

透過 ArchiMate 正型方法定義 SBC 六大金律

Formalizing the SBC Six-principles via ArchiMate as a Formal Method

韓孟麒
Meng-Chyi Harn
德明財經科技大學
資訊科技系
副教授
harn@takming.edu.tw

陳文賢
Wen-Hsien Chen
德明財經科技大學
資訊管理系
教授
wchen@takming.edu.tw

巫宇昕
Yu-Hsin Wang
德明財經科技大學
資訊管理系
研究生
yendotest@gmail.com

摘要

本研究的目的，在於將中山大學趙善中博士所發明的結構行為合一論 (Structure-Behavior Coalescence, SBC) 中的六大金律正型化，並使用開放群組制定的標準 ArchiMate，來定義六大金律。目前結構行為合一論是以 Visio 作為它的繪圖輔助工具，而繪製出來的企業架構視圖，其視點元素都獨立，無法符合企業架構師的繪圖要求。其原因在於它無法做「對齊」和「重複繪製」的動作，當圖形重覆再使用時，每張圖都是各自獨立且無關聯性。所以造成無法很完善地將企業架構中的「靜態結構」與「動態行為」確切地表達出來，因而受限於應用。它雖有自己正型的表達方法，然而全都是使用文字敘述，而尚未使用數學的方式來定義。本研究之貢獻乃首創以 ArchiMate 來定義與繪出 SBC「靜態結構」與「動態行為」；且在繪製過程中，將構件、互動、操作名稱、輸出參數、輸入參數、...等，用 ArchiMate 的視點元素及關係來表達。最後，我們以實際案例為例，來加以說明與驗證。

關鍵詞：正型方法、企業架構、架構描述語言、結構行為合一論。

Abstract

The purpose of this research is to formalize and define the six-principles of Structure-Behavior Coalescence (SBC) invented by Dr. William S. Chao, National Sun Yet-sen University, by using the ArchiMate that is standardized by The Open Group. The view diagram of SBC six-principles drawn via the Diagram tool Visio cannot satisfy the needs of the enterprise architect due to the independent viewpoint element. The lacks of the alignment and duplication cause the problem to reuse the view diagram that has non-dependency. Therefore, it is hard and limited to represent the static structure and dynamic behavior of the enterprise architecture. Though the original SBC has its own formal representation, the definitions are specified with verbal description but mathematics. The contribution of our research is to originate the view diagrams of the static structure and dynamic behavior of the SBC via the viewpoint elements and relations of ArchiMate for the processes to describe the component, interaction, operation name, input and output parameters, and so forth. Finally, we take a real example to explain and verify our study.

Keywords: Formal Method, Enterprise Architecture, Architecture Description Language, Structure-Behavior Coalescence

1. 緒論

1.1 研究背景與動機

正型方法(Formal Method)是一種非常精確、不模糊的方法，它可以釐清學理中，不精確和模糊的部分，通常以數學符號和數學運算式來表示；使得學術論述，有明確的定義。正型可以解決學理不完整和不一致現象所衍生的問題；並可藉由定義和推理，來證明學理論述的正確性。本研究發現：企業架構論述正型化後，可建構出一套正型的數學符號和數學運算式體系。

目前 SBC 六大金律(The Six-principles of Structure-Behavior Coalescence)，有它自己一套的正型方法，但問題出在它的正型方法不夠嚴謹；而且，也沒有一套實現架構描述語言(Architecture Description Language, ADL)的繪圖輔助工具，可以把 SBC 六大金律的特色，完全展現出來。目前 SBC 六大金律，是以 Visio 作為替代的繪圖輔助工具，所繪製出來的企業架構視圖(View Diagram)，無法完全符合企業架構師(Enterprise Architect)的架構塑模要求。其原因在於：它無法做視點元素對齊(Alignment)和重複(Duplication)繪製的動作。當視點元素圖形欲重覆再使用時，因為每個視點元素圖都是各自獨立且無關聯性，所以產生了閱圖和繪圖的障礙。

1.2 研究目的

本研究的目的，乃在於將中山大學趙善中博士所發明的 SBC 中的六大金律，以正型方法來定義它，用 ArchiMate 的視點元素(Viewpoint Element)及關係(Relation)，來完整表達 SBC 六大金律的企業架構視圖。如此一來，任何使用者能快速又明確地找到他們所需的視點元素及關係，來完成應用領域的企業架構視圖，進而得使用符合架構描述語言的工具，達到最佳的表達方式。

1.3 研究方法

本研究之方法與步驟，共分五個階段來進行：

1. 確定研究目標、方法。
2. 蒐集探討相關文獻。
3. 以正型方法來定義 SBC 六大金律。
4. 實例驗證。
5. 結論。

1.4 研究貢獻

本研究之貢獻乃首創以 ArchiMate 來定義及繪出 SBC 靜態結構(Static Structure)與動態行為(Dynamic Behavior)；且在繪製過程中，將構件、互動、操作名稱、輸出參數、輸入參數等，用 ArchiMate 的視點元素及關係來表達。本研究透過正型化，將 SBC 六大金律經過嚴謹的正型化，再結合 ArchiMate，利用架構描述語言工具 Archi，繪出 SBC 靜態結構與動態行為。

2. 文獻探討與相關技術

2.1 企業架構

企業架構(Enterprise Architecture)最早由 John Zachman 在 1987 年所提出，由美國聯邦政府、美國國防部、及開放群組發揚光大[2][6]。國外的企業，使用企業架構做策略規劃(Strategic Planning)已是非常普遍的事；反觀國內，卻仍處於啟蒙階段。企業架構建構後，就為企業運作遵循的規範，因其共通性的參考視圖，企業可以將之發揮到最大綜效(Synergy)[2]。

2.2 TOGAF

TOGAF(The Open Group Architecture Framework)是一個架構規範，或稱之為架構框架，可以滿足不同企業的企業架構需求。TOGAF 將企業架構分為四類：業務架構(Business Architecture)、應用架構(Application Architecture)、資料架構(Data Architecture)與技術架構(Technical Architecture)，其發展內涵，全部定義於 TOGAF 的架構開發方法(Architecture Development Method, ADM)中[11]。

TOGAF ADM，係建構企業架構的過程，實質上它定義了一個完整的生命週期，可用來規劃、設計、實現和管理企業架構。TOGAF ADM 的架構開發週期(Architecture Development Cycle)，一共有十個階段(Phases)，如圖 1 所示；列示如下[10][11]：

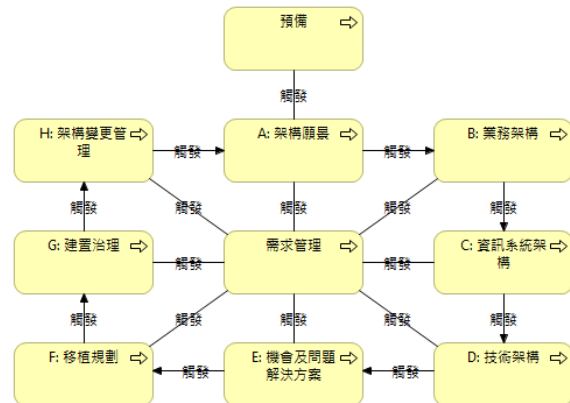


圖 1 架構開發週期

1. 預備(Preliminary)
2. 階段 A：架構願景(Architecture Vision)
3. 階段 B：業務架構(Business Architecture)
4. 階段 C：資訊系統架構(Information Systems Architectures)
5. 階段 D：技術架構(Technology Architecture)
6. 階段 E：機會及問題解決方案(Opportunities and Solutions)
7. 階段 F：移植規劃(Migration Planning)
8. 階段 G：建置治理(Implementation Governance)
9. 階段 H：架構變更管理(Architecture Change Management)

10. 需求管理(Requirements Management)

2.3 SBC 架構(SBC Architecture)

為了尋找企業架構的本質，趙博士發明了結構行為合一(SBC)架構，他認為：「架構是由結構與行為合一而成」，即[1][2][7]：

$$Architecture = Structure + Behavior$$

且行為是被結構所負載(即：*Behavior is loaded on structure.*)，另外還得加上「其他觀點」來做補充。也就是說，在建構企業系統架構過程中，先有物件的靜態結構，才会有物件的動態行為。

趙博士認為：系統學 2.0 使用了 SBC 架構描述語言(SBC Architecture Description Language, SBC-ADL)，來完成對「系統」的定義。SBC 架構描述語言，係由「六大金律」的圖形所組成。

結構行為合一(SBC)架構，是具有「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)必須合一，因此其核心思維為[3][4][5]：

$$架構 = 結構 + 行為 + 其他觀點$$

趙博士所發明的 SBC 理論內容，乃以結構、行為、其他等三個觀點為本體，說明企業架構具多重觀點(Multiple View)，如圖 2 所示。

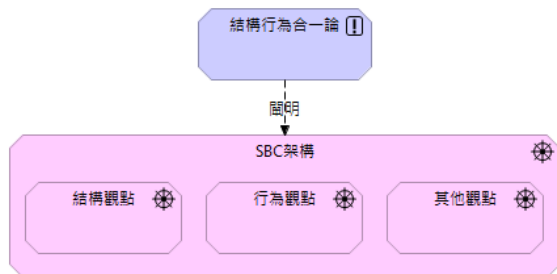


圖 2 SBC 架構之多重觀點

「結構行為合一」的意思，就是在「架構」的建構過程中，須先將「靜態結構」與「動態行為」合而為一，再結合「其他觀點」，作為補充。他認為三者須相輔相成，即[1][2][8]：

$$Architecture = Structure + Behavior + Other Views$$

SBC 架構，須藉助 SBC 架構描述語言來描述，並以 SBC 六大金律的圖形來表達，六類的圖如圖 3 所示[2]，分別為：

1. 架構階層圖(Architecture Hierarchy Diagram, AHD)
2. 規範圖(Framework Diagram, FD)
3. 構件操作圖(Component Operation Diagram, COD)

4. 構件連結圖(Component Connection Diagram, CCD)
5. 結構行為合一圖(Structure-Behavior Coalescence Diagram, SBCD)
6. 互動流程圖(Interaction Flow Diagram, IFD)。



圖 3 SBC 六大金律

SBC 架構可以完完全全地將企業架構領域內的靜態結構與動態行為描述清楚，而且其所有「構件」(Component)，也都可以被靜態結構與動態行為所描述。

SBC 架構的「企業架構動態行為塑模準則」，係配合 SBC 六大金律所律定的法則，一共有七個步驟，如圖 4 所示[2]，說明如下：

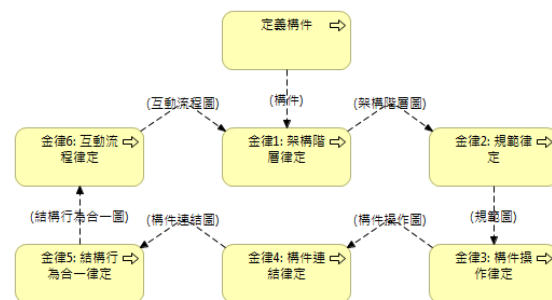


圖 4 建構 SBC 架構的七個步驟

1. 定義構件

首先以正向工程(Forward Engineering)或反向工程(Reverse Engineering)的方式，從整體觀(Integrity View)找出所有企業架構領域內的相關構件。

2. 金律 1：架構階層律定

將所有構件給予分類，以顆粒觀(Granularity View)繪出架構階層圖。

3. 金律 2：規範律定

將分類好的構件，依階層繪出規範圖。

4. 金律 3：構件操作律定

於規範圖內之構件，找出操作要點(Operational Slot)，繪製構件操作圖。

5. 金律 4：構件連結律定

利用構件操作圖，找出構件連結方式，繪出構件連結圖。

6. 金律 5：結構行為合一律定

利用構件連結，繪出結構行為合一圖。

7. 金律 6：互動流程律定

利用結構行為合一圖，繪製互動連結圖，將企業架構領域內所有相關靜態結構構件的動態行為繪製出來。

由於「企業架構動態行為塑模準則」及六大金律所建構的架構，可以成為其他架構的先導架構(Advanced Architecture)。因此，所有的企業架構建構工作，都可以從 SBC 的「結構行為合一」核心思維開始，進而勾勒「動態行為」。也就是說，為了降低企業架構建構的複雜度(Complexity)，我們可以先以「企業架構動態行為塑模準則」，配合 SBC 六大金律的產出，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。

2.4 相關技術

1. ArchiMate

ArchiMate 是 The Open Group 發展出來的一個標準，係由不同的工具開發商和顧問諮詢公司一同支持的塑模語言[10]。經過十幾年的驗證，這個圖形化架構描述語言(ADL)，已成為 TOGAF 最好的圖形工具塑模語言。ArchiMate 提供了一個一般化的語言(Generic Language)，以描述業務過程(Business Process)、組織結構(Organization Structure)、資訊流(Information Flow)、資訊系統(Information System)及技術基礎建設(Technology Infrastructure)的建構和運作[12]。

2. Archi

Archi 是由 The Open Group 所認可的塑模工具，主要使用對象係以企業架構師和塑模師為主。它是一個免費的跨平台塑模工具，滿足了大多數企業架構師和相關的使用者的需求，並且被全球著名的銀行界、保險公司、EA 顧問、培訓機構等各種行業及大學學生和研究生所使用，更是全世界以 ArchiMate 為基礎且最流行的建模工具。

3. 正型化

正型化(Formalizing)了一個特定的問題，通常被稱為正型化規範。所謂的成型(Well-formed)又稱之為正型、成形、正形、正規、... 等，它是工程(Engineering)界的普遍現象，以技術(Technology)為主，通常都是以定義(Definition)、定理(Theorem)、公式(Formula)、法律(Law)、規章(Principle)、規則(Rule)、準則(Discipline)、標準(Standard)、證照(Certificate)、品質(Quality)、成熟度(Maturity)... 等為手段[9]，運用數學的方式來定義出系統特性的技術方法。這樣的正型化方法訂定出較為硬性(Hard)標準，進而減少認知上誤解所產生出來的問題，以便在該標準中來進行規劃、開發和驗證，這系統架構大大地增加了論述的成功性。正型化方法可以用來補充不正型的方法，可在整體系統開發過程中來使用[9]。到目前為止，正型化方法，已展現出它的規範和驗證實力。

3. SBC 六大金律正型化

SBC 六大金律分別是：架構階層圖、框架圖、構件操作圖、構件連結圖、結構行為合一圖、互動流程圖 [1][2][8]，以 ArchiMate 之部分視點元素及關係定義如下：

Definition (Structure-Behavior Coalescence) The Structure-Behavior Coalescence is a set of architectures $SBC = AHD \cup FD \cup COD \cup CCD \cup SBCD \cup IFD$ where

1. *AHD* is a set of views for architecting the architecture hierarchy diagram by using viewpoint elements: *business actors, application components and work packages*;
2. *FD* is a set of views for architecting the framework diagram by using viewpoint elements: *business actors and application components*;
3. *COD* is a set of views for architecting the component operation diagram by using viewpoint elements: *business actors, application components and application interfaces*, and relations: *junctions*;
4. *CCD* is a set of views for architecting the component connection diagram by using viewpoint elements: *business actors and application components*, and relations: *associations*;
5. *SBCD* is a set of views for architecting the structure-behavior coalescence diagram by using viewpoint elements: *business events, business actors and application components*, and relations: *triggering flows*; and
6. *IFD* is a set of views for architecting the interaction flow diagram by using viewpoint elements: *business events, business actors, business functions, application components and application collaborations*, and relations: *triggering flows*.

本研究定義的結構行為合一論，是由架構 *SBC* 所成的集合，而 $SBC = AHD \cup FD \cup COD \cup CCD \cup SBCD \cup IFD$ 。其中，*AHD* 為架構階層圖，*FD* 為框架圖，*COD* 為構件操作圖，*CCD* 為構件連結圖，*SBCD* 為結構行為合一圖，*IFD* 為互動流程圖。建構 *AHD*、*FD*、*COD*、*CCD*、*SBCD*、*IFD* 的目的，乃是藉圖形化的表現，可以很簡單、很清楚地描繪出整個複雜的企業架構模型。

經由六大金律，本研究找出了表達 SBC 六大金律的 Viewpoint Elements 及 Relationships，並使其合理化，用 Lisp-like 表達如下：

1. 在架構階層圖中，運用之 Viewpoint Elements 為：
(Business (Business_Actor))
(Application (Application_Component))
(Implementation_and_Migration (Work_Package))

2. 在框架圖中，運用之 Viewpoint Elements 為：
 (Business (Business_Actor))
 (Application (Application_Component))

3. 在構件操作圖中，運用之 Viewpoint Elements 與 Relationships 為：
 (Business(Business_Actor Business_Function))
 (Application(Application_Component Application_Interface))
 (Relationships (Junction))

4. 在構件連結圖中，運用之 Viewpoint Elements 與 Relationships 為：
 (Business (Business_Actor))
 (Application (Application_Component))
 (Relationships (Association))

5. 在結構行為合一圖中，運用之 Viewpoint Elements 與 Relationships 為：
 (Business (Business_Event Business_Actor))
 (Application (Application_Component))
 (Relationships (Triggering Flow))

6. 在互動流程圖中，運用之 Viewpoint Elements 與 Relationships 為：
 (Business(Business_Event Business_Actor Business_Function))
 (Application(Application_Component Application_Collaboration))
 (Relationships (Triggering Flow))

4. 實例驗證

以國立中山大學資管研究所研究生周賢輝碩士論文：架構導向法院業務管理模型之研究為例：
 1. 架構導向法院業務管理模型之架構階層圖，如圖 5 所示。

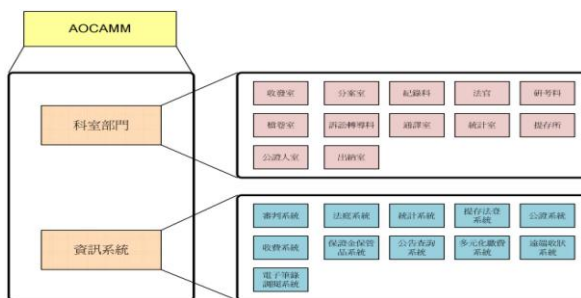


圖 5 法院業務管理模型之架構階層圖

經本研究正型化後，以 ArchiMate 表達之架構導向法院業務管理模型之架構階層圖，如圖 6 所示。

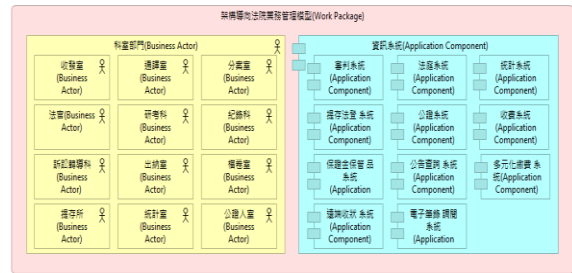


圖 6 以 ArchiMate 表達之法院業務管理模型之架構階層圖

2. 架構導向法院業務管理模型之框架圖，如圖 7 所示。

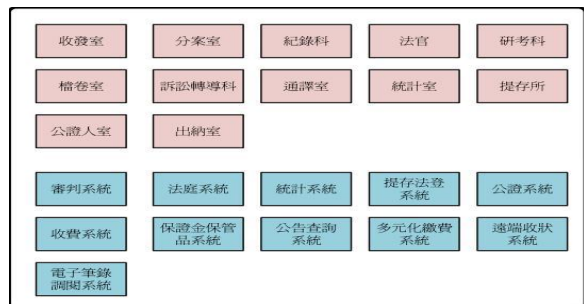


圖 7 法院業務管理模型之框架圖

經本研究正型化後，以 ArchiMate 表達之架構導向法院業務管理模型之框架圖，如圖 8 所示。

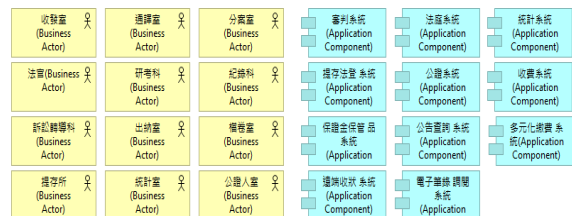


圖 8 以 ArchiMate 表達之法院業務管理模型之框架圖

3. 架構導向法院業務管理模型之構件操作圖，如圖 9 所示。

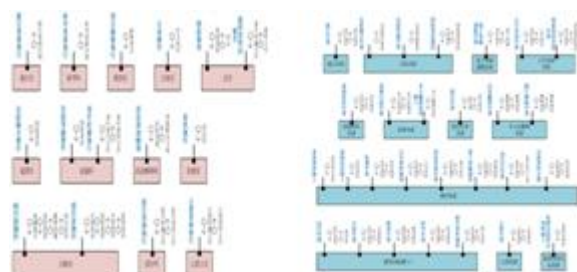


圖 9 法院業務管理模型之構件操作圖

經本研究正型化後，以 ArchiMate 表達之架構導向法院業務管理模型之構件操作圖，如圖 10 所示。

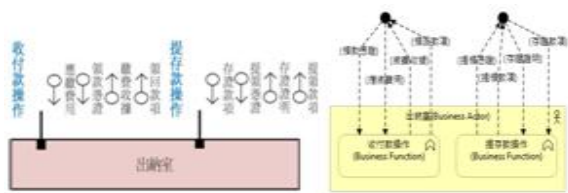


圖 10 以 ArchiMate 表達之法院業務管理模型之構件操作圖

4. 架構導向法院業務管理模型之構件連結圖，如圖 11 所示。

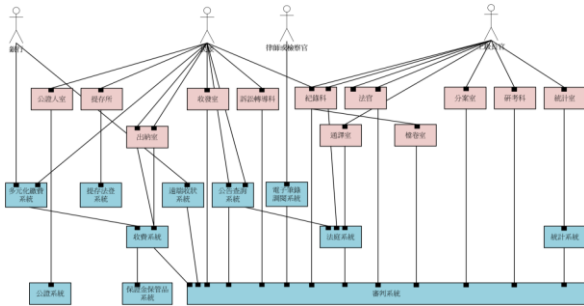


圖 11 法院業務管理模型之構件連結圖

經本研究正型化後，以 ArchiMate 表達之架構導向法院業務管理模型之構件連結圖，如圖 12 所示。

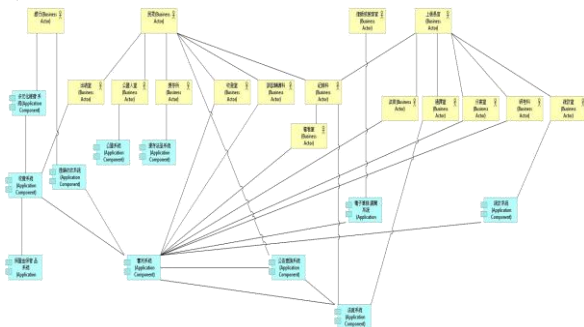


圖 12 以 ArchiMate 表達之架構導向法院業務管理模型之構件連結圖。

5. 架構導向法院業務管理模型之結構行為合一圖，如圖 13 所示。

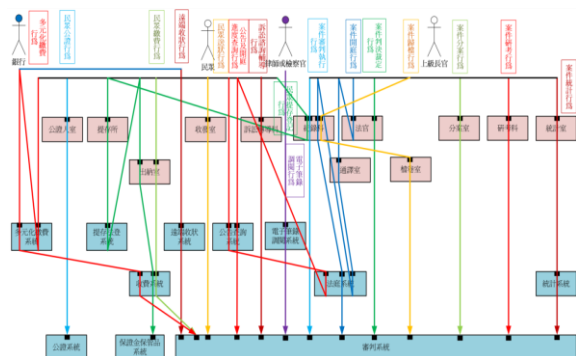


圖 13 法院業務管理模型之結構行為合一圖

經本研究正型化後，以 ArchiMate 表達之架構導向法院業務管理模型之結構行為合一圖，如圖 14 所示。

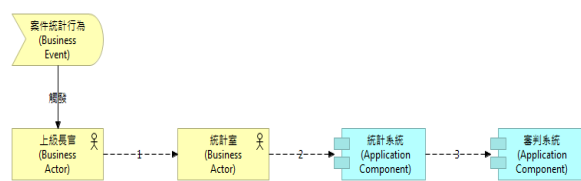


圖 14 以 ArchiMate 表達之法院業務管理模型之結構行為合一圖

6. 架構導向法院業務管理模型之互動流程圖，如圖 15 所示。

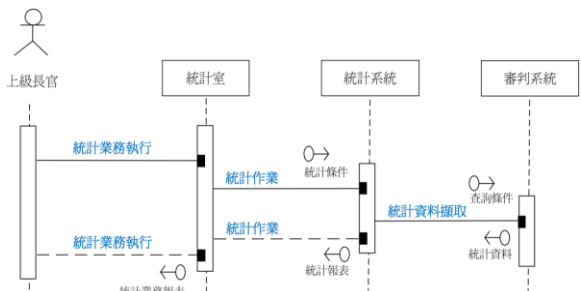


圖 15 架構導向法院業務管理模型之互動流程圖

經本研究正型化後，以 ArchiMate 表達之架構導向法院業務管理模型之互動流程圖，如圖 16 所示。

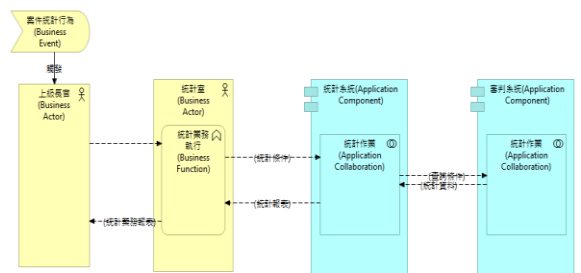


圖 16 以 ArchiMate 表達之法院業務管理模型之互動流程圖

5. 結論

SBC 六大金律經由離散數學的集合或定義方式正型化後，完整闡述了結構行為合一論的原意；運用 ArchiMate 企業架構塑模語言繪圖後，展現了企業架構視圖的精髓。正型化乃是利用數學的方式，產生精確的定義，以便解決各種分歧等問題及整合更細匿賦的工作；同時，它也可以當作一種系統塑模的基礎。本研究主要在說明這個方法的重要性，並強調運用 ArchiMate 的視點元素與關係，來組成 SBC 六大金律。所繪製的視圖，其視點元素與關係，分別來自 SBC 完整的構件操作與外部互動行為。而 SBC 架構視圖的主要構件及操作，可以具體的被分類及擴充為架構階層圖、框架圖、構件操作圖、構件連結圖、結構行為合一圖、互動流程圖等。

本研究未來研究方向，可以朝先依 SBC 六大金律，再依 ArchiMate 視點元素與關係，來進行繪

製出企業架構視圖。SBC 六大金律被 ArchiMate 視點元素規與關係範後，即能方便檢核 SBC 六大金律的完整性及一致性。

參考文獻

- [1] 趙善中、趙薇、趙鴻 (2008)，*系統架構學：軟體架構，企業架構，知識架構，思考架構*，科技圖書股份有限公司出版。
- [2] 韓孟麒、趙善中 (2013)，「企業架構師」師資培訓先導班上課講義，中華企業架構師學會。
- [3] 趙善中、孫述平、韓孟麒 (2013)，*企業架構－企業營運創新的基石*，義守大學系統架構研訓中心。
- [4] 趙善中、孫述平、韓孟麒 (2013)，*系統思考法－使用 SBC 架構*，義守大學系統架構研訓中心。
- [5] 趙善中、孫述平、韓孟麒 (2013)，*系統學－使用 SBC 架構描述語言*，義守大學系統架構研訓中心。
- [6] Scot A. Bernard, EA3: *An Introduction to Enterprise Architecture*, AuthorHouse, 2008.
- [7] William S. Chao, Shuh-Ping Sun, and Meng-Chyi Harn, *Systems Architecture: Hardware, Software, Enterprise, Knowledge, Thinking*, Research and Training Center of System Architecture, I-Shou University, 2013.
- [8] William S. Chao, *Software Architecture: SBC Architecture at Work*, National Sun Yet-sen University, 2011.
- [9] Meng-Chyi Harn, "A Formal Model for C4ISR Systems," *The Second Taiwan Conference on Software Engineering*, Taipei, Taiwan, June 9-10, 2006.
- [10] Henk Jonkers, Iver Band, and Dick Quartel, *ArchiSurance Case Study*, The Open Group, 2012.
- [11] The Open Group, *TOGAF Version 9: The Open Group Architecture Framework (TOGAF)*, 2009.
- [12] The Open Group, *Archi - ArchiMate Modelling, User Guide, Version 2.3*, 2009.