

國防企業架構與 C4ISR 系統的正型化關係

The formalization relationship between defense enterprise architecture and C4ISR systems

韓孟麒 副教授

德明財經科技大學資訊科技系

Meng-chyi Harn

Department of Information Technology,
Takming University of Science and Technology

Email: harn@takming.edu.tw

摘要

目前企業的運行，已慢慢地朝向正型化的方向發展，走向正型化的第一步，就是要將其獨特的企業架構建立起來，有了完整的企業架構，企業電腦化才有成功的可能。國防企業是全國最大的企業，它的分支十分的細緻，若是沒有完整的國防企業架構描述，要去實踐「指揮與管制」的自動化，簡直是不可能。現在國軍積極發展的 C4ISR 系統，就是要藉助國防企業架構來正型化其功能，美國國防部所提供的 DoDAF 標準架構方法，就是數位化戰場中建置 C4ISR 系統的規範。

關鍵字：正型方法、構型管理、企業架構、C4ISR 系統、DoDAF

一、緒論

正型方法(Formal Method)是實現正型理論(Formalization)的方法，所形成的論述稱之為正型主義(Formalism)[8]。企業的正型化(Formalize)其實就是電腦化(Computerize)的第一步，也是企業進入二十一世紀在資訊科技推波助瀾下的必然趨勢，它是將企業從不成型(Ill-formed)中轉成正型(Well-formed)的過程。本研究是從正型理論的角度來看國防事業正型化的走向，國防事業在追求效率與效能的過程中，一定要將相關的國防企業架構定義出來，有了清楚的架構，才能將國防事務電腦化。

現在國防事務電腦化最典型的作為就是藉由 C4ISR 系統將指揮管制走向自動化，美國及先進的國家都致力於發展其獨特的 C4ISR 系統，我們國家也是一樣[7]。但是我們國家的 C4ISR 系統並不多，有鑑於國防自主的重要，我們嘗試於自我研發，後來才發現到許多技術都需要引進國外的高科技。這些高科技中，除了精良的武器、導航、監視與偵察等系統外，就是要靠資訊系統對這些子系統的整合。由於一個 C4ISR 系統有獨特的構型管理(Configuration Management)[4]，所以其架構就顯得重要，這其中的關鍵就是建構的方法論，有了明確的方法論，才不會在建構時找不到藍圖。

美國國防部在開發 C4ISR 系統時，大都先將其

企業架構建立好，再建案開發，使開發者有所依循，也使需求方可以按照正型的方法來描述系統的使用者需求。這個正型的方法就是美國國防部架構規範 (Department of Defense Architecture Framework, DoDAF)，它是將傳統的系統分析[11]與物件導向技術[1]結合在一起，由美國國防部所律定的規範，目前是世界有能力發展 C4ISR 系統國家的公認標準[2]。

本文在說明正型化的定義、國防企業架構與 C4ISR 系統及其正型化的關係。

二、正型化的定義

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[6]。

正型後的系統較為硬性(Hard)且比較沒有彈性，它是屬於較為死板的形式邏輯(Formal Logic)或形式語言(Formal Language)所呈現出來的機器(Machine)行為，運用到企業活動上稱之為資訊模式(Information Model, IM)，能夠利用電腦這種數位媒介(Digital Medium)產生出數位資訊(Digital Information)，例如多媒體(Multi-medium)型式的數位文件(Digital Documentation)。

所謂的不成型(Ill-formed)又稱之為不正型、不成形、不正形、不正規、非正型、非成形、非正形、非正規、...等，它是商業(Business)界的普遍現象，以管理(Management)為主，而管理是一門藝術(Art)，公說公有理，婆說婆有理，沒有一定的基調。

正型前的系統較為軟性(Soft)且比較有彈性，它是屬於較為靈活的人類(Man)思維所呈現出來的非機器(Non-machine)行為，運用到企業活動上稱之為企業模式(Business Model, BM)，能夠利用抽象知識(Abstract Knowledge)產生出電腦化前的企業決策[6]。

以國家為例，歐美國家是一個「正型」的國家，非洲國家是一個「非正型」的國家，全世界的

國家都朝著正型國家的方向走，這是個趨勢，擋也擋不住，我國就在這個轉換的過程中，所以美國稱之為已開發的國家，非洲國家稱為未開發的國家，而我們中華民國稱之為開發中的國家，也是一個由「非正型」到「正型」的過渡型國家。

以人為例，軍人是一種「正型」的人，因為受過正規的訓練，一個口令一個動做，所中共稱其軍隊叫正規軍；而普通老百姓是一種「非正型」的人，因為沒受過正規的訓練，比較散漫。

而知識份子及專業人員也是一種「正型」的人，因為受過良好教育和訓練；剛出生的嬰兒或野蠻人是一種「非正型」的人，因為沒受過良好教育和訓練，比較原始，人類都是朝著正型人的方向走，這是天性，擋也擋不住，學生就在這個轉換的過程中，所以教育、訓練、獲得證照就是一種正型化的過程，目的在「塑造」出一個具有知識且有用的專業人才。

以企業為例，有資訊科技輔助的企業是一個「正型」的企業，沒有資訊科技輔助的企業是一個「非正型」的企業，全世界的企業都朝著正型企業的方向走，這也是個趨勢，擋也擋不住，電腦化的企業就在這個轉換的過程中；但企業不能太過於正型，也就是說不能太過於依賴資訊科技，因為人性的因素在企業中仍扮演了一個重要的角色，人類不可能刻意的缺席，而完全由電腦來取代。

三、國防企業架構與 C4ISR 系統

「企業架構」(Enterprise Architecture, EA)的研究領域十分的廣泛，凡是有關各領域架構的靜態結構 (Static Architecture) 與動態建構 (Dynamic Construction)，都涵蓋進去，它的涵蓋範圍包括了：政府、國防、產業及家電的應用領域；因此，只要是企業中任何「系統化」的架構研究與相關資訊科技的應用，都是我們所關注的。國防企業架構 (Defense Enterprise Architecture) 是「企業架構」的一環，而國防部又是政府部門最大的企業，它有完整的靜態結構與迫切的動態建構需求。國防企業的正型化是要將不正型的人為作為 (Human Behavior) 轉換為正型的電腦作為 (Computer Behavior)，也就是美國海軍研究院 Luqi 教授在 [8] 所說的，要將「不正型的濕」(Informal Wet) 轉變為「正型的乾」(Formal Dry)。

在軍事上，所謂「不正型的濕」就是國防企業未實現電腦化的極端現象，例如二次大戰以前的軍隊；所謂「正型的乾」就是國防企業實現電腦化的極端現象，例如目前數位化戰場的 C4ISR 系統，這極端的二者都不是好現象。但是為了在戰場中求勝，並追求效率與效率能的戰場綜效 (Synergy)，國防企業大都從「不正型的濕」走向「正型的乾」。第三世界的軍隊由於無能力藉助電腦化來作戰，所以是最「不正型的濕」；美國的軍隊由於是邁向電腦化的最前端，所以是最「正型的乾」，我們國家和其他先進國家一樣，都是從「不正型的濕」邁向

「正型的乾」。在邁向「正型的乾」過程中，國防企業架構的描述，是發展 C4ISR 系統的關鍵所在。

C4ISR 系統是一個「系統中的系統」(System of Systems, SoS)，它本來就是一個非常堅固的架構，它是由五類的戰場物件 (Battlefield Objects) 所組成，即：軍事人員 (Military People)、武器系統 (Weapon Systems)、導航系統 (Navigation Systems)、平台感應器 (Platform Sensors) 及通訊鏈路 (Communication Links) [3][4]。目前美國國防部在描述一個 C4ISR 系統時，大都要藉助 DoDAF 來完成，並從「不正型濕」的作戰觀點 (Operational View)，經由系統觀點 (Systematic View)，到「正型乾」的技術觀點 (Technical View)；將人性面較強的不正型行為，轉換為機器面較強的正型行為 [2]。若一個戰場，全然是人性面較強的不正型行為，這時打仗只靠人，全由人來指揮，由人來作肉搏戰，這不是一個好現象，因為在二十一世紀的數位化戰場中，單靠傳統式的打法，人類是打不過有 C4ISR 系統的軍隊。反之，一個數位化戰場，若全然走向機器面較強的正型行為，這時打仗就不需要靠人類了，全由機器來指揮，人無用武之地，這不是一個好現象，因為人類世界可能會受控於機器人，並被機器人所摧毀。我們認為，最佳的策略是：依照現況，發展適合自己國家的 C4ISR 系統。

四、DoDAF

DoDAF 是由 26 項產品所組成 [2]，利用這些產品，就足以說明國防企業架構與 C4ISR 系統的正型化關係。國軍在建軍備戰的過程中，把不正型的作戰構想，轉換為較為正型的白紙黑字，這就是在做正型化的工作，最明顯的例子就是各軍種「作戰要綱」的訂定。由於各軍種的「作戰要綱」是文字描述，解讀的結果又是千變萬化，莫衷一是，我們很難依其條文去實踐數位化戰場的目的。而這些「作戰要綱」的條文，是某一特殊作戰領域的軍事專家依據作戰時的寶貴經驗撰寫而成，但根本沒有用正型方法來描述，使得 C4ISR 系統的開發增加了許多困難度。況且 C4ISR 系統的開發是要跨國合作，許多高科技需要美國的支援，若沒有一個共通的語言，是描述不清楚的。就連美國本國的共通語言—英語，對美國人來說，還是對電腦化幫助有限，原因是它並不是一個正型化過程中，表達使用者需求的最好方法；否則就不會在軟體工程中，有許多的正型方法出現，例如 1979 年 Constantine 及 Yourdon 的結構化設計 (Structured Design)、1979 年 Gane 及 Sarson 的結構化系統分析 (Structure System Analysis)、1983 年 Jackson 的系統開發方法 (System Development Method)，一直到 1990 年代 Robinson、Booch、Rumbaugh 的物件導向法 (Object-oriented Method) [1][10]。

在我國發展 C4ISR 系統的過程中，最難的是使用者需求的獲得與確認，若是僅憑「作戰要綱」是不夠的，更何況要由中文轉譯成英文，再讓美國人

理解我們的作戰構想，這是會有誤差的。若世界上有一種方法，或一種規範，可以讓全世界的人都可以上面溝通使用者需求，那麼就可以減少能力成熟度模式整合－需求模組(Capability Maturity Model Integration－Acquisition Module, CMMI－AM)中所言的供給方(Supplier)與需求方(Acquirer)認知的差距[5]。根據我們的研究，這個方法就是正型方法，這個規範就是 DoDAF。

DoDAF 是結合了傳統的結構化方法及現代的物件導向法製訂出來的規範[1][2][11]。在未用 DoDAF 描述前，國防企業架構本來就存在著，只是沒有用正型方法表達出來而已；C4ISR 系統是電腦化後的產物，兩者的關係全都在 DoDAF 的產品中。所以要了解 DoDAF，就必須要先去了解軟體工程中的結構化方法及物件導向法，這樣才有可能將國防企業架構正型化出來。

在DoDAF中，除了架構中提供摘要資訊與標準辭彙的綜合觀點(All View, AV)外，律定出三種描述國防事務的觀點：作戰觀點(Operational View, OV)、系統觀點(Systematic View, SV)及技術觀點(Technical View, TV)[7]；茲將各觀點定義如下：

Definition 1 (All View) An *all view* is a set of architectures $AV = AV-1 \cup AV-2$ where

1. *AV-1* is a set of *views* for architecting the overview and summary information by using *texts* and *tables*, and
2. *AV-2* is a set of *views* for architecting the integrated dictionary by using a *structured tool: data dictionary*, and *tables*.

Definition 2 (Operational View) An *operational view* is a set of architectures $OV = OV-1 \cup OV-2 \cup OV-3 \cup OV-4 \cup OV-5 \cup OV-6a \cup OV-6b \cup OV-6c \cup OV-7$ where

1. *OV-1* is a set of *views* for architecting the high-level operational concept by using *graphics*, *use-case charts* or *deployment charts* with *properties*,
2. *OV-2* is a set of *views* for architecting the operational node connectivity by using *deployment charts* or *data flow diagrams* with *properties*,
3. *OV-3* is a set of *views* for architecting the operational information exchange by using *matrixes* with *properties*,
4. *OV-4* is a set of *views* for architecting the organizational relationships by using *class charts* or *hierarchical trees* with *properties*,
5. *OV-5* is a set of *views* for architecting the operational activity model by using *use-case charts*, *collaboration charts*, *sequence charts*, *activity charts* or *IDEF0* with *properties*,
6. *OV-6a* is a set of *views* for architecting the operational rules model by using *decision rules* or *IDEF3* with *properties*,

7. *OV-6b* is a set of *views* for architecting the operational state transition by using *state charts* with *properties*,
8. *OV-6c* is a set of *views* for architecting the operational event trace by using *sequence charts* with *properties*, and
9. *OV-7* is a set of *views* for architecting the logical data model by using *class charts* or *entity-relationship diagrams* with *properties*.

Definition 3 (Systematic View) A *systematic view* is a set of architectures $SV = SV-1 \cup SV-2 \cup SV-3 \cup SV-4 \cup SV-5 \cup SV-6 \cup SV-7 \cup SV-8 \cup SV-9 \cup SV-10a \cup SV-10b \cup SV-10c \cup SV-11$ where

1. *SV-1* is a set of *views* for architecting the systems interface by using *deployment charts* and *component charts* with *properties*,
2. *SV-2* is a set of *views* for architecting the systems communications by using *deployment charts* or *component charts* with *properties*,
3. *SV-3* is a set of *views* for architecting the systems-systems matrix by using *matrixes* with *properties*,
4. *SV-4* is a set of *views* for architecting the systems functionality by using *class charts*, *object charts*, *activity charts*, *data flow diagrams* or *hierarchical trees* with *properties*,
5. *SV-5* is a set of *views* for architecting the operational activity to systems function traceability by using *matrixes* with *properties*,
6. *SV-6* is a set of *views* for architecting the systems data exchange by using *matrixes* with *properties*,
7. *SV-7* is a set of *views* for architecting the systems performance parameters by using *matrixes* with *properties*,
8. *SV-8* is a set of *views* for architecting the systems evolution by using *graphics* with *properties*,
9. *SV-9* is a set of *views* for architecting the systems technology forecast by using *tables* with *properties*,
10. *SV-10a* is a set of *views* for architecting the systems rules model by using *decision rules* or *IDEF3* with *properties*,
11. *SV-10b* is a set of *views* for architecting the system state transition by using *state charts* with *properties*,
12. *SV-10c* is a set of *views* for architecting the system event trace by using *sequence chart* with *properties*, and
13. *SV-11* is a set of *views* for architecting the physical schema by using *component charts* or *entity-relationship diagrams* with *properties*.

Definition 4 (Technical Views) A *technical view* is a set of architectures $TV = TV-1 \cup TV-2$ where

1. *TV-1* is a set of *views* for architecting the technical architecture profile by using *tables* with *properties*, and
2. *TV-2* is a set of *views* for architecting the standards technology forecast by using *tables* with *properties*.

Definition 5 (DoD Architecture Framework) A

DoD Architecture Framework is a tuple *DoDAF* = (*AV*, *OV*, *SV*, *TV*) where

1. *AV* is a set of *all views* for architecting a C4ISR system,
2. *OV* is a set of *operational views* for architecting a C4ISR system,
3. *SV* is a set of *systematic views* for architecting a C4ISR system, and
4. *TV* is a set of *technical views* for architecting a C4ISR system.

五、結論

正型方法的研究早在有數學的時代就有了，其實也是一種系統塑模的方法。本研究主要是說明這個方法的重要性，並強調一個企業若要走向電腦化，就必須要先做正型化的準備。國防企業也是一樣，它必須走向自動化，因為現在是數位化戰場的時代。我們國軍目前正要將國防事務電腦化，指揮與管制的自動化有著太多的問題要描述，若只光靠文字是不夠的，還必須藉助結構化工具(Structured Tools)、IDEF 工具(Integrated Computer-Aided Manufacturing DEFINition Tools, ICAM DEFINition Tools)、實體關係圖(Entity Relationship Diagrams, ER Diagrams)、層級樹(Hierarchical Trees)、表格(Tables)、矩陣(Matrixes)、本文(Texts)及圖(Graphics)等正型化的工具。DoDAF 的產品對企業架構有著系統性的描述，只要受過訓練的人就會懂得使用，並懂得解讀。一個國防企業架構就在使用 DoDAF 的產品中被建構，一個 C4ISR 系統就在解讀 DoDAF 的產品中被建置出來。

參考文獻

- [1] P. P. Biennu, D. Kim and A. H. Levis, *C4ISR Architectures III: An Object-Oriented Approach to Architecture Design*, C3I Center, George Mason University, Fairfax, VA., April 2000.
- [2] DoD Architecture Framework Working Group, *DoD Architecture Framework Version 1.0 Deskbook*, February 9, 2004.
- [3] M. Harn, V. Berzins, Luqi, and W. Kemple, "Evolution of C4I Systems," *Proceedings of 1999 Command and Control Research and Technology Symposium*, United States Naval War College, Newport, Rhode Island, June 29 - July 1, 1999, pp.1361-1380.
- [4] M. Harn, S. Hsu, V. Berzins, and Luqi, "Battlefield Object Control via Internet Architecture," *Proceedings of 2002 Command and Control Research and Technology Symposium*, United States Naval Postgraduate

School, Monterey, California, June 11 - June 13, 2002.

- [5] M. Harn, Shu-hsien Lee, Chia-ling Hu and Cheng-hang Wang, "Applying CMMI-AM to a C4ISR Project from the Buyer's View," *Proceedings of 2006 Command and Control Research and Technology Symposium*, San Diego, California, June 20-20, 2006.
- [6] M. Harn, "A Formal Model for C4ISR Systems," *The Second Taiwan Conference on Software Engineering*, Taipei, Taiwan, June 9-10, 2006.
- [7] M. Harn and Cheng-hang Wang, "The DoDAF Support for a Huge-grain C4ISR System," *The Third Taiwan Conference on Software Engineering*, Taichung, Taiwan, June 8-9, 2007.
- [8] Luqi, "Formal Methods Promises and Problems," *IEEE Software*, January 1997, pp. 73-85.
- [9] S. Malerud, E. H. Feet, and U. Thorsen, "A Method for Analyzing Command and Control Systems," *Command and Control Research and Technology Symposium*, Naval Postgraduate School, California, June 29 - July 1, 1998, pp. 231-240.
- [10] I. Sommerville, *Software Engineering*, Addison-Wesley, 2004.
- [11] L. W. Wagenhals, I. Shin, D. Kim, and A. H. Levis, *C4ISR Architectures II: A Structured Analysis Approach to Architecture Design*, C3I Center, George Mason University, Fairfax, VA., April 2000.