情報管理研究 Vol. 13 No. 4 1980. 8.

實驗的 SDI시스팀 開發

강 한 철
〈KORSTIC 企劃室〉

1. 머릿말

개발도상국들의 과학기술 개발을 촉진하기 위해서는 무엇보다도 먼저 이들 국가의 과학기술 정보시스팀을 발전시키고 이를 토대로 국가간의 네트웍을 形成함으로써 국경을 초월하여 과학기술정보의 흐름을 원활히 해야한다는 論議가 1960年代부터 1970年代에 걸쳐 과학기술개발에 관한 각종 政府間 회의 및 국제회의를 통하여 꾸준히 展開되어 왔다. 그러나 국내, 국제적인 情報流通시스팀의 形成 및 運営過程에서 많은 개발도상국들이 직면한 공통적 障碍要因은 현대적 情報処理知識과 技術을 갖춘 유능한 情報専門家의 절대부족 현상이었다. 이러한 難関을 타개하기 위한 노력의 一環으로 유네스코, 유엔디피 그리고 미국, 영국의 해외원조기관들의 후원과 필리핀, 태국, 인도네시아, 말레이지아, 싱가폴 등 소위 「아세안국가연합」국들의 공동사업으로 1978년 필리핀대학교에 제1차 「과학정보 전문가 훈련과정」(Science Information Specialists Course)이 開設되었고 필자는 1979/1980 学年度 제2차 과정을 이수한 바 있다.

6개국에서 22명이 참가한 본 코스에서 도서관학 및 정보과학의 一般理論, 情報処理의 컴퓨터응용, 이와 관련된 컴뮤니케이숀, 시스팀分析 資料의 統計的 処理, 情報시스팀 개발 등의 과목이 연수되었는데 마지막 한달간은 학생들을 5개그룹으로 나누어 각 그룹별로 컴퓨터를 이용한 情報検索시스팀을 개발하는 실험을 실시하였다.

이 글은 독자들이 SDI서비스의 定義, 필요성, 효과, 실태 등 기본적인 지식을 이미 알고 있다는 前提下에 필자가 참가한 「文献速報選択的情報提供(Slective Dissemination of Information, SDI) 시스팀 개발」에 관한 節次方法 등 技術的側面에 대하여 記述하고 본 시스팀을 스스로 分析, 評価하는 것이 내용을 이룬다.

2. 入出力 設計

2.1 入力(INPUT) 設計

1) 데이터베이스개발

데이터베이스(Database)—다른 표현으로 情報蓄積파일(Information Storage File)—제작을 위한 情報源(Information sources)으로서 우리는 영국에서 발행되는 도서관학 및 정보과학분야 抄録誌(Library and Information Science Abstracts, LISA)를 선택하였다. 데이터베이스의 情報源을 選定하는 데에는 予想利用者(Potential Users)들의 관심분야와 一致시켜야 하는데 우리가 만드는 데이터베이스는 실험용이기 때문에 情報源 選定은 어느 分野 어느 情報源을 択해도 이론상으로 문제가 되지 않는다. 그러나 情報源이 予想利用者들의 관심분야와 相異할 때는 検索率(Retrieval Rate)이 零으로 되거나 매우 낮은 결과를 가져올 것이기 때문에 우리가 予想利用者들로 선택한 学友들과 교수님들의 専

Worksheet for Database

Doc. No.

Data Elements	Description
Descriptors	
Name of Author	
Title of Article	
Title of Periodical (Journal)	
Volume No. (Issue No.)	
Date of Publication	
Pagination	
Number of references.	

그림 1.

攻分野에 맞도록 「LISA」를 情報源으로 決定했다. 또한 우리가 계획한 시스템이 오프라인배치方式(Off-line Batch Mode)이어서 데이터베이스의 生産者이며 동시에 利用者 프로파일作成者가 될 우리들 자신이 서치 叙述者를 選定하기에 容易하게 되는 점도 고려하였다. 특정 학문분야를 대상으로 만들어진 데이터베이스검색에서 서치 프로파일에 나타나는 검색어가 그 분야에 있어서 가장 보편적인 用語 예를 들면 우리가 택한 도서관학 및 정보과학에서 「정보서비스」 또는 「利用者」 등이 들어있는 서치프로파일은 자칫 데이터베이스에 入力된 문헌의 대부분을 검색해낼 가능성이 많고 따라서 검색된 문헌의 適切性을 감소시킨다. 이점에 유의하여 우리는 서치프로파일 작성은 물론 데이터베아스에 入力될 문헌의 選定에도 同一하거나 類似한 主題의 문헌이 重複되지 않도록 각별한 주의를 기울였다.

데이터베이스에 入力될 문헌의 書誌的 情報 및 叙述子들을 기록하기 위하여 우리는 먼저 (그림 1)과 같은 「워크시트」를 작성한 후 여기에 기록된 데이터를 코딩紙에 옮겨적고 끝으로 홀러리스 카드에 穿孔하였다.

데이터베이스 入力用 카드는 세가지 종류로 設計하여 첫번째 카드에는 일련번호, 카드번호와 함께 叙述者들을, 두번째 카드에는 일련번호 카드번호와 함께 著者, 論文標題를, 세번째 카드에는 일련번호, 카드번호 雜誌名(Title of Source Document), 巻/号, 出版年月日, 페이지, 참고문헌수를 각각 入力하였다. 그림 2는 이와 같은 排列順序에 따라 穿孔하여 入力시킨 세가지 카드의 見本을 보여주고 있다.

그림 2에서 보는 바와 같이 各 데이터 및 레코드欄은 固定化(Fixed Length Field)되어 있기 때문에 데이터베이스의 出力例에서와 같이 各 레코드마다 데이터의 길이에서 큰 차이를 나타내어 —실제로 문헌입력번호D001의 著者 WALSH, MYLES. E와 D002의 UNESCO 사이에는 7개의 文字数 차이를 나타내고 있다(그림 3 참조)—결국 데이터 蓄積의 浪費를 가져올 뿐만 아니라 各 데이터欄에서 相異한 書誌情報의 入力을 排斥하게 되기 때문에 情報源의 単一性을 避할 길이 없다. 우리가 이러한 短点을 갖고 있는 固定欄方式을 択한 이유는 検索프로그램作成이 容易한 長点때문이었는데 이를테면 하드웨어部門에서의 非能率을 甘受하고 소프트웨어部門의 便利性을 択했다고 할까?

데이터베이스의 情報源으로서 「LISA」를 利用하였다고 前述하였으나 사실 우리는 情報源의 単一性을 克服하기 위한 노력으로 유네스코가 発刊한 몇권의 単行本을 入力시키면서 検索時間을 단축시키기 위해 著者가 유네스코인(Corporate author) 문헌을 검색할 때는 利用者 프로파일의 書誌事項番号 3 (INSTITUTION) 앞에 달러사인($)을 부착하여 컴퓨터가 바로 著者·을 읽어나갈 수 있도록 검색프로그램을 작성하였다.

우리가 데이터베이스에 入力시킨 문헌의 数는 모두 107件이었는데 107이란 総入力件数는 任意的인 것으로서 더 많이 入力시킬수록 検索率이 向上되어 보다 충실한 데이터베이스가 될 것은 분명하지만 본 실험에 주어진 제한된 기간 때문에 더이상 入力시키지 못했다. 그러나 우리의 予想利用者 20~30名중에서 몇사람만이라도 문헌검색이 이루어지고 나머지는 단 한 件의 검색도 없다 할지라도 「관련문헌없음(No DOCUMENT)」이라는 出力만 나오면 본 시스템 개발실험은 일단 成功하는 것이기 때문에 総入力件数는 그다지 문제될 것이 없었다. 정작 중요한 것은 데

Card Type No. 1

Character Position	No. of Character	Description
1 – 5	5	Serial No.
6	1	Card No.
7 – 78	72	Descriptor

Card Type No. 2

Character Position	No. of Character	Description
1 – 5	5	Serial No.
6	1	Card No.
7 – 25	19	Author
26 – 78	53	Title of Article

Card Type No. 3

Character Position	No. of Characters	Description
1 – 5	5	Serial No.
6	1	Card No.
7 – 55	49	Title of journal
56 – 63	8	Volume No. /issue No.
64 – 67	4	Date of Publication
68 – 75	8	Pagination
76 – 78	3	References

그림 2.

```
D00011MANAGEMENT INFORMATION SYSTEM
D00012WALSH,MYLES E.      MIS:WHERE ARE WE,HOW DID WE GET HERE AND WHERE ARE WE
D00013JO OF SYSTEM MANAGEMENT                              29(11)  N0786-21  4R
D00021INFORMATION SERVICES
D00022UNESCO              WORKSHOP ON SCIENTIFIC,TECHNICAL-INDUSTRIAL INFOR.
D00023BACA                                                 5(1)      781-3
D00031LIBRARY EDUCATION
D00032CARTER,JANE R.      MULTICULTURAL GRADUATE LIB EDUCATION
D00033EDUC LIBRARIANSHIP                                   18(4)   SP78295-314
D00041CLASSIFICATION
D00042SMITH,LYLNN S.      TO CLASSIFY OR NOT TO CLASSIFY
D00043SERIAL LIBRARIAN                                     2(4)    SU78371-385
D00051INDEXING
D00052COOPER,WILLIAM S.   INDEXING DOCUMENTS BY GEDANKEN EXPERIMENTATION
D00053JO OF AMERICAN SOC. FOR INFO.SC.                     29(3)   0578107-109 11R
D00061NATIONAL BIBLIOGRAPHY
D00062ELROD,J. MCREE      UNIVERSAL AVAILIBILITY OF BIBLIOGRAPHIC RECORDS
D00063IFLA JO                                              4(4)    1178347-350
```

그림 3.

User Profile Sheet

Profile No.
Name and address of user :

Please state your search request in narrative form(brief but detailed). Do mention references(portaining to your subject interest) which have been of use to you previously.

그림 4

User Profile Term Sheet

Profile No.

Type Code	Alphabetic Code	Profile Terms
	A	
	B	
	C	
	⋮	
	⋮	
	2	

Search Expression

그림 5.

이터베이스 生産과 利用者 프로파일 작성, 검색 프로그램, 서치알고리듬, 매칭시스팀 등 본 실험에 核心的인 要素들이 하나의 시스팀으로 작용하여 목적하는 결과 즉「컴퓨터 문헌검색」이 実現되는지의 与否이다.

우리는 우선 25件의 문헌과 4개의 이용자 프로파일을 結合시켜 보았다. 그러나 4개의 프로파일이 모두 한건의 문헌도 검색되지 않아 결과는 실패였다. 우리는 검색프로그램에 修正을 加하고 10件의 문헌을 追加入力시킨 후 실험을 다시 해보았으나 또 실패였다. 한동안 이와 같은 試行錯誤(Trial and Error)를 거듭한 끝에 매칭시스팀에 異状이 있음을 알고 서치과정에서 매칭의 우선순위를 表示하는 괄호를 무시하고 左에서 右로 直進서치方式인 接頭符号表現(POLISH NOTATION)을 사용함으로써 문제를 解消할 수 있었다.

2) 利用者 프로파일 作成

이용자들의 情報検索主題를 一次的으로 파악하기 위한「利用者 프로파일 워크시트」를 作成하였는데 이 워크시트의 특징은 이용자들이 찾고 싶어하는 주제를 획일적으로 記入하는 대신에 정보검색을 요구하는 動機라든가 이미 사용하고 있는 参考文献, 資料調査 또는 研究上의 애로사항 등을 形式에 관계없이 서술하도록 한 점이다(그림 4 참조).

이와 같은 워크시트는 필요로 하는 情報를 어떤 特定主題로 표현하기 곤란한 정보이용자에게 편리할 뿐 아니라 우리가 정보이용자를 대신하여 서치 프로파일을 작성할 때 両者間의 個別的

인 접촉없이도 이용자의 真意를 쉽게 파악할 수 있는 長点이 있었다.

이용자들이 기입한 워크시트로부터 검색 敍述子(Search Descriptor)들을 골라내고 各 서술자를 符号化하기 위하여 로마알파벳文字를 부여한 다음 서술자의 書誌的 性格에 따라 다음과 같이 到達位置番号(Access Point No.)를 주었다(그림 5 참조).

1. =Author
2. =Subject
3. =Institution
4. =Title of Journal Article
5. =Title of Source Document

이리하여 컴퓨터는 가령 「A」라는 검색 서술자의 到達位置番号「2」를 기억했다가 데이터 베이스에 入力된 데이터를 읽을 때 主題欄에 곧바로 接近(Direct Access)할 수 있도록 했다.

문헌검색방법으로 선택한 5가지 書誌情報중에서 어느 것을 검색서술자로 결정하느냐 하는 문제는 이용자의 요구가 우선 고려되어야 한다. 어떤 이용자는 「○○○」이 쓴 문헌을 모두 찾아달라는 수도 있고 다른 이용자는 特定主題에 관한 문헌이면 누가 썼건 모두 찾아달라는 수도있고 또 다른 사람은 주제와 저자를 提示할 수도있다. 그러나 일반적으로 검색서술자를 主題로부터 선택하는 것이 가장 보편적인 반면 특정한 論文의 이름(Title of Article)을 提示하는 경우는 SDI서비스에서 가장 드문 例이다.

우리는 多様한 이용자의 정보검색요구에 応할 수 있도록 서술자 뿐만 아니라 두가지 이상의 書誌事項사이도 서로 논리적 관계를 맺어 주는 블린대수(Boolean Algebra)를 이용하여 서치数式(Search Expression)을, 다시 말하면 서술자 코드의 組合을 만들고 이를 이용자의 이름주소와 함께 홀러리스카드에 다음과 같은 排列로 入力시켰다.

우리가 作成하여 실험한 이용자 프로파일은모두 37件이었는데 그중 27件만 이 데이터베이스에서 1개이상의 문헌이 검색되었고, 나머지 10個의 프로파일은 「관련자료없음」이란 결과를 가져왔다. 이것은 약 73%의 検索率을 나타내는 셈인데 検索率이 더 올라가지 못한 것은 데이터베이스에 入力된 문헌의 수가 제한되었고 (107件), 이용자 프로파일作成에 경험이 부족했기 때문으

Character Position	No.of Character	Description
1 — 3	3	Serial No.
4	1	Card No.
5 — 24	20	User's name
25 — 78	54	Office addr.

Character Position	No. of Character	Description
1 — 3	3	Serial No.
4	1	Card No.
5	1	Descriptor Type Code
5 — 78	73	Descriptors

Character Position	No. of Character	Description
1 — 3	3	Serial No.
4	1	Card No.
5 — 78	78	Search expression

그림 6. Card Design for User Profile

```
                                ILS / SPGTCSTSSA / NCI

                                SDI PROJECT
999                                                                                    PALE
                                                                                         9
XXXX USER NAME XXXXX                                                                   DATE
XXXXX-----OFFICE ADDRESS ------ ---XXXXXXXX                                            99/99/99

9          XXX----AUTHOR---- -XXX          XXXXXXXX-----TITLE OF JOURNAL - ---- - XXXXXX
XXXXXXXXX--- TITLE OF ARTICLE---  -------XXXXXXX
99999999  XX  99  99999999  XXX

XXXXXXX---- --- ---DESCRIPTOR---------- ---------- ---XXXXXX

      (  ) RELEVANT        (  ) RELEVANT, NOT NEEDED        (  ) NOT RELEVANT
```

그림 7 OUTPUT FORMAT DESIGN

```
0011MAMANCE,PATCHARTE   INSTITUTE OF LIBRARY SCIENCE LAUA DILIMAN
00122THESAURUS#COMPUTERIZED#8
0013A*B
00218ARLI  SUMANTRI     LIPI-PDIN JAKARTA INDONESIA
00222MANAGEMENT#INFORMATICN#SYSTEM#8
0023A*(B*C)
0031PRABHASANOBOL,DAFAN INSTITUTE OF LIBRARY SCIENCE U.P. DILIMAN
00322   COMPUTERIZED#INFORMATION#SYSYEM#SERVICES#8
0033A*B*(C*D)
0041DARDJAT,RUKASHI     PSLIPI BANDUNG INDONESIA
00422 LIBRARIANSHIP#INFORMATION SCIENCE#8
0043A*B
0051SWATOEE,CHAWEEWAN   MAHIDOL UNIVERSITY BANGKOK THAILAND
00522LIBRARIANSHIP#LIBRARY AUTOMATION# NDEXING#8
0053A*B(B*C)
0061MANATAD,JOSEFINA M. NSDB BICUTAN TAGUIG METRO MANILA
00622LIPRARY#COMPUTERIZED#8
0063A*B
0071NUR IMA KAHAR       UNIVERSITY OF MALAYA KUALA LUMPUR MALAYSIA
00722INFORMATICN CENTERS#COMPUTERIZED  LIBRARY#PLANNING#8
0073A*B*C
0081ABDUL ROCHIM ADNAN  CBS JAKARTA INDONESIA
00822INFORMATICN SYSTEM#DATA-BASE#8
0083 A*B
```

그림 8

로 생각된다. 그러나 데이터베이스에 문헌을 얼마든지 追加시킬 수 있고 검색프로파일 작성은 경험에 의하여 숙달될 수 있다는 점을 고려할 때 73%의 検索率은 결코 낮은 수치라고 할 수 없다. 오히려 검색프로파일에 보편적인 単語(Universal Terms)가 사용됨으로써 데이터 베이스에 入力된 대부분의 문헌이 검색되는 가능 성을 더욱 염려하였는데 5件이상의 문헌이 검색된 프로파일은 단 2개뿐이어서 실험의 결과는 매우 고무적인 것이었다.

2.2 出力(OUTPUT) 設計

出力様式(OUTPUT FORMAT)은 하나의 시스팀을 設計하고 운영함으로써 얻어지는 결과를 담는 그릇이라고 할 수 있다.

우리는 375㎜ ×275㎜ (150列×66行) 코딩紙를 사용하여 그림 7과 같은 出力様式을 만들어 上端에 이용자의 이름, 주소欄이 그리고 검색된 書誌情報(Bibliographic Citation)가 著者, 論文名, 源誌名, 巻/号, 出版年月日, 페이지, 参考文献의 順으로 排列되도록 하였다.

한가지 특이한 점은 下端部에 문헌검색에 사용된 叙述者들을 挿入하고 그밑에 이용자로부터 검색된 문헌의 適切性 与否를 묻는 欄을 만들어 出力으로부터 피드백 機能을 수행하도록 設計한것이다. 만일 검색된 문헌이 適切하지 않다고 피이드백되면 이용자와의 협의를 통하여 검색 叙述者들을 다시 選定하거나 서치数式의 変更을 함으로써 向後 검색되는 문헌의 適切性을 提高시키려는 것이 본 出力様式에서 追求한 意図이었

다. 그림 8은 이와 같은 出力様式에 따라 실험에서 실제 出力된 例를 보여주고 있다.

3. 서치 알고리듬

알고리듬이란 하나의 문제를 해결하기 위하여 밟아야 할 階段的 節次를 말한다.

本橋에서는 컴퓨터로 하여금 이용자 프로파일 속의 叙述子들을 論理的으로 結合시켜 데이터베이스의 레코드중에서 適切한 것만을 골라내도록 指示하는 方法과 節次를 서치 알고리듬으로 規定하였다.

각기 독립적인 個体인 叙述子들을 연결시키기 위하여 다음과 같은 세가지 컴퓨터 可読形 블린 演算子를 사용하였다.

*——→「AND」
+——→「OR」
——→「AND NOT」

블린演算子「*」는 찾고자 하는 문헌 속에 두가지 서술자가 모두 들어 있어야 한다는 数式(Search Expression)을 만들고「+」은 찾고자 하는 문헌속에 둘중 어느 하나만 들어 있어도 좋다는 数式을「−」는 특정한 서술자가 들어있지 않은 문헌을 찾는 数式을 만든다.「−」는 검색문헌의 適中率(Relevancy rate)을 높이기 위한 方法의 하나로 가령「LANCASTER」라는 사람과「RAWLINSON」이라는 사람이 우연히도「INFORMATION RETRIEVAL」이라는 표제의 책을 썼다고 할 때 정보이용자는 이미「LANCASTER」가 쓴 문헌은 읽었기 때문에「정보검색」이라는 標題를 가진 책중에서「LANCASTER」가 쓴 것 이외의 책을 찾고 싶다고 할 때 이를 서치 数式으로 表示하면「A×B−C」로 된다.

그림 8의 이용자 프로파일 出力例에서 프로파일 번호「003」,「Institute of Library Science U. P. Diliman」을 주소로 하고 있는 PRABHASANNOBOL DARAN이라는 사람이 찾고자 하는 정보는 카드번호 2의「COMPUTERIZED # INFORMATION # SYSTEM # SERVICE #」검색 서술자로 미루어「自動정보 시스팀」또는「自動情報서비스」를 주제로 하는 문헌임을 알 수 있다. 사실 이 사람은 우리 学友로서 正式姓名은 이보다 길어「PRABHASSANNOBOL DARANEE」인데 述前한 바와 같이 入力카드 設計에서 固定欄方式을 취함에 따라 姓名欄에 할당된 最大 20文字 制限으로 右側切削(Right Truncation)을 당하여 이름이 DARANEE에서 DARAN으로 준 것이다.

검색서술자 사이의 #(더블해치)는 서술자가 독립된 個体임을 表示하는 다시 말하면 서술자 사이의 論理的 断切(LOGICAL BREAK−UP)을 컴퓨터가 알아 볼 수 있도록 숫자 또는 알파벳 文字(ALPHANUMERIC)와 뚜렷이 구별되는 符号를 사용한 것이고 맨끝의 £(파운드사인)은 더이상 서술자가 없다는 表示이다.

「0033」세번째 카드에 入力된 본 이용자의 서치数式「A*B*(C+D)」을 풀어보면 서술자의 알파벳 文字코드

A=COMPUTERIZED
B=INFORMATION
C=SYSTEM
D=SERVICE를 代入해 보면「A*B*C」=「"A" AND "B" AND "C"」또는「"A" AND "B" AND "D"」로 되어 결국 DARAN이라는 사람이 찾고 싶어하는 문헌의 주제「컴퓨터 정보 시스팀」또는「컴퓨터 정보 서비스」와 一致함을 알 수 있다.

4. 매칭시스팀

문헌정보는 磁性디스크에,이용자프로파일은 홀러리쓰 카드에 각각 入力되어 있다. 컴퓨터는 맨먼저 이용자의 이름과 주소가 穿孔된 카드를 읽고 그다음 두번째 카드의 서술자들을 接近位置番号(Access Point No.)와 함께 읽고나서 이들을 일단 緩衝記憶装置(BUFFER STORAGE AREA)에 저장한다. 그 다음 컴퓨터는 제 3카드의 서치数式을 읽고 이를 接頭符号表現(POLISH NOTATION)으로 変換시킨 후 이것을 緩衝記憶装置에 저장한다.

매칭이 시작되면 컴퓨터는 완충기억장치에서 우선 데이터베이스의 어느 欄을 읽을 것인가를 나타내는 接近位置番号를 확인하고 接頭 符号表現의 順序에 따라 데이터베이스의 데이터를 左

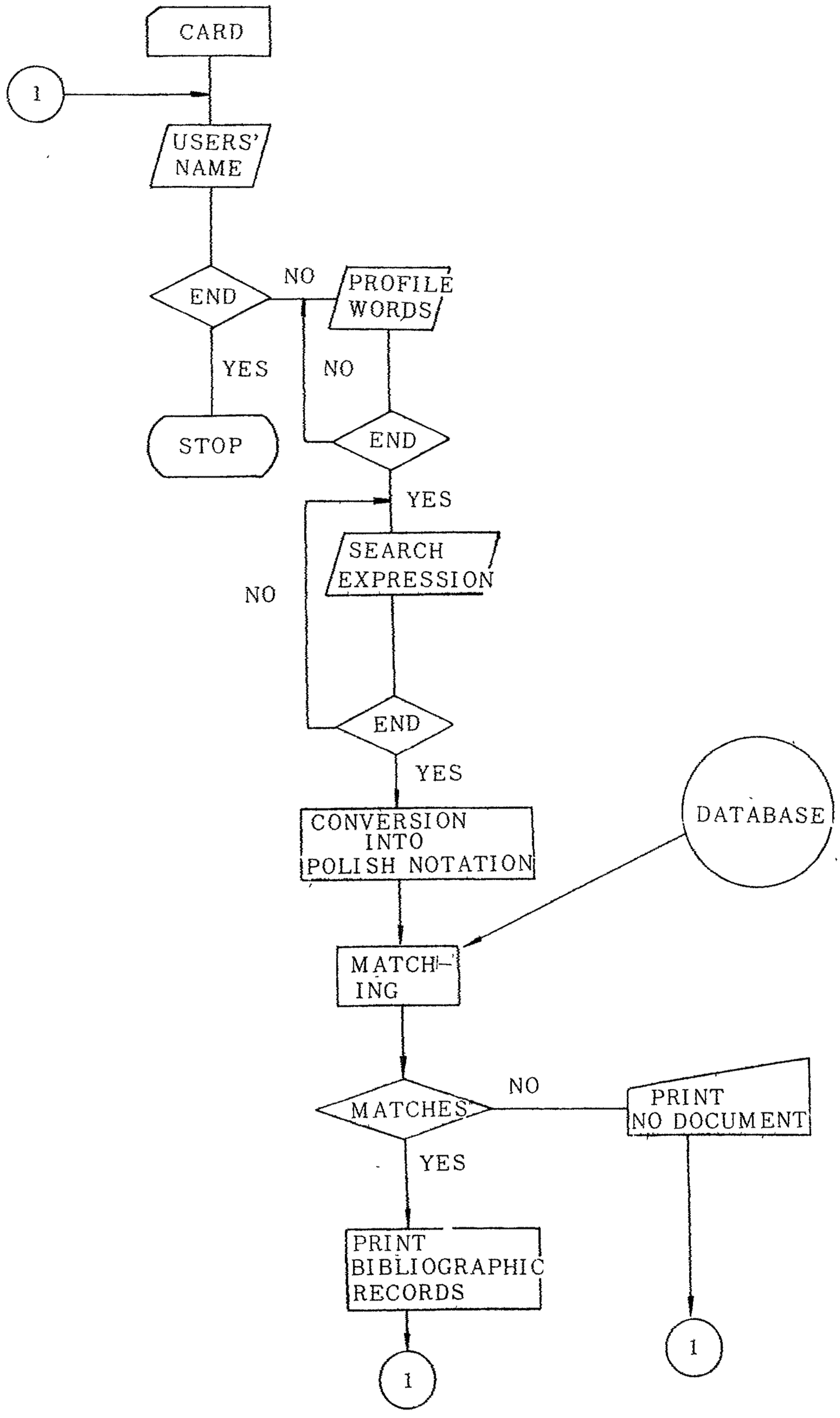

그림 9, Simplified Flow-chart of Matching Process

입력데이터	일련번호, 이용자이름, 주소, 서술자서지번호, 서술자, 서치수식	일련번호, 서술자, 저자, 문헌주제명, 잡지명, 卷1号, 출판년월일, 페이지, 참고문헌
사용회수	계 속	1回
番地指定方式	固 定 欄	固 定 欄
파일크기	레코드別최대240文字총37個레코드	레코드別최대240文字총107레코드
파일구조	順 次 파 일	無順到達파일
데이터저장수단	천 공 카 드	磁性디스크

그림 10. 파일明細書

에서 右로 읽어나가다가 이용자 프로파일의 서술자와 같은 데이터가 발견되면 被演算함수(OPERAND)는 「1」의 값을,매칭이 안되면 被演算함수는 「0」의 값을 갖게 된다. 이와 같은 過程이이용자 프로파일의 서술자 全部가 消盡될 때 까지 계속 反復되고 난 후 서술자 A, B, C, D, … Z의 被演算함수 값이 정해지면 演算子「*」를 가진 数式은 論理和回路 또는 論理積回路를 따라 真偽表(TRUTH-FALSE TABLE)에 나타난 값이 「1」이면 이용자의 이름, 주소와 함께 데이터베이스의 書誌情報가 出力되고 「0」이 나오면 「관련자료없음」이란 결과가 出力되게 된다(그림 9, 매칭과정흐름도 참조).

5. 소프트웨어, 하드웨어

手作業(Manual Searching)의 限界를 초월한 尨大한 情報蓄積속에서 정보이용자가 원하는 정보만을 컴퓨터로 하여금 연속적으로 골라내게 하기 위한 準備, 入力, 貯藏, 検索, 出力 등에 관한 설명이 이글의 대부분을 차지하고 이것이 곧 본 시스팀 개발의 소프트웨어이가 때문에 소프트웨어 부분만을 別度로 論議할 필요는 없을지 모르나 이 들의 核心이 되는 두가지 데이터파일의 明細(FILE SPECIFICATION)를 하나의表로서 整理함으로써 두 파일의 특징을 비교 검토할 수 있을 것이다(그림 10참조).

본 시스팀의 검색프로그램作成은 코볼이 사용되었고 하드웨어는 기억容量 131KB의 UNIVAC 90/30, 周辺機器로는

磁性디스크(容量·28,95MB; 移送率 625 KBS)

카드 判読器(判読率 500CPM)

라인 프린터(500 LPM) 등이 이용되었다.

6. 시스팀分析

우리가 실험한 매칭시스팀은 이용자 프로파일과 데이터베이스에 自然語(Natural Language)를 入力시키고 文字 対 文字의 매칭(CHARACTER BY CHARACTER CHECKING) 方式을 사용하기 때문에 同意語라든가 綴字의 変形, 뜻의 차이가 거의 없는 表記上의 失手 등에 대한 融通性(FLEXIBILITY)이 없다. 예를 들면 「DATABASE DEVELOPMENT」와 「DATABASE CREATON」, 「AUTOMATIC INDEXING」과 「COMPUTERIZED INDEXING」등 심지어 「, 」「·」「-」등의 있고 없음 즉 「DATA-BASE」와 「DATA-BASE」는 물론 「COMPUTERIZATION」과 「COMPUTERISATION」의 「Z」와 「S」의 変形 등 이용자 입장에서 보면 전혀 의미의 차이가 없는 단어들을 한단어로 취급하지 못하는 것이다.

우리는 이와 같은 制約点을 미리 予見하였고, 또 이를 解決하는 方法 즉 디소오러스(THETHAURUS) 作成을 통한 어휘통제(VOCABULARY CONTROL)를 시도하고 싶었으나 디소오러스 作成은 많은 시간을 요하는 작업이기 때문

에 졸업일자에 쫓기던 우리로서는 이 方式을 포기하지 않을 수 없었다. 그러나 비록 디소오러스 작성을 통한 어휘통제의 自動化는 実現하지 못했으나 그대신 手作業과 어휘의 左右側切捨(RIGHT-LEFT TRUNCATION) 方法을 사용하여 앞서 지적한 제약점을 최소화하려고 노력했다.

어휘 切捨란 데이터베이스에는 正式어휘(FULL SPELLING)를 入力시키고 이용자 프로파일의 검색 서술자는 切捨가 가능하도록 한 것이다.예를 들어 「COMPUTERIZATION」, 「INFORMATION」과 같은 데이터베이스의 어휘는 검색 서술자를 「COM #」 「INFO #」와 같이 縮語를 사용할 수 있도록 했고 그다음 어휘와의 논리적 斷切 및 縮語임을 表示하기 위한 「#」(더블해치)를 부착시켜 컴퓨터가 나머지 文字는 읽지 않고 뛰어넘도록 操作하였다.

또한 手作業을 통한 어휘 통제는 데이터 베이스와 이용자 프로파일을 作成할 때 두 파일에 入力된 어휘중에서 同意語나 綴字変形을 갖고 있는 어휘는 미리 골라내어 가장 대표적인 어휘로 統一하였고 데이터베이스에 入力될 名詞形 어휘는 반드시 復数形으로 入力시킴으로써 검색서술자보다 어휘의 길이가 같거나 크도록 함으로써 同一한 어휘가 単復数形의 차이때문에 검색되지 않는 경우를 予防하였다. 그러나 이와 같은 手作業에 의한 어휘 통제는 두 파일의 데이터量 이 제한된 실험용으로서는 그 효과를 충분히 발휘할 수 있으나 레코드의 축적량이 수천件 이상의 본격적인 SDI시스템에서는 일일이 두 파일의 어휘를 対照 確認하는 것이 不可能할 것이므로 디소오러스 작성에 의한 自動어휘통제 方法이 쓰여져야 할 것이다.

우리가 매칭 시스템으로 設計한 文字 対 文字 매칭 시스템은 검색시간이 많이 걸린다는 문제를 제기시킨다. 만일 하나의 이용자 프로파일에 관한 문헌을 검색하기 위하여 데이터베이스에 入力된 모든 데이터를 文字 하나 하나別로 모두 읽어 나간다면 엄청난 시간이 소요될 것이며 이러한 매칭시스템은 컴퓨터의 비능률적 이용이라는 비판을 면치 못한다. 이러한 短点을 補完하기 위하여 우리는 컴퓨터가 검색 서술자앞에 書誌事項番号를 먼저 읽고 데이터베이스의 해당 書誌事項에 직접 접근할 수 있도록 매칭시스템을 設計하였으나 최근에는 이보다 더 빠르고 능률적인 시스템이 개발되고 있다. 즉 일련번호(Serial No.), 서술자, 書誌情報로 구성되는 데이터 파일로부터 各 어휘마다 데이터 파일에 출현하는 頻度数와 変換파일에 대한 포인터(POINTER)를 포함하는 辞典体파일(DICTIONARY FILE)을 만들고 다시 이 파일을 기초로 各 어휘들의 문헌속에서의 위치를 나타내는 変換파일(INVERTED FILE)을 작성하여 컴퓨터가 어휘의 문자를 하나하나 읽어 나가는 대신 사전체 파일에 있는 어휘코드와 変換파일에 있는 어휘의 위치를 확인하고 이러한 어휘 또는 어휘의 組合(SET OF DESCRIPTORS)을 포함하고 있는 문헌을 검색해 내는 가장 진보적인 매칭시스템으로서 KORSTIC이 美 IBM社로부터 도입하여 CAC, INSPEC, ISMEC, USGRA 등의 데이터베이스에 대한 SDI서비스를 실시하고 있는 STAIRS(STORAGE and INFORMATION RETRIEVAL SYSTEM)도 이 범주에 속한다. 精巧하고 보다 더 效果的인 SDI서비스를 위해서는 검색된 문헌들을 이용자의 관심도에 따라 그 우선순위를 부여해 주는 시스템이 요청된다. 이와 같은 시스템을 개발하기 위한 방법으로는 검색 서술자에 이용자의 관심 정도에 따른 상대적 비중(Relative Weightage)을 매기는 방법과 중요한 어휘일수록 하나의 문헌에서 出現하는 頻度가 높다는 점에 착안하여 이용자 프로파일작성시 검색서술자의 出現頻度를 삽입하는 방법이 사용되고 있으며 세계적으로 널리 이용되고 있는 「COMPENDEX」데이터베이스는 「CARD-A-LERT-CODE」시스템을 개발하여 사용하고 있는데 이는 후자의 방법을 이용한 것이다. 이와 같은 精巧한 SDI시스템은 검색된 문헌의 適中率(PRECISION RATE=NO. OF RELEVANT DOCUMENT RETRIEVED/NO. OF DOCUMENT RETRIEVED)을 크게 증가시키는 반면 검색률(RECALL RATE=NO. OF DOCUMENT RETRIEVED/NO. OF DOCUMENT IN THE FILE)은 감소시킨다.

우리가 개발한 시스템은 검색 서술자마다 동