

你能否通俗地介紹一下，你的“軟件設計編程自動化”SDDA 技術是什么？

真正製作電腦軟件的語言僅有三代：(1) 第一代是“機器代碼語言”，依據記錄了客戶要求的設計文件，再人工編制“機器代碼程序”；(2) 第二代是“程序語言”，依據記錄了客戶要求的設計文件，再人工編制“語言代碼程序”；(3) 第三代是“軟件設計語言”，它不需要人工編制程序，只要用它寫下客戶要求，這種記錄了客戶要求的設計文件，就能自動轉換成電腦上直接運行的“機器代碼程序”。這種強有力軟件設計語言都使用了軟件設計編程自動化 SDDA 技術。

我十二年的工作是創建了一種模式邏輯工具，用它首次突破了 SDDA 技術。這種 SDDA 技術，可以為任何系統任何種類的軟件，製作它的第三代的“軟件設計語言”。為了證明它“既強有力又有普遍意義”，我從二零零一年開始，為二個最複雜的軟件系統：(1) 為企業經營管理的 Window 視窗軟件，製作了軟件設計語言 SDDA_window；(2) 為互聯網網站軟件，製作了軟件設計語言 SDDA_web。這二種“軟件設計語言”工具都將在“可視化 D++ 語言”一書中介紹。

當今時代的狀況是一個工廠要成為汽車製造廠，不必要去學習怎麼提煉橡膠，怎麼加工輪胎和加工發動機。歷史悠久的專業廠已提供

自己原先都很杰出）。另外，在我的時態邏輯文章在“中國科學”外文版上發表，又受到美國“數學評論”評論員（有關語言理論與邏輯）邀請。這樣的經歷，把我推入了計算機理論科學的陣地。（我回憶了這些，不是我有什么特殊智慧。事實上，中國聰明的人很多，只要選準方向，沉下心來干上一二十，都能作出不錯的成績。）

怎麼會來美國搞計算機理論研究？

在我的時態邏輯文章在“中國科學”上發表之後，我已有足夠的理論基礎可以去國外進修。一九八六年，我曾經想去牛津大學學習和工作。有一次，應中科院軟件所的兩位資深研究員（最早的院士）唐稚松和周巢塵的邀請，給他們的研究生介紹我在軟件時態邏輯方面的研究成果。會後，因為周巢塵老師常被邀請去牛津大學工作，我就請他引薦我去牛津大學計算機系工作，而他建議我去找內基·梅隆大學計算機系，不去牛津大學計算機系。他對我說：“那兒的人是大 X' 級水平，而你已是‘元 X' 級水平。你應該去 CMU”。我接受了他的建議。回上海後，我帶着我的論文，請上海交大的孫永強老教授寫一封信，介紹我去 CMU 卡內基·梅隆大學 Ed Clarke 教授（後為圖靈獎獲得者）那裡學習。孫老師先看了我的論文，然後在一張紙上寫了一句話，中文意思是“此人對你必有用”，簽了名後交給我。一九八六年底，我飛到匹茲堡機場，見到了

系統。一月以後，Clarke 教授告訴我，他向論文作者要一個製作了的系統，作者回復說，他的（軟件）系統才開始編。此時，教授才要我寫出，為什麼作者的理論不可能製作了一個（軟件）系統。

在計算機軟件理論的研究中，你是否建立了與別人完全不同的新方法？

計算機軟件理論科學中，早期提出了兩大難題：一是算法複雜性的 NP 問題，另一個是程序正確性的驗證問題。在程序正確性的驗證的領域里，為驗證軟件程序中的大大小小循環語句，人們想起完全類似的“數學歸納法”。研究人員都自然而然地使用數學工具，去研究程序正確性的驗證問題。此問題太困難了。因此，為驗證程序正確性而提出每一種能讓人理解而接受的，邏輯或公理化理論的創始人，一般都能獲得圖靈獎。但是，軟件程序太複雜，而在執行過程中產生的狀態也多的無法估量，用嚴謹的數學邏輯方法僅能驗證簡單的小段的程序語句。而且，用數學表達式去驗證簡單的程序語句，這些複雜的數學表達式讓程序員難於理解。我在不同的研究組干了多年的程序正確性研究之後，我認為數學方法的潛力到此為止，就這麼一點能力，不能徹底解決問題。因而，人們需要化較長的時間，去探索尋找其它的工程的方法試試，當然不會立竿見影，前幾年不會有結果。但一般的研究組的課題，往往以年填寫研究計劃，年

下面我引用某網頁公佈的一段關於我的 PROLOG 新編輯器的技術討論：“大約在一九九六年前後，我曾給 CMU 的埃德蒙·克拉克教授演示這種自動製作 PROLOG 編譯器軟件的技術。要構造一款 PROLOG 編譯器軟件，除了一般的編譯，還要用到幾種特殊算法。埃德蒙曾問了三個問題，(1)“你的編譯器是否使用了沃倫機器？”，我回答說：“不用”，(2)“你的編譯器使用的垃圾回收算法是什么？”，我回答說：“不需要”，(3)“在編譯器里，什么是你的數學合併算法呢？”我回答說：“合併算法也不需要。我只是用一個簡單的一個數據到另一個數據的‘指向’，來代替複雜的合併算法”。突然，埃德蒙站了起來，說：“是的，你是對的”。就是平時這些新技術堆積，我才有基礎去考慮軟件工程的革新問題。

你的軟件工程的革新方法是怎樣的，能讓我知道一個大概意思嗎？

在真正確定軟件工程的改革方向之前，先要瞭解已有的軟件工程方法到底存在哪些根本性的問題。原來的軟件工程方法，說它是“工程”方法，好像還差很遠，我們知道，以現代工程化方法建造一幢大樓，大樓是什麼樣子，完全決定於設計文件，而不依賴於誰去做。一種先進的軟件製作技術也應該是這樣。同樣一份軟件設計文件，由不同的軟件公司產生的軟件應該完全一樣。這種機械化工程化製作過程，才真正實現

邏輯學家十二年實現“軟件工程方法”上的革命

介紹匹茲堡華人精英唐同誥先生與他的 D++ 語言教程

各種零部件，讓你選購。同樣，人們在進入第三代的“軟件設計語言”時代之後，一般軟件人員也根本不需要學習怎樣製作數據庫和數據庫程序。你只要寫上你要的數據庫表的名稱是“圖書館目錄”一詞，再寫上它的每條記錄由“書目”與“作者”二項目。總共，你只不過打了三個詞語，一個大型數據庫就產生，並提供各種需要的操作讓你選用。所以，人們在進入第三代的“軟件設計語言”時代之後，這種新技術已提供這樣的一種可能：一位普通人都可以為他熟悉的業務，製作他自己的電腦視窗應用軟件。在以前，這是不可想想吧。

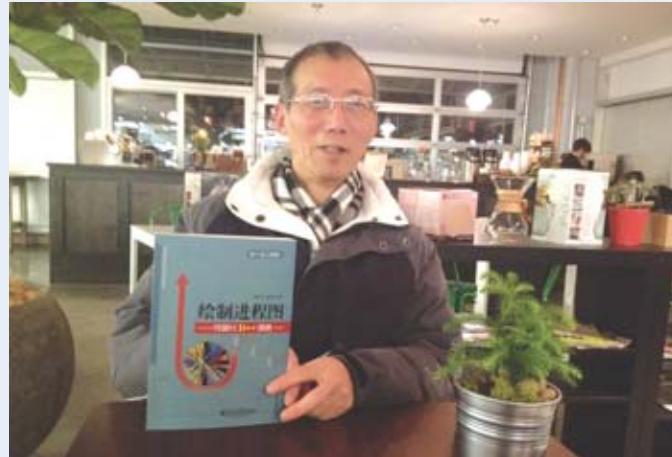
這裡要指出的是，可視化 D++ 語言是屬於一種“通常的，用于軟件設計的各種基本工具”的大集成，它的操作使用方式，都屬於人們使用電腦的常識。它內部的高智能的協調管理，保證它使用非常非常簡單方便，又通用可靠。附帶也指出一下：正象第二代語言能具有第一代語言的所有功能，第三代語言也能具有第二代語言的所有功能。因而，若有人問“能產生大軟件，能顯示美麗的畫面嗎？”一類問題，就可知這些問題都是次要的，都是不言而語的。

你原來是復旦大學數學教師，怎麼會去搞計算機軟硬件的？

因為日本、美國解決了用計算機去“光順”船體數據，有了正確的船體模型數據，從而能用計算機去控制船體製造。一九六七年，上海的造船研究所和計算機研究所，邀請了復旦大學數學系三位老師和二位學生加入他們“船體計算機數學光順”研究小組。二年下來，有一個特大難題解決不了。江南造船顧師傅組織領導了一次“攻克難題”大會戰。二個半月後，我在原有同事工作的基礎上，“模仿人工智能”、“機器學習”的方法，使用加‘權’與精選‘權’的非傳統方法，攻克了電腦船型光順的會戰難關”（引自“可視化 D++ 語言第 1 冊”一書的後言）。這使我看到了計算機的威力。我先打了一份報告，沒成功，後又補充打了一份報告，要求在復旦大學內建立“計算機製造廠”。感謝數學系領導的工宣隊張連長的支持，一九六九年，“復旦計算機製造廠”成立了。次年一九七零年，復旦大學“計算機結構”專業正式成立並招工農兵學生。當時，我是計算機製造廠長兼任計算機教研組組長，前後做了約十年。就這樣，我走上計算機的道路。嗨，中國的計算機製造廠成立比台灣宏碁和美國的 DELL 早，但是做的都比他們差。我不同意做得差的原因是“工人吃大鍋飯”。簡單地說，中國大部分人，無論在國營企業、私營企業，以及在美國打工者，毋庸置疑大都是智慧與勤奮的。

你十年搞計算機硬體之後，怎麼會轉去搞計算機理論科學？

文化革命剛結束不久，新幹部與恢復工作的老幹部一大堆，每一位都得安排一個職位。一大批職位，又引出一大批職權。於是，大事小事都要經過層層不同職權的又不瞭解生產的人員批准。我也無可奈何，最後說“計算機工廠我上繳，你們去管理吧。”。一九八零年，我加入了朱洪老師為領導的新建立的“計算機理論”教研組。大家都知道，中國的計算機軟件，以及基于電子元件工業基礎的計算機硬件，與世界先進水準相差一大截。但是，不涉及工業基礎設施的計算機理論科學，要趕上世界先進水平，還是有可能的。一九八一年，我暗暗下決心，在算機理論科學上，三年達國內先進水平，五年達國際先進水平。我把我的職責干好之外，不丟掉我的決心。即使領導安排我去做‘班主任’行政工作，但我晚上也可為計算機系、數學系的學生開專題“討論班”。幾年後，我的願望達到了。復旦校刊登載了：唐同誥五年左右為學生開了五門新課，以及參加了某個討論班的“六人中有三位直升博士生，二位考取碩士研究生”（要說明一下的是，這些學生



Clarke 教授。他駕車把我帶到他的辦公室。我把孫永強教授的信轉交他。他問我“你能幹什麼？”。到了美國，我就不必要過分謙虛，我回答說“我能做到你叫我干的任何事情”。過後想想，我的回答不精確，應該補充說，任何事情是指他的科研上的任何事情。如有一次，他建議我去 XXX 大學做教師，我就沒干。因為，當時我的英語水平，只能看懂別人寫得，基本上聽不懂學生說什麼。

在美國你得到的最大的受獲是什么？

我到美國是一九八六年十二月底，Clarke 教授交給我美國頂級計算機雜誌為他刊登一本專輯（共有八章），說這本專輯是我到這裏進修的主要學習材料，並說過了聖誕節後一月十日再談。到了一月十日，我去了 Clarke 教授辦公室，他問我“你看過這本專輯的文章嗎？”。我說“看了，而且看完了”。他說“你有什么問題要問我嗎？”。我說：“沒不清楚的問題。整篇文章的結論正確。不過，第八章的證明改寫一下更好”，他問我“哪兒？”。接着，我們就討論起來。以後，他不斷的送其他人的稿件給我，要我幫他初審一下，提個意見。這是我的工作，這些論文大都與我的研究領域無直接關係，有些是偏向代數領域，不太熟悉。為了正確審閱這些論文，有的結論查不到資料，我需自己動手證明，我需要花大量時間。事實上，我在 CMU 參加了三個有興趣的研究課題：軟件驗證、硬件驗證，以及人工智能的知識表達。這些需要邏輯知識，但都沒有超出我以前已經掌握的知識。對我而言，比審稿簡單，化的時間不多。

要說我真正從 CMU 幾位美國教授那兒學到的，是學術“思想方法”，也就是“在計算機科學領域里，對每一個概念、主意、想法、思想、方法，都要考慮用一個‘可計算的算法和程序’去描述和實現”。這方面，當時感覺 CMU 比斯坦福大學大學更強調一些。因此，當我以後拿到論文，我就思考把它的內容要用可計算的程序寫一下。這種做法的結果，有時會令人意外的，這兒僅僅舉一個例子。研究人工智能的人，幾乎沒有人沒讀過某位極為著名的人工智能專家的“機器學習”有關“表達式學習”的論文。此論文不長並有軟件，看過論文後沒人提出有錯誤。偶然機會，我參加的一個研究小組要用此論文的軟件。我拿來論文，依照論文提出的方法，用自編的程序去描述一下，發覺論文的數學中有關“極限”概念錯誤，導致論文提出的算法錯誤。如果不用自編的程序去實現一下論文的內容，論文的錯誤很難識別（錯誤已在某會議上報道）。但必須指出此論文的主體思想還是很了不起的。還有一次，我為校外的一位教授的論文審稿。他論文中提出一種複雜的理論，並在文末尾說：他已經根據他的理論，製作了一個（軟件）系統。在審稿時，我用可計算的算法和程序，模仿他的理論。發現他的理論根本“不可計算”，然後回答 Clarke 教授說：此論文作者根本不可能製作了一個（軟件）

編者按：在匹茲堡有一位從上海來美國近 30 年的華人唐同誥先生，他任教于復旦大學數學系，他在 1968 年在上海復旦大學開了計算機工廠，1986 年到匹茲堡，一生研究，最近他的第一本書《繪製進程圖》D++ 語言教程課本由電子工業出版社出版。浦瑛

了軟件工程化 (Software Engineering) 的原旨，但已有的軟件工程還不能做到此點。我決心探索，用建造一幢大樓的類似方式，根據設計圖紙，自動製作模塊，自動安裝模塊，來機械化建造軟件。想來，百年後軟件製作方法應該是如此。在一九九七年前後，我向我所在的美國第一個數據公司的子公司提交了我的這個建議。後來由於子公司撤銷，建議無回應。我化了幾年時間，分析了語言理論、實際編程，以及新業務的可行性之後，于二零零一年正式開始研製 SDDA (S 軟件，D 設計，D 編程，A 自動化) 技術和“可視化 D++ 語言”的設計與製作。于二零零四年，可行性方案測試成功。于二零零六年開始廣泛對外表演我的可視化 D++ 語言工具。

作為從軟件理論科學過來的人，總會想到，怎樣改變現有軟件工程產生軟件的“慢、差、貴”的現象。而這種現狀也主要是人工編制程序過程以及程序的複雜性引起的。其實，各大軟件公司都想過，直接從記錄客戶要求的設計文件（或稱“業務模型”），不用編制程序而直接生成編輯好的高速軟件代碼，該多好啊。擁有衆多軟件尖端人才的德國的大公司 SAP，美國的大公司 Oracle & PeopleSoft，以及大公司 IBM（曾經收購了 Rational Rose）都做過這方面的巨大努力。即使不談專用軟件，就是通常的企業經營管理軟件和網頁平台 CGI，都實在太複雜了。要生成整個軟件更是困難。這種困難現象，正如 Software Development 雜誌 [2006 年 4 月卷。14 日，第 4 期] 指出的：“實施模型到代碼和模型到模型轉換的任務，還需要做一些工作”。閱讀了該篇文章之後，我發 Mail 給該文作者，告訴他，我已成功“實施模型到代碼轉換”。他回信說，希望我公開我的軟件代碼。當然，作為公司的財產，我只能以產品的方式，公開技術。

你的“可視化 D++ 語言”書出版後，你有什么打算？

電腦軟件製作人員是不斷的學習新語言，以適應工作的需要。正如前面提到的，真正製作電腦軟件的語言僅有三代 (1) 第一代是機器代碼語言；(2) 第二代是編程語言，如 C, JAVA, PROLOG 等；(3) 第三代是軟件設計語言，如使用 SDDA 技術的可視化 D++ 語言。每代語言都有實質差別，一句話“一代比一代使用更方便有效”。寫完這本套書“可視化 D++ 語言”後，我打算給年輕學生上課，並提供的免費軟件。希望有利于年輕人創業。此書也將會很快有英文版出版。

這個先進的可視化 D++ 語言發行後，對現有軟件行業有什么影響與衝擊？

電腦軟件製作人員是不斷地學習新語言，這是他們的習慣。所以，對程序員個人來說，他們能學到更強有力更方便的語言，使他們能化較少的時間製作更多的軟件。最終，有關的軟件人員會喜愛它的。但對大中型的應用軟件公司是挑戰。如果停滯不進（當然很少可能），大中型的應用軟件公司的利潤，好多年後會分流給小公司群。

但是，我們也可預見到，在軟件歷史上臨時用于過渡的代用技術，將會逐漸萎縮。例如當網頁 CGI 軟件會自動生成，那麼 ASP 平台的歷史作用就完成了；又例如，SAP 公司和 Oracle PeopleSoft 公司採用的低速的“業務模型的解釋”技術，將會逐漸被簡便又高速軟件替代；又例如，若主要功能是畫流程圖，又僅僅產生構架式的程序。那麼使用複雜而又昂貴的 IBM UML 技術，使用會收到限制。總之，要存活，就要改革，

還再補充一下，“可視化 D++ 語言”軟件，對不同的電腦操作系統有不同的版本，先發行的版本 SDDA_window 和 SDDA_web 都是為 MS Window 操作系統的電腦用的。為其它計算機操作系統用的版本，將會比大中型的 Window 操作系統用的版本更簡單一些。大家一起干吧。