

〈後記〉

計算作為理解的模型和思考的語言

The method is more important than the discovery, because the correct method of research will lead to new, even more valuable discoveries.

方法比發現還更重要，因為正確的研究方法將會帶來新的、甚至更有價值的發現。

Lev Landau

Johann Sebastian Bach(中譯：巴哈、巴赫, 1685-1750)在18世紀初葉創作的「平均率(The Well-Tempered Clavier)」，可以說是音樂界的“牛頓定律”。在這兩套各有24首前奏曲與24首賦格的偉大作品中，Bach不但用大小調在五度圈(circle of fifths)上的杯觥交錯，從混沌未開的聲音世界中劃出一條條清晰的尺規，更是透過和弦轉換的變幻莫測及聲音線的交織舞步，傾訴與吶喊出心中的共鳴與悸動。

音樂理論中的“規則”既能刻畫我們對音樂合理性的感受，又能讓人們在其基礎之上建立語言描述和分享內心的觸動。如此在“模型”和“語言”中的巧妙平衡，是否也正是本書的主角“計算”扮演的角色呢？

將近一百年前，Turing採用機械論的角度，開啟了對計算這個概念的數學研究。在本書的第一部份中，我們看到計算如何從簡單的規則，在經過多年研究後漸漸演化和發展出各種創新的模型及概念。電腦科學在科技上的蓬勃發展提供了其他領域大量新的研究工具，同時我們在本書的第二和第三部份中見識到計算的視角如何在物理和生命世界中提供不同的觀點。透過這些例子，筆者想要強調計算不只可以作為一個方便的工具，它更能有如音樂理論一樣，作為“理解的模型”和“思考的語言”。

「每個科學方法論某種程度上不都是同時有模型和語言嗎？計算有什麼特別的見長之處？」

我認為計算中的「composability(合成與創作性)」和「modularity(模組性)」，是其同時作為模型和語言的優勢。有趣的是，音樂也具有這兩種性質。如果讀者具有相關的背景，不妨在閱讀以下的深入討論時，試著將裡頭的例子和音樂做類比。

■ Composability

英文的“compose”一詞有兩種意思：合成以及創作。剛好在這邊想要強調的，正是計算同時具有compose一詞的這兩個含義。

在物理世界的模型中，大多使用所謂的偏微分方程(partial differential equations)或是更廣義的動力系統(dynamical systems)，這些數學工具不但具有豐富的描述能力，也具有十分強大完善的理論根基，因此引領著理論學家走得非常遠。然而，這些數學模型通常因為太過細緻，於是不太容易拼湊組裝。相較之下，電腦科學從圖靈機開始，就具有容易把不同功能/演算法合成拼裝的特性。如此彈性和自由的思考方式，更是帶給我們在“創作”上的方便性。

乍聽之下，創作似乎和嘗試理解自然的科學方法走著相反的方向：前者是一個“人造(artificial)”的過程，而後者則是一個“自然(natural)”的過程。如果在科學模型中加入太多人造的元素，那真的還能夠精準地理解大自然嗎？這時候我想重提一下本書前言中曾約略著墨的Penrose三個世界理論：在面對真實世界的時候，我們無可避免地將會透過心智世界，經由抽象的形式語言，建立出一套理解。既然這樣的科學過程必然是個創作的人造過程，計算身為一個方便讓人構思如何創作的介面，是否能夠用來建立我們的科學理解，甚至提供不同領域間的共同語言？

■ Modularity

筆者在就讀大學時的第一個專題研究方向是軟體工程中的「設計模式(design pattern)」，這個領域非常精湛地將計算中modularity的概念發揮得淋漓盡致。以下讓我們用餐廳做為例子來闡述幾個核心想法。

假想我們要開一家台式餐廳，主廚特色料理有糖醋排骨、番茄炒蛋、宮保雞丁、蒼蠅頭等等。每道菜雖然做法不太一樣，但是在結構上都具有(1)備料切菜、(2)肉/蛋的前處理、(3)辛香料爆香、(4)食材混合拌炒、(5)最終調味。於是在撰寫菜餚的食譜時，可以根據這五個不同的階段分別設計，如此一來之後廚房內的分工合作也更容易協調。這樣的概念又被稱作「工廠模式(factory pattern)」。

現在到了菜單規劃的階段，除了主要料理之外，前菜、甜點、飲料等等是決不能少的。而且根據不同時段和特殊日子，我們可能會想要提供特殊的套餐方案，讓客人可以有多種不同的料理組合可以搭配。如果根據所有可能的料理做排列組合，那菜單大概需要個幾十頁跑不掉。所以一般來說會使用樹狀結構的方式來彈性地表達套餐的選項。這樣的概念又被稱作「聚合模式(composite pattern)」。

最後，在制定餐廳作業流程及人員流動時，會有許多“指令”動作。例如主廚A要副廚B去切菜、副廚C要助手D去洗碗、領班E要服務生F去問客人有沒有食物過敏等等。這些指令的參與者可能會隨著餐廳職員的來來去去而發生變動，同時指令本身的內容或是之間的連接關係也可能會更新(例如疫情時要增加防疫措施，或是當餐廳規模變大後需要更多層次的命令結構)。於是，與其鉅細靡遺地描述所有人與人之間的互動，一個方便的抽象方式，便是將這些角色之間的“指令(command)”視為一個“物件(object)”，如此一來不但能夠彈性地變動指令的內容，又可以輕易地更改指令的發送者與接受者。這樣的概念又被稱作「命令模式(command pattern)」。

以上的例子只包含了設計模式的冰山一角，但希望藉此已經能讓讀者約略感受到所謂modularity的意思：透過將事物和功能抽象成為模組，讓操作和理解更加便利與清晰。計算的視角和語言很自然地能夠承載modularity，這樣的優勢也許能讓計算用來將量化/機械化的分析方法發揮到原本比較“敘述性(descriptive)”的研究課題上？

■ 結語

希望本書作為探索計算現代綜合論的初步嘗試，能讓讀者從大量的例子中，感受到計算的多種面貌，並體會到以上強調的composability和modularity。就像在學了量子理論後，有些人會開始從平行宇宙的視角看待日子裡的機緣巧合。或是在學了演化理論後，有些人會開始從物競天擇的觀點理解生命中的跌宕起伏。計算的視角，在未來或許也能帶給你不一樣的理解模型和思考語言！？

最後，由於筆者在本書大部份的主題中都非領域內的專家，難免會在用字遣詞和難易度拿捏等之上有些不完美的地方。因此，如果發現內容上有任何不精確甚至錯誤之處，歡迎來信指教。