

교환시스템의 서비스 품질수준 향상을 위한 소프트웨어 블록의 중요도 결정

조 재 균*

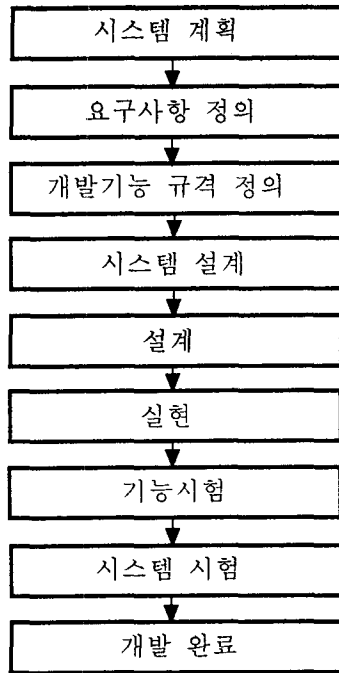
<목 차>

I. 서 론	IV. 결정지수(criticality index)의 평가 방법
II. 서비스 품질수준의 규정 및 평 가방법	V. 결 론
III. PERT 모델링 방법	참고문헌
	Abstract

I. 서 론

교환시스템의 개발은 <그림 1>에 보인 것과 같은 개발단계에 따라 이루어진다 (한국전자통신연구소, 1994a). 시스템 계획 단계는 사용자의 요구사항(user requirement)을 분석하여 시스템 개발의 전반적인 계획을 수립하는 단계이며, 요구사항 정의 단계에서는 사용자 요구사항에서 제시된 불분명한 요구사항을 개발자가 사용자 관점에서 명확히 정리하여 개발 도중 발생 가능한 사용자와 개발자간 개발정보의 불확실성과 불완전성을 초기에 최소화시키고 문제점을 조기에 발견하여 해결하는 단계로서, 시스템이 갖추어야 할 기능과 서비스 품질수준, 시스템 용량, 트래픽 처리용량, 신뢰도 등의 요구사항을 정의한다. 개발기능 규격 정의 단계는 시스템에서 실현할 개발기능(function)들을 요구사항 정의로부터 추출하여 정의하는 단계이며, 시스템 설계 단계에서는 시스템의 전체적인 구조를 정의하고, 개발기능을 실질적인 구현 단위인 하드웨어 및 소프트웨어 블록(block)으로 변환시키는 업무를 수행하게 되며, 블록 간 메시지 정의 및 순서도(sequence chart) 작성 등도 이 단계에서 수행된다. 설계 단계에서는 각 블록에 대한 설계 작업을 수행하는 단계로, 소프트웨어 구현 단위인 소스 프로그램 파일(source program file) 및 하드웨어 구현 단위인 인쇄회로판(PBA, Printed circuit Board Assembly) 정의, 블록 상호작용 다이어그램(BID, Block Interaction Diagram) 작성, 블록별 프로세스(process) 정의, SDL(functional

* 동의대학교 경영정보학과

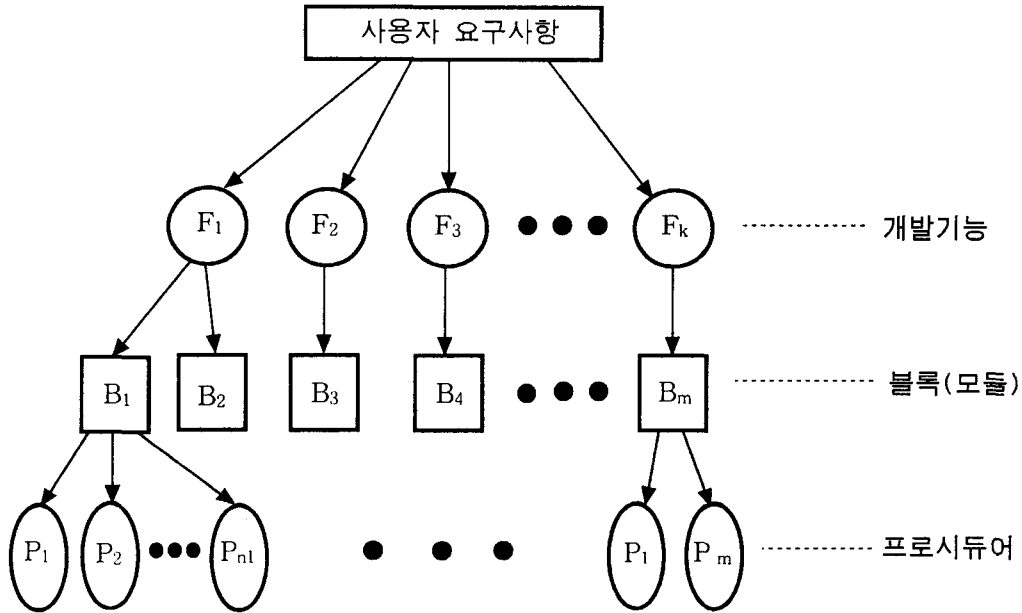


<그림 1> 교환시스템 개발단계

Specification and Description Language) 블록 다이어그램 작성 등의 업무가 수행된다.

설계 단계에서 작성된 설계 문서에 따라 소프트웨어를 구체적으로 구현하는 실현 단계를 거쳐 코딩이 완료된 소프트웨어는 시험 단계에서 검증을 받게 된다. 시험은 모듈 (즉, 블록) 내부의 오류를 찾기 위한 단위시험(unit testing), 독립적으로 검증한 모듈들을 연동하여 개발기능을 검증하기 위한 통합시험(integration testing), 그리고 통합시험에서 검증된 소프트웨어를 실제의 운용 조건하에서 시스템의 모든 기능들을 연동시켜서 시험하고, 한계 조건하에서 시스템이 정상적으로 동작하는가를 검증하기 위한 시스템 시험(system testing)으로 구분되어 수행된다. 시스템 시험에는 사용자가 기대하는 서비스 품질을 제공하기 위해서 시스템이 호처리 용량, 서비스 품질수준(QOS, quality of service) 등의 성능 설계 목표치(performance design objectives)를 만족하는지를 확인하는 성능시험(performance testing)이 포함된다. 성능시험의 수행 결과 목표치를 만족하지 못할 경우에는 성능을 저하시키는 요인이 무엇인가를 파악하고 문제점을 해결함으로써 성능을 향상시키는 노력을 기울이게 된다.

교환시스템의 소프트웨어 구조는 효율적인 개발, 시험 및 유지보수를 위하여 <그림 2>에 보인 것과 같은 계층적 구조로 설계되어진다. 예를 들어, 최상위 레벨의 사용자 요구사항은 중간 레벨의 개발기능들로 분할되고, 각각의 개발기능은 그 하위 레벨인 블록들로, 그리고 각 블록은 다시 그 하위 레벨인 프로시듀어(procedure)들로 분할된다.



<그림 2> 소프트웨어의 계층적 구조

교환시스템의 서비스 품질수준 항목 중 지연시간을 평가하는 방법에 대해서는 많은 연구가 되어 있다(여환근 외 3인, 1998; 윤복식 외 2인, 1995; 이일우 외 2인, 1996; Shuichi Sumita, 1989).

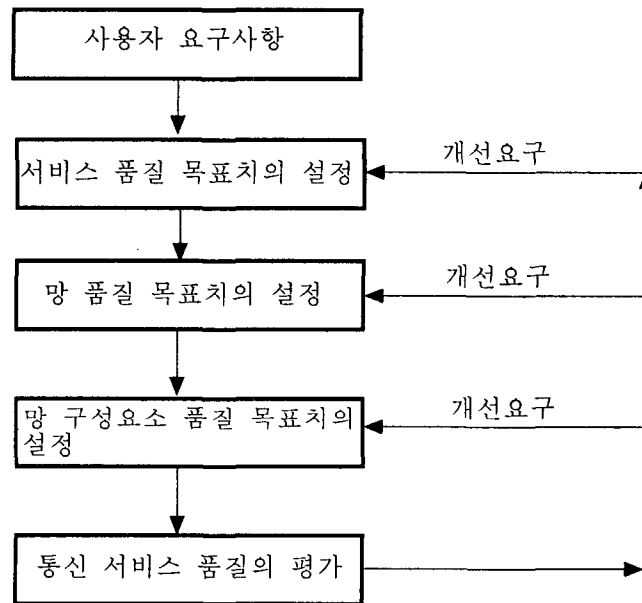
본 논문에서는 지연시간을 평가한 결과 설계 목표치를 만족하지 못할 경우, 소프트웨어의 개선을 통한 서비스 품질수준 향상을 위해서 메시지와 블록의 중요도를 결정하는 방법을 제시한다. 제안된 방법은 설계 단계에서 작성되는 도식적 설계 표현 기법 중의 하나인 순서도를 확률적 PERT(Program Evaluation and Review Technique) 네트워크로 모델링한 후, 몬테칼로 시뮬레이션을 이용하여 중요도 척도(importance measure)를 평가함으로써 서비스 품질수준에 중요한 영향을 미치는 메시지와 블록이 무엇인지를 밝힌다. 이때, 메시지와 블록의 중요도 척도로서 결정지수(criticality index)를 사용한다. II절에서는 교환시스템에서의 서비스 품질수준 규정 및 평가 방법에 대하여 살펴보고, III절에서는 순서도를 확률적 PERT 네트워크로 모델링하는 방법을 제시한다. 그리고 IV절에서는 메시지와 블록의 중요도 척도로 사용되는 결정지수를 평가하는 방법을 설명한다.

II. 서비스 품질수준의 규정 및 평가 방법

이 절에서는 교환시스템의 서비스 품질수준 규정 방법과 평가 방법에 대하여 살

펴본다.

먼저, 통신 서비스의 품질 설계 순서를 살펴보면 <그림 3>에 나타낸 바와 같이 (淺谷耕一, 1993), 설문조사나 주관적 품질평가 방법 등을 통하여 파악된 사용자가 기대하는 통신 서비스 품질을 바탕으로 서비스 품질의 목표치를 설정하고, 망 품질 (network performance) 목표치는 서비스 품질 목표치를 달성할 수 있도록 사용자간 통신을 제공하는 망에 대해 설정된다. 망 품질 목표치는 망 품질에 영향을 미치는 망 구성요소의 품질(network element performance) 목표치로 배분된다. 그리고 통신 서비스 품질의 평가 결과는 해당 품질 설계 단계로 피드백되어 품질 향상을 위한 설계 자료로 활용된다.



<그림 3> 통신 서비스의 품질설계 순서

망 구성요소인 교환시스템은 가입자의 호(call) 연결 요구를 호손실(blocking)과 지연시간(delay time)의 관점에서 만족시켜주는 것이 필요하다. 교환기가 하나의 호를 연결하기 까지는 많은 자원 (예: 스위칭 네트워크, 신호채널 및 각종 서비스 장치들)을 필요로 하며, 가입자들이 이들 공통 자원들을 경쟁적으로 사용할 경우에 자원 부족으로 인해 호연결을 해주지 못하는 호손실이 발생하게 된다. 그리고 여러 프로세서들로 구성되는 제어계에 의해 호를 연결하는데 지연시간이 발생하게 된다. 따라서 사용자를 만족시키는 서비스 품질을 제공하기 위해서는 호손실 확률과 지연시간이 일정한 값 이하가 되는 것이 바람직하다.

ITU(International Telecommunication Union, 국제전기통신연합)에서는 교환시스템의 호손실 확률과 지연시간들에 대한 설계 목표치들을 Q.543에서 권고하고 있다 (ITU-T, 1993). Q.543에서 권고하고 있는 지연시간 목표치 항목으로는 호요구 지연시

간(local exchange call request delay), 호설정 지연시간(exchange call set-up delay), 통화로 접속 지연시간(through-connection delay), 호착신 지연시간(incoming call indication sending delay), 호출음 송출 지연시간(alerting sending delay) 및 호해제 지연시간(exchange call release delay) 등이 있으며, <표 1>은 호요구 지연시간 목표치를 나타낸다. 아날로그 가입자 회선에 대한 호요구 지연시간(발신음 지연시간이라고도 함)은 교환기가 가입자 회선 정합장치에서 후크 오프(hook off) 상태를 감지한 순간부터 다이얼 신호음을 가입자 회선에 공급하는 순간까지의 지연시간으로 정의되며, 디지털 가입자 회선에 대한 호요구 지연시간은 착신가입자 번호의 개개 디지털(digit)가 별도의 메시지에 담겨 송신되는 오우버랩(overlap) 방식의 경우에는 가입자 신호시스템으로부터 SETUP 메시지를 수신한 순간부터 SETUP ACKNOWLEDGE 메시지를 반송할 때까지의 지연시간으로 정의되며, 착신가입자 번호가 한 개의 메시지에 담겨 한꺼번에 송신되는 엔블록(en-bloc) 방식의 경우에는 가입자 신호시스템으로부터 SETUP 메시지를 수신한 순간부터 CALL PROCEEDING 메시지를 반송할 때까지의 지연시간으로 정의된다. 기준부하는 교환기의 성능 목표치들을 설정하기 위한 기준이 되는 부하를 말하며, 기준부하 A는 교환기가 최번시에 가입자 회선이나 중계선에 제공하고자 하는 평균 트래픽량을 나타내며, 기준부하 B는 계획된 정상 수준을 초과하는 트래픽량을 나타내는데 교환기의 용량은 기준부하 B를 만족시킬 수 있도록 설계된다. 예를 들면, 아날로그 가입자 회선에 대해서 기준부하 B하에서 호요구 지연시간 설계 목표치는 평균이 800ms이고, 1000ms를 초과하지 않을 확률이 95%가 되어야 함을 의미한다.

<표 1> 호요구 지연시간 목표치 (단위: ms)

구분	기준부하 A		기준부하 B	
	평균	95% 값	평균	95% 값
에널로그 가입자 회선	400	600	800	1000
디지털 가입자 회선	overlap 방식	400	800	1000
	en-bloc 방식	600	800	900

서비스 기준 지연시간의 평가 방법으로는 해석적인 방법, 시뮬레이션에 의한 방법 그리고 실측에 의한 방법이 사용된다. 해석적인 방법과 시뮬레이션 방법은 전형적으로 시스템 설계 단계에서 수행되며, 시스템의 규모, 처리 용량 및 서비스 품질수준 등을 평가함으로써 요구사항 정의 단계에서 설정된 사항들을 충족하는 시스템의 전체적인 구조를 정립할 목적으로 사용된다. 반면에, 실측은 시스템 시험 단계에서 수행 가능하며, 시스템의 처리 용량 및 서비스 품질수준 등이 목표치를 충족하는지의 여부를 확인할 목적으로 사용된다.

예를 들어, 한국전자통신연구원에서 수행한 TDX-10 교환시스템의 서비스 기준 지연시간 평가 방법 및 그 결과를 간략히 살펴보면 다음과 같다(한국전자통신연구소, 1994b). 해석적인 방법과 시뮬레이션에 의한 방법은 다중 프로세서 분산제어 방식을

채택하고 있는 TDX-10 시스템의 제어계를 프로세서들과 프로세서 상호간 메시지 교환을 위한 IPC(Inter Processor Communication) 네트워크를 서버(server)로 하는 큐잉 네트워크(queueing network)로 모델링한다. 큐잉 네트워크를 분석하기 위해서, 해석적인 방법에서는 QNA(Queueing Network Analyzer)(Whitt, 1983)와 같은 큐잉 네트워크 분석용 소프트웨어 패키지가 사용되며, 시뮬레이션 방법에서는 SLAM II와 같은 시뮬레이션 언어가 사용된다. 실측에 의한 방법은, 트래픽 발생장치인 LCS(Local Call Simulator)와 UCS(Universal Call Simulator) 및 가상 트래픽을 발생시키는 소프트웨어인 부하 박스(load box)를 사용하여 시스템의 제어계를 구성하는 각 프로세서에 트래픽을 발생시킨다. 해석적인 방법과 시뮬레이션 방법의 분석 결과 또는 실측에 의한 방법의 측정 결과, 각 프로세서별로 트래픽 부하에 따른 CPU 점유율 및 서비스 기준 지연시간값들이 얻어진다. 이때, 각 프로세서의 호처리 용량은 프로세서 CPU 점유율이 90% 이하이면서 지연시간값들이 ITU-T Q.543에서 권고하고 있는 목표치들을 만족하는 트래픽 부하의 최대값이 된다. 그리고 프로세서들의 호처리 용량들 중에서 최소값을 시스템의 호처리 용량으로 삼는다. <표 2>는 음성호의 대표적인 지연시간 항목인 호요구 지연시간 즉, 발신음 지연시간(dial tone delay)에 대해서, 해석적인 방법과 시뮬레이션 방법으로 추정된 트래픽 부하 증가에 따른 발신음 지연시간값들을 나타낸다. 결과적으로, TDX-10 교환시스템의 제어계 호처리 용량은 150만호로 추정되며, 그때의 발신음 지연시간값들 (해석적인 방법에 의한 226.2ms, 시뮬레이션 방법에 의한 221.8ms)은 <표 1>에 나타나있는 목표치 800ms를 만족한다. 이때 물론 호요구 지연시간 외의 다른 지연시간 추정값들도 그들의 목표치를 만족한다.

<표 2> TDX-10 교환시스템의 발신음 지연시간 추정값

트래픽 부하	발신음 지연시간 (단위:ms)	
	해석적 방법	시뮬레이션 방법
90만	223.3	220.7
120만	224.7	221.2
150만	226.2	221.8

위의 결과는 150만호 이상의 트래픽이 교환기에 추가되면 지연시간 항목 중 적어도 한개 이상의 지연시간값이 목표치를 초과하게 됨을 의미한다. 즉, 교환시스템의 호처리 용량을 증가시키고자 하면 지연시간값들이 증가하게 되고, 지연시간값의 증가는 호처리 용량의 증가를 제한하는 결과를 가져온다.

개발단계의 시스템 설계 단계에서는 해석적인 방법이나 시뮬레이션 방법 등을 사용하여 요구사항 정의 단계에서 설정된 호처리 용량이나 서비스 품질수준 목표치 등을 만족할 수 있도록 시스템의 전반적인 구조를 설계한다. 그러나, 해석적인 방법이나 시뮬레이션 방법을 사용할때의 가정사항들, 고려하지 못한 시스템 특성 및 부정확한 입력 데이터값들로 인하여 호처리 용량이나 지연시간의 추정값들은 부정확할 소지가

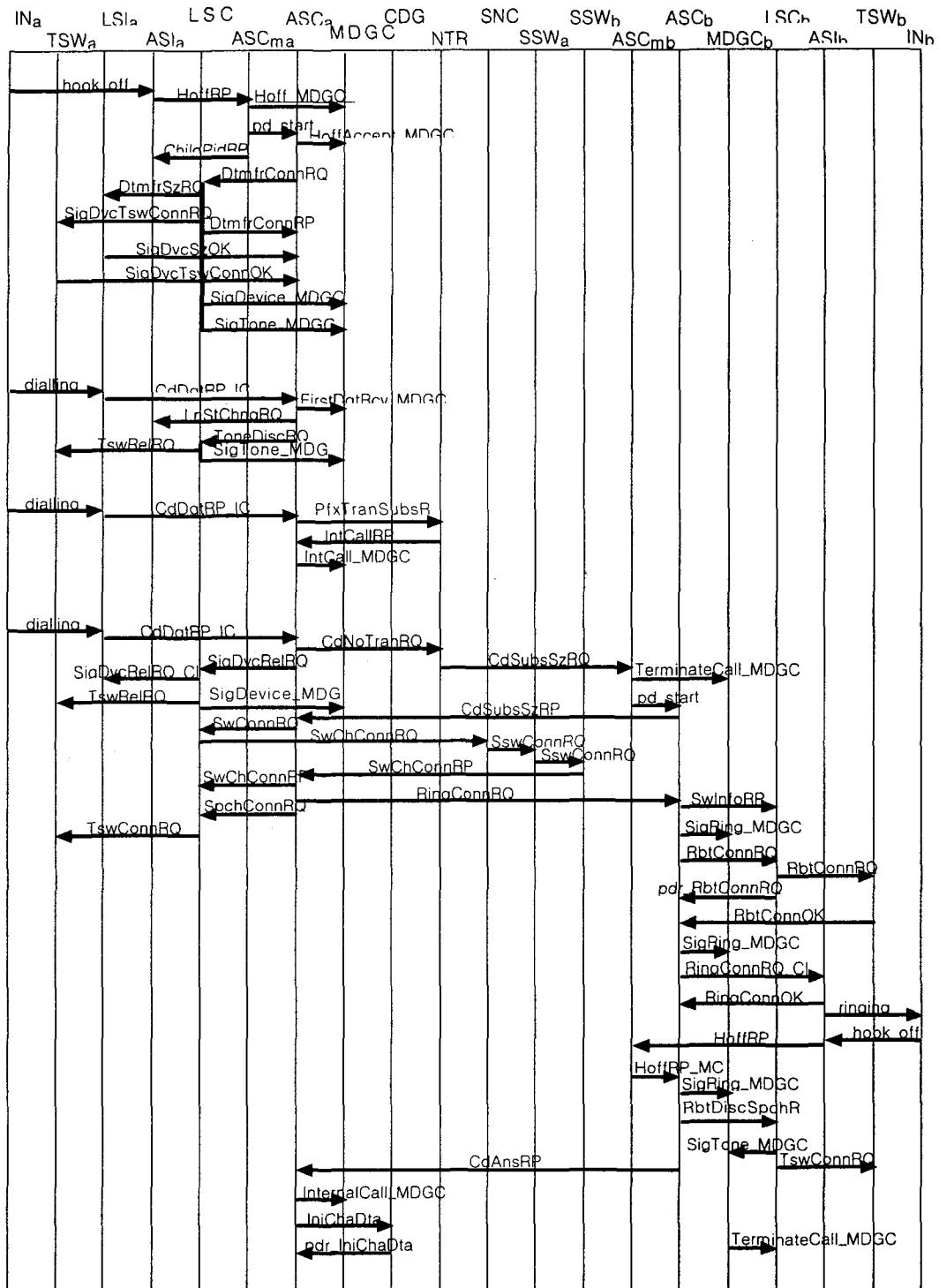
많다. 만약, 실측에 의해 측정된 호처리 용량이나 지연시간값들이 목표치를 만족하지 못하는 경우를 생각해보자. 이러한 상황은 시스템의 성능 개선 노력이 있기 전에는 흔히 발생 가능한 일이다. 다시 말하면, 시스템 시험 단계의 성능시험 결과 요구사항 정의 단계에서 설정된 성능 설계 목표치를 단번에 만족하는 것은 오히려 쉽지 않다.

시스템의 성능 개선을 위해서는 시스템 하드웨어 및 소프트웨어의 수정이 가능하다. 그러나, 제어계 구조 변경 및 프로세서 용량 증대 등과 같은 하드웨어 수정은 개발단계에서 볼 때 시스템 설계 단계로의 피드백(feedback)을 뜻하며, 시스템 전체에 대한 파급효과가 크기 때문에 개발기간과 비용면에서 현실적으로 어려움이 많다. 반면에, 프로세서에 적재되는 응용 프로그램(예: 블록들)을 수정하는 것은 실현 단계로 피드백하는 것으로 하드웨어에 비해 수정도 훨씬 용이하며 개발기간과 비용도 상당히 줄일 수 있어 성능 향상을 위한 합리적인 대안이 될 수 있다. 하드웨어의 수정이 시스템의 재설계를 통한 근본적인 성능 개선의 성격을 띠는데 반해, 소프트웨어의 수정은 성능 개선을 위한 보완적인 성격을 띠므로 시스템 성능의 대폭적인 향상을 기대하기는 어려운 점이 있다.

III. PERT 모델링 방법

이 절에서는 개발기능 순서도로부터 분석하고자 하는 서비스 품질수준 항목과 관련된 부분을 추출한 후, 그것을 확률적 PERT 네트워크로 모델링하는 방법을 제시한다.

순서도는 개발기능별로 작성되며, 블록간에 어떠한 메시지를 주고 받음으로써 그 개발기능이 구현되는지를 시간 개념을 도입해서 나타낸 그림이다. <그림 4>는 TDX-10 교환시스템의 개발기능들 중의 하나인 '자국호처리' 기능의 순서도 일부이다(한국전자통신연구소, 1994b).

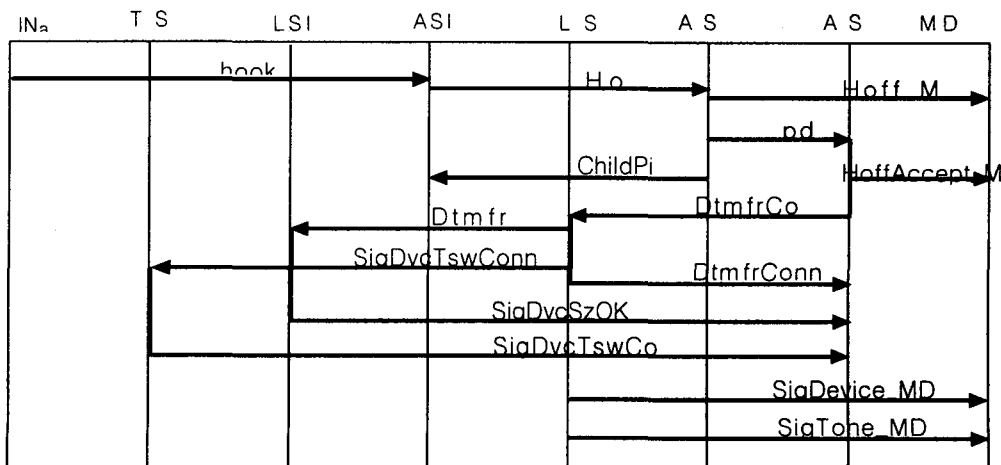


<그림 4> TDX-10 자국호처리 기능 순서도

<그림 4>에서 발신 단말을 나타내는 IN_a 와 IN_b 를 제외한 나머지 TSW_a , LSI_a , ..., ASI_b , TSW_b 등은 자국호처리 기능의 구현을 위해 개발되어지는 블록들이며, 블록이름 뒤의 a와 b는 각각 발신가입자 및 착신가입자가 관련된 블록임을 나타낸다. LSC_a , LSC_b , ASC_{ma} , ASC_a , ASC_b , $MDGC_a$, $MDGC_b$, CDG 블록들은 프로세서 ASP에 적재되며, NTR 블록은 프로세서 NTP에, SNC 블록은 프로세서 INP에, TSW_a 와 TSW_b 블록들은 프로세서 TSP에, LSI_a 와 LSI_b 블록들은 프로세서 LSP에, ASI_a 와 ASI_b 블록들은 프로세서 ASIP에, SSW_a 와 SSW_b 블록들은 프로세서 SSP에 적재된다. 블록간에 주고 받는 메시지는 블록에서 블록으로의 화살표로 나타내며, 메시지의 이름은 화살표 위에 기입된다. 그리고 순서도에서 먼저 나타나는 메시지가 시간적으로 앞서게 된다. 예를 들어, 메시지 HoffRP가 ASI_a 블록에서 ASC_{ma} 블록으로 송신된 후에, 메시지 Hoff_MDGC가 ASC_{ma} 블록에서 $MDGC_a$ 블록으로 송신된다.

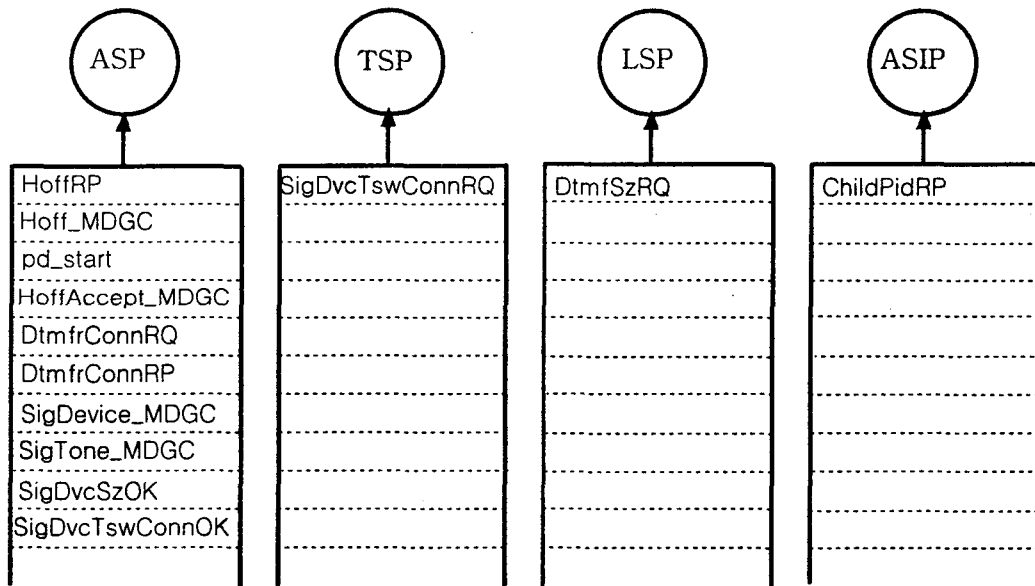
각 블록에서는 메시지를 수신하게 되면 메시지의 처리를 위해서 일련의 프로시유어들을 수행하게 된다. 이때, 각 메시지의 처리시간은 일정하지 않고 불확정적(uncertain)이다. 왜냐하면, 해당 메시지를 처리하는 프로세서의 부하(load) 정도가 시간에 따라 변동될 뿐만 아니라, 메시지의 처리를 위해 수행되는 프로시유어들에서의 제어 흐름(control flow) 및 데이터 탐색 처리 등이 메시지의 처리시마다 상이해지기 때문이다. 따라서, 메시지 처리시간은 확률변수로 나타낼 수 있으며, 메시지 처리시간 분포는 개발단계의 기능시험 단계에서 실측을 통하여 추정 가능하다.

서비스 품질수준 항목 중 발신음 지연시간의 경우를 예로 들어 PERT 네트워크로 모델링하는 방법을 설명하면 다음과 같다. II절에서 논의된 바와 같이, 아날로그 가입자 회선에 대한 발신음 지연시간은 교환기가 가입자 회선 정합장치에서 후크 오프 상태를 감지한 순간부터 다이얼 신호음을 가입자 회선에 공급하는 순간까지의 지연시간으로 정의되므로, <그림 4>에서 아날로그 가입자 회선에 대한 발신음 지연시간과 관련된 부분만을 추출하면 <그림 5>와 같다.



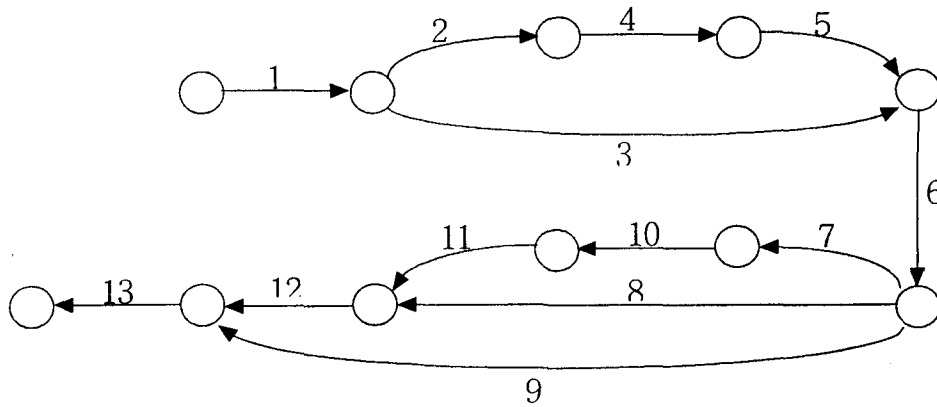
<그림 5> 발신음 지연시간 순서도

<그림 5>에 있는 순서도를 PERT 네트워크로 모델링하기 위해서는 블록들의 메시지 처리 선후행 관계를 분석해야 한다. <그림 5>에 있는 메시지들이 처리되는 프로세서 및 프로세서에서의 처리 순서를 나타내면 <그림 6>과 같다. 각 프로세서에서는 큐(queue)에 먼저 입력되는 메시지가 먼저 처리된다. 예를 들면, ASP 프로세서에서는 HoffRP 메시지를 처리한 후 Hoff_MDGC, pd_start, HoffAccept_MDGC 메시지 등을 순서대로 처리하게 된다. 한편 ASIP 프로세서에서 처리되는 ChildPidRP 메시지는 ASP 프로세서에서 HoffAccept_MDGC 메시지가 처리 완료되기 전까지 처리를 끝내면 된다. 그리고 ASP 프로세서에서의 HoffAccept_MDGC 메시지 처리 및 ASIP 프로세서에서의 ChildPidRP 메시지 처리가 종료된 후, ASP 프로세서에서는 DtmfrConnRQ 메시지를 처리하게 된다.



<그림 6> 프로세서에서 처리되는 메시지 종류 및 처리 순서

<그림 7>은 발신음 지연시간의 PERT 네트워크를 나타낸다. <그림 7>의 PERT 네트워크에서 화살표로 표시된 '활동' (activity), 활동을 수행하는데 걸리는 시간인 '활동시간' (activity duration) 그리고 첫번째 노드(node)에서 마지막 노드까지의 수행시간인 '사업완성시간' (project completion time)은 각각 메시지, 메시지 처리시간 및 발신음 지연시간과 대응된다.



- | | | |
|---------------------|---------------------|----------------------|
| 1 : HoffRP | 6 : DtmfrConnRQ | 11 : SigTone_MDGC |
| 2 : Hoff_MDGC | 7 : DtmfrConnRP | 12 : SigDvcSzOK |
| 3 : ChildPidRP | 8 : DtmfrSzRQ | 13 : SigDvcTswConnOK |
| 4 : pd_start | 9 : SigDvcTswConnRQ | |
| 5 : HoffAccept_MDGC | 10 : SigDevice_MDGC | |

<그림 7> 발신음 지연시간의 PERT 네트워크

다른 서비스 품질수준 항목 (호설정 지연시간, 통화로 접속 지연시간, 호착신 지연시간, 호출음 송출 지연시간 및 호해제 지연시간 등)에 대해서도 PERT 네트워크로 모델링하고자 하는 지연시간과 관련된 부분을 <그림 4>에서 추출한 후 발신음 지연시간의 경우와 유사하게 메시지 처리의 선후행 관계를 분석함으로써 PERT 네트워크로 모델링 가능하다.

IV. 결정지수(criticality index)의 평가 방법

PERT 네트워크 분석의 주된 목적 중의 하나는 프로젝트 완성에 중요한 역할을 하는 활동들을 밝히는 것이다. 활동의 중요도를 측정하기 위한 척도인 활동의 결정지수(criticality index of an activity)는 그 활동이 주경로(critical path)상에 있을 확률로 정의되며, 사업완성기간의 평균값을 감소시키기 위하여 어떤 활동의 평균 시간을 감소시켜야 할지를 밝히기 위하여 사용된다. 여기서 주경로란 그 경로상에 있는 활동들의 활동시간이 증가하게 되면 사업완성시간의 증가를 초래하게 되는 그러한 경로를 말한다.

결정지수에 관해서는 많은 연구가 되어 왔으며(Bowman, 1995; Dodin and Elmaghraby, 1985; Elmaghraby, 1977; Van Slyke, 1963), 결정지수는 해석적인 방법

과 시뮬레이션 방법에 의해 평가될 수 있다. 본 논문에서는 Van Slyke(1963)에 의해 제시된 몬테칼로 시뮬레이션(Monte Carlo simulation) 방법에 의한 결정지수의 평가 방법을 이용한다. 몬테칼로 시뮬레이션을 이용하여 활동의 결정지수를 평가하기 위해서는 먼저, PERT 네트워크를 구성하는 각 활동들의 활동시간 분포로부터 확률 표본값(random sample)들을 추출한다. 그 다음 CPM(critical path method)과 같은 '가장 긴 경로 알고리즘'(longest path algorithm)을 이용하여 사업완성시간이 가장 긴 경로 즉, 주경로를 찾는다. 그리고 주경로상에 있는 활동들을 기록해 둔다. 위의 과정을 시뮬레이션 실행횟수 동안 반복한다. 이때, 시뮬레이션 실행횟수 중에서 어떤 활동이 주경로상에 있을 횟수즉, 비율을 그 활동의 결정지수로 한다. 예를 들어, 시뮬레이션 실행횟수 3,000회 중에서 어떤 활동이 2,700회 주경로상에 있었다면 그 활동의 결정지수는 0.9가 된다. 결정지수가 1에 가까울수록 그 활동의 활동시간 증가(감소)는 사업완성시간의 증가(감소)에 크게 영향을 미치며, 0에 가까울수록 그 활동의 활동시간 증가(감소)는 사업완성시간의 증가(감소)에 영향이 작다. 따라서 사업완성시간을 줄이기 위해서는 결정지수가 큰 활동들의 활동시간을 줄이는 것이 효과적임을 의미한다.

<그림 7>에 있는 PERT 네트워크에 대해서 활동시간 (즉, 메시지 처리시간) 분포가 정규분포를 따른다고 가정하고, 또한 각 활동시간의 평균과 분산이 <표 3>과 같다고 가정했을때, 몬테칼로 시뮬레이션 방법에 의해 계산된 활동 (즉, 메시지)의 결정지수는 <표 4>와 같다. <표 4>에서 결정지수가 큰 순서대로 메시지의 처리시간을 줄이는 것이 발신음 지연시간을 감소시키는데 효과적일 것이다. 메시지의 처리시간을 줄이기 위해서는 그 메시지를 처리하는 일련의 프로시듀어들을 따라가면서 데이터 처리 알고리즘, 제어흐름 등의 재설계 및 수정이 요구된다.

<표 3> 활동시간 분포 (시간단위: ms)

활동	활동시간 분포	활동	활동시간 분포
1	(3.53, 1.0) †	8	(4.50, 3.5)
2	(2.16, 0.8)	9	(5.20, 4.8)
3	(0.68, 0.1)	10	(0.74, 0.1)
4	(1.78, 0.8)	11	(1.06, 0.2)
5	(0.74, 0.1)	12	(0.67, 0.1)
6	(1.89, 0.9)	13	(0.67, 0.1)
7	(0.74, 0.1)		

†: (평균, 분산)

<표 4> 활동의 결정지수

활동 i	$CA(i)$	활동 i	$CA(i)$
1	1.000	8	0.231
2	0.998	9	0.002
3	0.002	10	0.767
4	0.998	11	0.767
5	0.998	12	0.998
6	1.000	13	1.000
7	0.767		

위에서 계산된 메시지의 결정지수로부터 발신음 지연시간에 각 블록이 차지하는 중요도는 <표 5>와 같이 계산할 수 있다. 이때 블록의 중요도는 관련된 메시지 결정지수들을 합하여 구한다. 예를 들어, 블록 ASC_a 의 결정지수 3.763은 ASC_a 에서 처리되는 메시지들 pd_start , $DtmfrConnRP$, $SigDvcSzOK$, $SigDvcTswConnOK$ (<그림 5> 참조)의 결정지수 0.998, 0.767, 0.998 및 1.000을 합한 값이다.

<표 5> 블록의 중요도

블록	$CB(i)$
TSW_a	0.002
LSI_a	0.231
ASI_a	0.002
LSC_a	1.000
ASC_{ma}	1.000
ASC_a	3.763
$MDGC_a$	3.530

V. 결론

교환시스템은 사용자에게 만족스러운 서비스 품질을 제공하기 위해서 호처리 용량, 서비스 품질수준(QOS, quality of service) 등의 성능 설계 목표치(performance design objectives)를 만족하도록 개발되어진다. 시스템 개발단계의 요구사항 정의 단계에서 성능 설계 목표치가 정의되며, 시스템 설계 단계에서는 해석적인 방법 및 시뮬레이션 방법을 이용하여 호처리 용량 및 서비스 기준 등에 대한 성능분석을 수행하고, 분석 결과를 시스템의 전체적인 구조 설계에 반영함으로써, 성능 설계 목표치를 만족하도록 시스템이 설계된다.

그러나 시스템 설계 단계에서의 성능분석 결과는 성능분석시의 가정사항들, 고려하지 못한 시스템 특성 및 입력 데이터값들의 불확실성 등으로 인해 부정확할 가능성

이 크다. 만약 시스템 시험 단계의 실측 결과 성능 설계 목표치를 만족하지 못할 경우에는 성능을 저하시키는 요인이 무엇인가를 파악하고 문제점을 해결함으로써 성능을 향상시키는 노력이 필요하게 된다.

본 논문은 교환시스템의 서비스 품질수준 항목 중 지연시간이 설계 목표치를 만족하지 못할 경우, 소프트웨어의 개선을 통한 서비스 품질수준 향상을 위해서 메시지와 블록의 중요도를 결정하는 방법을 제시한다. 제안된 방법은 설계 단계에서 작성된 순서도에서 분석하고자 하는 서비스 품질수준 항목과 관련된 부분을 추출하여 확률적 PERT 네트워크로 모델링한 후, 몬테칼로 시뮬레이션을 이용하여 결정지수(criticality index)를 평가함으로써, 서비스 품질수준을 향상시키기 위하여 어떤 메시지의 처리시간을 줄이는 것이 효과적인가를 평가한다. 그리고 각 블록들이 해당 서비스 품질수준에 대해 차지하는 중요도는 어느 정도 되는가를 평가한다.

시스템의 성능 개선을 위해서 하드웨어를 수정하는 것은 시스템의 재설계를 통해 근본적인 성능 개선은 가능하나, 시스템 전체에 대한 파급효과가 크기 때문에 개발기간과 비용면에서 현실적으로 어려움이 많다. 반면에, 제안된 방법과 같은 응용 프로그램의 수정을 통한 성능 개선은 하드웨어에 비해 수정도 훨씬 용이하며 개발기간과 비용도 상당히 줄일 수 있으나, 하드웨어 수정에 비해 성능 개선을 위한 보완적인 성격을 띠므로 성능의 대폭적인 향상을 기대하기는 어려운 점이 있다. 따라서, 시스템 시험 단계의 실측 결과가 성능 설계 목표치를 크게 만족하지 못할 경우에는 하드웨어의 수정을 통한 근본적인 성능 개선이 요구되지만, 실측 결과와 성능 설계 목표치와의 차이가 크지 않을 경우에는 제안된 방법의 적용은 서비스 품질수준 향상을 위한 합리적인 대안이 될 수 있다.

참고 문헌

- [1] 여환근, 송광석, 노승환, 기장근, “분산 ATM 교환 제어시스템에서 프로세서간 통신 정합부에 대한 성능 분석,” 전자공학회논문지, 35권 S편 6호, pp.25-35, 1998.6.
- [2] 윤복식, 백장현, 이창훈, “교환기의 지연시간 분포 분석방법,” 한국통신학회논문지, 20권 6호, pp.198-211, 1995.6.
- [3] 이일우, 조기성, 임석구, “시뮬레이션에 의한 이동통신 교환기 제어계의 성능 분석,” 한국통신학회논문지, 21권 10호, pp.2605-2619, 1996.10.
- [4] 한국전자통신연구소, *TDX-10 개발지침서(I)*, TDX-10 총서 제8권, 1994a.
- [5] 한국전자통신연구소, *TDX-10 성능평가*, TDX-10 총서 제11권, 1994b.
- [6] 浅谷耕一, 通信 ネットワークの 品質設計, 電子情報通信學會, 1993.
- [7] Bowman, R. A., “Efficient estimation of arc criticalities in stochastic activity networks,” *Management Science*, Vol. 41, No. 1, pp.58-67, 1995.
- [8] Dodin, B. M., and Elmaghraby, S. E., “Approximating the criticality indices of the activities in PERT networks,” *Management Science*, Vol. 31, pp.207-223, 1985.
- [9] Elmaghraby, S. E., *Activity Networks: Project Planning and Control by Network Models*, Wiley, New York, 1977.
- [10] ITU-T Recommendation Q.543, “Digital exchange performance design objectives,” 1993.
- [11] Shuichi Sumita, “Performance analysis of interprocessor communications in an electronic switching system with distributed control,” *Performance Evaluation*, Vol. 9, pp.83-91, 1989.
- [12] Van Slyke, R. M., “Monte Carlo methods and the PERT problems,” *Operations Research*, Vol. 11, pp.839-860, 1963.
- [13] Whitt, W., “The queueing network analyzer,” *Bell System Technical Journal*, Vol. 62, No. 9, pp.2779-2815, 1983.

<Abstract>

**Determination of Importance of Software Blocks
for Improving Quality of Service in Switching System**

Jae-Gyeun Cho

The switching system is designed and developed to satisfy the performance design objectives recommended by ITU for call processing capacity and quality of service(QOS), etc. When the results by actual measurement at the system testing phase do not satisfy the performance design objectives, however, an effort is required to improve the performance.

This paper presents a method for improving QOS by modifying the application programs for the switching system. In the proposed method, the sequence chart related to a *delay time* for call connection is modelled using PERT(Program Evaluation and Review Technique) network. Then, the criticality index of a message is calculated using Monte Carlo simulation to evaluate which message's processing time to decrease in order to decrease the delay time and thus to improve QOS. The criticality index of a block is also calculated to identify those software blocks that significantly contribute to the delay time.