

以結構行為合一改善架構之動態行為 Improving the Dynamic Behavior of Architecture via Structure-Behavior Coalescence

韓孟麒

德明財經科技大學
資訊科技系
副教授

harn@takming.edu.tw

趙善中

中華企業架構師學會
理事長

architectchao@gmail.com

李金龍

德明財經科技大學
資訊管理系
研究生

jacky00745@hotmail.com

摘要

由於目前的企業架構方法，將結構與行為分離，所以無法將企業架構的整體觀、顆粒觀、靜態結構與動態行為有效地表達出來。有鑑於此，本研究的目的，乃在提出先以「企業架構動態行為塑模準則」，配合 SBC 六大金律，來繪製企業架構視圖；再來建構以 UML、Zachman、TOGAF、DoDAF、MoDAF 等為規範的企業架構應用。SBC 的「結構行為合一」論，簡單、易懂、易學，可以迅速建構複雜的企業架構。本研究以物品監控系統為例，驗證所提的方法，可以有效的把領域內所有靜態結構的構件，緊密地與動態行為結合。

關鍵字：SBC 架構、架構方法論、SBC 架構描述語言、物品監控系統架構

Abstract

Due to the distraction of structure and behavior in the current enterprise architecture method, the representation for integrity view, granularity view, static structure and dynamic behavior is inefficiency. The purpose of this paper is to propose the Modeling Principle of EA Dynamic Behavior (MPEADB) with the six golden principles of Structure-Behavior Coalescence (SBC) for describing the enterprise architecture views in advance, and then constructing the enterprise architecture application with framework such as UML, Zachman, TOGAF, DoDAF, or MoDAF. SBC is quiet easy to understand and learn for complicate enterprise architecture. This paper takes the Object Monitor System (OMS) for example to verify the method proposed that can efficiently combine all of the component of static structure with dynamic behavior in a specific domain.

Keywords: Structure-Behavior Coalescence (SBC) architecture, architecture method, SBC architecture description language, object monitor system.

1. 緒論

目前所有的企業架構(Enterprise Architecture, EA)模式的設計，大都是靜態結構(Static Structure)與動態行為(Dynamic Behavior)分離，不能有效描述動態行為(Dynamic Behavior)。[1] 有鑑於此，本研究提出了先植基於企業架構的靜態結構，再於其

上建構企業架構的動態行為，形成「企業架構動態行為塑模準則」(Modeling Principle of EA Dynamic Behavior, MPEADB)；其目的在改善系統靜態結構與動態行為分離的缺失，讓靜態結構與動態行為合一，以符合企業架構的基本要求。

目前學術及實務界所慣用的企業架構，有：Zachman[9]、DoDAF[2]、MoDAF[6]、TOGAF[8]、...等，由其許多的案例與論文中得知，它們無法有效地表達動態結構。[7] 為能將靜態結構與動態結構合一表達，本研究使用中山大學資管系趙善中教授所發明之「結構行為合一」(Structure-Behavior Coalescence, SBC)論，運用其六大金律圖，來建構企業架構。[1]

最早應用「以架構為中心理論」(Architecture-centric Theory)的是軟體工程[4]，在其應用軟體(Application Software)的系統分析與設計(System Analysis and Design)工作，大都以傳統軟體工程中的功能導向(Function-oriented)或過程導向(Process-oriented)為基礎，使得應用軟體在做開發與演進時，無法有效地描述使用者需求(User Requirements)，也無法有效地表達應用軟體的靜態結構與動態行為。現在業界所使用的物件導向軟體工程(Object-oriented Software Engineering)，大都以統一塑模語言(Unified Modeling Language, UML)為基礎，雖然解決了部分應用軟體的靜態結構與動態行為問題，但是仍然沒有架構的整體觀(Integrity View)與顆粒觀(Granularity View)。[3] 我們研究發現，有關企業架構動態行為的開放性問題，有：

- (1) 一般的企業架構只重視描述靜態結構，對動態行為描述明顯不足。
- (2) 企業亟需架構正型化(Formalizing)的描述。[5]
- (3) 企業轉型(Business Transformation)必須借助企業架構。
- (4) 軟體工程缺乏對動態行為的描述。

SBC 藉助了「SBC 架構描述語言」(SBC Architecture Description Language, SBC-ADL)來塑模系統，它律定了六大架構圖，稱之為「SBC 架構框架」(Structure-Behavior Coalescence - Architecture Framework, SBC-AF)或六大金律[1]；即：架構階層圖(Architecture Hierarchy Diagram)、框架圖(Framework Diagram)、構件操作圖

(Component Operation Diagram)、構件連結圖(Component Connection Diagram)、結構行為合一圖(Structure-Behavior Coalescence Diagram)及互動流程圖(Interaction Flow Diagram)等。本研究之學術與實務貢獻，乃是提出了「企業架構動態行為塑模準則」，以物品監控系統架構為例，將其業務架構(Business Architecture)與軟體架構(Software Architecture)利用 SBC 來描述，並分析 UML、Zachman、DoDAF、MoDAF、TOGAF 等架構規範的動態行為，驗證 SBC 架構與「企業架構動態行為塑模準則」的優越性，以彌補上述架構規範的不足。

2. 相關技術與文獻探討

2.1 相關技術

Zachman、DoDAF、MoDAF、TOGAF 等企業架構，是目前被公認最有名的企業架構。Zachman 架構是企業架構的濫觴，因為優早被提出來，所以成為各家企業架構創造者參考的對象。[9] DoDAF 是 Department of Defense Architecture Framework 的縮寫，為美國軍方的架構規範(或稱之為框架)，專門應用在軍事領域，也可應用在企業。[2] MoDAF 是 Ministry of Defence Architecture Framework 的縮寫，為英國國防部選定的英國國防部架構規範，主要是應用在作戰領域。[6] TOGAF 為 The Open Group Architecture Framework 的縮寫，是一個標準的及開放的架構規範。TOGAF 是 The Open Group 為了組織設計、組織評估和建立正確的組織架構而律定的，它包含詳細的方法論及一些支援性的工具。[8] 這個方法論稱之為架構開發方法(Architecture Development Method, ADM)，一共有十個階段(Phases)。而 SBC 架構是唯一具有「結構行為合一」特徵的一種系統架構，SBC 架構嚴格要求結構觀點(Structure View)與行為觀點(Behavior View)兩者必須合一，其核心思維指的就是[1]：

架構 = 結構 + 行為 + 其他觀點

本研究所定義的「結構行為合一」係指：在企業架構中，屬於領域內靜態結構的構件(Component)，都必須是動態行為的一部分。若靜態結構的構件不是動態行為的一部分，就不應被描述；否則就是對企業架構的動態行為，描述不足。

2.2 文獻探討

除了 SBC 架構，現有企業架構對動態行為均描述不足，可以由 UML、Zachman、DoDAF、MoDAF、TOGAF 等對企業架構的動態行為設計的圖或規範中可以看出，說明如下：

(1) UML 動態行為

UML2.0 所提供的 13 個圖中，只有活動圖(Activity Diagram)、狀態機圖(State Machine Diagram)、使用個案圖(Use Case Diagram)是屬於行為式圖形(Behavior diagrams)；而 UML2.0 其他圖所屬的靜態結構構件，不完全被這三個行為式圖形

所描述。UML 方法中其他靜態結構圖形之構件，若遷就這三個行為式圖形而建構，會造成其他圖形構件的增減，這個增減現象，又會造成所屬靜態結構構件，不完全被這三個行為式圖形所描述。企業架構師在增減的過程中，並無所有構件的整體觀與顆粒觀。

(2) Zachman 架構動態行為

Zachman 架構所提供的 36 個規範中，只有業務過程模式(Business Process Model)、工作流模式(Work Flow Model)是屬於行為式圖形；而 Zachman 架構其他規範的所屬靜態結構構件，不完全被這兩個行為式圖形所描述。[9]

(3) DoDAF 動態行為

DoDAF1.0 所提供的 26 個規範中，只有作戰觀點的 OV-5 作戰行動模式圖(Operational Activity Model)、OV-6b 作戰狀態轉換模式(Operational State Transition Description Model)、OV-6c 作戰循序圖(Operational Sequence Diagram)、SV-4 系統功能描述圖(System Function Description Diagram)、SV-10b 系統狀態轉換描述模式(System State Transition Description Model)及 SV-10c 系統循序圖(System Sequence Diagram)是屬於行為式圖形；而 DoDAF 其他規範的所屬靜態結構構件，不完全被這些行為式圖形所描述。[2]

(4) MoDAF 動態行為

MoDAF 所提供的規範中，只有作戰觀點的 OV-5 作戰行動模式圖(Operational Activity Model)、OV-6b 作戰狀態轉換模式(Operational State Transition Description Model)、OV-6c 作戰事件追蹤模式(Operational Event-Trace Description Model)、SV-4 系統功能描述圖(System Function Description Diagram)、SV-10b 資源狀態追蹤轉換描述模式(Resource State Transition Description Model)及 SV-10c 資源事件追蹤描述模式(Resource Event-Trace Description Model)是屬於行為式圖形；而 MoDAF 其他規範的所屬靜態結構構件，不完全被這些行為式圖形所描述。[6]

(5) TOGAF 動態行為

TOGAF 所提供的視圖中，只有業務架構的業務過程視圖(Business Process View)、是屬於行為式圖形；而 TOGAF 其他規範的所屬靜態結構構件，不完全被這個行為式圖形所描述。[8]

3. 企業架構動態行為塑模準則

SBC 架構可以完完全全地將企業架構的靜態結構與動態行為描述清楚，而且其所有「構件」，都可以被靜態結構與動態行為所描述。本研究提出的「企業架構動態行為塑模準則」，配合 SBC 六大金律，一共有七個步驟，說明如下：

(1) 首先以正向工程(Forward Engineering)或反向工程(Reverse Engineering)的方式，以整體觀(Integrity View)找出所有相關的構件。

(2) 其次將所有構件，依照給予分類，以顆粒觀

(Granularity View)繪出架構階層圖。

- (3) 將分類好的構件，依階層繪出框架圖。
- (4) 將框架圖內之構件，找出操作要點(Operational Slot)，繪製構件操作圖。
- (5) 利用構件操作圖，找出構件連結方式，繪出構件連結圖。
- (6) 利用構件連結，繪出結構行為合一圖。
- (7) 利用結構行為合一圖，繪製互動連結圖，將領域內所有相關靜態結構構件的動態行為繪製出來。

4. 實例驗證：SBC 動態行為塑模

本研究以物品監控系統(Object Monitor System, OMS)為例，使用 Microsoft Office Visio SBC 模組，依「企業架構動態行為塑模準則」七個步驟，繪製六大金律圖，其說明如下：

(1) OMS 架構階層圖

本研究是以物品監控系統為例，並且直接將「小偷」與「系統管理員」抽離監控系統監控人員層，做為系統行為驅動者，形成整體防護體系來研究，讓原本的監控系統流程體系變得簡單明瞭，使程式設計師撰寫監控系統時，變得更容易明瞭使用者需求。

首先從 SBC 架構所衍生的鑽石模式(Diamond Model)[1]，來思考物品監控系統所包含四個構面間的互動關係，其中組織結構及科技(包括軟體、硬體、設備等)為結構觀點，人員(包括技能、動機與合作等)、任務(包括流程與專案等)為行為觀點。組織結構及科技部分，採重點式內容設計為主。

OMS 架構階層圖，如附錄圖 1 所示，第一層的分解是以物品監控的專業性人員來分解，接著將監控設備依據用途來分解，最後將電腦軟體依據資訊系統分類再分解。

物品監控系統以專業性分類，可分為人員、硬體和軟體，本研究找出物品監控系統共需要 17 個構件所組合而成。這 16 個構件分別為「安裝人員」、「保全人員」、「監控人員」、「維護人員」、「管理人員」、「物品標籤」、「節點位置」、「路由器」、「接收端」、「閘道器」、「登入軟體構件」、「硬體連接軟體構件」、「定位軟體構件」、「追蹤軟體構件」、「記錄軟體構件」、「警示軟體構件」等。

而這 16 個構件以專業性分類可以分為三層：監控人員層、監控用硬體設備層和電腦軟體層，本研究將這 16 個構件分門別類，並且繪製出架構階層圖，說明如下：

· 第一層：監控人員層

本研究以使用物品監控系統遇到的使用者或管理人員，分為：「保全人員」、「安裝人員」、「管理人員」、「監控人員」和「維護人員」等，並且將「小偷」抽離監控人員層，置放於物品監控系統外部，做為行為驅動者，「小偷」是偷竊人員的代表，他會觸發物品監控系統的警示軟體構件。

· 第二層：監控用硬體設備層

此層為物品監控系統所使用的機器設備硬體描述，本研究物品監控系統是使用 ZigBee 硬體構件，分別為：「物品標籤」、「節點位置」、「路由器」、「接收端」、「閘道器」等。

· 第三層：電腦軟體層

此層為物品監控系統所使用的軟體構件，本研究開發的軟體構件分別為：「登入軟體構件」、「硬體連接軟體構件」、「定位軟體構件」、「追蹤軟體構件」、「記錄軟體構件」、「警示軟體構件」。

(2) OMS 框架圖

OMS 框架圖是由 OMS 架構階層圖中，將每一階層繪出框架並將構件重新組合及詳細描述。本研究將架構階層圖分為三個框架，分別為：「監控人員層(即人員或組織)」、「儀器設備層(即硬體)」、「電腦系統層(即軟體)」，每一層的構件沒有重覆，且為結構觀點之一，如附錄圖 2 所示。

(3) OMS 構件操作圖

經由上述架構階層圖與框架圖，找到所需的 16 個構件後，即可著手繪製構件操作圖。繪製構件操作圖時，首先要先找出所有構件的功能，分析它們在物品監控系統中的功能面。

依現行法規、職責權限、硬體功能與軟體功能，將 16 個構件逐一檢視與分析其功能，繪製構件操作圖，並且將每個構件的輸出參數與輸入參數明細寫出來。每個構件會有不同的行為與權限，以管理人員為例，相對應的行為有：可以接收狀況(即接收狀況)、下達指令給其他人員或軟體設備(即下達指令作業)，並且可以解除警報系統(即解除警報作業)。所以本研究將以此類推，繪製構件操作圖，並利用表單來詳細描述所有構件功能，如附錄表 1。進而將每個構件的操作名稱、輸出參數與輸入參數，詳細繪製成一張構件操作圖；藉此圖我們可以繪製 SBC 六大金律中的構件連結圖、結構行為合一圖及互動流程圖。本研究將構件操作圖分為兩大塊，如附錄圖 3 所示。

其一為人員與硬體設備，如附錄圖 3(a)所示。本研究不將「小偷」與「系統管理員」放入人員構件內，但假設「小偷」與「系統管理員」是外部行為驅動者。

其二為軟體構件，如附錄圖 3(b)所示。雖然硬體設備與軟體息息相關，本研究未來將詳細描述軟體階層，並且將軟體功能詳細劃分，繪製出複合數據類型的規格表。

(4) OMS 構件連結圖

OMS 構件連結圖是從結構觀點來看一個系統，將系統內構件與構件及構件與外界環境做連結。連結後本研究進一步可導出每個行為是由哪些結構重覆連結而成，進而設計 OMS 構件連結圖。如附錄圖 4 所示，「小偷」與「系統管理員」之間有直接連結，而「系統管理員」下面所轄管的「管理人員」、「安裝人員」、「安全人員」、「監控人員」和「維護人員」等所有人員的動作都能用構件連結圖描述出來，「物品監控軟體構件模型」連結也一

一被呈現出來；因此，結構觀點就更清晰可見。

(5) OMS 結構行為合一圖

由於「物品監控軟體構件模型」是透過一個群集的方式共同形成的，本研究根據物品監控所涵蓋的環節，從中找出幾個在物品監控常見的位置佈署行為及防護竊取行為，分別為：破壞物品標籤、物品離開安全區域、監控位置佈點行為、檢視歷史資料、刪除記錄資料、設備異常維護行為等 6 個動作，本研究將這 6 個動作分別用線描述 6 個結構，完成 OMS 結構行為合一圖，如附錄圖 5 所示。

(6) OMS 互動流程圖

OMS 互動流程圖主要是從行為觀點來看問題，詳細描述每一個行為其外界環境、構件之間互動順序及過程中所傳遞的訊息、資料、物質或能量等。對於「物品監控軟體構件模型」而言，因為不同人講同一件事，卻有多種說法，也就是必須依靠經驗完成。而一般流程圖也只能呈現出單一決策觀點且沒有對應結構(Aligned Structure)支援，容易造成流程理想過於完美，而實際運作是有落差的。OMS 互動流程圖最重要的是有能力整合散落的其他觀點，表達出多重觀點，也就是說有能力可以將經驗標準化，這樣不同人講出同一個「物品監控軟體構件模型」行為會是一致的，且有依據性。

OMS 互動流程圖的設計方法是將 OMS 結構行為合一圖中的 6 個物品監控系統行為，繪製出 6 個互動流程圖，詳細描述出物品監控系統行為在互動過程中結構的操作連結順序，及其所攜帶的輸入與輸出參數。所謂的參數係指其他觀點，是可以對應到現行已存在的文件，包括溝通訊息、表單或報表資料等，茲將 6 個物品監控系統行為逐一說明如下：

[行為 1：破壞物品標籤]

「破壞物品標籤」行為的互動流程圖，如附錄圖 6 所示，說明如下：

- (1) 首先，外界環境「小偷」與構件「管理人員」發生「接收狀況」操作的互動，並帶著輸入參數「標籤訊號異常」與之互動。
- (2) 接著，構件「管理人員」與構件「開道器」發生「接收訊號」操作的互動，並帶著輸入參數「訊號資訊」與之互動。
- (3) 構件「開道器」與構件「警示軟體構件」發生「警示作業」操作的互動，並帶著輸入參數「標籤訊號異常」與之互動。如狀況排除，構件「管理人員」與構件「警示軟體構件」發生「解除警示」操作的互動，接收「標籤訊號異常」並發出警報，再「取回物品」時解除警報。
- (4) 異常狀況之後構件「管理人員」與構件「安全人員」發生「通報安全人員」操作的互動，回應輸出參數「指令下達」的互動。
- (5) 最後，外界環境「小偷」與構件「安全人員」發生「攔截作業」操作的互動，並帶著輸入參數「取回物品」與之互動。

[行為 2：物品離開安全區域]

「物品離開安全區」行為的互動流程圖，如附錄圖 7 所示，說明如下：

- (1) 首先，外界環境「小偷」與構件「管理人員」發生「接收狀況」操作的互動，並帶著輸入參數「標籤移動資訊」與之互動。
- (2) 接著，構件「管理人員」與構件「追蹤軟體構件」發生「追蹤標籤作業」操作的互動，並帶著輸入參數「標籤位置資料」與之互動。
- (3) 構件「追蹤軟體構件」與構件「記錄軟體構件」發生「記錄作業」操作的互動，並帶著輸入參數「日期時間地點資訊」與之互動。
- (4) 構件「追蹤軟體構件」與構件「警示軟體構件」發生「警示作業」操作的互動，並帶著輸入參數「標籤離開安全區」與之互動。
- (5) 構件「追蹤軟體構件」與構件「監控人員」發生「異常狀態通知」操作的互動，並帶著輸入參數「標籤離開安全區」與之互動。
- (6) 構件「監控人員」與構件「安全人員」發生「通報安全人員」操作的互動，並帶著輸入參數「物品位置資訊」與之互動。如狀況正常則構件「監控人員」與構件「警示系統」發生「解除警示作業」操作的互動，帶著輸入參數「解除警示確認」的互動並回應參數「正常狀態」。
- (7) 接著，外界環境「小偷」與構件「安全人員」發生「攔截作業」操作的互動，並帶著輸入參數「取回物品」與之互動。
- (8) 最後「安全人員」利用密碼解除警報。

[行為 3：監控位置佈點]

「監控位置佈點」行為的互動流程圖，如附錄圖 8 所示，說明如下：

- (1) 首先，外界環境「系統管理員」與構件「安裝人員」發生「硬體安裝作業」操作的互動，並帶著輸入參數「節點位置」與之互動。
- (2) 接著，構件「安裝人員」與構件「物品標籤」發生「安裝標籤」操作的互動，並帶著輸入參數「物品標籤」與之互動。
- (3) 構件「物品標籤」與構件「登入軟體構件」發生「權限判斷作業」操作的互動，並帶著輸入參數「設置許可」與之互動。
- (4) 構件「登入軟體構件」與構件「位置軟體構件」發生「地圖位置」操作的互動，並帶著輸入參數「定位」與之互動。

[行為 4：檢視歷史記錄]

「檢視歷史記錄」行為的互動流程圖，如附錄圖 9 所示，說明如下：

- (1) 首先，外界環境「系統管理員」與構件「安全人員」發生「查詢記錄」操作的互動，並帶著輸入參數「狀態資料」與之互動。
- (2) 接著，構件「安全人員」與構件「登入軟體構件」發生「資料狀態」操作的互動，並帶著輸入參數「密碼」與之互動。
- (3) 構件「登入軟體構件」與構件「記錄軟體構件」發生「檢視歷史記錄」操作的互動，並帶著輸

出參數「歷史資料」與之互動。

[行為 5：刪除記錄資料]

「刪除記錄作業」行為的互動流程圖，如附錄圖 10 所示，說明如下：

- (1) 首先，外界環境「系統管理員」與構件「安全人員」發生「查詢記錄」操作的互動，並帶著輸入參數「狀態資料」與之互動。
- (2) 接著，構件「安全人員」與構件「登入軟體構件」發生「狀態資料」操作的互動，並帶著輸入參數「密碼」與之互動。
- (3) 構件「登入軟體構件」與構件「記錄軟體構件」發生「檢視歷史記錄」操作並回傳「資料」。
- (4) 最後將「刪除資料」並回傳「刪除確認」。

[行為 6：設備異常維護]

「設備異常維修」行為的互動流程圖，如附錄圖 11 所示，說明如下：

- (1) 首先，外界環境「系統管理員」與構件「安裝人員」發生「硬體資訊」操作的互動，並帶著輸入參數「狀態資料」與之互動。
- (2) 接著，構件「安裝人員」與構件「維護人員」發生「更換硬體設備」操作的互動，並帶著輸入參數「更換確認」與之互動。
- (3) 構件「維護人員」與構件「閘道器」發生「啟動所有設備」操作的互動，並帶著輸入參數「硬體」與之互動。
- (4) 構件「閘道器」與構件「定位軟體構件」發生「訊號傳遞測試」操作的互動，並帶著輸入參數「訊號輸入」與之互動。
- (5) 外界環境「系統管理員」與構件「維護人員」發生「排除雜訊狀況」操作的互動，並帶著輸入參數「排除」與之互動。
- (6) 然後，構件「維護人員」與構件「定位軟體構件」發生「確認硬體作業」操作的互動，並帶著輸入參數「測試狀態」與之互動，以及回應輸出參數「訊號」。
- (7) 然後，外界環境「系統管理員」與構件「維護人員」發生「更換設備電池」操作的互動，並帶著輸入參數「電池安裝」與之互動。
- (8) 最後，構件「維護人員」與構件「定位軟體構件」發生「確認硬體作業」操作的互動，並帶著輸入參數「確認硬體狀態」與之互動，以及回應輸出參數「訊號資料」。

由實證中得知，本研究之物品監控系統，其 OMS 架構階層圖所有的靜態結構構件，均為構件連結圖、結構行為合一圖及互動流程圖之一部分。

5. 結論

目前所有的企業架構規範，表達架構的動態行為稍顯不足，而架構的動態行為是要描述應用領域內所有靜態物件的操作行動。本研究以物品監控系統為例，將 SBC 六大金律以「企業架構動態行為塑模準則」的七個步驟繪製相關視圖，充分表達了第一步驟中所有的靜態物件的行為面。也就是在

企業架構中，屬於物品監控系統領域內靜態結構的構件，都是動態行為的一部分。

本研究的貢獻在於將 UML、Zachman、TOGAF、DoDAF、MoDAF 等架構規範做了一個重要的檢視，發現到對企業架構的動態行為設計有不足的地方，並積極尋求解決之道，提出了「企業架構動態行為塑模準則」，配合 SBC 六大金律，以補其不實之缺口。

未來的研究方向有二，分別為：將上述架構規範行為式圖形，一一做細部分析，提出具體建議，並改良之；以及針對每個軟體構件，再做顆粒化之細部建構。由於上述原創人或單位，其原先之設計理念，不是從「結構行為合一」為出發點，所以要改良十分困難。唯一可行的方法是：爾後所有的企業架構建構工作，都要以 SBC 的「結構行為合一」為基礎；也就是先以「企業架構動態行為塑模準則」，配合 SBC 六大金律，來繪製企業架構視圖，再來建構以 UML、Zachman、TOGAF、DoDAF、MoDAF 等為規範的企業架構應用。SBC 的「結構行為合一」的優點是簡單、易懂、易學，可以在最短的時間內，把企業架構的整體觀、顆粒觀、靜態結構、動態行為表達出來。物品監控系統的軟體構件的顆粒化，可以成為以「以架構為中心理論」的軟體工程議題。

參考文獻

- [1] W. Chao, *Software Architecture: SBC Architecture at Work*, National Sun Yat-sen University, 2011.
- [2] DoD Architecture Framework Working Group, *DoD Architecture Framework, Version 2.0 Volume 1: Introduction, Overview, and Concepts Manager's Guide*, 2009.
- [3] M. Harn and Cheng-hang Wang, "The DoDAF Support for a Huge-grain C4ISR System," *Proceedings of the Third Taiwan Conference on Software Engineering*, Taichung, Taiwan, June 8-9, 2007.
- [4] A. J. Lattanze, *The Architecture Centric Development Method Technical Report CMUISRI-05-103*, Institute for Software Research International, Carnegie Mellon University, 2005.
- [5] Luqi, "Formal Methods Promises and Problems," *IEEE Software*, January 1997, pp. 73-85.
- [6] *The MoDAF Version 1.2*, pp.4-9, Sept., 2008,
- [7] S. J. Ring, D. Nicholson, J. Thilenius and S. Harris, "An Activity-Based Methodology for Development and Analysis of Integrated DoD Architectures," *Proceedings of Command and Control Research and Technology Symposium*, San Diego, California, June 15-17, 2004.
- [8] The Open Group, *TOGAF Version 9: The Open Group Architecture Framework (TOGAF)*, 2009.
- [9] J. A. Zachman, "A framework for information architecture," *IBM System Journal*, Vol. 26, Issue 3, pp. 276-292, 1987.

附錄



圖 1 OMS 架構階層圖

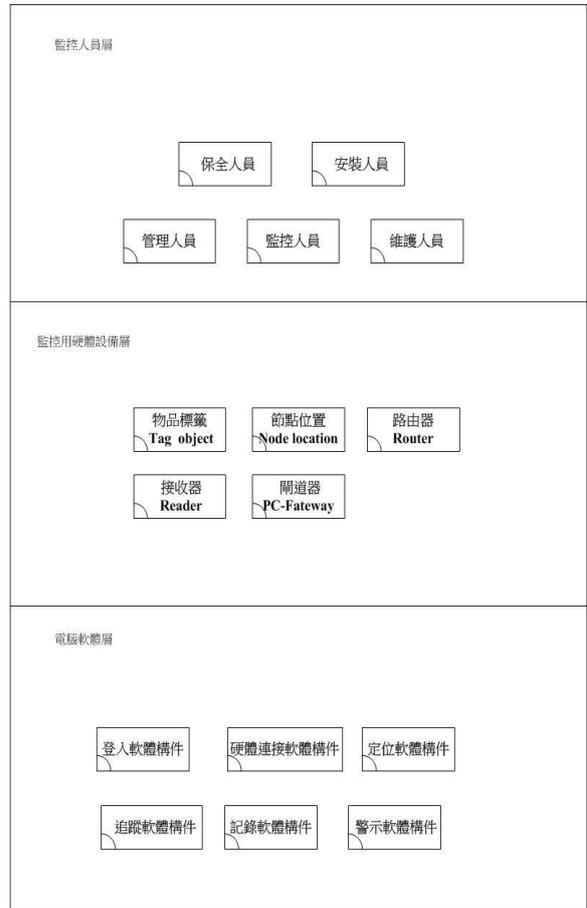
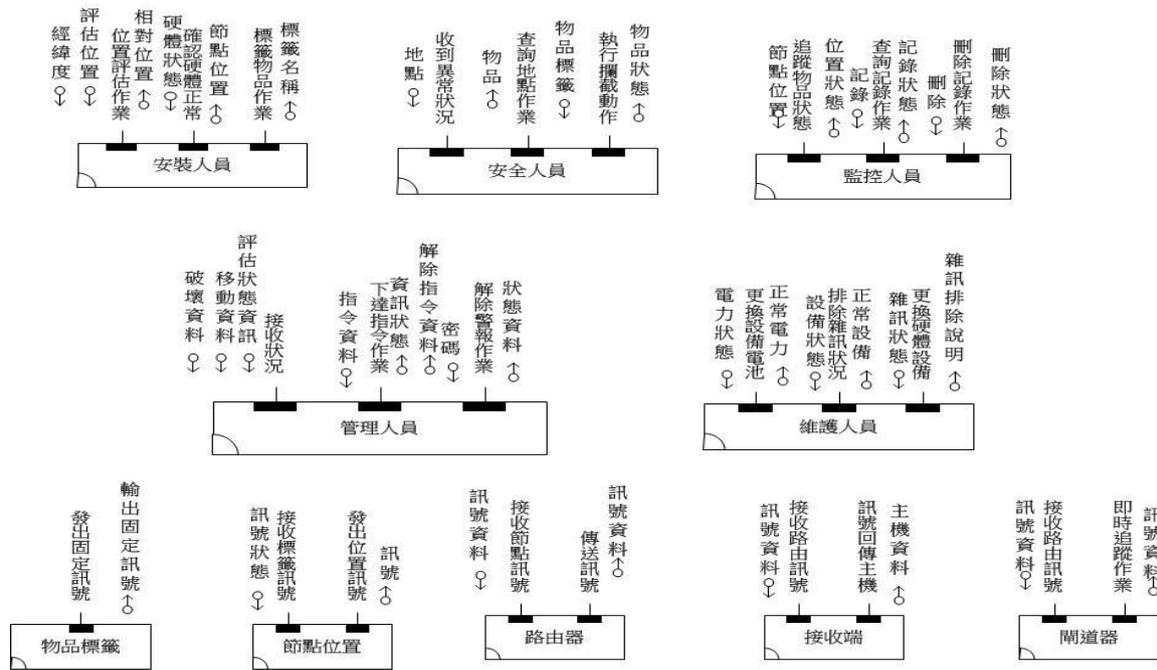
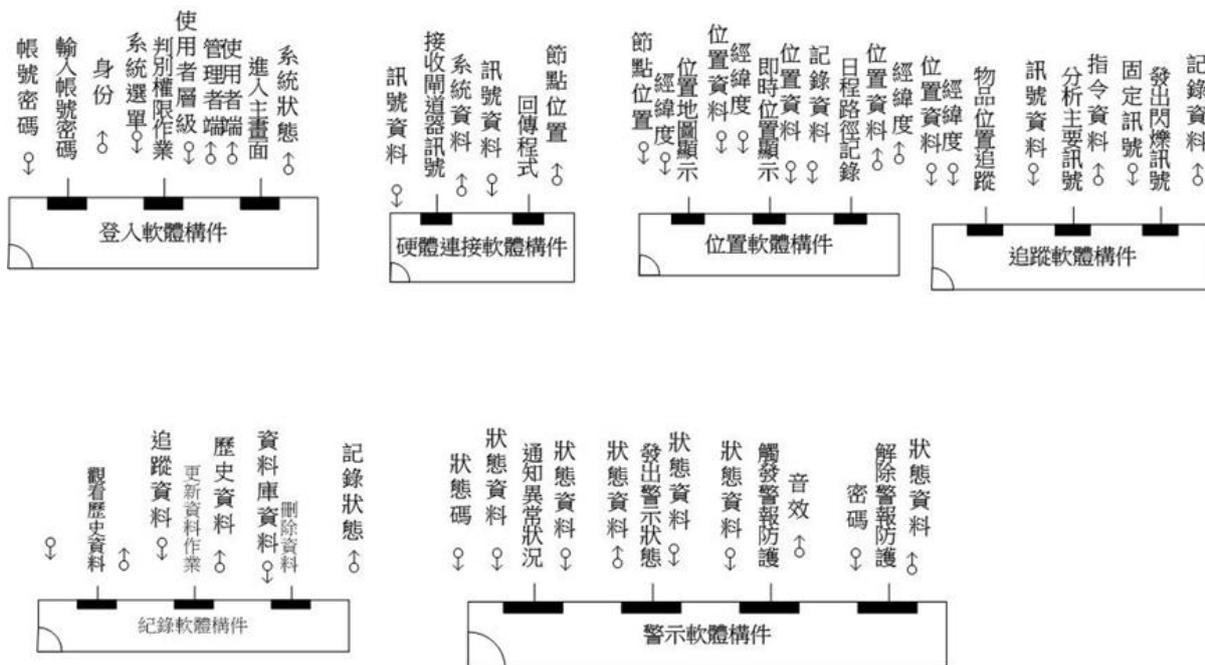


圖 2 OMS 框架圖



(a)



(b)

圖 3 OMS 構件操作圖

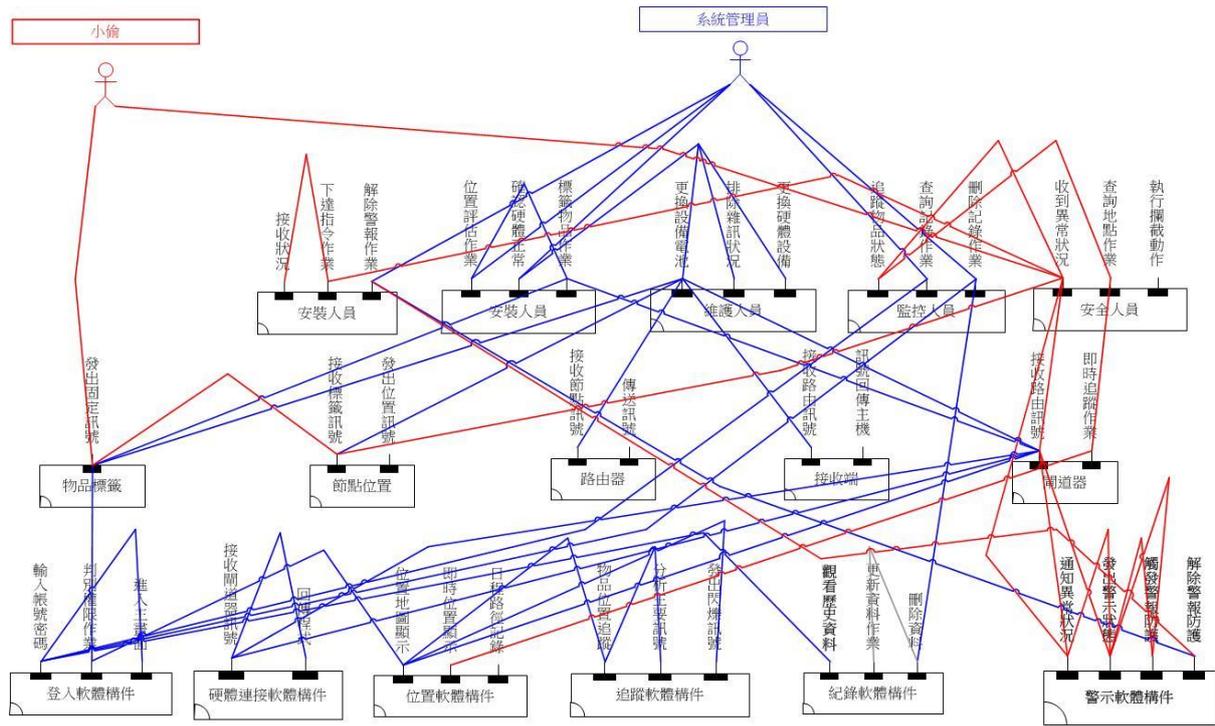


圖 4 OMS 構件連結圖

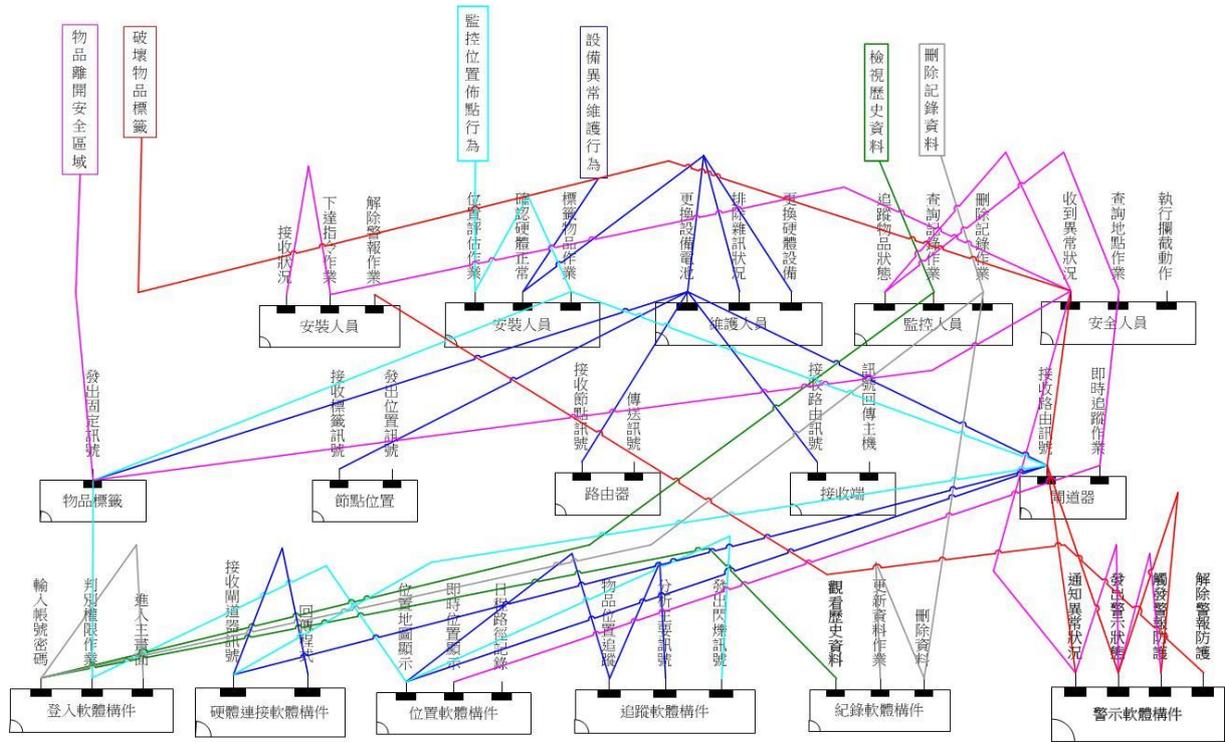


圖 5 OMS 結構行為合一圖

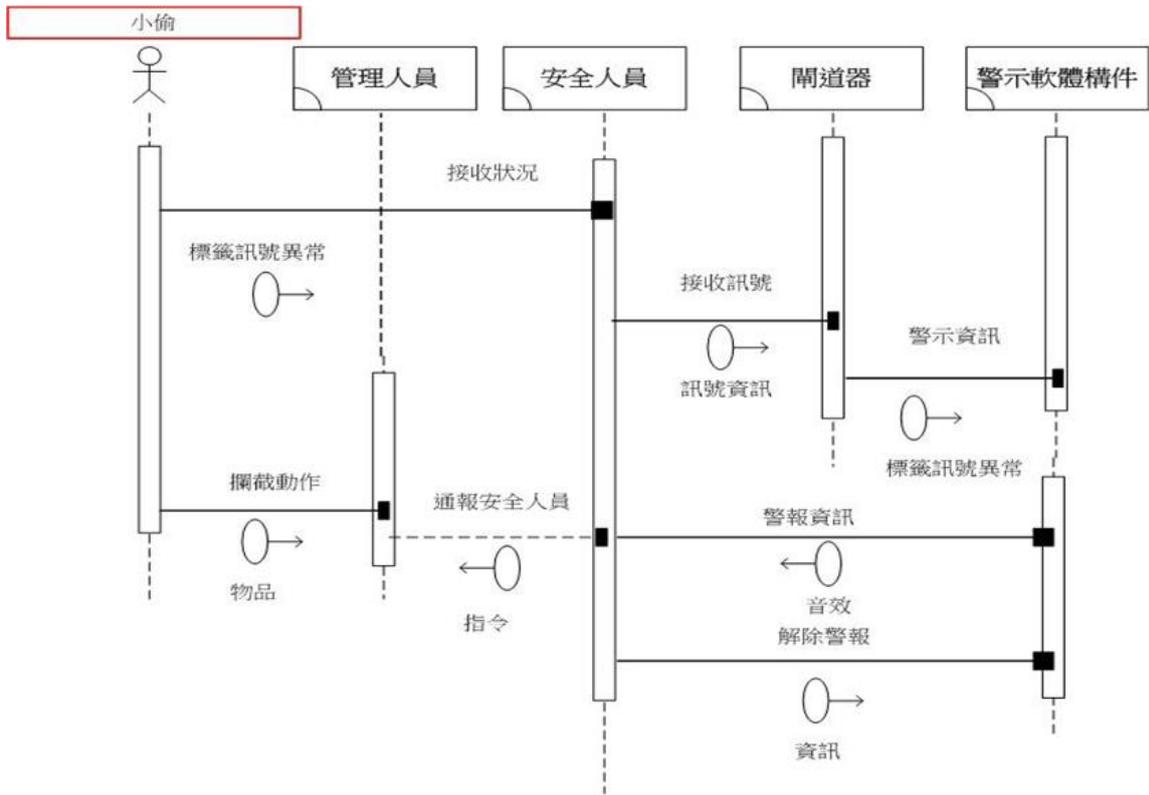


圖 6 「破壞物品標籤」行為的互動流程圖

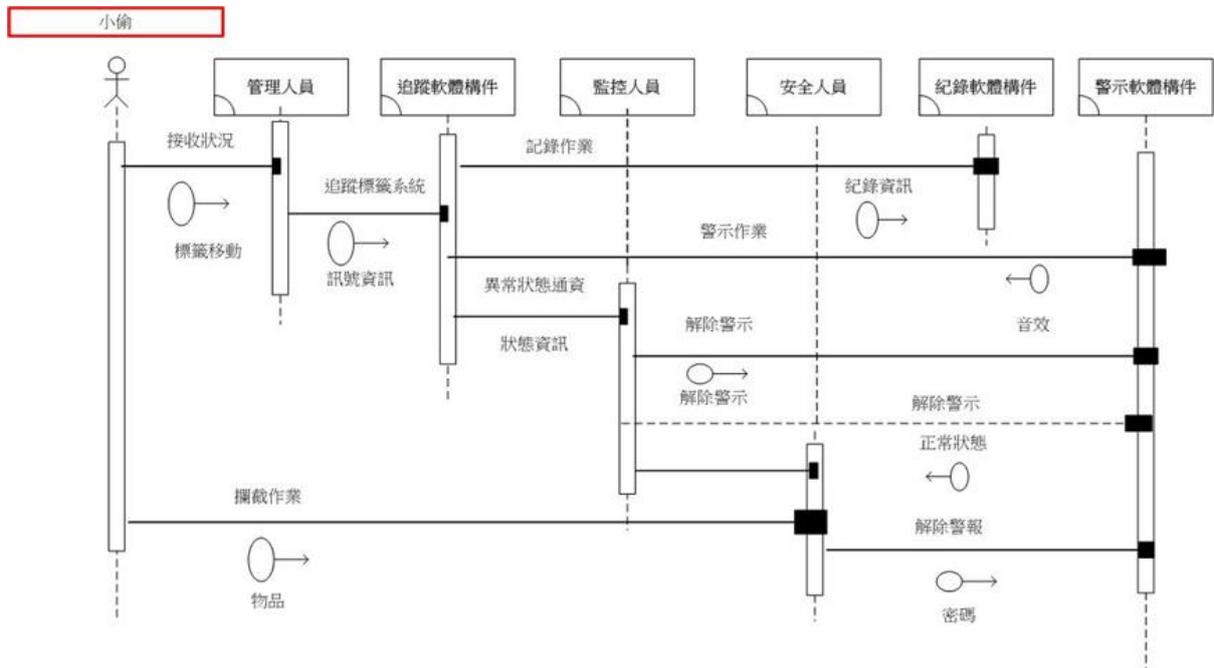


圖 7 「物品離開安全區」行為的互動流程圖

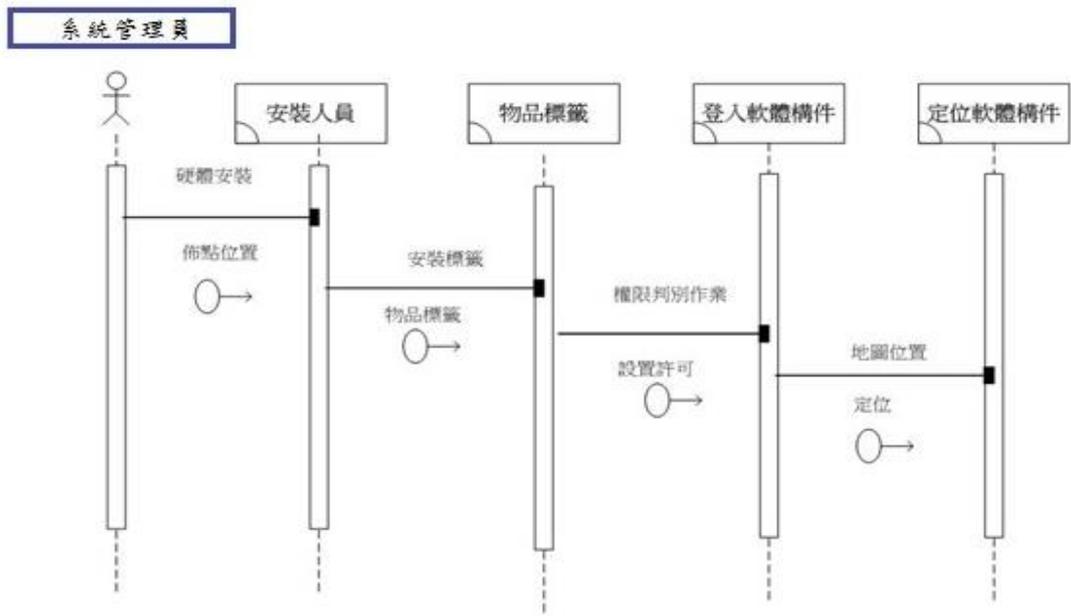


圖 8 「監控位置佈點」行為的互動流程圖

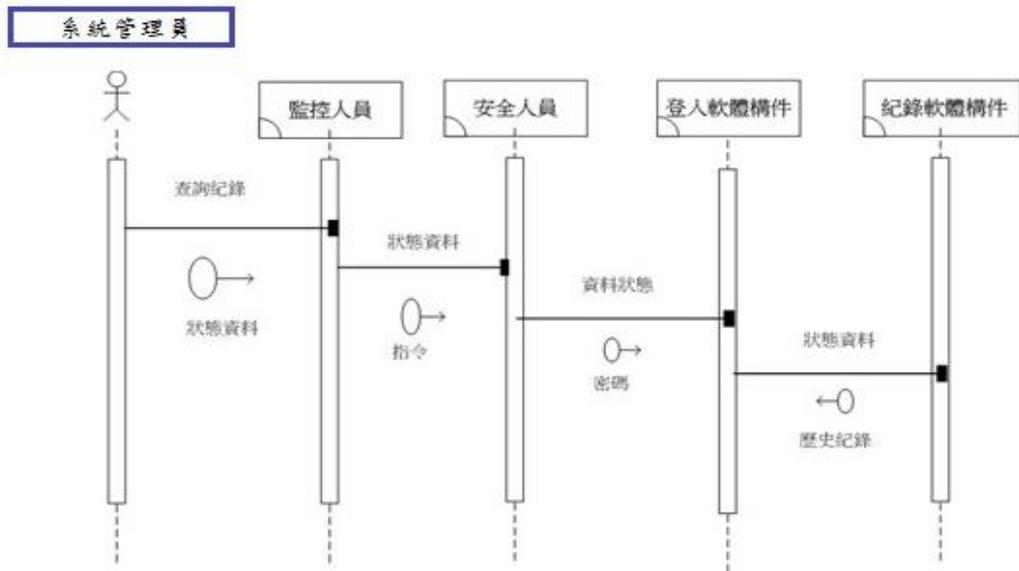


圖 9 「檢視歷史記錄」行為的互動流程圖

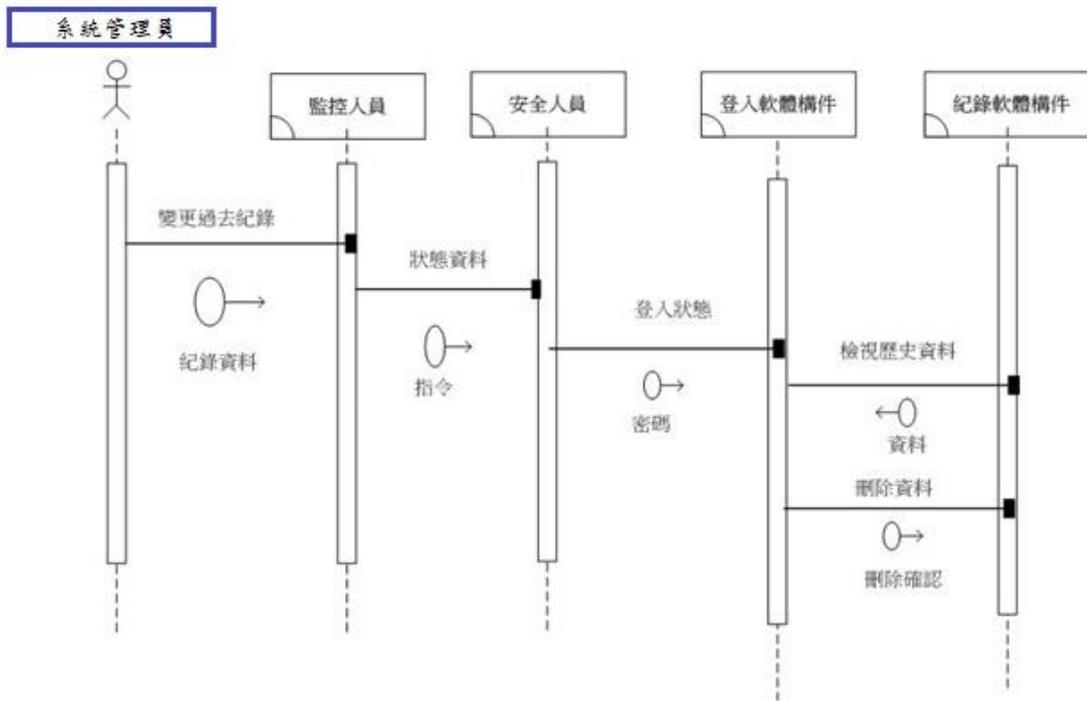


圖 10 「刪除記錄作業」行為的互動流程圖

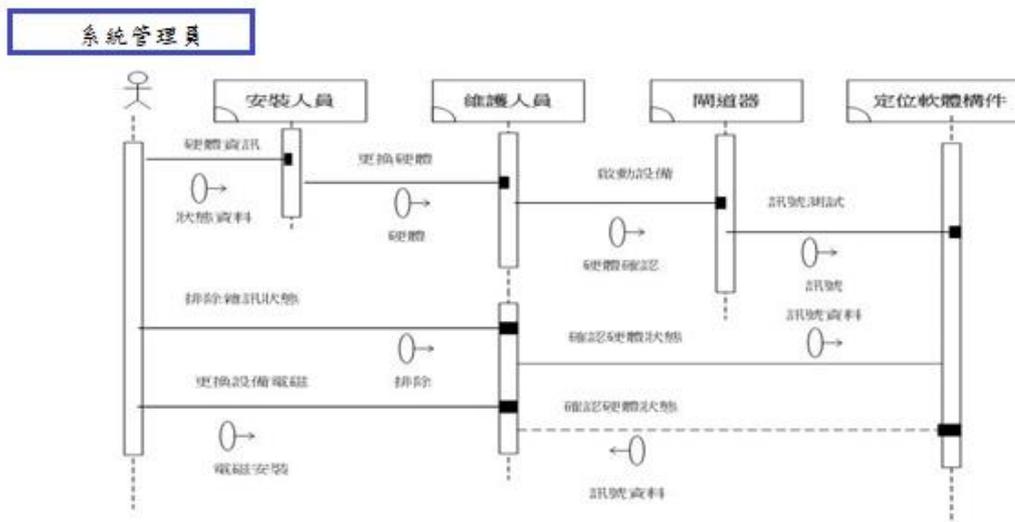


圖 11 「設備異常維修」行為的互動流程圖

表 1 操作構件表

項次	構件名稱	操作名稱	輸入參數	輸出參數
1	管理人員	接收狀況	評估狀態 資訊 移動資料 破壞資料	—
2		下達指令作業	指令資料	解除指令 資料 資訊狀態
3		解除警報作業	密碼	狀態資料
4	安裝人員	位置評估作業	評估位置 經緯度	相對位置
5		確認硬體正常	硬體狀態	節點位置
6		標籤物品作業	—	標籤名稱
7	安全人員	收到異常狀況	地點	物品
8		查詢地點作業	—	—
9		執行攔截動作	物品標籤	物品狀態
10	監控人員	追蹤物品狀態	節點位置	位置狀態
11		查詢記錄作業	記錄	記錄狀態
12		刪除記錄作業	刪除	刪除狀態
13	維護人員	更換設備電池	電力狀態	正常電力
14		排除雜訊狀況	設備狀態	正常設備
15		更換硬體設備	雜訊狀態	雜訊排除 說明
16	物品標籤	發出固定訊號	—	訊號資料
17	節點位置	接收標籤訊號	訊號狀態	—
18		發出位置訊號	—	訊號
19	路由器	接收節點訊號	訊號資料	—
20		傳送訊號	—	訊號資訊
21	接收端	接收路由訊號	訊號資料	—
22		回傳主機訊號	—	主機資料

23	閘道器	接收路由訊號	訊號狀態	—
24		即時追蹤作業	—	訊號資料
25	登入軟體構件	輸入帳號密碼	帳號密碼	身份
26		判別權限作業	系統選單 使用者層 級	管理者端 使用者端
27		進入主畫面	—	系統狀態
28	硬體連接 軟體構件	接收閘道器訊號	訊號資料	系統資料
29		回傳程式	訊號資料	節點位置
30	位置軟體 構件	顯示地圖位置	節點位置 經緯度	—
31		即時位置顯示	位置資料 經緯度	—
32		記錄日程路徑	位置資料 記錄資料	位置資料 經緯度
33	追蹤軟體 構件	追蹤物品位置	位置資料 經緯度	—
34		分析主要訊號	訊號資料	指令資料
35		發出閃爍訊號	固定訊號	記錄資料
36	紀錄軟體 構件	觀看歷史資料	紀錄狀態	資料
37		更新資料作業	追蹤資料	歷史資料
38		刪除資料	—	記錄狀態 資料庫資 料
39	警示軟體 構件	通知異常狀況	狀態碼 狀態資料	—
40		發出警示狀態	狀態資料	狀態資料
41		啟動警報防護	狀態資料	音效
42		解除警報防護	密碼	狀態資料