

# 追溯性矩陣在 DoDAF 架構產品的驗證研究

## The Study of Validation in DoDAF Products by Traceability Matrix

曾淼泓

國防大學理工學院  
國科所 博士候選人  
tmh5735@mail2000.com.tw

陳子江

國防大學理工學院  
電機電子系教授  
johnrita514@gmail.com

韓孟麒

德明財經科技大學  
資訊科技系 副教授  
harn@takming.edu.tw

楊家宏

國防大學理工學院  
電機電子系助理教授  
yang.jiahorng@gmail.com

### 摘要

在 DoDAF 的作戰、系統、技術三個觀點的產品中，相互間存在著對應關係。當作戰需求變動或更改其作戰觀點中的某項內容，在對應關係的機制中，系統觀點與技術觀點的某產品內容，將隨著變動。本文透過系統分析法及正型方法，對實踐 DoDAF 各產品活動中的具體情況，進行歸納與分析，並將該結論提出一有效率的 59×59 追溯性矩陣，以獲得各產品關連性之追溯性解。其目的在於讓企業架構師在操作 DoDAF 的過程中，能在更動某項規格內容之後，知道必需連帶更動何項對應產品的規格內容。為了驗證此 59×59 追溯性矩陣的正確性，我們將事前並沒有規劃的「無人駕駛飛機」(UAV)這個戰場物件加入到原有的系統中。

關鍵詞：DoDAF、正型方法、企業架構師、追溯性矩陣。

### Abstract

There exists relationship among the DoDAF products in the three views: operation, system and technology. While operational requirements are changed or their product contents are modified, the related DoDAF product of the systematic and technological view will be changed. This paper concludes and analyzes the substantial situation about coping with the DoDAF product activities via the system analysis method and formal method. We have proposed a 59×59 traceability matrix to obtain the solution to the product traceability. The purpose of this paper is to understand which component is changed and what specification content is related after the architect updates certain specification in a component during the process of the DoDAF product manipulation. We verify the correctness of the 59×59 traceability matrix by adding a battlefield object UAD that is not planned in advance to the original system.

Keywords: DoDAF, formal method, architect, traceability matrix.

### 一、緒論

由於 DoDAF 的各產品間具有關連性 (Dependency)，當企業架構師 (Enterprise Architect) 完成了所屬專案的 DoDAF 產品設計後，往往會因為甲方企業模式 (Business Model) 的改變，而導致要修改部分的 DoDAF 產品規格。在多變的環境裏，使用者需求 (User Requirements) 改變的現象相當頻繁，而且是一種常態，這種現象與常態，迫使企業架構師所設計系統必須去適應環境的變化。在修改的過程中，各個 DoDAF 觀點中的元件會受到連動，也會讓企業架構師，搞不清楚哪些觀點的相關資料元素會被修改。另由於 DoDAF 的 26 項產品間具有一致性與完整性，所以企業架構師在製作 DoDAF 產品時，會與甲方不斷地溝通，不斷地訪談，並來回多次的編輯 DoDAF 產品。在修改的過程中，常使各個觀點中的相關資料元素產生不一致與不完整的現象。

我們透過系統分析法 (System Analysis Method) 及正型方法 (Formal Method)，在 DoDAF 各項產品製作的活動裏，進行歸納與分析，使製作過程系統化及理論化；我們將研究成果，著重在將 DoDAF 資料庫中的資料元素，隨時保持一致與完整。

當企業架構師面對 DoDAF 某項產品的需求與規格須要變動後，就必須追溯 26 項產品的關連性，這種追溯關係隱含著：作戰觀點 (OV)、系統觀點 (SV) 與技術觀點 (TV) 各觀點內部產品間的橫向追溯；作戰觀點 (OV)、系統觀點 (SV) 與技術觀點 (TV) 跨觀點產品間的縱向追溯及跨產品間具遞移性的間接追溯。我們惟有詳盡檢討，並發展一套追溯規則，才能讓後續之建構與維護工作得以持續進行；因此，我們必須透過各資料元素間的對應關係，來檢討各項產品間的追溯關係，並發展出追溯性矩陣來解決相關資料元素間不一致與不完整的問題。本論文聚焦在找到需更動的資料元素後，對 DoDAF 後台資料庫的各資料元素作出驗證 (Validation) 的工作。

依照本研究所作出來的驗證工作，驗證出於[1]先前研究之 59×59 追溯性矩陣之縱向追溯、橫向追溯、間接追溯、直接追溯的實用性，而獲得必須配合修改之產品，再修改此產品的規格及內容。將為爾後的 DoDAF 相關研究人員，對 DoDAF 後台資料庫的維護，提供一個效益強大的輔助工具。

## 二、文獻探討

### 2.1 追溯方向性

Soonsongtanee 等學者，對追溯方向性的討論提到，它涵蓋著橫向(Horizontal)、縱向(Vertical)、雙向(Bidirectional)及間接(Indirect)等追溯方向性，各種型態的內容敘述如下[2]：

#### (1) 縱向的追溯性

縱向的追溯性確認了該元素(Items)的來源(如客戶需求)，且遵循該元素的層級(Hierarchy)結構，自顧客需求開始，一直到專案團隊的探討上；縱向的追溯性保證需求遵守著原始構想，其中雖經過多次變化，而達至最後的型態；縱向關係如：需求的探討、需求與專案計畫間、專案計畫與產品間、產品與產品組件間的關係。

#### (2) 橫向的追溯性

橫向的追溯性確認著工作或產品間的關係，其目的在避免層級上的衝突；它促成了專案的實踐，減少了整合測試上的潛在問題，橫向關係例如：需求間、產品組件間、功能與功能間的關係。

#### (3) 雙向的追溯性

雙向的追溯性被定義為二或更多邏輯實體(Logical Entities)的組合，且在任一方向都是可識別的。在一軟體專案內，雙向的追溯性可有效的管理原始需求間、產品間及各組件間的關係；雙向的追溯性使用向前(Forward)及向後(Backward)的雙向追溯(如：從需求到最終產品及從最終產品到需求的追溯)。

#### (4) 間接的追溯性

追溯性是有方向性(Directions)的，向前追溯其起始點是需求，向後追溯其目標亦是需求。間接關係常發生在兩個邏輯實體下的變動，例如：A 產生變動影響到 B，B 產生變動影響到 C，則 A 產生變動會間接影響到 C，此種間接關係在需求管理(Requirements Management)過程是必需要注意的；以上各解釋如圖 1 所示。

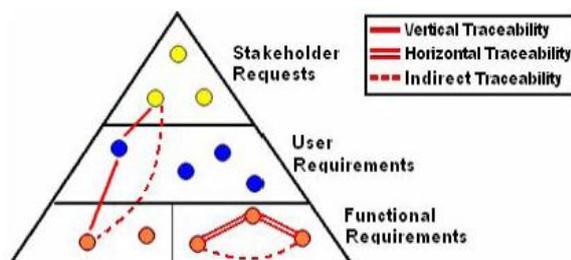


圖 1 縱向、橫向及間接追溯[2]

### 2.2 追溯性工具探討

Xie 等學者[3]提到，2001 年 Standish Group's 所發佈的一個 CHAOS Report 中描述到，公司專案(Projects)的無法準時驗收，有 74% 是歸因於需求的改變而失敗的；換句話說，這些失敗專案的致命傷，皆發生在缺乏需求管理，所以需求管理對一專案來說是重要的；尤其是對一包含多項需求的複雜系統專案。

早期對需求管理上的工作，都透過 Word 或 Excel 來完成，先選擇每一需求品項(Items)的屬性(Attribute)，再在需求間建立起相對關係；明顯的，此方法太依靠人為作業了，也很容易造成錯誤；需求管理的工具如 RequisitePro[4]及 DOORS[5]，現已被廣泛的使用。

對於使用 RequisitePro 這需求管理工具來管理火車控制系統(Train Control System)的需求，在研究過程，Xie 將需求管理的方法可分為四步驟，包括：分類(Categorize)、標籤化(Tag)、產生追溯性矩陣(Generate Traceability Matrix)及建立 UML 內各元素(Elements)的相互關係(Correlation)等程序。

第一步驟是分類，目的在使需求的各分類，能與 UML 模式中的各元素能建立相互關係，並使後續建立 UML 模式的規則及關係時，更簡單。從分類到規則都已建立了，他們再使用所定的規則去作第二步驟，對全部的需求文件標籤化；此步驟有兩個目的，第一是貫穿整個文件，讓所有的資訊皆可以辨識(Distinguish)，避免錯誤的刪除；第二是確保在後續的塑模(Modeling)中，不會遺漏或誤加入資訊；換句話說，需求的追溯性已經建立，並可輕易的管理全部文件。

當需求文件的標籤化動作完成之後，第三步驟的屬性矩陣與追溯性矩陣將被建立完成。

(1) 屬性矩陣：它顯示需求上的全部屬性，它可協助需求者構建出最佳化需求(Optimal Requirements)。

(2) 追溯性矩陣：它顯示需求上的一或兩個型態(Types)間的關係，協助需求者在需求改變時，達致鏈接反應(Chain Reaction)的關係管理。

第四步驟是建立與 UML 模式內各元素的關係；過去使用自然語言(Natural Language)描述的需求文件，可能都存在著無法避免的模糊(Ambiguity)關係，現在使用統一塑模語言(UML)來操作，俾明確化需求管理。

使用 RequisitePro 這需求管理工具，在標籤化需求管理工作上最傑出的效果，在於可以直接接觸到 UML 各模式中的元素；所以在 UML 模式與需求文件兩者間，確認了資料的一致性。此種一致性是後續工作的基礎，如果無法維持其一致性，則塑模與後續的工作，都將失去系統需求的原來特性。

在 Xie 等學者所操作的火車控制系統的案例中，透過 RequisitePro 這工具，使得專案內的需求文件不只紀錄到所有的系統功能，還可以清楚的規

定出全部的安全要求。他們並說明，會選擇 RequisitePro 當工具，來作出需求管理，是因為它與 UML 模式有一直接的界面；除了保證 UML 模式可完全反映需求品項外，更重要的，它在模式解決(Model Solution)上，提前獲得了整個需求狀況的一致性。

現存的追溯性工具無法提供充分的支援，在整合到現有軟體生產的工具，雖然還在持續發展，但是這些工具仍是無法滿足，軟體生產線(Software Production Line, PPL)的發展。

遑提現有軟體追溯工具之優點，在目前軟體生產所採用的方法及技術，皆無法在各軟體產品間的定義及運用技術過程中，建立其追溯性的機制。換句話說，目前各追溯性工具皆無法在軟體生產線上，提供軟體新產品，及發展過程間，產品的追溯性資料；其結果導致整個軟體發展，各互動現象及變數的特性皆無從納入明確的追溯性描述及管理。在歐洲多國大學參與研究的 AMPLE 計畫中[6]，已對現有追溯性工具在軟體生產線發展過程中的追溯表現，作出調查評估。

各工具使用以下方式作出評估：1.追溯連接管理(Management of Traceability Links) 2.追溯查詢(Traceability Queries) 3.展現方式(Views) 4.擴充性(Extensible)，如表 1 所示。其調查結果，顯現出沒有一個工具可內植(Built-in)到軟體內，協助其發展；各工具的主要功能，卻常因無法解決所冒出的各種需求問題，而被迫關掉其使用功能。

表 1 三種需求追溯工具之綜合比較表[6]

工具軟體	RequisitePro	Caliber RM	DOORS
發行公司	IBM	Borland	IBM Rational
連接管理	各需求間	貫穿全壽期	貫穿全壽期
支援方式	手動	手動	手動及資料注入方式
追溯查詢	透過需求屬性完成	透過需求定義來連接完成	透過資料連接來完成
展現方式	追溯性矩陣、追溯樹	追溯性矩陣、追溯示意圖、報表	追溯性矩陣、追溯樹
擴充性	可以 ODBC 方式與外部檔案運作	可以 ODBC 方式與外部檔案運作	可自行創造新型態作出連接 可以 ODBC 方式與外部超過 25 種設計、分析、構型管理的工具相容

表 1 綜整各主要軟體追溯工具，自管理的角度而論，以上工具皆是手動(Manual)，但卻能以 MS-Word, Excel, ASCII and RTF 等檔案方式來執行。Caliber RM 及 DOORS 可在不同類型的軟體間創造追溯連接，RequisitePro 則將焦點放在各需求定義上的追溯連接。無論如何，這些外部工具(如 DOORS 及 RequisitePro)在傳統的系統中，都無法在軟體生產線上，作出具體而明確的追溯性描述。例如各變數的追溯性描述，亦無法在軟體發展過程

中，需求及特性的衝突分析(Impact Analysis)下，提前提供特定的支援。

RequisitePro 可透過需求分割(Filtering on Requirements)來提供查詢功能，Caliber RM 及 DOORS 允許查詢需求及追溯連接；Caliber RM 可偵知各軟體產品連接上的不一致(Inconsistency)，DOORS 可偵測孤立碼(Orphan Code)的衝突分析再產生報表；各追溯工具在展現方式上有：追溯樹(Traceability Tree)、追溯示意圖(Traceability Graphical Diagram)、追溯性矩陣等，所有功能的目的，在協助了解各軟體產品，相互間的追溯性連接。

本研究經數度的操作後，選擇了追溯性矩陣，來遂行 DoDAF 各產品的追溯性管理。

### 三、開放性問題探討

對 DoDAF 產品的探討，具有以下幾點目前存在的開放性問題：

#### 1. DoDAF 的產品具有關連性

企業架構師完成了所屬專案的 DoDAF 產品設計後，往往會因為需求方(甲方)企業模式(Business Model)的改變，導致要修改部分的 DoDAF 產品。企業模式改變的原因很多，根據美國海軍研究院(Naval Postgraduate School, NPS) Dr. Luqi 的研究，可歸納為政治環境、經濟、法律、國防…等環境因素。在多變的環境裏，需求的改變相當頻繁，而且是一種常態，這種現象與常態，迫使所設計系統可以去適應環境的變化。在修正的過程中，各觀點中的元件(Component)會受到連動，也會讓企業架構師，搞不清楚哪些觀點的相關元件需被修正[7]。

#### 2. DoDAF 的產品具有一致性與完整性(Complete)

企業架構師在製作 DoDAF 產品時，與需求方(甲方)不斷地溝通及訪談，來回多次的修訂，常使各個相關觀點中的產品產生不一致(Inconsistency)與不完整(Incompleteness)的現象；因此，企業架構師會不斷的根據使用者需求(User Requirement)與規格(Specifications)，作產品的確認(Validation)及驗證(Verification)工作。在過去探討一致性及完整性的過程，常發生無法驗證各產品的正確性(Correctness)問題。

#### 3. DoDAF 產品在邏輯與概念上的修正落差。

在目前市面上使用之軟體，如：IBM 的 SA(System Architecture)、Rhapsody 與 Sparx System 的 EA 等工具，在實體設計上，對追溯關係已作出初步之功能。它們允許設計人員直接在實體設計上作出增加、刪除、修改之動作，因為 DoDAF 產品之相依性大，在 OV-2 直接作出刪除某作戰節點(組織)之動作，系統或允許直接之暴力刪除。惟此作戰節點(組織)

的刪除，在該產品的實體系統檢討似是無誤，惟其卻影響到 OV-4 的組織關係圖、OV-6c 的循序圖，並使 OV-4 及 OV-6c 失卻意義。只有透過人工之檢核，才能了解問題之嚴重性，並在邏輯概念上作出修正。

因此，本研究透過製作 DoDAF 各產品之過程及經驗，定義出各產品資料元素(Data Elements, DE)，進而從 DoDAF 各產品結構最底層的資料元素去找出各產品之關連性(Dependency)，再創造出一個追溯性矩陣，並對此追溯性矩陣的正確性作出驗證工作，俾貢獻予企業架構師，作出需求管理(Requirement Managements)，避免前後需求規格落差的錯誤控管。

#### 四、追溯方向性的各項定義

承續以上之描述，本研究對追溯方向性的各項定義，如：縱向追溯(Vertical Traceability)、橫向追溯(Horizontal Traceability)、間接追溯(Indirect Traceability)、直接追溯(Direct Traceability)等內容敘述如下：

##### Definition 1 (Traceability)

Let  $V1$  be a set of *data element of view 1*.  
Let  $V2$  be a set of *data element of view 2*.  
Let  $x, y$  be a *data element*.  
There exists  $x$  and  $y$  such that  
 $x \in V1$  and  $y \in V2$   
or  $x \in V1$  and  $y \in V1$ ,  
or  $x \in V2$  and  $y \in V1$ ,  
or  $x \in V2$  and  $y \in V2$ ,  
where  $V1 \neq V2$   
if and only if  $T[x, y]$  is called *Traceability*.

##### Definition 2 (Vertical Traceability)

Let  $V1$  be a set of *data element of view 1*.  
Let  $V2$  be a set of *data element of view 2*.  
Let  $V3$  be a set of *data element of view 3*.  
Let  $x$  and  $y$  be a *data element*.  
There exists  $V1, V2, V3, x$  and  $y$  such that  $x \in V1$  and  $y \in V2 \cup V3$  or  $x \in V2$  and  $y \in V1 \cup V3$  or  $x \in V3$  and  $y \in V2 \cup V1$  where  $V1 \neq V2 \neq V3$  if and only if  $V[x, y]$  is called *vertical traceability*.

##### 【定義說明】

縱向追溯(Vertical Traceability)是在定義 DoDAF 各觀點(View)資料元素間縱向的追溯性，假設  $V1$  為觀點一資料元素的集合， $V2$  為觀點二資料元素的集合， $V3$  為觀點三資料元素的集合； $x, y$  皆為資料元素，若  $x$  屬於  $V1$  則  $y$  屬於  $V2$  或  $V3$ ，若  $x$  屬於  $V2$  則  $y$  屬於  $V1$  或  $V3$ ，若  $x$  屬於  $V3$  則  $y$  屬於  $V2$  或  $V1$ ；若且為若  $V[x, y]$  我們稱之為縱向追溯。

##### Definition 3 (Horizontal Traceability)

Let  $V1$  be a set of *data element of view 1*.

Let  $V2$  be a set of *data element of view 2*.

Let  $V3$  be a set of *data element of view 3*.

Let  $x$  and  $y$  be a *data element*.

There exist  $V1, V2, V3, x$  and  $y$ , such that  $x \in V1$  and  $y \in V1$  or

$x \in V2$  and  $y \in V2$  or  $x \in V3$  and  $y \in V3$ , where  $V1 \neq V2 \neq V3$

if and only if  $H[x, y]$  is called *Horizontal Traceability*.

##### 【定義說明】

橫向追溯(Horizontal Traceability)是在定義 DoDAF 各觀點資料元素間橫向的追溯性，假設  $V1$  為觀點一資料元素的集合， $V2$  為觀點二資料元素的集合， $V3$  為觀點三資料元素的集合； $x, y$  皆為資料元素，若  $x$  屬於  $V1$  則  $y$  屬於  $V1$ ，若  $x$  屬於  $V2$  則  $y$  屬於  $V2$ ，若  $x$  屬於  $V3$  則  $y$  屬於  $V3$ ，若且為若  $H[x, y]$  我們稱之為橫向追溯。

##### Definition 4 (Indirect Traceability)

Let  $x, y$  and  $z$  be a *data element*.

Let  $V[x, y]$  notes the *vertical traceability* between  $x$  and  $y$ .

Let  $H[x, y]$  notes the *horizontal traceability* between  $x$  and  $y$ .

There exist  $x, y$  and  $z$  such that

$H[x, y]$  and  $H[y, z]$  or

$V[x, y]$  and  $V[y, z]$  or

$H[x, y]$  and  $V[y, z]$  or

$V[x, y]$  and  $H[y, z]$

if and only if  $I[x, z]$  is called *indirect traceability*.

##### 【定義說明】

間接追溯是在定義 DoDAF 各觀點資料元素間透過第三個資料元素而產生的追溯性，假設  $V1$  為觀點一資料元素的集合， $V2$  為觀點二資料元素的集合， $V3$  為觀點三資料元素的集合； $x, y$  皆為資料元素，若  $H[x, y]$  則  $H[y, z]$ ，若  $V[x, y]$  則  $V[y, z]$ ，若  $H[x, y]$  則  $V[y, z]$ ，若  $V[x, y]$  則  $H[y, z]$ ，在合乎以上四條件的任一條件下，若且為若  $I[x, z]$  我們稱之為間接追溯。

##### Definition 5 (Direct Traceability)

Let  $x, y$  be a *data element*.

Let  $V[x, y]$  notes the *vertical traceability* between  $x$  and  $y$ .

Let  $H[x, y]$  notes the *horizontal traceability* between  $x$  and  $y$ .

There exist  $x$  and  $y$  such that  $V[x, y]$  or  $H[x, y]$

if and only if  $D[x, y]$  is called *direct traceability*.

##### 【定義說明】

直接追溯(Direct Traceability)是在定義 DoDAF 各觀點資料元素間未透過第三個資料元素而產生的直接追溯，假設  $x, y$  皆為資料元素，存在著  $H[x, y]$  或  $V[x, y]$ ，若且為若  $D[x, y]$  我們稱之為直接追

溯。

DoDAF 的產品製造過程中，將會發生以上定義的橫向追溯、縱向追溯、直接追溯、間接追溯等情形，以上之定義，如圖 2 所示。

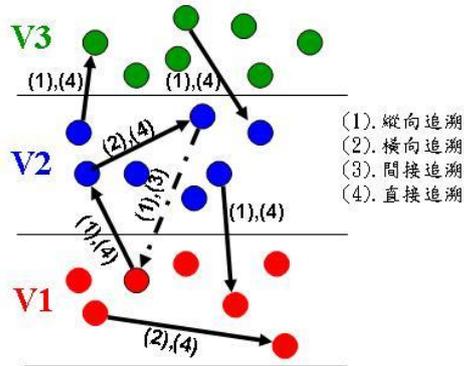


圖 2 各觀點、產品之間的追溯方向性

### 五、解決方法

我們針對「反溯河突擊」C4ISR 系統，因另規劃「無人駕駛飛機」戰場物件，更動了 OV-3[InformationExchange]展開追溯過程，一直到佇列 Q = null，所以追溯動作停止。資料元素集合 A 內的資料元素，A = (OV-3[InformationExchange] OV-5[MechanismFlowConnector] OV-6c[Event] OV-7[AttributeType] OV-7[EntityType] SV-6[SysDataExchange] SV-4[IoSysDataFlow] SV-11[SysAttributeType] SV-11[SysEntityType]) 即為要更動改的資料元素。

在資料元素內容修正前，我們先簡述本研究內之無人駕駛飛機(UAV)之特性：

- (1) 已加裝超高頻 UHF(Ultra High Frequency, UHF) 視頻數據網路。
- (2) 在全球衛星定位系統(Global Positioning System, GPS)的引導下，可飛至目標區自動搜索與識別目標。
- (3) 所獲目標影像透過數據網路可直接回傳到指揮中心。
- (4) 操作人員可直接透過回傳之數據影像，於確認目標後，再指揮 UAV 進行攻擊。

在以上之條件下，現針對集合 A 的九個資料元素，做出內容修正，以確認此過程的正確性，修正過程如下：

- (1) OV-3[InformationExchange]的內容變更(如表 2 所示)

海巡署岸際雷達為目前(As-is)之情況，其所後傳之情報資訊有帶著「船隻編號」及「船名」；而更易為以 UAV 作偵蒐工具後，UAV 以其直接即時影像後傳之特性，已無時間再賦予「船隻編號」及「船名」。

表 2 OV-3 的內容變更情形

	field 1	field 2	field 3	field 4	field 5	field 6	field 7	field 8
後傳情報資訊 (海巡署岸際雷達)	編號	距離	方位角	航向	航速	威脅模式	船隻編號	船名
後傳情報資訊 (UAV)	編號	距離	方位角	航向	航速	威脅模式		

- (2) OV-5[MechanismFlowConnector]的內容變更如圖 2 所示

當更易為以 UAV 作偵蒐工具後，全部的作戰程序中，UAV 以其直接即時影像後傳之特性，在原 OV-5 的 A1.3「雷情呈核」這動作，已因直接後傳的即時特性而取消了。

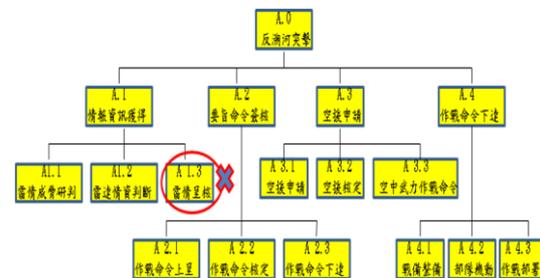


圖 3 OV-5 的內容變更情形

- (3) OV-6c[Event]的內容變更(如圖 4 所示)

當更易為以 UAV 作偵蒐工具後，在 OV-6c 的 A 1.1「雷情威脅研判」動作上，在目前(As-is)的作業上，自情報官一路上傳之情資，在情資會誤解時有回傳要求補強；在未來(To-be)的作業程序上，因 UAV 以其即時影像後傳之特性，在此動作上，已因直接後傳的即時特性而取消了(如圖紅色之描述)。

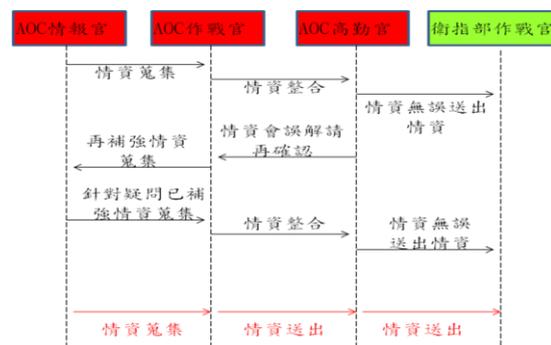


圖 4 OV-6c 的內容變更情形

- (4) OV-7[AttributeType]及 OV-7[EntityType]的內容變更(如圖 5 所示)

當更易為以 UAV 作偵蒐工具後，以 UAV 的直接即時影像後傳之特性，不再像海巡署岸

際岸雷達的面向海面搜尋特性; UAV 飛行於海面、陸地、高山、平原各地形上, 所以在 OV-7 的資料內容中, 需加入原來所沒有的「地形資料」這項類別(如圖粉紅色之欄位), 並帶出高度、地物種類、人口稠密性等資料屬性。

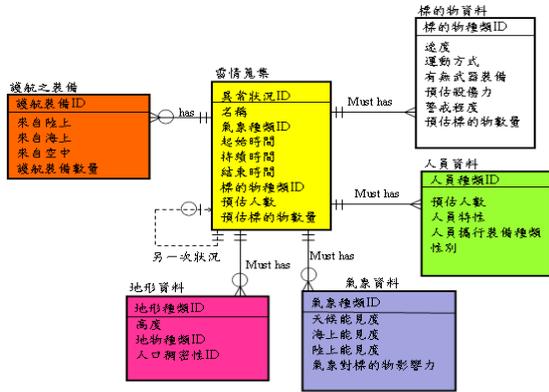


圖 5 OV-7 的內容變更情形

接即時影像後傳之特性, 不再像海巡署岸際雷達的面向海面搜尋特性, UAV 飛行於海面、陸地、高山、平原各地形上, 所以遵循在 OV-7 的資料內容中, 加入原來所沒有的「地形資料」這項類別(如圖粉紅色之欄位), 並帶出高度、地物種類、人口稠密性等資料及資料屬性。

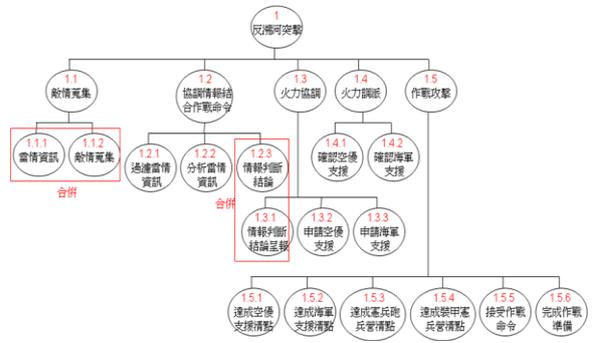


圖 6 SV-4 的內容變更情形

(5) SV-6[SysDataExchange]的內容變更(如表 3 所示)

當更易為以 UAV 作偵蒐工具後, 系統資訊的傳遞上, UAV 以其即時影像後傳之特性, 「戰場狀況」與「敵運動路線」的分析, 因指揮中心已可直接目視戰場而可將此戰場狀況的全部資訊傳遞取消, 不再透過各單位的逐步傳遞來完成。

表 3 SV-6 的內容變更情形

Interface Name	System Data Exchange	Source Type	Source System/Element/Software	Source System Note	Destination Type
臺全部_福山(地理)	戰位資訊	System Entity	地理偵查系統_4.01	臺兵司令部(衛戍指揮部)	System Entity
臺全部_福山(地理)	戰場分析表	System Entity	地理偵查系統_4.01	臺兵司令部(衛戍指揮部)	System Entity
臺全部_福山(地理)	戰場路線分析	System Entity	地理偵查系統_4.01	臺兵司令部(衛戍指揮部)	System Entity
臺全部_福山(火控_兵員)	戰火砲彈分析	System Entity	火控系統_2	臺兵司令部(衛戍指揮部)	System Entity
大直_臺全部(地理)	戰術命令	System Entity	火控系統_2	臺兵司令部(衛戍指揮部)	System Entity
大直_臺全部(地理)	戰術命令列表	System Entity	地理偵查系統_3.02	大直分區指揮部	System Entity
大直_臺全部(地理)	戰術路線分析	System Entity	地理偵查系統_3.02	大直分區指揮部	System Entity
臺全部_大直(地理)	戰術路線	System Entity	地理偵查系統_3.02	大直分區指揮部	System Entity
臺全部_大直(地理)	戰術路線	System Entity	地理偵查系統_4.01	臺兵司令部(衛戍指揮部)	System Entity
臺全部_大直(南山)	戰場分析表	System Entity	南山系統_4.01	臺兵司令部(衛戍指揮部)	System Entity
臺全部_大直(火控_南山)	戰火砲彈分析	System Entity	南山系統_4.01	臺兵司令部(衛戍指揮部)	System Entity
臺全部_大直(火控_南山)	戰術路線分析	System Entity	火控系統_4.01	臺兵司令部	System Entity
臺全部_大直(火控_南山)	戰術路線分析	System Entity	火控系統_4.01	臺兵司令部	System Entity
臺全部_大直(火控_南山)	戰術路線	System Entity	火控系統_4.01	臺兵司令部	System Entity
臺全部_大直(火控_南山)	戰術路線	System Entity	火控系統_4.02	臺兵司令部	System Entity
臺全部_大直(火控_南山)	戰術路線	System Entity	火控系統_4.02	臺兵司令部	System Entity
福山_臺全部(地理)	戰術命令	System Entity	地理偵查系統_3.03	福山分區指揮部	System Entity
大直_臺全部(南山)	戰術命令	System Entity	南山系統_3.02	大直分區指揮部	System Entity
臺全部_大直(南山)	戰術命令	System Entity	南山系統_3.02	大直分區指揮部	System Entity
臺全部_大直(兵員)	戰術命令	System Entity	兵員系統_4.02	臺兵司令部	System Entity
臺全部_大直(兵員)	戰術命令	System Entity	兵員系統_4.02	臺兵司令部	System Entity
臺全部_大直(火控)	戰術命令	System Entity	火控系統_11	臺兵司令部	System Entity
臺全部_大直(火控)	戰術命令	System Entity	火控系統_11	臺兵司令部	System Entity

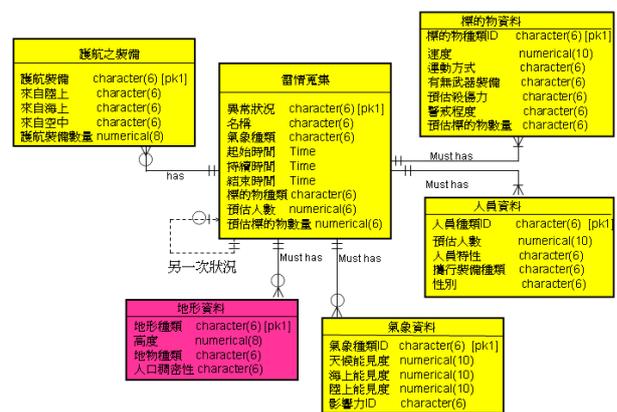


圖 7 SV-11 的內容變更情形

(6) SV-4[IoSysDataFlow]的內容變更(如圖 6 所示)

當更易為以 UAV 作偵蒐工具後, 系統設計上的程序中, UAV 以其即時影像後傳之特性, 原 SV-4 的 1.1.1 「雷情資訊」及 1.1.2 「敵情蒐集」這二動作, 已因直接後傳的即時特性而合併為一, 另 1.2.3 「情報判斷結論」及 1.3.1 「情報判斷結論呈報」這二動作亦合併為一。

(7) SV-11[SysAttributeType]、SV-11[SysEntityType]的內容變更(如圖 7 所示)

當更易為以 UAV 作偵蒐工具後, 以 UAV 的直

經過以上因另規劃「無人駕駛飛機」戰場物件, 更動了 OV-3[InformationExchange]展開追溯過程, 一直到佇列 Q = null, 所以追溯動作停止。資料元素集合 A 內的資料元素, A = (OV-3[InformationExchange] OV-5[MechanismFlowConnector] OV-6c[Event] OV-7[AttributeType] OV-7[EntityType] SV-6[SysDataExchange] SV-4[IoSysDataFlow] SV-11[SysAttributeType] SV-11[SysEntityType]) 即為要更動改的資料元素; 亦驗證了本研究所發展 59x59 追溯性矩陣的正確性。

## 六、結論

DoDAF 的 26 項產品相依的追溯過程隱含著: 作戰觀點(OV)、系統觀點(SV)及技術觀點(TV)各觀點內的橫向追溯; 作戰觀點(OV)、系統觀點(SV)、技術觀點(TV)各產品間, 跨觀點的縱向追溯及跨產品間間接追溯, 惟有發展一特定的軟體追溯工具才能解決此追溯性問題。

為解決 DoDAF 各產品間的追溯性問題，我們發展出一套追溯性矩陣的方法；此方法在執行上，先定義 DoDAF 綜合觀點、作戰觀點、系統觀點及技術觀點，各觀點與產品的資料元素，作出一個 59x59 的追溯性矩陣；本研究於尋獲 DoDAF 各資料元素相互間的追溯特性後，對後台資料庫的各該資料元素作出驗證的動作，本研究亦驗證了所發展 59x59 追溯性矩陣的正確性。

#### 參考文獻

- [1] Tseng, M., Harn, M. and Yang, J., "The Traceability Research of Changing Data Elements in the DoDAF Products," *Journal of C.C.I.T.* Vol. 41, No. 2, December 2012. (Accepted)
- [2] Soonsongtanee, S. and Limpiyakorn, Y., "Enhancement of Requirements Traceability with State Diagrams," *The proceedings of the 2nd International Conference on Computer Engineering and Technology*, IEEE, April 2010.
- [3] Xie, Y., Tang, T., Xu, T., and Zhao, L., "Research on Requirement Management for Complex Systems," *The Proceedings of the 2nd International Conference on Computer Engineering and Technology*, 2010.
- [4] <http://www-01.ibm.com/software/awdtools/reqpro/>
- [5] <http://www-01.ibm.com/software/awdtools/doors/productline/>
- [6] Sousa, A., Kulesza, U., Rummler, A., Moreira, A., Amaral, V., Araújo, J., Anquetil, N., and Mitschke, R., "A Model-Driven Traceability Framework to Software Product Line Development," 2010, <http://www.ample-project.net/>.
- [7] *DoDAF Version 2.0*, Volume I, pp. 15-52, May 2009.