

Applicable SIB에 의한 IMT-2000 분산 망관리 에이전트의 인터페이스 스펙 모델링

박수현*

Interface Specification Modeling for Distributed Network
Management Agent of IMT-2000 Based on Applicable Service
Independent Building Blocks

Soo-Hyun Park*

■ Abstract ■

It is noteworthy that IMT-2000 communication network based on All-IP/AIN(Advanced Intelligent Network) should accommodate current and future wire/wireless AIN service easily through integration and gearing AIN construction elements. In this paper, Intelligent Farmer model(I-Farmer Model) and methodology are suggested in order to solve the several problems including standardization on implementation of Q3 interface in Telecommunication Management Network(TMN) agents which is caused by heterogeneous platform environment and future maintenance. Also this paper proposes ITI algorithm transforming the system which is designed by I-Farmer model to Interface Specification Model(ISM) applying the I-Farmer model. In addition to ITI algorithm, we suggest NTS(Node to SIB) algorithm converting entity node and ILB/OLB component in agent system designed by the I-Farmer model to SIB of AIN GFP(Global Functional Plane) and to ASIB for application program.

Keyword : IMT-2000, AIN, I-Farmer Model, ITI Algorithm, NTS Algorithm

1. 서론

현재 우리나라를 포함하여 전세계적으로 서비스를 실행할 예정인 IMT-2000은 ITU-R의 TG/1에서 연구/권고한 제 3세대 이동통신 시스템(TGMS : Third Generation Mobile System)으로서 이동통신 사용자의 요구자에 따라 PSTN(Public Switching Telephony Network), PCN(Personal Communication Network), DCN(Digital Cellular Network) 및 B-ISDN(BroadBand-ISDN) 등과 같은 타 이동통신 서비스 및 고정망에 의해 지원되는 다양한 통신 서비스를 다양한 무선 링크를 이용하여 접속하도록 하는 이동통신 시스템이다[1, 5-7].

IMT-2000 망은 All-IP/AIN(Advanced Intelligent Network) 개념을 기반으로 구성하게 되면 자연스럽게 지능망의 구성요소들과 통합 및 연동이 가능하게 되어 기존에 개발되었거나 앞으로 개발될 유선 지능망의 서비스들을 쉽게 수용할 수 있게 된다[5-6]. 하지만 IMT-2000 및 IMT-2000과 연동되는 PSTN, PCN 등과 같은 타 네트워크는 이미 구축과정에서 서로 다른 하드웨어와 운영체제 등 상이한 플랫폼 환경 하에서 개발되어 있고 향후의 유지보수과정에서도 동일한 상황이 발생할 것으로 예측되고 있다. 대표적인 문제점으로는 TMN(Telecommunication Management Network)[8-10] 시스템내의 에이전트들이 Q3 인터페이스 구현상의 표준을 이룰 수 없을 뿐만 아니라 다중 플랫폼을 지원할 수 없게 된다. 또한 IMT-2000과 연동되는 서로 다른 망의 유지보수 시스템 사이의 호환성이 보장되지 않는다.

이와 같은 문제점을 해결하기 위하여 본 논문에서는 I-Farmer 모델(Intelligent-Farmer Model)을 제안하였다. I-Farmer 모델은 PICR(Platform Independent Class Repository)[12-13, 16-18]에 기반한 ASR(Applicable SIB Repository)을 이용한 TMN Agent 생성모델인 Farmer 모델[12-13] 및 지능망 서비스 생성 모델인 SCSM(Service Creation Sequence Model) 개념[1, 15]을 상호연동시킨

모델이다.

I-Farmer 모델에 기반한 I-Farmer 방법론은 이미 정의되어 있는 SIBs(Service Independent Building Block)[1, 15]를 이용하여 통신망 상의 신규 서비스를 생성할 수도 있을 뿐만 아니라 기존에 정의되어 있는 SIB로는 신규 통신 서비스의 생성이 불가능할 경우 이를 위하여 새로이 만들어지는 신규 SIB를 SMP(Service Management Point)/SCEP(Service Creation Environment Part)[1, 15]내의 DB에 저장하도록 하는 기능을 포함하고 있다. 특히 신규통신 서비스 뿐만 아니라 NE NMS Agent 등과 같은 네트워크 구성요소를 구성하는 응용프로그램을 생성하기 위하여 ASIB(Applicable SIB)의 개념을 뒀으므로 SMP ASR에 저장된 ASIBs를 이용하여 I-Farmer 모델에 의하여 설계된 NMS(Network Management System) 에이전트[8-10, 12]등의 응용 프로그램을 구성할 수 있다. I-Farmer 방법론의 단계들은 전단계(Front step) 후단계(End Step)으로 구분된다. 또한 본 논문에서는 I-Farmer 모델링 방법론에 의하여 디자인되어진 시스템을 컴포넌트 기반개발의 인터페이스 명세모델로의 변형(transformation) 메카니즘인 ITI 알고리즘을 제안하여 JAVA RMI 및 분산 컴포넌트 개발환경인 IDL 및 EJB[2-4, 11]와의 상호관련성에 대하여 예를 들어 설명하고 있다.

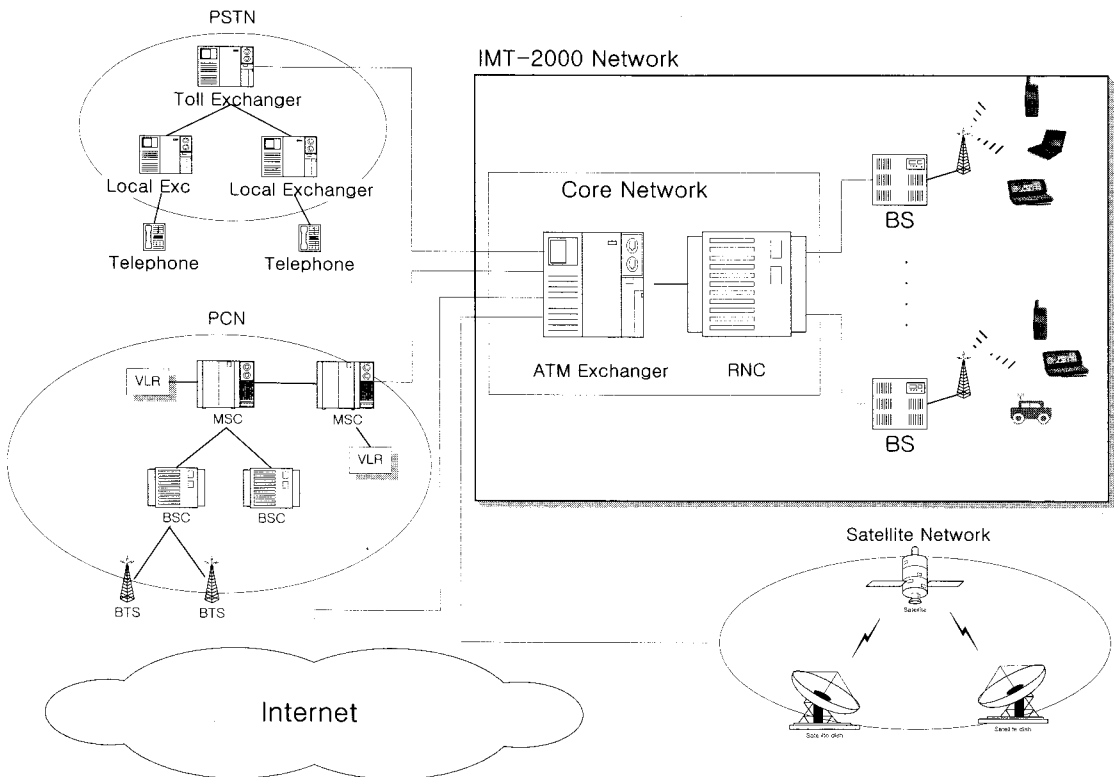
본 논문의 구성은 다음과 같다. 2장에서 IMT-2000의 망관리 요소시스템에 대하여 언급하고 있으며 3장에서는 TMN Agent 생성모델인 I-Farmer 모델과 지능망 서비스 생성 모델인 SCSM(Service Creation Sequence Model) 개념을 상호연동시킨 모델인 I-Farmer Model을 소개하고 있다. 4장에서는 I-Farmer 모델에 의하여 정의된 개념을 인터페이스 명세모델(ISM)에서의 개념으로 변형시키는 방법 및 알고리즘에 대하여 설명하고 있다. 5장에서는 I-Farmer 모델로 설계한 에이전트 시스템 중 Entity Node 및 ILB/OLB 컴포넌트를 AIN의 GFP(Global Functional Plane)내의 SIB 및 시스템 응용프로그램 생성 컴포넌트인 ASIB로 변형(trans-

formation)하는 알고리즘인 NTS(Node to SIB) 알고리즘에 상술하고 있다. 마지막으로 결론을 내리고 있다.

2. IMT-2000 망관리 요소시스템

IMT-2000망 구조[1, 5-7]는 PCN(Personal Communication Network)이나 DCN(Digital Communication Network)와 큰 차이가 없으나 이동통신 핵심망(CN : Core Network)과 무선접속망(RAN : Radio Access network)을 구분된다는 점에서 차이점을 찾을 수 있다. 이는 UMTS의 GRAN개념을 받아들인 것으로서 무선규격의 변화를 수용하는 RAN 부분이 변경되더라도 전달망을 포함하는 CN 부분은 최대한 다시 이용할 수 있도록 한 것이다. 따라서 그동안 핵심망의 이동교환기에 거의 대부분 집중되어 있던 중요기능들 중 핸드오버 제어 등

과 같이 무선자원관리와 밀접한 관계가 있는 많은 기능들이 RAN으로 넘겨지고 CN은 본래의 교환 및 전달 기능을 충실히 수행할 수 있게 됨과 동시에 지능망 및 B-ISDN의 발전을 수용하여 진화시키기 용이하게 되었다. IMT-2000은 AIN(Advance Intelligent Network) 서비스를 처리하기 위한 SCP (Service Control Point), SDP(Service Data Point), SSP(Service Switching Point) 등이 부가되어 추후 유.무선의 통합 및 서비스의 진화추세를 수용하기 용이하게 된다는 점도 큰 차이점의 하나이다. 기존의 이동통신서비스는 SMC, FMC, VMC 등과 같은 별도의 서비스 장치를 설치해야만 각종 부가 서비스를 제공할 수 있었다. 그러나 지능망을 기반으로 IMT-2000망을 구성하게 되면 자연스럽게 지능망의 구성요소들과 통합 및 연동이 가능하게 되어 기존에 개발되었거나 앞으로 개발될 유선 지능망의 서비스들을 쉽게 수용할 수 있게 된다. 이밖



[그림 1] IMT-2000 요소 시스템

에도 위성망과의 연동 및 고속 DATA서비스와 멀티미디어 서비스의 지원을 위한 별도의 기능 및 장치가 추가되었다는 점이 기존의 이동통신망과의 큰 차이점이라고 할 수 있다.

[그림 1]에서 보여주는 IMT-2000 요소 시스템은 현재로서는 기본 모델만이 제시되고 있을 뿐이며 각각의 요소시스템들이 담당하는 기능은 기존 이동통신 시스템과 기본적으로는 크게 다르지 않다. IMT-2000 NE(Network Element) 망관리 요소 시스템은 다음과 같이 구분할 수 있다.

2.1 무선 접속망(RAN : Radio Access Network)

1) BS(Base Station)

송수신 및 유.무선 변환을 담당

2) RNC(Radio Controller)

단말기의 망접속 제어, 무선자원 관리, 기지국 운용 관리 등 이동통신의 핵심 망과의 접속을 담당

2.2 핵심망(CN : Core Network)

1) MSC(Mobile Switching Center)

호 제어, 연결제어 등을 포함하여 교환 및 전달을 담당, MSC는 그 역할에 따라서 AMSC, DMSC, GMSC 등으로 구분

- AMSC(Anchor MSC) : 교환 및 전달. 교환기간 handover 제어·교환기간 핸드오버시 제어의 주체
- DMSC(Drift MSC) : 교환 및 전달
- GMSC(Gateway MSC) : 타 망으로의 루팅(routing)이나 타 망으로부터의 루팅을 수행하는 관문 역할을 담당

하나의 MSC는 AMSC와 GMSC의 역할을 모두 수행할 수도 있다.

2) VLR(Visitor Location Register)

위치영역내 사용자의 사용자 정보 및 위치정보

저장

3) HLR(Home Location Register)

자신의 망에 소속된 사용자의 사용자정보 및 위치정보 저장

4) GLR(Gateway Location Register)

타망에 소속된 사용자의 사용자의 사용자 정보 및 위치정보 임시저장

이밖에 인증을 담당하는 AC(Authentication Center), 지능망의 구성요소인 IP(Intelligent Peripheral), SCP(Service Control Point) 및 SDP(Service Data Point) 등이 망관리 요소로 구분할 수 있다.

IMT-2000망의 표준화에 있어서 2단계에 속하는 망구조 및 정보흐름도가 완성되면 모든 정보흐름을 INAP으로 규정하고 MSC, LR, AC, SCP, SDP 등의 동작절차 및 데이터 모델도 지능망과 호환될 수 있도록 표준화하게 된다. 그러나 최근의 안정화 버전의 권고초안들(Q.1211-1229)이 작성된 IN CS-2의 프로토콜은 이러한 IMT-2000의 표준화를 지원하기에는 부족한 점이 많으며 향후의 추세는 All-IP망을 중심으로 IMT-2000의 모든 동작절차와 정보흐름 및 데이터 모델링이 진행될 전망이다. 이에 따라 INAP이 적용되더라도 IMT-2000 서비스 초기에는 MSC와 SCP 사이의 인터페이스, 그리고 SCP 상위에 존재하게 될 서비스 생성 및 관리 부분에 국한될 예정이다.

3. I-Farmer Model

국제 표준화 기구인 ITU-T의 IN CS-2에서는 "SCE는 망 구성 및 망 유형과는 독립적으로 새로운 서비스를 손쉽게 개발하도록 하는 환경으로서, 서비스 스크립트를 개발할 수 있는 과정을 지원하기 위하여, 서비스 구현과 독립적으로 도구, 기술(description), 프로그래밍 언어, 절차 등을 제공하여야 한다"고 규정하고 있다.

ITU-T를 비롯한 세계 여러 표준화 기구에서는

지능망의 표준화를 위해 활발히 활동하고 있지만, 아직도 SCE에 대한 구체적인 표준안은 제시하지 못하고 있다. 이것은 지능망과 지능망 서비스 구조의 표준이 완성되지 않은 상태에서 SCE의 구체적 표준화가 어렵기 때문이다. 따라서 통신 선진국에서는 SCE의 국제 표준화 노력과 더불어, 각자 독자적으로 개발을 진행하고 있는 상황이다. 그러므로 실제 발표되고 있는 SCE들은 개발 환경을 구성하고 있는 지원 도구들은 물론, SCE의 개발에 이용한 컴퓨팅 환경이나 구현언어 등에서도 차이가 있다.

3.1 Farmer Model

Farmer 모델의 주요 목적은 실제 디자인 하고자 하는 에이전트를 분석하여 에이전트를 구성하고 있는 컴포넌트 요소들을 측면[14]에 따라 분리, 추출해 내는 데 있다. 이러한 결과 추출된 컴포넌트 요소들은 Farmer 모델 트리의 리프노드(leaf node)에 위치하게 되며 이러한 컴포넌트 요소들은 네트워크를 통하여 최종적으로 PICR에 저장된다. 또한 Farmer 모델은 플랫폼독립형 클래스저장소(PICR)에서 컴포넌트 요소를 에이전트로 동적 또는 정적으로 다운로드하는 Farming의 개념을 추가한 형식모델이다. Farmer 모델은 실세계의 객관적, 추상적 대상체를 표현하는 개체노드 타입(entity node type), 이러한 개체를 표현하는 관점인 측면노드 타입(aspect node type), 개체와 측면간의 관계성을 나타내는 링크타입(link type) 그리고 개체가 가지는 성질을 나타내는 속성타입(attribute type), 동일한 이름을 가지는 2 노드는 동일한 속성과 동일한 서브트리를 갖는다는 균일성의 원리에 의해 정의되는 균일성 개체노드 타입(uniformity entity node type)과 균일성 측면노드 타입(uniformity aspect node type) 그리고 Farming 시 ILB(Initial Loading Block) 컴포넌트에 해당하는 속성을 지니는 IM-컴포넌트 타입 노드(IM component type node : ILB Multiplicity Component type node), OLB(on-demand Loading Block) 컴포넌트의 속성

을 지니는 OM-컴포넌트 타입 노드(OM component type node : OLB Multiplicity component type node) 등의 구성요소들과 generalization, aggregation, multiplicity 등의 3가지 추상화 개념들로 구성된다.

3.2 Intelligent Farmer Model(I-Farmer Model)

I-Farmer모델(Intelligent-Farmer Model)은 PICR(Platform Independent Class Repository)을 이용한 TMN Agent 생성모델인 Farmer 모델 및 지능망 서비스 생성 모델인 SCSM(Service Creation Sequence Model) 개념을 상호연동시킨 모델이다. I-Farmer 모델에서는 PICR을 대신하여 ASR(Applicable SIB Repository)의 개념을 두고 있는데 ASR과 PICR의 차이점은 다음과 같이 구분할 수 있다. I-Farmer 모델에서의 PICR은 ILB 및 OLB 등과 같은 비 표준화형태의 컴포넌트의 저장소(repository)의 의미를 지니고 있는 반면 ASR은 지능망에서 제공되는 SIB의 개념에 의거한 표준화된 컴포넌트의 의미를 지니고 있다. 따라서 ASR내의 ASIB는 IMT-2000망 내의 NE 및 IMT-2000과 연동하는 대부분의 타 네트워크의 구성요소들의 망관리 에이전트 등과 같은 응용 프로그램들의 제작에 사용될 수 있는 표준화된 응용프로그램이라고 볼 수 있다.

I-Farmer 모델은 Farmer 모델에 기본을 두고 있으며, SCSM의 핵심개념인 SIB를 중심으로 하여 다음과 같은 형식론을 갖는다.

SIB는 표준화된 인터페이스를 가지고 있으며 SIB에 대한 정의는 서비스 생성 관점에서의 1단계 기술과 구현을 위한 2단계 기술로 나눌 수 있다. 1단계 기술에 의한 SIB는 다음과 같이 정의된다.

[Definition 1] Information Structure

F 를 SIB에 의하여 수행되는 Operation 집합내의 Information Flow Set이라고 하면, F 는 다음과 같이 정의된다.

$$\forall f \in \mathbf{F}, f = \langle \mathbf{Fe}, \mathbf{R}, \lambda, \mathbf{RN} \rangle$$

where, \mathbf{Fe} : Function Entity

\mathbf{R} : Inter-Relationship between
Function Entities

$$\lambda : \mathbf{Fe} \rightarrow \mathbf{Fe}$$

for $\exists f_e \in \mathbf{Fe}$,

$$\lambda(f_e) = \mathbf{Have_Relations_Of}(f_e)$$

\mathbf{RN} : Reference Number of Function
Entity Operation ■

SIB에 의하여 수행되는 operation들의 집합은 다음과 같이 정의할 수 있다.

[Definition 2] Operation Structure

\mathbf{Op} 를 SIB에 의하여 수행되는 Operation 집합이라고 하면, \mathbf{Op} 는 다음과 같이 정의된다.

$$\forall o \in \mathbf{Op}, o = \langle \text{oid}, \mathbf{F} \rangle$$

where, oid : 동작 o의 이름

\mathbf{F} : Information Set ■

Definition 1과 Definition 2를 기반으로 하여 SIB 형식구조를 정의하면 다음과 같다. SIB 형식구조는 predicate login에 의하여 정의된 구조(structure)로서 정의된다.

[Definition 3] SIB Formal Structure

\mathbf{S}_{ib} 를 SIB 구조들의 집합이라고 하면 다음과 같이 정의된다.

$$\forall b \in \mathbf{S}_{ib}, b = \langle \text{Sid}, \mathbf{S}_f, \mathbf{S}_v, \mathbf{I}, \mathbf{O}, \tau_n, \Gamma_n, \mathbf{L}_p \rangle$$

where, Sid : SIB의 이름

\mathbf{S}_f : Service Feature Set

AIN Service Plane에서 정의되는 서비스 특징들을 의미하며 AIN CS-X의 경우에 따라 구성요소는 달라진다.

$$\mathbf{S}_f = \{ \text{ABD, ATT, AUTC, AUTZ, ACB, ... , TCS, TDR} \}$$

\mathbf{S}_v : 적용대상 서비스의 집합

본 SIB가 적용될 수 있는 AIN 서비스 집합

$$\mathbf{S}_v = \{ \text{ABD, ACC, AAB, CD, CF, CRD, CCBS, CON, ... , MAS, APP} \}$$

\mathbf{I} : 입력데이터 집합으로 다음과 같은 요소들을 갖는다.

$$\mathbf{I} = \{ \text{Service_Support_Data, Call_Instance_Data, C}_{isp} \}$$

\mathbf{O} : 출력데이터 집합으로 다음과 같은 요소들을 갖는다.

$$\mathbf{O} = \{ \text{Call_Instance_Data, C}_{lep} \}$$

$$\tau_n : \mathbf{S}_v \rightarrow \mathbf{S}_f$$

각 서비스를 생성하기 위하여 서비스 평면(service plane)에서 정의된 Service Feature들과 서비스사이의 관계를 정의

$$\text{From } \mathbf{S}_v = \{ v_i \mid 1 \leq i \leq n \} \text{ and } \mathbf{S}_f = \{ f_j \mid 1 \leq j \leq n \},$$

For $\exists v_i \in \mathbf{S}_v, v_i \leftarrow$

SSL($\exists f_j \in \mathbf{S}_f$)

$$\Gamma_n : \mathbf{S}_f \rightarrow \mathbf{S}_{ib}$$

서비스 평면(service plane)에서 정의된 Service Feature들과 이를 지원하기 위하여 필요한 GFP(Global Function Plane)의 SIB들의 관계를 정의

$$\text{From } \mathbf{S}_f = \{ f_i \mid 1 \leq i \leq n \} \text{ and } \mathbf{S}_{ib} = \{ b_j \mid 1 \leq j \leq n \},$$

For $\exists f_i \in \mathbf{S}_f, f_i \leftarrow \text{GSL}(\exists b_j \in \mathbf{S}_{ib})$

\mathbf{L}_p : Logical Point Constant Set

Logical Point Constant Set은 다음과 같이 정의된다.

$$\mathbf{L}_p = \{ C_{isp}, C_{lep} \}$$

where, C_{isp} : Logical Start Point
Constant

C_{lep} : Logical End Point
Constant ■

ASIB는 Service Creation에 사용되는 SIB가 아닌 TMN Agent나 교환시스템 내에서 사용되는 응용 프로그램을 생성하는 사용되는 SIB를 의미한다. ASIB는 I-Farmer 모델에 의하여 정의된 중간 enti-

ty node, ILB/OLB 등의 lead node 등이 NTS 알고리즘을 통하여 변형되어 생성되어지게 된다. ASIB는 SIB 집합 중에 여러 조건을 만족하는 SIB 요소들로서 정의된다.

[Definition 4] ASIB Structure

Applicable SIB (ASIB)는 다음과 조건을 갖는 SIB로 정의된다.

- 1) A_{sib} 를 **Applicable SIB (ASIB)** 집합이라고 하면

$$A_{sib} \subset S_{ib}$$

- 2) For $\forall b \in S_{ib}, \forall a \in A_{sib}, \text{ iff } \{ APP \} \in b.S_f$

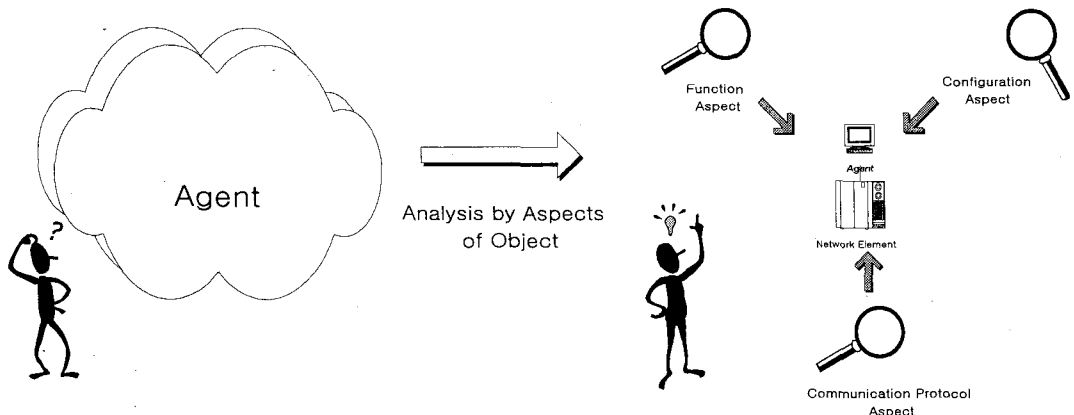
4. 인터페이스 정의를 위한 I-Farmer 모델변형 알고리즘

I-Farmer 모델은 Tangled-mess-of-object 형태의 실제계의 사물을 속성별로 다중측면에 의하여 분석한 후 컴포넌트 Farming 개념에 의하여 디자인하는 개념을 설명하고 있다. [그림 2]는 이와 같은 I-Farmer 모델의 문제분석(problem-decomposition) 개념을 보여주고 있다. 본 절에서는 I-Farmer 모델에 의하여 정의된 개념을 인터페이스 명

세모델에서의 개념으로 변형시키는 방법 및 알고리즘에 대하여 설명하고 있다.

<표 1>은 이러한 양쪽 모델의 개념상의 매핑관계를 보여주고 있다. I-Farmer모델에서의 측면노드의 개념을 지원하기 위하여 측면 인터페이스(Aspect Interface)의 개념을 두었으며 Aspect Interface에 대한 상세정보를 보관하기 위하여 측면 인터페이스 카탈로그(Aspect Interface Catalog)를 별도로 유지하고 있다. 균일성 개체노드(uniformity entity node)는 인터페이스 카탈로그내의 동일한 이름을 갖는 인터페이스로 매핑이 되고 균일성 측면 개체노드는 측면 인터페이스 카탈로그내의 동일한 이름을 갖는 측면 인터페이스로 매핑이 이루어진다. I-Farmer 모델에서의 다중 추상화(multiplicity abstraction)개념에 의한 대표개체노드(representative entity node)는 대표 인터페이스(Representative interface)로 변형되며 ILB 컴포넌트는 정적인터페이스 호출(Static Interface Invocation)을 지원하는 인터페이스로 매핑되는 데 속성 중 Type_Of_BasicComponent값을 ILB로 정한다. OLB LP컴포넌트의 경우 동적인터페이스 호출(DII : Dynamic Interface Invocation)을 지원하는 인터페이스로 매핑되는 데 속성 중 Type_Of_Basic Component 값을 OLB로 정한다.

I-Farmer 모델에 의하여 정의된 개념을 인터페이스 명세모델에서의 개념으로 매핑시키기 위하여



[그림 2] I-Farmer 모델의 문제분석(problem-decomposition) 개념

<표 1> I-Farmer 모델과 Interface Specification Model 모델의 Concept Mapping 관계

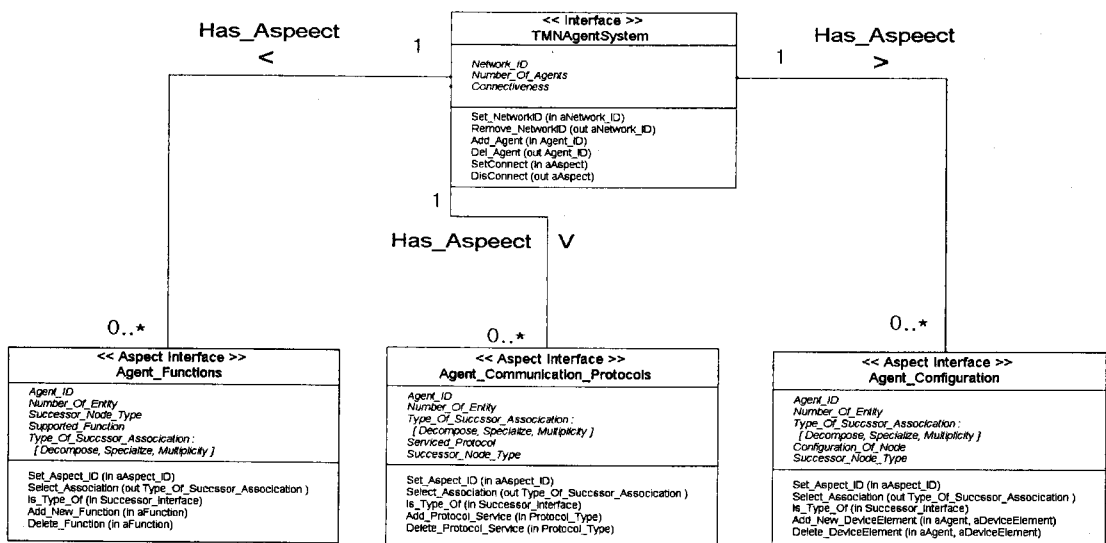
| The Farmer Model | Interface Specification Model |
|-------------------------------|--|
| Entity node | Interface |
| Aspect entity node | Aspect Interface |
| Uniformity entity node | Interface which has the same name in the interface catalog |
| Uniformity aspect entity node | Aspect Interface which has the same name in the aspect interface catalog |
| IM-Component | Interface which supports Static interface Invocation(SII), and attribute Type_Of_BasicComponent has "ILB" value |
| OM-Component | Interface which supports Dynamic interface Invocation(DII), and attribute Type_Of_BasicComponent has "OLB" value |
| Decomposition | Decomposition |
| Specialization | Specialization |
| Multiplicity Abstraction | Refinement & Representative interface |

I-Farmer모델의 구성요소를 기준으로 다음과 같은 사항들을 고려하여야 한다.

- Aspect interface에 추가되는 기본속성
 Agent_ID
 Number_Of_Entity
 Successor_Node_Type
 Supported_Function
 Type_Of_Successor_Association :
 [Decompose, Specialize, Multiplicity]
 Serviced_Protocol
 Configuration_Of_Node

4.1 I-Farmer 모델에서의 Aspect entity node

- ISM의 Aspect Interface로 매핑
- Aspect Interface Catalog 운영
- IFR(Interface Repository)에 AIFR(Aspect IFR)을 별도로 유지



[그림 3] Aspect Interface

- Aspect interface에 추가되는 메소드
Set_Aspect_ID (in aAspect_ID)
Select_Association
 (out Type_Of_Succssor_Association)
Is_Type_Of (in Successor_Interface)
Add_New_Function (in aFunction)
Delete_Function (in aFunction)
Add_Protocol_Service (in Protocol_Type)
Delete_Protocol_Service (in Protocol_Type)
Add_New_DeviceElement (in aAgent,
 aDeviceElement)
Delete_DeviceElement (in aAgent,
 aDeviceElement)

[그림 3]은 이와 같은 측면 인터페이스(aspect interface)의 예를 보여주고 있으며 이는 다음과 같은 CORBA IDL로 프리프로세싱된다.

- IDL Description<표 2>

4.2 I-Farmer 모델에서의 Multiplicity abstraction에 의한 representative entity node

- interface로 매핑, interface이름을 representa-

tive_NODE_ID로 정함

- 기본속성
Attribute_Of_BasicComponents : [ILB, OLB]
Number_Of_BasicComponents
Type_Of_Succssor_Association :
 [Decompose, Specialize, Multiplicity]
- 기본 operations
Assign_Attribute_To_BasicComponent
(in BasicComponent)
Is_ComponentType_Of (in BasicComponent)
Is_InterfaceType_Of (in Successor_Interface)
Add_BasicComponent_To_MultiplicityLink
(in aComponent)
Delete_BasicComponent_To_MultiplicityLink
(in aComponent)
Select_Association
(out Type_Of_Succssor_Association)

4.3 I-Farmer 모델에서의 ILB

- Static Interface Invocation
- Interface로 매핑

<표 2> TMN_Agent_Aspects.IDL

```
// TMN_Agent_Aspects.idl
Module TMN_Agent_Aspects {
    Aspect Interface Agent_Functions {
        void Set_Aspect_ID (in string aAspect_ID)
        void Select_Association (out any Type_Of_Succssor_Association )
        void Is_Type_Of (in any Successor_Interface)
        void Add_New_Function (in any aFunction)
        void Delete_Function (in any aFunction) }
    Aspect Interface Agent_Communiation_Protocols {
        void Set_Aspect_ID (in string aAspect_ID)
        void Select_Association (out any Type_Of_Succssor_Association )
        void Is_Type_Of (in any Successor_Interface)
        void Add_Protocol_Service (in any Protocol_Type)
        void Delete_Protocol_Service (in any Protocol_Type) }
    Aspect Interface Agent_Configuration {
        void Set_Aspect_ID (in string aAspect_ID)
        void Select_Association (out any Type_Of_Succssor_Association )
        void Is_Type_Of (in any Successor_Interface)
        void Add_New_DeviceElement (in any aAgent, in string aDeviceElement)
        void Delete_DeviceElement (in any aAgent, in string aDeviceElement) }
```

- 속성 중 Type_Of_BasicComponent 값이 ILB 인 것으로 판단

4.4 OLB

- Dynamic Interface Invocation 수행
- Interface로 매핑
- 속성 중 Type_Of_BasicComponent 값이 OLB 인 것으로 판단

4.5 개체노드(entity node)

- Interface로 매핑

I-Farmer 모델을 ISM 모델로 변환시키는 ITI 알고리즘의 설명을 위하여 여기서 I-Farmer Model에서 사용하는 기본적인 definition에 대하여 알아볼 필요가 있다. 먼저 개체노드에 대한 정의는 다음과 같이 구조체(structure)로서 정의할 수 있다.

【 Definition 5 】 Entity Node Structure

If E is *the set of the entity node structure*, E can be defined as follows.

$$\forall e \in E, e = \langle \text{Eid}, A, S, LT \rangle$$

where, Eid : Name of the entity node e

A : Attribute Set (Refer to **Definition 6**)

S : It is the view that e has, and it signifies the set of the aspect node name (id) (Refer to **Definition 7**)

LT : It is the loading type of the entity

$$LT \in \{ \text{Dynamic}, \text{Static}, \text{none} \} \blacksquare$$

【 Definition 6 】 Attribute Set of Entity Node Structure

The *attribute set* A of E can be defined as the following structure.

$$\forall a \in A, a = \langle \text{Aid}, AT \rangle$$

where, Aid : Attribute name

AT : Attribute type set

$$AT \in \{ \text{char}, \text{string}, \text{integer}, \text{real}, \text{boolean} \} \blacksquare$$

측면개체 노드는 다음의 구조체(structure)로서 정의할 수 있다.

【 Definition 7 】 Aspect Node Structure

If S is *the set of the aspect structure*, it can be defined as the following.

$$\forall s \in S, s = \langle \text{ASPid}, \text{OWNER} \rangle$$

where, ASPid : Name of aspect s

OWNER : Set of the entity node name that has s as the aspect \blacksquare

I-Farmer 모델에서 OLB와 ILB는 다음과 같이 정의하고 있다.

【 Definition 8 】 Structure of OM-Component Type Entity

The similarity type E_o of the entity structure set E is the set, such that,

$$\{ e \in E \mid e.LT = \text{Dynamic} \}.$$

This similarity type E_o is called *the OM-Component type entity structure*. \blacksquare

【 Definition 9 】 Structure of IM-Component Type Entity

The similarity type E_i of the entity structure set E is the set, such that,

$$\{ e \in E \mid e.LT = \text{Static} \}.$$

This similarity type E_i is called *the IM-Component type entity structure*. \blacksquare

위의 정의들을 바탕으로 I-Farmer 모델 구성요소 및 추상화 개념들을 ISM의 인터페이스 및 관계(relationship)으로 변형하는 ITI(I-Farmer model To Interface specification model) 알고리즘은 다

음과 같다.

[Algorithm] ITI(E : Entity Type Node)

// Trans-mapping I-Farmer Model to Interface Specification Model

1. WHILE End of FMD DO

// IMD : I-Farmer Model Diagram

1.1 Traverse I-Farmer Model Diagram from node E to leaf node by the BFS(Breadth First Search)

2.1 Read Current Node

2.2 CASE type_of(current_node) OF

// I-Farmer Model Diagram의 current node가 entity node type인 경우

2.2.1 Entity Node Type :

// In case former relationship of current Entity Node is multiplicity relationship, skip.

IF Former relationship of current Entity Node = multiplicity

THEN

Exit Case

END IF

// In case the next relationship of current Entity Node is multiplicity relationship, generate representative Entity Type Node.

IF the next relationship of current Entity Node = multiplicity

THEN // Generate Representative Entity Type Node

Get e.Eid // get name of entity node e

Get e.A // get attribute set of entity node e

Set interface id by using "Representative_" + e.Eid

Set e.A to interface attributes

Add new attribute to interface attributes.

Add methods to list of operations of interface

ELSE // Generate interface

Get e.Eid // get name of entity node e

Get e.A // get attribute set of entity node e

Set e.Eid to interface id

Set e.A to interface attributes

Add methods to list of operations of interface

END IF

// I-Farmer Model Diagram의 current node가 aspect entity node type인 경우

2.2.2 Aspect Entity Node Type :

// Generate aspect interface

Get a.Aid // get name of aspect node

Set a.Aid to Agent_ID

Set any values to attribute set.

// attribute set의 내용

// Number_Of_Entity, Successor_Node_Type,

// Supported_Function,

// Type_Of_Succssor_Association :

// [Decompose, Specialize, Multiplicity]

// Serviced_Protocol, Configuration_Of_Node

Add methods to list of operations of aspect interface.

// I-Farmer Model Diagram의 current node가 Uniformity Entity Node Type 인 경우

2.2.3 Uniformity Entity Node Type :

Find any interface which has the same name of current Entity Node in the interface catalog

```

IF check_if_current_node = exist THEN
    Entity_Node ← Current Uniformity
    Entity Node Type
    Call FTI(Entity_Node)
ELSE
    Exception(No_Exist)
ENDIF

```

// I-Farmer Model Diagram의 current node가
Uniformity Aspect Entity Node Type 인 경우

2.2.4 Uniformity Aspect Entity Node Type :

Find any aspect interface which has the same name of current Aspect Entity Node in the aspect interface catalog

```

IF check_if_current_node = exist THEN
    Aspect_Entity_Node ← Current
    Uniformity Entity Node Type
    Call FTI(Aspect_Entity_Node)
ELSE
    Exception(No_Exist)
ENDIF

```

// I-Farmer Model Diagram의 current node가 ILB
Multiplicity Component Type Node 인 경우

2.2.5 ILB Multiplicity Component Type Node :

// Generate interface

```

Get e.Eid // get name of entity node e
Get e.A // get attribute set of entity node e
Set e.Eid to interface id
Set e.A to interface attributes
Add new attribute Type_Of_BasicComponent &
set it to [ILB ].

```

// I-Farmer Model Diagram의 current node가
OLB Multiplicity Component Type Node 인 경우

2.2.6 OLB Multiplicity Component Type Node :

// Generate interface

```

Get e.Eid // get name of entity node e
Get e.A // get attribute set of entity node e
Set e.Eid to interface id
Set e.A to interface attributes
Add new attribute Type_Of_BasicComponent &
set it to [OLB ].

```

2.3 END CASE

2.4 Read Current Relationship between nodes

2.5 CASE type_of(Current_Relationship) OF

// I-Farmer Model Diagram의 current relation-
ship 이 decomposition relationship 인 경우

2.5.1 Decomposition :

Generate Decomposition Relationship.

// I-Farmer Model Diagram의 current relation-
ship 이 specialization relationship 인 경우

2.5.2 Specialization :

Generate Specialization Relationship.

// I-Farmer Model Diagram의 current relation-
ship 이 Multiplicity relationship 인 경우

2.5.3 Multiplicity :

Generate Association Relationship.

// I-Farmer Model Diagram의 current relation-
ship 이 Multiplicity relationship 인 경우

2.5.4 Multiplicity Instance Link

// E_c : It signifies the set of the elements that have dynamic or static loading type, that is, the set of OM-Component type entity node and IM-Component entity type node

Get E_c from the I-Farmer Model Diagram.

For $\forall e \in E_c = \{ e_i \mid 1 \leq i \leq n \}$,

FOR $i = 1$ to n

// Generate interface

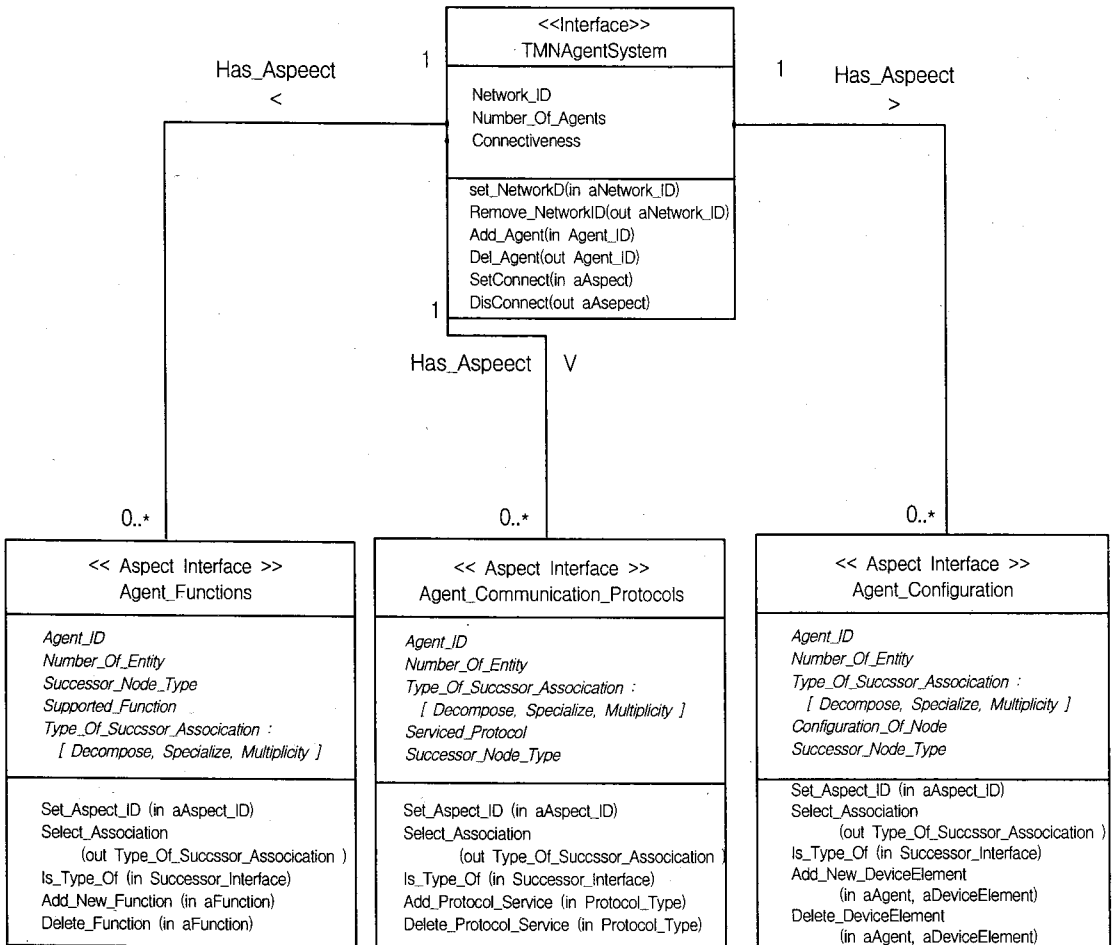
Get e_i.Eid // get name of entity node e
 Get e_i.A // get attribute set of entity
 node e
 Set e_s.Eid to interface id
 Set e_s.A to interface attributes

END IF
 END FOR

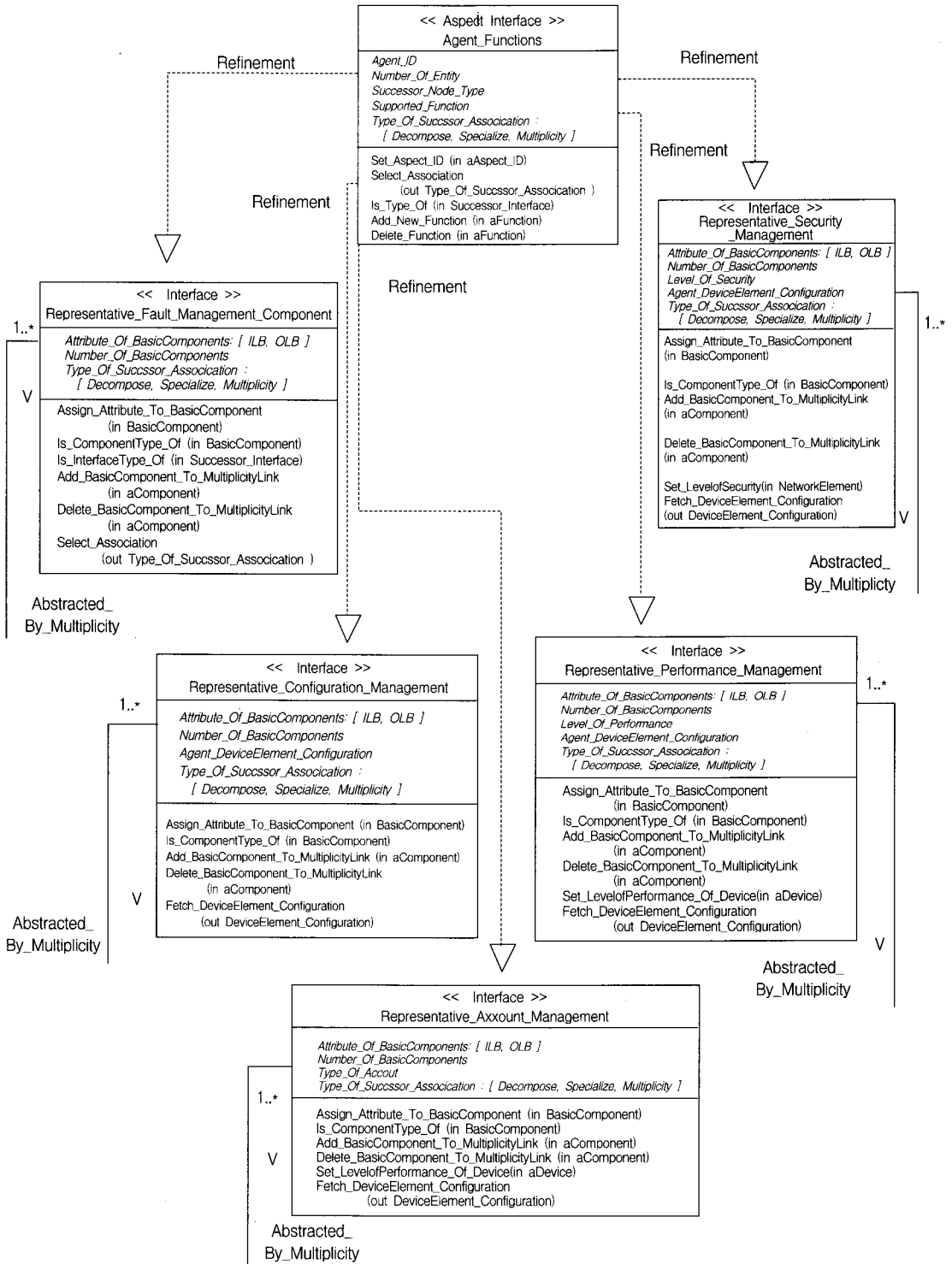
IF e_s.a.LT = Dynamic THEN
 Add new attribute Type_Of_Basic
 Component & set it to [OLB].
 ELSE
 Add new attribute Type_Of_Basic
 Component & set it to [ILB].

2.6 END CASE
 3. END WHILE

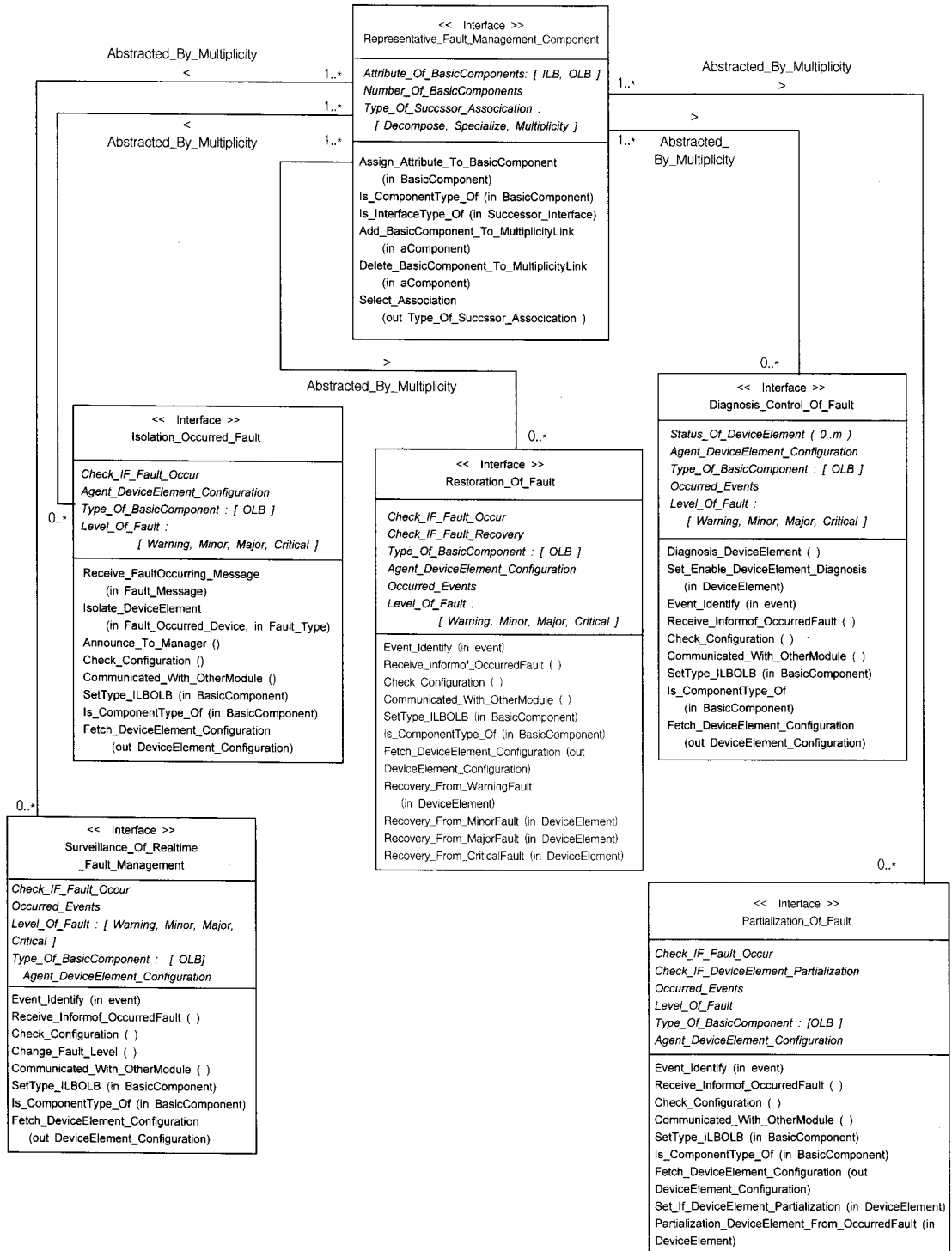
ITI 알고리즘의 correctness에 대한 증명은 trivial하다. ITI 알고리즘을 이용하여 I-Farmer 모델에 의하여 디자인된 TMN Agent 설계는 [그림 4]~[그림 7]에서 보여주는 바와 같이 인터페이스 명세모델에 의한 인터페이스 명세로 변형됨을 알 수 있다.



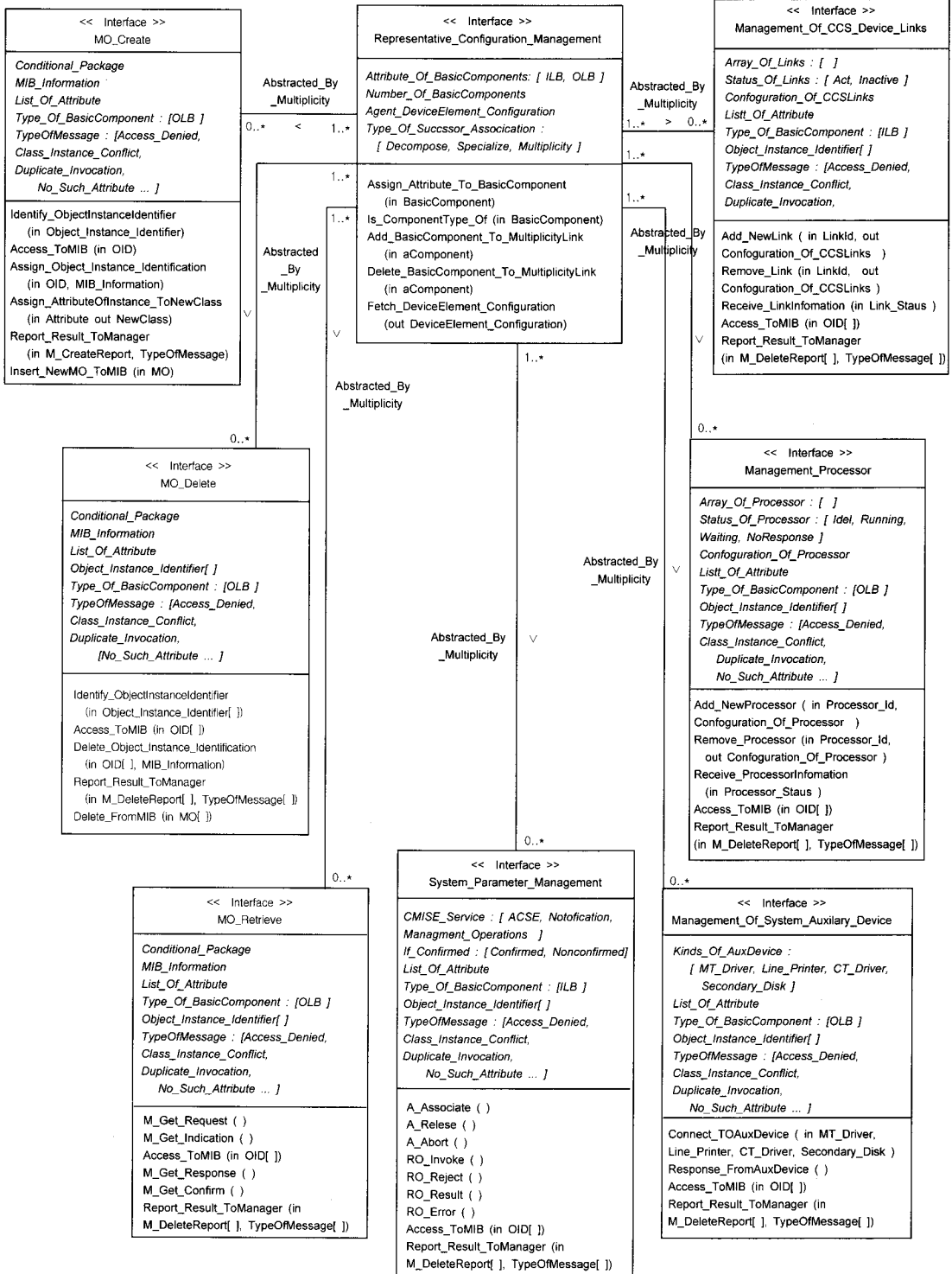
[그림 4] Aspect Interface



[그림 5] I-Farmer 모델에 의한 TMN 에이전트 모델의 인터페이스 명세모델로의 매칭(1)



[그림 6] I-Farmer 모델에 의한 TMN 에이전트 모델의 인터페이스 명세모델로의 매핑(2)



[그림 7] I-Farmer 모델에 의한 TMN 에이전트 모델의 인터페이스 명세모델로의 매핑(3)

5. NTS 알고리즘을 응용한 I-Farmer 방법론

5.1 NTS 알고리즘

NTS(Node to SIB) 알고리즘은 I-Farmer 모델로 설계한 에이전트 시스템 중 Entity Node 및 ILB/OLB 컴포넌트를 AIN의 GFP(Global Functional Plane)내의 SIB 및 시스템 응용프로그램 생성 컴포넌트인 ASIB로 변형하는 알고리즘이다.

[Algorithm] NTS (N : Node) // Node to SIB

```
// I-Farmer Model로 설계한 에이전트 시스템 중
Entity Node 및 ILB/OLB를 SIB/ASIB로
transformation 하는 알고리즘
// I-Farmer Model Diagram 중 abstraction concept
부분은 ISM 모델로의 변형 메카니즘을 따름
```

1. WHILE End of FMD DO

```
// IMD : I-Farmer Model Diagram
```

1.1 Traverse I-Farmer Model Diagram from node E to leaf node by the BFS(Breadth First Search)

1.2 Read Current Node

1.3 CASE type_of(current_node) OF

```
// If the current node of I-Farmer Model Diagram
is entity node type
```

1.3.1 Entity Node Type :

```
// In case former relationship of current Entity
Node is multiplicity relationship, skip.
```

```
IF Former relationship of current Entity Node =
multiplicity
```

```
THEN
```

```
Exception (Multiplicity)
```

```
END IF
```

```
// In case the next relationship of current Entity
```

```
Node is multiplicity relationship, generate re-
presentative Entity Type Node.
```

```
IF the next relationship of current Entity Node
≠ multiplicity
```

```
THEN
```

```
// Generate SIB
```

```
Get name of entity node e (e.Eid)
```

```
Get attribute set of entity node e (e.A)
```

```
Set name of entity node e (e.Eid) to
name of SIB (b.Sid)
```

```
According to attribute set of entity
node e,
```

```
Define service feature set (b.Sr)
```

```
Assign Service set to SIB
```

```
Construct SSL
```

```
Construct GSL
```

```
END IF
```

1.3.2 Aspect Entity Node Type :

```
Exception(Aspect Entity Node Type )
```

1.3.3 Uniformity Entity Node Type :

```
Find any interface which has the same name of
current Entity Node in the interface catalog
```

```
IF check_if_current_node = exist THEN
```

```
Entity_Node ← Current Uniformity
```

```
Entity Node Type
```

```
Call NTS(Entity_Node)
```

```
ELSE
```

```
Exception(No_Exist)
```

```
ENDIF
```

```
// I-Farmer Model Diagram의 current node가
Uniformity Aspect Entity Node Type 인 경우
```

1.3.4 Uniformity Aspect Entity Node Type

```
Exception(Uniformity Aspect Entity
Node Type)
```

```
// I-Farmer Model Diagram의 current node가
```

ILB Multiplicity Component Type Node 인 경우
1.3.5 ILB Multiplicity Component Type Node :
 OLB Multiplicity Component Type Node :
 // Generate ASIB(Applicable SIB) for ILB/OLB
 Get name of entity node e (e.Eid)
 Get attribute set of entity node e (e.A)
 Set name of entity node e (e.Eid) to name of ASIB (b.Sid)
 According to attribute set of entity node e,
 Define service feature set (b.S_f)
 Assign Service set to ASIB
 IF ASIB can be applicable to BCP
 THEN
 Construct SLP(Service Logic Program)
 Construct GSL(Global Service Logic)
 ENDIF
1.4 END CASE
1.5 Call Register_SIB(b) // b : instance of SIB or ASIB

2. END WHILE

Register_SIB 알고리즘은 신규로 생성된 SIB와 I-Farmer 모델의 entity node와 ILB/OLB의 변형체인 ASIB를 SMP내의 데이터베이스 또는 ASR에 저장하는 기능을 수행하는 알고리즘이다.

[Algorithm] Register_SIB(S : SIB/ASIB) //

Register SIB/ASIB to SMP

1. CASEOF(S)

1.1 SIB :

 Read SIB table index
 IF Can not find S in SIB table index THEN
 Register S into SMP
 ELSE
 exception(already exists)
 ENDIF

1.2 ASIB :

 Read SMP ASR Registry
 IF Can not find S in ASR (Applicable SIB Repository) Registry of SMP
 THEN Register S into SMP ASR
 ELSE
 exception(already exists)
 ENDIF

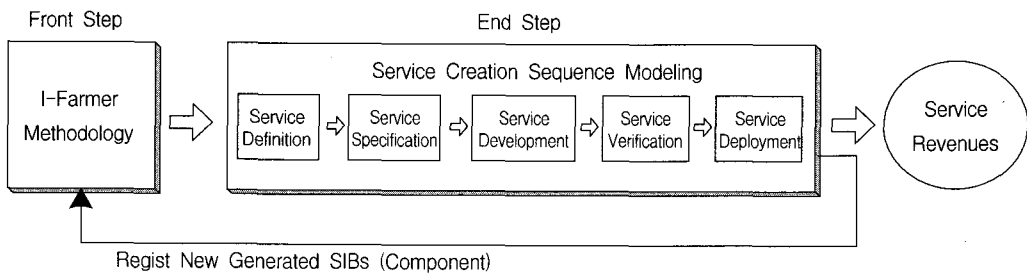
1.3 ELSE :

 exception(undefined type)

2. ENDCASE

5.2 I-Farmer 방법론

I-Farmer 방법론은 이미 정의되어 있는 SIBs (Service Independent Building Block)를 이용하여 통신망 상의 신규 서비스를 생성할 수도 있을 뿐만 아니라 기존에 정의되어 있는 SIB로는 신규 통신 서비스의 생성이 불가능할 경우 이를 위하여 새로이 만들어지는 신규 SIB를 SMP/SCEP내의 DB에



[그림 8] I-Farmer 방법론을 이용한 IMT-2000 NE/Service 구축

저장하도록 하는 기능을 포함하고 있다. 특히 신규 통신 서비스 뿐만 아니라 NE NMS Agent 등과 같은 네트워크 구성요소를 구성하는 응용프로그램을 생성하기 위하여 ASIB (Applicable SIB)의 개념을 돕으로서 SMP ASR(Applicable SIB Repository)에 저장된 ASIB를 이용하여 I-Farmer Model에 의하여 설계된 NMS agent 등의 응용 프로그램을 구성할 수 있다. [그림 8]은 I-Farmer 방법론의 기본개념을 보여주고 있다.

I-Farmer 방법론의 단계들은 다음과 같은 전단계(Front step) 후단계(End Step)으로 구분된다.

5.2.1 전단계(Front Step)

AIN의 서비스 생성 순기 모델(Service Creation Sequence Model)의 전단계를 의미하며 주로 IMT-2000 망내의 망관리 에이전트 시스템의 구축 및 NE내에서 수행되는 응용프로그램의 설계 및 구현과 관련된 단계이다.

(1) 1 단계 : 망관리 대상 시스템 범위 선정

[그림 1]에서 보여주는 바와 같이 IMT-2000 시스템의 어느 NE를 망관리 대상으로 할지에 대한 정의를 한다. 특히 PSTN, PCN, DCN 위성망 등과 같은 다른 망과 연동하여 통합 망관리 시스템(INMS : Integrated Network Management System)을 구축시 IMT-2000내의 지능망 SMP/SCEP내의 ASR과의 연동과 관련된 정의가 필요하다.

(2) 2 단계 : 분산객체 시스템 분석

- 망관리 Agent 생성을 위한 ILB/OLB 컴포넌트 추출
- Entity node/ILB/OLB 컴포넌트를 ASIB(Applicable SIB : 응용프로그램 생성용 SIB)로 변환 NTS 알고리즘 응용
- ITU-T 권고안에 의한 AIN 서비스생성을 위한 AIN CS-2 또는 CS-3 제공 basic SIB 정의

(3) 3 단계 : ASIB를 위한 ASR(Applicable SIB Repository) 구축

- SMP / SCEP에 ASR 구축
- SMP / SCEP SIB DataBase 구축

(4) 4 단계 : Agent 생성

기존의 Farming 방법론과는 달리 SMP/SCEP에서 IMT-2000내의 NE를 위한 에이전트를 생성하여 각 NE로 download

- I-Farmer 모델에 의한 Agent 설계 : ADL 생성
- ITI 알고리즘에 의한 Agent 구현

5.2.2 후 단계(End Step)

AIN 서비스 생성을 위한 단계로서 서비스 생성 순기모델(Service Creation Sequence Model)로서 불리는 단계이다.

(1) 1 단계 : 서비스 개념화(Service Conceptualization)

고객 및 시장의 요구사항을 분석하여 신규로 필요로 하는 통신 서비스를 개념화하는 단계이다.

(2) 2 단계 : 서비스 정의(Service Definition)

기존에 존재하는 재사용이 가능한 기능과 새로 구현해야하는 기능의 분류작업으로 이 단계에서는 Farming 방법론[16, 17]의 2단계의 개념을 응용한다.

사업적인 측면(business model)과 기술적인 측면을 고려하여 서비스 도입의 타당성을 상세히 분석한 후 서비스 생성 비용을 평가한다. 이때의 고려사항으로는 컴포넌트(component)의 재사용 여부를 판별하여야 하며 서비스 생성 비용 평가(service creation cost evaluation model)이 존재하는지 여부에 대한 사전조사가 필요하다.

현 단계에서 분리된 기존에 존재하는 재사용이 가능한 기능의 컴포넌트와 새로 구현해야하는 기능을 가진 컴포넌트를 구분하여 Service Specification Catalog(SSC)를 생성한다. SSC는 ASR내의 Catalog로서 존재하게 된다.

(3) 3 단계 : 서비스 명세화 (Service Specification)

논리적으로 정의된 서비스를 물리적으로 존재하

는 프로그램이나 데이터베이스로 매핑시키는 과정을 의미한다.

출력으로 서비스 개발과 검증단계에 필요한 서비스 규격과 상위구조 설계문서이다. 상위구조 설계문서에서는 ISM(Interface Specification Model)을 사용한다.

(4) 4 단계 : 서비스 개발(Service Development)

상위구조 설계문서를 상세 구조 설계문서로 변환하는 역할을 수행한다. 이를 실현하기 위하여 요구되는 서비스 소프트웨어, 데이터 정의 등을 개발하는 단계이다. 이 단계에서 사용되는 SIBs 중 재사용이 아닌 신규로 생성된 SIBs는 SMP/SCEP의 ASR에 신규로 저장되며 SSC에 catalog로서 저장된다.

(5) 5 단계 : 서비스 검증(Service Verification)

개발된 소프트웨어가 서비스 응용에 대한 규격을 완전히 만족하는지를 시험하기 위한 과정이다.

(6) 6 단계 : Service Deployment

서비스를 지원하는 모든 소프트웨어와 하드웨어를 포함한 요소들을 망에 설치하는 과정이다.

6. 결 론

AIN 및 All-IP 개념을 기반으로 하는 IMT-2000 망은 지능망의 구성요소들과 통합 및 연동을 통하여 기존에 개발되었거나 앞으로 개발될 유선 지능망의 서비스들을 쉽게 수용할 수 있게 된다. 본 논문에서는 IMT-2000 및 IMT-2000과 연동되는 다네트워크의 상이한 플랫폼 환경의 개발환경 및 향후 유지보수과정에서 발생할 것으로 예측되는 TMN 시스템 내 에이전트의 Q3 인터페이스 구현상의 표준을 이룰 수 없다는 등의 문제점을 해결하기 위하여 I-Farmer 모델 및 I-Farmer 방법론을 제안하였다. I-Farmer 모델은 향후의 TINA 기반 망관리 시스템 및 응용 시스템 개발 등을 위한 전 단계로서의 의미를 지니고 있다. 이 모델을 응용하여 I-

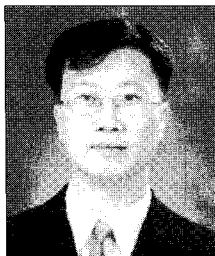
Farmer 모델에 의하여 디자인된 시스템을 인터페이스 명세모델(ISM)에서의 개념으로 변형시키는 ITI 알고리즘을 제안하였으며 I-Farmer 모델로 설계한 에이전트 시스템 중 Entity Node 및 ILB/OLB 컴포넌트를 AIN의 GFP(Global Functional Plane) 내의 SIB 및 시스템 응용프로그램 생성 컴포넌트인 ASIB로 변형하는 알고리즘인 NTS(Node to SIB) 알고리즘을 또한 제안하였다.

참 고 문 헌

- [1] 최고봉, 김기령, 김태일, 윤병남, 「지능망 기술」, 한국통신학회, 홍릉과학출판사, 1997.
- [2] Alistair Cockburn, *Surviving Object-Oriented Projects(The Agile Series for Software Developers)*, 2001.
- [3] Deepak Alur, John Crupi and Dan Malks, *Core J2EE Patterns : Best Practices and Design Strategies, SUN Micoro Systems*, 2001.
- [4] Floyd Marinescu, *EJB Design Patterns : Advanced Patterns, Processes, and Idioms, Wiley*, 2001.
- [5] ITU-T, Draft Recommendation Q.FNA, "Network Functional Model for IMT-2000," USA, Sep. 1998.
- [6] ITU-T, Draft Recommendation Q.FIF, "Information Flows for IMT-2000," Geneva, Sep. 1997.
- [7] ITU-T AIN Q.1210 - Q.1229.
- [8] ITU-T Recommendation M.3010, "Principles for a TMN," 1992.
- [9] ITU-T Recommendation M.3020, "TMN Interface Specification Methodology," 1992.
- [10] ITU-T Recommendation M.3100, "Generic Network Information Model," 1992.
- [11] James Rumbaugh, Ivar Jacobson and Grady Booch, *The Unified Modeling Language*

- Reference Manual(Addison-Wesley Object Technology Series), Addison-Wesley, 1997.*
- [12] Soo-Hyun Park, "Intelligent Farmer Model for Network Management Agent Design of IMT-2000," *to appear in Perspectives on Information Systems Development : Theory, Methods and Practice, Kluwer Academic/Plenum Publishers, U.S.A., 2002.*
- [13] Soo-Hyun Park, Sung-Gi Min, Tai-Suk Kim, "FTI Algorithm for Component Interface Meta Modeling of the TMN Agents," *Contemporary Trends in System Development, Kluwer Academic/Plenum Publishers*, Edited by W.Gregory Wojtkowski, Wita Wojtkowski, 2001, pp.129-145.
- [14] Zeigler B.P, *Multifaceted Modeling and Discrete Event Simulation, Academic Press, 1984.*
- [15] Soo-Hyun Park, "Convergence Mechanism for Transforming from Mobile Loading Block to Building Components of SMP in Advanced Intelligent Network," *In Proceeding of The 2002 International Symposium on Future Software Technology (ISFST'2002)*, Wuhan, China, 2002(Forthcoming).
- [16] Soo-Hyun Park and Doo-Kwon Baik, "Platform Independent TMN Agents Based on the Farming Methodology," *The IEICE Transactions on Fundamentals of Electronics, Communications and Computer Sciences*, The Institute of Electronics, Information and Communication Engineers (IEICE), Japan, 1998, pp.1152-1163.
- [17] Soo-Hyun Park, Doo-Kwon Baik, "The Farmer Model with the Component Farming Concept for Developing TMN Systems," *Journal of Circuits, Systems, and Computers*, World Scientific Publishing Co., Vol. 9, No.1-2(1999), pp.1-22, Singapore.
- [18] Soo-Hyun Park, "Integrating Heterogeneous Network Elements via IMT-2000 by Using ASIB of AIN," *In Proceeding of The 2001 International Symposium on Future Software Technology (ISFST'2001)*, Ghengzou, China.

◆ 저 자 소 개 ◆



박수현 (l-Farmerark21@kookmin.ac.kr)

고려대학교 컴퓨터학과에서 학사, 석사 및 박사학위를 취득하고 LG전자 중앙연구소 선임연구원, 고려대학교 컴퓨터과학기술연구소 연구원, 부산테크노파크 정보통신/소프트웨어분과 책임위원 및 동의대학교 소프트웨어공학과 조교수를 거쳐 현재 국민대학교 정보관리학부 조교수로 재직 중이다. 현 관심 분야는 액티브 네트워크에 기반한 망관리 시스템, 소프트웨어공학 등이다.