

運用遺傳和社會關係建構用以協助分析

社區醫療資訊之家系圖系統

俞旭昇 吳偉信 張哲嘉 國立暨南國際大學資訊管理學系

摘 要

為提昇國人健康品質，減少醫療資源的浪費，中央健保局近年來大力推行以家庭醫師制度為中心的社區醫療工作。家庭醫師負責照護家庭中所有成員的健康，其任務包括家庭評估、家族病史記錄與遺傳疾病推論，都需要使用家系圖（Genogram）。家系圖以圖形與線條來表現家族成員的相關資訊，便利醫生以視覺化的方式來分析家庭的健康狀況。然而現今家系圖的使用仍停留在紙筆或電腦圖繪，會產生圖形重複繪製，資訊不同步之問題，亦無法與家族成員在醫院留下的醫療資訊整合，難以支援現今複雜的社區醫療工作。本研究提出遺傳、婚姻與人際社會關係的分類架構，並且設計了成員擷取演算法、層級指定演算法與版面配置演算法，以案主為中心，動態繪出家系圖；最後利用XML語法定義查詢標籤去取得家系圖成員在社區醫療環境中所留下的相關資訊，且可利用這些資訊以不同顏色標記成員圖像。本研究建立的家系圖系統可以避免家系圖重複繪製，累積大量的家庭健康資訊，並可與醫院資訊系統相結合，相信可充分協助醫師和社工人員分析社區醫療資訊。

關鍵詞：家系圖，genogram，社區醫療

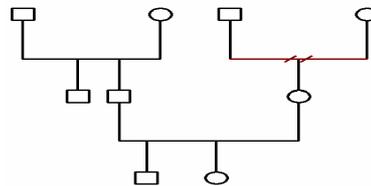
1. 導論

社區醫療是一門確認和解決有關社區群眾健康照顧問題的醫療學科[1]，是基層醫學極重要的一環。全民健保開辦後，逐漸形成以大型醫院為主體的醫療體系，民眾跨區就醫的情形日益嚴重，基層醫療失去平衡，大醫院因為病人過多而導致醫療品質下降，中小型醫院面臨病人流失可能倒閉的窘境。為了導正此現象，國內醫學界普遍希望加強推廣社區醫療，建立完善的社區醫療照護網，對在地民眾推廣衛教、疾病預防宣導等措施，使民眾有病先找家庭醫生並習慣向自己所在地的院所就醫。長期下來，醫生在臨床診斷時就可從片面的疾病觀點擴展到病人完整的病史，進一步分析家庭因素，並納入整體社區的觀察，達到全人健康為導向之醫療保健[2]。

一般而言，社區醫療的業務包括了家庭醫學門診、優生保健諮詢、社區活動支援、健康檢查、山地巡迴診療、衛生健康教育宣導、預防接種等非專科所能完全處理的事務，另外像是無法完全康復，需長期照料的慢性病照護也多納入其中。近年來健保局推動許多跟社區醫療相關的新政策，如89年7月1日起試行「呼吸器依賴患者整合性照護系統」試辦計

畫、91年9月1日起代辦行政院衛生署國民健康局「門診戒菸治療試辦計畫」之代收代付作業。92年3月1日起推動家庭醫師整合性照護制度試辦計畫更與社區醫療密不可分，計畫中提到「家庭醫師制度的順利推動必須依賴社區照護網路的建立，亦即經由“疾病管理”、“個人個案管理”到最終的“家庭照護管理”三階段的推展，以“共同照護門診”為基礎，主動關心特定病人的健康，並逐步在社區中擴大照顧的範圍，建立照護網絡系統[3]。

在社區醫療活動中，家庭評估工作主要收集家庭結構、家庭生活週期、家庭功能、家庭資源及家庭的生活壓力事件等紀錄，以作為日後照護患者的重要參考。其中在記錄家庭結構時，家庭醫生或社工師最常使用的繪圖工具是家系圖（Genogram）（如圖一），用以顯示成員間親屬關係及互動關係、家族病史、遺傳疾病等等，能使照護人員迅速獲知患者家庭的許多資訊。有些家庭醫學科的初診病歷，也都會留下空白區塊供家庭醫生繪製家系圖[4]。



圖一、家系圖範例

現今家系圖使用情況是在一張表單或病歷上，把患者的家庭結構用紙筆描繪而成，然後放入檔案櫃中歸檔，此法對要調閱或欲修，都非常不便。縱然在市面已發展出一些專門繪製家系圖的軟體，如GenoPro、Genoware、Smart-Draw等[5-7]，但此類軟體都還是將家系圖以檔案方式儲存，不具備集中管理功能，更無法和醫院中醫療資訊系統(Healthcare Information Systems, HIS)，如電子病歷、電子表單等進行整合，因此其效益有限。

針對上述缺點，本研究提出一個家系圖系統將遺傳、婚姻與社會關係儲存於資料庫中，由這些關係動態繪製家系圖，以達成下列目的：

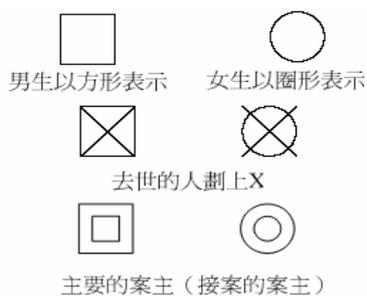
1. 以人為主：本研究提出以案主為中心，將人與人的關係記錄在資料庫，以動態即時的方式繪製出家系圖全貌，節點即為個人，連結兩個節點的邊為人際關係，改善紙本繪製無法擴展的問題。
2. 取得患者醫療資訊：本研究可結合患者在醫院資訊系統中所留下的各類資訊，如個人電子表單資料[13]，讓使用者在系統執行期間自行設定家系圖中欲觀看的成員資訊，達到全體分析之用。
3. 以色彩標注患病成員：本研究藉著取得成員歷史資訊的功能，得知家系圖中成員有無罹患欲調查的疾病後，可以進一步將罹病的成員標注上色彩。
4. 模組化設計：本系統設計實作將個個環節製作成物件模組，減低彼此的耦合性；系統內部機制如成員擷取、成員呈現或關係線條描繪都以具彈性，可靈活抽換為設計目標。

本文第二節探討敘述家系圖使用方法、介紹市面上現有的家系圖軟體、並提出他們的不足之處。第三節說明開發系統所遇到的難處與解決方法。第四節介紹執行環境、實作程式模組與功能介紹，第五節提出本研究結論。

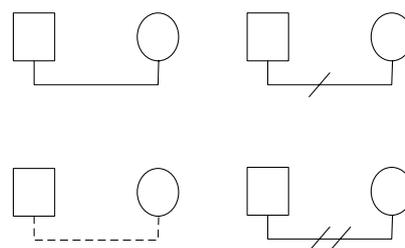
2. 文獻探討

家系圖(Genograms) 是用符號與線條呈現至少三代的家族系統訊息之方法，又稱為家族樹，其親屬關係由圖示可一目了然。利用家系圖可以從各種角度推論疾病與症狀如何隨著血緣關係延伸。家系圖在家庭治療與醫療上廣泛被使用，但並沒有標準的繪製方法。即使是理論背景相當雷同的研究學者們，對於究竟要收集哪些訊息，如何記錄，如何繪製等環節上仍各執一詞。

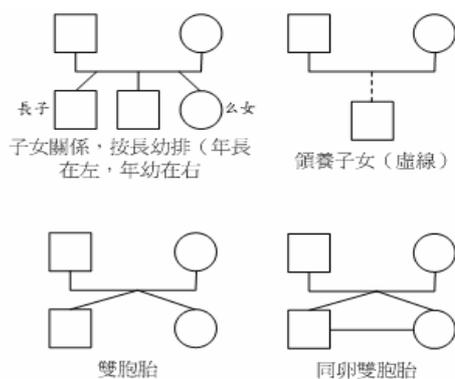
本研究經訪談數間醫院之社工室與參考McGoldrick & Geerson ; Helton & Jackson,1997[8-13]學者的研究後，整理出一般常見的家系圖表達符號與線條如圖二至圖五所示：



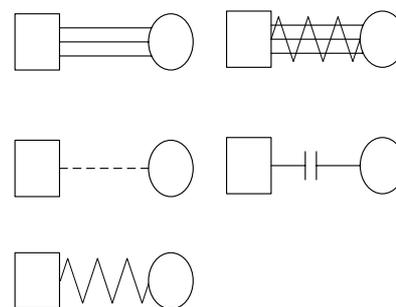
圖二、基本圖形表示



圖三、婚姻關係表示



圖四、家系圖子女表示



圖五、家庭互動模式圖

現今家系圖使用狀況上，有下列幾種處理方式：

1. 紙筆手繪家系圖：某些病歷或是社區醫療業務的表單上會留下一個空白區域用以描繪家系圖，護理人員或社工師以本身對案主的了解在上面繪出家族情況。
2. 家系圖專用繪圖軟體：國外有發展出一些專門繪製家系圖的軟體，比較有代表性的如SmartDraw, WonderWare, GenoPro, Genoware, Geneweaver等等[5-7][15]，在使用上大同小異，皆定義好家系圖專用的圖像符號，以方便繪製。

上述方法與工具皆存在以下這些問題：

1. 無法將成員的關係儲存下來：其儲存單位都以圖為基礎，需由使用者目視判讀，無法提供資訊系統做進一步的分析運用；使用者亦無法自動取得患者的醫療資訊，不易進行分析判斷。
2. 資料重複導致不同步：在不同作業中，可能都需要繪製家系圖，但因使用上述工具，

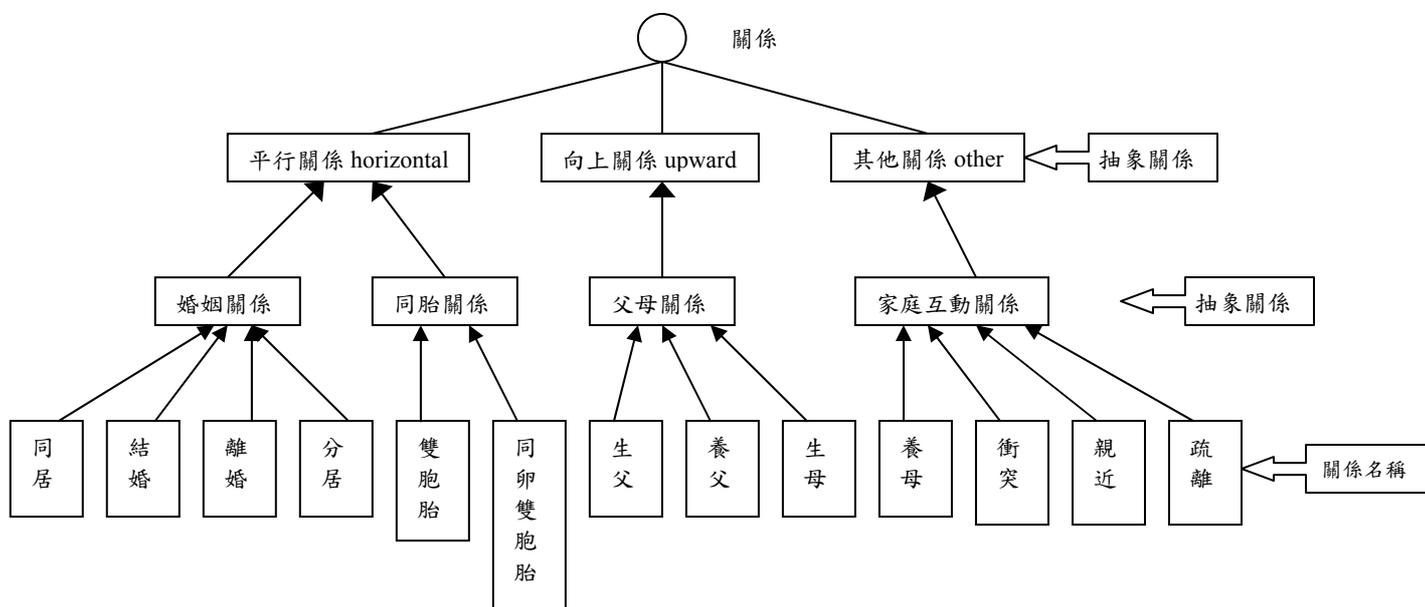
圖與圖之間各自獨立，無法連結整合，同一家庭可能會有數張類似的家系圖。

3. 系統分析與設計

本節以五個觀點進行分析與設計，首先思考遺傳、婚姻與社會關係之分類；其次是分類後的關係資料庫設計；第三為成員擷取機制的設計與分析；第四是圖像呈現排版演算法；最後定義取得患者醫療資訊的方法以進行進階的分析功能。

3.1. 家系圖成員之關係分析

本研究以兩成員彼此間關係為出發點，將關係分類為垂直關係（帶有上下層級之分，如父母與兒女）、平行關係（如夫婦結婚，雙胞胎關係。兄弟手足關係因為同父母生下，不需要刻意記錄。）、與其他關係（關係之間無上下或平行之情形，如人與人的關係）。以上列三種抽象關係概念，我們將實際的人際關係歸類到抽象關係中，如圖六表示：



圖六、關係類別圖

3.2. 關係資料庫設計

基於上述的關係分析，我們設計出兩個資料庫表格來儲存關係定義，如表一與表二：

表一、RelationshipMetadata 資料庫表格欄位

表格名稱：RelationshipMetadata								
欄位名稱	型態	解釋						
RelationshipMetadataId	Int	Primary Key						
RelationshipType	String	平行關係、垂直關係、其他關係。						
RelationshipSubType	String	存放實際的關係分類，如婚姻關係、父母關係...。等等。						
IsTwoWay	Boolean	辨別這種類別的關係是不是雙向。雙向的意思是人與親戚之間的關係（稱謂）是互有意義的，在儲存關係時，要存入兩筆人與親戚對調的資料，如「結婚關係」會存入如下資料 <table border="1" style="margin: 5px 0;"> <tr> <td>Husband</td> <td>Wife</td> <td>結婚</td> </tr> <tr> <td>Wife</td> <td>Husband</td> <td>結婚</td> </tr> </table> 不是雙向關係的例子，如「生父」即是。	Husband	Wife	結婚	Wife	Husband	結婚
Husband	Wife	結婚						
Wife	Husband	結婚						

表二、RelationshipName 資料庫表格欄位

表格名稱：RelationshipName		
欄位名稱	型態	解釋
RelationshipNameId	Int	Primary Key
RelationshipMetadataId	Int	Foreign Key
RelationshipName	String	關係名稱，如結婚、離婚...等等。
RelativeSex	String	三種選項：男生、女生、無。表示這條關係可以用運到什麼性別的親戚，如「生父」只能用在性別為男的親戚。

設計關係定義的資料庫表格後，接著定義儲存人與人關係的資料庫表格（如表三）；最後我們畫出關係儲存之E-R圖（如圖七）。

表三、Relationships 資料庫表格欄位

表格名稱：Relationships		
欄位名稱	型態	解釋
PersonnelId	String	PK, 個人 ID
RelativeId	String	PK, 親戚的 ID
RelationshipNameId	String	PK,FK, 與這位親戚的關係名稱代碼
RelationshipDate	Date	關係建立的日期
RelaitonshipComment	String	關係備註

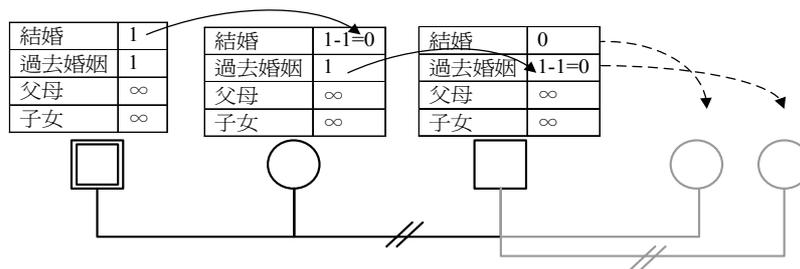


圖七、關係儲存表格ER圖

3.3. 家系圖人員擷取機制的分析與設計

一般家系圖繪製範圍為家族三代成員，而本系統是將個體間關係存入資料庫，以收案主為中心，設定展開的層次，使用遞迴（Recursion）方式，根據人際關係，一層層把人員從資料庫擷取出來。以一個人為中心展開的話，其展開關係包括五項：家系圖最大展開關係、婚姻關係、過去婚姻關係、父母關係與子女關係。這樣的設計，可以在關係上有層級控制，可避免不必要的資訊。

假如使用者要求收案主A以家系圖最大層次為3層；婚姻關係為1層；過去婚姻可以向外長1層；父母與子女關係皆為無限等5個參數展開。其結果如圖八為示：



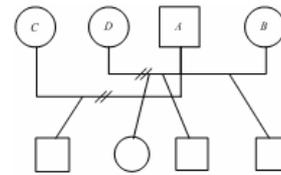
圖八、關係層次展開示意圖

3.4. 家系圖圖像呈現排版的分析與設計

3.4.1. 家系圖多重結婚呈現問題的分析與設計

家系圖規格中有明確定義夫妻關係的畫法，但對於結婚離婚多次這種複雜的婚姻關係群體並無規範，圖像要怎麼排列、線條要怎麼繞徑以提高可讀性，就由實際繪圖的工作者自由發揮[5-7]。本研究採取如圖九畫法。

此畫法採取男左女右的配偶安排，過去的配偶則放到另外一邊；下垂線條必須不重疊，以清楚表示親子關係。演算法分兩部分：1. 複雜的婚姻群主要依男左女右，前任配偶放另一邊的方法排序好。2. 把線條以不重疊的方法描繪出來。第一部分的 pseudo code 如下：



圖九、多重婚姻畫法

Input: 一組未經排列的婚姻群 Set: personSet 。

Output: 一組排列好的婚姻群 List: personList 。

```

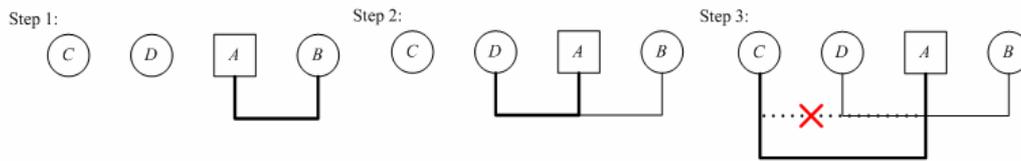
1.
2. Person: personA ← 從 PersonSet 中找出一個婚姻關係次數最多的人
3. personList.add(personA)
4. ArrangePeople(personA)
5.
6. ArrangePeople(Person: person){
7.     IF personSet.Have(person)
8.         personSet.remove(person)
9.     ELSE
10.         return, Function terminate
11.     end IF
12.
13.     List: coupleList ← 找出 person 的配偶們，依照年份，從近到遠排序
14.
15.     index ← personList.indexOf(person)
16.     FOR i←1 to coupleList.size
17.         Person: couple ← coupleList.get(i);
18.         IF personList.notHave(couple)
19.             IF i=1
20.                 IF couple 是女生
21.                     couple.setIndex(index+i)
22.                     personList.add(couple)
23.                 ELSE
24.                     couple.setIndex(index)
25.                     personList.add(couple)
26.                 end IF
27.             ELSE
28.                 IF couple 是女生
29.                     couple.setIndex(index-i)
30.                     personList.add(couple)
31.                 ELSE
32.                     couple.setIndex(index+i)
33.                     personList.add(couple)
34.                 end IF
35.             end IF
36.         end IF
37.     end FOR
38.
39.     coupleList ← coupleList 以婚姻關係次數排序，從多到少
40.     FOR i←1 to coupleList.size
41.         Person: couple ← coupleList.get(i);
42.         ArrangePeople(couple)
43.     end FOR
44.
45. }end ArrangeCouple
    
```

排序好後，接著線條處理，重點在於每一條平行的線皆不重疊。以圖十說明，案主A與妻子B結婚，過去曾跟C與D女士有婚姻關係，其繪製線條步驟如下：

Step 1：以婚姻次數最多的人開始，A有三次婚姻關係，所以由A開始。再來是以婚姻關係時間由近到遠畫線，所以先跟B畫線。

Step 2：A跟D畫線。

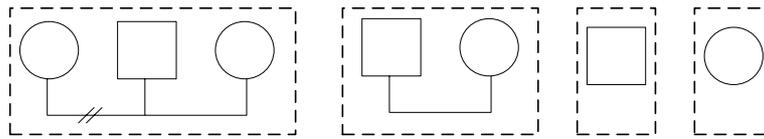
Step 3：AC直接路徑已經被AD路徑佔走，故AC路徑就要往下移一格畫水平線。



圖十、多重婚姻畫線步驟

3.4.2. 家系圖整體圖像版面配置問題

本研究架構以資料庫儲存人際關係來動態繪出家系圖，故在版面呈現時，如何安排成員圖像的位置，是一個非常重要的議題。家系圖成員間關係皆有長輩、平輩、晚輩之分，整張圖含有層級概念，長輩應該排在最上層的圖像，如同根節點。以此觀之，家系圖近似樹狀結構中的有根樹，但子節點都可能有一個父親節點與一個母親節點，因此家系圖不算是正統的樹狀結構，而是 tree-like graph(樹狀圖)。故要將樹狀家系圖動態繪製之演算法，必須考慮包括：樹狀的層級架構、樹狀結構外觀維持平衡、點與點不重疊與婚姻群應該要聚在一起。在演算法節點定義中，考量婚姻群應該要聚在一起以便於判讀，我們提出婚姻群與所有具有圖六定義的婚姻關係的節點皆視為一個聚合節點(Concrete Node)概念，圖十一以虛線方框代表一個聚合節點。



圖十一、聚合節點圖例

基於上述考量，我們設計家系圖圖像層級演算法以達成父母應落於子女之上的視覺效果。而後再設計動態繪製演算法，將設定好層級的圖像節點，動態繪製出來。

3.4.3. 層級指定演算法

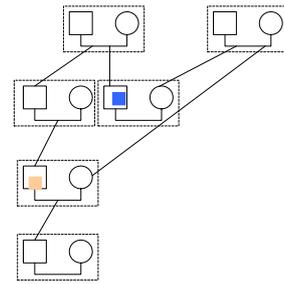
我們提出的層級指定演算法基本理念如下：先將所有節點的層級設定為-1，然後由那些沒有父節點的節點開始，設定他們層級為 0，然後層級+1 往下遞迴指定給子節點，如果遇到子節點的層級小於欲指定的層級，就指定層級並往下繼續遞迴。底下是層級演算法的 pseudo code：

Input: 一組還沒指定層級的 ConcreteNode

Output: 一組指定指定好層級的 ConcreteNode

1. assign -1 to the level of all ConcreteNodes
- 2.
3. visit all the ConcreteNodes without father {
4. setLevel(no father ConcreteNode, 0, Emptyvisited List)
5. }
- 6.
7. setLevel(ConcreteNode, level, visitedList){
8. add ConcreteNode to visitedList
9. if(level > ConcreteNode.level){
10. ConcreteNode.level ← level
11. visit all the children of ConcreteNode not in visitList{
12. setLevel(child ConcreteNode, level+1, visitedList)
13. }
14. }
15. Remove concreteNode from visitedList
16. }

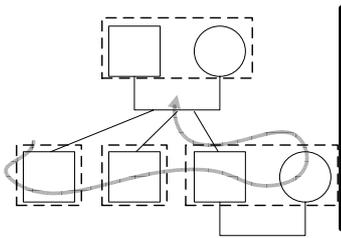
經過我們層級指定演算法的安排之後（如圖十二），叔姪同娶一對姊妹的特殊情況，也能夠妥善的依照父母在子女之上的規則來指定。



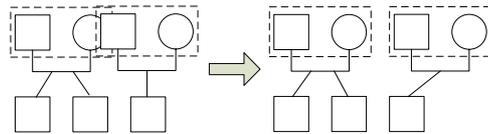
圖十二、叔姪同娶姊妹的層級指定圖

3.4.4. 動態繪製演算法

在動態繪製演算法基本理念為：先記錄整張圖最右邊的 X 座標，命名為 MaxXCoordinate，然後以“先深後廣（DFS）”，“後序（Postorder）”拜訪法來把整棵樹，用遞迴方式往下生長，到末端無子女之葉節點圖像時，葉節點便依照 MaxXCoordinate 所表示的 X 座標為位置與自己所在層級的 Y 座標把自己動態繪製出來，繪製完之後再把 MaxXCoordinate 加上本身節點寬度，最後再把本身 X 座標資訊回傳給父節點。等所有子節點都拜訪完後，表示父節點已收集完所有子節點圖像的 X 座標資訊，父節點以這些子節點位置資訊算出自己應該擺放的中間點位置把自己繪製出來（如圖十三）。



圖十三、節點拜訪圖



圖十四、圖像重疊解決示意圖

另一個重點為每一個階層亦須紀錄各自最右邊 X 座標，因為有些情形為子節點寬度小於父節點寬度，那麼父節點可能就會跟他的同儕相疊，所以還是要紀錄每階層的最右 X 座標，以解決相疊的問題（如圖十四）。底下是 Layout 演算法的 pseudo code：

Input: 一組根節點的 ConcreteNode 串列: topConcreteNodeList 。

Goal: 把所有圖像的位置安排好，繪製出來

```

1. double: maxXCoordinate
2. double[]: levelMaxXCoordinate
3.
4. FOR i ← 1 to topConcreteNodeList.size
5.     ConcreteNode: topConcreteNode ← topConcreteNodeList.get(i)
6.     Layout(concreteNode)
7. end FOR
8. IF 整體版面太過向左偏移而有圖像被遮蔽
9.     整體向右移動到沒有圖像被遮蔽
10. end IF
11.
12. double[] Layout(ConcreteNode: concreteNode)
13.
14.     IF concreteNode 已經畫過
15.         return null
16.     end IF
17.
18.     IF concreteNode 是葉節點，沒有小孩了
19.         double: startX ← maxXCoordinate
    
```

```

20.         double: startY ← concreteNode.level
21.         concreteNode.draw(startX, startY)
22.         maxXCoordinate ← maxXCoordinate+concreteNode.width
23.         levelMaxXCoordinate[level] ← maxXCoordinate
24.         return double[] {startX, maxXCoordinate}
25.     end IF
26.
27.     double: leftX ← double.max
28.     double: rightX ← double.min
29.     FOR i←1 to concreteNode.children.size
30.         ConcreteNode: childNode ← concreteNode.children.get(i)
31.         double[]: childRecord ← Layout(childNode)
32.         IF leftX >= childRecord[0]
33.             leftX ← childRecord[0]
34.         end IF
35.         IF rightX <= childRecord[1]
36.             rightX ← childRecord[1]
37.         end IF
38.     end FOR
39.
40.     IF leftX != double.max AND rightX != double.min
41.         double: centerX ← (rightX-leftX)/2
42.         concreteNode.drawCenter(centerX)
43.     ELSE
44.         concreteNode.draw(maxXCoordinate)
45.     end IF
46.
47.     IF concreteNode 跟 levelMaxXCoordinate[concreteNode.level] 重疊
48.         concreteNode 向右移動到不與 level_maxXCoordinate 重疊
49.     end IF
50.
51.     IF concreteNode.getRightX > levelMaxXCoordinate[concreteNode.level]
52.         levelMaxXCoordinate[concreteNode.level] ← concreteNode.getRightX
53.     end IF
54.
55.     return double: {concreteNode.getLeftX, concreteNode.getRightX}
56.
57. end Layout

```

3.5. 取得患者醫療資訊之分析與設計

根據患者在醫院留下的醫療資訊，進行長期追蹤、病史分析，以達到疾病預防全人照護一直都是醫療產業所要追求的目標。在我們的家系圖系統中，除了取得成員個人基本資料外，還要考量如何取得成員的醫療資訊。我們定義查詢方法（查詢標籤）並考量以下要點：

1. 查詢標籤定義的紀錄方法：我們使用 XML 標籤記錄查詢標籤的定義，一個個標籤組合起來就變為一張可交換的 XML 文件，如果往後系統有離線版本，就可以不用受限於資料庫連結的問題。
2. 屬性內容的資料型態：文字、數字、布林型態為一個執行命令，以啟動某個程式。
3. 內容取得的來源：由結構化電子表單系統之參數值[14]可以查詢人員是否參加慢性病專案來得知有無罹患疾病、查詢人員在某張電子表單的某個欄位所填資料、由某一個程式計算得出的值。

故查詢標籤我們定義了以下如表四所示屬性：

表四、查詢標籤屬性定義表

名稱	解釋												
Id	編號。												
Name	名稱。												
DataType	資料型態，現有兩種選項 Boolean ：去取得 boolean 值，boolean 值可以應用在標注患病成員，boolean 值為 true 的成員，其圖像就變色。所以如果這個額外屬性標籤是在查詢是否有罹患疾病，就可以用視覺快速得知家族中患病的成員。 String ：去取得文數字值。												
DataSource	資料來源的類型，現有四種選項，跟 DataType 屬性 各有其對應。 Form ：從表單欄位中淬取資料，DataType 屬性可為 Boolean 或 String。 Method ：使用某個 Class 中的 Method，DataType 屬性可為 Boolean 或 String。 Project ：查詢人員有無參加專案，DataType 屬性應該只有 Boolean 對應。 Command ：在家系圖程式的介面出現一個程式執行啟動點，DataType 屬性應該留白。												
Parameters	詳細的參數設定，對應各種 DataSource，底下列出其對應的內容： <table border="1" data-bbox="418 833 1353 1048"> <thead> <tr> <th>DataSource</th> <th>Parameters</th> <th>解釋</th> </tr> </thead> <tbody> <tr> <td>Project</td> <td>ProjectId:專案 Id</td> <td>查詢成員有無參加專案。</td> </tr> <tr> <td>Form</td> <td>FormId:表 Id， NodeId:節點 Id</td> <td>查詢某張表單的某節點數值。</td> </tr> <tr> <td>Method</td> <td>ClassName:類別名稱， MethodName:方法名稱</td> <td>使用某個類別的某個 static method 得到數值。</td> </tr> </tbody> </table>	DataSource	Parameters	解釋	Project	ProjectId:專案 Id	查詢成員有無參加專案。	Form	FormId:表 Id， NodeId:節點 Id	查詢某張表單的某節點數值。	Method	ClassName:類別名稱， MethodName:方法名稱	使用某個類別的某個 static method 得到數值。
DataSource	Parameters	解釋											
Project	ProjectId:專案 Id	查詢成員有無參加專案。											
Form	FormId:表 Id， NodeId:節點 Id	查詢某張表單的某節點數值。											
Method	ClassName:類別名稱， MethodName:方法名稱	使用某個類別的某個 static method 得到數值。											
Comment	解釋與備註												
Keywords	這個查詢標籤的關鍵字，可在搜尋時使用												

下面是查詢標籤的例子，這個額外屬性的意思就是要去查詢成員有無參加糖尿病專案來判斷他是不是糖尿病病患。

```
<Query Name="糖尿病" Id="1" DataType="Boolean" DataSource="Project">
  <Parameters>
    <ProjectId>1</ProjectId>
  </Parameters>
  <Comment>...</Comment>
  <Keywords>diabetes</Keywords>
</Query>
```

另一例為利用 static method 去查詢的標籤，其目的是要去得知成員是否過重。

```
<Query Name="體重是否過重" Id="13" DataType="boolean" DataSource="Method">
  <Parameters>
    <ClassName>package.tool.Measure</ClassName>
    <MethodName>tooFat</MethodName>
  </Parameters>
  <Comment>...</Comment>
  <Keywords>diabetes</Keywords>
</Query>
```

在社區醫療的環境下，醫護工作者的角色隨著業務內容而有所不同，如糖尿病收案醫生、氣喘收案醫生、病友團體管理師等等，其可觀看的資訊也各有不同，所以我們在查詢標籤也必須要加上角色權限的功能。

考量到未來屬性可能累積越來越多，使用者並不需要一次就把所有屬於他的查詢標籤載入，或期望一次只看某一種類群的查詢標籤，如只想看跟糖尿病相關的查詢標籤，其他查詢標籤就不需要出現以避免造成干擾。因此我們導入 View 的觀念，View 是一份由查詢標籤所組成的 XML 文件，角色與使用者可以擁有多張 View。家系圖執行時，就接收一張 View 文件決定可以看到什麼查詢標籤，使用者在操作過程中可以自行切換或是編輯自己擁有 View，來決定自己想要觀看的查詢標籤。使用 View 文件的另一個好處是此架構使用亦可用於離線版本，以方便醫師位於無網際網路之地區使用本系統。

運用 XML 對層級化文件的支援性，View 的內容還可以將查詢標籤分群，讓使用者增加便利性。底下為 View 範例：

```

<View>
  <QueryGroup Name="糖尿病相關">
    <Query Name="糖尿病" Id="1" DataType="boolean" DataSource="Project">
      <Parameters>
        <ProjectId>1</ProjectId>
      </Parameters>
      <Comment></Comment>
    </Query>
    <Query Name="血糖值" Id="2" DataType="string" DataSource="Form">
      <Parameters>
        <FormName>1</FormName>
        <NodeId>5</NodeId>
      </Parameters>
      <Comment></Comment>
    </Query>
  </QueryGroup>
  <Query Name="體重" Id="10" DataType="string" DataSource="Form">
    <Parameters>
      <FormName>10</FormName>
      <NodeId>3</NodeId>
    </Parameters>
    <Comment></Comment>
  </Query>
</View>

```

假如使用者在某個查詢標籤的觀看權力已經喪失，可是 View 文件上還有這個查詢標籤時，可能會讓使用者看到他不該看的資訊，因此家系圖在使用一張 View 時，還是會根據現在的使用者向資料庫查詢查詢標籤與角色是否匹配，再決定是否進行查詢的動作。

4. 系統實作

本節主要說明執行環境，系統模組的分割與實作功能介紹等系統運作核心架構；對於詳細的資料流程與系統畫面則不多贅述[16]。

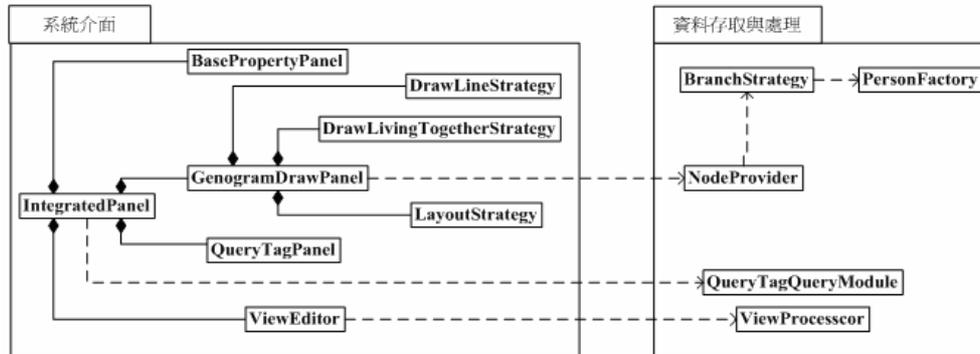
4.1. 系統執行環境

系統執行點為結構化表單管理系統[14]上一張表單的按鈕，執行後便出現以這張表單的患者為中心的家系圖。故在表單系統那主要傳遞下列參數給本系統：「患者 ID」、「展開層次」、「使用者角色」、「依照角色與應用所給予查詢標籤 View 文件」。

4.2. 系統模組

系統設計實作將系統介面與資料處理部分切割為資料存取與處理套件與系統介面套

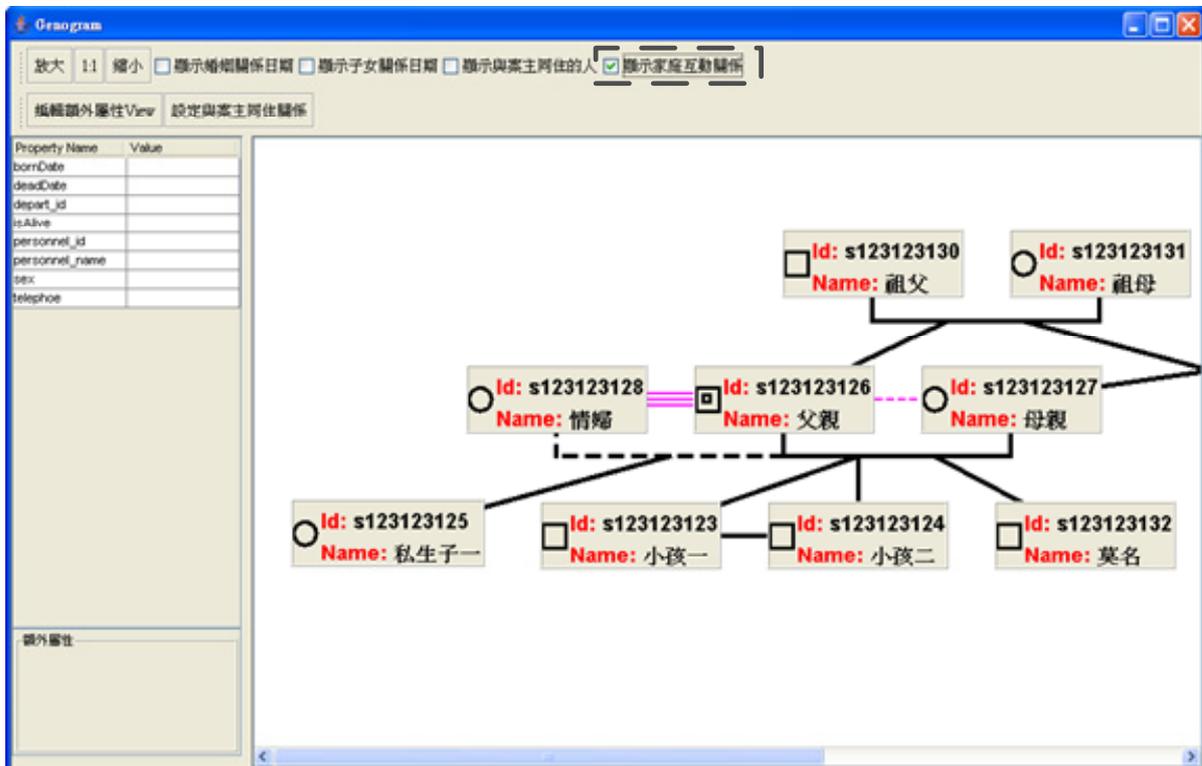
件，減低彼此的耦合性；系統內部機制如患者擷取、患者呈現或關係線條描繪皆具彈性，可靈活抽換。如對圖像 Layout 的方式更換，可再設計一個圖像 Layout 的演算法替換上去，整個繪圖版面毋需再更改（如圖十五）。



圖十五、系統套件圖

4.3. 系統實作功能

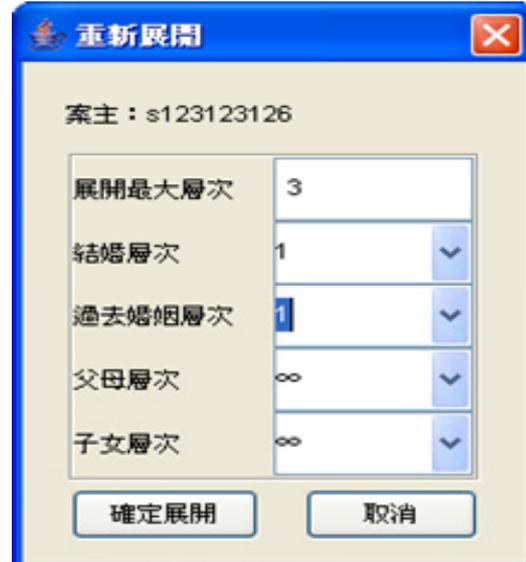
在系統主要呈現的畫面，由許多各司其職的模組包含組合而成。對於家系圖可進行圖像拖曳；關係表示可由連結線上顯示，如婚姻結婚、兒女生日關係日期顯示；家庭互動關係線條亦可在勾選後顯示（圖十六）。在成員關係設定上，可由資料庫判斷是否有此關係存在，若無則可在系統中直接新增（圖十七）。在查詢上，使用者可自行定義 View 來進行進階化查詢，也可將相關疾病之家族成員給予著色，最後使用者亦可選擇以個人為中心，展開關係層級數量（圖十八）。



圖十六、顯示家庭互動關係



圖十七、成員關係設定對話視窗



圖十八、重新展開對話視窗

5. 結論

在本研究中，我們從家系圖的觀點探討親屬關係的儲存方式，定義取得患者病史的方法，實作出以親屬關係動態產生的家系圖分析系統，本研究的成果如下：

1. 解決家系圖重複繪製問題：在過去紙本的情形，對同一家庭同一案主的處理多是一份表單就繪製一張家系圖，這樣就會出現家系圖重複繪製，資料內容不整合的問題。此問題在本研究所提出的架構得到解決。
2. 方便調閱家系圖：本系統集中管理親屬關係之家系圖，解決以往紙本版本不一致問題，方便使用者查閱。
3. 儲存親屬關係：本研究所提出之家系圖系統能將成員之間的親屬關係儲存下來，也可實行非視覺化之家族性分析，如遺傳疾病推論。
4. 家系圖結合患者歷史資訊：藉由查詢標籤的定義，可以取得患者歷史資訊，這是傳統作業之家系圖所難達成的目標。家系圖整合患者的歷史病史，讓醫療人員可以全面性的對整個家族進行家族病史之分析，推敲遺傳疾病之可能進而達到預防醫學。
5. 提供未來社區醫療家庭照護的基礎：雖然現今的社區醫療與政府政策還停留在疾病管理與個人專案照護，可是未來的發展目標一定是以血緣親屬為主的家庭全人照護政策。藉由本研究所提出的系統，可以在現階段使用家系圖的過程中，漸漸的將家族脈絡建立起來，替未來的社區照護網絡最終階段家庭照護管理打下基礎。

參考文獻

- [1] 邱泰源，「社區醫學教育與健康照護」，台灣醫學，第5卷，第二期，pp.212-220，民國90年03月。
- [2] 李怡娟，「運用資源整合策略於基層醫療保健」，台灣醫學，第2卷，第三期，pp.348-350，民國87年05月。
- [3] 中央健康保險局網站，<http://www.nhi.gov.tw/>。
- [4] 李孟智，家庭醫學與全民健保醫業管理，合記出版社。

- [5] GenoPro , www.genopro.com.
- [6] GenoWare , www.genogram.com.
- [7] SmartDraw , www.smartdraw.com.
- [8] Monica McGoldrick and Randy G-erson, *Genograms in Family Assessment*,1985.
- [9] Helton, L.R. and Jackson, M, *Social work practice with families. A diversity model*. Boston: Allyn and Bacon,1997
- [10]吳涯,「家族圖在輔導工作上的應用」,諮商與輔導,第九十八期,pp23-27。
- [11]劉念肯,「家族圖的基本概念及其繪製方法」,諮商與輔導,第九十八期,pp14-22。
- [12]陳麗英,「家庭功能的評估」,中華心理衛生學刊,第8卷,第三期,pp31-44。
- [13]周月清,家庭社會工作-理論與方法,pp251-252。
- [14]趙咸欣,「應用於社區醫療資訊的結構化表單管理系統」,國立暨南國際大學資訊管理學系碩士論文。
- [15] WonderWare , www.interpersonaluniverse.net.
- [16]吳偉信,「運用遺傳和社會關係建構用以協助分析社區醫療資訊之家系圖系統」,國立暨南國際大學資訊管理學系碩士論文。