한국통신학회 1987년도 추계학술발표회 논문집 (1987. 11. 14)

LOTOS 기술 동향 연구 분석

이 명직 홍 만표 김 동규
아주 대학교 전자 계산 학과

A STUDY ON THE STATE OF ART IN LOTOS

Myung-Jik Lee Man-Pyoo Hong Dong-Kyoo Kim
Ajou University, Department of Computer Science

ABSTRACT

In this paper, we study the precise concept of LOTOS which is one of the FDTs used in OSI. A various studying about LOTOS (ie. LOTOS itself, graphical representation for LOTOS, verification oriented LOTOS specification, and et al.) is under going. This paper is made to explain the precise concept of LOTOS, this state of art, and our concept about verification tool developing project.

1. 서 론

Open System Interconnection(OSI)에 대한 기본 Reference Model은 많은 OSI standard개발에 관한 청사진으로 여겨져 왔다. 그러나 기존의 Reference Model은 완벽하지 못하였고 TC97/SC21/WG1(Reference 모델에 관한 그룹)은 이런 Reference 모델의 개선과 개발을 계속하여왔다.

Fomal Description Technique(FDTs)의 나타남은 이런일들에 새로운 계기가 되어 ISO/TC97/SC21/WG1은 OSI standard를 성형하게 specfication할 목적으로 FDTs의 개발계획에 착수하게 되었고 그 결과로 Estelle(Extended Finite State Machine Langage)와 LOTOS(language Of Temporal Ordering Specification)가 만족 할만한 결과로써 얻어지게 되었다.

이 논문에서는 LOTOS에 중점을 두어 LOTOS가 가지는 기본 특성및 구조를 서술한뒤 현재 LOTOS에 관련된 연구과제들을 소개한다.

2. LOTOSD의 기본 특성

LOTOS의 한 system에 대한 기술은 프로세서 정의의 계층으로 구성된다. 한 system은 상호작용하는 몇개의 써브-프로세서로 구성된 한 프로세스이고 그 서부 프로세스는 다시 써브-써브 프로세스들로 구성되어진다

분산되어 있는 system 사이의 상호작용은 외부에서 관찰될 수 있는 행동만을 각 프로세스의 value선언과 variable선언 그리고 synchronize opertor 들에의해 정의 되어진다.

LOTOS에서는 Non-determinism을 허락한다. LOTOS에서 허락되는 Non-determinism은 2가지 경우에 발생한다. 그림 2-1 에서볼수 있듯이 외부에서 발생하는 Non-determinism과 임의의 내부 사건을 의미하는 사건 i 에의해 system내부에서 발생하는Nondeterminism이 존재한다

```
   a; b; stop
[] a; b; stop

- 외부 non-determinism

  i; alternative
[] i; alternative
[] i; alternative
    ...
    ...

- 내부 non-determinism
```

- Fig 2-1 -

3. LOTOS의 구조

LOTOS에의한 SPEC은 process defnition, data type 정의, behavior expression, process사이의 interaction 으로 이루어 진다. [2]

(1). Data type 정의

ACT ONE language에 기반을 두어 abstract한 data type을 정의한다. 몇 개의 기본적인 data type은 (boolean, string, natural number 등)은 standard library 에서 제공하여 기본적인 data type 정의 에서 오는 오류 를 제거 시킨다. Data type을 정의 하는 방법은 이미제공되어 있는 standard library를 이용하거나 새로운 abstract data type을 정의하여 기본 data type을 정의한후 combination, parameterization, renameing 방법으로 이미 정의 되어있는 기본data type을 이용하여 datatype을 정의한다.

그림 3-1 에서는 이런 data type 정의 방법으로 정의된 natural number를 element로 갖는 data type queue를 정의한다.

```
TYPE general_queue IS data WITH
  SORT  queue
  OPNS  create: → queue
        add   : data, queue → queue
        first : queue → data
  EQUS
        ...
        first( create) = d0
        ...
ENDTYPE

TYPE data IS
  FORMALSORT  data
  FORMALOPNS  d0: → data
ENDTYPE

TYPE nat_number_queue IS
  general_queue  ACTUALLIZED
        BY nat_numer USING
  SORTNAME nat FOR data
  OPNSNAME 0   FOR d0
ENDTYPE
```

- Fig 3-1 -

(2). Behavior expression

LOTOS에서는 외부에서 관측될 수 있는 행동을 behavior expression으로 표현한다. 이런 expression은 event가 일어날 수 있는 순간에 밀접하게 관계된다.

그림 3-2 에서는 LOTOS가 갖는 behavior expression을 소개한다.

```
inaction              stop
terminate             exit, exit( E1...EN)
choice                B1 [] B2
action prefix         g:B, g?X:t;B, g!E;B
parallel_composition  B1I[g1...gln]IB2
                      B1 III B2 (interleaving)
                      B1 II B2 ( full sync.)
hiding                hide g1...gn in B
guarding              [BE] → B
disabling             B1 [> B2
```

- Fig 3-2 -

이중 몇가지 특이한 behavior expression을 소개하면 parallel operator는 independent process의 parallelism을 표현하는것, dependent process의 paralleism을 표현하는것 그리고 이 모두를 표현하는 general parallel operator로 3 가지가 있다. Independent parallel operator는 B1과 B2를 subprocess라 하면 'B1 III B2'로 표현되어지고 B1 III B2인 behavior는 B1과 B2들중 하나만의 event가 참가할 수 있다면 그 behavior는 진행되어진다. 이에반해 dependent parallel operator는 'B1 II B2'로 표현되어지고 B1과 B2 둘다의 event가 참가할 수 있을때 behavior는 진행되어 진다. 이 모두를 합한 general parallel operator는 'B1 |[a1...an]| B2'로 표현되어지고 B1과 B2는 a1...an에 의해 synchronization된후에 behavior는 진행되어 진다.

다음으로 hiding operator는 그 위에 존재하는 프로세서에게 지금 프로세서가 synchronization해야할 요소를 숨김으로써 위의 프로세서가 자기의 환경만으로 표현가능하게 만들어 준다. Hiding operstor는 'hide a1..an in B'로 표현되어진다.

(3). Interaction

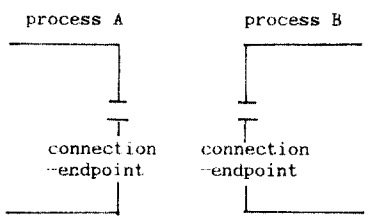
프로세서 사이의 interaction은 value declaration, variable declaration, parallel operator 를 이용하여 기술 되고하나 또는 그 이상의 똑같은 gate가 양 프로세서 사이에서 존재할때 이루어질 수있다.

그림 3-3 에서는 양 프로세스 사이에서 가능한 interaction을 기술한다.

process A	process B	sync. condition	interaction sort	effect
g!E1	g!E2	value(E1) = value(E2)	value maching	sync.
g!E	g?X:t	value(E) in domain(t)	value passing	sync. x=value(E)
g?X:t	g?Y:u	t=u	value generate	sync. x=y=v

- Fig 3-3 -

그림 3-4 에서는 connection-end point를 gate로 갖는 프로세스 사이의 interaction을 기술하고있다




```
connection_endpoint          connection_endpoint
!ident?addr:address          !ident?addr?address
?text:string                 !'test'
```

- Fig 3-4 -

4. LOTOS의 현황과 앞으로의 기대

이미 LOTOS는 distributed, concurrent information processing system에 대한 formal description으로 널리 이용되고 특별히 service definition과 OSI의 layer에 대한 protocol을 formal하게 describe할 수 있을 것으로 기대되어지고있다.

LOTOS toolset에 관한 연구는 LOTOS가 abstract한 data type 을 사용하고 있어 LISP을 이용한 toolset개발이 진행되어 완성되어있고 이 toolset을 이용하여 LOTOS에 의해 기술된 system은 event의 진행을 하나하나 선택하여 관찰되어질 수 있다. 이렇게함으로써 한 system이 LOTOS에의해 올바르게 specify될 수 있다면 그 system에 대한 verification은 event의 관찰을 통해 쉽게 이루어질 수 있다.

LOTOS graphical representation에 관한 연구는 LOTOS에 의해 specify 된 system을 보기 편하게 나타내기 위하여 여러가지 모양의 graphical model을 사용하여 표현하고 있다.

지금 우리가 가지고 있는 verification에 관한연구에 대한 개념은 현재 개발중에 있는 Temporal Logic을 이용한 verification toolset에 기반을 두어 LOTOS SPEC을 Temporal Logic expression(TLE) 변환시킨후 Temporal Logic verification toolset으로 검증하려한다.

이 밖에 많은 연구가 LOTOS에 관계 되어 진행중이다

참고 문헌

[1] ISO/TC 97/SC 21, "A tutorial on LOTOS", Draft Proposal ISO/DP, October 1986

[2] ISO/TC 97/SC 21, "Information Retrieval, Transfer and Managenent for OSI", Draft proposal ISO/DP, October 1986

[3] ISO/TC 97/SC 21, "Grapbical representation for LOTOS",Proposal for an extention of the OSI basic reference model, June 1987

[4] K.J. Turner, "An architectural semantics for LOTOS", IFIP proceeding, May 1987

[5] G.J.Leduc, "The interwining og data type and processes in LOTOS", IFIP proceeding, May 1987

[6] E.Naim, "A verification oriented specification in LOTOS of the transport protocol", IFIP proceeding, May 1987

[7] D.Gllbert, "Executable LOTOS", IFIP proceeding, May 1987

APPENDIX

< 예제 프로그램 >

* Vending machine

: 동전을 받아 choc 두개와 toffee 세개를 가지고있는 vending machine 이 임으로 한개를 선택하여 내어주는 system을 기술한다.

```
SPECIFICATION chocs[money,goods]: exit

TYPE currency IS
  SORTS money
  OPNS  coin :  -> money
ENDTYPE
TYPE goodstype IS
  SORTS goods
  OPNS  choc   : -> goods
        toffee : -> goods
ENDTYPE

BEHAVIOR
     chocs[goods]
  III(  toffee[goods]
      III(   coinslot[money]
           [> EXIT
        )
      )

II(      pay[money,goods]
   [> EXIT
  )
WHERE PROCESS chocs[goods]:EXIT :=
        goods!choc;
        goods!choc;
        EXIT
      ENDPROC

      PROCESS toffee[goods]:EXIT :=
        goods!toffee;
        goods!toffee;
        goods!toffee;
        EXIT
      ENDPROC

      PROCESS coinslot[money]:exit :=
        money!coin;
        coinslot[money]
      ENDPROC
      PROCESS pay[money,goods]:NOEXIT :=
        money?m:money;
        goods?g:goods;
        pay[money,goods]
      ENDPROC
ENDSPEC
```