

國立暨南國際大學資訊管理研究所

碩士論文

具自訂觸發規則能力之表單資料驅動架構  
—以社區醫療照護為例

Applying A Data Driven Architecture With  
User Defined Rules To Community Health  
Care

指導教授：俞旭昇 博士

研 究 生：李志良

中華民國九十五年六月

## 中文摘要

社區醫療指的是在特殊的社區範圍內，所提供的整合性醫療服務。主要目標是整合各個不同的專業資源，來對病患提供以人為服務，具有跨組織、流程不固定等特性，和一般醫療行為或企業流程的資訊需求有很大的不同。

針對社區醫療的資訊需求，可行的方法，是建立一個可以自訂表單結構，並將表單用 XML 檔案格式儲存的資料庫系統，並利用主動式資料庫的 ECA 規則做流程的整合。

本研究針對社區醫療照護中所使用的表單，提出資料驅動模型，利用社區醫療中產生各式各樣的表單，經由使用者自訂的規則，將表單中個各項資訊視為運算元，由使用者自行搭配 and、or、(、) + - \* / 等運算子，以 Expression 的方式定義成規則，隨著每張表單資料的內容，驅動後續相關的照護流程，來連結社區醫療中所需的照護資源。

關鍵字：社區醫療照護(Community health care)、ECA 規則

## Abstract

Community Health Care (CHC) is an integrated medical service in which delivered to specific community. Compare to general medical services, the need of information system (IS) is quite different due to the essence of CHC's operations such as crossing divisions and unroutine procedures.

Establishing the mechanize of developing user-defined-form based system is the best way to integrate the diversify medical services which occurred in CHC. With the concept of Active Database ECA rules, a structured form based software system with XML technique has been used to achieve this goal.

The purpose of this study is to develop a system that applying Data Driven Architecture with User Defined Rules to CHC. By using all varieties of data, users can define rules, and derive the necessary medical care procedures, by taking all items from the structured form based System into the element of operation, users can arrange the operation unit like "and、or、(、) + - \* /" by themselves, and define the rules by using Expressions. Then by following the content from the data, derive the relevant health care procedures to reinforce the community medical information needs.

Keyword: Community Health Care, ECA Rules

## 誌 謝

職場多年後又回到學校再次進修，首先感謝我的指導教授俞旭昇博士。俞老師在我研究的過程中，不論是在論文研究上或是在各種觀念上，悉心的給予我殷切的指導，能讓我順利的完成論文，在此致上我誠摯的感謝。

口試期間，感謝吳帆老師、蔡孟峰老師、尹邦嚴老師的悉心審閱。提供我寶貴的意見，幫助我匡正論文的不足之處，使得本論文才能更加的完整，在此致上我深忱的感激與謝意。

感謝埔里基督教醫院的黃蔚院長、趙文崇院長、劉建志副院長及同事們，在我進修的期間，給我非常大的鼓勵及支持，感謝這幾年來不斷給我鼓勵，陪我一路走來的實驗室的伙伴及各位同學們：咸欣、偉信、慶穎及學弟妹們，謝謝你們給我的幫助，使我的論文得以順利完成。

最後特別感謝我的家人，媽媽、太太麗雪及彥伯、李宜兩位乖巧的小孩。因為你們的支持，使我在無後顧之憂下，得以順利完成學業，在此深表由衷感激！

李 志 良 謹誌

民國 95 年 1 月於暨大

第一章	緒論	1
1.1.	研究背景與動機	1
1.2.	研究目的	2
1.3.	研究限制	3
1.4.	論文架構	3
第二章	文獻探討	4
2.1.	社區醫療	4
2.2.	主動式資料庫	7
2.3.	現有的電子化表單產品	17
2.3.1.	商用電子表單產品	17
2.3.2.	相關研究	23
第三章	系統架構與設計	25
3.1.	表單結構	25
3.1.1.	表單結構的描述	25
3.1.2.	表單資料的描述	29
3.2.	資料驅動模型	31
3.3.	事件處理單元	33
3.4.	條件規則處理單元	34

3.4.2.	條件的描述 .....	35
3.4.3.	動作的描述 .....	35
3.5.	動作處理單元 .....	36
第四章	系統建置實作與案例探討 .....	38
4.1.	事件處理單元 .....	38
4.1.1.	事件觸發者 .....	38
4.1.2.	事件佇列資料表 .....	38
4.2.	條件規則處理單元 .....	39
4.2.1.	條件規則定義資料表 .....	40
4.3.	動作的描述 .....	42
4.3.1.	動作分派者 .....	42
4.3.2.	動作執行者 .....	43
4.3.3.	動作要求佇列資料表 .....	43
4.4.	硬體架構 .....	45
4.5.	軟體架構 .....	45
第五章	討論 .....	47
第六章	結論與未來研究方向 .....	49
6.1.	結論 .....	49
6.2.	未來研究方向 .....	49

第七章 參考文獻..... 50

表目錄

表 2-1	主動規則的特性表 .....	8
表 2-2	執行模式特性表 .....	12
表 2-3	管理模式特性 .....	15
表 3-4	節點的資料型態 .....	27



## 圖目錄

圖 2-1	流程功能方塊圖 .....	17
圖 3-1	社區醫療中的表單範例 .....	25
圖 3-2	表單的樹狀結構 .....	26
圖 3-3	表單架構 XML 檔案 .....	29
圖 3-4	表單資料 XML 檔案 .....	30
圖 3-5	系統架構 .....	33
圖 3-6	事件處理單元架構 .....	33
圖 3-7	條件處理單元架構 .....	34
圖 3-8	動作處理單元架構 .....	36
圖 4-1	事件佇列資料表 SCHEMA .....	39
圖 4-2	規則定義輸入畫面 .....	40
圖 4-3	條件規則定義資料 SCHEMA .....	41
圖 4-4	動作要求佇列資料 SCHEMA .....	44

# 第一章 緒論

## 1.1. 研究背景與動機

社區醫療[1][2][3]指的是在特殊的社區範圍內，所提供的整合性醫療服務，主要目標是整合各個不同的專業資源，來對病患提供以人為主服務，和一般醫療行為或企業流程的資訊需求有很大的不同：

一、社區醫療不同於一般治療單一疾病的醫療模式，其涵蓋了預防保健、一般診療、精神醫療與長期照護等服務[1]，並且採取以人為主的照護方式。每一種疾病所需的專業、資源、療程、檢查等，都不相同，因此社區醫療照護，不但是以人為主體，而且必須能夠配合不同疾病的照護流程，去整合不同的醫療資源，不同的療程，來達到全人照護的目的。

二、在社區醫療中，同一種疾病的療程，不但因著每個人病情的變化不同，同時也是醫病雙方互動的結果，所以其流程的彈性，遠大於一般企業流程，所需要的控制機制更為複雜。

三、所有療程產生的病歷資料，皆會歸屬到同一個人；而每一個療程，皆有可能需要擷取過去的病歷資料。一般企業的流程資料只需要存在幾年，而社區醫療資訊，必須跟著病患一輩子，所以其資料的量，遠大於一般企業流程。

四、在一般的企業流程中，有著明確的組織階層架構與界線，而社區醫療的「社區」，其本身就沒有明確的範圍，任何一個社工師、或是衛教師都有可能隨時加入照護團隊，甚至同一流程可能由不同的院所共同完成。

針對社區醫療的資訊需求，可行的方法，是建立一個可以自訂表單結構，並將表單用 XML 檔案格式儲存的資料庫系統，並利用主動式資料庫的 ECA 規則做流程的

整合；而表單資料庫系統的建立，已經有陳金榮[4]及趙咸欣[5]等相關的研究，因此本研究的重點，希望在原本已經既有的表單資料庫系統上，建立符合社區醫療資訊需求的驅動架構。

註解 [v1]: 陳金榮<sup>a</sup>, \*溫嘉憲<sup>a,b</sup>, 葉大成<sup>c</sup>, 陳琦文<sup>d</sup> A  
Web-based Platform for  
Implementing Clinical Notes  
2004 年國際醫學資訊研討  
會論文集 17-21

## 1.2. 研究目的

為了提供良好的醫療服務，醫療機構可以透過資訊系統大量而精確的收集及分享資訊[6]，具有跨組織、動態流程、異質資訊、以人為本的醫療活動，更須嚴謹且深具彈性的資訊系統，來實現其願景。

為了能夠因應各種不同流程上的需求，本研究希望提供一個具有下列特性的系統架構，來充分整合電子化表單與作業流程，使得利用表單所蒐集的資訊，能夠有更適切的運用，創造更大的價值。

一、社區醫療的流程是與個案有關，也就是說流程的驅動是與表單的內容相關，因著個案的疾病情況不同，表單內容的資料不同，所相關的流程也會跟著改變，這與一般的企業表單流程有很大的不同。

二、因著沒有組織的界線存在的特性，參與的人員並不限於單一組織，更有可能隨時加入或離開，因此在工作流程中，並不能期望使用者，一定是系統的使用者，所以在與流程結合時，必須讓系統能夠與外界流程溝通，如發送 EMAIL 給不同機構之人員。

三、在流程的定義上，必須允許使用者隨時新增表單，而系統必須去適應隨時新增的表單結構，並且讓使用者可以依據新增的表單結構，隨時定義新的流程。

運用具有以上特性的系統架構，就可以使得大量的照護流程中的表單，不僅只有收集儲存的功能，更能充分應用其中的資訊，驅動後續相關的流程，使照護的品質更加提升，使其照護的病患更能邁向健康。

### 1.3. 研究限制

本研究的架構，有下列的限制存在：

一、假定每一個流程是由單獨一張表單所驅動，因此規則限定為針對單一表單，暫不考慮跨表單的條件支援。

二、為了能夠清楚的定義條件規則，所以任何新增的表單結構必須固定，而不能於執行時期新增資料項目。

### 1.4. 論文架構

本研究首先討論現今社區醫療照護流程的特性，並提出本研究所要達成的目標，與研究的限制。而第二章進行文獻的探討，了解何謂社區醫療，疾病管理，及有何相關的架構可以加以應用。第三章提出系統的架構與設計，說明此架構如何幫助社區醫療團隊的運作，第四章則在實作與建置的過程中，配合實際的情況，解決設計上的問題，並展示開發完成的系統功能及測試其可行性。最後第五章提出結論與未來研究方向。

## 第二章 文獻探討

在本章中首先探討社區醫療的定義，及主動式資料庫的 ECA 規則，並探討市面上的電子表單產品，及國內相關的研究。

### 2.1. 社區醫療

「社區」一詞的定義相當的多，以不同的角度來說明，就有不同的定義，而管敏秀[7]在對於社區健康營造的研究中，曾經為社區下定義，其指出「社區意即一群人居住在一固定的區域內，因為共同的利益、共同的問題、共同的需要，而產生一種生命共同體的社區意識，並且為保障利益、解決問題、應付需要，這群人乃組織起來，互相依賴，互助合作，共同行使社會功能」。

社區醫療其目的是為了建立一個更健康的社區，並提供整合性的醫療服務。陳慶餘[1]指出「台大醫學院以結合臨床醫學與公共衛生的方式融入社區，辦理以社區為基礎的醫學繼續教育及醫療服務，藉此建立一個均衡社區醫療與醫院醫療的整合性健康醫療服務系統，來促進社區健康。而對於促進健康社區的推展策略，則是提供整合性醫療服務，執行健康改善計畫，並推動健康社區營造方案」。

另外奇美醫院的社區醫療部吳重達曾說，「社區醫療的概念主要著重在促進健康及預防疾病上，但最主要的目的是建構一個更健康的社會」。而該院社區醫療部整合公共衛生、家庭醫學及預防醫學科，進行青少年的保健、及小型社區的健康營造。

上述皆說明社區醫療最主要的便是提供整合性的醫療服務，何謂整合性的醫療服務上，可以說是在特殊的社區範圍內，運用整合各項專業資源，來改善促進社區內居民的健康，其涵蓋的範圍包含預防保健、一般診療、精神醫療、與長期照護[1]。

張煥禎[2]等人曾提出健康管理新概念，「其包含的活動必須涵蓋上游到下游的整體性健康照護，不論是一般性的需求管理或特定疾病的健康管理，皆包括了專業人員諮詢、轉介服務網路、專業服務項目與資訊系統的支援服務等」。說明了在整合型醫療服務上，是整合許多不同類型資源的一種服務，而黃源協[3]更指出整個服務體系的成功關鍵，是在於如何建構一套以社區為基礎的長期照護團隊，為需要服務的社區民眾提供無縫隙的服務。其中對於團隊的定義作以下描述，「團隊可界定為滿足或解決多元或複雜的照顧需求或問題所組成的工作團體，其成員係來自不同的專業領域，但彼此透過專業分工、協調與合作方式，致力於共同確認的使命、宗旨或目標。」

邱泰源 [8]認為以社區醫療為基礎的醫療服務，其內容涵蓋社區健康促進計畫，並落實於良好之社區營造基礎上，讓健康意識可以融入社區公共政策中，是一種較理想且長遠的醫療服務方式，因此甄瑞興 [9]等人以某區域醫院的經驗為例，對於一個針對失智老人問題，結合了醫師、護理人員、神經心理師、社工師、職能治療師等不同專業的資源，提供相關的評估、診斷與衛生教育等服務，可以改善病人的神經精神行為，並對家屬日後照顧上有明顯的幫助。

且沈雅倫[10]等人也曾運用糖尿病關懷志工，對社區糖尿病個案，推動自我照護及血糖控制的重視，顯示社區關懷對糖尿病個案自我照護，有正向的提升影響，

而中央健康保險局亦為提升醫療品質，提供民眾周全性、協調性與持續性的服務，以及建立家庭醫師制度，於九十二年三月起，提出「全民健康保險家庭醫師整合性照護制度試辦計畫」，以加強醫療照護及轉診之整合。此計畫為同一地區的基層診所醫師 5-10 位為單位，組成一個「基層醫師團隊」，並與當地醫院共同合作成立「社區醫療群」，來為社區民眾提供整體性之醫療照護，與預防性之保健服務，且九十三年度更預定擴大推動達到 200 個社區醫療群。

以上例子皆可說明，整合性的社區醫療是跨專業，由不同的專業如醫師、護理人

員、神經心理師、社工師、職能治療師等不同專業，跨機構如不同的基層診所，所形成的複雜流程，其所結合的團隊，共同為了提升社區的健康而努力，並且產生正面的效果。

就社區醫療團隊中醫院所扮演的角色，李孟智 [11]曾對於以醫院為基礎的社區醫療，提出下列原則與方法：

重新界定社區：考慮地理因素與可能利用性。

切確的與行政資源(公權力結合)：結合地方上的資源進入社區實施家訪、個案追蹤管理、和特定調查工作，其中更提到將社區居民資訊連結到醫院之資訊系統上。

利用各項醫療及預防保健加以不同等級的資訊化管理。

界定社區中迫切需解決的健康問題，利用醫院的資源與資訊化管理系統，加以追蹤管理，並定期評估其績效。

在其中強調了資訊系統在社區醫療上，所扮演的重要角色，而邱泰源[8]。也曾提出資訊導向為其台大醫療體系各醫療群的宗旨之一，並使用電腦處理社區中的病患資料，林璦[12]也曾在大里市開發健康管理檔案，建立區域民眾健康資料庫，落實糖尿病疾病管理及個案追蹤管理。使糖尿病患之疾病管理，能有效建立疾病管理評估指標。

註解 [v2]: [社區醫學教育與健康照護,台灣醫學 2001年 5 卷 2 期 P212-220 ,]

而為了整合參與社區醫療的各個資源，達到最佳的效率，沈慶盈[13]在個案管理應用於社區照顧之適合性探討中指出，「個案管理是一種有系統與計畫的整合性服務的方法與步驟，以幫助案主有效的運用服務資源，進而增進案主的社會功能。運用個案管理於發展及推動社區照顧，應是一個可行的方向。應用個案管理於社區照顧有其正向的效果。」，其中隱含了在社區醫療照護上，是以人為中心的照護方式，結合了個案管理的方法，結合各種專業資源，來對於社區中的民眾提供不中斷的照護，

達到使民眾健康的目的。

## 2.2. 主動式資料庫

傳統的資料庫是屬於被動形式，只處理使用者請求的查詢或交易。如利用標準的 SQL 語法，將使用者的身體狀況資料加以儲存，紀錄其血糖值多少，是否有其他併發症等等。而當資料有所異動時，如果我們要同時異動病歷資料，及衛教通知單時，就必須在病歷管理程式或處理衛教通知單的應用程式，去增加一些對於如血糖值等相關資料的檢查程式，這樣是相當沒有效率的作法，主動式資料庫(Active Database)[14]卻是提供了持續對於關注的狀況加以監視，當狀況發生時適時的驅動適當反應的機制。主動式資料庫具有下列特性[15]，

一、主動式資料庫是一種傳統式資料庫，使用者如果沒有使用主動式資料庫的功能，則應該完全的像傳統式的資料庫。

二、主動式資料庫支援 ECA 模型，必需提供定義事件、條件、動作的方法。

三、主動式資料庫必須支援規則管理，可以定義 ECA 規則，並加以保存，並讓使用者可以存取，及支援規則的開發，必須能夠修改規則，更必須支援可以啟用及停用規則。

四、支援執行模式，必須能夠偵測到事件發生，必須支援 binding modes，可以評估規則，可以執行動作。

五、必須提供不同的耦合方式，亦即當事件觸發後進行規則評估的時機。

六、必須支援 consumption modes。

七、必須支援管理事件的歷史紀錄，當支援組合式的事件時，就必須能夠提供事



件發生的歷史紀錄以便支援組合式事件。

八、必須發展出當法則衝突時的解決機制。

主動式資料庫包含了 Knowledge Model、 Execution Model 及 Management 三大部分[14]：

一、Knowledge Mode 包含了三個項目，事件(Event)、條件(Condition)及動作(Action)，依據此三個項目所訂出的規則，稱為主動規則(Active Rule)，其語法如下

On event

If condition

Then action

這樣的規則可被資料庫的事件(Event)所觸發，如資料庫的操作等，並且依據條件(Condition)的內容加以評估，如果內容符合，就執行此規則所定義的動作(Action)，而此規則是定義及儲存在資料系統中，並由資料庫系統來執行。

而主動規則具有表 2-1 中之特點：

表2-1 主動規則的特性表

內容	特性	說明
Event	Source $\subset$ {Structure Operation ,Behavior Invocation , Transaction ,Abstract ,Exception ,Clock ,	Source：事件的來源包含了 Structure Operation ；如傳統的關聯式資料庫中的 insert、

	<p>External }</p>	<p>update、delete 等資料操作指令。</p> <p>Behavior Invocation 如物件資料庫中的函式呼叫等。</p> <p>Transaction 如其他的資料庫交易指令的 commit、abort 等。</p> <p>Abstract 可以讓使用者撰寫特殊的應用程式送出信號加以觸發。</p> <p>Exception 是由例外情況所觸發，如未經授權的存取資料庫等。</p> <p>Clock 在某一個設定的時間點所觸發。</p> <p>External 由外部狀況所觸發。</p>
	<p>Granularity</p> <p><math>\subset</math>{Member ,Subset ,Set}</p>	<p>Granularity 此事件的物件是定義在集合(如所有的員工)或子集合(如所有的員工不包含醫師)或特殊的成員(如有特殊授權的物件)中。</p>

	Type $\subset$ {Primitive ,Composite}	<p>Type 包含兩種:</p> <p>原始(primitive)的事件：如呼叫一個方法 (method invocation) 、資料的異動、交易的操作、時間的事件等等。</p> <p>組合(composite)：指將一些原始的事件組合而成。</p>
	Operators $\subset$ {or,and,seq,closure,times,not}	Operations 運算子包含如 And、 Or、 Closure、Seq、Times、Not 等等。
	Role $\in$ { Mandatory ,Optional ,None }	<p>Mandatory 只支援 ECA 的模式</p> <p>Optional 表示可以支援沒有 Event 的定義，只有 condition-action 的方式。</p> <p>None 表示不可以定義 Event，只支援 condition-action 的方式</p>
Condition	Role $\in$	Role 在 ECA 規則中條件大

	<p>{ Mandatory ,Optional ,None }</p>	<p>部分是 Optional 的，如果為 None 則只支援 event-action 的方式，Mandatory 表示一定要定義條件。</p>
	<p>Context <math>\subset</math> {DB<sub>T</sub>,Bind<sub>E</sub>,DB<sub>E</sub>,DB<sub>C</sub>}</p>	<p>Context 定義哪一個條件被評估，包含下列 4 種資料庫的狀態</p> <p>DB<sub>T</sub> 資料庫開始目前的交易。</p> <p>Bind<sub>E</sub></p> <p>DB<sub>E</sub> 資料庫事件開始產生。</p> <p>DB<sub>C</sub> 當條件已經被評估。</p>
<p>Action</p>	<p>Options <math>\subset</math> { Structure Operation ,Behavior Invocation ,Update –Rules,Abort,Inform, External }</p>	<p>Structure Operation 資料庫的操作。</p> <p>Behavior Invocation 。</p> <p>Update-Rule 更新其他法則。</p> <p>Abort 放棄或終止交易。</p>

		<p>Inform 通知使用者。</p> <p>External 執行一些外部的動作。</p>
	<p>Context <math>\subset \{DB_T, Bind_E, DB_E, Bind_C, DB_C, DB_A\}</math></p>	<p>Context 在作用中的定義，類似於條件，可作用於 <math>DB_T</math>、<math>Bind_E</math>、<math>Bind_C</math>、<math>DB_E</math>、<math>DB_C</math>、<math>DB_A</math>。</p> <p><math>DB_A</math> 當動作已經被執行。</p>

二、執行模式是說明在執行期間(runtime)，如何處理的一套模式，可以分為以下列步驟：

1. 信號產生階段：由「事件」的來源，來檢視那些「事件」發生。
2. 觸發階段：依據目前為止的事件觸發相關的規則。
3. 排程階段：處理規則的排序，以解決規則衝突。
4. 評估階段：評估這些被觸發規則的「條件」，決定是否執行「動作」。
5. 實施階段：執行這些法則的「動作」。

每一個步驟的執行程序如何制訂，乃是依據表 2-2 的屬性來控制。

表2-2 執行模式特性表

特性	說明
<p>Condition-Mode <math>\subset</math> { Immediate, Deferred, Detached }</p>	<p>Immediate 事件觸發後即刻評估條件。</p> <p>Deferred 事件觸發後不即刻評估條件，如交易結束後再評估條件。</p> <p>Detached 事件觸發後評估在不同的交易中。</p>
<p>Action-Mode <math>\subset</math> { Immediate, Deferred, Detached }</p>	<p>Immediate 條件成立後動作即刻執行。</p> <p>Deferred 條件成立後不即刻執行動作，如交易結束後再執行動作。</p> <p>Detached 條件成立後執行動作在不同的交易中。</p>
<p>Transition Granularity <math>\subset</math> { Tuple ,Set }</p>	<p>Tuple 單一事件觸發一個主動法則。</p> <p>Set 許多事件收集後一起觸發主動法則。</p>
<p>Net-effect policy <math>\in</math> { Yes,No }</p>	<p>是否將許多資料庫的操作相同的資料視為單一操作，例如先新增資料再更新資料。</p>

	料，則會視為新增更新後的資料。
Cycle policy $\subset$ {Iterative , Recursive }	<p>Iterative 原始的事件所驅動的動作，會觸發其他的事件</p> <p>Recursive</p>
Priorities $\in$ {Dynamic ,Numerical ,Relative ,None}	<p>動態優先權沒有優先權及靜態優先權，靜態又可分為數值形式或相對形式。</p> <p>Dynamic 由最近的更新情況或是複雜的條件產生</p> <p>Numerical 事先定義一個數值代表優先權</p> <p>Relative 給定一個規則當兩個法則同時被觸發時，其中一個必須修先另外一個執行。</p> <p>None 沒有優先權。</p>
Scheduling $\in$ {All Parallel ,All Sequential ,Saturation ,Some}	<p>Scheduling 執行的方式</p> <p>All Parallel 全部平行處理法則。</p> <p>Saturation 在某種情境之下才處理特殊的法則。</p>

	<p>Some 指執行其中的一些或單獨的法則。</p> <p>All Sequential 全部循序的處理法則。</p>
<p>Error handling <math>\subset</math> {Abort,Ignore,Backtrack,Contingency}</p>	<p>Error handing 在法則執行期間錯誤的處理方式。</p> <p>Abort 直接交易中止。</p> <p>Ignore 忽略錯誤及續處理其他法則。</p> <p>Backtrack 回到法則開始處理狀態，</p> <p>Contingency 加入意外處理計畫來處理此錯誤，一般如資料庫提供的例外處理。</p>

### 三、管理部份

指的是系統提供對於規則的管理操作能力，其屬性如表 2-3。

表2-3 管理模式特性

特性	說明
<p>Description <math>\subset</math> {Programming Language ,Query Language ,Objects}</p>	<p>法則的描述是用資料庫的程式語言，查詢語言或是物件資料庫中的物件</p>
<p>Operations <math>\subset</math> { Activate, Deactivate,</p>	<p>Activate 指法則可以被觸發執行。</p>



Signal}	<p>Deactivate 指法則將暫時不允許被觸發執行。</p> <p>Signal 法則由外部系統產生 Abstract events 來觸發執行法則。</p>
Adaptability $\in$ { Compile Time , Run Time }	<p>Compile Time 法則只允許在編譯時期做變更。</p> <p>Run Time 法則可以在 run time 中直接變更。</p>
Data Model $\in$ {Relational ,Extended Relational, Deductive ,Object-Oriented	Data Model 可支援 Relational Extended Relational Deductive Object-Oriented 等形態
Programmer Support $\subset$ {Query,Trace}	Programmer Support 是否支援查詢法則及追蹤法則的執行。

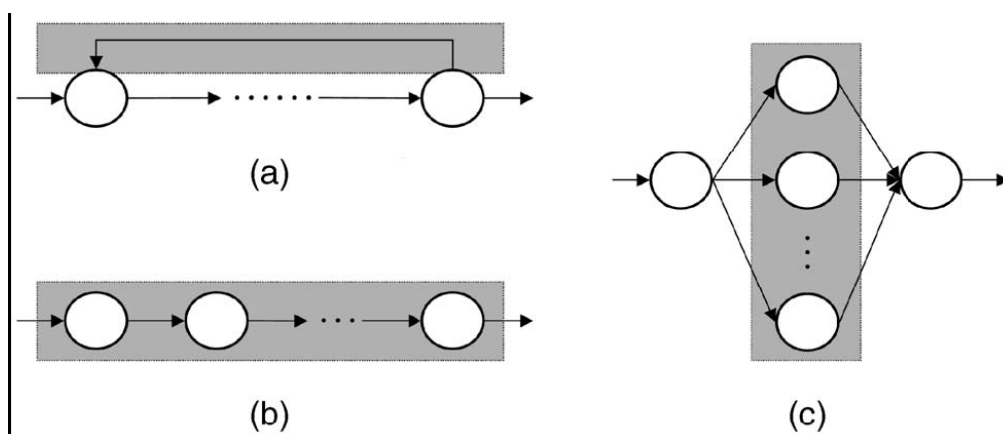
ECA 模型在資料庫本身的應用上，可以應用在維護資料的一致性、觸發程序、警示、維護衍生的資料等方面。

在相關的研究上，A. Goh [16] 等人曾經將 ECA Rules 運用在工作流程上，而其結論認為成功的關鍵在於，ECA 模型可以清楚地描述當政策改變時，有關動作應如何被執行，及關於此動作執行的限制。因此當一份表單的內容資料有異動時，我們

**註解 [v3]:** [A. Goh, Y.-K. Koh, D.S. Domazet ECA rule-based support for workflows, Artificial Intelligence in Engineering 15 (2001) 37-46]

可以藉由 ECA 規則，詳細的描述整個流程系統的反應。Joonsoo[17]等人提出利用流程分析，將流程型態分為為各個功能方塊，而將各個方塊歸類成如圖 2-1 三種行為模式，循序性(b)、並行性(c)、重複性(a)，再將各種行為模式的狀態轉換，利用 ECA 的法則一一建立，最後可自動將流程管理，運用 ECA 的法則一一對應，來支援流程控制，在其中提出對於流程控制，較為困難的是建立流程控管的 ECA 法則，這是需要許多的人工力氣去建立的。

圖2-2 流程功能方塊圖



## 2.3. 現有的電子化表單產品

### 2.3.1. 商用電子表單產品

目前市面上一般大眾較為知名的電子表單軟體，有 Lotus Notes 及 Microsoft InfoPath，分述如下：

Lotus Notes 是目前市面上最成功的群組軟體之一，改變了人們工作與協同合作

註解 [v4]: <http://www-8.ibm.com/software/tw/lotus/index.html>

的方式，為領先業界的訊息傳遞與協同合作用戶端。

Lotus Notes 是一個分散式的文件資料庫，其基本的架構是一個擁有通訊、溝通及共享資料的群組軟體。根據 Notes 的特性，適合在 Notes 上開發的應用程式共有下列五大類：

工作流程(Workflow)

廣播(Broadcast)

參考(Reference)

追蹤(Tracking)

討論(Discussion)

Lotus Notes 6 提供使用者功能齊全的協同合作環境，可透過 Notes Headlines 輕鬆個人化內容管理。Lotus Notes 提供電子郵件、行事曆、群組排程、網站存取及資訊管理功能，這些功能全部整合入一個易於使用且特製的環境，且無論線上或離線都可使用。

而在 Lotus Notes 家族中的 Lotus Domino Designer 幫助快速地建立及部署開放且更安全的協同合作應用程式。Lotus Domino Designer 為一個開放且整合的應用程式開發環境，提供多種平台支援及工作流程。提供使用者快速建置並部署開放、具高度安全性與關鍵任務的應用程式，並能輕鬆管理複雜的應用程式與分散各地的開發團隊。其具有完整、開放的開發環境，企業員工可以使用產業標準的程式設計技術，如 LotusScript (BASIC)、Java™ Runtime Environment (JRE) 版本 1.3、JavaScript 語言版本 1.4、HTML、XML、C 應用程式介面 (APIs)、CORBA、元件物件模型 (COM) 及物件導向程式設計等，直接在現有的平台建置與部署應用。而且 Lotus Domino Designer 能讓開發人員建立用於行動裝置的協同應用，以在具無線應用程式通訊協定

(WAP) 功能的裝置、簡訊服務 (SMS) 裝置，以及個人數位助理 (PDA) 上執行。此一功能讓開發人員建置必須達成企業需求的應用程式，同時讓一般使用者擁有維持生產力的彈性，甚至是移動當中也是如此。

Domino Workflow 是 Domino 內建工作流程功能的延伸產品，提供企業、組織功能更強大的工作流程管理系統，以便快速開發多樣化的 Domino 工作流程應用程式。Lotus 新推出的知識管理產品 Domino Workflow 讓企業可以非常容易地開發、管理及監視企業所有的商業流程，透過簡易視窗點選之方式可迅速開發完成高度複雜之工作流程應用程式，並可針對日後類似之流程應用進行再利用，即 Reusable Workflow Object，大幅提高生產力及日後之維護成本。

#### 重要特色和效益:

##### — 更快的程序、更短的週轉期

從客戶訂單到設計檢查，Domino Workflow 幫您更快速地處理程序以改進您企業的整體財務表現。以自動化的程序來取代仰賴紙張作業的處理程序，便能更妥善地管理現有的工作和流程。免去對耗時諮詢的需求，以及不知企業瓶頸何在。

##### — 與 Domino 相整合

採用 Domino 現有的工作流程功能並利用 Domino Workflow 作進一步的擴充。Domino Workflow 充分利用 Domino 的安控功能、訊息派送、複製及其他合作性的功能。利用您已有的訓練和專業知識來開發和管理更具效率的工作流程應用系統。您並可利用 Domino Designer 的設計功能來合併您所擁有的表單。

##### — 工作流程標準化

現在您可將工作流程標準化，這包括文件核准之派送路線和授權的簽署人姓名，而所有這一切都集中在單一的地點，並可將那些程序運用到任何應用系統上。您無

須再為每一套應用系統重新編寫相同工作流程邏輯的程式碼了。

#### — 視覺化介面

一個易於使用的視覺化介面可立即顯示您的工作程序，以方便您設計和隨即編修。對於瓶頸之分析、監控、和顯示的能力可讓您工作進度超前並維持領先的局面。工作流程中的每一步驟都可以視覺方式呈現，並以滑鼠點取的方式來進行變更。

#### — 加強的程序控制

透過更完善的流程管理可增進您的生產力。透過 Domino Workflow Architect，您能以更快更好的方式來應付截止日期、工作自動化、工作分派、和定義程序。工作之權責可指派給個人、工作群組、甚至整個部門。在無須任何額外費用的情況下，您可將 Web 客戶端加到您的 Domino Workflow 系統以便將工作流程擴及全世界。

#### — 降低擁有總成本

一個豐富多元的開發環境可讓您架構工作流程解決方案，而無需專門的程式設計經驗。藉由儲存並重複使用工作流程元件可節省開發的時間。由於它是運作於 Notes 和 Domino 之上，因此無須額外的訓練。

#### — 與 Domino.Doc 相整合

Domino.Doc 和 Domino Workflow 是天生的最佳拍檔。現在您可充分利用 Domino.Doc 所有強大的文件管理功能以及 Domino Workflow 最具威力的工作流程管理能力。您在 Domino.Doc 中儲存並管理文件的全生命周期，而後利用 Domino Workflow 來設計和管理更為強固的工作流程系統，以便在您的企業中派送那些文件和您的工作。

由此可知 Lotus Notes 是一套功能非常強大的系統，在此系統上可開發出與工作流程項配合的各式表單，但其為專屬之資料庫架構，對於表單的內容須與工作流程結合時，就必須由有經驗的程式開發人員，針對每一張表單的每一個欄位，進行程式設計，若要達到本系統欲達到的目的，隨時由使用者訂定工作流程規則時，將是不可行的。

#### InfoPath :

註解 [v5]: <http://www.microsoft.com/taiwan/office/preview/infopath/default.htm>

Microsoft 稱 InfoPath 為：「XML Web Service 的頭號智慧客戶端」。可把 InfoPath 當作是 XML 檔案之上的一個友善介面。它與在 Access, Excel 或是 Visual Basic 中開發表單是非常類似的。其可以讓客戶製做具有彈性的動態表單，即時獲取符合需求的資訊。並支援客戶自訂的 XML 描述結構，讓企業組織成員可以精確地將蒐集的資訊定義成結構化的資訊，並將資訊整合到既有的企業營運系統，從客戶關係管理、企業資源整合規劃、供應鏈管理，到工作群組的專案狀態及小組的專案進度報告等。

如此一來，所有資訊都可以重複使用，彙整資料更方便，提昇員工的效能和生產力，讓組織中更多的人員檢視企業的資訊，更快速地訂定出商業決策並採取行動，使組織得以提昇經營成果。IT 部門也能藉 InfoPath 快速開發或修改表單，並與其他微軟產品整合，達更快速因應商業需求的變更。

InfoPath 2003 最大的功能是讓小組及組織成員能夠很容易共同建立及使用豐富、動態的表單，同時以 XML 結構儲存以及整合 Web services 的強大功能，不但降低資訊科技成本，更提昇辦公室的工作效率。

InfoPath 是微軟 2003 Office 專業版裡的一個表單應用軟體，可以設計、填寫日常應用的各種表單，在設計上有許多控制項目可用，像是圖形使用介面、文字方塊、核取方塊、捲軸、指令按鈕，以及用來顯示資料、執行動作。填入表單裡的資料，

與 Word 類似，可拼音檢查、並以 RTF 格式儲存，同時可匯入匯出至其他檔。InfoPath 的重點是，除了在畫面上展現表單以外，還儲存了一份 .xsn 檔，是整個表單的 XML 資料結構，包括 XML 檔、XSD Schema 檔、XSL Stylesheet 檔、與 XSLT 轉換檔。這一 XML 資料結構，就能讓表單很容易的移到別的應用軟體，因為有越來越多的應用軟體接受 XML 資料。

InfoPath 表單看起來很類似於其他的 Office 表單。雖然使用者不需要了解螢幕畫面後到底在做什麼，但其實每當填好了一個 InfoPath 的表單後便產生一個正確標準的 XML 檔案，這也是 InfoPath 最大的優勢所在：藉由提供友善的前端和與標準相容的後端，Microsoft 將 InfoPath 視為不同廠商之間共同客戶端程式。

而因為其所產生的，只是在前端使用 XML 檔案，無論有沒有連上網路都能夠使用 InfoPath 的表格，相當適合經常出差的用戶來做各種運用。使用者能夠在旅途中在筆記型電腦上以 InfoPath 來填寫差旅費或旅遊券表單。之後，當使用者可以連上網路後，其他的應用程式就能夠使用那些已存檔的 XML 檔案。

事實上，縱使沒有重新連線，仍有絕佳的方式來讓資訊回存到使用者的網路裡：使用 XML Web Service。告訴 InfoPath 它應該要使用 Web Service 來傳送資料是一件容易的事情。在出差的例子中，使用者可以藉由 Internet 連上企業伺服器中的 XML Web Service，然後直接傳送完整的表格到 Web Service。在伺服器中，資訊會被重新導向 CRM、會計和其他的應用之中。

InfoPath 鎖定的目標是一般用戶，但不要指望用戶能夠自行做出設定。不論是否擁有 WYSISYG(所見即所得)的環境和精靈指引，光是設計和調校 InfoPath 表單就非得由 IT 專業人員出馬(雖然有些超級使用者(power user)可能會試試看自己設計表單)。值得加分的是，Microsoft 沒有因此省略了它其中的強大功能。因為 InfoPath 包含了腳本(scripting)和物件模型，所以資訊科技專家可以藉此而大幅度的自定表格行

為。每個事件都用標準的 XML 格式儲存，因此使用者也可以自選喜歡的 XML 工具 (toolbox) 來做修改。

當然，InfoPath 被設計與 Office 2003 其他部分共同作業。它可以直接輸出資料到 Excel，而且 InfoPath 的表格也可以儲存到 SharePoint 區，輕易地供給團隊成員使用。你也會發現 InfoPath 表單可在 Outlook 2003 和 Internet Explorer 中分享。

雖然 InfoPath 擁有如此大的功能，並且可以利用 XML Web Service 的方式與其他系統相容，但在流程結合方面，則必須依賴其他的軟體來配合，如 Share Point Server 等來搭配整個工作流程，因此如同 Microsoft 所言，只是智慧的客戶端，可將其視為單純的表單產生工具。

上述兩個軟體雖然都可稱為功能強大的電子表單系統，但是卻需要程式人員投入大量的人力，進行程式的撰寫，才能適切的運用表單的內容資料，觸發相關的工作流程。因此本研究決定提出另外不同的處理方式，來解決上述的問題。

### 2.3.2. 相關研究

在社區醫療多樣化的資料下，表單的結構有著與一般企業表單很大的不同，相關的研究，如陳金榮[4]等人為了解決臨床紀錄總類及格式繁多，建立了一個網基平台的臨床紀錄表單系統，將一份臨床記錄分為樣式(template)與內容(content)兩部份，讓使用者自行定義表單輸入畫面，產生樣式，並將樣式以 XML 格式儲存在伺服器，同時依據樣式產生輸入畫面，來收集表單資訊(content)，並儲存為 XML 檔案格式，且可加以列印輸出。

而趙咸欣[5]提出之應用於社區醫療資訊的結構化表單管理系統，具有自行定義表單結構，並能提供一個能幫助醫護人員蒐集與管理資料，而且整合與包容各種制式化的表單資料的資訊系統。

註解 [v6]: 陳金榮<sup>a</sup>, \*溫嘉憲<sup>a,b</sup>, 葉大成<sup>c</sup>, 陳琦文<sup>d</sup> A  
Web-based Platform for  
Implementing Clinical Notes  
2004 年國際醫學資訊研討  
會論文集 17-21



但上述兩項研究皆僅止於將資料蒐集，陳金榮[4]所提到其系統無法使用傳統式  
關連資料庫的 SQL 語言進行查詢或統計分析，而趙咸欣[5]亦將後續資料的運用列在  
未來研究上。

本研究採用趙咸欣[5]所提出之研究基礎，在其上加以應用，試圖解決在社區醫  
療的多樣化表單格式中，如何以表單為中心，把流程的定義表現出來，將表單資料  
賦予更大的意義，使得在社區醫療中收集的資料，不僅是紀錄，而是能夠緊密的運  
用於流程上。

**註解 [v7]:** 陳金榮<sup>a</sup>, \*溫嘉  
憲<sup>a,b</sup>, 葉大成<sup>c</sup>, 陳琦文<sup>d</sup> A  
Web-based Platform for  
Implementing Clinical Notes  
2004 年國際醫學資訊研討  
會論文集 17-21

## 第三章 系統架構與設計

在本章中，首先討論表單結構，接著提出資料驅動模型，及描述其中的定義

### 3.1. 表單結構

在社區醫療中，使用者因著各種環境的變化，必須隨時新增表單，套用在新的流程中，為了讓系統能夠運用新增的表單，必須先定義表單結構，讓系統能在既有的表單資料庫系統上，加以整合運用。

一份使用者所填寫的表單，可分成表單結構的描述與表單本身的內容資料。

#### 3.1.1. 表單結構的描述

為了描述表單的結構，可將表單視為由許多節點以樹狀結構來組成，其結構的描述具備有以下的特色：

以圖 3-1 此張表單為例，可將表單內的每一項資料或欄位，視為一個節點，再依據表單中的項目、內容等結構關係，可得出圖 3-2 的樹狀結構：如個案資料包含病歷號碼、身分證字號等項目，則在個案資料節點下，就連結著病歷號碼、身分證字號等節點。

圖3-2 社區醫療中的表單範例

# 兒童發展遲緩專案

## 綜合報告書

### 個案資料

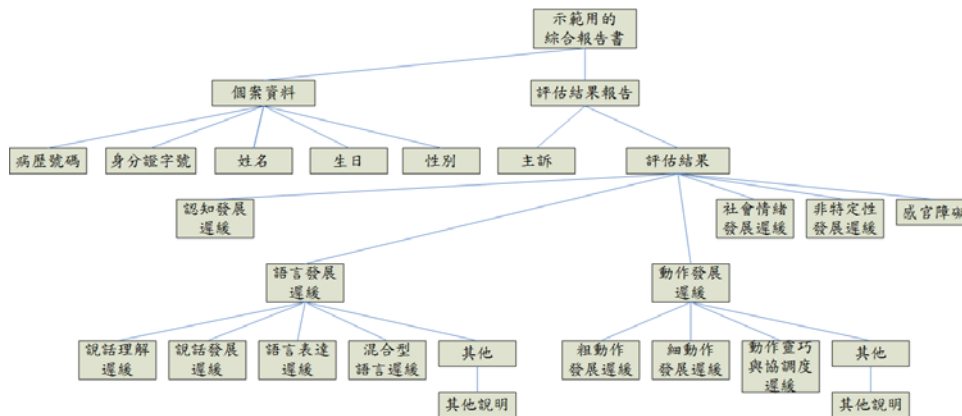
病歷號碼	身分證字號		
姓名	性別	生日	

### 壹、評估結果報告

類別	項目	內容
主訴與就診問題	主訴	
	<input type="checkbox"/> 發展遲緩 <input type="checkbox"/> 懷疑遲緩 <input type="checkbox"/> 發展遲緩	<input type="checkbox"/> 認知發展遲緩 <input type="checkbox"/> 語言發展遲緩 <input type="checkbox"/> 說話發展遲緩 <input type="checkbox"/> 語言理解遲緩 <input type="checkbox"/> 語言表達遲緩 <input type="checkbox"/> 混合型語言遲緩 <input type="checkbox"/> 其他
評估結果	<input type="checkbox"/> 動作發展遲緩	<input type="checkbox"/> 粗動作發展遲緩 <input type="checkbox"/> 細動作發展遲緩 <input type="checkbox"/> 動作靈巧與協調度遲緩 <input type="checkbox"/> 其他
	<input type="checkbox"/> 社會情緒發展遲緩	
	<input type="checkbox"/> 非特定性發展遲緩	
	<input type="checkbox"/> 感官障礙	

圖3-2

表單的樹狀結構



而各個節點有其屬性來描述節點的特性，

一、表單結構編號(form\_id)，每張表單結構皆有唯一的表單結構編號，代表此種類表單。

二、表單內的每一個節點，都具有節點代號(node\_id)，在此表單內為唯一。

三、每張表單的樹狀結構中，有一個特殊的節點，其父節點代號(parentNodeId)為-1，節點代號為 0，代表此張表單的根節點(rootNode)。而其他的節點，皆為根節點的子節點或孫節點，依次展開，如上圖的示範用的綜合報告書節點即為根節點。

四、每一節點為了描述在樹狀結構中的位置，皆具有父節點(parentNodeId)，及在同一個父節點底下的排行位置(levelSequenceNo)等屬性。

五、表單結構內的每一個節點，有一些其他的屬性如、資料型態(data\_type) 描述使用者輸入畫面時，以哪一種方式呈現、合法的型態列於小數點位置(dot)輸入時小數點位數限制、項目代號(item\_id)系統中相同項目的代號、是否為樹葉節點(leafNode)表示底下沒有接其他節點，或是沒有人的父節點指向此節點、資料長度(length)在輸入資料時的長度限制、如何產生輸入畫面描述及限制(uiConstrainDescription)、單位(unit)輸入畫面所顯示的單位，如公斤等資料。

本研究所提供的資料型態如表 3-4

表3-4 節點的資料型態

節點資料型態名稱	英文名稱	備註
單行的文字輸入欄位	Text Field	
多行的文字輸入欄位	Text Area	
單選	Single Choice	Radio Box
整數	Integer	
浮點數	Decimal	
日期	Date	
時間	Time	
多選	Multiple Choice	Check Box
布林值	Boolean	
家系圖	Genogram	

圖 3-3 的 XML 檔案為圖 3-1 設計後的結果

```
<?xml version="1.0" encoding="utf-8" ?>
<綜合報告書 form_id="4" rootNode="rootNode" data_type="0" dot="0" itemId="0" leafNode="True" length="0" levelSeqNo="0" node
parentNodeId="-1" uiConstrainDescription="" unit="" repeat="N">
- <封面 data_type="1" dot="0" form_id="4" itemId="283" leafNode="False" length="0" levelSeqNo="1" nodeId="1" parentNodeId="0"
uiConstrainDescription="%CTL:壹、"; unit="" repeat="N">
- <個案資料 data_type="1" dot="0" form_id="4" itemId="11" leafNode="False" length="0" levelSeqNo="1" nodeId="5" parentNode
uiConstrainDescription="" unit="" repeat="N">
<病歷號碼 data_type="1" dot="0" form_id="4" itemId="10" leafNode="True" length="0" levelSeqNo="1" nodeId="23"
parentNodeId="5" uiConstrainDescription="" unit="" repeat="N" />
<身分證字號 data_type="1" dot="0" form_id="4" itemId="303" leafNode="True" length="0" levelSeqNo="2" nodeId="24"
parentNodeId="5" uiConstrainDescription="%FUN:"ID_NUM";" unit="" repeat="N" />
<姓名 data_type="1" dot="0" form_id="4" itemId="2" leafNode="True" length="0" levelSeqNo="3" nodeId="25" parentNodeId=
uiConstrainDescription="%FUN:"NAME";" unit="" repeat="N" />
<性別 data_type="3" dot="0" form_id="4" itemId="3" leafNode="True" length="0" levelSeqNo="4" nodeId="26" parentNodeId=
uiConstrainDescription="%OPT:"1=男,2=女";%DEF:"1";" unit="" repeat="N" />
<生日 data_type="6" dot="0" form_id="4" itemId="4" leafNode="True" length="0" levelSeqNo="5" nodeId="27" parentNodeId=
uiConstrainDescription="" unit="" repeat="N" />
</個案資料>
</綜合報告書>
```

圖3-2

表單架構 XML 檔案

此表單結構是為了讓系統依據此結構自動產生表單輸入的畫面，並且依據表單結構的描述來儲存合法的表單資料，亦可依據此結構來設定條件規則。

### 3.1.2. 表單資料的描述

使用者所處理的資料，是符合某表單結構規定的 XML 檔案。當使用者填寫一份表單時，將其資料依據此表單的樹狀結構，加以儲存。而所填寫的表單資料，具有以下特性：

```

<?xml version="1.0" encoding="UTF-8" ?>
- <綜合報告書 form_id="4" form_no="0">
- <封面 node_id="1">
  - <個案資料 node_id="5">
    <病歷號碼 node_id="23" />
    <身份證字號 node_id="24">M[REDACTED]</身份證字號>
    <姓名 node_id="25" />
    <性別 node_id="26">2</性別>
    <生日 node_id="27">2004/03/16</生日>
  </個案資料>
- </綜合報告書 form_id="4">

```

圖3-2

表單資料 XML 檔案

每份表單資料皆有一個獨特的表單資料編號(form\_no)加以識別。其編號為唯一。

每份表單資料皆記錄其表單代號(form\_id)，表示此份表單的結構。

每個節點內容包括節點代號(node\_id)、節點名稱與節點內容。

每份表單只屬於一個主體，在實際的例子中，如病患若是加入糖尿病試辦計劃，就會填寫糖尿病初診評估單，保存其基本的糖尿病家族病史，生理狀況等基本資料，若該病患亦患有氣喘病，也可能加入氣喘的照護專案中，在這些多樣化的表單中，皆有其共同的特點，皆是記載著某個人的資料，因此表單有一特點，皆是屬於某個人，以人為中心。

為了容納具有不同表單結構的資料，我們利用 XML 來解決資料儲存格式上的問題，將一份表單存成一份 XML 檔案，依據表單結構，完整保存資料，又為了更快速的整理資料，因此每一個有意義的節點，另外存一份 INDEX，在關聯式資料庫，以正規化後 TABLE 的方式保存。

依據上述的特性，就可在表單系統中，利用表單代號及節點代號，來識別表單中的各個資料項目(節點)，而為了後續的處理應用各個資料項目，定義以[f?n?]為節點表示符號，如[f4n25]表示，表單結構代號為 4 節點編號為 25 的節點，意指綜合報告

書中姓名的欄位，再配合表單資料編號，就可以在全部表單資料中，輕易的取出某張綜合報告書內的姓名資訊。

### 3.2. 資料驅動模型

在一個許多角色互動的流程當中，若其流程是固定的，有其一定的關係，那可以依據流程中間的關係，來設計資訊系統提供支援，但若是其中的角色互動並不是很單純的單一模式，而是依據前一個流程的結果，來改變往後的行為模式，若其中銜接各個流程的資訊流通，又是透過大量不同格式的表單，則情況就很複雜了。

依據主動式資料庫的主動法則 ECA 的模型，在當事件發生後，能夠依據條件判斷，來觸發後續的動作，可以完全支援上述複雜的情形。當我們有了結構化的表單資料後，需利用其資料的內容，進行後續的作業，資料的內容不同，後續的作業規則也不同。如兒童發展遲緩的照護模式，與糖尿病患者的照護模式，絕對不相同。但在這些照護模式之下，有著相同的特點或特質可以符合 ECA 模式。其共同的處理原則，皆是當一個特殊的事件發生後，依據條件而產生後續的流程。依據此共同模式，本研究設計出以表單資料為驅動者，依據使用者自定的規則來達成社區醫療照顧流程的驅動架構。

在社區醫療中，當照護團隊依據病人的狀況，填寫一份表單，紀錄病患有價值的資訊時，我們可以利用其資訊的內容，再加上由各個專業對於資訊內容獨特的見解，來處理後續的動作，形成照護的流程。例如：某病患的血糖控制不良，就主治醫師的立場，認為必須要有合格的糖尿病衛教護理師，加入對此病患的治療團隊，我們可在填入表單同時，依據使用者的希望，發 EMAIL 通知糖尿病衛教護理師。以上的例子可以清楚地看出，在整個醫療照護中，其後續的流程，是因著表單資料的內容不同而改變的，其中所牽涉的角色也是如此。

依據 ECA 法則的語義：



ON Event

IF Condition

Do Action

以口語化的表達為，在某個事件發生時，如果條件符合，就執行其所設定的動作。

依據前述的實例，可以表達為如下的語義：

ON 填寫糖尿病表單

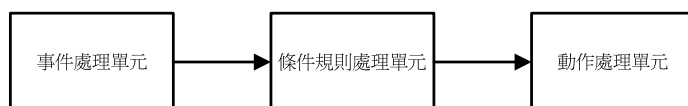
IF 血糖控制不良

DO 發 EMAIL 通知糖尿病衛教護理師

也就是說當使用者填寫糖尿病表單時，若其血糖值資料的內容經過判斷，為符合血糖控制不良的定義時，就執行發送 EMAIL 通知糖尿病衛教護理師的動作。在表單的填寫及異動中，當一張表單填入系統或更改一表單時，我們稱之為一事件。此一事件發生後，代表實際與此有關的作業將此展開，可能只是存檔備查，或是依據此表單進行篩檢，此種後續的作業稱為動作。而判斷此動作該不該發生的依據，稱為條件規則。如此可將 ECA 模型運用於整個照護流程中，可充分利用各個表單內容的資訊，來進行後續的作業流程。

為了符合社區醫療的資訊需求，我們提出以下表單資料驅動架構，來提供一個具有主動式資料庫 ECA 法則的處理能力，且以表單為基礎的資訊系統。

系統架構可細分為三個處理單元，一為事件處理單元，二為條件規則處理單元，三為動作處理單元，正對應 ECA 法則的 Event、Condition、Action。



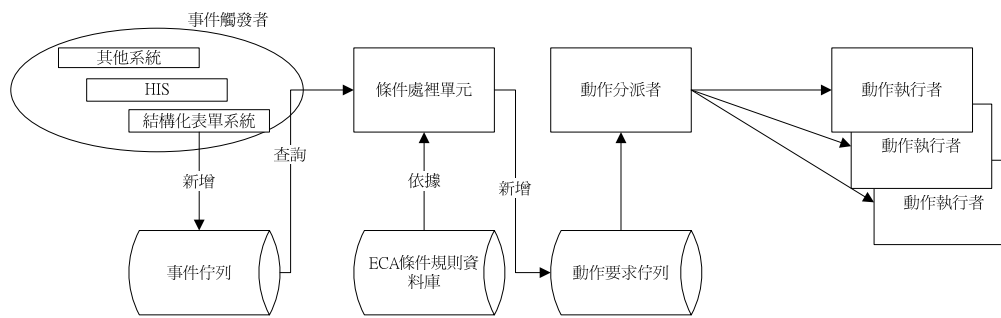


圖3-2

系統架構

而其詳細內容分述如下：

### 3.3. 事件處理單元

事件處理單元，主要的功能是處理所有的事件，也可稱是本架構的輸入端。

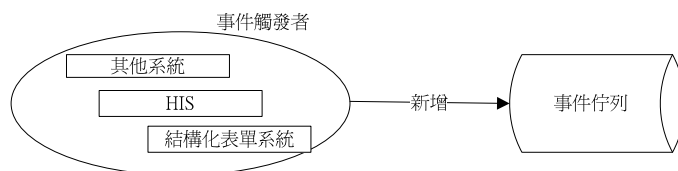


圖3-2

事件處理單元架構

在事件處理單元中，最主要的是事件觸發者。當結構化表單系統資料發生變化時，如使用者填寫一份新的兒童初診評估單時，事件觸發者會產生一個預先定義的事件，將事件的內容描述，新增到事件佇列中。為了能夠容納其他異質系統，只要能夠將資料新增到事件觸發紀錄，就可成為事件觸發者，而後續的流程就可以依此展開。

事件佇列的功能，主要是紀錄哪一張表單的哪一些事件已經產生，可視為是一個外界通知系統的窗口，不限定是哪一種事件。只要事先加以協定，而系統能夠認知，

就可以成為本架構下可以運作、支援的事件。此事件佇列可由條件規則處理單元，依據內容執行條件判斷，為了後續的流程能夠辨別事件的種類，因此事件的類別需事先協定。

### 3.4. 條件規則處理單元

條件規則處理單元，主要是當一個事件發生後，依據事件佇列中的資料，得知事件的種類，及表單的種類，再依據條件規則定義資料，使用者針對該份表單所自訂的條件規則，依據此條件規則，一一判斷條件是否滿足，若為滿足，則產生一個動作要求，在動作要求佇列資料中新增一筆資料。

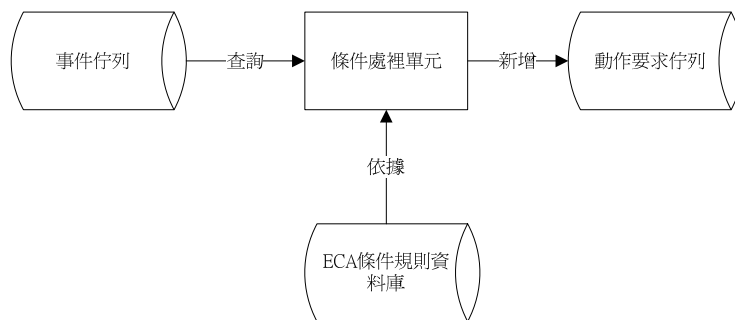


圖3-2

條件處理單元架構

ECA 條件規則定義資料，是主要使用者對於整個 ECA 法則的描述，是使用者針對各個所需面對的情形，自行訂定的規則，其中分成幾個要件：

該條件規則套用於哪一類的表單。

此條件規則的條件內容為何，亦即使用規則語言設定的條件。

當條件規則成立後，執行動作內容描述為何。

使用者自訂的條件規則其中最重要的一點，就是規則語言了，不同的規則語言可

能有不同的特性。

### 3.4.2. 條件的描述

在條件規則定義資料中，條件的描述其內容為一個字串的資料型態。本研究實作了一種條件表示式，其表示式是利用節點表示符號代表節點。如兒童發展遲緩初診評估單，其表單代號為 4，而其中系統功能的神經系統功能節點代號為 26，若規則需要以神經系統功能為判斷條件，則此節點代號的表示式為 [ f4n26 ]，再加上邏輯判斷式 >、<、>=、<=、!=、及四則運算符號及優先順序運算符號（）與 \$|! 等構成。而規則判斷之依據，利用整個規則的運算結果，為真則代表規則成立，可進行後續的動作，若為假則停止。

例如若是初診的評估項目神經功能異常，則通知社工人員，以此簡單的例子，由表單的定義中可以知道，該節點為單選的選項內容，1 代表正常，2 代表異常，3 代表不確定，因此只要在條件上設下 [ f4n26 ] = 2，若此條件式的運算結果為真，則會依據設定的條件內容，加入一筆動作要求佇列資料中。

條件規則的定義所依據的是表單結構，但是在實際的執行時，卻是將節點代號的表示式，置換成觸發的表單內容，來加以評估。

### 3.4.3. 動作的描述

當使用者在設定條件規則時，必須針對當條件規則符合後，其後續執行動作加以描述，在此僅定義其語義：其語義內容為，針對<對象> 執行<動作執行者> 依據<內容> 在<執行時間> 持續 <期間>，其所描述的情況是，當動作需執行時，其作用的對象是誰，而執行什麼樣的動作，依據或根據什麼樣的內容，在何時執行，執行多久，利用這樣的語義來描述動作的運作情形，至於詳細的內容，將由實作時加以協定。

### 3.5. 動作處理單元

動作處理單元，主要是處理，當條件規則符合使用者自訂的條件規則後，所須處理的動作，其角色可分為動作分派者，及動作執行者，

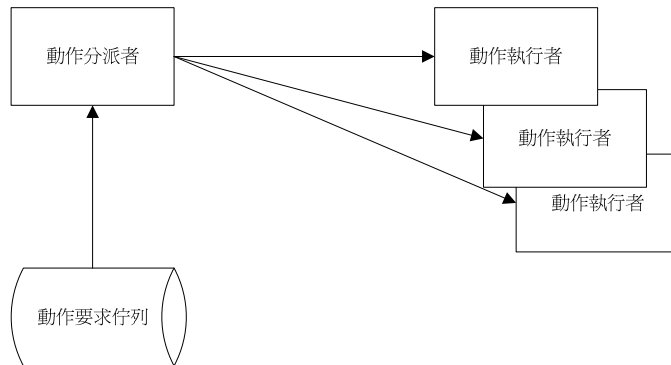


圖3-2

動作處理單元架構

動作分派者，依據動作要求佇列資料的內容，通知動作執行者執行預定的動作。動作分派者會持續監視動作要求佇列資料中的各項動作，若是發現其中有動作狀態為 Active 的動作項目時，便依據其動作型態，通知該行動所設定的處理方式來處理，亦即呼叫相對應的處理程式，如發送 E-mail 簡訊或是傳送呼叫器訊息等。

動作執行者，為實際動作執行程式，接受動作分派者的通知，而執行實際行動的程式，通知的內容利用參數的方式傳遞，參數的內容有兩個，以字串的形式來表達，一稱為對象、另外稱為內容，此兩參數的內容由使用者依據動作執行者的不同，各自訂定，因此動作執行者可依據實際需求，而加以增加，當現有需求的改變時，可加入新的動作執行者，來支援需求的變更。

依據此架構，可提供以表單為基礎的處理系統，擁有主動式資料庫的 ECA 法則的處理能力，更因使用者的可自行定義條件規則，不會受限於系統開發的支援程度，讓如此複雜的跨專業跨機構組織的團隊流程，可以依據使用者定義出最適當的條件

規則，依據表單的內容資料，觸發相關的處理動作，將各個流程密切的配合在一起。

## 第四章 系統建置實作與案例探討

在前面所提出的以表單資料為驅動的架構裡，在以下討論許多實作上的考量：

### 4.1. 事件處理單元

為了支援複雜的社區醫療流程裡，對於日常處理的事件，並不能限定於由本身的系統所產生，在整個照護團隊的各個醫療機構中，原本都有其各自的醫療資訊系統存在，為了能與各自獨立的異質系統結合，因此在事件處理單元中，必須設計出能夠相容的介面標準。

#### 4.1.1. 事件觸發者

當社區醫療照護的團隊中，發生一個事件或狀況時，必須通知本系統，才能加以觸發後續的流程。而通知的方式，就是在事件佇列資料表中，新增一筆事件紀錄，為了搭配既有的表單資料庫系統[5]，不修改目前的表單輸入程式，因此利用設定資料庫本身的 Trigger 功能，當表單資料一有新增或修改異動時，將此一事件資料，新增一個事件到事件觸發紀錄資料表中。

#### 4.1.2. 事件佇列資料表

事件要求佇列是以關聯式資料庫的形式來實作，其最重大的原因是關聯式資料庫是目前各系統支援最完整的，案例醫院的醫療資訊系統，使用的程式語言是 INFORMIX 資料庫專屬的 4GL 語言，難以利用其他的方式來溝通，如 WEB SERVICE 等，目前較新的技術架構皆無法支援，但其運用的資料庫也是關聯式資料庫，因此若是使用關聯式資料庫的形式來實作，則該系統的所發生的事件，可直接寫入事件

佇列資料表中，展開後續的動作，增加本系統的擴充性。

事件佇列資料表，記載事件被觸發的詳細資訊，其中包含

master_event [事件佇列資料表]	
event_no [事件編號]	
event_type [事件類別]	
form_no [表單資料編號]	
create_date [建立日期]	
notify_status [通知狀態]	

圖4-2

事件佇列資料表 SCHEMA

一、事件編號，每一個事件記錄皆有唯一的流水編號，在往後的流程中，用以識別觸發的事件。

二、事件本身的類型，目前定義有新增表單的事件及修改表單的事件。以社區醫療表單系統，此兩種事件是目前表單系統一切資料異動的觸發動作，若是其他異質系統要觸發事件，就必須先議定。往後的流程，才可以加以利用。

三、所異動的表單資料編號，紀錄哪一張表單資料觸發此事件。

四、並預設通知的狀態，在此預設為未曾通知，以便後續的程式，用以辨別事件是否經過處理，而後續程式處理完成此事件後，將狀態改為通知完成。

## 4.2. 條件規則處理單元

條件規則處理，主要是針對使用者所自訂的條件規則加以處理，在使用者定義的條件中，目前實作出一種規則語言，以節點表示符號[f?n?]，代表表單中節點的資料內容，在實際執行時，會依據觸發事件的表單編號，取得表單的資料內容，將節點符號置換為實際使用者所填的資料，再加以判斷，整個使用者所訂的條件是為真或



假。若為真則將條件規則內所定義的動作描述，填入動作記錄資料中，在填入資料的同時，若是資料的內容包含節點表示符號，亦會用該節點的實際表單內容資料，置換節點表示符號，使得動作記錄資料中，記載的是原本觸發事件的表單內容，例如在動作對象欄位中，所設定的是衛教師電子郵件帳號的節點代號，此時存入動作記錄資料的內容就會被置換為電子郵件帳號節點的內容。

#### 4.2.1. 條件規則定義資料表

圖4-2

規則定義輸入畫面

規則定義資料是由使用者於規則定義輸入介面程式中，加以逐項設定，其內容如下：

conditions [條件規則定義資料]	
condition_no [規則編號]	
event_type [事件類別]	
form_id [表單結構代號]	
description [說明]	
conditions [條件式]	
contents [內容]	
active_date [執行時間]	
action_code [動作代碼]	
target [動作對象]	
duration [持續期間]	
repeat [是否重複]	
create_date [建立日期]	
id [建立人員]	

圖4-2

條件規則定義資料 SCHEMA

表單代號：指此條件的設定是針對某一份特定的表單結構，而同一份表單結構可以設定許多條不同的條件。

事件類別：代表此條件規則，是針對哪一種事件類別所設定的。

說明：針對此條件的說明，只是為了留下紀錄說明，不在系統中運用。

條件式：為存放使用者自訂規則的內容，資料格式是一個文字字串，其內容為一種特定的規則語言，讓條件規則處理程式加以解析。

動作對象：指的是當規則符合時執行動作的對象，資料格式是一文字字串，其內容是由使用者，依據各個動作執行程式議定的規則填寫，目前系統實作出以下兩種動作執行程式，E-Mail 及簡訊。如果執行的動作設為 E-Mail 通知，則可能為對方的電子郵件信箱，如果為簡訊通知，則可以記載目標的手機號碼。將來陸續開發其他動作，如列印衛教資料，則可能為網路印表機代號，並且可以運用節點表示符號 [f?n?]，來表示實際動作時表單的內容。

動作代碼：指的是當條件符合後所要做的動作的代碼，即所謂的動作執行者，亦

指的是哪一隻執行程式，如是 E-Mail 通知或是簡訊等。

內容：則紀錄為通知執行這個動作的應用程式的內容，如簡訊通知的內容，請安排某某病患於兩週內進行衛教課程，其中若是有節點表示符號，則系統將會將實際表單的該節點內容，進行置換動作，如”請安排[f4n25] 病患於兩週內進行衛教課程”，可在系統執行時置換成，「請安排王小明病患於兩週內進行衛教課程」。

執行時間：當規則成立條件符合後，依其時間因素的考量，設定動作執行的時間，如果為立即執行，則為空白，若預定時間的話，則填入動作日期，亦即設定其符合條件內容後動作的日期，格式為[f?n?]+?天，表示依據當某個節點的日期，往後一段日子之後再動作，如兒童發展遲緩綜合報告書的預定追蹤日期，為發評估報告後的 3 個月，則可設定，評估日期的節點加 90 天，為動作日期。

持續期間：基本上為無限期，若填入值 7，代表該動作期間由動作起日期到七天後為截止日期，例如連續通知一個星期。

是否重複：表示同一張編號的表單可能修改許多次，每次都會觸發動作，如果設定不重複，則相同編號的表單，相同的規則動作只執行一次。

### 4.3. 動作的描述

#### 4.3.1. 動作分派者

動作分派者會定時監視動作要求佇列資料的內容，當動作的狀態為 active，而開始的時間已經到了的時候，或當有新的立即性的動作產生後，會依據動作的種類，通知動作執行者，不同的動作種類，有相對應的動作執行者，如果動作的種類為 E-MAIL 通知，則相對應的動作執行者為一支 E-MAIL 發送程式，如動作的種類為呼叫資料庫中的預儲處理程序，則對應的動作處理則會呼叫資料庫中的預儲處理程序，動作分派者會將動作記錄資料中的目標及內容傳給動作處理者，如 E-MAIL 通

知則目標即是被通知者的 E-MAIL Address，而內容就是信件的內容，如果是預儲處理程序，則目標就是預儲處理程序的名稱，而內容就是預儲處理程序的參數。

### 4.3.2. 動作執行者

因社區醫療沒有組織界線存在的特性，實際參與的人員並不限於組織中，因此在系統的執行動作上，並不能限於資訊系統的本身，而是須保留彈性，讓系統能與外界溝通。而相互作用的管道，在架構中就是動作執行者的角色所扮演的，其為實際執行使用者預設動作的程式，接受動作管理者的呼叫後，依據傳遞的參數來執行預定的作業，依此定義下只要開發出符合可接受參數的程式，就可以增加系統的功能，目前系統只實作如發送 E-MAIL 程式，或是簡訊傳呼程式，若流程需要的動作不在目前系統所支援的範圍內，則可以適時的開發新的功能程式來加以支援，例如將某一特定表單的資料，轉換格式，自動填入目前現有的醫療資訊系統中。

### 4.3.3. 動作要求佇列資料表

各個符合條件規則須執行的動作，皆先行記錄在動作要求佇列資料中，再由動作分派者來通知執行，其內容包括：

動作編號：各個動作皆賦予一個唯一編號。

表單編號：說明該動作是依據哪一張表單資料的內容所觸發的。

說明：說明描述。

內容：要通知動作執行者的內容。

執行時間：當條件成立後，依據條件的內容所設定預計執行的時間，動作管理者會依據此一時間，通知動作執行者執行動作。

結束動作時間：使用者設定在動作執行一段時間後，自動停止此一動作的時間設

定。

目標對象：指的是此動作當動作執行者所執行動作的對象，如欲利用 E-MAIL 通知其他照護團隊人員時，通知對象的 E-MAIL Address。

持續期間：指動作需持續多久。

是否重複：指同一標單處發同一規則時，若符合是否執行兩次動作。

建立日期：此動作建立的日期，當規則符合的日期。

action_pool [動作要求資料]
<b>action_no [動作編號]</b>
condition_no [條件編號] form_no [表單資料編號] description [說明] contents [內容] active_date [執行時間] inactive_date [結束動作時間] action_code [動作代碼] target [目標對象] duration [持續期間] repeat [是否重複] action_status [動作狀態] create_date [建立日期] id [建立人員]

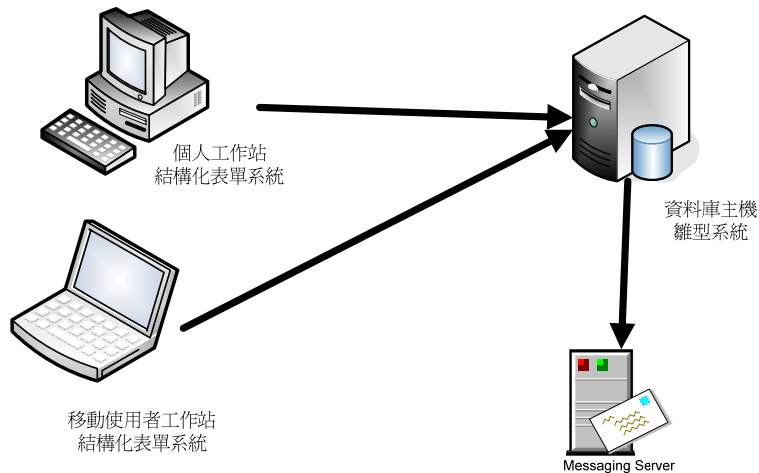
圖4-2

動作要求佇列資料 SCHEMA

依據此架構可容納在社區醫療的各式表單，將社區各個流程納入，依據使用者的需求，自行訂定驅動的條件，來整合整個照護的流程。如三段五級的照護中，當社區民眾第一次納入本系統中，則每年排定的健康篩檢計劃，皆可以通知了。若是患有糖尿病的病患，更可以依據病人不同的家族病史，生理徵狀符合規則而施以不同衛教通知。

#### 4.4. 硬體架構

在社區醫療的實際作業中，照護團隊不僅會在機構中產生表單資料，亦會到其他機構去訪視患者，因此前端表單蒐集系統，並不限於機構中的個人電腦，亦可能為離線工作者，但所有的表單資料皆會在事後，傳回資料庫主機中，而本系統也置於此資料庫主機中。



#### 4.5. 軟體架構

本研究之軟體架構，可分為下列幾項：

作業系統：於資料庫伺服器端採用 Microsoft Windows 2000 中文版搭配 Service Pack4，而其他皆是使用 Microsoft Windows XP SP2 中文版。

資料庫管理系統 (DBMS)：採用 Microsoft SQL Server 2000。

開發工具：採用 Microsoft Visual studio .Net 2003，採用的語言是 VC#，C# 是

一種先進的、物件導向的語言，可讓程式設計人員快速而輕易的建置 Microsoft .NET 平台的解決方案。

C#(唸成「C sharp」) 是 Microsoft 新開發的程式語言，用於發展在 .NET 平台上運作的元件式 (component-based) Internet 應用程式與服務，C# 擁有 C/C++ 的強大功能以及 Visual Basic 簡易使用的特性，和 C++ 與 Java 一樣亦為物件導向 (object-oriented) 程式語言；雖然微軟聲稱其 C# 的語法 95% 幾乎與 C++ 相同，但 C# 的語法其實更像 Java，更確切地說法，C# 的語法其實是由 C/C++、Visual Basic 和 Java 四種語言融合而成。

此語言不但提高了開發人員的生產力，同時還能消除可能造成開發成本增加的程式設計錯誤。C# 替 C 與 C++ 的程式設計人員帶來快速開發 Web 程式的方法，同時還能保有這些開發人員所需的強大功能與彈性。

## 第五章 討論

主動式資料庫所提供的主動法則，其目的是為了支援資料庫本身的功能，相對的本研究所提出的架構，是全然為了支援社區醫療活動。

在整個社區醫療的支援系統中，相關研究提出結構化表單系統，來支援各式各樣的社區醫療活動，在這些活動之中，人的活動與此系統之介面，是經由表單輸入來進行，因此最直接的事件就是表單的輸入及異動，雖然本架構沒有如同主動式資料庫所提供的各種不同組合事件，來支援整個流程，但為了與異質系統的連結，事件的定義是可以事先加以議定而增加的，並不會受限於目前系統所提供的事件類型。

在此驅架構中留下了一個介面，只要能夠依據事件觸發紀錄的資料格式，將資料填入，則不限定只有既有的表單資料庫系統，可以運用此架構來觸發展開後續的一切活動，例如在醫院急性醫療的資訊系統中，若是檢驗室的報告已經發出來，可以開發出一個簡單的介面，讓系統自動將此一個事件，加入事件觸發紀錄中，而不必經由表單的輸入來觸發此一事件，如此可達到與異質系統連結的目的。

Mario[18]等人在提出一個整合性的健康照護網路資訊架構時，即明確地指出，在設計系統架構上，最核心的幾項特性就是擴充性、適應性、整合性，依此特性來檢視本架構，本架構在條件規則部分由使用者自行定義，自行增加，並採取模組化架構，在事件方面提供事件觸發記錄資料的格式，可另行開發系統直接產生事件，連結本系統觸發規則，在動作方面只要能接受議定的呼叫方式介面，就可以自行開發增加系統功能，而條件規則語言方面也可以日後自行增加，因此符合上述的特性。

Xiaofmg[19]提到 rule-base 的程式是不容易維護，無法測試，不可靠的：

因為維護的人員與系統開發人員相交集應該只有法則，但維護法則的人員不能百

註解 [v8]: Mario Beyer , Klaus A. Kuhn , Christian Meiler , Stefan Jablonski , Richard Lenz , Towards a Flexible , Process – Oriented IT Architecture for an Integrated Healthcare Network , 2004 ACM Symposium on Applied Computing

註解 [v9]: Xiaofmg Li, NEC America 於 what's so bad about rule-base programming



分之百了解開發人員如何開發，就會造成維護法則上的困難，又其獨特的架構，並不像其他的系統可以適應外界的變化慢慢的演化修正，因而不易維護。但在本架構系統中，因著模組化的設計，利用表格資料的鬆散耦合，可以隨時抽換其中的模組，或增加模組的功能，因此可以輕易的適應外界的變化。

而 rule-base 不容易測試，是相較其他一般的系統，有著單一的輸入固定的輸出，非常的容易測試，但 rule-base 的系統卻因為 rule 的複雜性，如其中有許多的 IF—THEN 的不確定性而難以測試。在本架構中因著在整個規則進行的中間過程，皆是用表格資料的形式溝通，因此可以留下紀錄，非常容易測試，確保系統的正確性。

而可靠性上因著不容易測試，相較之下可靠性低。在此架構中亦有輔助的日誌記錄(LOG)，雖不在此架構中闡述，但對於上線後的偵錯或追蹤亦提供了幫助。

## 第六章 結論與未來研究方向

### 6.1. 結論

在沒有組織界線的社區醫療中，隨著個案的健康情形不同，其記錄資料內容也不相同，所相關的作業流程也會跟著改變。

本研究整合相關研究的表單資料庫系統，提供主動式資料庫 ECA 法則的支援，使得表單資料庫系統具有主動處理的能力，在實際的運用上，此架構不會受限於只支援目前已知的表單格式，而能隨著使用者隨時增加新的表單定義，就可以經由新表單實際的資料內容，依據使用者自行定義的規則，驅動後續指定的處理流程，可充分支援社區醫療多變而不確定的資訊需求。使得大量的照護流程中的表單，不僅只有收集儲存的功能，更能加以充分應用其中的資訊，驅動後續有意義的相關流程，來提升照護的品質，使其所照護的病患更能邁向健康。

### 6.2. 未來研究方向

在此架構下，未來研究可朝下列方向進行：

一、不限定只支援目前表單結構描述所產生的資料，利用加入 XPATH 與 XQL 語法的支援，能直接支援 XML 檔案格式的資料。

二、能夠提供支援跨表單的條件定義機制。

三、對於後續流程加上角色權限代理等，與實際團隊人員相關的互動機制。

## 第七章 參考文獻

- [1] 陳慶餘,「家庭醫學與社區健康營造」,台灣醫學 2001 年 5 卷 2 期 188-193
- [2] 張煥禎,「打造健康管理新紀元-從疾病治療照護到健康服務管理」,醫院.34 卷 44-47
- [3] 黃源協,「社區照顧團隊的建構與管理」,社區發展季刊第二十九期 P141-159
- [4]. 陳金榮,溫嘉憲,葉大成,陳琦文「A Web-based Platform for Implementing Clinical Notes」 2004 年國際醫學資訊研討會論文集 17-21
- [5] 趙成欣,「應用於社區醫療資訊的結構化表單管理系統」,民 93 國立暨南國際大學資訊管理研究所碩士學位論文
- [6] Robert B. Williams, Myde Boles, Robert E. Johnson,「Patient use of a computer for prevention in primary care practice」, Patient Education and Counseling 25 (1995) 283-292
- [7] 管敏秀,「社區健康營造之現況分析與策略研究」,民 91 銘傳大學公共事務學研究所碩士學位論文 P11-12
- [8] 邱泰源,「社區醫學教育與健康照護」,台灣醫學 2001 年 5 卷 2 期 P212-220,
- [9] 甄瑞興,賈淑麗,盧玉強,「多專業整合之失智老人照護模式 — 以某區域醫院經驗為例」,社區發展季刊第二十九期,206-214
- [10] 沈雅倫,葉莉莉,蕭尊鳳,「社區關懷對糖尿病個案照護之經驗分享」,社區發展季刊第 104 期 P422-431

- [11] 李孟智 , 家庭醫學與全民健保醫業管理 2003 ,合記圖書出版社 p147
- [12] 林 璫 , 「區域醫院及診所如何做好糖尿病照護個案管理? — 以臺中縣大里市為例」, <http://www.mc.ntu.edu.tw/department/cmrg/921paper/14-05paper.htm>
- [13] 沈慶盈 , 「個案管理應用於社區照顧之適合性探討」,社區發展季刊第十八期 P254-264
- [14] NORMAN W. PATON, OSCAR DI´AZ “Active Database Systems”, ACM Computing Surveys, Vol. 31, No. 1, March 1999
- [15] Klaus Dittrich, Stella Gatzju, Andreas Geppert , “The Active Database Management System Manifesto: A Rulebase of ADBMS Features”, SIGMOD Record, Vol. 25, No. 3, September 1996
- [16] A. Goh, Y.-K. Koh , D.S. Domazet “ECA rule-based support for workflows” , Artificial Intelligence in Engineering 15 (2001) 37-46
- [17] Joonsoo Bae , Hyerim Bae , Suk-Ho Kang , and Yeongho Kim “Automatic Control of Workflow Processes Using ECA Rules” IEEE TRANSACTIONS ON KNOWLEDGE AND DATA ENGINEERING, VOL. 16, NO. 8, AUGUST 2004
- [18] Mario Beyer , Klaus A. Kuhn , Christian Meiler , Stefan Jablonski , Richard Lenz , Towards a Flexible , “Process – Oriented IT Architecture for an Integrated Healthcare Network” , 2004 ACM Symposium on Applied Computing
- [19] Xiaofmg Li, NEC America, “What's So Bad About Rule-Based Programming? “ , IEEE Software, vol. 08, no. 5, pp. 103, 105, September/October 1991