

TMN 에이전트의 컴포넌트 인터페이스 메타 모델링을 위한 변형 방법론

박수현*, 민성기*, 백두권**

* 동의대학교 컴퓨터응용공학부

** 고려대학교 컴퓨터학과 소프트웨어시스템 연구실

Transformation Methodology for Component Interface Meta Modeling of the TMN Agents

Soo-Hyun Park*, Sung-Gi Min*, Doo-Kwon Baik**

* Faculty of Computer Application Engineering, Dongeui University

** Software System Lab., Dept. of Computer Science & Engineering,
Korea University

Abstract

미리 제작되어있는 소프트웨어 컴포넌트를 조립하여 소프트웨어 개발공정을 산업화하는 과정인 CBD(Component Based Development)는 Farmer 모델에 의하여 설계된 시스템을 실제로 구현하기 위하여 매우 적합한 개념이다. CBD개념의 도입을 위하여 Farmer 모델이 갖는 개체노드, 측면노드 및 ILB/OLB 등과 같은 객체들이 수행하는 기능에 대한 명세를 나타내는 인터페이스 명세 모델이 반드시 필요하게 된다. 본 논문에서는 Farmer 모델에 의하여 정의된 개념을 컴포넌트 인터페이스 메타모델에서의 개념으로 변형시키는 방안에 대하여 설명하고 있다.

I. 서론

Farmer 모델[1][2]에 의하여 설계된 시스템을 실제로 구현하기 위하여는 개체노드, 측면노드 및 ILB/OLB 등과 같은 객체들이 수행하는 기능에 대한 명세를 나타내는 인터페이스 명세 모델(ISM : Interface Specification Model)[3][4][5]이 반드시 필요하게 된다. Farmer모델에서의 측면노드의 개념을 지원하기 위하여 ISM에서는 측면 인터페이스(Asspect Interface)의 개념을 두었으며 Aspect Interface에 대한 상세정보를 보관하기 위하여 측면 인터페이스 카탈로그(Asspect Interface Catalog)를 별도로 유지하고 있다. 균일성 개체노드(uniformity entity node)는 인터페이스 카탈로그내의 동일한 이름을 갖는 인터페이스로 매핑이 되고 균일성 측면개체노드는 측면 인터페이스 카탈로그내의 동일한 이름을 갖는 측면 인터페이스로 매핑이 이루어진다.

본 논문에서는 Farmer 모델에 의하여 정의된 개념을

컴포넌트 인터페이스 메타모델에서의 개념으로 변형시키는 방안에 대하여 설명하고 있다.

II. 컴포넌트의 구성 및 컴포넌트 / 인터페이스 모델링

CBD[3][4][5]는 미리 제작되어있는 소프트웨어 컴포넌트를 조립한다라는 개념에 기반한 소프트웨어 개발공정의 산업화과정이라고 정의할 수 있으며 다음과 같은 두 가지 기본개념에 기본을 두고 있다. 첫 번째로 응용 프로그램을 개발 시에 미리 제작된 소프트웨어 컴포넌트들을 순서에 맞추어 조립함으로써 개발속도를 크게 향상시킬 수 있다. 두 번째로 상호호환가능한 소프트웨어 컴포넌트들이 급속히 증가함에 이미 개발된 소프트웨어 컴포넌트들을 카탈로그로 만들어 놓음으로서 응용프로그램의 개발자로 하여금 쉽게 접근할 수 있도록 한다. 소프트웨어 컴포넌트는 다음과 같이 3 측면으로 볼 수 있다.

1) 명세(specification)

시맨틱(semantic)에 대해 서술. 컴포넌트의 주요 기능 및 고객이 이를 사용할 수 있는 방법 등에 대하여 서술한다. 컴포넌트의 외부 뷰(external view)를 제공

2) 구현 디자인(implementation design)

명세에서 서술한 컴포넌트의 의도를 만족시키기 위하여 소프트웨어와 데이터저장의 디자인 및 구축에 대한 방안 에 대하여 서술

3) 실행(executable)

컴포넌트 상호간에 공동으로 동작할 수 있는 플랫폼에 대하여 서술해준다. 예를 들어 철자검색 기능 컴포넌트(spell checking component)는 사전관리 컴포넌트(dictionary management component)와 상호동작 해야지만 제대로 역할을 수행할 수 있다. 이와같이 컴포넌트의 실행에 필요한 제약조건들에 대하여 서술해주는 역할을 담당한다.

명세(specification)는 컴포넌트의 최종사용자로 하여금 보다 손쉽게 빠르고 정확하게 해당 컴포넌트가 제공하는 서비스를 이해할 수 있도록 도와주는 역할을 담당한다. 명세는 자연어 형태나 형식모델(formalized model)로서 주로 서술된다. 명세는 다음과 같은 2 부분으로 구성된다.

1) 서비스(service)

컴포넌트에 의해 제공되는 동작의 리스트(list of operations)를 제공한다.

2) 명세타입 모델(specification type model)

동작(operation)의 행위(behavior)를 보여주는 어휘(vocabulary)를 제공한다. 어휘는 각 동작들이 행위를 서술한 문장내의 명사들을 나타내는 데 이들 명사는 객체로서 나타내어진다. 이러한 객체를 어휘라 한다.

명세는 인터페이스들로서 구성되는 데 인터페이스는 컴포넌트에 의해 제공되는 서비스(동작의 리스트)들의 클러스터들이라 말할 수 있다.

컴포넌트는 하나이상의 인터페이스로서 구현될 수 있으며 동일한 인터페이스가 다른 여러 컴포넌트에 의하여 지원될 수 있다.

II. Farmer 모델

Farmer 모델[1][2]의 주요 목적은 실제 디자인 하고자 하는 에이전트를 분석하여 에이전트를 구성하고 있는 컴포넌트 요소들을 측면에 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 PICR[1]에 저장이 된다. 또한 Farmer 모델은 플랫폼 독립형 클래스저장소(PICR)에서 컴포넌트 요소를 에이전트로 동적 또는 정적으로 다운로드하는 Farming[1]의 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 컴포넌트에 해당하는 속성을 지니는 IM-컴포넌트 타입 노드(IM component type node: ILB Multiplicity Component type node), OLB 컴포넌트의 속성을 지니는 OM-컴포넌트 타입 노드(OM component type node: OLB Multiplicity component type node) 등의 구성요소들과 generalization, aggregation, multiplicity 등의 3가지 추상화 개념들로 구성된다.

III. Farmer 모델에서 인터페이스 명세모델(ISM : Interface Specification Model)로의 변형(Transforming)

Farmer 모델에 의하여 설계된 시스템을 실제로 구현하기 위하여는 개체노드, 측면노드 및 ILB/OLB 등과 같은 객체들이 수행하는 기능에 대한 명세를 나타내는 인터페이스 명세 모델이 반드시 필요하게 된다. 본 절에서는 Farmer 모델에 의하여 정의된 개념을 인터페이스 명세모델에서의 개념으로 변형시키는 방안 에 대하여 설명하고 있다.

표 1은 이러한 양쪽 모델의 개념상의 매핑관계를 보여주고 있다. Farmer 모델에서의 측면노드의 개념을 지원하기 위하여 측면 인터페이스(Asspect Interface)의 개념을 두었으며 측면 인터페이스에 대한 상세정보를 보관하기 위하여 측면 인터페이스 카탈로그(Asspect Interface Catalog)를 별도로 유지하고 있다. 균일성 개체노드(uniformity entity node)는 인터페이스 카탈로그내의 동일한 이름을 갖는 인터페이스로 매핑이 되고 균일성 측면개체노드는 측면 인터페이스 카탈로그내의 동일한 이름을 갖는 측면 인터페이스로 매핑이 이루어진다. Farmer 모델에서의 다중 추상화(multiplicity abstraction)개념에 의한 대표개체노드(representative entity node)는 대표 인터페이스(Representative

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

표 1 Farmer 모델과 Interface Specification Model 모델의 Concept Mapping 관계

interface)로 변형되며 ILB 컴포넌트는 정적인터페이스 호출(Static Interface Invocation)을 지원하는 인터페이스로 매핑되는 데 속성 중 Type_Of_BasicComponent 값을 ILB로 정한다. OLB 컴포넌트의 경우 동적 인터페이스 호출(DII : Dynamic Interface Invocation)을 지원하는 인터페이스로 매핑되는 데 속성 중 Type_Of_BasicComponent 값을 OLB로 정한다.

Farmer 모델에 의하여 정의된 개념을 인터페이스 명세모델에서의 개념으로 매핑시키기 위하여 Farmer 모델의 구성요소를 기준으로 다음과 같은 사항들을 고려하여야 한다.

1) Farmer 모델에서의 Aspect entity node

- 개념매핑(Concept Mapping) 및 관리방안
 - ISM의 Aspect Interface로 매핑
 - Aspect Interface Catalog 운영
 - IFR(Interface Repository)에 AIFR (Aspect IFR) 을 별도로 유지

- 측면 인터페이스(Aspect interface)에 추가되는 기본 속성

Agent_ID
Number_Of_Entity
Successor_Node_Type
Supported_Function
Type_Of_Successor_Association :

[*Decompose, Specialize, Multiplicity*]
Serviced_Protocol
Configuration_Of_Node

- Aspect interface에 추가되는 메소드
 - Set_Aspect_ID (in aAspect_ID)*
 - Select_Association*
(out Type_Of_Successor_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 1은 이와 같은 ISM에 의하여 디자인된 측면 인터페이스(aspect interface)의 예를 보여주고 있으며 이는 다음과 같은 CORBA IDL[7]로 프리프로세싱된다.

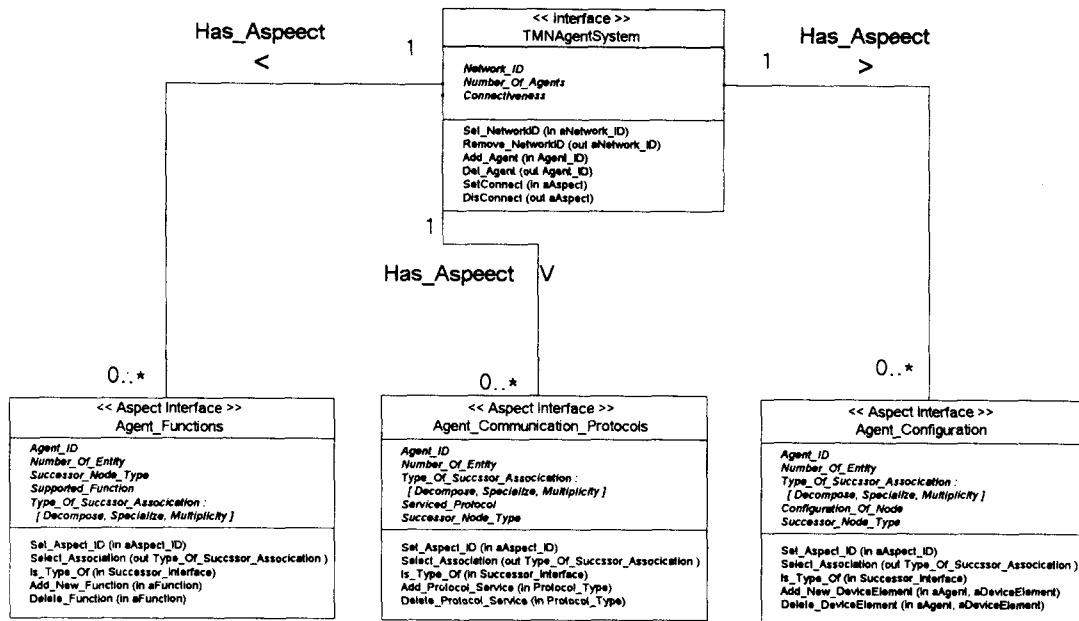


그림 1 ISM에 의하여 디자인된 Farmer 모델의 Aspect 개체

- IDL Description

// 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_Communication_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) }
```

2) Farmer 모델에서의 Multiplicity abstraction에 의한 representative entity node

■ 개념매핑(Concept Mapping) 및 관리방안

- Interface로 매핑, interface 이름을 representative_NODE_ID로 정함

■ 측면 인터페이스(Aspect interface)에 추가되는 기본 속성

```
Attribute_Of_BasicComponents:[ILB,OLB ]
Number_Of_BasicComponents
Type_Of_Succssor_Association :
    [ Decompose, Specialize, Multiplicity ]
```

■ Aspect interface에 추가되는 메소드

```
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_Assocication)

3) Farmer 모델에서의 ILB

- Static Interface Invocation
- Interface로 매핑
- 속성 중 Type_Of_BasicComponent값이 ILB인 것으로 판단

4) OLB

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

5) 개체노드(entity node)

- Interface로 매핑

IV. 결론

Farmer 모델은 TMN 에이전트 시스템과 같은 다중측면(multi-facet)[8]을 가진 복잡한 구조를 가진 시스템의 분석 및 설계에 매우 유용한 모델이다. Farmer 모델에 의하여 설계된 시스템을 실제로 구현하기 위하여 Farmer 모델에 의해 지원되는 독립적인 컴포넌트의 개념인 ILB와 OLB 등의 구현을 위하여 컴포넌트 기반 개발(CBD : Component Based Development)에 기초한 인터페이스 명세 모델(ISM : Interface Specification Model)의 필요성이 대두되었다. 본 논문에서는 이와 같이 Farmer 모델에 의하여 정의된 개념을 컴포넌트 인터페이스 메타모델에서의 개념으로 변형시키는 방안에 대하여 TMN 에이전트의 디자인의 예를 들어 CORBA IDL의 예와 더불어 설명하였다.

참고문헌

[1] Soo-Hyun Park, 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),

pp.1152 - 1163, Japan, 1998

[2] 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, Nos. 1 & 2, 1 - 22, Singapore, 1999

[3] Keith Short, "Component Based Development and Object Modeling", *White Paper*, Sterling Software, 1997

[4] Robert Orfali, Dan Harkey, and Jeri Edwards, *The Essential Distributed Objects, Survival Guide*, John Wiley & Sons, Canada, 1996.

[5] ComponentWare Consortium, "ComponentWare Architecture : A technical product description", *I-Kinetics, Inc*, 1995.

[6] Robert Orfali and Dan Harkey, *Client/Server Programming with JAVA and CORBA*, John Wiley & Sons, Canada, 1996.

[7] Zeigler B. P, *Multifaceted Modeling and Discrete Event Simulation*, Academic Press, 1984.