

유한요소 구조해석 프로그램의 전후처리 접속장치의 설계 Data-Exchange Interface Design of Pre-& Post-Processing System for Finite Element Structural Analysis Program

신 영 식* · 서 진 국**

Young-Shik Shin*, Jin-Kook Suh**

<Abstract>

In general, FORTRAN is used for numerical analysis and OPS5 or LISP is used for expert systems. This causes problems at the interface because the various applications require different computing languages or environments. This paper describes the approach used to take AutoCAD as a user-interface for an existing finite element structural analysis package. Some principles concerning database management related to data-exchange interface of pre- and post-processing system for FORTRAN structural analysis program are discussed, and numerical examples demonstrate the power of the combination of these programs.

Key Words : AutoCAD, Postprocessor, Preprocessor, Structural Analysis Program, User-interface

1. 서 론

근래에 유한요소 구조해석용 패키지(package)도 윈도우즈용으로 개발되어 이전보다 훨씬 더 개선된 사용자 환경을 제공하고 있는 추세에 있다. 그러나 공학적 문제해결에 가장 많이 애용되어 왔던 언어인 FORTRAN으로 된 구조해석용 소프트웨어도 수치해석용 모듈(module)로서, 교육용으로 또는 전문분야용 소규모 프로그램으로서 그 명맥을 유지하고 있는 실정이다. 이 언어는 절차적인(procedural) 특징을 가지고 있어서 수치해석용으로는 적합하

나, 그래픽 처리가 매우 불편하다는 단점을 가지고 있다. 따라서 FORTRAN 프로그램들은 GUI(Graphic User Interface)개념을 도입한 전후처리가 쉽지 않으므로 수치로만 구성된 파일 입출력 방법을 사용할 수밖에 없다.

반면, 윈도우즈용 언어는 사용자와의 친숙성을 구현하는 것이 그 주된 개발배경이므로 이를 활용하여 다양한 그래픽 입출력 형식을 쉽게 설계, 구현할 수 있다. 그러나 이러한 방법으로 완전한 윈도우즈용 구조해석 패키지를 개발하는 것은 부분적으로 전후처리기만을 개발하는 것에 비하여 많은 시간과 경비가 요구된

* 정회원, 영남대학교 공과대학 토목공학과 교수·工博
Tel: 053-810-2415

Professor, Dept. of Civil Engineering, Yeungnam Univ.
Ph.D.

** 정회원, 경동정보대학 토목과 조교수·工博
Tel: 053-850-8231

Assistant Professor, Dept. of Civil Engineering,
Kyungdong College, Ph.D.

E-mail: jksuh@kyungdong-c.ac.kr

“이 논문은 1995학년도 영남대학교 학술연구조성비 지원 연구에 의한 것임”

다고 할 수 있다.

따라서 본 연구에서는, 구조해석 모듈은 수치 해석에 적합한 장점을 가지고 있는 기존의 FORTRAN 모듈을 그대로 사용하고, GUI 개념이 사용자 환경에 가장 필요한 전후처리기(pre- & post-processor)만을 설계하고자 한다. 또한, 전후처리기의 설계에서도 그 개발시간 및 비용을 줄이기 위하여 가장 범용적으로 사용되고 있는 AutoCAD[1,2]를 기본환경으로 사용함으로써 가장 간편한 전후처리 접속장치(interface)를 설계하고자 한다.

2. 전후처리 접속장치의 구성

2.1 프로그래밍 도구

(1) AutoCAD 및 프로그래밍 언어

주처리기(main processor)와 별도로 전후처리기를 설계, 제작할 때에는 그래픽 프로그램 전체를 직접 개발하거나 기존의 그래픽 프로그램과 접속할 수 있는 모듈만 개발하는 방법 등을 사용할 수 있을 것이다.

전자의 경우는 개발자가 원하는 요소들로 구성하여 다양한 기능들을 모두 포함할 수도 있으나 전후처리 모듈을 개발하는 것 외에 그래픽 도구를 추가로 제작하여야 하므로 개발시간 및 비용이 많이 들게 되어 특정 전문용 패키지를 개발해야 하는 경우에는 오히려 부수적인 작업이 훨씬 더 많아져 원래의 개발초점에서 어긋나게 된다. 반면 후자의 경우는 그래픽 도구[2]가 미리 만들어져 있으므로 전후처리 모듈 중 수치해석 모듈과 연계된 부분만을 설계 및 코딩함으로써 신속한 구현이 가능하다.

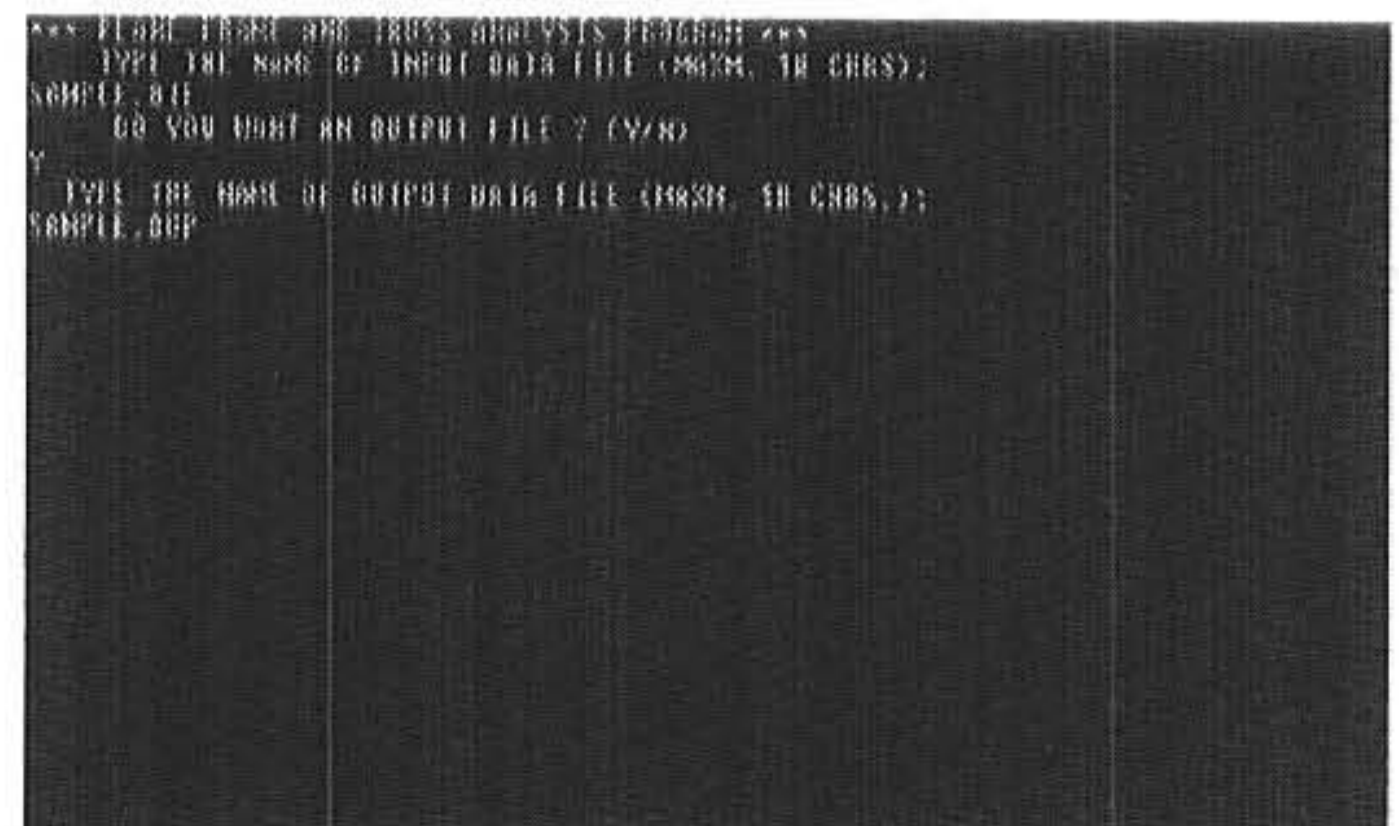
본 논문에서는 구조해석용 전후처리기의 간편한 설계방법에 관한 것이 연구의 목적이므로 후자의 방법을 택하였다. 그래픽 프로그램으로는 가장 범용적이며, 필요한 자료(data)를 추출, 가공하기가 용이하다는 장점을 가지고 있는 AutoCAD를 사용하였다. AutoCAD는 일종의 데이터베이스(database) 개념으로 제작되었기 때문에 AutoCAD에서 자체적으로 사용되는 자료 처리 형식을 그대로 사용하고, 필요한 자료를 여기에 추가해서 활용할 수 있다. 따라서, 본 연구에서는 2차원 평면이나 3차원 작업이

가능하며, 각 개체(entity) 마다 자료를 검색하고 필요한 자료를 질의(query)할 수 있는 장점을 이용하여 이 것을 수치해석을 위한 입력자료 파일의 형식(input data file format)으로 만들고, 해석결과를 CAD용 자료로 받아들여 다시 개체로 전환시키는 접속장치를 설계하였다.

AutoCAD에서는 내부에서 자체적으로 지원해주는 언어를 가지고 있으므로 CAD를 활용하기 위하여 외부에서 별도로 'Port'시킬 필요가 없어진다. 이 언어는 인공지능 언어인 LISP(List Processor)의 라이브러리를 확장하여 만든 AutoLISP[3]를 말한다. 그 외에, AutoCAD에서는 Visual-Basic과 C++도 지원하고 있다. 그러므로 그 중 하나의 언어만을 알고 있어도 이와 같은 전후처리 모듈을 개발하거나 확장시키기에 용이하다. 여기서는 자체 지원 언어인 AutoLISP를 사용하였다.

(2) 구조해석용 프로그램

본 연구에서 주처리 모듈의 예로 사용된 구조해석용 프로그램(NEWFR)[4]은 평면 뼈대구조의 선형 정적 해석을 수행할 수 있는 전형적인 학습용 FORTRAN 수치해석 프로그램이며, 이와 유사한 다른 FORTRAN 프로그램을 주처리 모듈로 사용할 수도 있다.



[Fig. 1] FORTRAN 프로그램의 실행장면

[Fig. 2] FORTRAN 프로그램 실행 결과의 예

Fig.1과 Fig.2는 FORTRAN 구조해석용 프로그램의 실행 화면과 그 결과 형식의 한 예를 보여준다.


```

** FORTRAN OUTPUT FILE
TITLE: ----SAMPLE----
THE DISPLACEMENTS ARE:
JOINT NO. ROTATION   VERT DEFL  HORIZ DEFL
1      .00000000    .00000000   .00000000
2     -0.38604890   -0.00084694 -1.32200700
3     0.39069910   -0.00081114 -1.32216600
4     0.00000000    .00000000   .00000000
MEMBER END FORCES ARE:
MBR-NO.  J-M    K-M    J-SHR  K-SHR  J-AX  K-AX
1      -2247.16 -4352.75  -43.20  43.20  239.47 -239.47
2      4352.75 -4279.80  239.47  238.55  43.21  -43.21
3      4279.80  2063.30  43.18  -43.18  238.55 -238.55
    
```

2.2 설계의 개념적 모델

일반적인 프로그램의 처리기는 전체적으로 다음의 3단계로 구성되어 있으며, 입출력기용 접속장치를 설계하기 위하여 필요한 각 단계별 입력자료의 정보들을 설명하였다.

(1) 전처리 단계

이 단계에서 요구되는 입력자료는 기하학적 형상과 재료의 물성(material property)에 관한 것들이며, 이 정보들은 본 연구에서 설계 구현된 DCL(대화상자)을 이용하여 직접 입력, 수집된다. 그 후 전처리 접속장치를 이용하여 대상이 되는 수치해석용 프로그램의 입력 자료 파일을 생성함으로써 그 정보가 주처리 모듈에 전달된다.

(2) 수치해석 모듈과의 접속 단계

별도로 외부에서 차용한 주처리 모듈을 이용하여 구조해석을 수행하며, 이 단계에서는 배치 파일(batch file)을 이용한 일괄처리로 수행된다. 전처리의 결과로 생성되는 파일의 자료로 후처리의 기본 입력 자료를 생성하기 위하여 그 자료를 전달하는 접속장치를 필요로 한다.

(3) 후처리 단계

전처리 단계의 입력자료 중 기하학적 형상 등에 관한 기초자료를 바탕으로 구조해석 결과를 출력(plotting)하게 된다.

상기에 설명된 요소들로 접속장치를 구성하면 후처리기만 있는 소프트웨어의 도화(drawing) 작업에 관련된 과정을 전처리기에서 해결하게 되고, 전처리기에서는 부가적으로 재료의 물성만 사용자에게 입력받음으로써 전후처리기의 모든 기능을 구현할 수 있게 된다.

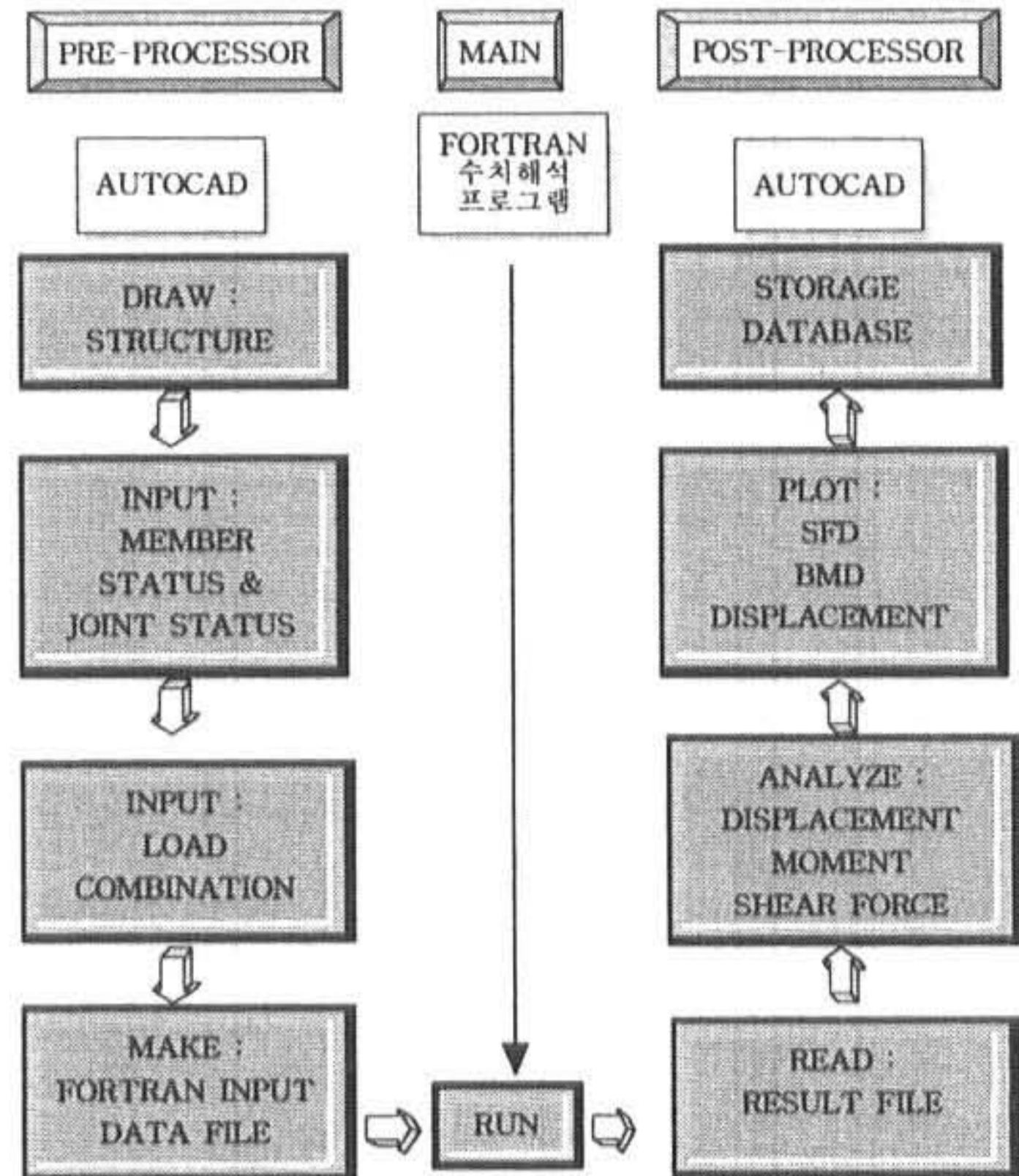
다음 Fig. 3은 각 단계별 자료변환의 전체적

인 흐름을 나타낸 것이다.

3. 전후처리 접속장치의 설계

접속장치의 설계 및 구현과정을 설명하기 위하여 부정정 뼈대 구조를 예제로 하여 전후처리 각 단계별로 모델링하는 과정을 나타내었다.

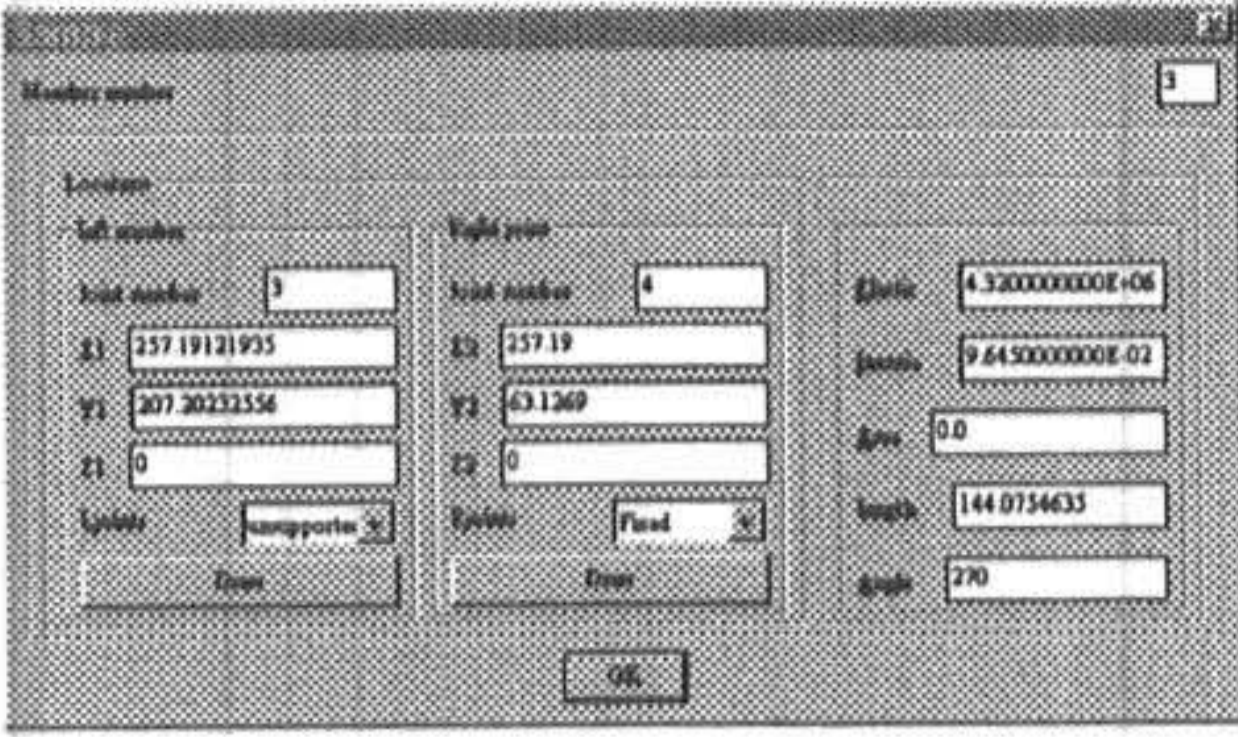
3.1 전처리 접속장치의 설계와 구현 예



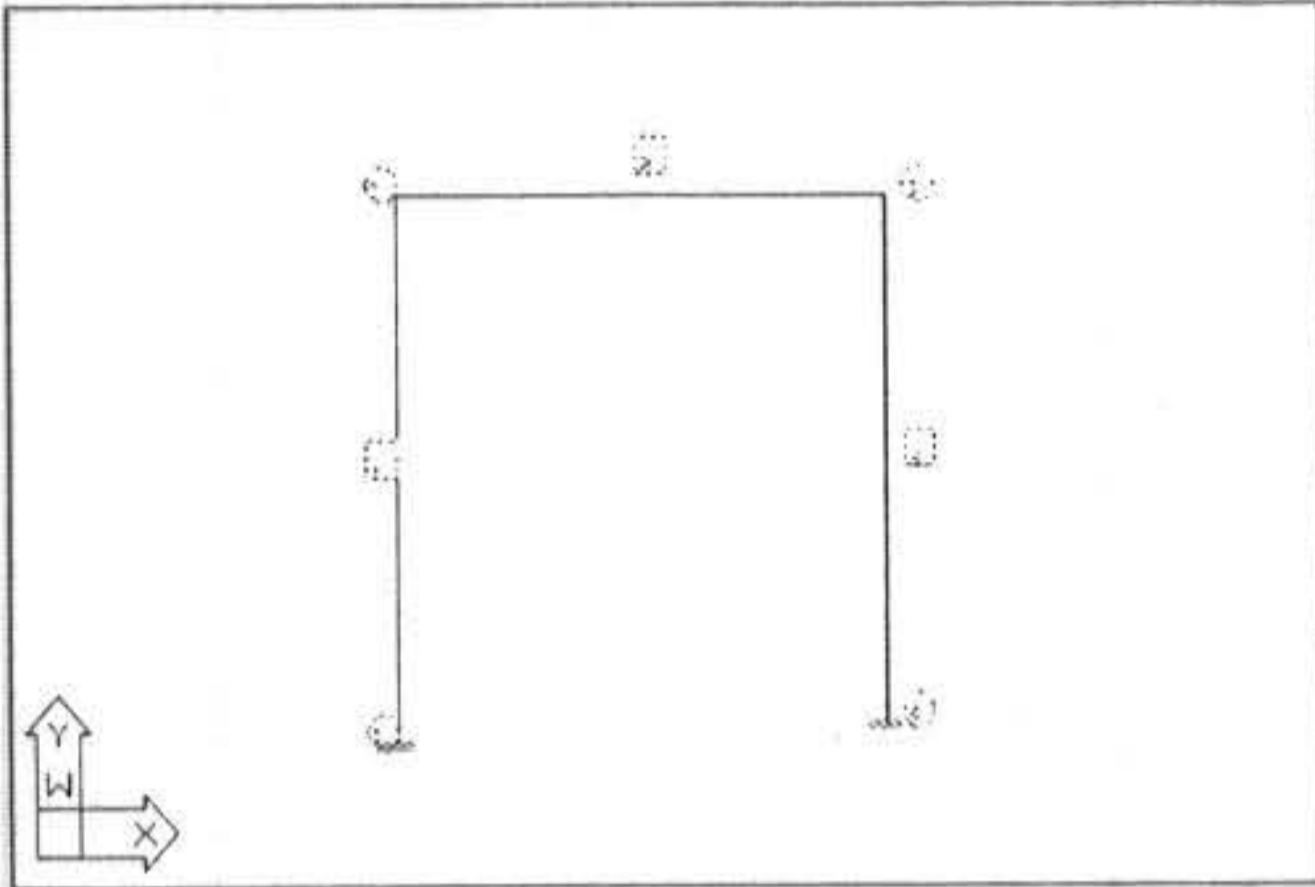
[Fig. 3] 자료변환의 전체적 흐름도

(1) 구조 형상의 작도

해석하고자 하는 구조 형태를 CAD에서 그리기 전에 부재의 치수를 실제 축척으로 하느냐, 임의 축척(free scale)으로 하느냐를 먼저 결정해야 한다. 본 연구에서는 전후처리기의 전체적 구성을 통해 볼 때 실 축척이 구조물의 거동을 파악하기 쉽고 작업하기가 용이하기 때문에 이 방법으로 도면을 세팅(setting)하여 미리 작업 환경을 조정한다. 기본 세팅이 끝나면 해석하고자 하는 구조를 실제 축척에 맞도록 상대 좌표 개념을 사용하여 구조물의 상호 관계에 따라 작도한다. 이 자료들은 기하학적 성질에 대한 정보로 저장되어 수치계산시 기본적인 자료가 됨과 동시에 후처리기에도 이용된다.



[Fig. 6] 부재 및 절점의 상태 설정



[Fig. 7] 부재 및 절점 정보를 가진 모델

(3) 하중상태 입력

부재에 재하되는 하중들을 사용자로부터 입력받아 그 자료를 저장한다.(Fig. 8, Fig. 9)

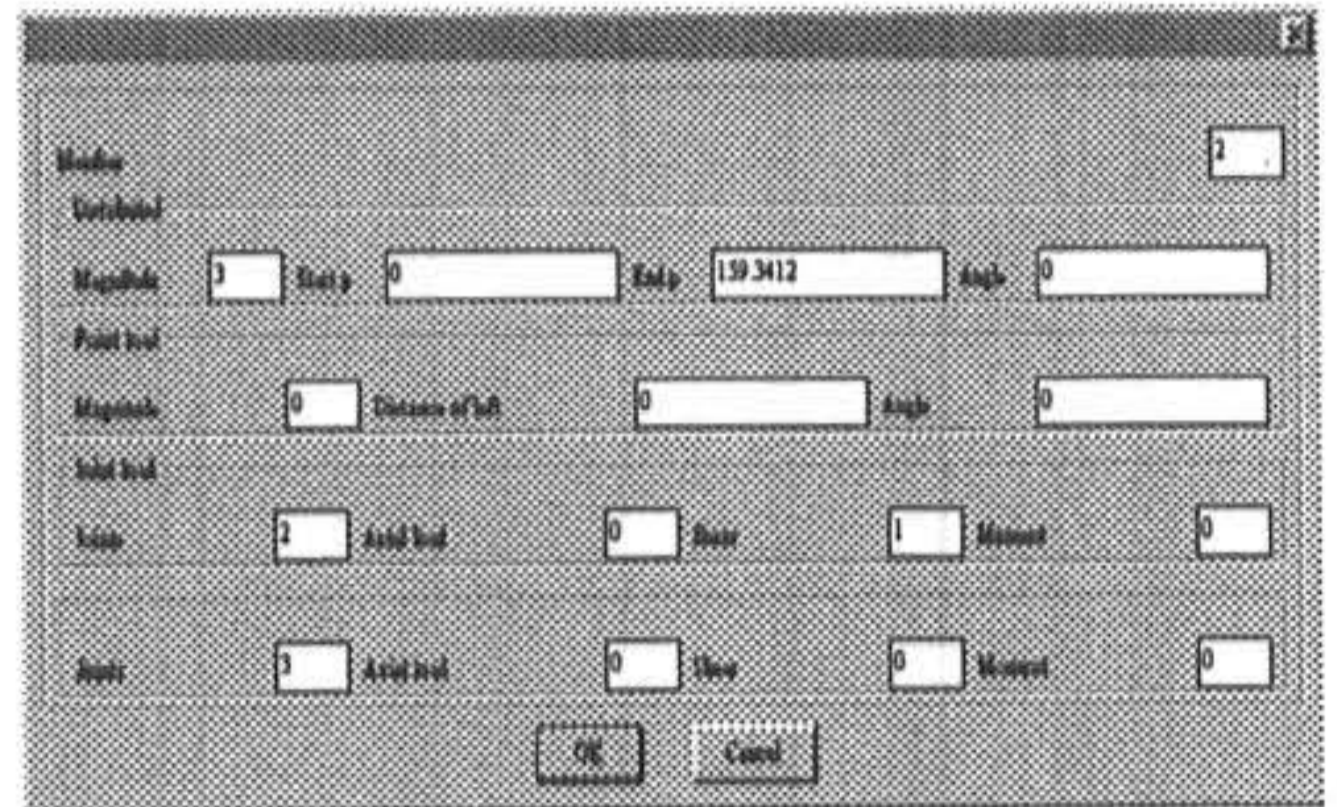
```
// 확장 자료에 하중 조합을 추가하여 저장
(setq pdata (list -3 (list "xdata"
                        (cons 1000 "FOR" )
                        (cons 1002 "(" )
// 미리 저장된 기본 자료
(cons 1070 imembers )
(cons 1070 iljoints )
(cons 1070 irjoints )
.....
// 하중 조합의 추가
(cons 1040 fmagnitued)
(cons 1040 fspl)
(cons 1040 fepl)
(cons 1040 fangd)
(cons 1040 fmagnituedep)
(cons 1040 fddistance)
(cons 1040 fangp)
(cons 1040 flaxial)
```

```
(cons 1040 flshear)
(cons 1040 flmoment)
(cons 1040 fraxial)
(cons 1040 frshear)
(cons 1040 frmoment)
(cons 1002 ")" )
)
```

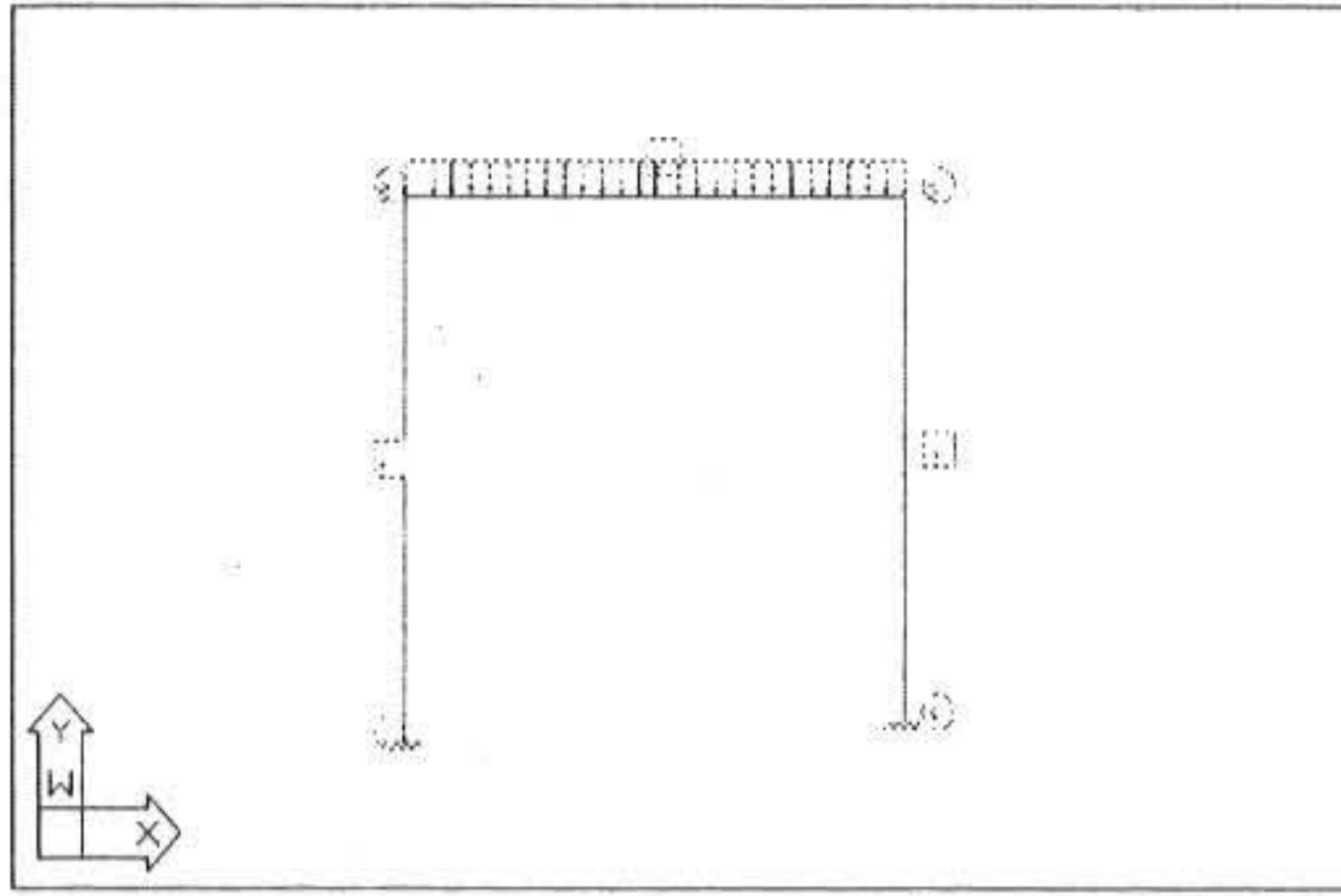
```
(setq edata (subst pdata extdata edata))
// UPDATE DATA
(entmod edata)
```

(4) 입력 자료 파일의 생성

전처리기에서 화면에 그린 것은 단지 보여주기 위한 기능이다. 그러나 도면에 그려진 것은 CAD 내에서 자료화되어 있고 또한 사용자가 여기에 첨가하여 입력한 정보는 확장 개체 자료(extended entity data)의 정보로 저장되어 있다. 이것은 개발자가 기존의 자료를 활용할 수 있도록 지원해 주는 기능을 CAD에서 가지고 있다는 것을 의미이다. 즉, 자체적으로 필요한 자료를 검색하고 추출하는데 데이터베이스 검색엔진을 따로 만들 필요 없이 기존의 CAD가 제공하는 라이브러리를 이용하여 필요한 기능을 가공할 수 있도록 해준다.



[Fig. 8] 하중 조합 설정



[Fig. 9] 하중이 재하된 상태

모든 자료가 첨가된 최종적인 개체의 형태는 다음과 같다.

```
((-1. <Entity name: 2280540>) (0. "LINE") ( 5.
"28") (100 . "AcDbEntity") (67 . 0 ) (8 . 0 )
(100 . "AcDbLine") ( 10 30.7478 30.2476 0.0)
(11 42.7478 30.2476 0.0) (210 0.0 0.0 1.0) (-3
("XDATA" (1000 . "FOR") (1002 . "(") (1070 .
2) (1070 . 2 ) (1070 . 0 ) (1070 . 0) (1040 .
12.0 ) (1040 . 0.072338) (1040 . 1.0) (1040 . 2.0)
(1040 . 0.0) (1040 . 21.0) (1040 . 0.0) (1040 .
0.0) (1040 . 0.0) (1040 . 0.0) (1040 . 0.0) (1040 .
0.0) (1040 . 0.0) (1040 . 0.0) (1040 . 0.0) (1040 .
0.0) (1040 . 0.0) (1002 . ")"))))
```

위의 형태는 LISP의 연상 리스트(list) 기법으로서 하나의 리스트가 이루는 모양은 '(CAR CDR)' 이다. 여기서는 여러 개의 쌍으로 이루어진 리스트로 각각의 연상을 나타내는데, 이때, CAR부를 키(key)로 그 나머지를 키에 연상되는 자료로 지정하는 형태의 자료구조를 가지므로 키를 사용하여 자료에 접근하게 된다. 기본적인 형식은 '((키1 . 자료1) (키2 . 자료2) ... (키n . 자료n))' 이다.

여기에서 생성된 자료 리스트를 임의의 FORTRAN 수치해석 프로그램 입력 파일의 자료 형식으로 가공할 수 있으며, 전처리 과정에서 독립적으로 자료를 생성할 수도 있다.

다음은 FORTRAN 구조해석 프로그램의 하나인 NEWFR의 입력 파일의 자료 형식으로 변환시키는 과정을 보여준다.

```
// 도면의 모든 개체를 선택
(while(setq entname(ssname readxdata temp1))
// 추가된 확장정보와 기하학적 자료의 추출
    (setq edata (entget entname ("xdata")))
// 기하학적 자료
    (setq fpoint (assoc 10 edata)
        spoint (assoc 11 edata))
    (setq fxpoint (nth 1 fpoint)
        fypoint (nth 2 fpoint)
        sxpoint (nth 1 spoint)
        sypoint (nth 2 spoint))
// 재료의 성질과 하중조합에 대한 자료의 추출
    (setq extdata (assoc -3 edata))
    (if (/= extdata nil)
        (progn
            (setq xdata (cadr extdata))
            (setq members (cdr (nth 3 xdata)))
            (setq nljoints (cdr (nth 4 xdata)))
            (setq nrjoints (cdr (nth 5 xdata)))
            (setq sljoints (cdr (nth 6 xdata)))
            .....
            .....
            (setq angp (cdr (nth 17 xdata)))
            (setq laxial (cdr (nth 18 xdata)))
            (setq lshear (cdr (nth 19 xdata)))
            (setq lmoment (cdr (nth 20 xdata)))
            (setq raxial (cdr (nth 21 xdata)))
            (setq rshear (cdr (nth 22 xdata)))
            (setq rmoment (cdr (nth 23 xdata)))
// 재료의 성질과 하중조합의 새로운 자료 리스트 생성
            (setq pdata (list (list members
                (cons 1 nljoints)
                (cons 2 nrjoints)
                (cons 3 sljoints)
                (cons 4 srjoints)
                (cons 5 elastic)
                .....
                (cons 21 fxpoint )
                (cons 22 fypoint )
                (cons 23 sxpoint )
                (cons 24 sypoint )
                ))) ; list
// 각 개체의 리스트를 전체로 묶음
            (setq arraydata (append arraydata pdata))
        ) ; progn
```



```

) ;if
  (setq temp1 ( + 1 temp1))
  ) ; while
// 지점에 대한 자료를 검색하여 리스트를 만듦
(setq readdata (ssget"x" '((-3 ("joint")))))
(setq sslen (sslength readdata))
(setq temp 0)
(setq arrayjdata nil)
(while (setq entname (ssname readdata temp))
  (setq edata (entget entname '("joint")))
  (setq extdata (assoc -3 edata))
  (if ( /= extdata nil)
    (progn
      (setq joindata (cadr extdata))
      (setq joinum (cdr (nth 3 joindata)))
      (setq jointangle (cdr (nth 4 joindata)))
      (setq jdata (list (list joinum jointangle)))
// 지점에 대한 리스트를 하나의 리스트로
      (setq arrayjdata (append arrayjdata jdata))
    )
  )
)
(setq temp ( + 1 temp ))
)
// 각 부재의 부재번호, 왼쪽 절점 번호, 오른쪽
// 절점 번호, 단면적, 탄성계수, 단면 1차 모멘트
// 등의 리스트
(setq memstatus (append memstatus (list (list
memberofnum tempdata1 tempdata2 area elastic
inercia))))
// 각 절점 좌표와 절점 연결 상태
(setq jointstatus (append jointstatus (list (list
tempdata1 (list fxpoint fypoint tempdata3))))))
(setq jointstatus (append jointstatus (list (list
tempdata2 (list sxpoint sypoint tempdata4))))))
// 부재에 재하된 하중 조합의 리스트
(if (or ( /= memberload1 0.0)( /= memberload2
0.0))
  (progn
    (setq memberload (+ memberload 1))
    (setq memberload (append memberload
(list (list memberofnum
memberload1 memberspl memberepl
memberangd memberload2
memberdist memberangp))))))
  )
)
// 왼쪽 절점에 재하된 하중조합

```

```

(if (or (or ( /= jointload1 0)( /= jointload2 0)) ( /=
jointload3 0))
  (progn
    (setq jointload (append jointload (list
tempdata1)))
    (setq jointload (append jointload (list
(list tempdata1 (list jointload1 jointload2
jointload3))))))
  ) ; progn
  ) ; if
// 오른쪽 절점에 재하된 하중조합
(if (or (or ( /= jointload4 0)( /= jointload5 0))( /=
jointload6 0))
  (progn
    (setq jointload (append jointload (list
tempdata2)))
    (setq jointload (append jointload
(list (list tempdata2 (list jointload4 jointload5
jointload6))))))
  ) ; progn
  ) ; if

```

** FORTRAN 입력 파일의 작성

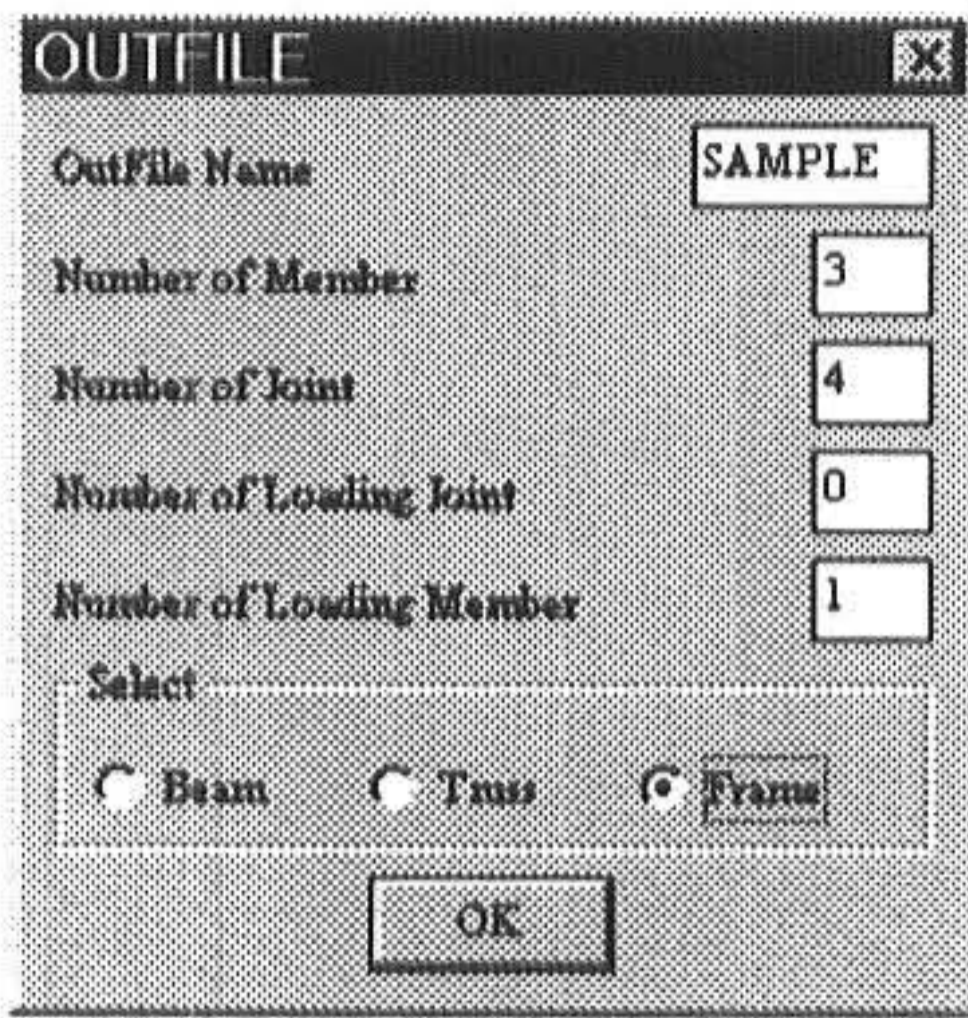
```

// NO.1: TITLE
// NO.2
(setq line2 (strcat smemnum "," stempdata ","
sflag "," sjlength "," smemberload))
(write-line line2 op)
// NO.3: 모든 하중 작용 절점에 대한 정보 작성
(setq line3 (strcat stt1 "," smoment "," sshear
"," saxial))
// NO.4: 등분포 하중이 재하된 모든 부재에
// 대한 정보 작성
(setq line4 ( strcat slomemno "," sdint "," sdspl
"," sdepl "," sdang "," smagpoint "," spdist ","
spang))
)
(write-line line4 op)
// NO.5: 모든 부재에 대한 정보 작성
(setq line5 (strcat smemnum "," cljoinum ","
crjoinum "," carea "," cmodulus "," cinercia))
(write-line line5 op)
// NO.6: 모든 절점에 대한 정보 작성
(setq line6 (strcat sjoinum "," sxdist "," sydist
"," ))
(setq line6 (strcat line6 sfljoints "," sjoinangle
"," "0" "," "0" ))

```



```
(write-line line6 op)
(setq nnn6 ( 1+ nnn6))
```



[Fig. 10] FORTRAN 입력 파일 생성용 대화상자

```

----SAMPLE----- // NO.1
3,4,3,0,1 // NO.2
2,3,0,159.3412,0,0,0,0 // NO.4
1,1,2,10,4320000,9.6450000000E-02 // NO.5
2,2,3,10,4320000,9.6450000000E-02
3,3,4,10,4320000,9.6450000000E-02
1,97.85,54.4155,3,0,0,0 // NO.6
2,97.85,207.2023,0,0,0,0
3,257.1912,207.2023,0,0,0,0
4,257.1912,60.3125,3,0,0,0
    
```

* 여기서, 절점 하중이 없으므로 NO.3는 없다.

[Fig. 11] 생성된 FORTRAN 입력 자료 파일

3.2 후처리 접속장치의 설계와 구현 예

후처리 접속장치는 구조해석을 수행한 후 그 결과 자료들의 정보를 분석하여 그래픽 정보로 변환하는 과정을 포함한다.

(1) 계산된 결과 파일의 분석

전술한 Fig. 1과 같은 전형적인 FORTRAN 결과 파일이 생성되는데, 이러한 자료 정보를 추출, 이용할 때 후처리 접속장치가 요구된다.

```
// 각종 변위 정보의 추출
(setq jointdisplace (append jointdisplace (list
(list dgijointnum dgfrotation dgfvertical
dgfaxial))))
```

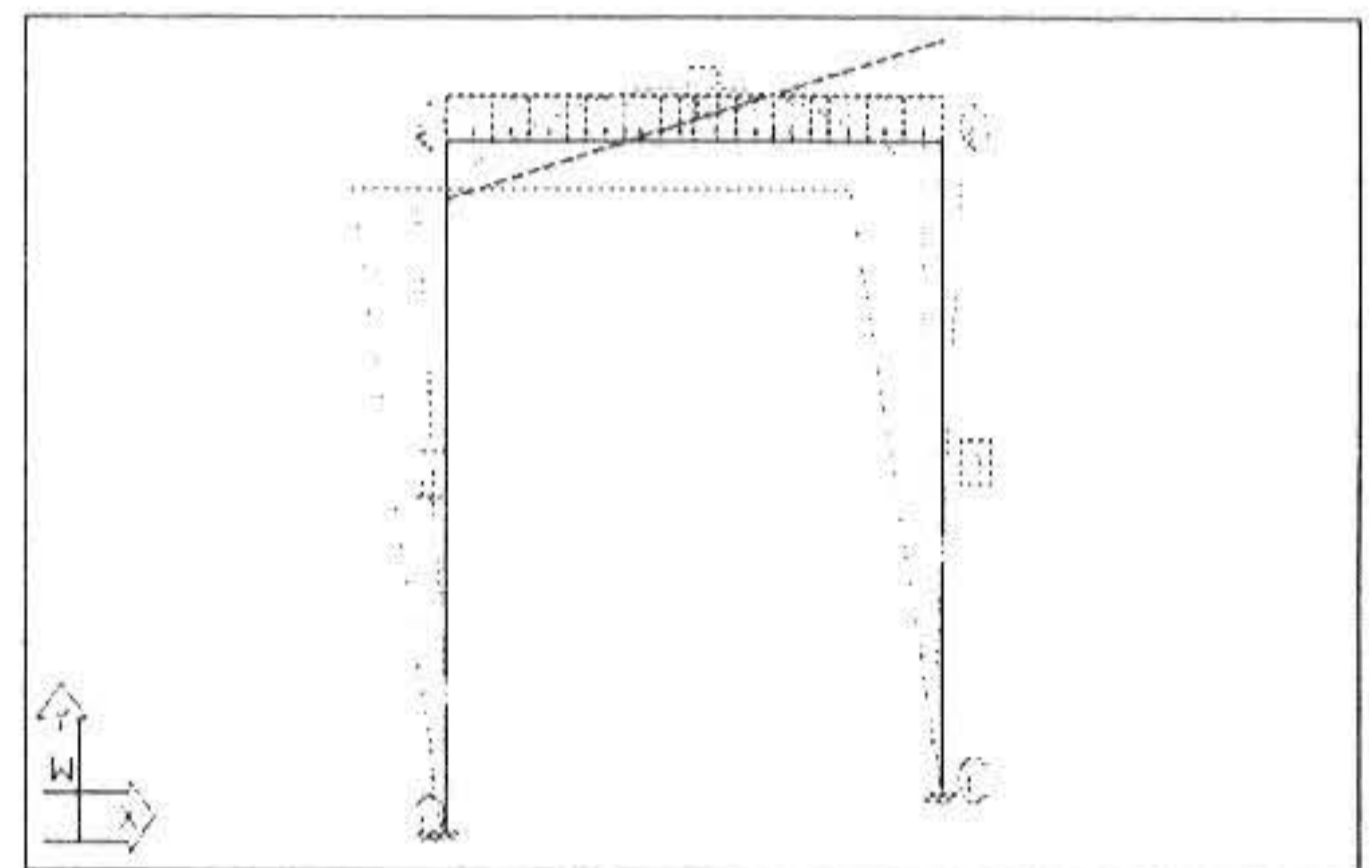
```
// 단면력 정보의 추출
(setq memberrigidity (append memberrigidity
(list (list dgimemnum dgfjmoment dgfkmoment
dgfjshear dgfkshear dgfjaxial dgfkaxial))))
```

(2) 변위 및 단면력 정보의 분석

- **변위도**: 각 절점의 변위 값에 관한 정보는 절점별 회전변위 값과 수직 및 수평변위 값으로 계산될 수 있다.
- **전단력도**: 이것은 각 절점의 전단력 값과 전처리 단계에서 주어지는 하중들의 조합을 이용하여 유도된다.
- **휨모멘트도**: 전단력도를 이용하여 미세하게 세분화된 면적을 적분하여 부재의 모든 지점의 모멘트를 구할 수 있다.

(3) 변위 및 단면력 정보의 작도

다음의 Fig.12와 같은 그래픽 정보가 최종 출력된다.



[Fig. 12] 최종 출력된 결과 화면

4. 결론

본 논문에서는 FORTRAN으로 작성된 기존의 구조해석용 프로그램에 편리한 사용자 환경을 부가적으로 제공하기 위하여 간편한 전후처리를 설계, 구현하였다. 이 전후처리는 개발비용과 시간을 절감하기 위하여 가장 흔히 사용되는 그래픽 패키지인 AutoCAD를 이용하여 그 접속장치를 설계하였는데, 이 때 자료의 교환은 기존 FORTRAN 프로그램이 채택하고 있는 파일에 의한 전달방식을 사용한다. 여기에

서는 데이터베이스의 4GL(4th generation language)의 개념을 설계개념으로 활용하였는데, 이것은 DBMS(database management system)와 사용자 사이에 어떤 특정한 메커니즘을 설정하고, 이 메커니즘에서는 데이터베이스 엔진과 사용자간의 접속장치를 제공한다. 사용자가 자료에 대해 질의하고 제어하기 위하여 그 접속장치를 사용한다면 사용자는 동시에 다른 데이터베이스 엔진을 사용해도 무방하게 된다. 결국 사용자는 이러한 4GL만 사용하면 되고, 데이터베이스 엔진은 이 4GL만 지원하면 되기 때문에 사용자는 4GL을 지원하는 데이터베이스 엔진을 동시에 사용할 수 있게 되는 것이다. 이 방법을 채용하면 전후처리는 주처리 프로그램에 독립적인 응용 프로그램이 되므로, 동일한 전후처리에 FORTRAN으로 작성된 임의의 수치해석 프로그램을 접속하여 사용할 수 있는 장점을 가진다.

참고문헌

- [1] *ADS Programmer's Reference - AutoCAD Release 14*, Autodesk (1998)
- [2] *Advanced Tools - AutoCAD Release 14* Autodesk (1998)
- [3] *AutoLISP Reference - AutoCAD Release 14*, Autodesk (1998)
- [4] Basu, P. K. and Woo, K. S., *NEWFR* Vanderbilt University (1986)
- [5] Medland, A. J., *The Computer-Based Design Process*, Chapman and Hall, New York (1992)
- [6] Machover, C., *The CAD/CAM Handbook* McGraw-Hill, New York (1996)
- [7] Pilkey, W. D., *Formulas for Stress, Strain and Structural Matrices*, John Wiley and Sons, New York (1994)
- [8] Earle, J. H., *Engineering Design Graphic - AutoCAD Release 12*, Addison-Wesley Publishing Company, Texas (1994)
- [9] Wertz, H., *An Introduction to Programming in LISP*, John Wiley and Sons, New York (1988)
- [10] Deal, R. M., *Getting to Know LISP*, Brown Publishers, Iowa (1991)
- [11] Leondes, C. T., *Computer-Aided Design/Engineering(CAD/CAE) - Techniques and Their Application*, Academic press, New York (1993)

(1999년 6월12일 접수, 1999년 10월10일 채택)