

# 以雲端運算平台建置電子病歷 檢視器系統之研究

The Study of designing a Viewer of Electronic Medical Records  
under Cloud Computing Platform

國立臺北護理健康大學 資訊管理所

指導教授：黃衍文 博士

學生：徐明睿

關鍵字：電子病歷、臨床文件架構、雲端運算、檢視器、超文本標記語言第五版

Keywords: Electronic Medical Records (EMR), Clinical Document Architecture (CDA),  
Cloud Computing, Viewer, Hyper-Text Markup Language 5 (HTML5)

# 大綱

- 摘要
- 緒論
- 文獻探討
  - 電子病歷
  - 資訊技術與服務
  - XML
  - Web 3.0
  - 雲端運算
  - 多國語系應用程式
  - 其它相關研究
- 研究方法
- 研究結果與討論
  - 樣板管理模組
  - 文件讀取及解析模組
  - 資料內嵌處理模組
  - 樣板輸出處理模組
  - 多國語系模組
- 結論與建議

# 摘要

- 國內衛生署推行電子病歷資料交換已經施行多年
- 必須透過健保局提供的虛擬私有網路與衛生署電子病歷交換平台交換電子病歷資料
- 病患的健保卡與醫師的醫事憑證卡
- 病歷資料的交換互通與即時處理顯示已經成為了未來的趨勢
- 以公告的標準文件、超文本標記語言第五版技術及資訊技術與服務的構想來設計俱有永續經營的電子病歷檢視器
- 部署至雲端平台
- 大幅增加電子病歷的可用性

# 緒論

# 研究背景

- 電子病歷的資料交換格式是由HL7制定的臨床文件架構所衍生的一種XML格式的文件
- 醫院必須透過各自的EEC Gateway，經過健保局提供的VPN，與衛生署電子病歷交換平台(EEC)交換電子病歷資料
- 經過衛生署認證後，才能上線至EEC平台開始做電子病歷交換
- 民國九十七，衛生署修訂了108項電子病歷單張範本成為國際表準的HL7 CDA R2格式
- 『醫院實施電子病歷及互通補助計畫』公告並開始使用四張醫院病歷單張與中醫、西醫與牙醫等三張門診單張交換使用

# 研究動機

- 雲端服務與雲端醫療的爭相競爭之下，以及有不少學者已經開始著手使用雲端運算技術、Web 3.0與HTML5等新技術應用至醫療領域的研究
- 目前國內政府力推全民健康雲

# 研究目的

1. 提供各家醫院能夠自行設計客制化的樣板檔來顯示各種電子病歷單張，並且不會因為資料交換而導致顯示不同
2. 提供上述各醫療院所設計的電子病歷單張樣板檔能夠上傳至雲端平台共用的功能
3. 能夠在有電子病歷檔案的時候任何地方使用，不需要依賴 HIS 的處理即可顯示病歷資料
4. 提供產生其他格式的 XML 文件以增加電子病歷的多用途與可用性
5. 能夠符合資訊系統共享及永續經營理念，提供醫療院所可以直接使用本檢視器平台，降低資訊系統重新造輪的狀況

# 文獻探討

依照研究目的，首先探討電子病歷的歷史與規格及國內交換的方式，並且依序探討資訊技術與服務概念，以及探討建置電子病歷檢視器平台所使用的技術等

# 電子病歷

- 電子病歷(Electronic Medical Record, EMR)是一種由電子化方式擷取、傳送、儲存、取回、連結與處理的多媒體資料
- 電子病歷主要的用途為協助醫療或其相關服務
- 內容包括病患的個人資料、問題、SOAP、病程紀錄、護理計畫、護理記錄、生命徵象記錄、藥物使用記錄、過去病史、家族病史、預防接種記錄
- 國內的電子病歷則是使用HL7的CDA R2的規格
- 必須經由健保局提供的VPN網路與EEC Gateway交換資料

# 資訊技術與服務 - 基礎構想

## (1/2)

- 源起於1980年代的英國政府商務部的前身組織 -- 顧客信用貿易協會所主導進行的「如何提升政府資訊科技與管理」研究專案
- ITIL之發展有三個階段：
  - 針對IT技術提供者(功能基礎)
  - 針對服務提供者(處理流程為導向)
  - 針對策略夥伴(生命週期管理)

# 資訊技術與服務 - 基礎構想

## (2/2)

- ITIL包含七個模組
  - 服務管理(Service Management)
  - 服務支援(Service Support)
  - 服務遞送(Service Delivery)
  - ICT基礎架構管理(ICT Infrastructure Management)
  - 規劃到建置服務管理(Planning to Implement Service Management)
  - 應用管理(Application Management)
  - 企業遠景(The Business Perspective)
  - 安全管理(Security Management)

# 資訊技術與服務 - 服務管理

## (1/2)

- 服務管理內細分的服務支援及服務遞送
- 服務遞送有以下五種管理：
  - 容量管理(Capacity Management)
  - 服務水準管理(Service Level Management)
  - 可用性管理(Availability Management)
  - 服務永續性管理(Service Continuity Management)
  - 服務的財務管理(Financial Management for Service)

# 資訊技術與服務 - 服務管理

## (2/2)

- 可用性管理有兩個處理流程階段：
  - 透過適當資源、方法和技術的部署，從而確保與顧客約定之服務應用(SA)可用性目標得以實現的流程。
  - 主要涉及維護活動的最佳化規劃，以及設計相關措施，將事件發生的次數降至最低。
- 服務永續管理：
  - 找出所需要的和協議的商業營運最低層級，以及系統、設備和服務需求所涵蓋的需求定義
  - 檢查對這些需求的風險及威脅
  - 開發新系統的風險降低或是減輕計劃

# XML

- 可擴展標記語言(Extensible Markup Language, XML)是一種標記式語言
- 標記指電腦所能理解的訊息符號，使用此種標記，電腦與電腦之間可以互相處理包含各種訊息的資料
- XML是從標準通用標記式語言(Standard Generalized Markup Language, SGML)中簡化修改出來
- XML是設計用來傳輸及儲存資料資訊，不是用來顯示或呈現資料，相對的HTML則是用來呈現資料

# XML的主要用途

- 豐富文件(Rich Documents)
  - 屬於以檔案為主的應用
  - 標記是用來定義資料應該如何呈現
- 中繼資料(Metadata)
  - 屬於以資料為主的應用
  - 標記是用來說明資料的意義
- 配置檔(Configuration Files)
  - 屬於以屬性為主的應用
  - 標記是用來配置資料或應用程式的設定

# XPath

- XPath即是表示資料或屬性在XML文件中所在的位置的一種表示方法
- 主要有以下表示及搜尋方式：
  - 使用任意搜尋符號"//node"：即任何符合"node"的元素都會被選取
  - 使用絕對搜尋"/nodeA/nodeB"：即是符合於根元素nodeA之下的nodeB將會被選取
  - 使用模糊搜尋"nodeA/nodeB"：即是符合nodeA元素之下的nodeB將會被選取
  - 使用含有屬性搜尋"//node[@attribute]"：即是符合node元素後，包含attribute屬性者，將會被選取
  - 使用條件搜尋"nodeA/\* except self::nodeB)"：即是符合nodeA元素之下的任意元素，但不包含nodeB的元素皆將會被選取

# Web 3.0

- 他的定義在2006年11月的Technet峰會上首次被提出
  - Netflix創始人Reed Hastings認為「Web 1.0是撥號上網，50K平均頻寬，Web 2.0是1M平均頻寬，那Web 3.0就該是10M頻寬，全視頻的網路，這才感覺像Web 3.0。」
- 而隔年的8月7日，Google首席執行官埃里克●施密特出席首爾數字論壇認為
  - 「Web 2.0只是一個行銷術語，而你剛才正好發明了Web 3.0這個行銷術語。創建應用程序的方法將不同。到目前為止Web 2.0一詞的出現主要是回應某種叫做AJAX的概念，而對Web 3.0，我的預測將是拼湊在一起的應用程序」

# Web 3.0 cont.

◆ Web 3.0將會有一些主要特性：

- 應用程式相對較小
- 資料處於Cloud中
- 應用程式可以在任何設備上運行（電腦或者行動裝置）
- 應用程式的速度非常快並能進行客製化
- 此外應用程式像病毒一樣地擴散（社交網路，電子郵件等）
- 包含了Web 2.0的語義網路與鏈結資料
- 資料與技術的混搭應用

# HTML5

- 超文本標記語言第五版草案的前身名為Web Applications 1.0，是在2004年由WHATWG提出，再於2007年獲W3C接納
- 廣義的說HTML5，實際指的就是包括HTML、CSS和JavaScript在內的一套技術集合
- 它希望能夠減少瀏覽器需要外掛程式的豐富性網路應用服務(plug-in-based Rich Internet Application, RIA)，如Adobe Flash、Microsoft Silverlight，與Oracle JavaFX的需求

S. Jobs. (2010, Thoughts on Flash. Apple Inc. Available: <http://www.apple.com/hotnews/thoughts-on-flash/>

# HTML5 cont.

- 離線儲存資料庫(離線網路應用程式)：可以讓網頁不需要網路的狀態下儲存資料或存取資料，等有網路的時候再依照設計的模型決定是否要與伺服器同步資料
- 編輯：讓網頁直接編輯，不再需要透過編輯器或所見即得編輯器
- 拖放：使檔案的選擇不需要經由系統的對話視窗選擇即可使用拖放的方式將檔案放置在網頁中讓應用程式讀取資料
- 瀏覽歷史管理
- MIME和協議處理程式時表頭登記
- 檔案API：處理檔案上傳和操縱檔案
- 目錄和檔案系統：這個API是為了滿足客戶端在沒有好的資料庫支援情況下的檔案儲存要求
- 檔案寫入：從網路應用程式向檔案裡寫資料內容

# 自適應網頁設計

- Responsive Web Design是一種網頁設計的概念
- 可以使網站可以適應到多種瀏覽的裝置上(從電腦到行動電話或者行動裝置)閱讀，同時可以減少縮放和捲動
- 採用RWD設計的網站會使用到CSS3的媒體查詢，即是對CSS的 @media 規則的擴充，以適應不同大小的裝置

# 雲端運算

- 雲端運算是一種基於網際網路的運算方式，這種方式是繼1980年代的大型電腦到主從式架構的大轉變之後的再次巨變
- 通常可以認為包括以下三個層次的服務
  - 基礎設施即服務(Infrastructure as a service, IaaS)
  - 平台即服務(Platform as a service, PaaS)
  - 軟體即服務(Software as a service, SaaS)

# 雲端運算服務應具備的特徵

- 需要時即可使用的自助服務
- 寬頻網路存取
- 資源池
- 可快速重新佈署
- 可被監控與量測的服務
- 基於虛擬化技術快速部署資源或獲得服務
- 減少使用者端的處理負擔
- 降低使用者對於IT專業知識的依賴

P. Mell and T. Grance, "The NIST Definition of Cloud Computing," National Institute of Standards and Technology Special Publication pp. 800-145, 2011.

# 雲端服務的普及度

- 2012年11月，VMware委託Forrester Consulting 進行的市場調查中
  - 發現台灣的企業雲端的普及率為44%
  - 略高於亞太地區的平均值2%
  - 27%表示尚未使用雲端解決方案，但未來有打算使用雲端方案
  - 56%表示會在未來的18個月內落實
- 該調查也歸納出企業評估投入雲端前的重要因素
  - 資料隱私、所在地，或「無法掌握」(loss of control) (71%)
  - 成本 (69%)
  - 規範/管理/法規遵循議題 (67%)

# 多國語系應用程式

- 很多物件的名稱都有不同的稱呼方式
- 讓使用者可以更容易瞭解物件的名稱
- 必須要標註不同語言的名稱在物品上
- 多國語系的應用程式也是如此
- 若應用程式要往國際化發展的話，則勢必要有不同語系的翻譯

# 其他相關研究

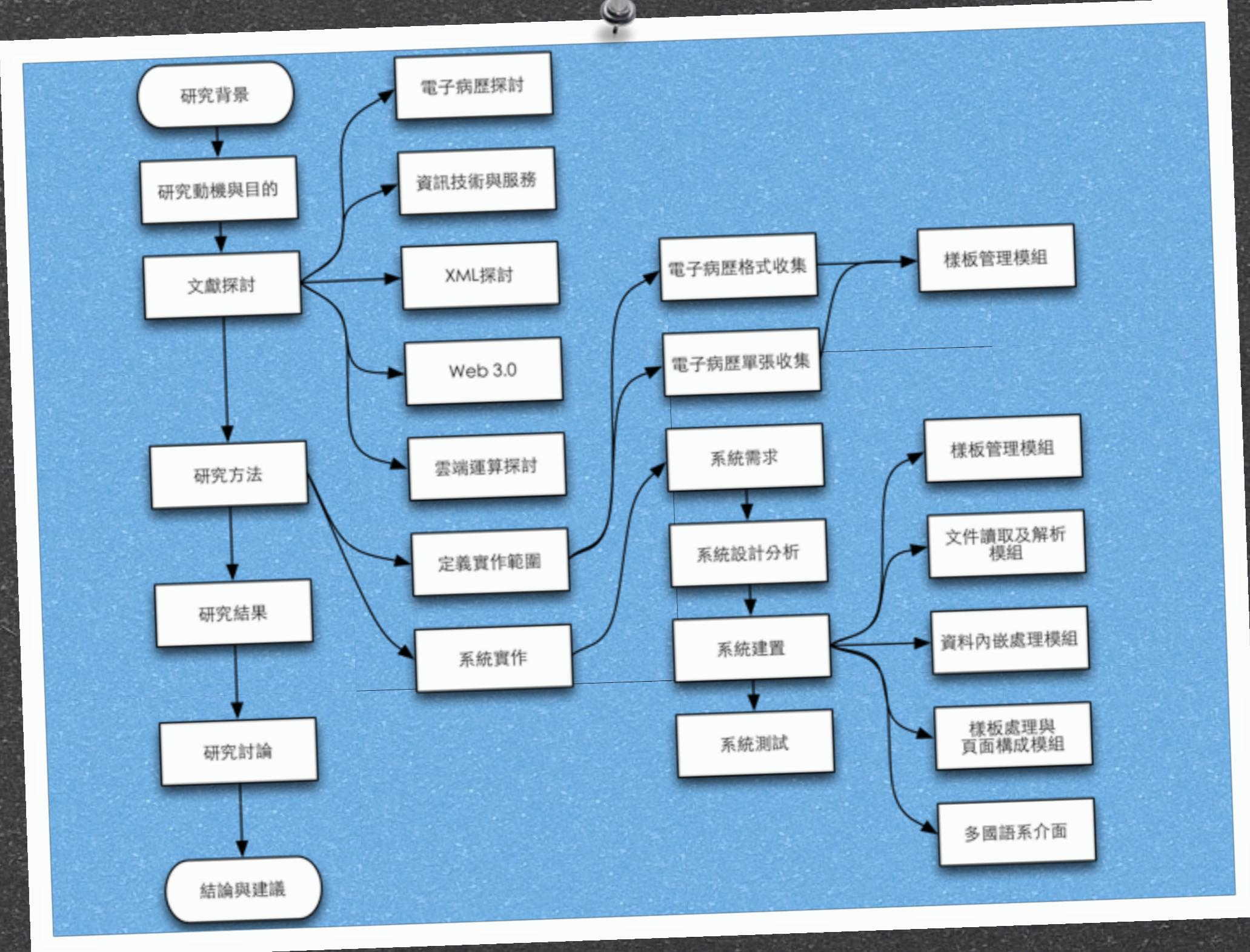
- Microsoft HealthVault與Google Health
- 陳榮駿(2011)撰寫之『應用於電子病歷呈現的臨床文件架構樣版管理平台之設計與建置』研究論文
  - 使用了可延伸樣式表語言來作為電子病歷檔案轉換套用成為顯示畫面的結果
  - 包含了一個醫學詞彙代碼的資料庫
  - 僅只有使用四張交換用單張，並無包含額外三張門診交換單張的內容
- 由Liwei, Hao等人之『使用HTML5增強電子病歷中的低劑量CT影像』研究使用了HTML5技術
  - 製作了一個純Web版本的增強式PACS檢視器

# 研究方法

本研究為實驗設計，並非一般商業軟體開發，故本系統建置流程不適合於一般商業軟體系統開發建置流程，所以較為簡化

# 研究步驟

- 本研究會設計一個電子病歷的檢視平台
- 提供各醫療院所及民眾使用
- 彙整技術的可行性及能否符合預期設計和目標
- 提供醫療院所做為一個電子病歷資料交換後的檢視平台
- 可於線上開發各種單張所需的檢視模板
- 最後驗證實作是否能達預期目標



# 研究流程與步驟

# 系統建置流程

1. 文件分析
2. 系統設計
  - 2.1. 建置系統框架
  - 2.2. 檔案解析介面
  - 2.3. 樣板檔管理
3. 雲端部署

# 系統需求分析與設計

根據系統建置流程中所提及的系統設計規劃中，將三個系統模組需要的功能及流程依序分析及設計

# 需求分析與系統架構 (1/2)

- 由於各醫療院所所使用的電子病歷樣板皆不相同，即使是同樣的單張，各自呈現的結果也不盡相同
- 讓各單位皆能夠快速的產生經由電子病歷提供的資料去產生表單
- 依照使用者，並且區分其代表的角色將會區分成三種使用者，詳細如下列所示：
  - 醫事單位表單開發設計人員：負責電子病歷樣板檔的設計以及擴充功能開發，讓醫事人員以及一般民眾可以很容易的使用本系統瀏覽由醫院所輸出的電子病歷資料，以及和其他系統的整合應用
  - 醫事人員：使用本系統檢視電子病歷資料的時候可以經由本系統匯出資料至該單位的其他系統中使用
  - 一般民眾：僅只有將電子病歷套用樣板檔的檢視功能

# 需求分析與系統架構 (2/2)

- 每個步驟中需要的基本功能以及處理的目標為下：
  - 系統框架
    - 必須可以將畫面分割顯示於不同的分頁之中
    - 能夠顯示頁碼，以及頁首與頁尾
    - 必須能夠提供檔案的讀取及寫出的能力
  - 檔案解析介面
    - 必須檢查檔案是否為CDA文件
    - 將CDA的Header與Body讀取出來，並加以區分
  - 能夠將樣板檔選取的XPath資料以元件的方式呈現
  - 能夠列舉XPath所選取的資料
  - 能夠有資料對應表來制作代碼區域對應的呈現
- 樣板檔管理
  - 必須自動的在找不到對應樣板檔的時候退回至該單張的預設樣板檔
  - 必須自動的在找不到對應的預設樣板檔時，使用系統範例樣板檔
  - 提供樣板檔上傳介面，讓設計者可以提供樣板檔給對應的電子病歷單張

# 系統分析與設計

- 系統中有六個系統模組來形成電子病歷檢視器
  - 樣板管理模組
  - 文件讀取及解析模組
  - 資料內嵌處理模組
  - 樣板輸出處理模組
  - 多國語系模組
  - 樣板檔編輯器模組

# 樣板管理模組

- 提供系統查詢與上傳系統或者使用者提供的樣板檔
- 使用md5 hash演算法作為資料識別ID，產生出的該字串將會用以下規則傳回適當的樣板檔以供使用：
  - 以單張代碼及醫療機構代碼查詢樣板檔
  - 若上述規則無樣板檔，則以單張代碼返回其使用的公用樣板檔
  - 若上述規則無樣板檔，則直接返回通用樣板檔
- 樣板檔中必須定義各樣版的名稱於根元素內，定義的規則為下：
  - 若為醫療機構客制的單張，則以此種方式命名：cda\_\_\_\_單張代碼\_\_\_\_醫療機構代碼
  - 若為單張公用樣板檔，則以此種方式命名：cda\_\_\_\_單張代碼
  - 若為通用樣板檔，則以此種方式命名：cdaDemo

# 文件讀取及解析模組

- 電子病歷文件需要透過使用者驅動瀏覽器讀取資料
- 使用Drag & Drop的方式讓使用者輸入檔案
- 接收到OnDrag Event之後系統就會將一個或多個檔案依序讀出來，並且檢查是否為系統可接受的檔案格式
- 以下規則來檢查是否為正確的文件格式：
  - 確認是否有XML的文件宣告
  - 檢查是否有ClinicalDocument的Root Element，並且有Element結尾
  - 檢查是否有xmlns宣告為urn:hl7-org:v3的schema
  - 其使用的RegExp檢查式為：/`<\?xml[^?>]+\?>\s*(<\?xml-stylesheet[^?\?]+\?>)?\s*<\?[^xml-][^?\?]+\?>\s*\}{0,}<([\w]*:)?ClinicalDocument[^>]+>([\s\w\W]**)<\/([\w]*:)?ClinicalDocument>([\s\w\W]*)/`
  - 但本系統不會根據宣告的schema去驗證XML是否符合schema的宣告

# 資料內嵌處理模組

- 將檔案區分為兩個部分：CDA Header與CDA Body
- 其區分的方式為下列XPath表示：
  - 取得CDA Header : \*:ClinicalDocument/\* except self::\*/\*:component)
  - 取得CDA Body : \*:ClinicalDocument/\*:component
- 分別以cdaHeader與cdaBody的元素物件包裝起來，放入cda2g這個元素物件中
- 使用Web Components的API處理在這個元素內的資料與讀取

# 樣板輸出處理模組

- 依照樣板檔的描述，輸出成最終的頁面結構，以及依照使用者在文件讀取階段所下的指令來決定是否要輸出樣板檔所描述相對應的輸出目標結構來產生XML文件輸出
- 畫面的顯示與輸出處理主要會以一個單張資料為一頁
- 若該單張的資料過多，則會在超過範圍的分頁處截斷至下一页處理，而被截斷的頁面屬性將會延伸到截斷後的新頁面上
- 電子病歷資料可以內嵌其他附件資料(例如：圖片、文件)，資料都會儲存於電子病歷內的**observationMedia**段

# 樣板輸出處理模組 cont.

- 使用HTML5的File API來產生檔案的轉換及下載輸出
- 檔案轉換輸出的時候會有以下流程：
  - Base64使用瀏覽器atob function轉換為binary string資料
  - 將binary string資料轉換回正確的binary資料
  - 使用Blob class將binary資料轉換成瀏覽器的blob資料
  - 建立blob URL
  - 輸出為圖片帶超連結或者文件資訊帶超連結

# 多國語系模組

- 系統會依照使用者的瀏覽器設定自動的設定到對應的介面與語言顯示
- 自動套用與設定字型
- 使用者在進行瀏覽的時候不會因為字型而造成辨識度不佳的狀況

# 樣板檔編輯器模組

- 將樣板檔以視覺化編輯
- 所見即得模式
- 簡單操作
- 僅需要瞭解XPath使用方式即可

# 相關技術及開發工具

- ✿ 本研究使用的開發環境、開發工具、開發程式語言如下：
  - ✿ 開發環境：Macintosh OS X Mountain Lion 10.8.3、Microsoft Windows 8 Pro
  - ✿ 開發工具：Sublime Text 2（純文字編輯器）、Google Chrome（瀏覽器開發環境）、SourceTree（git版本控管）、Mu（Markdown編輯器）
  - ✿ 開發語言：HTML 5、JavaScript、jQuery、jQuery UI、CSS、php（樣板檔上傳介面）、Markdown

# 研究結果與討論

# 研究結果

以系統中各個模組處理的過程與結果來分別敘述內容如下

# 樣板管理模組

- 模組對應的時候會經由md5編碼產生一段由單張代碼與機構代碼所組成的路徑，當查詢樣板檔時，如果沒有對應的樣板檔，則有下面兩種狀況處理：
  - 單張代碼有預設的樣板檔
  - 完全無法對應

```
curl --location --head --insecure \
--url https://cda2g.hp.af.cm/templates/9ccf3dfb3a08976b91b300a18ba6a0e7/8eed412f2ef2379272b620cfa8580ecb.xhtml
```

# 單張代碼有預設樣板檔的查詢命令

**HTTP/1.1 307 Temporary Redirect**

```
Server: nginx
Date: Thu, 30 May 2013 15:33:26 GMT
Content-Type: text/html; charset=iso-8859-1
Connection: keep-alive
Keep-Alive: timeout=900
Cache-Control: max-age=0, no-cache, no-store, must-revalidate
Pragma: no-cache
Expires: Wed, 11 Jan 1984 05:00:00 GMT
X-Template: 307
Location: https://cda2g.hp.af.cm/templates/9ccf3dfb3a08976b91b300a18ba6a0e7/8eed412f2ef2379272b620cfa8580ecb.xhtml/.../default.xhtml
Vary: Accept-Encoding
```

**HTTP/1.1 200 OK**

```
Server: nginx
Date: Thu, 30 May 2013 15:33:26 GMT
Content-Type: application/xhtml+xml
Content-Length: 11654
Connection: keep-alive
Keep-Alive: timeout=900
Cache-Control: max-age=0, no-cache, no-store, must-revalidate
Pragma: no-cache
Expires: Wed, 11 Jan 1984 05:00:00 GMT
X-Template: 307
Vary: X-FORWARDED-FOR
Last-Modified: Sat, 25 May 2013 10:50:30 GMT
Accept-Ranges: bytes
```

# 單張代碼有預設樣板檔的回應結果

```
curl --location --head --insecure \
--url https://cda2g.hp.af.cm/templates/078df904b021bea0ccf56f03ca7f971b/7a52d46f88c3001e4e2c1bf2732d86dc.xhtml
```

單張代碼沒有預設樣板檔的查詢命令

```
HTTP/1.1 307 Temporary Redirect
server: nginx
Date: Thu, 30 May 2013 15:35:44 GMT
Content-Type: text/html; charset=iso-8859-1
Connection: keep-alive
Keep-Alive: timeout=900
Cache-Control: max-age=0, no-cache, no-store, must-revalidate
Pragma: no-cache
Expires: Wed, 11 Jan 1984 05:00:00 GMT
X-Template: 307
Location: https://cda2g.hp.af.cm/templates/078df904b021bea0ccf56f03ca7f971b/7a52d46f88c3001e4e2clbf2732d86dc.xhtml/.../default.xhtml
Vary: Accept-Encoding

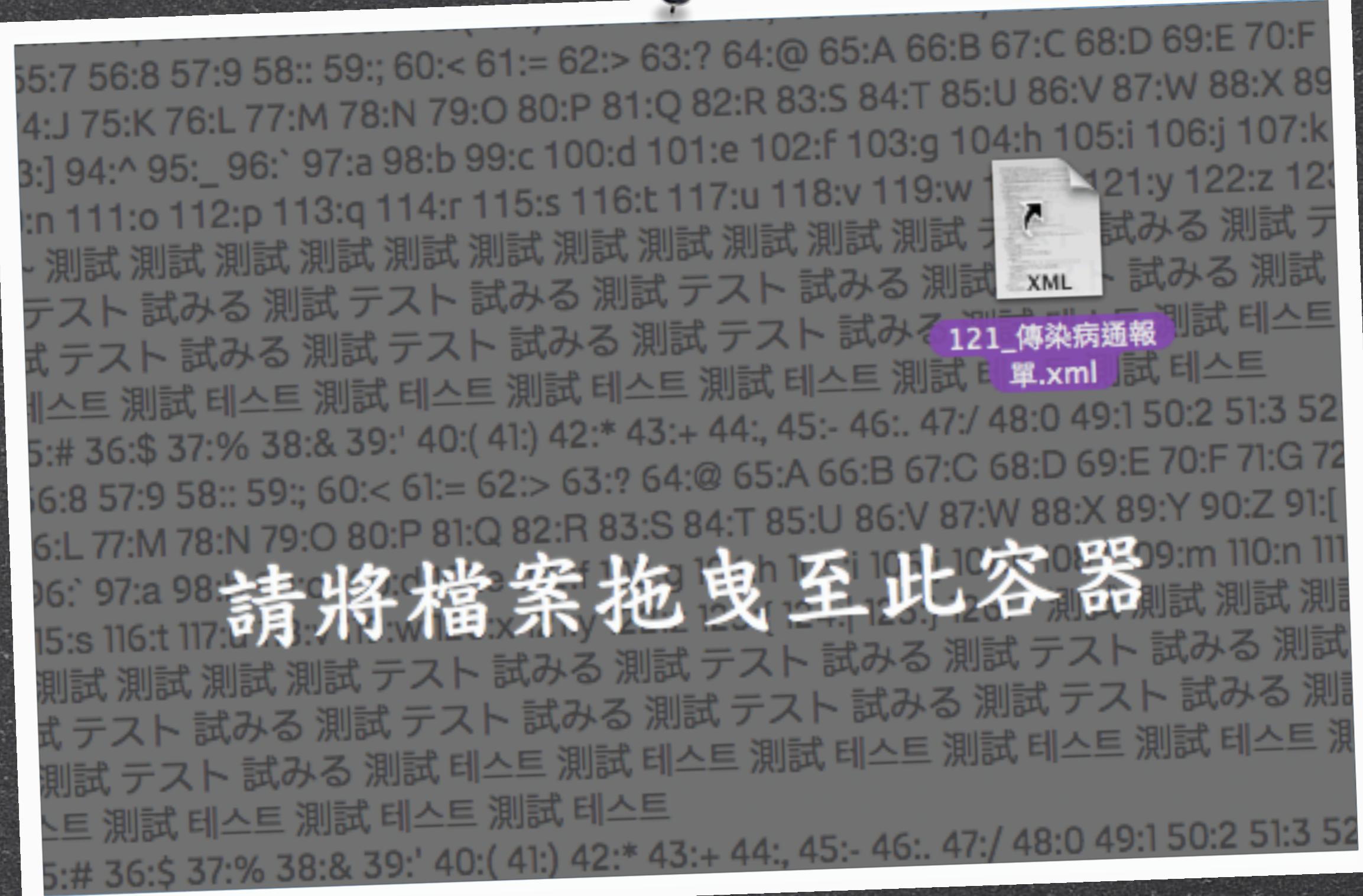
HTTP/1.1 303 See Other
server: nginx
Date: Thu, 30 May 2013 15:35:44 GMT
Content-Type: text/html; charset=iso-8859-1
Connection: keep-alive
Keep-Alive: timeout=900
Cache-Control: max-age=0, no-cache, no-store, must-revalidate
Pragma: no-cache
Expires: Wed, 11 Jan 1984 05:00:00 GMT
X-Template: 303
Location: https://cda2g.hp.af.cm/templates/078df904b021bea0ccf56f03ca7f971b/default.xhtml/.../components/cdaDemo.xhtml?template=078df904b021bea0ccf56f03ca7f971b/default.xhtml
Vary: Accept-Encoding

HTTP/1.1 200 OK
server: nginx
Date: Thu, 30 May 2013 15:35:45 GMT
Content-Type: application/xhtml+xml
Content-Length: 4562
Connection: keep-alive
Keep-Alive: timeout=900
Cache-Control: max-age=0, no-cache, no-store, must-revalidate
Pragma: no-cache
Expires: Wed, 11 Jan 1984 05:00:00 GMT
X-Template: 303
Last-Modified: Sat, 25 May 2013 10:50:39 GMT
Accept-Ranges: bytes
```

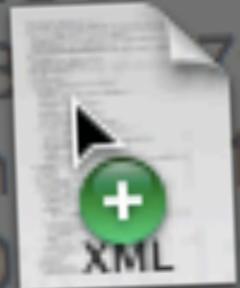
# 單張代碼沒有預設樣板檔的回應結果

# 文件讀取及解析模組

- 在觸發讀取文件的拖放事件的時候，系統會依據使用者附加按鍵會有下面兩種不同的回應鼠標
- 當使用者放入了不正確的文件的時候則會出現錯誤訊息



鼠標回應：未按附加按鍵

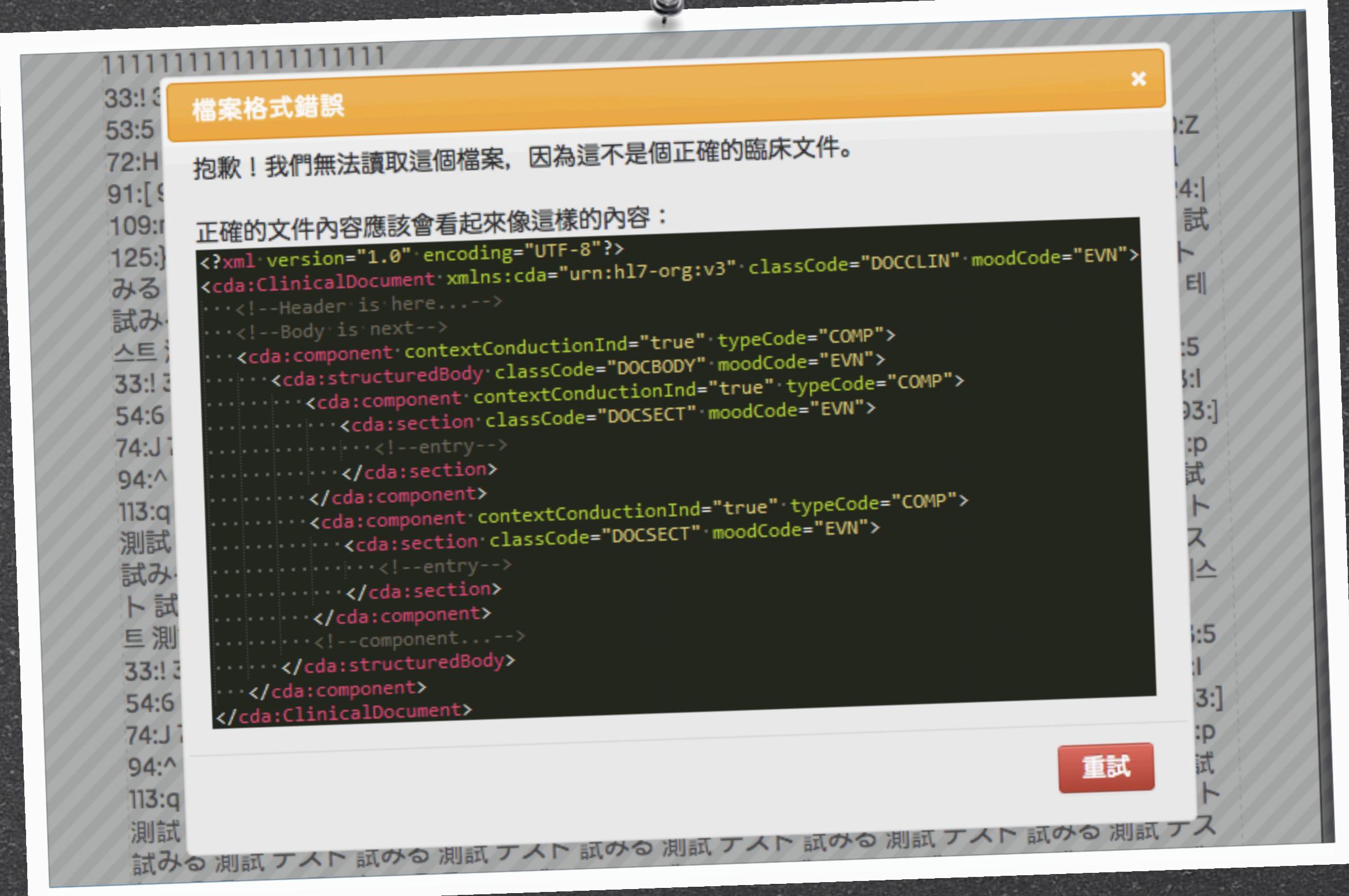


121\_傳染病通報

單.xml

請將檔案拖曳至此容器

鼠標回應：有按下附加按鍵



# 提示訊息：不正確的格式

# 資料內嵌處理模組

- 將資料內嵌至頁面中等待處理
- 內嵌的結構與CDA格式相同，但僅只有將CDA Header與CDA Body分開至兩個物件內包裝
- Web Components的描述結構皆在#document-fragment的物件內

## 檔案載入器初始化

載入檔案中

檔案載入即將完成...

載入檔案中 '121\_傳染病通報單.xml' (10759 bytes)...

```
307; 9ccf3dfb3a08976b91b300a18ba6a0e7/8eed412f2ef2379272b620cfa8580e  
cb.xhtml -> templates/9ccf3dfb3a08976b91b300a18ba6a0e7/default.xhtml
```

# 資料內嵌處理模組的系統畫面

nts Resources Network Sources Timeline Profiles Audits Console

```
dy>
section id="baseView" style="font-family: cdaFont, cdaFont-zh, 'Helvetica Neue', Helvetica, Arial, sans-serif;"></section>
section id="downloadURI" style="font-family: cdaFont, cdaFont-zh, 'Helvetica Neue', Helvetica, Arial, sans-serif;"></section>
section class="page-wrapper" style="font-family: cdaFont, cdaFont-zh, 'Helvetica Neue', Helvetica, Arial, sans-serif;">
<header>...</header>
<article class="page-viewpoint" style="height: 772px; top: 0px;">
  <section class="page-viewpoint">
    <header></header>
    <article class="pageBox">
      <section class="page-view">
        ><header class="outside">...</header>
        <article>
          <section class="cda-widget" style="position: relative;">
            <article>
              <parsing xmlns="http://www.w3.org/1999/xhtml" is="parsing">
                >#document-fragment
                <loading class="l10n">載入檔案中</loading>
                <filename>121_傳染病通報單.xml</filename>
                <filesize>10759</filesize>
              </parsing>
              <cda2g xmlns="http://www.w3.org/1999/xhtml" is="cda_51897-7" cda.filename="121_傳染病通報單.xml" style="display: block; overflow: hidden; height: 0px !important;" xmlnsPrefix="urn:hl7-org:v3">...</cda2g>
            </article>
          </section>
        </article>
        ><footer class="outside">...</footer>
      </section>
    </article>
  </section>
</article>
```

包含CDA資料內嵌的Web Components物件 -- cda2g

# 內嵌資料於頁面中

```

<cda2g xmlns="http://www.w3.org/1999/xhtml" is="cda__51897-7" cda.filename="121_傳染病通報單.xml" style="display: block; overflow: hidden; height: 0px !important;" xmlnsPrefix="urn:hl7-org:v3">
  #document-fragment
    <style scoped="scoped">...</style>
    <style scoped="scoped">...</style>
    <info filename="121_傳染病通報單.xml">...</info>
    <processable>...</processable>
    <div class="_XCD_Component" style="display: block;">
      <table>...</table>
      <br>...
      <table>...</table>
      <table>...</table>
    </div>
    <cdaHeader xmlns="http://www.w3.org/1999/xhtml">
      <typeId xmlns="urn:hl7-org:v3" root="2.16.840.1.113883.1.3" extension="POCD_HD000040">...</typeId>
      <templateId xmlns="urn:hl7-org:v3" extension="2.16.886.119.100118.10.121" root="2.16.886.119.100118.10.121">...</templateId>
      <id xmlns="urn:hl7-org:v3" extension="DOH_121" root="2.16.840.1.113883.19.4">...</id>
      <code xmlns="urn:hl7-org:v3" codeSystem="2.16.840.1.113883.6.1" codeSystemName="LOINC" code="51897-7" displayName="Healthcare Associated Infection Report">...</code>
      <title xmlns="urn:hl7-org:v3" value="法定及新興傳染病個案(含疑似病例)報告單">...</title>
      <effectiveTime xmlns="urn:hl7-org:v3" value="201207021704">...</effectiveTime>
      <confidentialityCode xmlns="urn:hl7-org:v3" code="N" displayName="normal" codeSystem="2.16.840.1.113883.5.25" codeSystemName="Confidentiality">...</confidentialityCode>
      <languageCode xmlns="urn:hl7-org:v3" code="zh-TW">...</languageCode>
      <recordTarget xmlns="urn:hl7-org:v3" typeCode="RCT" contextControlCode="OP">...</recordTarget>
      <author xmlns="urn:hl7-org:v3" typeCode="AUT" contextControlCode="OP">...</author>
      <custodian xmlns="urn:hl7-org:v3" typeCode="CST">...</custodian>
      <participant xmlns="urn:hl7-org:v3" contextControlCode="OP" typeCode="RCV">...</participant>
      <participant xmlns="urn:hl7-org:v3" contextControlCode="OP" typeCode="RCV">...</participant>
      <participant xmlns="urn:hl7-org:v3" typeCode="REFT">...</participant>
      <componentOf xmlns="urn:hl7-org:v3" typeCode="COMP">...</componentOf>
    </cdaHeader>
    <cdaBody xmlns="http://www.w3.org/1999/xhtml">
      <component xmlns="urn:hl7-org:v3" typeCode="COMP" contextConductionInd="true">
        <structuredBody classCode="DOCBODY" moodCode="EVN">
          <!-- 職業 **** History of occupation ****-->
          <component typeCode="COMP" contextConductionInd="true">...</component>
        <!-->
      </component>
    </cdaBody>
  </cda2g>

```

Web Components

CDA Header

CDA Body

# 內嵌CDA文件的Header與Body資料

# 樣板輸出處理模組

- 將Web Components的結果取出並取代原本內嵌的結構
- 處理使用者指定需要輸出成為樣板檔指定的XML格式文件



# 樣板處理與頁面構成模組的系統畫面



# 樣板處理與頁面構成模組的系統畫面

```
▼<疾病通報>
  <醫院名稱>臺大北護醫院</醫院名稱>
  <醫院代碼>0401190010</醫院代碼>
  <醫院電話>(02) 2371-7101</醫院電話>
  <診斷醫師>葉醫師</診斷醫師>
  <醫院地址>台北市萬華區康定路37號</醫院地址>

  ▼<病患>
    <姓名>張三</姓名>
    <性別>男</性別>
    <出生日>2000/02/11</出生日>
    <身分證字號>A123456789</身分證字號>
    <國籍>Mexican</國籍>
    <婚姻狀況>單身</婚姻狀況>

    ▼<電話>
      <公司電話>(02)8765-4321</公司電話>
      <家用電話>(02)1234-5678</家用電話>
      <行動電話>0933123456</行動電話>
    </電話>
    <地址>台北內江街89號</地址>
    <職業>教師</職業>
  </病患>

  ▼<疾病通報項目>
    <發病日期>2012/07/01</發病日期>
    <旅遊史>中南美探親，途經美國</旅遊史>
    <主要症狀>(文字敘述)</主要症狀>
    <相關疫苗接種史>(文字敘述)</相關疫苗接種史>
    <診斷日期>2012/07/02 01:11</診斷日期>
    <報告日期>2012/07/02 17:04</報告日期>

    ▼<收到日期>
      <衛生局>2012/07/03</衛生局>
      <疾管局>2012/07/05</疾管局>
    </收到日期>
    <備註>抗酸菌塗片、結核菌培養、聚合酶連鎖反應PCR、典型結核病理報告、胸部或其他X光檢查、自行檢驗結果等</備註>
  </疾病通報項目>

  ▼<通報項目>
    ▼<通報資料>
      ▼<採集檢體>
        <檢體名稱>Tuberculosis</檢體名稱>
        <是否採集>是</是否採集>
      </採集檢體>
    </通報資料>
  </通報項目>
</疾病通報>
```

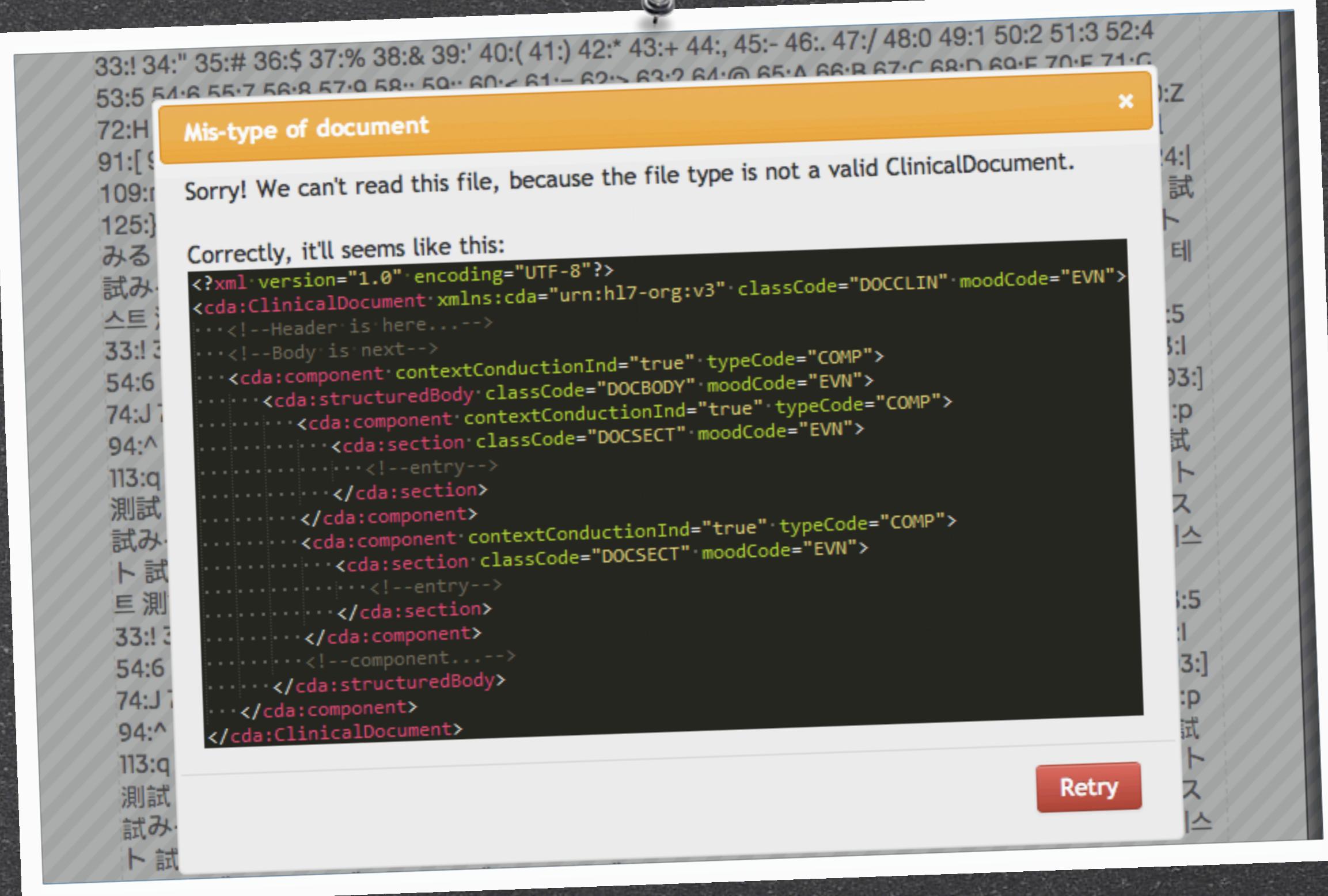
# 樣板檔輸出指定的XML格式文件範例

# 多國語系模組

- 依照使用者瀏覽器的語言不同，本系統會自動的調整顯示的語言



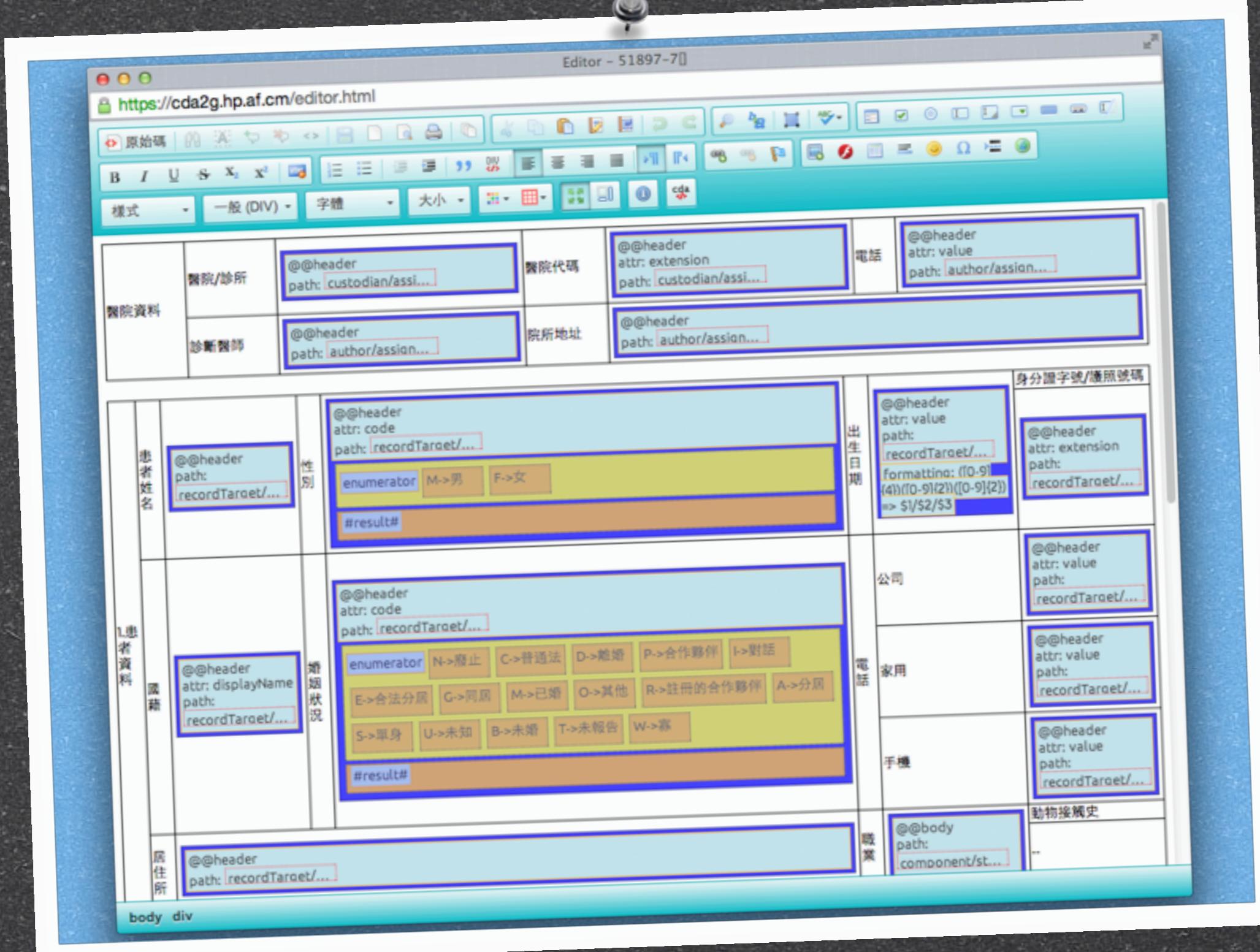
# 中文語系介面



# 英文語系界面

# 樣板檔編輯器模組

- 依照使用者需求會自動開啟編輯器



# 編輯器畫面

# 討論

本研究為了可以有更多的擴充性與靈活度，採用Web Application的方式寫作，並且結合MVC(Model View Controller)的設計方式，以及結合RWD理念，隨時可以提供不同的擴充功能

# 樣板檔編輯

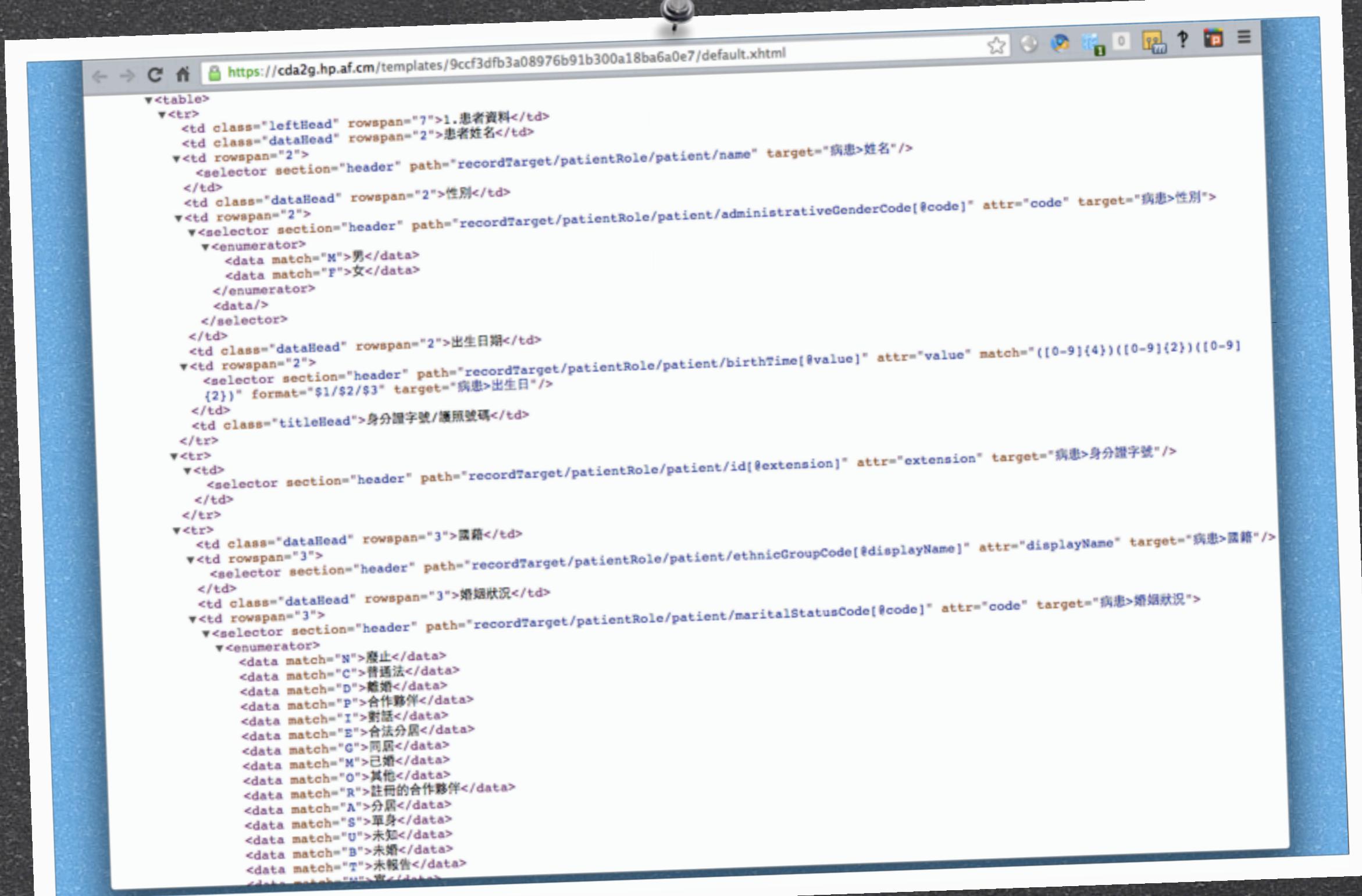
- 使用支援XHTML的編輯器編輯
- 將檔案中命名的屬性修改後即可變成專用樣板檔
- 僅需要有HTML的設計編寫能力即可自行客制化所需的電子病歷樣板檔

This XML file does not appear to have any style information associated with it. The document tree is shown below.

```
<element extends="cda2g" name="cda_51897-7" constructor="cdaDemo">
  <decorator>
    <template>
      <style scoped="scoped">
        table { border: solid 1px black; border-collapse: collapse; width: 100%; } table th, table td { border: solid 1px black; } th { font-size: 14pt; } td { font-size: 10pt; } margin: 0px; border-bottom: solid 1px #06F; } result { margin-top: 0px; } result - * { margin-top: 15px; } h4 { margin-left: 15px; } hr { margin: 5px 0px 5px 0px; } .data, .desH td { vertical-align: top; text-align: left !important; }
        .leftHead { text-align: center; width: 12px; } .dataHead { text-align: center; width: 34px; } .dataHead1 { text-align: center; width: 50px; }
        .dataHead2 { text-align: center; width: 75px; } .dataHead3 { text-align: center; width: 60px; } .titleHead { text-align: center; width: 150px; }
        .innerTB { border: none 0px black; vertical-align: top; } .innerTB table { border: none 0px black; margin: -1px -2px -1px -1px; }
        .desH { height: 105px; width: 678px; } .desH th { font-size: 10pt; font-weight: normal; height: 20px; }
      </style>
    </info>
    <header>
      <choose>
        <exist>
          <selector section="header" path="title"/>
        </exist>
        <otherwise>
          <selector section="header" path="code{@displayName}" attr="displayName"/>
        </otherwise>
      </choose>
    </header>
  </info>
  <processable>
    <green root="疾病通報" filename="121.green.xml"/>
  </processable>
  <div class="_XCD_Component" style="display: none;">
    <table>
      <tr>
        <td rowspan="2">醫院資料</td>
        <td>醫院/診所</td>
      <td>
        <selector section="header" path="custodian/assignedCustodian/representedCustodianOrganization/name" target="醫院名稱"/>
      </td>
      <td>醫院代碼</td>
      <td>
        <selector section="header" path="custodian/assignedCustodian/representedCustodianOrganization/id[@extension]" target="醫院代碼"/>
      </td>
      <td>電話</td>
      <td>
        <selector section="header" path="author/assignedAuthor/representedOrganization/telecom[@value]" attr="value" target="醫院電話"/>
      </td>
    </tr>
    <tr>
      <td>診斷醫師</td>
      <td>
        <selector section="header" path="author/assignedAuthor/assignedPerson/name" target="診斷醫師"/>
      </td>
    </tr>
  </table>

```

# 疾病通報單張原始樣板檔節錄1



The screenshot shows a web browser displaying an XML document at the URL <https://cda2g.hp.af.cm/templates/9ccf3dfb3a08976b91b300a18ba6a0e7/default.xhtml>. The XML code defines a table structure for a medical record form. The table has several rows and columns, with various sections like '患者資料' (Patient Information), '性別' (Gender), '出生日期' (Date of Birth), and '國籍' (Nationality). Each section includes header elements and selector elements for different data types like names, genders, dates, and nationalities. The XML uses XPath expressions to map fields to specific data elements.

```
<table>
  <tr>
    <td class="leftHead" rowspan="7">1. 患者資料</td>
    <td class="dataHead" rowspan="2">患者姓名</td>
    <td rowspan="2">
      <selector section="header" path="recordTarget/patientRole/patient/name" target="病患>姓名"/>
    </td>
    <td class="dataHead" rowspan="2">性別</td>
    <td rowspan="2">
      <selector section="header" path="recordTarget/patientRole/patient/administrativeGenderCode[@code]" attr="code" target="病患>性別">
        <enumerator>
          <data match="M">男</data>
          <data match="F">女</data>
        </enumerator>
        <data/>
      </selector>
    </td>
    <td class="dataHead" rowspan="2">出生日期</td>
    <td rowspan="2">
      <selector section="header" path="recordTarget/patientRole/patient/birthTime{@value}" attr="value" match="([0-9]{4})([0-9]{2})([0-9]{2})" format="$1/$2/$3" target="病患>出生日"/>
    </td>
    <td class="titleHead">身分證字號/護照號碼</td>
  </tr>
  <tr>
    <td class="dataHead" rowspan="3">國籍</td>
    <td rowspan="3">
      <selector section="header" path="recordTarget/patientRole/patient/id[@extension]" attr="extension" target="病患>身分證字號"/>
    </td>
  </tr>
  <tr>
    <td class="dataHead" rowspan="3">婚姻狀況</td>
    <td rowspan="3">
      <selector section="header" path="recordTarget/patientRole/patient/ethnicGroupCode[@displayName]" attr="displayName" target="病患>國籍"/>
    </td>
  </tr>
  <tr>
    <td class="dataHead" rowspan="3">配偶</td>
    <td rowspan="3">
      <selector section="header" path="recordTarget/patientRole/patient/maritalStatusCode[@code]" attr="code" target="病患>婚姻狀況">
        <enumerator>
          <data match="N">廢止</data>
          <data match="C">普通法</data>
          <data match="D">離婚</data>
          <data match="P">合作夥伴</data>
          <data match="I">對話</data>
          <data match="E">合法分居</data>
          <data match="G">同居</data>
          <data match="M">已婚</data>
          <data match="O">其他</data>
          <data match="R">註冊的合作夥伴</data>
          <data match="A">分居</data>
          <data match="S">單身</data>
          <data match="U">未知</data>
          <data match="B">未婚</data>
          <data match="T">未報告</data>
        </enumerator>
      </selector>
    </td>
  </tr>

```

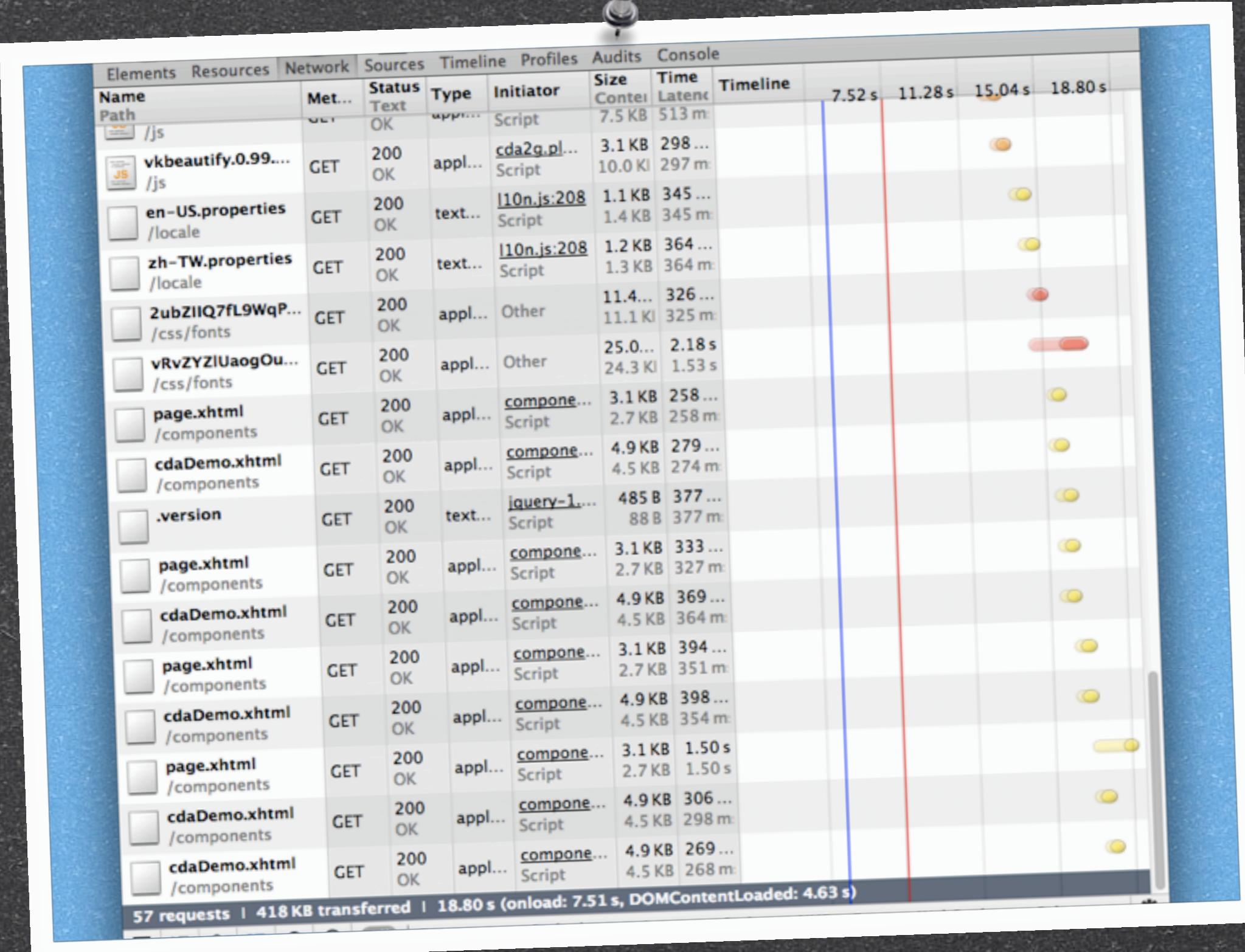
# 疾病通報單張原始樣板檔節錄2

```
1 <element extends="cda2g" name="cda_51897-7" constructor="cdaDemo">
2   <decorator>
3     <template>
4       <style scoped="scoped"><!--此文件專用的樣式表內容--&gt;&lt;/style&gt;
5       &lt;info&gt;
6         &lt;header&gt;&lt!--文件標題--&gt;&lt;/header&gt;
7         &lt;footer&gt;&lt!--文件結尾--&gt;&lt;/footer&gt;
8       &lt;/info&gt;
9       &lt;processable&gt;
10      &lt;green root="輸出XML根元素名稱" filename="輸出XML文件檔名"&gt;&lt;/green&gt;
11    &lt;/processable&gt;
12    &lt;div class="_XCD_Component" style="display: none;"&gt;&lt!--包裝輸出資料的元素--&gt;
13      &lt;selector section="使用header或body的資料" path="XPath選擇資料" target="輸出XML元素的位置"&gt;&lt;/selector&gt;
14      &lt;!--例如：&lt;selector section="header" path="recordTarget/patientRole/patient/name" target="病患&gt;姓名"&gt;&lt;/selector&gt;--&gt;
15      &lt;!--attr="使用CDA資料的屬性"--&gt;
16      &lt;selector section="header" path="recordTarget/patientRole/patient/administrativeGenderCode[@code]" attr="code" target="病患&gt;性別"&gt;
17        &lt;!--設定列舉資料--&gt;
18        &lt;enumerator&gt;&lt;!--&lt;data match="符合的字串"&gt;轉換輸出的資料&lt;/data&gt;--&gt;
19          &lt;data match="M"&gt;男&lt;/data&gt;&lt;data match="F"&gt;女&lt;/data&gt;
20        &lt;/enumerator&gt;
21        &lt;data&gt;&lt;!--selector 將資料輸出的位置(包含列舉中轉換的結果)--&gt;&lt;/data&gt;
22      &lt;/selector&gt;
23      &lt;!--設定資料格式化--&gt;
24      &lt;selector section="header" path="recordTarget/patientRole/patient/birthTime[@value]" attr="value" match="([0-9]{4})([0-9]{2})([0-9]{2})" format="$1/$2/$3" target="病患&gt;出生日"&gt;&lt;/selector&gt;
25      &lt;!--選取CDA Body的資料--&gt;
26    &lt;/div&gt;
27  &lt;/template&gt;
28&lt;/decorator&gt;
29&lt;/element&gt;</pre>
```

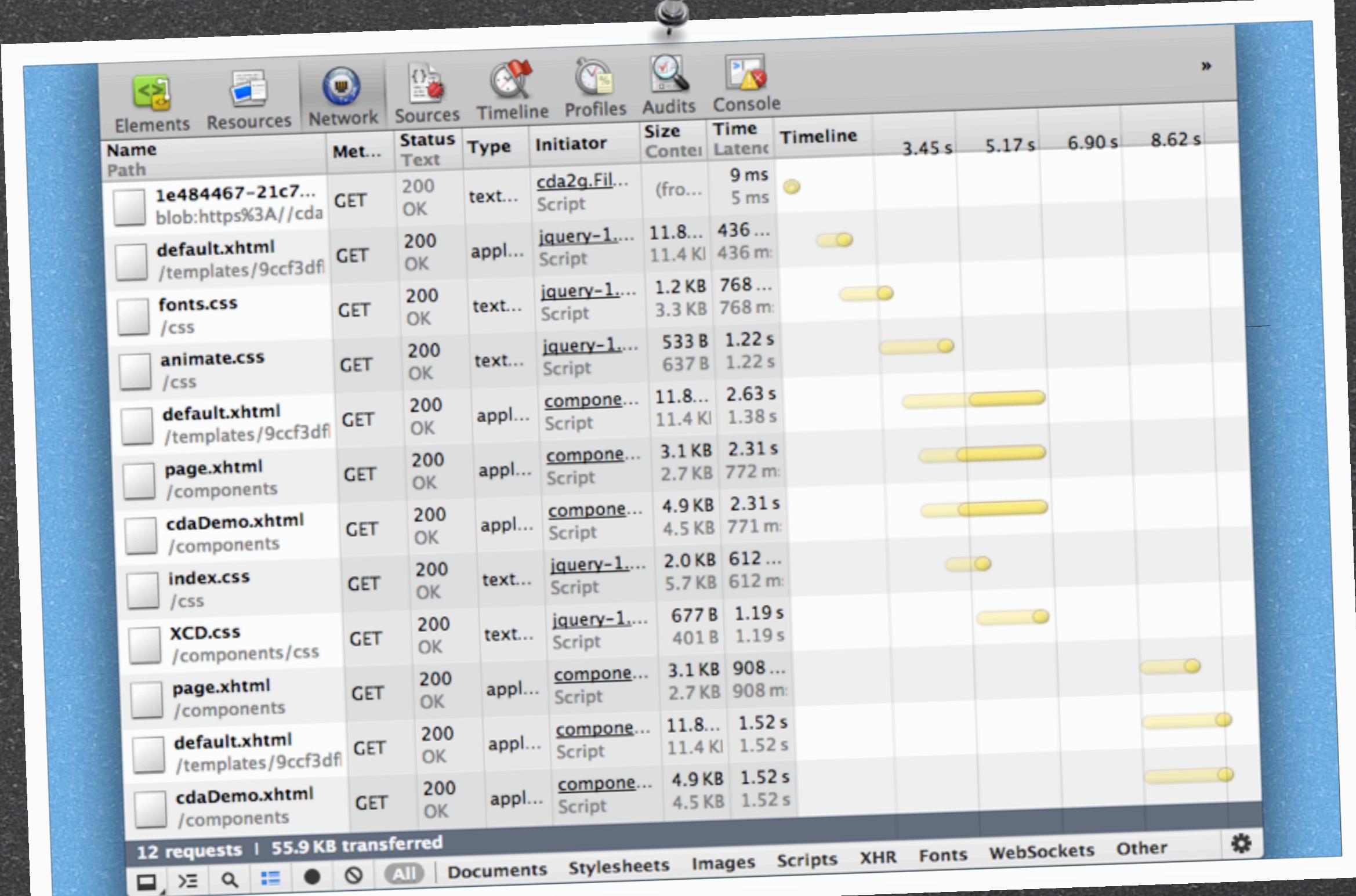
# 語法使用示意圖

# 檢視器存取能力

- 本研究之系統提供簡單的功能，令使用者僅需要簡單的動作即可套用樣板檔產生結果
- 有對應的樣板檔的時候，系統載入與套用樣板檔僅需要約不到30秒的時間即可完成一個單張的樣板的套用並顯示結果
- 一次載入多個單張，每多一張僅需要多3至5秒時間即可處理完成並且顯示結果



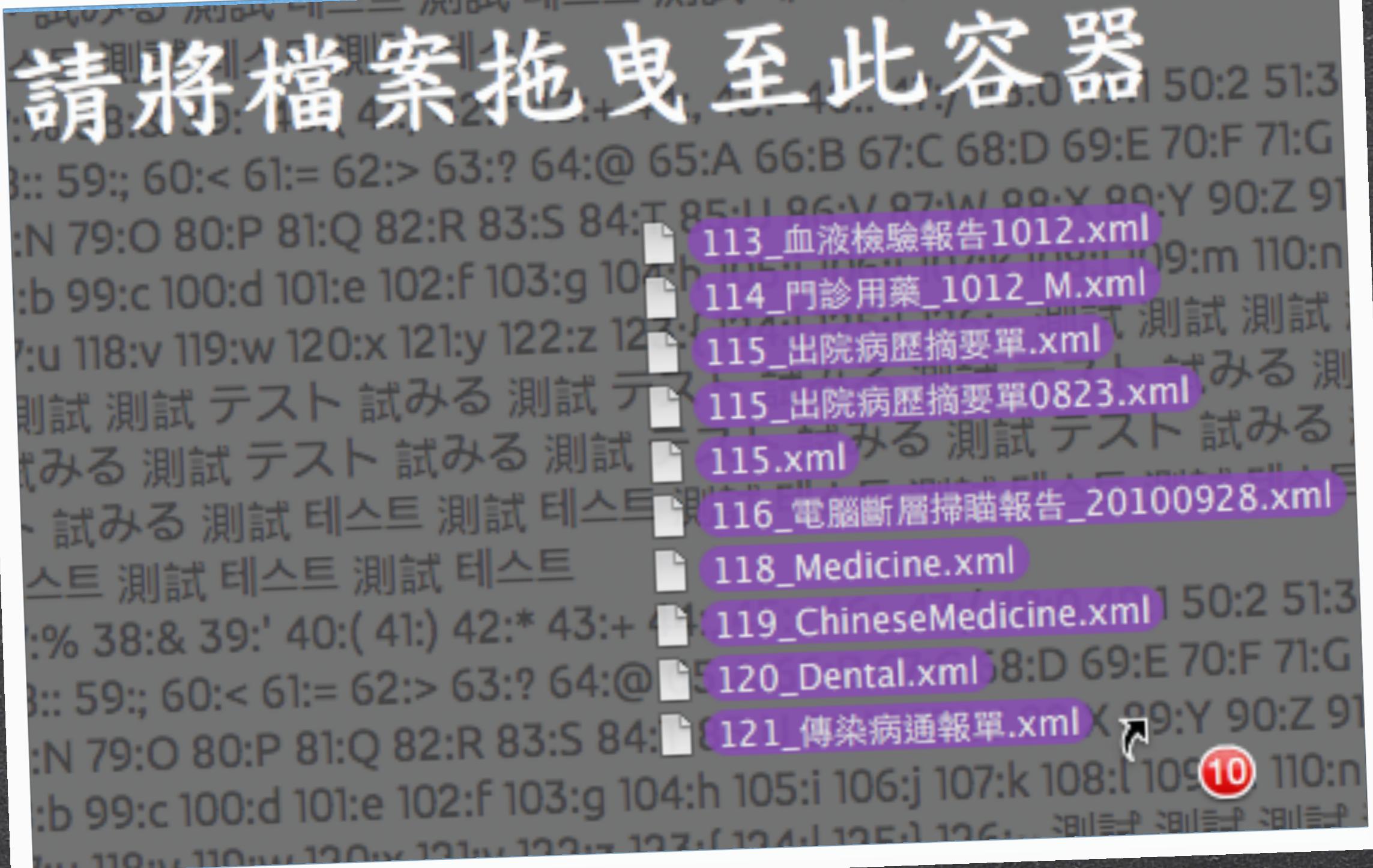
# 系統載入所耗用的時間



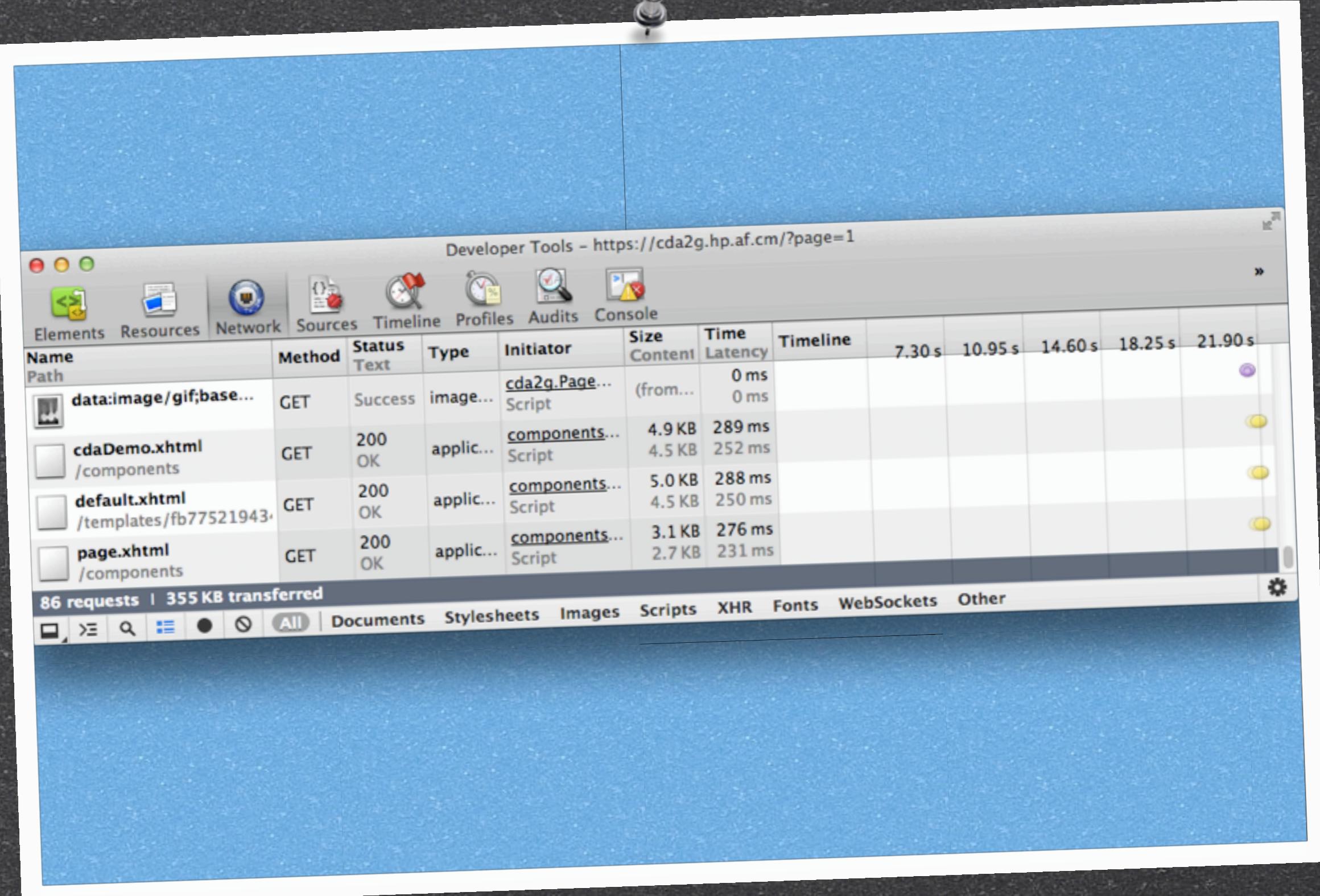
# 樣板檔套用的資料要求耗用時間

# 多個電子病歷檔案讀取與輸出能力

- 系統俱有批量處理樣板檔套用的能力
- 使用者一次拖拉超過一個檔案的時候會依照使用者的需求而批量處理資料



使用者拖曳多個電子病歷檔案示意圖



# 批量處理的耗用時間

# 研究限制

- 本研究的範圍，將會限制在系統的可行性研究上，會以主流的網頁應用程式設計的方式導入到電子病歷交換的層面，觀察能否應用於醫療領域
- 下列各項將不列入本研究範圍：
  - 電子病歷呈現用的樣板檔編輯器
  - 樣板擋存回電子病歷中增加可攜性之研究
  - 電子病歷的製作與交換
  - 電子病歷的格式與數位簽章之驗證及檢查
  - 資料安全與防止漏洞隱碼之研究
  - 雲端平台相關建置之研究
  - 國內其他非正式交換用的電子病歷規範之研究

# 結論與建議

# 結論

- 已經可以呈現行政院衛生署所公告的四類交換標準單張，以及疾病通報單張
- 操作方式讓使用者僅需要將電子病歷的檔案經由滑鼠拖曳拉入瀏覽器的畫面中即可開始解析文件資訊
- 可以匯出成樣板檔所設計的其他格式化之XML文件
- 能夠讓系統達到ITSM中的永續經營之目的，系統中提供使用者隨時可以執行資料維護的作業(包含新增、修改)
- 利用雲端運算平台的服務能力，提供使用者不間斷的服務
- 提供使用者直覺化的使用者操作經驗
- 節省掉電子病歷耗用在轉換解碼的程序及減少電子病歷表單在開發時所遇到的成本問題與教育訓練問題

# 未來努力方向

- 目前本系統中僅只有實作單張樣板之對應，目前並無結合代碼轉換之功能
  - 短期內尚可與其它研究醫療詞彙代碼相關的研究整合，成為一套完整的雲端服務平台
- 在結合醫學詞彙代碼資料庫之後
  - 長期來看就能夠提供醫療院所更完整的電子病歷開發環境

Thanks for your attention!!