

DAISY 3000 -  
A NEW APPROACH IN TEXT PROCESSING

TIMO RAUNIO

## DAISY/3000

### A NEW APPROACH IN TEXT PROCESSING

Once upon a time, Mr. Gutenberg discovered typography. It was a revolution in **information distribution**.

When IBM released the electronic typewriter with type ball and later advanced it by including magnetic memory, we were talking about **word processing**.

Only a few years ago did the rush of all kind of microprocessor controlled equipment with display units and floppy discs begin. Now we can use the term **text processing**, although their different features, hardware and prices cause confusion in our minds.

However, now we are standing at the front door of **integrated information processing**, which includes all conventional data processing as well as high quality printed output, electronic archives, automatic reproduction and mailing, etc. A key to slightly open that door is **DAISY/3000**, a program which fills the gap between your existing systems and high quality formatted output with reasonable costs, and meanwhile, satisfies all of your normal text processing demands, until we have "the paperless office".

### WHAT IS DAISY/3000

DAISY is a high level **document description language** with which almost any kind of document can easily be produced. It fully utilizes the intelligence of the Diablo printer and Hyfeed sheet feeder. Over 70 basic level commands are recognized by DAISY and these commands are used to define even higher level **macro commands**. To obtain high quality, even very complicated, formatted output, the user only inserts a few names of these previously defined macros within the text.

In addition to all normal text processing features, some sophisticated capabilities, such as graphics, user-written procedures, conditional execution etc., are included.

## COMMAND LEVELS

**Printer level** commands are escape sequences which are hard to remember and difficult to use. The user does not have to worry about them with DAISY.

**Basic level** commands have mnemonic names and are much easier to use. They also generate all of the required escape sequences for the desired operation. However, the user seldom has to be concerned about them, unless he is the person creating macro libraries.

**Macro level** commands are the main tools used when entering texts to be formatted. Macro commands are named and defined by the user. Thus, their names and function can be designed according to the user's demands.

## MACRO LEVEL BENEFITS

As known in the other programming languages where user defined macros are available, the macros are the key to effective and structured programming. The same also applies to text processing and especially to flexible text entry.

Some noticeable benefits of macro level commands are:

- Short commands do a lot. Thus, complex series of commands or often used phrases can be referred to by a single name.
- Flexibility. Just changing the macro library will result in a totally different output format or changing a macro definition will have its effect throughout the text.
- Mnemonic names can be used in any language.
- Less errors. A correctly designed macro does it right always and there remain only typing errors which rarely can be avoided.
- Less work. Once designed, macros can always be used.
- The user can concentrate on WHAT he is writing, NOT HOW he is writing.
- No regressive way of thinking. Each macro can cancel all unwanted states set by other macros. Thus, each macro STARTS a block of text.
- No need to adjust the text on the display. The user simply writes the text and the macros always do the proper formatting.

## RESTRICTIONS

DAISY/3000 is hardware dependant. It has been written in SPL, which is known only by HP computers. Conversion to PASCAL may be possible.

The escape sequences controlling the output devices have no standards, and neither do the features of printers. Therefore, only one printer could be selected as the output device, so that all of the advanced formatting features and special effects could be realized. Thus, formatted output is available only with the DIABLO printer.

Even if the escape sequences could be converted, a display terminal cannot show all the necessary formatting. An example: A very normal situation is to have margins at columns 9 and 79. With proportional spacing, this area contains 85-100 characters per line and an 80 column display will cut the line and could not possibly show the right justified margins.

## DAISY: COMMANDS

This section describes the basic level commands by classes. More detailed information can be found in the "DAISY reference manual" and in the application notes.

Command classes are:

- Define files
- Define macros
- Modes of operation
- Size of page
- Vertical movement
- Horizontal movement
- Special effects
- Automatic titles and footnotes
- Graphics
- Automatic table of contents and keyword index
- Conditional commands
- User procedures
- Hyphenation
- Miscellaneous
- End
- Edit

The commands can be located anywhere within the text and are indicated by a "command character". Some commands have numeric parameters which can be an integer expression containing variables and constants.

## USER PROCEDURES

If the capabilities of DAISY are not satisfactory, the user can write his own procedures or subprograms in SPL or FORTRAN. DAISY then loads such routines from the user's SL file when desired and the procedure can return commands and/or text for DAISY to process.

This feature allows direct linkages to existing systems, data bases and registers. Also, user-written "preprocessors" for the input can easily be done to totally change the default syntax of commands.

## INTERACTIVE APPLICATIONS

User-written procedures can be used to obtain interactive text processing applications. However, DAISY itself includes commands to make conversation with the user.

In many cases, it is desirable to alter some of the formatting parameters during execution. Form size and printwheel characteristics are two such parameters. Also, you may have a fixed text where only a few words should be changed from time to time. Editing of such text is unnecessary if you use interactive commands.

## EDIT

There are several methods to create input files or write texts for DAISY. One way is EDIT/3000, which as known is a very powerful tool when entering or editing any kind of source texts. EDIT/3000 is available in the normal way or as a son process of DAISY through the EDIT command.

Although EDIT/3000 is quite a meritorious program, it lacks a few desirable features when entering and editing texts in human languages. Therefore, DAISY has a built-in, full-screen editor which operates in block mode and fully utilizes the local edit facilities of HP2640 series terminals. (2620 series terminals are under investigation).

## DAISY IN PRACTICE

When compared to those stand-alone micro-monsters, DAISY has a totally different theory of operation. Which one is better depends on what you are doing. If you are only replacing a typewriter, DAISY may be too sophisticated. But, if you want real text processing, the macro level commands show their power. And if you are planning linkages to your existing systems or to raise your integration level, there are not too many other possibilities available.

## EXAMPLES

The following four pages show some examples of DAISY applications. They are not examples of any ordinary text processing applications which any system can do. Rather, they show some of the flexibility and graphical power of DAISY.

The first example shows part of an application note which describes a convenient way to draw flowcharts. This macro library consists of about 30 defined macros, 20 of which are not visible to the user. The flowchart in the example was drawn with 12 lines of input, using 9 of these predefined macros.

The second example shows the usage of multiple printwheels. It was printed with 3 different wheels: CUBIC PS, APL and SCIENTIFIC.

The third one is an example of multicolumn printing.

The last one shows mathematical expressions with user-defined, special-symbol macros, using graphical mode.

DAISY is developed in HELECON, HELSINKI SCHOOL OF ECONOMICS

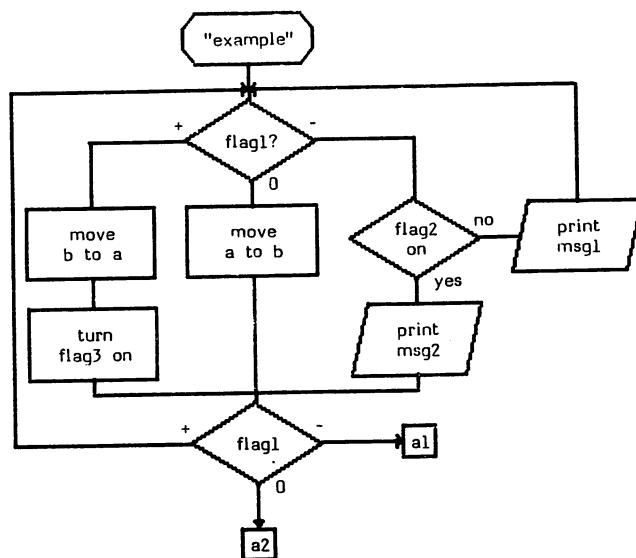
Distribution: Oy PORASTO Ah  
Töölöntullinkatu 8  
00250 Helsinki 25  
FINLAND  
Telex: 125194 PSTO SF

Helsinki 1.6.1981

DAISY: application note 2/81

## HOW TO MAKE FLOWCHARTS WITH DAISY

Macrolibrary "FLOWCH" provides a convenient way to draw flowcharts. Example:



User input defines which symbols are used and where they are located, text within symbols and where to draw lines.

Daisy automatically calculates the form of symbols, horizontal and vertical centering of the texts, start and end points of the lines, location of arrowheads, location of "yes/no-like" labels around the "if" symbol and the scale of the picture.

## 2.2 Ensimmäisen kertaluvun predikaattikalkyyli

Nimikkeistö  $L$  koostuu joukosta vakio- funktio- ja predikaattisymboleja. Jokaiseen funktio ja relaatiot symboliin liittyy kokonaisluku, joka ilmaisee symbolin paikkaluvun. Täsmällisemmin,

$L = (\text{Const}, \text{Func}, \text{Rel}, \text{ar}),$

missä

Const -- vakiosymbolien joukko,

Func -- funktiosymbolien joukko,

Rel -- relaatiot symbolien joukko ja

ar: Func  $\cup$  Rel  $\rightarrow \mathbb{N}$ .

Kun  $\text{ar}(f) = n$ ,  $f \in \text{Func}$ , sanomme, että  $f$  on  $n$ -paikkainen funktiosymboli. Vastaavasti, kun  $\text{ar}(r) = n$ ,  $r \in \text{Rel}$ , sanomme, että  $r$  on  $n$ -paikkainen relaatiot symboli.

Nimikkeistön  $L = (\text{Const}, \text{Func}, \text{Rel}, \text{ar})$  struktuuri on pari  $M = (M, I)$ , missä  $M$  on ei-tyhjä arvojoukko ja  $I$  on kuvaus joukolla Const  $\cup$  Func  $\cup$  Rel, siten, että

$I(c) \in M$  kun  $c \in \text{Const}$ ,

$I(f): M^n \rightarrow M$  kun  $f \in \text{Func}$ ,  $\text{ar}(f) = n$  ja

$I(r) \subset M^n$  kun  $r \in \text{Rel}$ ,  $\text{ar}(r) = n$ .

Käytämme usein merkintää  $c^M = I(c)$ ,  $f^M = I(f)$  ja  $r^M = I(r)$ .

Esimerkiksi ryhmäteoriassa käytetään nimikkeistöä

$L_1 = (\{0\}, \{+, \cdot\}, \emptyset, \text{ar})$

missä  $\text{ar}(+) = 2$ .  $L_1$  :n struktuuri on pari  $M = (M, I)$ , missä  $I(0) \in M$  ja  $I(+): M^2 \rightarrow M$ .

Lukuteoriassa taas käytetään nimikkeistöä

$L_2 = (\{0\}, \{S, +, *\}, \emptyset, \text{ar})$ ,

missä  $\text{ar}(S) = 1$  ja  $\text{ar}(+) = \text{ar}(*) = 2$ . Eräs tämän kielen struktuuri on  $M = (M, I)$ , missä  $0^M = 0 \in \mathbb{N}$ ,  $S^M(n) = n+1$ ,  $+^M(m, n) = m+n$  ja  $*^M(m, n) = m \cdot n$ , ts näillä symboleilla on standardi merkityksensä.

Seminaariharjoitukset tapahtuvat pienryhmissä (yhteensä 8-15 henkilöä). Niissä käsitellään keskustelemalla joko teoreettisia tai käytännöllisiä kysymyksiä. Seminaariharjoitusten yhteydessä opiskelijat laativat usein myös esitelmää keskustelun pohjaksi ja joissakin aineissa käsitellään seminaareissa myös heidän tekemiään tutkimussuunnitelmia. Seminaareissa on hyvä tilaisuus oppia tieteellisen keskustelun periaatteita ja tutustua oppiaineen metodiiikkaan. Ne auttavat tehokkaasti syventymään omaan aineeseen.

Ei ole liioiteltua väittää, että useimpien akateemisten aineiden opetus on nykyään niin tehokasta, että opiskelijoiden todella kannattaa seurata sitä niin paljon kuin mahdollista. Oppitunteihin, koekielisiin töihin ja harjoituksiin osallistuminen on useimmiten tentissä onnistumisen välttämätön edellytys. Harjoitukset antavat tarpeellisen taidon selviytyä soveltavista tentti tehtävistä, ja seminaareissa saa tärkeää ohjausta tieteelliseen työhön.

### 1.3.4 Aktiivisuuden merkitys

Ratkaisevaa on kuitenkin, että opiskelija ei vain "istu" erilaisissa opetustilaisuuksissa. Ainoastaan osallistumalla aktiivisesti työskentelyyn hän voi saada jotakin irti opetuksesta. Sen lisäksi hänen täytyy käyttää hyväkseen jokainen tilaisuus keskusteluun opettajan kanssa ja tehdä tälle kysymyksiä. Opettajat yleensä arvostavat sitä, että opiskelijat seuraavat heidän opetustaan aktiivisesti ja kiinnostuneina.

## 1.4 OPINTOJEN SUUNNITTELU

### 1.4.1 Pitkän tähtäyksen ohjelma

Tärkeämpiä johtopäätöksiä, joita voidaan tehdä teoriaosan perusteella, on se, että yhden ja saman aineen liian keskitetty oppiminen pitkähkön ajanjakson aikana tuskin on järkevää. Sen tähden on hyvä suunnitella vaihteleva pitkän ajan ohjelma, jotta välttyttäisiin tehottomilta ja usein masentavilta oppimistasanteilta.

Tehokkaan opiskelumetodiikan tärkeä apuväline on siis mahdollisimman yksityiskohtainen suunnitelma. Pitkän tähtäyksen suunnitelman on havaittu vaikuttavan positiivisesti opiskeluun. Jokainen saavutettu välitavoite antaa opiskelijalle tyydytyksen tunteen ja lisää opiskelutarmoa auttaen siten voittamaan seuraavan tehtävän mukanaan tuomat alkuvaikeudet.

Päämäärän tulee luonnollisesti olla realistinen: jos on asettanut itselleen liian suuret vaatimukset - toivoen siten kannustavansa omat kykynsä äärimmilleen - alkaa jonkin ajan kuluttua tuntea haluttomuutta huomataessaan, että päämäärää ei voikaan saavuttaa. Itseltään liian vähän vaatiminen veltostuttaa ja opiskelija oppii vielä vähem-

män kuin oli suunnitellut. Tehokkainta on asettaa päämääränsä siten, että sen voi tavoittaa maksimaalisen ponnistelun avulla. Sen on siis oltava saavutettavissa, mutta vasta todellisen työpanoksen jälkeen. Ei riitä, jos suunnittelee "ehtivänsä niin ja niin pitkälle aineessa ensimmäisten viikkojen aikana". On vähintään yhtä tärkeää yrittää suunnitella työnsä yksityiskohdat viikko viikolta ja päivä päivältä.

### 1.4.2 Lue sopivasti

On niinkään hyödyllistä kehittää kiinteä työskentelytapa. Myös työskentelytapa on suunniteltava realistisesti siten, että sitä voidaan noudattaa pitkän ajanjakson kuluessa. Sen tähden on työskentelysuunnitelmaan jätettävä tilaa esim. urheilulle, ostosmatkoille, kirjallisuudelle ja sanomalehdille, järjestötoimintaan osallistumiselle, elokuvissa ja teatterissa käynneille ja muille virkistysmahdollisuuksille. Näin välttytään suunnitelman alituiselta muuttelemiselta ja ennen kaikkea säästytään siltä tunteelta, että olisi "epäonnistuttu" hyvissä aikomuksissa. Voit esimerkiksi hankkia päiväkirjan, jossa jokainen päivä on jaettu tunteihin, ja merkitä siihen suunnitelmasi. Usein on sopivinta aloittaa merkitsemällä tenttipäivät, ja sitten laskea ne tenttiä edeltävät päivät, jotka tarvitsee kertaukseen ja sen jälkeen jakaa jäljellä oleva aika niiden kurssin osien kesken, jotka täytyy suorittaa, jotta ohjelma pitäisi paikkansa.

Yliopistoissa ja korkeakouluissa tiettyyn järjestelmään sidottu opetus on muotoiltu siten, että se vaihtelee viikottain. Silloin on vielä jokaista viikkoa harkittava erikseen. On kuitenkin miltei aina mahdollista sovittaa opiskelutösi kiinteisiin aikoihin ja täten luoda itselleen työskentelyrutiini, jota voi noudattaa.

### 1.4.3 Suunnittele yksityiskohtaisesti

Ajankäyttö ei suinkaan ole ainoa asia, joka pitää suunnitella sopivaa kaavaa noudattaen. Myös opittava aines tulee organisoida hyvin. Kurssin jokaisen osan kohdalla on suunniteltava, kuinka paljon aikaa kuluu kurssikirjojen, muun kirjallisuuden ja luentomuistiinpanojen lukemiseen, paljonko aikaa kuluu mahdollisiin harjoitustehtäviin ja esitelmien laatimiseen, ja lopuksi, kuinka paljon aikaa on varattava kertaukseen ennen loppuententtiä. Kertaamiseen varattu aika on usein tehokkaammin käytettyä kuin alkuperäinen oppimisaika. Edelleen voi olla hyvin edullista lukea eri kirjoja tai kurssin kohtia rinnakkain.

### 1.4.4 Realistinen päämäärä

Akateemisten opintojen suunnittelussa on erittäin tärkeää, ettei opiskelija yliarvioid omaa opiskelukapasiteettiaan. Suhteellisen vapaa opiskelu tarjoaa kyllä mahdollisuudet tähän. Monet opiskelijat aloittavat liian optimistisesti monta ainetta

the mean percentage mean absolute error

$$M = \frac{1}{n} \sum_{i=1}^n \left( \frac{1}{T} \sum_{t=1}^T \left| \frac{A_{it} - P_{it}}{A_{it}} \right| \right) 100 \quad \begin{matrix} t = 1, \dots, T \\ i = 1, \dots, n \end{matrix}$$

and

the mean percentage root mean square error

$$R = \frac{1}{n} \sum_{i=1}^n \left( \frac{1}{T} \sum_{t=1}^T \frac{\sqrt{\sum (A_{it} - P_{it})^2}}{\sqrt{\sum (A_{it} - \bar{A})^2}} \right) 100 \quad \begin{matrix} t = 1, \dots, T \\ i = 1, \dots, n \end{matrix}$$

are used.<sup>1</sup>

3. It is possible that an equation has a good statistical fit, but a poor tracking fit. This is due to the dynamic properties of the model which bear little relation to the way individual equations fit the historical data. It is for this reason that M and R tracking errors are an important criterion for the evaluation of a multi-equation model.

4. In the following comparisons of calculated vs. observed variable values will be performed between the AJKA and the ETLA models and trend forecasts:

- 1 for periods 1958-78, 1971-78 and years 1977 and 1978
- 2 for 22 common endogenous variables of AJKA and ETLA
- 3 for some of the most important variables separately.

<sup>1</sup> These are documented e.g. in Johnston, H.N., Klein L.R. and Shinjo K.: 'Estimation and Prediction in Dynamic Econometric Models', Chapter 2 in Sellekaerts, and in Pindyck and Rubinfeld.