

한국시뮬레이션학회 '00춘계학술대회 논문집 2000.4.15 영남대학교

NGN 콜 에이전트의 속성기반 분석과 객체 지향 설계

이종혁*, 박성공*, 김영준*, 백두권*
*고려대학교 컴퓨터학과
e-mail: jhlee@swsys2.korea.ac.kr

Attribute Based Analysis and Object-oriented Design of NGN Call Agent

Jong-hyeok, Lee*, Sung-Kong Park*, Young-Jun Kim*,
Doo-Kwon Baik*

*Dept of Computer Science and Engineering, Korea University

요약

인터넷의 발전에 따라 데이터의 전송량이 음성 전송량을 능가하고 있으나 현재의 네트워크로는 빠르고 신뢰성 있는 데이터의 전송을 보장할 수 없다. 데이터와 음성 모두의 빠르고 신뢰성 있는 전송과 관리를 위하여 개발된 패킷중심 네트워크를 NGN이라 한다. NGN에서는 효율적인 전송과 관리를 위하여 기존(PSTN)의 교환기의 콜 에이전트를 소프트웨어로 구성하여 범용컴퓨터에서 운용될 수 있도록 하였다. 본 논문에서는 콜 에이전트를 소프트웨어로 설계하기 위하여 H/W개발을 위하여 설계된 기본 호처리모델인 BCSM을 객체지향방법론의 표준인 UML을 이용하여 설계를 하였다. 그리고 설계된 모델의 검증을 위하여 LSL(Larch Shared Language)를 이용하여 정형명세를 하였다.

1. 서론

현재 데이터의 전송량이 급격히 증가하여 음성 전송량을 능가하고 있는 실정이다[1]. 현재의 망 구조는 이러한 현상을 만족시키기에는 한계점을 가지고 있다. 차세대 네트워크(NGN) 개발 일환의 하나로 위의 문제점을 해결하기 위해 통합 네트워크가 제안되었다. NGN기반 통합 네트워크의 중요 개념은 패킷망이 데이터 전송과 관리의 중심이 된다는 것이다. 통합 네트워크에서 가장 중요한 요소는 호처리를 관리하는 콜 에이전트(Call Agent)이다. 콜 에이전트는 제조사에 의존하는 스위칭 시스템 대신 범용 컴퓨터에서 동작한다 [2].

범용 컴퓨터에서 동작하는 NGN 콜 에이전트의

효율적인 디자인을 위해서는 기존의 네트워크 호 처리에 대한 분석과 설계 구조가 필요하다. 현재 망의 망 교환(Network Switching) 시스템은 H/W의존적인 구조를 가지고 있는 반면, NGN 콜 에이전트 S/W는 범용 컴퓨터에서 동작을 한다. 따라서 NGN 콜 에이전트 S/W는 순수한 S/W 관점에서 설계되어야 한다.

본 논문에서는 효율적인 NGN의 설계를 위해 콜 에이전트의 속성기반 분석과 객체지향 설계를 제안한다. 속성기반 분석은 객체간의 관계성을 표현하는 방법을 제안하고, 제안된 객체지향 설계방법을 BCSM[3]에 적용한다. 우리는 UML(Unified Modeling Language)[4]을 이용하여 객체지향 설계를 하였고, 만들어진 모델의 검

증을 위하여 LSL(Larch Shared Language)[5]이라는 정형 명세언어를 이용하여 명세를 하였다.

본 논문에서 제안한 객체 지향 설계의 새로운 접근 법은 객체지향 모델을 설계하기 위한 보다 편리한 방법을 제공한다. 본 논문에서 제안한 분석/설계는 객체지향 파라다임의 많은 이득을 제공할 것이다.

2. 관련연구

1990년대 S/W의 위기와 함께 S/W의 재사용과 유지보수성을 높이기 위한 시스템 개발 방법론으로 객체지향 방법론이 다양하고 활발하게 대두되기 시작했다. 객체지향 방법론은 객체로 구성된 모델을 이용하여 문제에 접근하는 사고방식으로, 주체성(identity), 분류화(classification), 다형성(polymorphism), 상속(inheritance)등의 특징을 가지고 있다[6]. 객체지향 방법론들은 고유의 객체지향 개발에 대한 개념과 프로세스, 다이어그램 기법들을 내포하고 있다. 다음은 몇 가지 방법들의 주요개념이다[7].

- OOAD/OOD[8] : Coad & Yourdon

개발단계에 있어서 spiral, incremental 개념 적용

- OMT[6] : James Rumbaugh

분석단계의 프로세스는 선형적인 단계가 아닌 반복적이고 점진적인 과정 거침

- OOADA[9] : Grady Booch

설계의 문서화 강조하여 풍부한 표기법 제공, 반복적인 프로세스 정의

- OOSE[10] : Ivar Jacobson

유즈케이스를 이용하여 시스템의 기능적인 측면으로부터의 요구분석에 치중

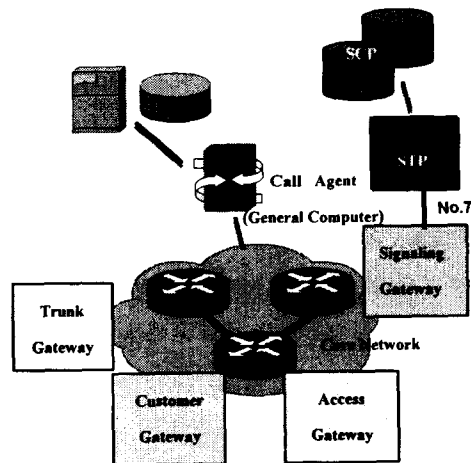
이러한 객체지향 방법론들은 다이어그램을 이용하여 가독성을 향상시켰으나, 정형성(formality)의 부족으로 인해 정확한 평가가 어렵다는 단점을 가지고 있다.

3. BCSM의 속성 기반 분석

1. 데이터의 음성, 비디오 통신이 가능한 하나의 범용 네트워크
2. 데이터 친화적이거나 데이터-기반의 패킷 전송과 스위칭 내부구조
3. 음성 통신과 데이터, QoS(Quality of Service)를 지원하는 유연한 서비스 제어 요소
4. 특성과 질을 표현할 수 있는 PSTN의 음성 부분

<그림 1> 통합망의 특성

빠른 속도로 증가하는 데이터 전송량의 효율적인 전송을 위하여 현존하는 음성 중심의 통신망에서 점차적으로 데이터와 음성을 통합하려는 노력들이 이루어지고 있고, 그 수단으로 급속한 발전을 하고 있는 인터넷을 선택할 것으로 예측된다[1]. 그러나 인터넷만으로는 현존하는 이종의 망들과의 호환이 불가능하다. 이러한 문제점을 해결하기 위해 NGN기반의 통합망이 제안되었고 통합망의 기본 개념 모델은 <그림 1> 과 같다[1].

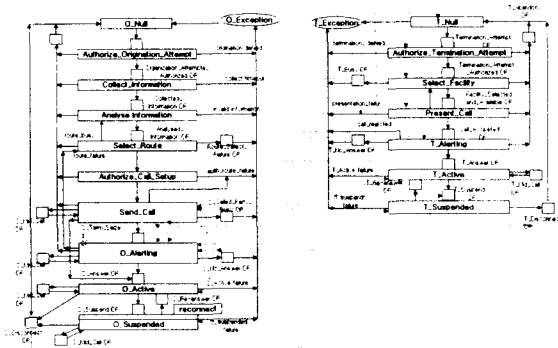


<그림 2> NGN의 기본 개념 모델[1]

<그림 2> 에서 볼 수 있는 것처럼, 중요한 망의 전송 경로를 제어하는 요소를 콜 에이전트라고 하고 이 콜 에이전트는 범용 컴퓨터에서 동작한다. 콜 에이전트의 중요 기능은 패킷기반 핵심망에서 음성 전송을 위한 호처리이다.

본 논문에서는 PSTN 스위치에서 데이터 관리

를 제어하는 역할을 하는 콜 에이전트를 다양한 기능적인 구조들간의 S/W관점으로 설계하였다. 이는 범용 컴퓨터에서 H/W적인 접근방법을 S/W적인 관점으로 구현함에 따라, NGN에서 PSTN과 IP 네트워크의 상호연결을 제공할 것이고, 통합과정에서 기존의 네트워크 토폴로지를 고려할 필요가 없고, 기존의 PSTN과 IP 네트워크에서 사용하던 응용과 서비스에서 사용하던 디렉토리(저장소)를 그대로 사용할 수 있다[1][2]. 그리고 새로운 프로토콜이나 디렉토리, 클라이언트 장치들을 NGN에 추가할 때 단지 새로운 게이트웨이와 장치서버만을 추가하면 된다[1]. NGN 콜 에이전트의 기능적인 모델로 선택된 ITU-T Q.1224 AIN BCSM은 통신 경로를 사용자가 상호연결하기 위한 통신 경로를 확보하기 위한 연결 제어와 기본 호를 구현하는 스위치의 부분에 대한 추상적인 모형을 제공한다[3].



〈그림 3〉 BCSM의 기본구조[3]

BCSM은 사용자에게 통신 경로를 관리하고 형성하기 위해 필요한 CCF(Call Control Function) 동작의 고수준 모델 설명을 제공하고, 기본 콜과 연결 이벤트를 설명하기 위한 프레임워크를 제공한다[11].

1. BCSM의 각 PIC를 하나의 객체로 정의
2. PIC로 들어오는 화살표를 INPUT으로 정의, INPUT으로 정의되는 경우는 해당 PIC로 들어오기 직전의 상태에 있는 경우를 의미
3. PIC에서 나가는 화살표를 OUTPUT으로 정의, OUTPUT으로 정의되는 경우는 해당 PIC에서 나가는 바로 다음 상태 의미
4. 각 PIC가 수행하는 기능과 Indicator의 PIC를 함수로 정의
5. 시스템 전체에 걸쳐서 이용되는 PARAMETER를 정의하여, 각 PIC에서 사용되는 것을 그 PIC의 PARAMETER로 정의

〈그림 4〉 속성기반 분석 방법

본 논문에서는 BCSM을 객체지향의 방법으로 설계를 하기 위하여 속성에 따른 분석법을 제안한다. 속성기반 분석은 특히 Functional Process를 분석하는데 이용하기 위한 분석법으로, 이 기법을 이용하면 기존의 프로그래밍 언어로 구현된 많은 S/W, 또는 BCSM과 같이 상태 도해도로 표현되는 모델을 쉽게 분석을 할 수 있고, 분석된 결과를 이용하여 객체지향 설계를 할 수 있다.

속성기반 분석을 하기 위하여 먼저 속성을 INPUT, OUTPUT, FUNCTION, PARAMETER 등으로 정의하였다. 속성기반 분석은 〈그림 4〉의 과정을 따르고, 〈그림 5〉는 위의 과정에 따라 분석된 결과의 일부이다.

4. BCSM의 객체 지향 설계

원래 BCSM은 H/W설계를 목적으로 만들어진 모델이다. H/W는 각 모듈별로 독립적으로 설계, 구현되기 때문에 재사용성에 대한 개념이 들어 있지 않다. 그러나 S/W구현을 위해서는 재사용성을 고려하여 개발비용과 유지 보수비용을 줄여 주는 것이 합리적이다[12]. 그래서 본 논문에서는 NGN 콜 에이전트의 설계를 위해서 UML(Unified Modeling Language)을 이용한 BCSM의 객체지향 설계를 하였다.

ASPECT	INPUT	OUTPUT	FUNCTION	PARAMETER
O_Null	O_Disconnect DP O_Abandon DP O_Exception	Authorize_Orgnation _Attempt		Orgnation_sub Occupy Orgnation_Device Occupy Charging off
Authorize_O rgnation_At tempt	O_Null	Organization_Attempt Authorized DF O_Abandon DP O_Exception	Authorize attempt Exception	Orgnation_sub Occupy Orgnation_Device Occupy Charging off
Collect_infor mation	Organization_Attem pt_Authorized DP	Collected_Information DP O_Abandon DP O_Exception	Collect information Exception	Orgnation_sub Occupy Orgnation_Device Occupy Charging off
Analyse information	Collected_Informa tion DP Select_Route	Analyse_Information DP O_Abandon DP O_Exception	Analyse information Exception	Orgnation_sub Occupy Orgnation_Device Occupy Charging off
Select_Route	Analyse_Information DP Send_Call O_Alerting	Analyse_Information Authorize_Call_Setup Route_Select_Failure DP O_Abandon DP	Select route Exception	Orgnation_sub Occupy Orgnation_Device Occupy Switch occupy Charging off
Authorize_C all_Setup	Select_Route	Send_Call O_Abandon DP O_Exception	Authorize call setup Exception	Orgnation_sub Occupy Orgnation_Device

〈그림 5〉 속성기반 분석 결과

1. INPUT 테이블 작성 : 각 행과 열에 각각의 PIC를 표시한 후, 같은 INPUT을 갖는 PIC에 1점을 준다.
2. OUTPUT 테이블 작성 : 각 행과 열에 각각의 PIC를 표시한 후, 같은 OUTPUT을 갖는 PIC에 1점을 준다.
3. FUNCTION 테이블 작성 : 각 행과 열에 각각의 PIC를 표시한 후, 같은 FUNCTION을 갖는 PIC에 5점을 준다.
4. PARAMETER 테이블 작성 : 각 행과 열에 각각의 PIC를 표시한 후, 같은 PARAMETER값을 갖는 PIC에 0.5점을 준다.
5. 가중치 테이블 작성 : 각 테이블의 같은 셀들에 대한 합을 내어 새로운 테이블을 작성한다. 셀 안의 값이 클수록 각 객체들의 관련성은 높다

〈그림 6〉 가중치 테이블 작성 알고리즘

4.1 가중치 테이블

BCSM을 UML의 클래스 다이어그램으로 그리기 위해서는 먼저 클래스를 추출하여야 한다. 클래스를 구성하기 위해서는 그들간의 관계성을 찾아내는 것이 중요한 요소이다. 본 논문은 클래스 추출과 그들간의 관계성을 찾아내기 위해 3장에서 분석된 결과를 기반으로 클래스들간의 관계성을 나타내는 가중치 테이블을 생성을 하였다.

클래스들간의 관계 중 일반화 관계는 가장 주목할 필요가 있다. 일반화 관계를 이용하면 S/W시스템의 재사용성을 증가시킬 수 있고, 시스템의 변경시 유지비용을 감소시켜준다[4].

〈그림 6〉은 가중치 테이블을 작성하는 방법이고 〈그림 7〉은 그 결과이다.

	SEND_C ALL	O_ALER TING	O_ACTI VE	O_SUSP ENDED	PRESEN T_CALL	T_ALER TING	T_ACTI VE	T_SUSP ENDED
SEND_C ALL		17.5	7	8.5	13	17	7	7.5
O_ALER TING			1	3	2	2	1.5	2.5
O_ACTI VE				14	8.5	9	20	14
O_SUSP ENDED					8	8.5	8.5	9.5
PRESEN T_CALL						8.5	9	9
T_ALER TING							9	9.5
T_ACTI VE								8.5
T_SUSP ENDED								

〈그림 8〉 가중치 테이블

BCSM의 경우 INPUT/OUTPUT보다 FUNCTION에 더욱 의존하기 때문에 FUNCTION에 보다 많은 가중치를 주었다. 이러한 가중치는 밀접도에 따라 다르게 구성될 수 있는데, 이는 NGN 콜 에이전트의 경우 적용되는 통신망의 종류와도 밀접한 관련성을 갖고 있다. 예를 들면 이동통신망과 유선통신망의 콜 에이전트는 가중치가 다를 수 있고, 이러한 가중치의 변경에 따라 클래스를 추출하는 과정의 유연성을 확보할 수 있다.

1. 각 PIC를 클래스로 정의하고 그들의 속성과 연산 찾아낸다.
2. 최소로 겹치는 클래스들간의 공통 속성과 연산 추출
3. 추출된 속성과 연산을 포함하는 새로운 부모 클래스 생성
4. 생성된 클래스의 임시 이름을 Temp #로 정의
5. 속성과 연산을 갖지 않는 부모클래스 삭제
6. 공통된 속성과 연산을 갖는 부모클래스가 없어질 때까지 2~5반복
7. 계층형 일반화 관계성 생성
8. 속성과 연산을 갖지 않는 클래스는 그 부모클래스와의 집단화관계 생성
9. 부모클래스의 적합한 이름 부여
10. BCSM을 참조하여, 일반화 관계와 집단화를 제외한 나머지 관계들을 연관관계로 나타낸다

〈그림 8〉 클래스다이어그램 작성 방법

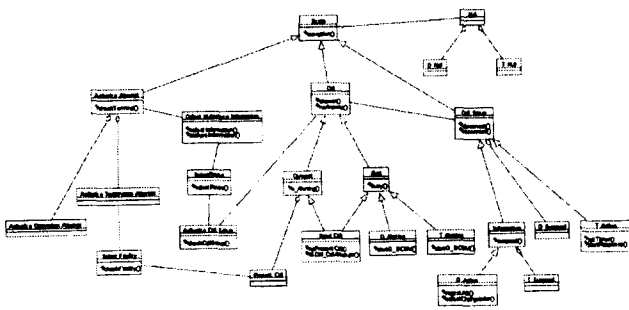
4.2 클래스 다이어그램의 작성

객체지향 방법론의 중심은 클래스 다이어그램이라 할 수 있다[12]. 클래스 다이어그램은 시스템에 있는 객체 타입과 그들 사이에 존재하는 여러

가지 정적인 관계를 설명해준다[4].

가중치 테이블을 이용하여 클래스 다이어그램을 만드는 방법은 <그림 8> 과 같고 그 결과로 만들어진 클래스 다이어그램은 <그림 9> 에 나타내었다. 이 방법론은 <그림 9> 에서 나타나 있듯이, 일반화 관계를 통한 상속관계를 설정하여 S/W개발 시 재사용성을 높이는데 주안점을 두었다.

4.3 동적 다이어그램[4][12]

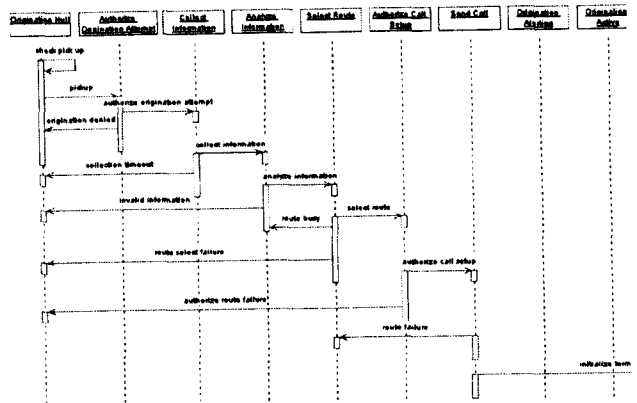


<그림 9> BCSM의 클래스 다이어그램

1. 각 PIC를 하나의 객체로 설정
2. <그림 5> 에서 분석된 결과(INPUT/OUTPUT)를 이용하여 하나의 객체에서 다음 객체로의 전이 정의
3. BCSM의 Indication을 참조하여 나머지 전이 정의

<그림 10> 순차도 작성 알고리즘

우리는 대상 시스템의 동적인 관계를 보여주기 위해서 교류도(Interaction Diagram)와 활동도(Activity Diagram)를 그렸다. 교류도는 어떤 행동 안에서 객체 그룹들이 어떻게 협력하는지를 설명해주는 모델로, 순차도(Sequence Diagram)와 협력도(Collaboration Diagram) 두 종류가 있다. 순차도는 작업이 어떤 순서대로 일어나는지를 쉽게 볼 수 있다는 장점을 가진다[4].



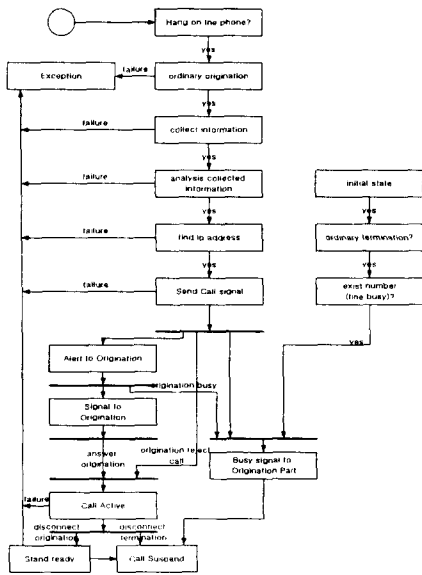
<그림 11> BCSM의 순차도

순차도를 그리는 방법은 <그림 10> 과 같고 그 다이어그램의 일부는 <그림 11> 과 같다.

활동도는 Jim Odell의 Event 도해, SDL의 상태 모델링 방법론, 그리고 Petri Net등에서 아이디어를 가져왔다. 활동도의 특징은 워크플로우와 연계했을 때와 많은 병렬처리를 가진 행동을 설명하는데 유용하다[12]. 활동도를 그리는 방법은 <그림 12> 와 같고 그 활동도는 <그림 13> 과 같다.

1. 시스템의 활동(Activity)을 나타내는데 필요한 PIC를 추출한다.
2. 추출된 PIC를 활동으로 정의한다.
3. 정의된 활동과 연결되어 있는 DP를 동기막대(Synchronization bar)로 정의한다.
4. <그림 5> 에서 분석된 결과(INPUT/OUTPUT, FUNCTION)을 이용하여 활동들과 동기막대간의 관계를 정의한다.

<그림 12> 활동도 작성 알고리즘



〈그림 13〉 BCSM의 활동도

다이아그램의 정형명세

Algorithm : Derive Class Specifications
Input : Let $V = \{V_1, \dots, V_k\}$ be the base types of object model O . Let $clssq$ be a queue of class names which initially contains the classes of V in any order.
Output: Formal specifications of classes
Procedure:
 let $clssq \leftarrow$ elements of V in any order
 while $clssq$ is not empty do
 let $stypes \leftarrow \{d_1, \dots, d_k\}$ where each d_i is a
 for each $d \in stypes$
 if d is not in $clssq \wedge$ specification(d) has not yet been computed,
 then enqueue d on $clssq$.
 endfor
 compute specification(c)
 end while

〈그림 14〉 클래스 명세 추출 알고리즘[13]

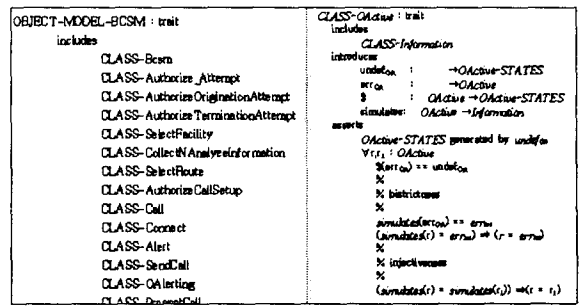
UML은 요구 분석과 시스템 설계 과정에 유용한 구조화된 언어이지만, 정형성의 결여로 인해 완결성(completeness), 지속성(consistency)과 설계 명세(des ign specification)의 평가가 어려운 실정이다[5][13].

UML과 같은 비정형적 접근법으로 개발된 시스템의 신뢰성을 확보하기 위해서는, 모든 개발 단계에서 복잡한 검증과정이 이루어져야 하고 확장된 테스트 방법의 사용이 요구된다. 본 논문은

설계되어진 클래스 다이어그램의 신뢰성 확보를 위해, 명세 언어로 LSL을 선택하였다. LSL은 간단한 구문을 가지고 있고, 구문 검사기, Larch prover, 그래픽 브라우저와 개발 환경 등을 포함한 툴들을 가지고 있다는 장점을 가진다.

클래스가 일반화관계를 가지고 있을 때, 클래스의 명세 순서가 중요한 역할을 하게 된다. 〈그림 14〉는 간단한 클래스 명세 추출 알고리즘이다[13]. 〈그림 15〉는 위의 알고리즘을 이용하여 〈그림 9〉의 클래스 다이어그램의 LSL 명세 일부이다.

4.4 클래스



〈그림 15〉 클래스 다이어그램의 LSL 명세

5. 결론

본 논문에서는 NGN 콜 에이전트를 구현하기 위하여 속성기반 분석을 통한 새로운 객체지향 설계 방법을 제안하였다.

ITU-T Q.1224의 BCSM을 대상으로 4가지의 속성을 기반으로 분석을 하였고, 분석된 결과를 이용하여 UML 기반의 객체지향 설계 기법과 변환 알고리즘을 제안하였다. 본 논문에서 제안된 설계 기법은 속성의 중요도에 따른 가중치 설정을 이용한 것으로, 기존의 기능 중심의 설계기법보다 유연성과 적용성 측면에서 보다 유리하다.

기존의 객체지향개발 방법론에서는 클래스를 추출하고 그들간의 관계성을 찾아낼 때, 객관적인 판단 기준이 아닌 설계자의 경험에 의한 주관적 판단에 의하여 관계성을 정의하였다. 반면, 시스템의 속성을 기반으로 분석한 결과를 이용하여 경험이 풍부하지 않은 설계자도 쉽게 관계성을

추출할 수 있는 방법을 제안한 본 논문의 방법은 처음 개발하는 시스템의 경우보다는 흐름 중심 설계되어 있는 시스템을 객체지향으로 설계하는데 적합하다. 또한 기존의 방법론에서는 정형성의 부족으로 인한 문제점을 가지고 있었으나, 본 논문에서는 제안된 알고리즘에 의하여 새로 설계된 클래스 다이어그램의 적용성을 확인하기 위한 방법으로 LSL이라는 정형 언어를 이용하였다.

참고문헌

- [1] "An introduction to Lucent tech. Softswitch -e nabling the promise of NGN", Lucent Tech., Inc., June 16, 1999
- [2] "Lucent Technology Softswitch", <http://www.lucent.com/IN/softswitch.html>
- [3] ITU-T Q.1224, "Draft Recommendation, Part1 INCS-2", International Telecommunications Union, September, 1997
- [4] Grady Booch, James Rumbaugh, Ivar Jacobson, "The Unified Modeling Language User Guide", Addison Wesley
- [5] "Larch Homepage", <http://www.sds.lcs.mit.edu/spd/larch>
- [6] James Rumbaugh, "Object-Oriented Modeling and Design", Prentice Hall
- [7] 신신애, "객체지향 방법에 기초한 모델링 다이어그램에 UML로의 변환 규칙", 고려대학교 석사학위논문, April, 1999
- [8] Peter Coad & Edward Yourdon, "Object-Oriented Analysis", Prentice Hall
- [9] Grady Booch, "Object-Oriented Analysis and Design with Application", Addison Wesley, 1994
- [10] Ivar Jacobson, "Object-Oriented Software Engineering", Addison Wesley, 1996
- [11] 최고봉, 김기령, "지능망 기술", 홍릉과학 출판사, Oct, 1996
- [12] Martin Fowler, Kendall Cott, "UML Distilled", Addison Wesley
- [13] Robert H. Bourdeau, Betty H.C.Cheng, "A formal semantics for object model diagrams", IEEE Transactions on SE, October, 1995