

资料库管理系统

SYSTEM-X 之设计与制作

黄为德 合著
李斯特

379

76

松岗电脑图书资料有限公司

72-1-1
5/10

資料庫管理系統

SYSTEM-X 之設計與製作

作 者

黃為德

德國慕尼黑工業大學自然科學博士

李斯特

國立清華大學數學研究所計算機科學組碩士

松崗電腦圖書資料有限公司 印行

547

資料庫管理系統 SYSTEM-X之設計與製作

版權所有



翻印必究

每本定價 280 元整

書號：510103

編著者：黃 為 德、李 斯 特

發行人：吳 守 信

發行所：道明出版社

台北市仁愛路二段一一〇號三樓

總經銷：松崗電腦圖書資料有限公司

台北市仁愛路二段一一〇號三樓

電話：3930255・3930249

郵政劃撥：109030

印刷者：東崗印刷設計股份有限公司

台北市仁愛路二段一一〇號三樓

電話：3930255・3930249

中華民國七十年十二月 初 版

本出版社經行政院新聞局核准登記，

登記證號為局版台業字第一七二九號

序 言

本書是作者將發展資料庫管理系統 SYSTEM-X 的經驗與資料整理出來，做有系統的說明。我們發表這些資料，特別希望對於實際從事資訊與電腦方面工作的同仁有所助益，包括：(1)對於發展大型複雜軟體系統有興趣，或是正在從事這項工作，尤其正在發展資料庫管理系統的人；(2)對於使用 SYSTEM-X 有興趣，或是正在使用類似這類系統來處理資料的人；(3)對於正在修習軟體設計，或資料庫管理系統製作的學生。本書撰寫的方式，並非資料庫管理系統的教科書，因此對於許多所應用到的基本觀念，僅提供參考文獻，並不作詳細解說，希望讀者自行選讀參考。

民國 66 年 12 月間，作者開始產生一種想法：由國人自行發展較大型而且較複雜的軟體系統，以求取發展經驗，並且希望因此能引起國人對於軟體系統的重視。當時國內對於發展資訊與計算機工業尚未有今日的認識，因此對於這種嘗試頗具冒險性，幸賴當時作者服務機構財團法人中國技術服務社的支持，得於民國 67 年 2 月間決定開始發展一套資料庫管理系統，我們之所以選擇發展這類系統的原因，乃是因為我們都有使用過這類系統的經驗，因此較為熟習，同時資料庫的理論已有相當程度的發展，也可以減少許多發展時的未知因素，最重要的是，國內使用資料庫管理系統來處理龐大資料的習慣並未普遍，即使應用，其系統的來源也大都引自於國外公司，不但價格昂貴，而且往往必須在特定的計算機上方可使用，對於系統本身的結構所知有限，如發生問題往往不能即時獲得滿意的系統修正，甚至於有些系統艱澀難用，不易獲得使用這類系統的確實效果。事實上，資料庫管理系統本身的昂貴售價或使用權費用倒還在其次，而使用機構所建立的資料庫本身却是價值無限，一旦資料庫管理系統發生問題，當有產生嚴重後果之慮，因此這種危險性必須減至最低，這是我們在國內發展可靠而且易學易用的資料庫系統 SYSTEM-X 的動機。

SYSTEM-X 發展至今已算告一段落，使用者亦可以安心使用，但是我們認為這個系統尚有不少發展的餘地，例如改進輸入方式，使它更為易學易用，或者連

結計算機系統中具有通訊公用軟體，使SYSTEM-X 具備某種程度的分散化能力，此外充分的測試，以及可靠性與效用性的衡量也是極為重要的工作。於是我們徵得中國技術服務社的同意，公開發展資料，提供有興趣的同業參考，如因此使得SYSTEM-X 獲得進一步的改進，也是作者最大的心願。

在發展SYSTEM-X 時，我們並未組成一個專案小組來全力工作，原因是支持我們發展的公司並非以軟體發展為主要業務，因此SYSTEM-X 的發展並非完全基於商業上的目的，不過當時公司的電腦作業中心成立不久，我們希望藉助SYSTEM-X 的發展，為公司累積軟體發展的經驗，進而帶動公司其他有關軟體的發展與資料處理的活動，實際上參與細節發展者祇有作者二人而已，因此用來支持發展的資源大部份都是利用公司的間接費用，並未做太多的直接投資，雖然這種發展方式，並非是真正要發展大型複雜，而且具備商業價值的軟體系統的合理方式，但是在軟體的價值尚未被我們的環境充分認識與肯定之前，至少我們可以用這種方式來開始。

雖然如此，我們在發展這個系統時，特別注意到軟體工程上的幾項基本原則，使系統儘可能地符合需求，而且具備可用性、可靠性與可保養性，我們根據模組化（modularity）的觀念來達到這些目的，因此特別在本書第一部討論模組化觀念之外，還利用所謂「流程描述法」（flow expressions）來說明SYSTEM-X 的結構，也利用圖表來表示SYSTEM-X 的程式模組與資料流程，這些圖表不但幫助讀者了解SYSTEM-X 的內容，同時對於系統裝置亦有所助益。

SYSTEM-X 的結構分「系統部」（System Mode），「定義部」（Definition Mode）與「資料部」（Data Mode）等三部。語言分：「泛系統命令」（System-Wide Commands），用來連繫三個部以及編輯工作；「擷取管制語言」（Access Control Language），用來做資料安全與會計管制；「資料定義語言」（Data Definition Language），用來定義資料結構；「資料處理語言」（Data Manipulation Language），用來處理資料，這些語言的規範與語法記錄在第二部（第3章至第6章）。第三部我們依照第一部所討論的軟體模組將SYSTEM-X 的原程式詳加分析與說明，對於修改SYSTEM-X 有興趣的同業定有幫助。在整本書中所列舉的例子，都是根據2.4 節的使用範例，因此在閱讀第3至6章時可能需要時常參閱。至於錯誤訊息在真正裝置時是另外建立一個獨立

檔案，以便於修正，我們將這些錯誤訊息載錄在附錄A。附錄B記載SYSTEM-X的界面子程式，使用者的應用示式可呼叫這些子程式，以擷取或更新SYSTEM-X內的資料庫，到目前為止，我們祇提供FORTRAN 應用程式的使用。附錄C為SYSTEM-X的指令節錄，可供使用者臨時參考之用。

由於發展這個系統，我們深深體會到軟體工程上的重要原則之一：「用好而少數的人」(Use better and fewer people) [BOEHM 79 b]，但是如何培養以及使用「好的人」，並不十分單純，最重要的必須建立良好的發展環境，這一點也許是我國資訊工業界的當務之急。

參與發展SYSTEM-X 的人士尚包括：國立清華大學的李家同、金陽和與陳文村等教授，以及國立交通大學的杜敏文教授，他們對於語言的設計方面有很大的貢獻，尤其李家同教授對於SYSTEM-X 的推廣費了很多的時間與精力，我們特別感謝他們的合作，以及對SYSTEM-X 的關心。同時對目前服務於中鼎工程公司的蔡文隆先生，我們也要表示由衷的謝意，蔡先生做了許多測試與維護方面的工作，同時將SYSTEM-X裝置在IBM 3031系統，提供經建會使用。我們特別要感謝的是支持我們的公司與人士，包括：中國技術服務社的金開英董事長，以及中鼎工程公司的王國琦兼總經理與朱德熙協理，沒有他們的鼓勵與支持，我們無從完成這件工作，如果這套系統對於國內軟體工業的萌芽有些微的貢獻，主要來自於這兩個公司的提倡。

為了名詞翻譯的困難，以及某些敘述的方便，本書的原資料係以英文書寫，我們原來也預定用英文來公開這些資料，但是經考慮再三，還是用中文來發表，以適合於國內使用，因此有部份易為誤解的名詞或加注英文，或仍然照用原文而不加翻譯。此外，本書編印時間十分匆促，幸賴松崗電腦圖書資料有限公司的全力支持，得以付梓出版，我們萬分感謝，不過疏漏難免，希望專家不吝指正，以待再版修改。

黃 爲 德
李 斯 特 謹識

民國70年11月於台北

目 錄

序 言

第一部 系統描述

第一章 SYSTEM-X設計背景	1
第二章 SYSTEM-X的簡介	5
2.1 系統結構	5
2.2 軟體流程	8
2.3 資料流程	11
2.4 SYSTEM - X使用範例	15

第二部 SYSTEM-X的規範與語法

第三章 泛系統操作(System-Wide Operation)	47
3.1 泛系統指令 (SWC)	47
3.1.1 SYSTEM MODE指令 (SWC - 1.1)	47
3.1.2 DEFINITION MODE指令 (SWC - 1.2)	47
3.1.3 DATA MODE指令 (SWC - 1.3)	48
3.1.4 MODE指令 (SWC - 3)	48
3.1.5 LINE指令 (SWC - 4)	48

3.1.6	EXIT指令 (SWC - 2)	48
3.1.7	COMMAND FILE指令 (SWC - 5)	49
3.1.8	REPORT FILE指令 (SWC - 6)	49
第四章	系統部(System Mode)	51
4.1	擷取管制語言 (ACL)	51
4.1.1	使用者密碼指令 (ACL - 1)	51
4.1.2	NEW DATA BASE指令 (ACL - 2)	52
4.1.3	DATA BASE指令 (ACL - 4)	52
4.1.4	ADD PASSWORD指令 (ACL - 3.1)	53
4.1.5	DELETE PASSWORD指令 (ACL - 3.2)	53
4.1.6	CHANGE PASSWORD指令 (ACL - 3.3)	54
4.1.7	RELEASE RIGHT指令 (ACL - 5.1)	54
4.1.8	RECOVER RIGHT指令 (ACL - 5.2)	55
4.1.9	ADD RIGHT指令 (ACL - 5.3)	56
4.1.10	DELETE RIGHT指令 (ACL - 5.4)	56
4.1.11	CHANGE RIGHT指令 (ACL - 5.5)	57
4.1.12	PURGE指令 (ACL - 6)	58
第五章	定義部(Definition Mode)	61
5.1	資料結構	61
5.2	資料庫構體	61
5.3	資料型態	63
5.4	資料定義語言 (DDL)	64
5.4.1	DEFINITION FROM指令 (DDL - 1.1)	64
5.4.2	DEFINITION FILE指令 (DDL - 1.2)	64
5.4.3	DEFINITION BEGIN指令 (DDL - 2)	64
5.4.4	CHANGE指令 (DDL - 3.1)	65
5.4.5	DELETE指令 (DDL - 3.2)	66

5.4.6	INSERT指令(DDL-3.3)	67
5.4.7	DISPLAY指令(DDL-5)	67
5.4.8	END指令(DDL-4)	68
5.4.9	PERMANENT指令(DDL-8)	69
5.4.10	FUNCTION指令(DDL-6)	69
5.4.11	REPORT定義指令(DDL-7)	70
第六章 資料部(Data Mode)		73
6.1	讀入步驟	73
6.2	層次資料結構	75
6.3	資料處理語言(DML)	79
6.3.1	DATA FROM指令(DML-1.1)	79
6.3.2	DATA FILE指令(DML-1.2)	79
6.3.3	DATA BEGIN指令(DML-2)	80
6.3.4	PRINT指令(DML-4)	80
6.3.5	DELETE指令(DML-3.1)	83
6.3.6	APPEND指令(DML-3.2)	84
6.3.7	INSERT指令(DML-3.3)	85
6.3.8	CHANGE指令(DML-3.4)	85
6.3.9	END指令(DDL-4)	86
6.3.10	統計計算指令(DML-5)	86
6.3.11	REPORT產生指令(DML-6)	87

第三部 SYSTEM-X製作

第七章 原程式		91
7.1	裝置程式	91
7.2	系統部與ACL處理程式	112

7.3	Scanner	152
7.4	檔案輸出入程式.....	183
7.5	構體表格保養程式.....	200
7.6	DDL 敘述處理程式.....	226
7.7	定義部資料庫定義程式.....	238
7.8	Display-Operator 程式.....	254
7.9	DDL Insert-Operator 程式	259
7.10	DDL Delete-Operator 程式	263
7.11	DDL Change-Operator 程式.....	271
7.12	輸出編輯程式.....	279
7.13	Function 定義程式	287
7.14	資料讀入程式.....	293
7.15	DML 索構體尋找程式	315
7.16	統計計算處理程式.....	326
7.17	自定計算式處理程式.....	331
7.18	Inverted List 處理程式.....	339
7.19	Boolean Operators 處理程式	349
7.20	條件子句與非索構體尋找程式.....	371
7.21	Normalization 處理程式	384
7.22	DML Insert-, Change-, Delete- 與 Append-Operator 處理程式.....	390
7.23	Data Mode, Data File, Data From 與 Data Begin 指令處理程式.....	399
7.24	詢問與擷取處理程式.....	401
7.25	資料庫安全處理程式.....	421
7.26	訊息處理程式.....	431
7.27	SYSTEM-X 主程式	440
7.28	列印 SYSTEM-X 表格程式	444

附錄 A 錯誤訊息 (Error Messages)	455
附錄 B 介面子程式 (Interface Subroutines)	459
附錄 C SYSTEM-X 指令摘要	473
索引	479
參考文獻	485

第一部 系統描述

第一章 SYSTEM-X設計背景

一般認為，人類無法有效處理大型複雜的軟體系統，是導致系統不可靠的主要原因，軟體系統可說是人類所遭遇到的最爲複雜的事務之一，因此在過去十幾年來，許多軟體工程專家致力於發展一些方法，來有效地設計與發展具備「良好工程化」(well-engineered) 結構的軟體，使這種軟體易爲人類所了解、使用、修改與擴充，這些方法包括軟體需求與規格 (requirement and specification)，程式規劃方法 (programming methodology)，軟體測試與確認 (testing and verification)，以及軟體管理 (software management) 等方面。在程式規劃方法之中，軟體結構的設計，似乎大都指向軟體模組化 (modularity) 的觀念，換句話說，軟體是由獨立的模組 (module) 以及它們之間的介面關係所組合而成，因此在軟體設計方面，最重要的問題之一是：如何來模組 (modulize) 一套軟體系統，使這套軟體容易被測試、修改與保養。軟體必須具備優良的測試性 (testability) 與保養性 (maintainability) 之所以重要，是因為在軟體發展的過程中，幾乎有一半的時間用在測試上，同時幾乎有 60 % 以上的費用使用在保養上「BOEHM 73」。因此一般的做法是限制程式的實質規模 (physical size) 來確保合理的模組化，但是這種做法並不十分妥當，譬如說，在 50 行的程式敘述 (statement) 中，如果其中包含 25 行 “IF...THEN” 結構時，該程式的管制道 (control path) 將高達 33,500,000 條，但是程式測試時，祇能測試其中的極少部份，許多以 FORTRAN 撰寫的程式，雖然實質規模不大，但是很難測試，就是這個原因。事實上，最佳的模組化，到目前爲止，並不爲人所充分瞭解[HALSTEAD 77]，不過模組化的觀念在程式規劃法中具有廣泛的意義，我們引用「ROSS et al. 75」的定義如下：「模組化所處理的是如何利用某一事物的結構，使我們易於達到某些目的，它是一種有目的的結構分解」。因此模組化的技術被

2 資料庫管理系統

用來增進軟體的可靠性，協助設計與促進程式的多重使用，並且使程式易於修改。總而言之，模組化的觀念可用來處理大型軟體的複雜性，包括軟體的設計與製作，同時亦可應用到軟體發展的人事組織上。SYSTEM-X 是基於這種觀念所設計與製作而成的大型而複雜的軟體。

應用模組化的觀念來設計軟體結構，可以使系統本身容易延伸（extension）或收縮（contraction），這種功能十分重要「PARNAS 79」，因此在SYSTEM-X 的設計中，我們特別注意到這種延伸與收縮的需要，包括：(1)軟體的某些子集合（subset）可以不必等到全部軟體完成即可工作使用；(2)容易在現有的程式中再加入某些功能；(3)為了簡化或加速系統的作用，甚至因使用上的經濟因素，容易在現有的程式中去除不需要的功能。因此我們在設計SYSTEM-X時 特別注意二項原則：(1)良好的系統結構；(2)明確的子系統間的介面（interface）。

由於SYSTEM-X是一種成套軟體（software package），因此它的需求（requirement）係根據市場的需要，而不是使用者的要求，這種軟體所被考慮的是「軟體的通用性」（software generality），同時這種軟體的使用，並不因各種使用環境而有大幅度的改變，因為這種改變的需要，已經在設計規格時加以考慮，（前面所談到的模組化功能是針對設計者與保養者而言，並非指使用者的環境），大部份的系統軟體都屬於這一類型。與軟體通用性相對的是「軟體的彈性」（software flexibility），具備這種性質的軟體，必須因使用環境的不同，而易於改變，一般使用者委託發展的應用軟體大都屬於這一類型。通用性的代價是機轉時間（run time）的費用增加，相反地，具備高度彈性的軟體雖然可以減低這種費用，但是將增加設計時間的費用。有些機構寧採前者多付機轉時間的費用，而不願發展具有高度彈性的軟體，以避免因維護數種版本所產生的保養問題，影響這種決定的因素包括下述各項：(1)可供額外使用的計算機資源；(2)可用來做程式修改與保養的技術；(3)某一程式的修改是否容易通用於其他多數版本。不過任何人都無法說清楚究竟一套軟體成品中應該含多大的彈性與適用性，但是在設計時必須決定軟體應該屬於那一類型。SYSTEM-X 的設計乃是選擇通用性的設計哲學。

在實際發展時，穩定的發展環境將提高發展者的生產力（productivity），其中最重要的是減少人與人間的介面複雜性與設計變更。依照IBM的記錄[WALSTON et al. 77]，包括小型單純的系統到大型複雜的系統，發展者與使用者

介面的複雜性對於生產力的影響，如果在正常的情況下，程式員每月可生產 295 行的原程式（包括說明），如低於正常複雜性可增加到 500 行，相反地，高於正常情況，將減少到 124 行，其間相差高達 376 行之多，所幸 SYSTEM-X 的發展並無這類複雜性。

第二章 SYSTEM-X的簡介

2.1 系統結構

SYSTEM-X 係依據下述需求而設計：

- (1) 具備廣泛的適應性，可裝置於多種廠牌計算機系統。使用整批或交談式作業。
- (2) 具備各種保護措施，可維護資料庫的安全與隱私性。
- (3) 具備類似自然英語 (natural English-like) 的高階語言，可建立、修改、保養與擷取資料庫，易學易用。
- (4) 具備計算能力，可直接用做統計及使用者自定的運算。
- (5) 具備同時處理中英文資料的能力，可產生使用者自定形式的各類中英文報表。
- (6) 具備高度適用性，可銜接其他應用程式。

依照上述各項需求，SYSTEM-X 由三個部 (mode) 組合而成：

- (1) 系統部 (System Mode)：專司安全與會計工作，包括：
 - (a) 檢查使用者的密碼 (password)；
 - (b) 確認使用者的帳錄；
 - (c) 配置使用者的資料庫，必要時創立新的資料庫。

系統部中所使用的語言為「擷取管制語言」 (Access Control Language 簡稱 ACL)。

- (2) 定義部 (Definition Mode)：在本部內，使用者可以定義資料庫結構，這種結構完全是以使用者的觀點來定義，是一種邏輯定義 (logical definition)，SYSTEM-X 的資料庫採用層次關係 (hierarchical relationship)，使用者利用「資料定義語言」 (Data Definition Language 簡稱 DDL) 來定義其資料庫結構，同時本部尚具備極易使用的編輯功能，讓使用者修改其資料庫定義。
- (3) 資料部 (Data Mode)：利用「資料處理語言」 (Data Manipulation

6 資料庫管理系統

Language 簡稱 DML)，使用者可以在本部內輸入，修改與擷取資料庫資料。

爲了增加使用 SYSTEM-X 的效率，當使用者在使用某一部時，其他二部與使用者之間並無直接關係，其間的連繫由「泛系統命令」(System- Wide Commands 簡稱 SWC)來擔任，因此使用者可利用 SWC 進入其他任何一部，同時 SWC 亦擔任進入與離開 SYSTEM-X 的工作，也是操作系統與 SYSTEM-X 之間的橋樑。圖 2-1 表示上述這些關係。爲確保資料庫的安全與隱私性，SYSTEM-X 的任何使用者(包括資料庫所有人)，進入系統時，都是經過系統部，以檢查其密碼，並且記錄其帳號，不過使用者可以從任何一部離開 SYSTEM-X，不必再經由系統部。

SYSTEM-X 的資料庫結構包括「記錄」(record)及其說明，以及其間的層次關係，合稱爲記錄型態(record type)，每一記錄型態內含有一項或多項「記體型態」(field type)，在每一記體型態上，使用者可自定是否爲「索」(key)，以及資料型態(data type)，如 name , integer , decimal , money 等型態。如圖 2-2 所示者爲 SYSTEM-X 功能圖說，由該圖可以了解到，系統保證能夠適時地、安全地以及隱私地儲存大量的資料，同時資料可以有效地被連結處理。

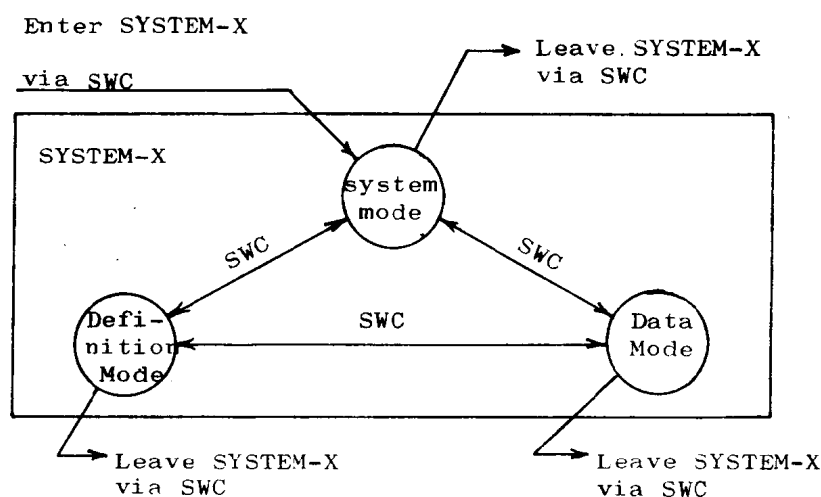


圖 2-1 系統部、定義部與資料部之間的關係