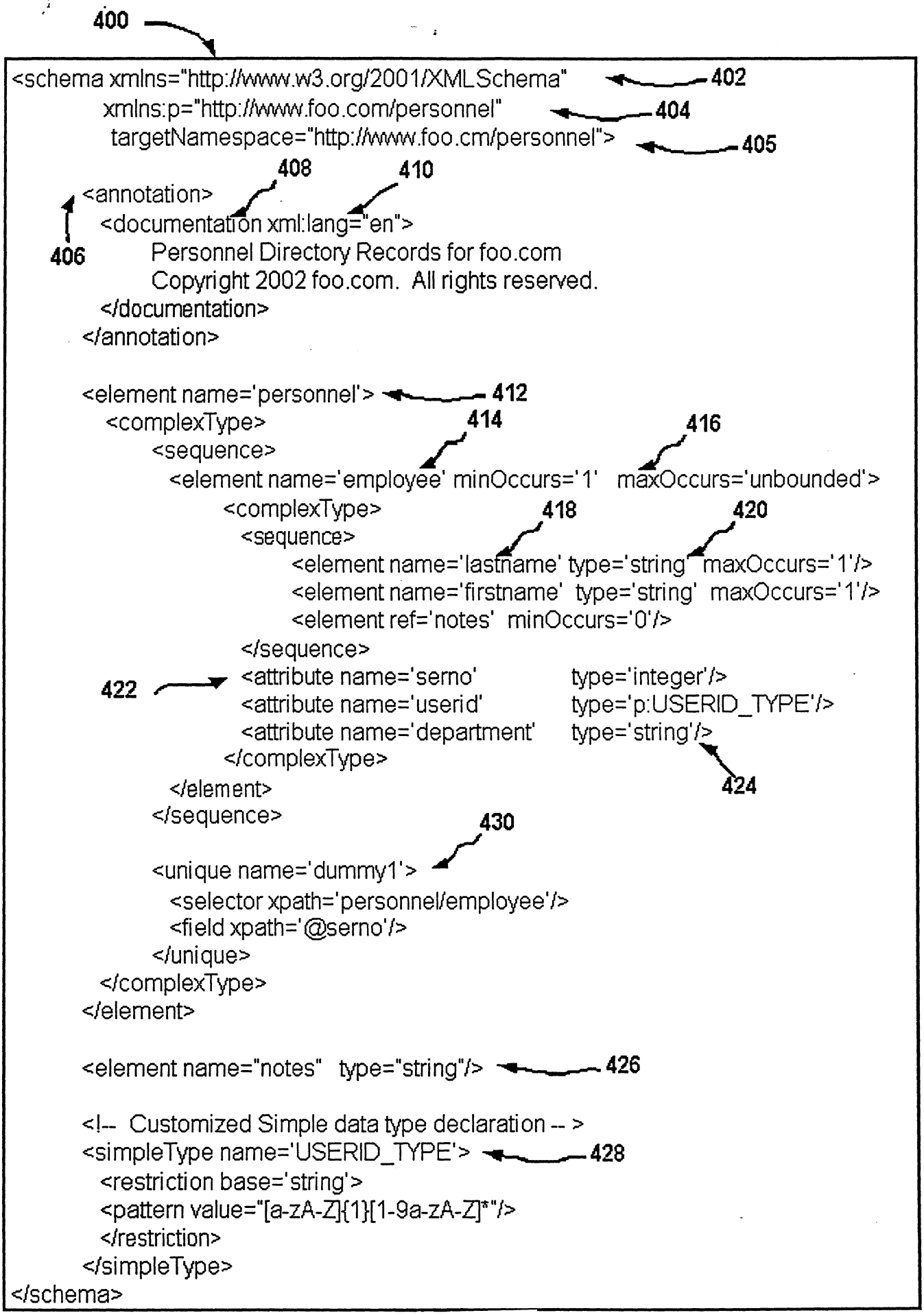

400
<schema xmlns="http://www.w3.org/2001/XMLSchema"
402
xmlns:p="http://www.foo.com/personnel"
404
targetNamespace="http://www.foo.cm/personnel">
405
408
410
<annotation>
<documentation xml:lang="en">
406
Personnel Directory Records for foo.com
Copyright 2002 foo.com. All rights reserved.
</documentation>
</annotation>
<element name='personnel'>
412
<complexType>
414
416
<sequence>
<element name='employee' minOccurs='1' maxOccurs='unbounded'>
<complexType>
418
420
<sequence>
<element name='lastname' type='string' maxOccurs='1'/>
<element name='firstname' type='string' maxOccurs='1'/>
<element ref='notes' minOccurs='0'/>
</sequence>
422
<attribute name='serno' type='integer'/>
<attribute name='userid' type='p:USERID_TYPE'/>
<attribute name='department' type='string'/>
424
</complexType>
</element>
</sequence>
430
<unique name='dummy1'>
<selector xpath='personnel/employee'/>
<field xpath='@serno'/>
</unique>
</complexType>
</element>
<element name="notes" type="string"/>
426
<!-- Customized Simple data type declaration -- >
<simpleType name='USERID_TYPE'>
428
<restriction base='string'>
<pattern value="[a-zA-Z]{1}[1-9a-zA-Z]*"/>
</restriction>
</simpleType>
</schema>

圖4

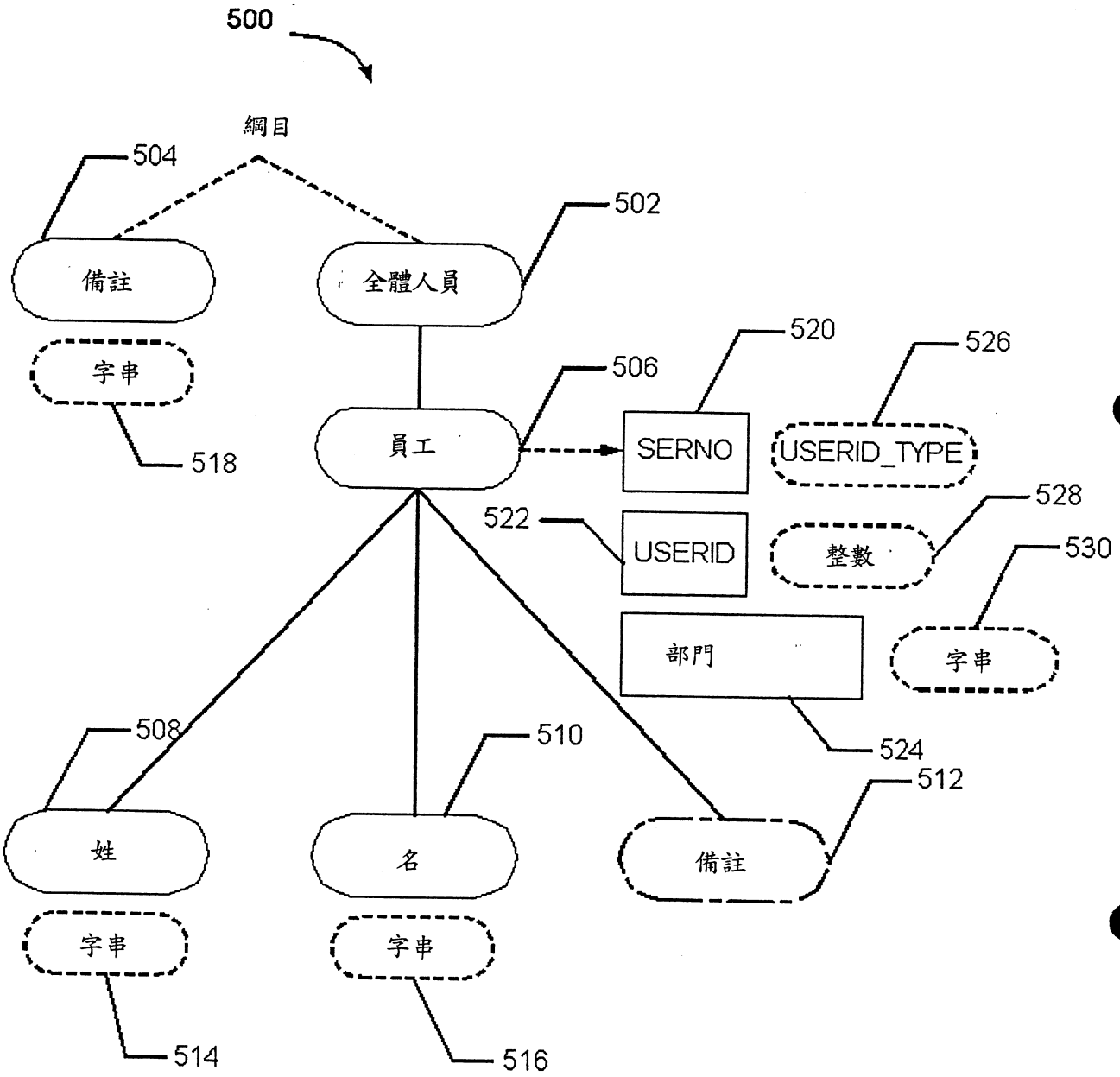

500
綱目
504
502
備註
全體人員
字串
518
員工
506
520
526
SERNO
USERID_TYPE
528
522
USERID
整數
530
部門
字串
524
508
510
512
姓
名
備註
字串
字串
514
516

圖5

```
personnel: employees

employees: employee
         | employee employee

employee: lastname firstname notes_01

notes_01: espilon
         | notes
```

圖6

```
personnel: PS employees PE

employees: employee
         | employee employees

employee: ES lastname firstname notes_01 EE

notes_01: epsilon
         | notes

lastname:   LS LE

firstname:  FS FE

notes:      NS NE
```

圖7

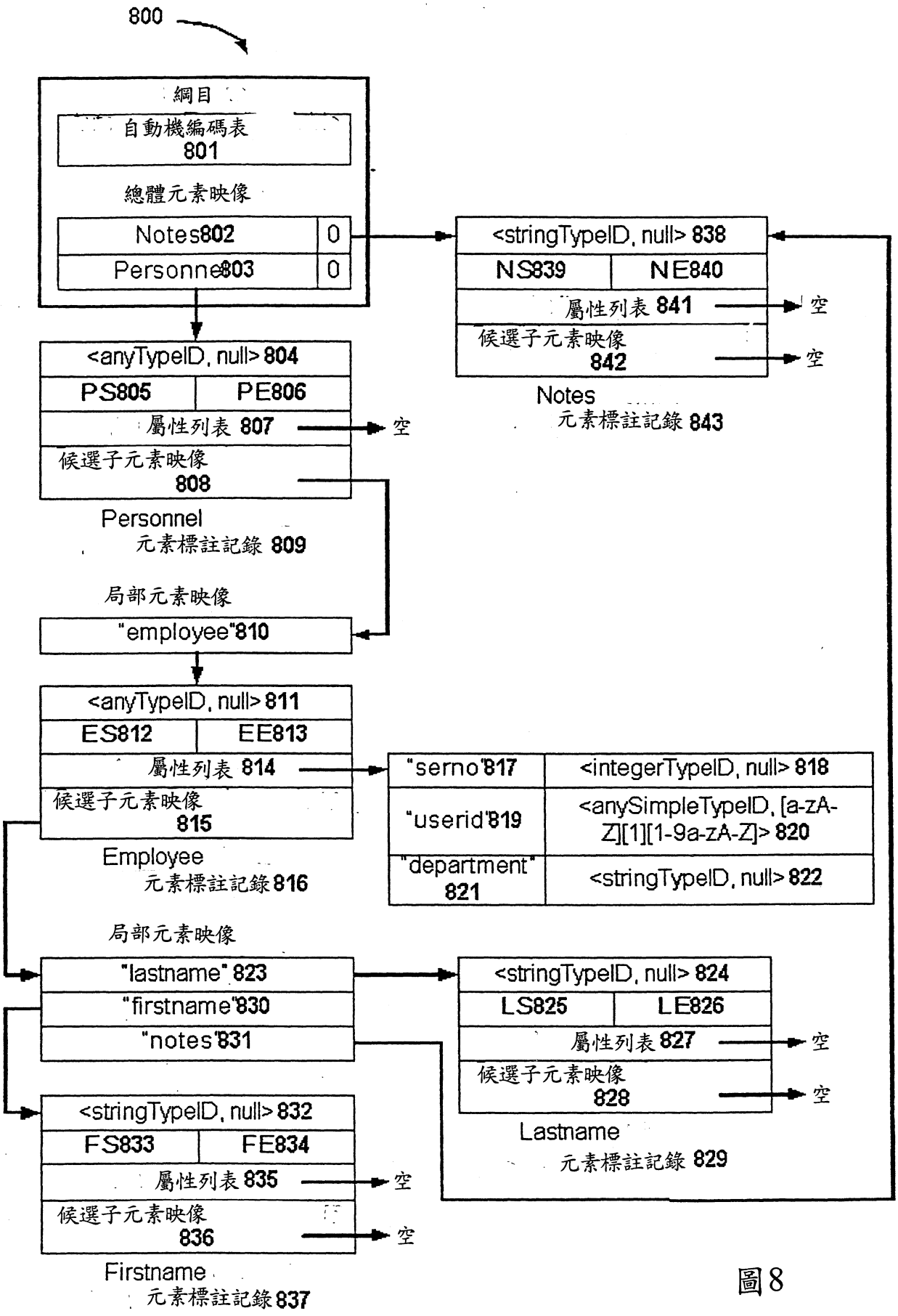

800
綱目
自動機編碼表 801
總體元素映像
Notes802
Personnel803
<stringTypeID, null> 838
NS839
NE840
屬性列表 841
空
候選子元素映像 842
空
Notes 元素標註記錄 843
<anyTypeID, null> 804
PS805
PE806
屬性列表 807
空
候選子元素映像 808
Personnel 元素標註記錄 809
局部元素映像
"employee"810
<anyTypeID, null> 811
ES812
EE813
屬性列表 814
候選子元素映像 815
Employee 元素標註記錄 816
"serno"817
<integerTypeID, null> 818
"userid"819
<anySimpleTypeID, [a-zA-Z][1][1-9a-zA-Z]> 820
"department" 821
<stringTypeID, null> 822
局部元素映像
"lastname" 823
"firstname"830
"notes"831
<stringTypeID, null> 824
LS825
LE826
屬性列表 827
空
候選子元素映像 828
空
Lastname 元素標註記錄 829
<stringTypeID, null> 832
FS833
FE834
屬性列表 835
空
候選子元素映像 836
空
Firstname 元素標註記錄 837

圖8

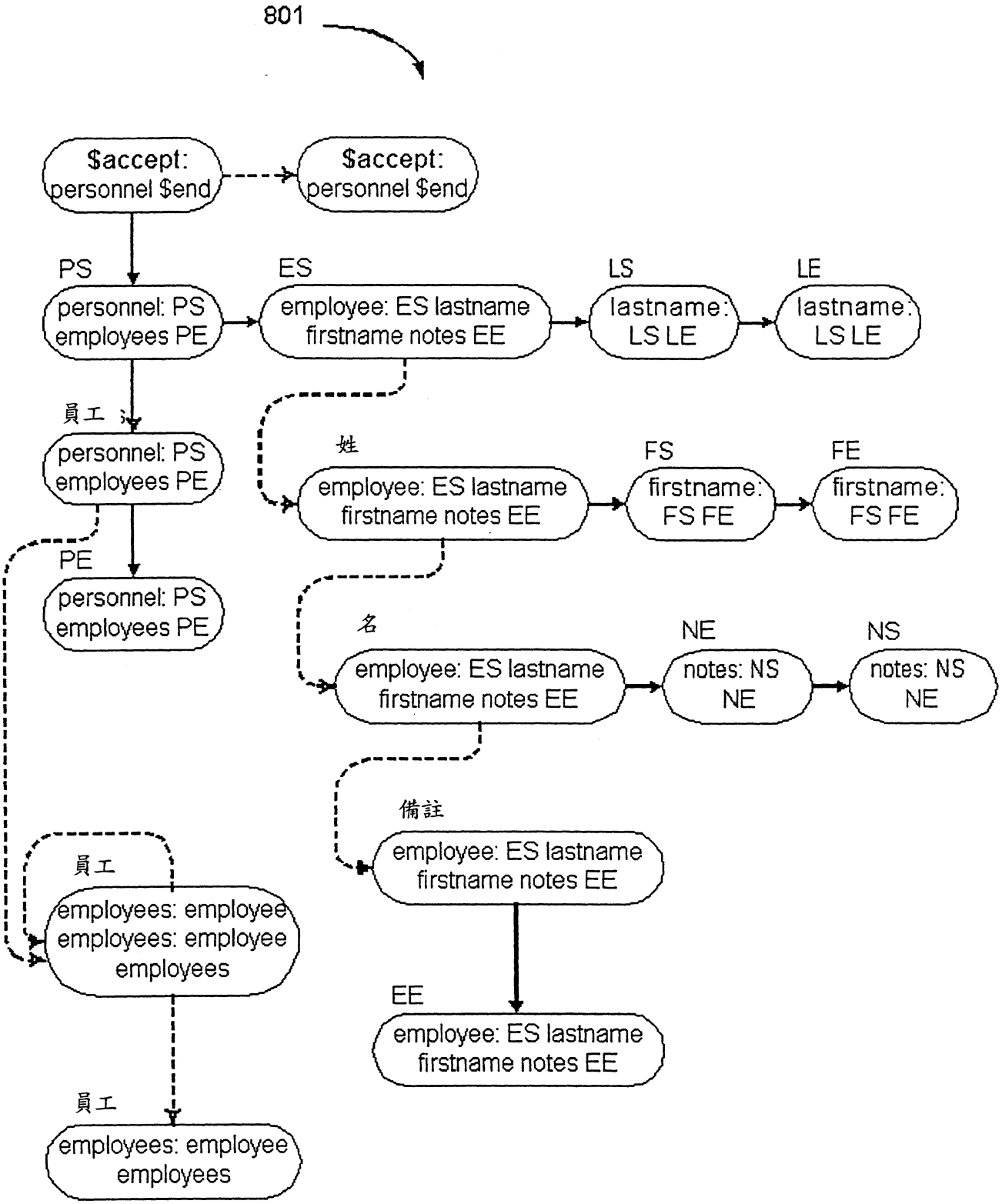

801
$accept: personnel $end
$accept: personnel $end
PS
personnel: PS employees PE
ES
employee: ES lastname firstname notes EE
LS
lastname: LS LE
LE
lastname: LS LE
員工
personnel: PS employees PE
姓
employee: ES lastname firstname notes EE
FS
firstname: FS FE
FE
firstname: FS FE
PE
personnel: PS employees PE
名
employee: ES lastname firstname notes EE
NE
notes: NS NE
NS
notes: NS NE
備註
employee: ES lastname firstname notes EE
員工
employees: employee employees: employee employees
EE
employee: ES lastname firstname notes EE
員工
employees: employee employees

圖9

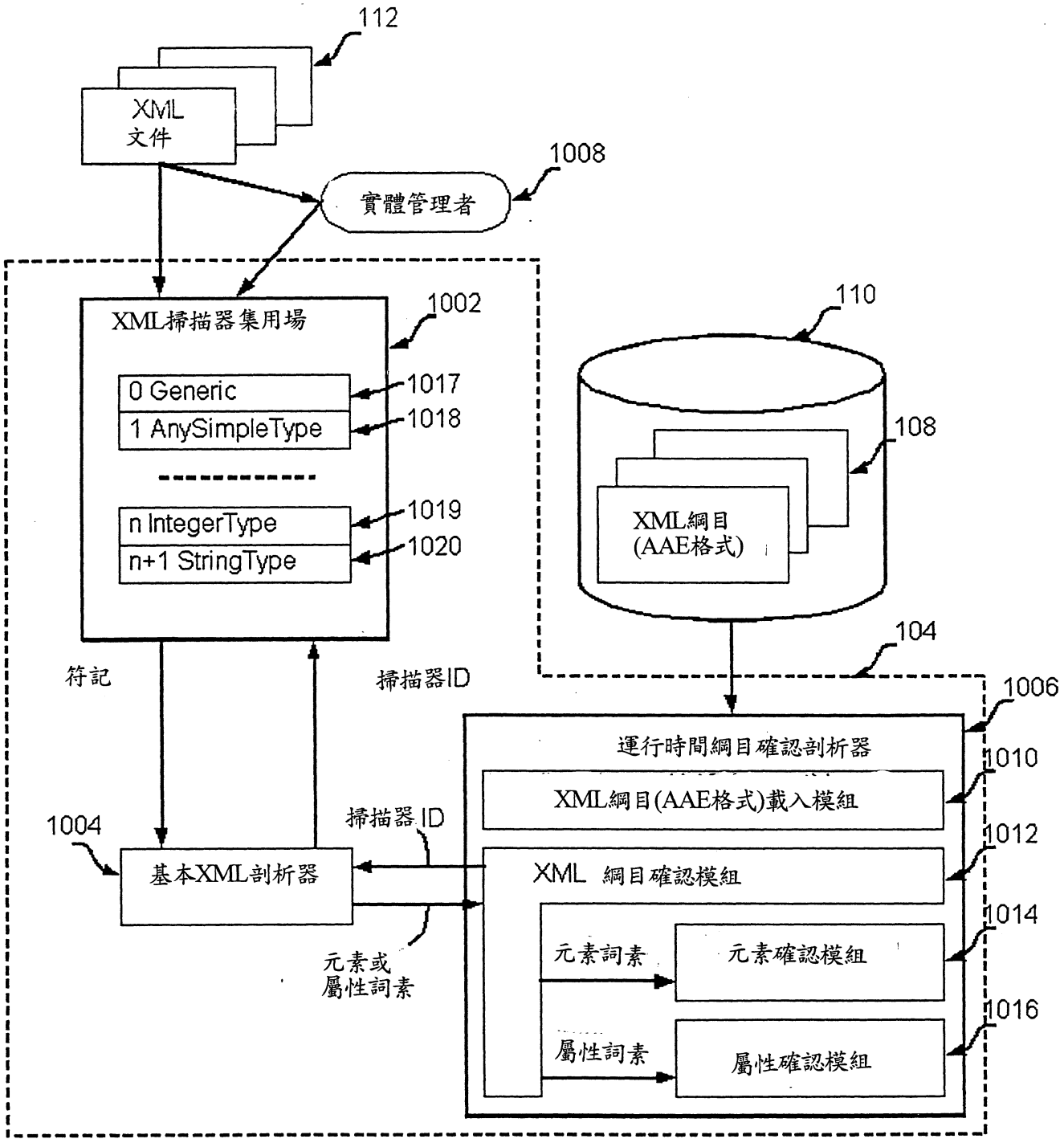

112
XML
文件
1008
實體管理者
XML掃描器集用場
1002
0 Generic
1 AnySimpleType
1017
1018
n IntegerType
n+1 StringType
1019
1020
110
108
XML綱目
(AAE格式)
104
符記
掃描器ID
1006
運行時間綱目確認剖析器
1010
XML綱目(AAE格式)載入模組
1004
掃描器ID
基本XML剖析器
1012
XML 綱目確認模組
1014
元素詞素
元素確認模組
元素或
屬性詞素
1016
屬性詞素
屬性確認模組

圖10

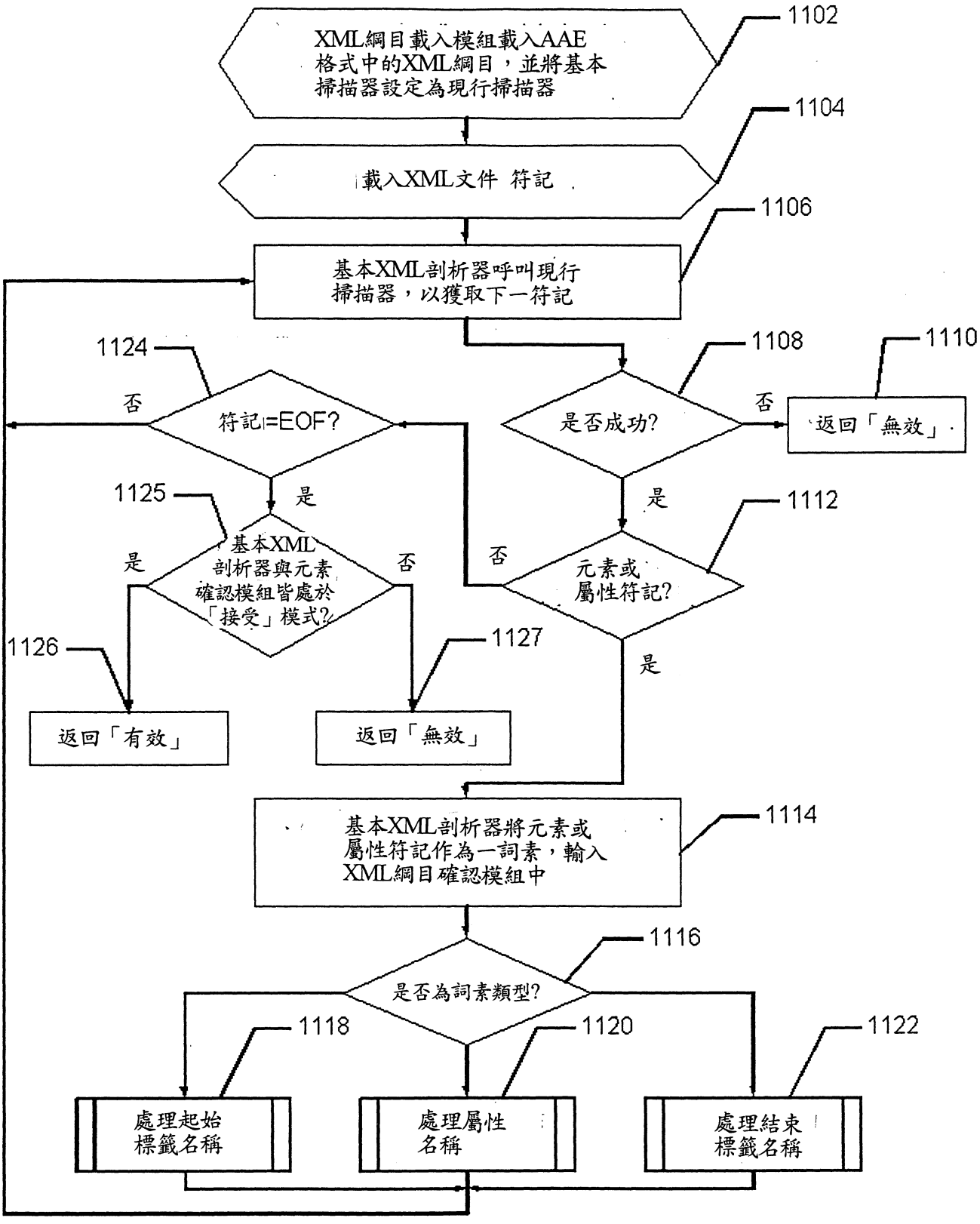

XML綱目載入模組載入AAE格式中的XML綱目，並將基本掃描器設定為現行掃描器
1102
載入XML文件 符記
1104
基本XML剖析器呼叫現行掃描器，以獲取下一符記
1106
符記=EOF?
1124
否
是
是否成功?
1108
否
返回「無效」
1110
是
基本XML剖析器與元素確認模組皆處於「接受」模式?
1125
是
否
元素或屬性符記?
1112
否
是
返回「有效」
1126
返回「無效」
1127
基本XML剖析器將元素或屬性符記作為一詞素，輸入XML綱目確認模組中
1114
是否為詞素類型?
1116
處理起始標籤名稱
1118
處理屬性名稱
1120
處理結束標籤名稱
1122

圖11

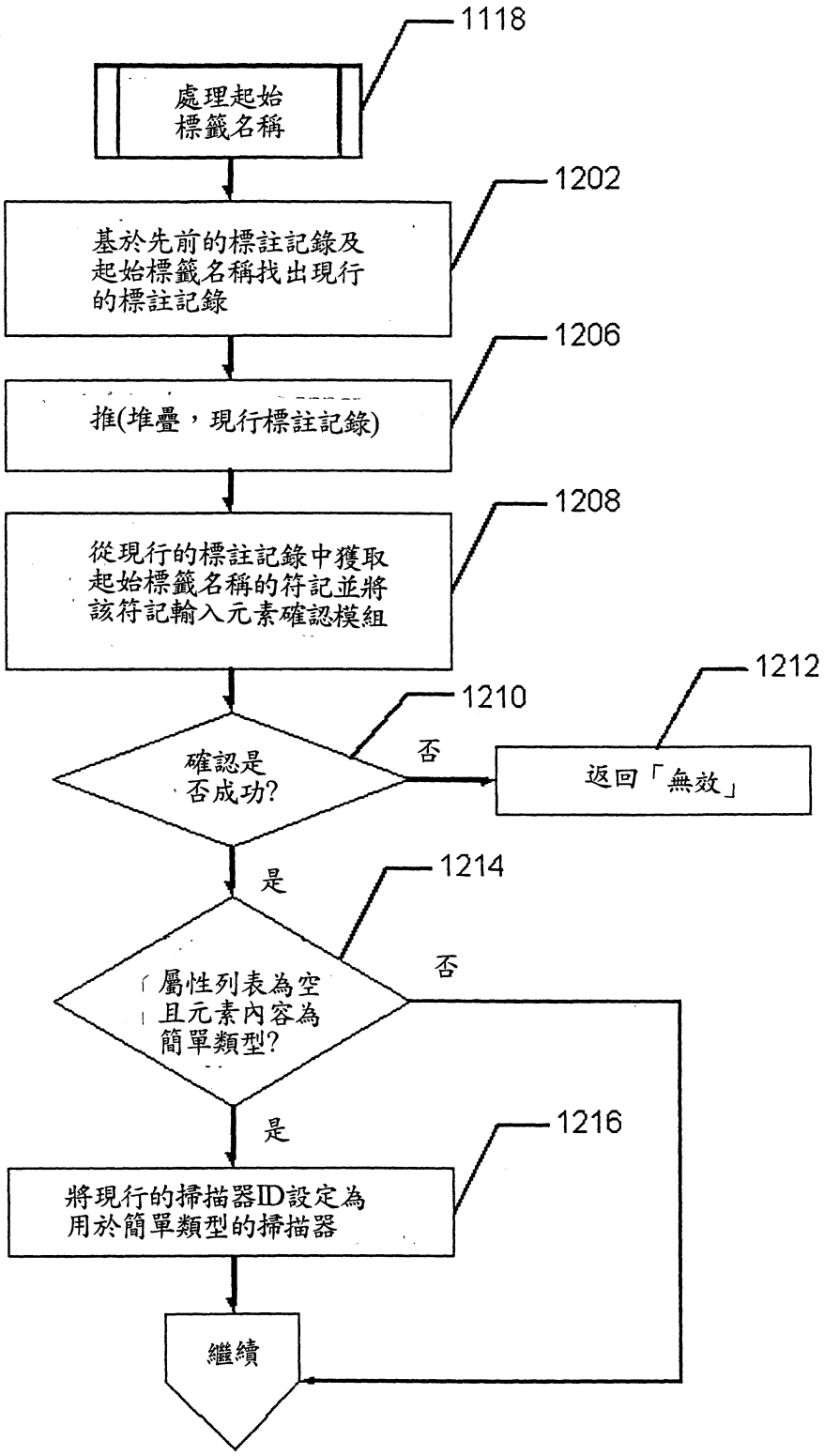

1118
處理起始標籤名稱
1202
基於先前的標註記錄及起始標籤名稱找出現行的標註記錄
1206
推(堆疊，現行標註記錄)
1208
從現行的標註記錄中獲取起始標籤名稱的符記並將該符記輸入元素確認模組
1210
確認是否成功?
否
1212
返回「無效」
是
1214
屬性列表為空且元素內容為簡單類型?
否
是
1216
將現行的掃描器ID設定為用於簡單類型的掃描器
繼續

圖12

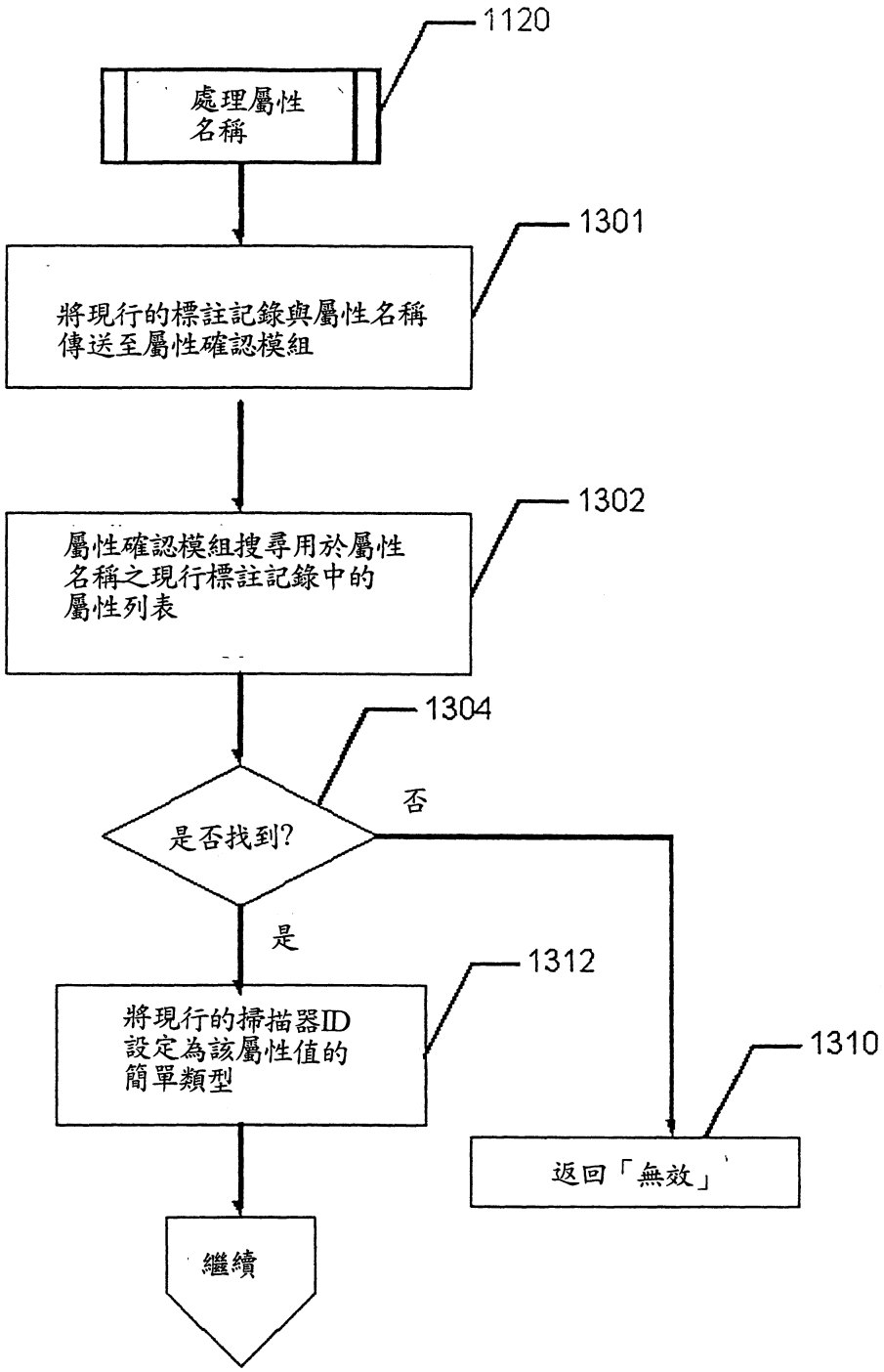

1120
處理屬性名稱
1301
將現行的標註記錄與屬性名稱傳送至屬性確認模組
1302
屬性確認模組搜尋用於屬性名稱之現行標註記錄中的屬性列表
1304
是否找到?
否
是
1312
將現行的掃描器ID設定為該屬性值的簡單類型
1310
返回「無效」
繼續

圖13

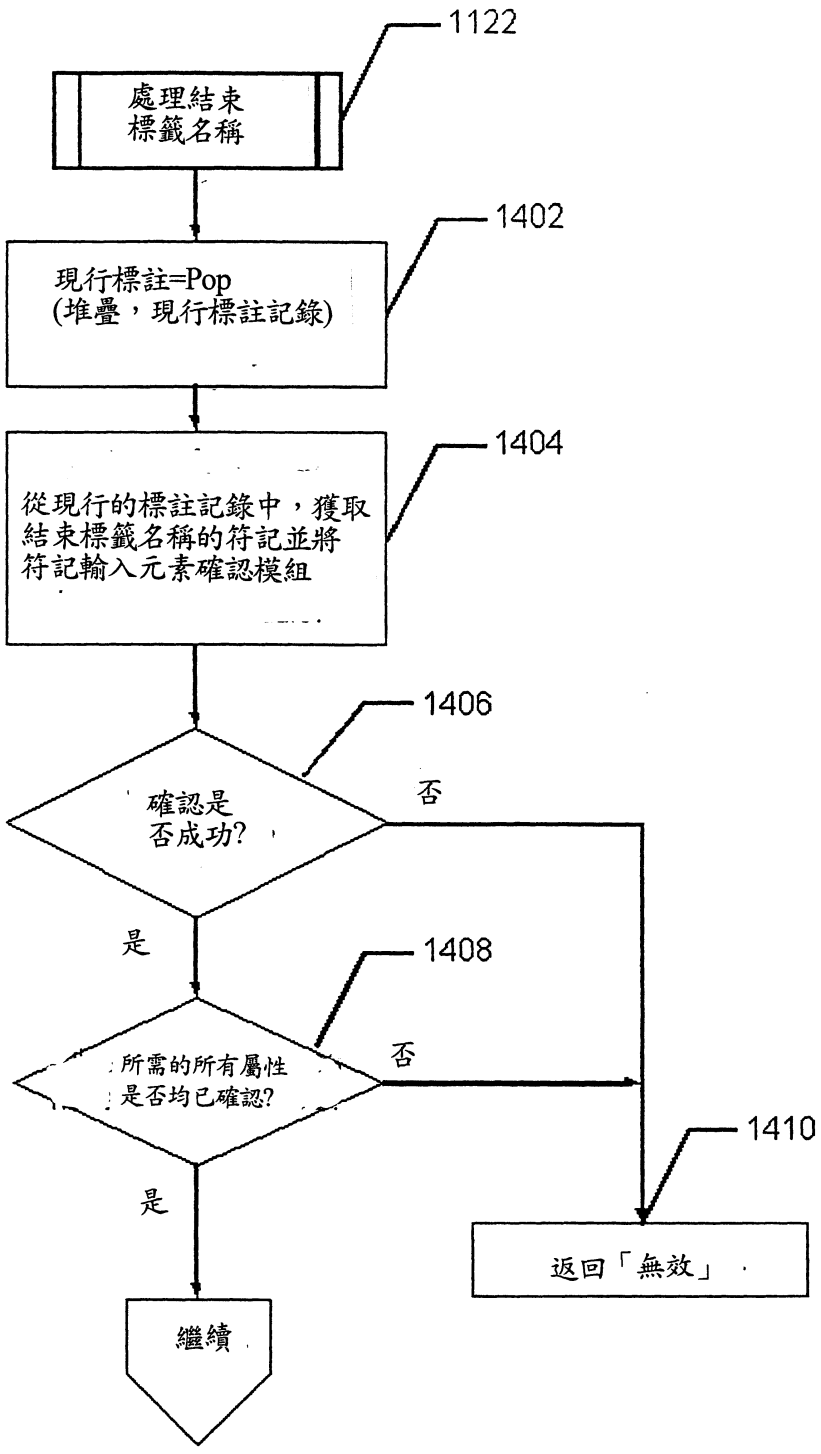

1122
處理結束
標籤名稱
1402
現行標註=Pop
(堆疊，現行標註記錄)
1404
從現行的標註記錄中，獲取
結束標籤名稱的符記並將
符記輸入元素確認模組
1406
確認是
否成功?
否
是
1408
所需的所有屬性
是否均已確認?
否
是
1410
返回「無效」
繼續

圖14

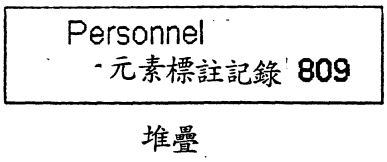


圖15A

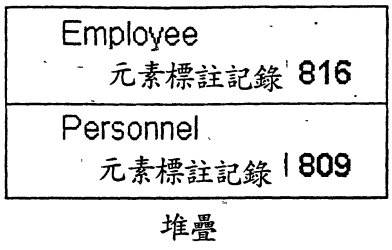


圖15B

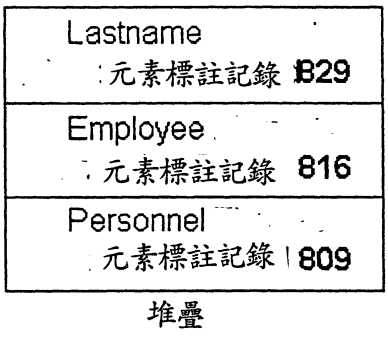


圖15C

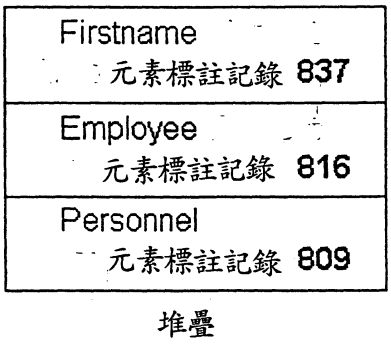


圖15D

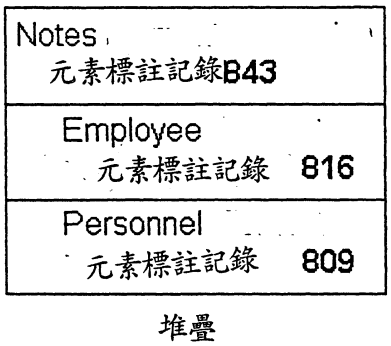


圖15E

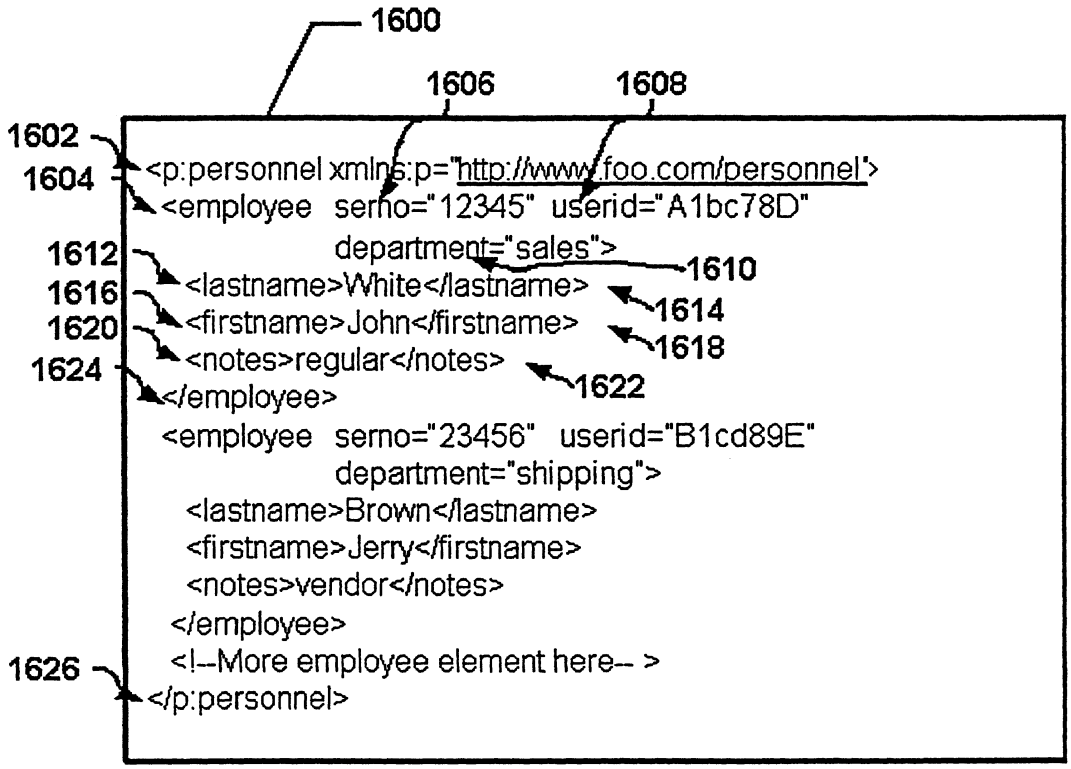

1600
1606
1608
1602
1604
<p:personnel xmlns:p="http://www.foo.com/personnel">
<employee serno="12345" userid="A1bc78D"
department="sales">
1612
1610
1616
<lastname>White</lastname>
1614
1620
<firstname>John</firstname>
1618
1624
<notes>regular</notes>
1622
</employee>
<employee serno="23456" userid="B1cd89E"
department="shipping">
<lastname>Brown</lastname>
<firstname>Jerry</firstname>
<notes>vendor</notes>
</employee>
<!--More employee element here-- >
1626
</p:personnel>

圖16

```
<complexType name="employeeType">
  <sequence>
        <element name='lastname'  type='string'  maxOccurs='1'/>
        <element name='firstname'  type='string'  maxOccurs='1'/>
        <element ref='notes'  minOccurs='0'/>
  </sequence>
</complexType>

<complexType name="regularEmployee">
  <complexContent>
        <extension base="p:employeeType">
          <sequence>
                <element name="salary"  type="decimal"/>
          </sequence>
        </extension>
  </complexContent>
</complexType>

<complexType name="tempEmployee">
  <complexContent>
        <extension base="p:employeeType">
          <sequence>
                <element name="wage"  type="decimal"/>
          <sequence>
        </extension>
  </complexContent>
</complexType>

<element name='employee'  type="p:employeeType"
        minOccurs='1'  maxOcurs='unbounded'/>
```

圖17

柒、指定代表圖：

(一)本案指定代表圖為：第（ 2 ）圖。

(二)本代表圖之元件代表符號簡單說明：

102　XML網目編譯

106　XML網目定義

108　標註自動機編碼(AAE)格式

110　資料庫

202　XML網目編譯器前端

204　XML網目編譯器後端

捌、本案若有化學式時，請揭示最能顯示發明特徵的化學式：

（無）