

미니 명세서의 구조적 한글 표현에 관한 방법론

조 성 희

숙명여자 대학교 전산학과

A Methodology on expression of structured HANGUL for Mini_Specification.

Sunghee Cho

Department of Computer Science, Sookmyung Women's University

요 약

본 논문은 효율적인 소프트웨어 개발을 위한 수단으로 구조적 분석 기법을 사용하는데 구조적 분석 도구중의 하나인 미니 명세서를 한글화하고 나아가서는 전반적 자동화 연구에 있어 비 전문 사용자가 쉽게 접할수 있도록 한글화를 실현하고자 하는데 그 목적을 둔다.

1. 서 론

최근 하드웨어의 개발기술의 발전에 따라 하드웨어 개발비용은 하락해가는 대신 소프트웨어에 투자되는 비용은 고도의 기술이 요구됨으로 인해 상대적으로 급증하게 되고 이로인해 소프트웨어의 자동화 생성에 관한 연구가 필수적으로 요구되어 지고 있다. 소프트웨어의 양적, 질적공급이 요구에 충족되어지지 못할때는 소프트웨어 위기에 봉착하게 된다. 소프트웨어의 자동화 생성에 관한 연구에 앞서 구조적 분석 기법은 소프트웨어 질적 향상을 위한 해결수단으로 연구되어졌다.[5]

여기서 구조적 분석이란 정해진 문제해결의 방법과 규칙을 가지고 큰 시스템을 작은 시스템으로 나누어 가면서 처리방법을 알아내 가는것을 말한다. 이를 위해서는 아래와 같은 도구를 사용한다.[1]

- 1). 자료 흐름도 (Data Flow Diagram)
- 2). 자료 사전 (Data Dictionary)
- 3). 미니 명세서 (Mini Specification)

위와 같은 도구를 사용하여 새로운 시스템 명세서를 만드는 것을 구조적 분석이라 할 수 있겠다.

본 논문에서 연구하는 바는 구조적 분석 도구중의 미니명세서를 표현하는데 있어서의 기존 문제중의 가장 큰 문제라고 할수 있는 미니 명세서의 한글화, 나아가서는 전반적 자동화 연구의 한글화 문제를 해결해 보자는데 그 의의를 둔다.

미니 명세서는 처리과정의 논리를 기술해 주는데 있어 일상언어가 아닌 구조적 언어, 의사 결정표, 의사 결정 나무(Decision Tree)등으로 표현된다.[1] 그러나 현재, 이는 모두 영문으로 기술하고 있으며 이로 인해 시스템이나 전산에 대한 전문 지식이 없는 현업 사용자들이 요구 사항을 표현하기 위해서는 난관에 부딪히는 빈번한 문제를 초래하게 되고 여러가지 제약을 가져온다. 따라서 사용자가 요구사항을 적당하게 표현하고 구조적 분석의 효율과 나아가서는 자동화 시스템을 구축하기에 이르러까지 우선적으로 미니 명세서의 한글 표현은 중요한 역할을 감당하리라 생각된다.

II. 본 론

미니 명세서의 구조적 한글 표현에 관한 연구를 위해 먼저 구조적 분석의 도구를 살펴볼 필요가 있다. 자료 흐름도, 자료사전 그리고 미니 명세서에 대해 본 장에서 간단히 언급한뒤, 한글의 특징, 특히 영어와의 차이점을 중심으로 한글의 특징을 살펴봄으로 인해 미니 명세서를 한글화하는데에 적합한 한글의 구조를 설정하는 과정을 보이도록 하겠다.

1. 구조적 분석 도구

구조적 분석의 목적을 달성하는데에는 다음과 같은 세개의 도구가 필요하다.

1). 자료 흐름도 (Data Flow Diagram)

시스템을 기능 형태별로 분할하여 표현할수 있는 도구

자료 흐름도의 구성요소는 자료 흐름선(Data Flow), 처리기(Processor), 화일(File) 또는 자료 저장소(Data Store), 관련자(Terminator)로 되어 있다.

2). 자료 사전 (Data Dictionary)

모델을 구성하고 있는 부문간의 접속요소를 엄격하게 정의해준다.

다시 말하면 자료 흐름도에 나타나는 자료 흐름선, 화일, 자료요소등과 같은 자료항목을 특별한 약속된 기호를 가지고 그 내용을 알기 쉽게 정의 해놓은 것이다. 약속된 기호는 표 1.에 나타내었다.

기호	의 미
=	정의 (is equivalent to OR is composed of)
+	구성(and)
[]	택일(select only one)
{ }	반복(iteration)
()	생략가능(option)
* *	설명(comment)

표 1. 자료사전 정의에 사용되는 약속된 기호와 그 의미

3). 미니 명세서 (Mini Specification)

처리과정의 논리를 기술해주는 것으로서 일상 언어가 아닌 구조적 언어, 의사 결정표, 의사 결정 나무(Decision Tree)등으로 표현한다.

자료 흐름도와와의 연관성을 설명한다면 자료의 이동을 나타내는 것은 자료 흐름도에 자료의 구성 내용은 자료 사전에 정의하고 자료의 변환을 나타내는 처리기의 내용을 명료하게 설명해 놓은 것을 미니 명세서라고 하는 것이다. 가장 적합한 미니 명세서의 크기는 한 페이지정도로 작성된 것이다. 이는 요구 표현의 명확한 전달과 복잡성의 기피를 위해 제안된 크기이다.

위의 분석 도구를 이용 하는데 있어 보다 효율적이고 비 전문가도 능숙하게 사용하도록 하고자하는 노력이 현재 각 업체 및 학계에서 진행되어 자료 흐름도의 작성, 자료 사전의 산출, 자료 흐름도의 처리기를 통해 산출되는 미니 명세서의 자동 산출을 생성하는 CASE tool의 완성에 대한 연구가 활발히 진행되고 있다. 나아가서는 생성된 미니 명세서에서 자동적으로 프로그래밍 언어를 생성해 내는 Automatic programming분야에 이르기까지 자동화에 관한 연구가 계속 추진되고 있으며 이러한 도구들이 하루빨리 한글화된 산물까지 이르기를 기대한다.

2. 한글 미니 명세서의 구현

2.1 한글의 특징 [4]

① 언어의 공통점

모든 언어는 자의적인 음성기호의 체계라는 점과 모든 언어는 어느것이나 자음과 모음으로 나뉘어 진다는 것, 그리고 반드시 명사와 동사가 있고 문장은 주어와 서술어로 구성된다. 이러한 보편적인 특질을 공유하면서 각 언어는 나름대로의 특질을 따로 갖는다.

② 한글의 어순

언어마다 문장을 구성하는 방식은 약간씩 다르지만 문장의 구성을 크게 나누어 보면 '주어 + 목적어 + 동사', '주어 + 동사 + 목적어', '동사 + 주어 + 목적어'의 세가지로 나눌수 있다. 한글은 이 중에서 '주어 + 목적어 + 동사'의 어순을 따른다. 이러한 어순을 갖는 한글에서는 몇가지 문장 구성상의 특징이 있다. 모든 문법적 형태소는 반드시 어근이나 어간 뒤에 온다는것, 즉 조사는 체언뒤에 붙어 쓰이며 활용어미는 용언의 어간뒤에 쓰인다.

이러한 제한을 제외한다면 국어에서는 문장 성분의 순서를 비교적 자유롭게 바꾸어 말할수 있다는 특징을 갖는다.

예를 든다면 "순희가 착하다고 말했다",
"착하다고 순희가 말했다",
"말했다 착하다고 순희가"

와 같이 말해도 그 의미가 같게 전달되어 질수 있다는 것이다.

③ 조사의 갈래

우리 국어에서 조사는 앞의 말이 모음이냐 자음이냐에 따라 교체되며 이렇게 음운론적 조건에 따라 교체되는 형태들을 음운론적 이형태라고 부른다. 조건에 따라 형태를 달리하는 조사에는 다음과 같은것들이 있다.

가) 는/은, 를/을, 야/아

나) 과/와, 으로/로, 이고/고, 이며/며, 이나/나, 이든지/든지, 이나마/나마/, 인들/ㄴ 들, 이랑/랑, 이라도/라도

다) 이/가

④ 존대법

국어의 존대법에는 규칙적인 용언의 활용에 의한 것으로서 주체 높임법, 상대 높임법이 있고 특수 어휘에 의한 것이 있다. (예: 밥->진지, 딸->따님,...) 하지만 존대에 대해서는 논의하지 않고 구조적 한글 표현을 위해서는 명령문, 평서문의 형태를 사용하기로 한다.

⑤ 그 밖의 특징

가) 주어 가 없는 문장이 많다.

2.2 구조적 한글표현

구조적 분석단계에서 미니 명세서를 표현할때 프로그래밍 언어로 직접 작성한 다든지 pseudo-code로 작성한다면 이용자들은 이해하기 어렵고 작성을 기피하게 된다. 반면 일상언어로 작성을 한다면 의사 소통에 곤란을 가져오게 되어 혼돈을 일으킬수 있게 되므로 구조적 프로그래밍의 기본 제어 구조와 제한된 일상언어를 택하여 미니 명세서를 작성하기 위한 구조적 언어를 한글화하여

확립해 놓은것을 구조적 한글 표현이라 하였다.[1] 구조적 한글의 표현에 있어서 문장의 구성에 따라 적당한 semantic와 keyword를 산출하는 과정을 아래에서 설명한다.

① 치환문 (assignment statement)

가장 기본적인 치환문의 형태는 우변의 논리식, 산술식, 관계식 혹은 변수의 값, 상수 등을 좌변의 변수에 기억시키는 문장이다. 결과적으로 expression의 결과, 즉 value나 변수의 값을 어떤 변수에 치환(기억)시키는 의미를 가지며 치환문의 일반적인 형식은 아래와 같다.[2]

$$\text{변수} = \left[\begin{array}{c} \text{산술식} \\ \text{논리식} \\ \text{관계식} \\ \text{변수} \\ \text{상수} \end{array} \right]$$

문장으로서 표현하면

_____ 을(를) _____ 에 (으로/로) 치환하라

로 나타낼수 있다.

치환문의 예를 영문으로 표현하면 MOVE A TO B or ASSIGN A TO B 가 되겠지만 한글 표현으로서는 변수와 동사 즉 명령어의 어순이 바뀌어야 하며 어순은 다음의 두가지를 모두 포용하도록 했으며, 변수뒤의 조사는 한글 변수를 허용토록 함에 따라 앞의 말이 모음이나 자음이나에 따는 교체성을 허용한다.

A 를(을) B 에(으로/로) 치환하라.

B 에(으로/로) A 를(을) 치환하라.

혹은 Equal sign에 의해 B = A 표현도 사용한다.

그리고 산술식을 표현하는데 있어서는 산술기호 '+, -, /, *'를 사용하며 변수로의 치환은 equal sign에 의한 표현에 대신하도록 하였다.

$$\text{평점} = (\text{국어} * \text{수강시간1} + \text{수학} * \text{수강시간2} + \text{영어} * \text{수강시간3}) / (\text{총수강시간})$$

② 입, 출력문

계산, 처리에 필요한 데이터를 읽어들이며 주기억 장치에 기억시키는데 필요한 명령어가 입력문이고 반대로 기억장치에 기억되어 있는 내용을 적당한 출력 장치를 통해 표시해 주도록 하는 명령어를 출력문이라고 한다.

입, 출력문에 해당하는 영문 명령어들은 다음과 같은 것들이 있다.

INPUT, OUTPUT, READ, WRITE.

참고되어야 할 명령어로는 OPEN과 CLOSE가 있다.

문장 구성 형식을 분류하여 살펴보면

가) INOUT, OUTPUT

먼저 영문 형식을 검토한뒤 한글과 비교 하겠다.

INPUT문과 OUTPUT문은 OPEN과 같이 사용하며 의미는 File을 OPEN하는데 그 File이 INPUT을 위한것이나, OUTPUT을 위한 것이나, 아니면 INPUT과 OUTPUT을 같이 수행하기 위한 것이나를 알려주는것이 OPEN문이다.

영문 형식은 일반적으로 다음과 같다.

```
OPEN [ INOUT  filename1, {filename2},....
      OUTPUT filename1, {filename2},....
      I-O    filename1, {filename2},.... ]
```

이에 대응하는 한글 표현은 명령어인 OPEN 과 file을 지시하는 구가 어순의 차이에 의해 교체되어 다음과 같이 표현할수 있다.

```
[ 입력파일  filename1, {filename2} .... ] 을(를) 연다.
  출력파일  filename1, {filename2} ....
  입출력파일 filename1, {filename2} .... ]
```

반대로 open한 file을 닫을때는 CLOSE문을 사용하는데

영문은 CLOSE filename1, {filename2}

한글표현으로는 filename1, {filename2} ... 을(를) 닫는다.로 나타낸다.

나) READ, WRITE

OPEN된 file에서 record를 읽어들이거나 내용을 쓸때 사용하는 명령어로서
영문 :

```
READ filename RECORD [ KEY IS data-name]
```

```
WRITE record-name
```

한글 :

filename 파일로부터 [키는 data-name으로(로)] 레코드를 읽는다.
record-name을(를) 레코드에 쓴다.

③ 제어문

program을 효과적으로 작성하기 위하여 실행 순서를 변경시키거나 일시 중
단, 개시, 반복등을 위해 사용하는 명령어이다.

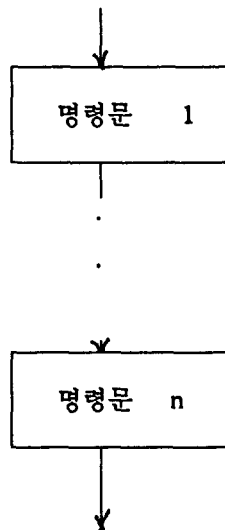
특히 이 제어문에서는 또 세 구조로 구분할수 있다.

이 기본 제어 구조 세가지로 모든 알고리즘이 표현될수 있다는 것이 1966년
jacopini에 의해 수학적으로 증명이 되었다.

세계의 기본 제어구조는 아래에 설명한다.

가) 순차 구조 (sequence buliding block)

문장들이 순차적으로 처리되어지는 구조



예를 들면 근태파일로부터 근태레코드를 읽는다.

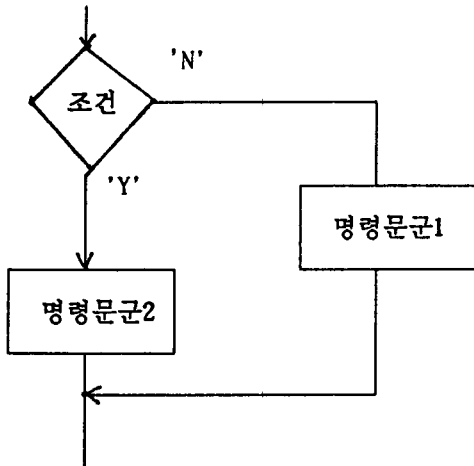
급여 = 근무일수 * 일당

나) 선택 구조 (select buliding block)

두개 이상의 명령문들 중에 하나 명령문을 처리하는 구조이다.

영문에서는 일반적으로 IF __ THEN __ ELSE or CASE문이 여기에 속한다.

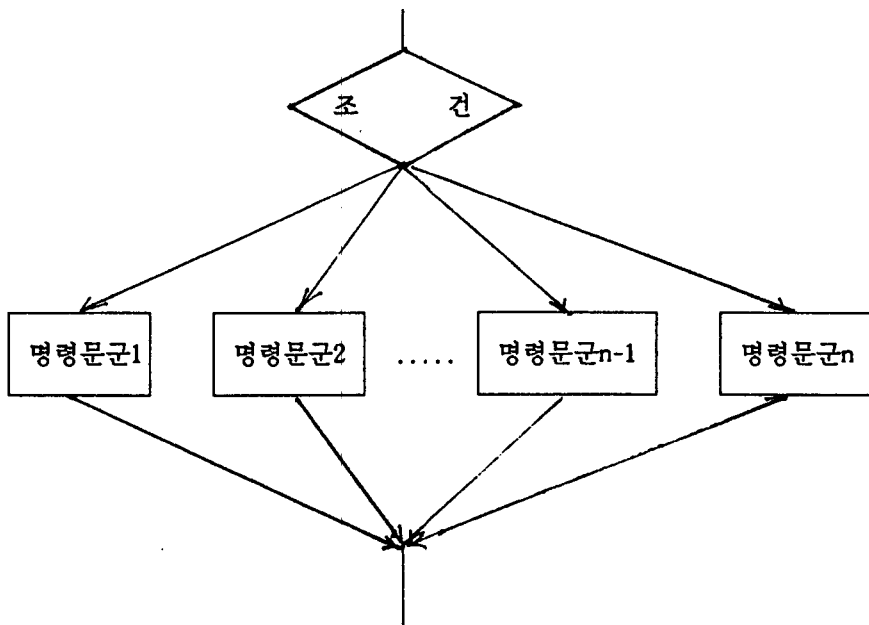
i) IF __ THEN __ ELSE 문



영문 : IF p THEN statements1
ELSE statements2

한글 : 만약 P 이면 명령문군 1
아니면 명령문군 2
만약끝

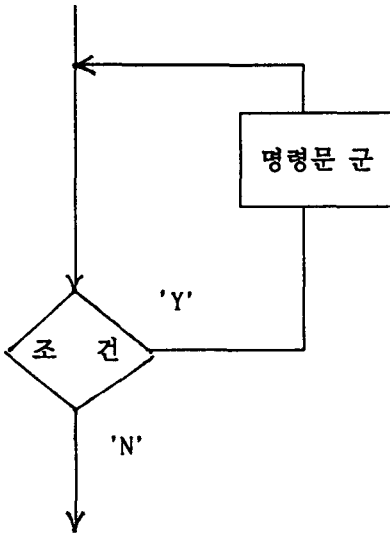
ii) CASE문



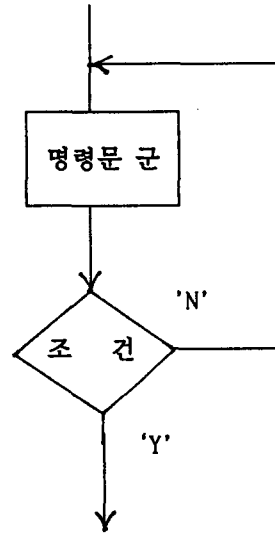
영문 :	DO	한글 :	P에 따라 택일
	CASE p = 10		10인 경우
	명령문군1		명령문군1
	CASE p = 20		20인 경우
	명령문군2		명령문군2
	CASE p = 30		30인 경우
	명령문군3		명령문군3
	.		.
	.		.
	ENDCASE		택일끝

다) 반복 구조 (Iteration building block)

주어진 조건이 만족되는 동안 명령문들을 반복해서 처리하는 구조이다.



i) while 구조



ii) until 구조

i) while 구조

영문 : while 수 < 100
수 = 수 + 1
합 = 합 + 수
endwhile

한글 : 다음을 수 < 100 동안
수 = 수 + 1
합 = 합 + 수
반복한다.

ii) until 구조

영문 : repeat

수 = 수 + 1

합 = 합 + 수

until 수 = 100

한글 : 다음을

수 = 수 + 1

합 = 합 + 수

수 = 100 일때까지 반복한다.

III. 결 론

미니 명세서의 구조적 한글 표현을 하는데에 되도록이면 전산에 대한 전문지식이 없어 programming에 익숙하지 않은 이용자들에게 어색하지 않은 표현을 만들어내는데 주력하였으나 일상언어와는 구분되는 어순과 표현으로 다소 어색함을 의사 소통 및 효율성에 감안하여 어느정도 허용할수 밖에 없었고 위에서 작성된 구조적 한글을 사용하는데도 어느정도는 학습이 필요하리라 생각된다. 그러나 이러한 연구가 소프트웨어위기를 극복하는 노력에 일조하리라 기대하고, 앞으로는 이 작성된 한글 미니 명세서를 통하여 자동적으로 program을 산출해 내는 자동화 프로그래밍에 관한 연구가 요구된다.

< 참고 문헌 >

- [1] 박재년. "구조적 분석 기법"
- [2] 박재년, 황종선, 장육배. "전산개론 및 프로그래밍" 유풍 출판사, 1984.
- [3] 이정열 편역. "MS-COBOL활용" 크라운 출판사, 1991.
- [4] 남기심, 고영근. "표준 국어 문법론" 탐출판사, 1987.
- [5] Tom de marco. "Structured analysis and System specification" Yourdon, 1978.
- [6] Tom de marco. "Structured analysis and System specification workshop" Yourdon, Edition 9, 1984.