

□ 특 집 □

객체 모델링

중앙대 컴퓨터공학과 이 경 환*

● 목

차 ●

| | |
|----------------------------|-------------------------------------|
| I. 객체 모델링 방법론 | 3.3 2단계 : 객체추출과 활동의 기술 |
| 1.1 의미정의에 의한 객체 모델링 | 3.4 3단계 : 객체의 관련성 기술 |
| 1.2 객체 모델링 | 3.5 4단계 : 동적 모델링 |
| II. 여러가지 객체 모델링의 비교분석 | IV. 객체 모델링의 사례연수 |
| 2.1 Edwards | 4.1 해결하고자 하는 문제의 영역의 설명과 시스템의 목표 |
| 2.2 Page와 Jones의 이벤트 영역 분류 | 4.2 행위 기반 분석에 의한 문제 분석 |
| 2.3 Shlaer와 Mellor의 OOA | 4.3 객체, 속성 및 행위의 추출 |
| 2.4 Rumbaugh의 OMT | 4.4 모델링 카드 |
| III. 객체 모델링의 단계 | V. 결 론 |
| 3.1 기본단계 : 분석의 계획 | |
| 3.2 1단계 : 문제 분석 | |

I. 객체 모델링 방법론

소프트웨어 개발의 가장 큰 문제점은 복잡성, 요구의 변화, 그리고 개발기간의 시간적인 제약성이다. 이러한 문제 해결에 제안된 기본적인 개념으로 분해 기법과 추상화기법을 들 수 있다. 분해기법은 크고 복잡한 문제를 작고 단순한 서브프로그램들로 나누는 것이다. 이 기법의 기본 가정은 나누어진 문제는 전체 문제보다 해결하기 쉽고, 서브프로그램들을 다시 조합하면 원래 의도한 문제를 해결한다는 것이다. 추상화 기법은 다양하고 복잡한 개념을 단일한 단순 개념으로 매핑하는 과정으로 서로 다른 많은 것들을 불필요한 사항으로 무시한 채 같은 것으로 취급함으로써 복잡함을 줄이는 방법이다. 객체지향 방법론 역시 이 두 가지 개념을 바탕으로 소프트웨어를 개발한다.

객체지향 방법론은 해영역의 실체(entity)에 의미

(semantic)를 부여하고 실체를 객체로 추상화시키는 의미객체 모델링(정적모델) 방법과, 처음부터 동적모델을 고려한 객체추출을 기반으로 하는 객체 모델링 방법으로 나눌 수 있다.

1.1 의미정의에 대한 객체 모델링

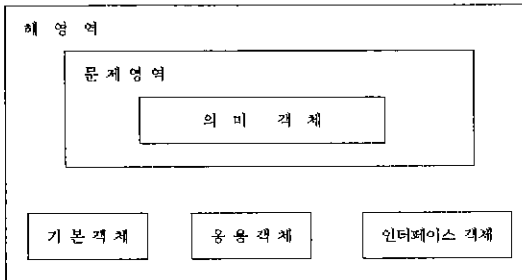
정보처리 시스템은 문제 영역(problem domain) P에 대한 해 영역(solution domain) S를 양식으로 나타낸 것으로,

PCS

인 관계로 나타난다. 실세계에서 문제의 의미를 분석하여 기술하고 분석한 양식에 의한 설계를 하여 해를 기술한다. 또 P는 의미 객체(semantic object)를 포함한 집합으로서 P의 객체들은 해보다는 문제를 기술하는 개념으로 사용한다. 상품을 주문하는 시스템의 의미 객체는 고객, 주문, 항목, 피고용자들로서

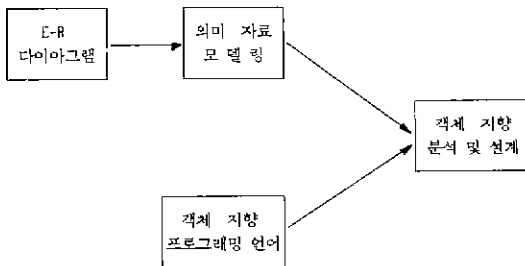
* 중신회원

주어진 시스템의 실체(entity)들이 된다. S는 문제 영역 P이외에도 인터페이스, 응용 및 기본 객체들을 포함한다. 인터페이스 객체는 사용자와 정보처리 시스템간의 인터페이스를 설명하고 문제 영역 자체를 설명하지는 않는다. 즉, 의미 객체에 대한 사용자의 관점에서 기술한다. 응용객체는 정보처리 시스템에 관한 제어 메카니즘이나 유도기능을 가지고 개발된 시스템의 기능들을 순차적으로 제어한다. 즉, 시스템의 주된 모듈로서 다른 객체들에 대한 메시지 전달이나 메뉴를 전달한다. 기본 객체는 문제 영역이나 응용되는 독립된 객체이다. 예를 들어서 string, array, 또는 디비징과 같은 기능의 루틴들이다.



(그림 1) 문제영역과 해영역

의미객체 모델을 바탕으로 해서 객체지향 분석을 하는 과정은 그림과 같이 의미자료 모델이 갖추어야 할 자료 추상화, 즉 의미 객체의 식별, 집합관계(agggregation), classification과 instantiation, 속성의 승계, 그리고, subclass와 superclass를 정의하여 수행한다.



(그림 2) 의미정의에 의한 객체 모델링

분석단계는 문제점을 기술하므로 의미 객체별 요구와 양식 중심으로 기술하고, 설계 단계는 그 해를 기술한다. 객체지향 분석은 문제 영역을 모델링하기

위하여 의미객체의 집합을 식별하여 그 양식을 정하고 시스템 요구에 의해서 관련성을 기술한다. 객체지향 설계는 해영역을 모델링하는데 의미 클래스, 응용 및 기본 클래스를 설계한다.

결국, 실체(entity)를 식별하여 의미객체로 추상화하고 객체들의 속성과 관련성, 집합관계(agggregation)를 기술하여 의미자료 모델링을 하고, 여기에 객체지향 프로그래밍 언어가 가진 특성, 즉 속성의 외부 서비스, 객체간의 인터페이스 메시지, classification 및 행위의 승계, 그리고 정보은닉과 캡슐화(encapsulation)를 추가하여 객체지향 분석 및 설계를 수행한다.

1.2 객체 모델링

객체 모델링은 문제분석과정부터 객체를 추상화시켜서 클래스로 정의하고, 관련성을 분석하여 상속성을 정의하는 분석과정이다.

1.2.1 객체

객체는 사람, 장소, 물건 등과 같이 물리적이거나 온도, 액수, 구좌 등과 같이 개념적일 수 있다. 또한, 실제 시스템의 구성요소와 일대일 대응관계를 갖는다. 그러나 객체는 개개의 요소를 의미하는 것이지 항목들의 일반적인 범주나 그룹을 의미하는 것은 아니다. 즉, 차 자체가 객체일 수 없고 각 개인이 지닌 고유 엔진 번호를 갖고 있음으로써 차가 객체일 수 있는 것이다. 다시 말해서 객체는 하나의 고유 식별자를 갖는다는 것을 의미한다.

또한 객체는 외부에 제공하는 행위를 갖는다. 객체가 제공하는 행위에 대한 기술서는 객체에 대한 작업 기술서와 비슷하다. 작업 기술서는 시스템이 어떠한 일을 해야하는가를 기술한 것이다.

행위는 객체가 제공하게 될 서비스와 수행하게 될 책임을 의미하는 것으로 외부에서 객체를 사용하는 방법과 시스템에서 주어진 일을 수행하는 방법이 된다. 임의의 조건하에서 어떠한 사건이 발생할 때 취하는 객체의 행동 파악이 필요하다. 행위는 또 다른 사건을 유발할 수도 있고 객체를 생성 또는 파괴하기도 하며 메시지의 교환일 수도 있다. 또한 객체는 상태를 갖고 이 상태는 시간에 따라 변화를 일으킨다. 상태란 객체의 현재 상황이나 전체 흐름에서의 단계 따위를 의미한다. 객체에 대한 상태를 정의하는 방법으로는 여러가지가 있을 수 있으나 가장 가능한

방법으로는 현재 가동중인 시스템인 경우 시스템 운영의 흐름내에서 객체 관찰을 통해서, 그리고 수행되고 있는 시스템이 없는 경우는 전체의 흐름을 가 정하여 파악하는 것이다.

1.2.2 객체 클래스

서로 비슷한 객체들끼리 하나의 추상화된 것으로 만들 필요가 있다. 이러한 것을 분류 작업으로 간주할 수 있다. 각 객체 클래스는 이름을 갖게 되면 이들이 포함하게될 객체의 일반적인 성질을 갖게 된다. 이는 생물학에서 비슷한 것들끼리 하나의 속으로 구분하는 것과 비교될 수 있다.

1.2.3 관련성

객체는 서로 다른 객체와 어떠한 형태로든 관련성을 갖고 있다. 여기에서 관련성이란 객체들간의 논리적 관계를 나타낸 것이다.

• 일반화 Is-A 관련성 집합: 예를 들어 ‘모든 학생은 사람이다’는 가정이 성립할 경우 ‘학생’ 객체 클래스에 속한 원소는 ‘사람’ 객체 클래스에 속함을 의미한다. 이때 학생과 사람간에는 ‘Is-A’ 관계가 성립된다.

여기에서 한 객체가 다른 객체의 상위집합인지 하위집합인지는 중요한 요인이다. 이때 상위집합을 일반화라고 하고 하위집합을 상세화라고 정의한다. 이는 집합의 개념을 판단하는 것으로 집합에 속하는 원소들을 고려하여 하위집합에 속하는 객체가 상위집합에 속할 수 있어야 한다.

• 상속: 하위 집합의 원소는 상위집합의 원소라고 정의했다. 이는 일반화가 지닌 모든 관련성 집합에 새로 정의된 상세화 역시 포함됨을 의미한다. 상세화는 일반화가 지닌 모든 관련성을 상속받는다. 예를들어 어느 회사의 ‘사원’이나 ‘고객’은 일반화인 ‘사람’의 상세화로 간주될 수 있고, ‘사람’이 ‘이름’과 ‘주소’를 갖는다는 관련성을 갖는다면 특별한 정 의없이도 ‘사원’과 ‘고객’은 ‘이름’, ‘주소’를 갖는다는 관련성을 갖는다. 어떤 경우에는 하나의 객체 클래스가 두 가지 이상의 일반화의 상세화일 수도 있다. 이때 하위집합은 여러개의 상위집합이 지닌 모든 관련성을 상속받는다. 이를 다중상속이라고 정 의한다.

• 집합관계: 한 객체가 또다른 객체를 자신의 서브파트나 부품으로 구성되는 경우 이들간에는 집합 관계가 존재한다. 이 집합관계에는 참가제한 사항이

추가된다. 예를 들어 차는 차체와 엔진, 바퀴로 구성 된다.

그러나 차는 오직 하나의 차체와 하나의 엔진, 4 개의 바퀴로 구성되고 각 차체, 엔진, 바퀴는 어느 차에 속한 부품일 수 있으나 속하게 되면 오직 하나의 차의 부품이 될 뿐이다.

• Is-Member 관련성: 하나의 객체로 취급하려는 객체와 그들 원소들간의 관계를 말한다. 학생들은 그룹을 지어 컴퓨터 동아리 등을 조직하기 위해 서로 관련을 가질 수 있다. Is-Member는 항상 이진관계 이다.

II. 여러가지 객체 모델링의 비교 분석

2.1 Edwards

분석결과에서 직접 C++로 코딩할 수 있고, 정적 및 정적차원에서 지원할 수 있으나, 승계 정의를 고려하지 않았다.

(1) 정적 차원의 지원: 클래스, 인스턴스, 관계, 기능(관계)

(2) 동적 차원의 지원: 다중 객체로 일어난 이벤트를 중심으로하여 분석하고, 단순 이벤트는 STD에 의해서 분석한다. 이벤트의 5개 영역은 다음과 같다.

- 객체 생성
- 용어 정의
- 클래스 확장
- 클래스의 재구성
- 객체의 재분류

여기에서 클래스의 확장과 재구성, 객체의 분류는 객체를 클래스로 구분하는 과정이다.

2.2 Page와 Jones의 이벤트 영역 분류

Page와 Jones는 이벤트의 영역분류를 통해서 다음과 같은 과정으로 객체를 모델링하는 과정을 설명하였다.

- (1) 객체 생성
- (2) 용어 정의
- (3) 객체 정의와 분류에 의한 클래스 정의
- (4) (3)단계와 반복 분석
- (5) 재분류

- (6) 관계의 재정의
- (7) 서로 다른 객체의 표준화와 통합

2.3 Shlaer와 Mellor의 OOA

활동(action)은 전이대신에 상태(state)와 대응하여 그 안에 들어가도록 한다. 즉, 인스턴스가 상태에 대응되었을 때 수행되는 활동상태(activity)나 연산처리 작용이다.

객체의 상호관련은 이벤트를 통해서 정의되고 이때 활동을 통해서 이벤트를 생성한다. 전이는 이벤트가 발생할 때 일어난다.

활동을 확장할 때 DFD를 채용하며, DFD는 외부 속성을 표현한다. STD에 의해서 객체의 라이프 싸이클을 결정한다.

2.4 Rumbaugh의 OMT

Rumbaugh는 활동상태(activity)와 활동(action)을 구분하여 설명하고 있다. 활동은 전이가 가능한 이벤트와 동시에 일어나고, 활동상태(activity)는 상태가 추가된 연산과정으로 생각한다. 객체의 상태는 객체가 가지고 있는 다른 객체들과의 결합으로 모은 집합이다. 활동상태는 전이가 가능한 이벤트에서 끝나고, 항상 활동으로 바뀔 수 있다. 그래서 자료의 동적 및 정적 성질을 표현하는데 중점을 두고 있다.

객체지향 분석은 다음의 4가지 과정을 반복해서 이루어진다.

- (1) 객체 모델링
- (2) 동적 모델링
- (3) 기능 모델링
- (4) 모델간의 관계 정립

특히, Booch와 Gibson은 실체와 서비스시스템을 사용하여 객체를 정의함으로써 설계과정의 단계를 상세하게 추가하고 있다. 즉, 확장가능한 객체를 서비스시스템에 정의하고 그 자료구조를 설계하고 행위의 알고리즘을 설계하면서 새로운 객체를 식별하는 기회를 갖게한다. 본 연구에서는 Rumbaugh, Booch 및 Gibson의 방법을 참조하면서 객체 모델링 방법을 설명하고자 한다.

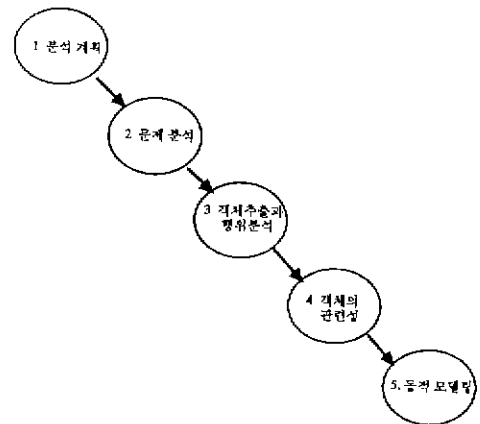
III. 객체 모델링의 단계

객체지향 분석은 시스템의 시나리오에 나타난 객

체를 식별하고 그들간의 상호작용을 파악한 다음에, 각 객체에 부과된 책임, 즉 행위를 분석한다. 객체의 행위분석은 인터뷰하기 전부터 시작할 수 있으므로 시스템의 시나리오를 문서로 작성한 이유도 객체추출을 쉽게하기 위해서이다. 여기에서 관심을 두고 보아야 할 점은 시스템을 개발하게 된 동기가 무엇이며, 동기별로 어떠한 사건이 일어날 것인가하는 점이다. 기술문서와 함께 동기를 부여하는 요구자와 요구자의 요청에 따라서 활동하는 행위자의 행위(객체의 책임, 혹은 서비스) 등은 어떤 것들이 있는가를 추출한다. 즉, 기술문서와 함께 객체와 그들의 속성 및 행위를 식별하는 것이 객체지향 분석의 중요한 과제이다.

이상과 같은 분석과정을 단계별로 나누면 다음과 같다.

- (1) 기본단계 : 분석을 위한 계획서 작성
- (2) 1 단계 : 행위에 초점을 맞춘 행위분석
- (3) 2 단계 : 객체를 추출하고 그 행위의 기술
- (4) 3 단계 : 객체들간의 관련성 파악
- (5) 4 단계 : 동적 모델링



(그림 3) 객체 모델링의 수명주기

3.1 기본단계 : 분석의 계획

시스템의 시나리오를 바탕으로 해서

- (1) 개념적 모델링(conceptual modeling)
- (2) 외부 양식(external specification)
- (3) 구현(implementation)

의 과정을 정하고 시스템의 시나리오에 나타난 목표

설정을 하고, 시스템의 범위와 문제공간의 기술, 분석을 위한 자료의 선정, 그리고 분석단계의 계획을 세운다. 목표설정에서는 인력, 자금, 시간과 같이 계획에 따라서 얻을 수 있는 자원과 품질목표 등을 세운다. 시스템의 범위와 문제공간의 기술은 시스템의 환경과 조건을 파악하고 문제분석의 공간영역이 어디까지인가를 결정한다. 분석을 위한 자원식별은 사용자의 파악과 영역을 설명할 수 있는 전문적인 지식, 그리고 적당한 참고문서 등을 찾아낸다. 이상과 같이 분석과정의 상위단계에서 나타난 행위를 설명하고 다음에 연결되는 1단계부터 4단계까지의 분석을 위한 계획을 세우고, 모델의 개념을 정립한다.

기본단계에서 기술되는 문서는,

- (1) 분석계획
- (2) 현 시스템의 목표
- (3) 요구 정의
- (4) 품질목표
- (5) 면담내용의 기술

들로서 시스템의 시나리오와 비교하여 모순점이 없는가를 평가할 수 있는 문서와 객체 모델링의 전과정에 관한 계획문서이다. 요구정의와 품질목표, 면담 등은 인터뷰와 참조문서를 바탕으로 해서 이루어지는데 다음과 같은 내용을 포함해야 된다.

- 객체들의 활동을 설명하는 시나리오
- 에러 조건과 처리방법
- 모델에 포함된 서비스
- 서비스의 수행 책임자인 행위자의 식별

시스템의 목표와 계획이 세워지면, 개념적인 모델링 단계에 들어가는 데 이 과정으로 문제분석, 객체의 추출, 그리고 객체의 관련성 정의에 들어간다. 1단계부터 4단계까지 구현되는 객체 모델링을 모델개념을 중심으로하여 설명해 보자.

3.2 1단계 : 문제분석

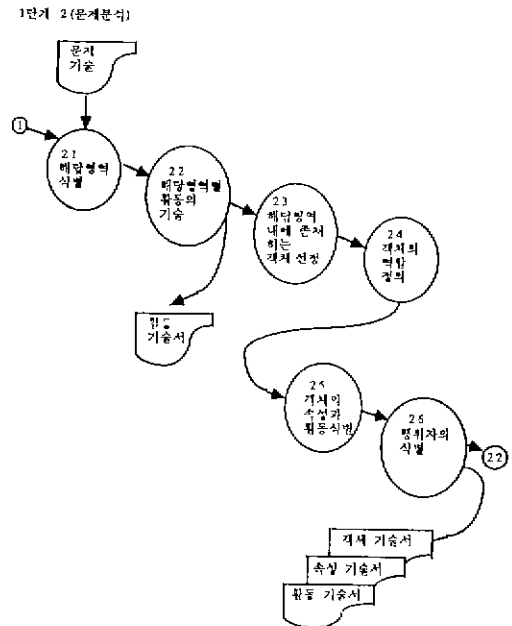
문제분석 단계에서는 시스템의 기능을 정하고, 그 기능을 누가 수행하며, 누구의 도움을 받아서 집행할 것인가를 결정한다. 시스템 기술서에 나타난 동기와 이벤트를 중심으로 하여 야기되는 문제를 파악하고 그 해당영역을 찾는다. 그리고 나서 해당영역별로 어떤 활동이 일어날 것인가를 예측하고, 활동을 할 객체를 선정한다.

선정된 객체는 어떤 역할을 담당할 수 있는가를 분석하여 그 객체의 행위와 속성을 정의한다. 끝으로 행위자의 활동을 중심으로하여 객체들의 행위를 결정하여 기술문서를 작성한다. 기술문서의 내용은 객체와 속성, 그리고 요구자의 행위와 행위자의 서비스를 기술한다. 기술문서는 사용될 용어를 충분히 이해하고, 시스템 시나리오를 상세하게 세분해 놓은 요소들로 생각할 수 있도록 설명해야 한다.

기술문서에 사용될 용어는 다음과 같다.

- 행동(action) : 시스템 안에서 객체들이 수행할 서비스와 TASK
- 서비스 : 행위자가 수행할 활동
- 요구자 : 서비스를 요청할 객체
- 협약 : 요구자와 행위자간의 계약
- 트레이스(trace) : 특정한 결과를 정당화시키기 위한 참조문서
- 참여자 : 요구자와 행위자

이상과 같은 문제분석 과정의 세부활동을 순서대로 기술하면 다음과 같다. 활동기술서는 요구자와 행위자의 객체가 수행하는 활동을 의미한다. 객체가 식별되면 요구자의 활동(initiator behavior)은 행동이라고 하고, 행위자의 활동(participant behavior)은 서비스라고 정한다.



(그림 4) 문제분석

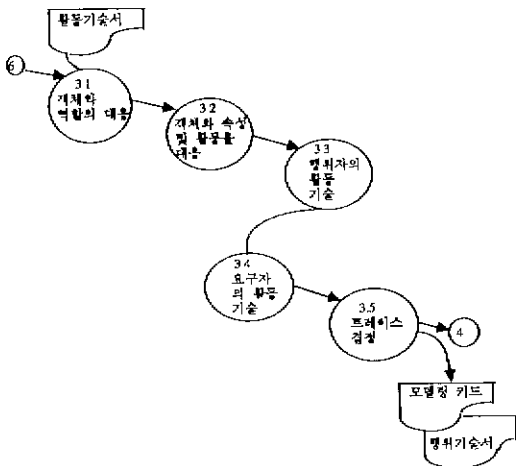
3.3 2단계 : 객체추출과 활동의 기술

문제분석에서 추출된 모든 객체의 역할을 정하고, 객체의 속성과 행위를 비교 분석하여 미비점과 모순점을 찾아낸다. 그 다음에 객체들을 구분하여 요구자인지, 행위자인지를 식별하고 각각의 행위를 기술한다. 끝으로 트레이스의 방법을 기술하면 객체의 기초적인 모델링은 끝난다.

객체의 속성은 논리적인 특성이기 때문에 객체의 활동에 관련된 성질을 정의하면 된다. 객체의 행위는 객체의 책임이 무엇인가를 파악하고, 그 객체가 제공할 서비스와 다른 객체에 요청할 서비스를 구분하여 정해 나간다. 모든 참여자가 객체로 나타나는 데 참여자는 동기를 부여하고, 서비스를 요청하는 요구자와 서비스를 수행할 행위자로 나눌 수 있다. 그러나 한 개의 객체가 요청과 수행의 두 가지 기능을 같이 가질 수도 있다. 이럴때는 객체의 기능은 두 가지로 구분해서 설명할 수 있는 행위를 정의해야 된다. 객체의 추출과 그 행위의 기술을 하는 과정을 순서대로 세분화하면 <그림 5>와 같다.

3.2의 객체는 요구(또는 제안자)와 행위자가 되는데 요구자의 활동은 능동적인 행동을 의미하고, 행위자의 활동은 수동적인 서비스를 의미한다. 본 단계에서는 활동기술서에서 행위기술서를 구별하여 설명하는 과정이 중요하다.

2단계 3(객체 추출과 행동의 기술)



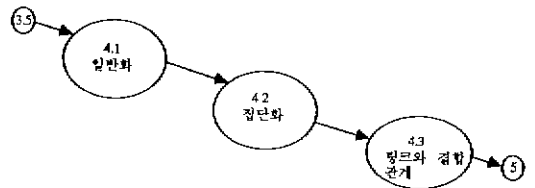
(그림 5) 객체추출과 행위기술

3.4 3단계 : 객체의 관련성 기술

객체의 관련성은 추상화(abstraction)에 의한 일반화(generalization), 특성화(specialization)에 의한 집합관계(aggregation), 그리고 링크와 결합(concatenation)에 의한 연결관계를 찾아서 객체들을 그룹으로 나누는 단계이다.

추상화는 객체에 어떤 기능을 부여할 것인지를 기술함으로써 외부적 관점에서 설명코자 하는 것으로, 구현에 의해서도 그 기능이 변하지 않는 캡슐화(encapsulation)를 시도할 수 있다. 또 공통된 특성과 행위를 특성화시켜서 다른 객체에 승계시켜줄 수 있도록 일반화시키고, 추상화된 객체를 새로운 객체로(새로운 특성과 행위를 추가시켜서) 재정의되는 특성화를 통해서 보다 차원 높은 추상화를 유도내 낼 수 있다. 즉 추상화된 객체를 일반화와 특성화를 통해서 재차 추상화시킴으로써 승계를 위한 차원높은 추상화 객체에 대해서 집합관계를 설계할 수 있다.

3단계 : 4(객체의 관련성)



(그림 6) 객체의 관련성 정의

집합관계(aggregation)은 'a-part-of'나 'part-whole'로 표현되는 객체의 그룹화를 위한 관계정의를 하는 분석과정이다. 자원관리의 BM구조와 같이 제안된 방법안에서 설계할 수 있는 DB에서 쉬운 예를 찾아볼 수 있다.

링크와 결합(link/association)을 사용자가 정의하는 내부레코드의 관계와 DB에서 parent-child 관계와 같이 정의한다. 관계의 비중을 나타내는 차수와 대응수, 그리고 종속성이 정의된다.

객체모델에서 일반화, 집합관계 및 링크와 결합관계는 매우 중요하기 때문에 이들을 각 4절, 5절 및 6절에서 상세히 설명한다.

3.5 4단계 : 동적 모델링

앞 단계에서 설명한 문제분석, 객체추출과 행위 기술, 그리고 객체들간의 관련성을 중요하게 생각하는 분석이 정적(static) 모델링이라고 한다면, 시스템 전체의 생명주기를 모델링하는 것을 동적(dynamic) 모델링이라고 할 수 있다.

동적 모델링에서는 객체의 상태(state)가 변환되어 가는 과정을 중요시한다. 객체의 상태는 기술문서에 나타난 전후조건들로부터 파악된다. 정적 모델링이 객체의 구조에만 관심을 두고, 시간조건을 배제한 환경에서 설계된다면, 동적 모델링은 시간에 따라서 변하고 시스템의 관점에서 객체를 관찰하게 된다. 따라서 두 가지 모델링의 관점은 서로간에 균형을 유지할 수 있도록 설계되어야 한다.

동적관점의 설계내용은 각 객체에 대한 상태를 정의하고 동적 행위를 가진 객체의 상태가 객체의 생명에 어떤 영향을 줄 수 있는가를 정의해야 된다. 객체의 상태는 다음과 같이 여러가지 형태로 나타날 수 있다.

- 객체가 처한 상태에 따라서 행위가 달라질 수 있다.
- 물리적인 범위, 규정 및 정책을 적용하는 동안에 달라지는 객체의 여건이나 조건을 표현한다.
- 객체의 시간적인 차이점을 규명한다.
- 시간적인 제약을 '높은 순서', 혹은 '낮은 순서'의 상태로 표현한다.
- 다른 상태나 속성에 주어지는 조건을 정의한다.
- 여러개의 객체들이 중첩되지 않는 상태에서 동시에 존재할 수 있다.

IV. 객체 모델링의 사례연구

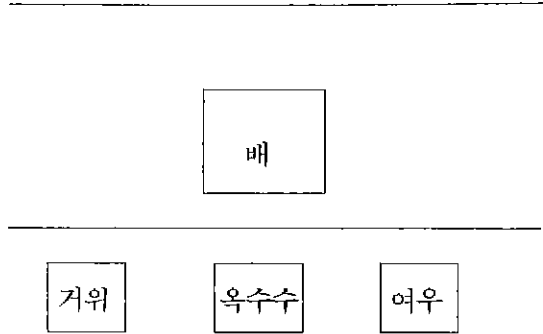
4.1 해결하고자 하는 문제영역의 설명과 시스템의 목표

주어진 문제는 우리들이 잘 알고 있는 수수께끼 문제를 택하였다. 이를 객체 모델링으로 분석하여 문제안의 객체를 추출하고 그들간의 관련성을 정의하여 객체 모델링을 한다.

문제는 <그림 7>에서 보는 것과 같이 한 강가에 있는 새 가지 실체, 거위, 옥수수, 여우를 배에 태워 다른 쪽 강가로 이동하는 문제이다. 배에 태울때는 최대 하나의 실체만을 실어 이동시킬 수 있고, 거위와

옥수수만 남겨놓을 경우, 거위는 옥수수를 먹어버리게 되어 문제가 종결되고, 거위와 여우만 남겨둘 경우도 여우가 거위를 먹어 버리게 되어 목표하는 해를 구하지 못한채 문제는 끝나게 된다. 해를 구하고 종료되는 경우를 정상적인 종료라 하고, 그렇지 않은 경우는 비정상적인 종료라고 한다.

문제의 목적은 이미 알고 있는 답만을 시뮬레이션 하는 것이 아니라 이미 어느 것도 먹히지 않고 건너게 할 수 있는 답을 우리가 알고는 있지만 이를 하나의 오락이라고 생각하고 사용자가 행하는 모든 행위에 의해 발생할 수 있는 다양한 경우를 처리하는 시스템으로 구현한다.



(그림 7) 수수께끼 문제의 도형화

4.2 행위 기반 분석에 의한 문제 분석

활동기술서를 작성하는 과정에서 택한 기준은 사용자의 시스템에 대한 초기요구 방법과 이에 따른 결과를 기준으로 하였다. 즉, 시스템에 대한 상이한 요구 방법의 활동 기술서를 작성하고, 동일한 요구 방법에 대하여도 서로 다른 결과가 일어날 수 있다면 분리된 활동기술서로 작성하였다.

본 문제를 보면 초기의 사용자 요구 방법을 찾는 것이 용이한 편이다. 왜냐하면 강을 건널 실체가 정해져 있고 시스템의 흐름은 이것들의 선정순서에 따라 달라지기 때문이다.

- ① 거위를 먼저 태운다.
- ② 여우를 먼저 태운다.
- ③ 옥수수를 먼저 태운다.

②,③의 경우는 시스템이 비정상적으로 종료한다는 결과를 내며 동일한 요구에 대해서 다른 결과가 있을 수 없고, ①의 경우는 정상적으로 종료되는 경우도

존재하지만, ②,③의 경우처럼 비정상적으로 종료될 수도 있으므로 각각의 활동기술서를 갖게 된다.

- 활동기술서 1 : 거위를 택하여 정상적으로 종료한 경우.

◎ 활동기술서 1

| 요청자 | 활동 | 행위자 | 서비스 |
|--------|------------------|-------------|--------------|
| sailer | select | goose | move to boat |
| sailer | load to goose to | boat | take a goose |
| sailer | sail | boat | move |
| sailer | unload a goose | other side | place goose |
| sailer | sail | boat | move |
| sailer | select | fox | move to boat |
| sailer | load a fox to | boat | take a fox |
| sailer | sail | boat | move |
| sailer | unload a fox | other side | place fox |
| sailer | select | goose | move to boat |
| sailer | load a goose to | boat | take a goose |
| sailer | sail | boat | move |
| sailer | unload a goose | origin side | place goose |
| sailer | select | corn | move to boat |
| sailer | load a corn to | boat | take a corn |
| sailer | sail | boat | move |
| sailer | unload a corn | other side | place corn |
| sailer | sail | boat | move |
| sailer | select | goose | move to boat |
| sailer | load a goose to | boat | take a goose |
| sailer | sail | boat | move |
| sailer | unload a goose | other side | place goose |

- 활동기술서 2 : 거위를 택하여 비정상적으로 종료한 경우.

◎ 활동기술서 2

| 요청자 | 활동 | 행위자 | 서비스 |
|--------|-----------------|------------|--------------|
| sailer | select | goose | move to boat |
| sailer | load a goose to | boat | take a goose |
| sailer | sail | boat | move |
| sailer | unload a goose | other side | place goose |
| sailer | sail | boat | move |
| sailer | select | fox | move to boat |
| sailer | load a fox to | boat | take a fox |
| sailer | sail | boat | move |
| sailer | unload a fox | other side | place fox |
| sailer | select | goose | move to boat |
| sailer | sail | boat | move |
| fox | eat | goose | eaten |

- 활동기술서 3 : 여우를 택하여 비정상적으로 종료한 경우.

◎ 활동기술서 3

| 요청자 | 행동 | 행위자 | 서비스 |
|--------|---------------|-------------|--------------|
| sailer | select | fox | move to boat |
| sailer | load a fox to | boat | take a fox |
| sailer | sail | boat | move |
| goose | eat | corn | eaten |
| sailer | unload a fox | origin side | place fox |
| sailer | sail | boat | move |

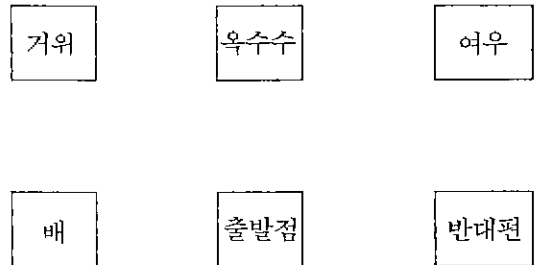
- 활동기술서 4 : 옥수수를 택하여 비정상적으로 종료한 경우.

◎ 활동기술서 4

| 요청자 | 행동 | 행위자 | 서비스 |
|--------|----------------|-------------|--------------|
| sailer | select | corn | move to boat |
| sailer | load a corn to | boat | take a corn |
| sailer | sail | boat | move |
| fox | eat | goose | eaten |
| sailer | unload a corn | origin side | place corn |
| sailer | sail | boat | move |

4.3 객체, 속성 및 행위의 추출

활동기술서에서 추출한 객체는 다음 <그림 8>과 같다.



(그림 8) 추출된 객체

이들 객체들에 대한 역할 기술서와 속성의 기술서를 작성한다.

1) 객체 역할 기술서

| 객 체 | 역 할 설 명 |
|-------------|----------------------------------|
| goose | load which eats corn |
| fox | load which eats goose |
| corn | load |
| boat | viechle to move the load |
| origin side | palce where are goose, corn, fox |
| origin side | palce where are goose, corn, fox |

2) 객체 속성 설명서

| 객 체 | 속 성 |
|-------|---------------------------------|
| goose | has direction |
| fox | has direction |
| corn | has direction |
| boat | has direction(goose, fox, corn) |
| boat | has location(goose, fox, corn) |

4.4 모델링 카드

객체에 대해 선정된 속성과 행위를 하나의 모델링 카드에 기록하여 후후 설계나 구현 단계에서 참고할 수 있게 된다.

(그림9)은 기본 모델링 카드 형태이다.

| | |
|-------------------|---------|
| (1) 모델링 카드명 | |
| <hr/> | |
| (2) 객체명 | |
| (3) 객체의 책임 한계(행위) | |
| (4) 행동 | (5) 서비스 |
| | |
| (6) 속성 | |
| (7) 트레이스 | |

(그림 9) 모델링 카드의 기본 형태

각 구성요소를 알아보자. 번호는 이해를 돕기위해

임의로 기술한 것이다.

(1) 모델링 카드의 이름을 기술한다. 이는 후후의 단계에 참조를 위한 식별자가 된다.

(2) 객체의 이름을 기술한다. 이는 행위 기술서를 바탕으로 추출된 객체 기술서에 나타난 이름을 기술한다.

(3) 객체의 책임은 (4), (5)에서 기술되는 행위를 통해 나타난다. 즉, 객체가 실제로 수행하는 기능을 나타내는 (5)는 요구를 수행하는 행위로 서비스에 해당한다. 또한 객체가 다른 객체의 기능을 요구하는 행동은 (5)에 기술한다.

(6) 객체가 지닌 속성을 기술하는 것으로 속성 설명서를 바탕으로 한다.

(7) 트레이스는 이러한 모델링 카드를 얻게된 입력을 기술하는 것으로 모델링 카드의 오류 발생시 이 경로에 기술된 문서를 참조하여 수정하게 된다.

V. 결 론

모델링 카드 작성이 끝난 다음에도 관련성 정의 단계를 확장하여 일반화와 상세화를 통해서 공통성을 분석하고 새로운 객체를 추출하고, 객체간의 결합관계를 정의하여 결합기능과 관련된 객체를 추출한다.

또 활동 기술서에 작성된 시스템의 수행과정을 바탕으로 하여 객체의 상태를 파악함으로써 객체들이 가진 조건과 제약사항을 분석한다. 각 객체들의 상태를 종합적으로 기술하는 상태변화와 이 상태에서 일어날 수 있는 사용자요구(event), 이 요구에 따라서 변화하는 객체들의 상태와 사용자에게 주어진 반응(response)을 나타내는 후행조건(postcondition)으로 작성한다.

이렇게하여, 분석설계한 객체 모델링은 모델링카드를 작성하고 외부양식으로 표현한 다음에 C++로 코딩한다. C++로 코딩된 클래스는 품질평가를 거쳐서 라이브러리 안에 부품으로 저장되고 재사용된다.

참 고 문 헌

1. 이경환, 소프트웨어 재사용시스템-CARSL0 메뉴얼, 중앙대학교, 1991.
2. 이경환, 재사용을 위한 객체 모델링, 교학사(출판예정), 1993. 4.
3. G. Booch, Object-Oriented Design with Application, Benjamin/Cummings Inc., 1991.

4. ParcPlace Systems, Object-Oriented Methodology Course Notes, ParcPlace Sys. Inc., 1992.
5. J. Rumbaugh, Object-Oriented Modeling and Design, Prentice Hall, 1991.
6. S. Shlaer, S. J. Mellor, Object Life Cycle: Modeling the World in States, Yourdon Press, 1992.
7. J. Rumbaugh, Objects in the Constitution: Enterprise Model, JOOP, Jan., 1993.
8. D. J. Chen, S. K. Hung, Interface for Reusable Software Components, JOOP, Jan., 1993.



이 경 환

1980 중앙대 대학원 수학과 응용
수학 전공(이학박사)
1982 미국 Aurband대학에서 S/W
Engineering에 관하여 연구
1986 서독 Bonn대학 Institute
fuer Informatik과 공동 연구
1993 현재 중앙대 컴퓨터 공학과
교수
관심 분야: 소프트웨어공학 분야,
특히 객체지향 프로그래밍,
소프트웨어 재사용에 관하여
연구중
