

# 객체 지향과 수자원 관리 시스템 개발

## Water Resources Management System and Object-Oriented Approach

최진용\*  
Choi, Jin-yong

### 1. 머리말

컴퓨터가 발명된 이래로 이를 운영할 수 있는 프로그램(OS, Operating System, 예: DOS, UNIX, Windows 98, LINUX 등)과 이를 바탕으로 컴퓨터의 자원을 이용하여 사람이 하기 어려운 작업을 반복적으로 수행하는 프로그램(Application program)이 지속적으로 개발되고 발전되어 왔다. 이와 같은 발전과정에 있어 가장 큰 변화는 사용자의 확대라고 볼 수 있다. 특정인에 한정되어 사용되던 컴퓨터는 이제 유치원생부터 노인까지, 컴퓨터에 대한 지식의 유무와 관계없이, 특정 분야에 이르는 사회 모든 분야에 걸쳐, 일정지역에서 사용되는 것이 아니고 네트워크에 의하여 전세계에 이르기까지 확대된 것이다. 이는 컴퓨터의 하드웨어 발전을 근간으로 함은 두 말할 나위도 없거니와, 프로그램의 개발에 있어서 사용자와 개발자가 서로 분리되는 불특정 다수를 대상으로 하는 프로그램이 많아 졌을 뿐만이 아니라, 여러 계층의 사용자를 위하여 그래픽을 이용한 인터페이스와 통신을 위한 통신프로그램을 개발하여, 이제는 다양하고 복잡한 업무의 해결을 위해서는 반드시 컴퓨터의 힘을 빌릴 수밖에 없게 되었다.

결국 이는 프로그램의 내용 측면에서 살펴

보면 복잡하게 되고 크기가 방대해졌을 뿐 만 아니라 구현해야 하는 것이 반복적인 계산과 더불어 업무의 흐름이나 상태를 반영할 수 있어야 함을 나타내고 있다. 반면 프로그램을 개발하는 측면에서 보면 많은 사람이 동시에 하나의 프로젝트에 참여하여 프로그램을 개발하기에 이르렀다는 것을 의미한다.

이는 소프트웨어의 개발도 '계획-분석-설계-구현-보수' 라고 하는 수명주기(Life cycle)상에서 이루어져야 한다는 것이며, 이와 같은 배경을 바탕으로 보다 효율적인 소프트웨어 개발을 위해 1980년도 말에서 1990년에 이르러 Booch, Jacobson 등에 의해 새로이 제시된 소프트웨어 개발 패러다임이 바로 객체지향이다.

객체지향은 현실의 업무 또는 현상을 전산화하면서 순서와 과정을 근간으로 하여 프로그램을 개발하는 구조적 분석 기법(Structural analysis and design)과 절차적 프로그래밍(Procedural programming)에 반하여 현실의 업무나 개념, 대상 물체를 모두 객체(컴퓨터상의 가상 물체)화하여 프로그램을 구현하려는 것을 말한다.

따라서 본 소고에서는 객체지향의 탄생의 배경, 개념과 방법론에 대하여 간단히 알아보고 수자원 관리를 위한 객체지향 소프트웨어 개발 현황과 전망에 대하여 간단히 알아보고자 한다.

\* 서울대학교 농업생명과학대학

## 2. 객체지향 이론

### 가. 탄생의 배경

앞에서 언급한 바와 같이 업무 전산화와 같은 대형 소프트웨어의 개발 프로젝트는 수명주기에 의하여 이루어지게 되는데, 지금까지 사용하던 과정 중심의 소프트웨어 개발 방법은 코드의 수정, 반복이용, 유지보수에 어려움이 많았다. '과정 중심'이란 소프트웨어를 현실세계에 대한 컴퓨터에서의 구현이라고 할 때, 현실세계를 구현하는 방법을 '순서나 방법 중심으로 해석'하여 옮기는 것을 의미한다. 이러한 방법은 과정이 바뀌거나 업무나 환경의 변화를 적절히 프로그램에 반영할 수 있는 유연성이 적어 방대하고 체계적인 프로그램을 작성하기 어려울 뿐 만 아니라, 코드의 반복이용, 프로그램의 유지보수가 어려워 시스템을 지속적으로 관리하기가 사실상 불가능하다. 이와 같은 단점으로 말미암아 프로그램의 수명이 짧고 실패하는 프로젝트가 생기게 되었으며, 소프트웨어 개발 위기론이 대두되었다. 따라서 이를 극복하기 위한 대안으로 현실의 물체를 객체로 표현하는 객체지향 프로그램 언어, 객체지향 분석과 설계론이 탄생하게 된 것이다.

### 나. 객체지향의 개념

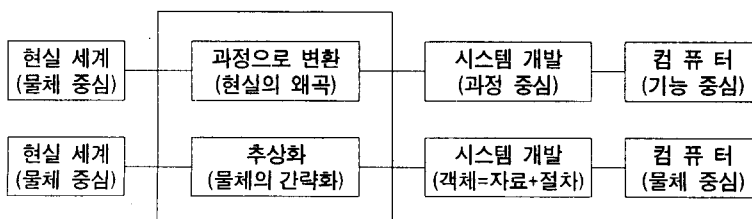
객체지향은 근본적으로 프로그램 개발에 있

어 문제의 해결을 과정 중심에서 물체 중심으로 사고하는 생각의 전환이라고 할 수 있다. 이를 바탕으로 소프트웨어의 개발을 위한 방법론으로서 객체지향 분석과 설계가 있으며, 구현을 위한 객체지향 프로그램 언어가 있다고 할 수 있다. 즉 분석과 설계단계에서 소프트웨어의 문제영역 또는 대상을 여러 개의 객체로 구분하여 각 객체를 자료(Data 또는 Properties)와 방법(Method)으로 정의한 후, 객체별로 프로그램을 개발하여 연결함으로써 각 객체에 의해 문제의 해답을 얻을 수 있도록 소프트웨어를 개발하는 기법이다. 즉 <그림 -1>에서 보는 바와 같이 객체지향은 과정중심의 기존 개발 방법과 비교할 때 현실 세계와 컴퓨터의 세계가 일치함을 볼 수 있다. 즉 현실과 구현이 같고 쉽다.

객체지향 접근은 프로그래밍의 문제만이 아니고, 시스템 개발 전체에 관련된 근본적인 문제도 해결해 주고 있다. 이를 위한 여러 가지 개념 중에서 이해를 돕기 위해 간단히 '추상화(Abstraction)', '캡슐화(Encapsulation)', '상속(Inheritance)'에 대하여 설명하기로 한다.

#### 1) 추상화(Abstraction)

현실세계의 물체를 객체에 사상(Mapping)할 때에는 현실을 그대로 객체로서 표현하는 것이 아니고, 문제의 중요한 측면, 혹은 주목하고 싶은 문제의 측면에 대한 내용을 강조하는데 이것을 추상화(Abstraction)라 한다. 이와 같이 현실세계의 추상화를 수행하는 점이 객



<그림 - 1> 과정중심과 객체지향의 프로그램 개발의 차이점

체지향의 특징중 하나이다. 이를 통하여 실제 계를 필요한 것만 간략화하여 자연스럽게 표현할 수 있는 것이다.

## 2) 캡슐화(Encapsulation)

객체지향 접근은 자료와 절차를 그룹화한 객체 단위로서 소프트웨어를 개발한다. 이때, 객체가 보유하고 있는 자료와 절차에 대해 외부에 가시성(Visibility)을 적절히 제공한다. 즉 객체 외부로부터 접근할 수 있는 것은 절차를 통해 간접적으로 자료를 접근하지만, 객체 자료에는 직접 접근할 수 없게 할 수 있다. 이 상태는 마치 자료가 캡슐로 둘러 쌓여진 상태에 비유할 수 있기 때문에 이것을 캡슐화(Encapsulation) 혹은 ‘정보은닉(Information hiding)’이라고 한다. 이와 같은 캡슐화는 객체화된 프로그램에 대하여 독립성을 유지할 수 있도록 하여 객체의 이용에 대한 혼란을 감소시켜 준다.

## 3) 상속(Inheritance)

객체지향 접근은 상위 클래스의 개념을 하위 클래스에 상속시킴과 동시에 하위 클래스에는 상위 클래스에 존재하지 않는 새로운 성질을 추가할 수 있는데, 이것은 상속(Inheritance)이라는 메카니즘을 통하여 이루어진다. 이 개념은 기존의 자원을 유효하게 재 이용해서 생산성을 향상시킬 방법을 제공하고 있다. 상속의 상위 클래스의 것을 슈퍼 클래스(Superclass), 하위 클래스의 것을 서브 클래스(Subclass)라고 부른다. 예를 들면 자동차에는 승용차, 승합차, 등으로 구분할 수 있고, 승용차에는 소형, 중형 등으로 구분할 수 있을 것이다. 이와 같은 모든 형태의 자동차를 통칭하여 자동차라고 하는 클래스로 정의할 수 있다. 이 때 슈퍼 클래스는 자동차, 서브 클래스는 승용차, 승합차이며, 소형, 중형은 또 다시 승용차의 서브 클래스가 된다. 이 때 자동차라고 하는 슈퍼 클래스

는 자동차가 가지는 일반적인 특징을 추상화해서 가지게 되며, 이러한 특징은 승용차와 승합차에게 그대로 전수된다. 이것이 상속이며, 승용차는 자동차로부터 상속받은 특징에 자신의 특징을 더하여 승용차라고 정의할 수 있다. 한편 정의된 승용차가 구체적인 자료를 입력 받아 프로그램 상에서 구현되는 것을 객체 또는 인스턴스라고 한다. 상속의 효과로서는 클래스를 체계화하는 방법으로써 유용하다는 것이며, 기존 클래스를 확장하기가 쉽다는 것이다.

## 다. 객체의 정의

일반적으로 객체라는 것은 물체와 같은 의미로 이해할 수 있다. 여기서 물체라 함은 인공물이나 움직이는 생물 등과 같은 것이다. 이와 같은 물체를 판단해 보면 이를 구분할 수 있는 모양을 가지고 있고, 어떤 목적을 위해 움직이고 있으며, 움직임에 의한 결과는 다른 물체에 직접적으로 또는 간접적으로 영향을 미치게 된다. 또한 각 물체는 각각의 특성에 독특한 행동 방식을 가지고 있으며, 이로부터 물체와 물체를 구분하게 된다. 하지만 여기서 언급하는 객체는 그것이 정형의 모양을 갖추지 않은 개념이나 수식과 같은 것도 포함한다. 즉 객체는 State(형세), Behavior(작용), Identity(정체)를 가지는 것이며, 이와 같은 것으로 다른 것과 구분되어 정의될 수 있다면 물체가 아닌 추상적인 것도 객체로 정의될 수 있다. 즉 소프트웨어로 형세와 작용, 그리고 정체를 정의하여 객체로 구현할 수 있다면 대상이 사물이거나 개념 또는 수식도 객체로 정의될 수 있는 것이다. 또한 구조와 행태가 비슷한 object를 모아서 하나의 클래스로 분류된다. 클래스의 실체를 인스턴스(Instance)와 또는 객체(Object)라고 하며 서로 같은 의미로 사용될 수 있다.(Booch, 1991)

## 라. 객체지향 개발과정과 개발 툴(CASE Tools), 언어

객체지향 소프트웨어 개발은 '프로그램의 개발 계획 수립→객체지향 분석→객체지향 설계→객체지향 프로그램→유지보수 및 관리'의 순환과정을 반복하게 된다. 이와 같은 개발과정에 있어서 계획수립, 분석과 설계에 적용되는 개념이 바로 객체지향 소프트웨어 방법론이며, 객체지향 프로그램 단계에서 이용되는 것이 바로 객체지향 언어이다. 프로젝트 규모의 프로그램 개발은 계획, 분석과 설계를 프로그램 개발자들과 프로그램의 사용자들이 서로 협의하여, 요구 분석과 문제점 분석, 객체의 추출, 객체의 설계 등을 하게 되며, 이 때 각 단계의 산출물로서 프로그램의 분석 및 설계서가 작성되게 된다. 이와 같은 산출물은 소프트웨어의 설계서로 사용되면서, 이 때 사용자와 개발자, 또는 개발자들간의 의사소통을 위한 규칙이 필요하게 된다. 이 때 사용되는 언어를 모델링 언어(Modeling language)라고 한다. 이와 같은 모델링 언어는 객체지향이론에 따라 여러 가지가 있지만, Booch에 의한 Booch 객체지향 분석과 설계, Rumbaugh에 의한 OMT 방법론, Jacobson에 의한 OOSE- (Object Oriented Software Engineering) 방법론의 통합하여 탄생한 객체지향 분석 설계 모형화 언어인 UML(Unified Modeling Language)이 대표적이다. 또한 프로그래밍 단계에서 사용되는 언어로서는 최근 많이 사용되고 있는 JAVA™과 C++ 등이 있다. 객체지향 모델링 언어와 객체지향 프로그램언어의 관계는 어떠한가. 결론적으로 서로 독립적이다. 하지만 객체지향 분석과 설계 그리고 프로그램 코딩에 이르는 일련의 과정을 보다 쉽게 도와주기 위하여 도구들이 개발되어 있으며 이를 바로 CASE (Computer Aided Software Engineering) Tool 이라고 한

다. CASE Tool은 모델링 저작도구와 다이어그램의 템플리트를 제공하여 분석과 설계의 표현을 일정한 규칙(Semantics and notations)에 따라 일목요연하게 그림으로 표현할 수 있도록 지원해 주며, 객체의 속성과 방법에 대한 코드를 작성해 주는 기능이 있어 분석과 설계, 코딩을 효율적으로 이뤄낼 수 있도록 한다.

## 3. 수자원 시스템과 객체지향

### 가. 집중물관리 시스템과 객체지향

#### 1) 객체지향의 도입 필요성

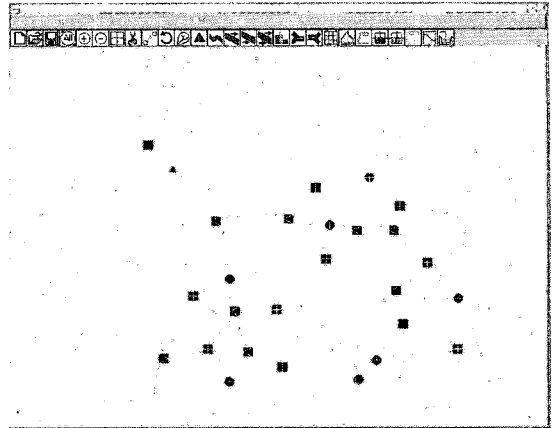
집중물관리 시스템은 관개조직을 단일의지로 운영하기 위한 시스템으로서, 이 시스템은 자료관리시스템, 모델시스템, 자료수집시스템 등으로 구성된다. 물론 집중물관리 시스템은 컴퓨터 프로그램으로 구현되는 시스템이며, 물관리 시스템이 운영하려는 대상은 당연히 관개조직이다. 관개조직은 여러 가지 물체 즉 여러 종류의 구조물과 필지 등으로 구성되어 있다. 유역과 수원공, 용배수로, 필지 등이 있으며, 수원공은 저수지, 양수장 등으로 분류가 가능하고 용배수로에는 각종 분수공과 수로 구조물이 존재한다. 또한 이와 같은 구조물들은 유기적으로 상호 연결되어 운영되고 있다.

관개조직은 변화한다. 변화한다는 것은 유지 및 개보수, 보완 등을 통하여 새로운 보조수원공이 설치되기도 하고 토공이 라이닝 수로로, 토공이 콘크리트 폴룸으로 교체되기도 하는 것을 말한다. 또한 집중물관리 시스템은 개발자와 운영자가 다르다. 변화하는 관개조직을 시스템에 반영하는 것은 운영자의 몫이라는 뜻이다. 이와 같이 다양한 구조물이 존재하며, 변화하는 관개조직을 기존의 과정 중심의 프로그래밍 방법으로 시스템을 개발하고 운영한다는 것은 개발자보다 모형이나 컴퓨터에 대한 지식이 많지

많은 사용자에게는 어려움이 많다. 이 보다는 관개조직의 여러 가지 구조물을 객체화하여 이들이 현실 세계와 같이 반응하도록 하면 개발자나 사용자가 모두 쉽게 이해할 수 있을 것이다. 개발자는 쉽게 객체를 다른 지구나 업무에 재이용할 수 있으며, 사용자는 이해가 쉬울 뿐 아니라 조직의 변화를 운영에 직접 반영할 수 있다. 여기에 객체지향 기법을 관개조직 모델링에 사용해야하는 필요성이 존재한다고 할 수 있다.

## 2) 객체지향의 적용사례

객체지향 기법을 이용하여 구성된 시스템은 성주지구 집중용수관리 시스템에서 살펴볼 수 있다. 지금까지 개발된 물 관리 프로그램은 관개조직의 변화 즉, 수로의 개보수에 따른 매개변수의 변화, 수로의 폐쇄, 보조수원공의 개발과 같은 사업의 시행 결과를 물관리 프로그램에 반영하기 위해서는 개발자가 모두 프로그램을 손봐야 하는 어려움이 있었다. 하지만 성주지구 집중용수관리 시스템에서는 관개조직 설계 프로그램(INDES-Irrigation Networks Design System)을 제시하였다. 관개조직 설계 프로그램에는 용수조직의 각종 구조물을 아이콘화하고 그래픽 에디터와 아이콘을 통하여 관개조직을 모식화한 후, 각종 자료를 입력하면 그 자료가 관개조직의 운영 모형에 입력자료로 사용될 수 있도록 되어 있어, 관개조직 모의 조작 모형에 대한 깊은 지식이 없는 운영자도 쉽게 관개조직의 변화를 입력자료로 작성할 수 있도록 하였다. 이는 또한 미리 관개조직을 변화시키기 전에 변화 후의 상황을 운영자도 쉽게 모의 조작하여 볼 수 있는 효과가 있다. <그림 -2>는 INDES의 사용자 그래픽 인터페이스(GUI-Graphic User Interface) 화면이며, 이를 이용한 관개조직의 모식도를 작성한 것을 보여 주고 있다.



<그림 - 2> INDES의 편집 화면

## 나. 수자원 관련 객체지향 프로그램

수자원관리를 위한 모의 조작모형은 지금까지 주로 FORTRAN 이나 C 언어를 사용하여 개발되어 왔다. 이러한 과정 중심의 언어는 복잡하고 반복적인 계산을 효과적으로 수행할 수 있는 장점이 있으나, 개발자 이외의 사람이 운영하기에는 어려움이 많고, 수정과 보완이 용이하지 않으며, 코드의 반복 이용이 까다롭고, 그래픽 사용자 인터페이스의 구성이나 순차적인 계산이 아닌 일상적인 업무의 구현에 특히 적용하기 어려운 단점이 있어 왔다. 수자원 분야에서 객체지향기법에 의한 객체의 개발과 이를 이용한 시스템의 구축이 이루어 지고 있으며 이중 대표적인 것이 USBR의 River Ware라고 할 수 있다. RiverWare는 USBR(US Bureau of Reclamation)과 TVA (Tennessee Valley Authority)가 지원하고 University of Colorado의 CADSWEA(Center for Advanced Decision Support for Water and Environmental System)가 개발한 수자원 전용 Object oriented software이다. RiverWare는 수자원 시스템에서 필요한 객체와 각종 관계식을 지원하며 이를 연결하여 시스템을 구성할 수 있도록 되어 있다 (김영오,1998). 이외에도 Kutija (1998)은 수치해

석 모형 개발에 객체지향 기법을 적용하였으며, John(1996)은 객체지향 모의 조작 기법을 이용하여 수문모형을 개발한 사례를 찾을 수 있다.

#### 4. 향후의 전망

현재 객체지향 기술은 객체지향 분석과 방법론에 있어서는 OMG(Object Management Group)라고 하는 컴퓨터 및 소프트웨어 개발업체의 콘소시움으로 이루어진 비영리 단체에서 표준화 작업등과 관련된 사업을 하고 있으며, 표준화에 따른 모델링 언어를 이용한 CASE Tool의 개발은 여러 기업에서 하고 있다. 그 중 대표적인 것이 Rational software사의 Rational Rose 이며, 한국 벤처기업인 PLASTIC software사의 plastic이 있다. 객체지향 언어는 마이크로 소프트(MS)사의 Visual 계열의 언어(Visual Basic, C++ 등)와 SUN사의 JAVA™ 가 서로 힘 겨루기를 하고 있으며, 보통 MS사 제품 군을 이용한 프로그램 개발에서는 Visual 계열이 강력하며, 일반적으로 플랫폼에 독립적인 프로그램 개발을 강점으로 하는 JAVA™ 는 대학을 중심으로 많이 사용되고 있는 실정이다. 이는 또한 오픈 데이터베이스의 활용과 네트워크 프로그램의 필요성이 폭발적으로 증대되면서 향후 모든 프로그램은 객체지향의 개념을 바탕으로 하는 자료구조와 프로그램의 분석과 설계, 객체지향 언어의 사용이 가일층 증대될 전망이다.

#### 5. 맺는 말

수자원관리 특히 농촌용수를 관리하기 위한 집중 용수관리 시스템은 사용자 중심으로 프로그램을 개발하여 유연하고 쉬우며, 편리하고 성능의 개선과 유지관리를 효율적으로 할 수 있는 시스템으로 개발해야 하는 것이 과제로

남아 있다. 프로그램을 개발해 본 사람은 소프트웨어도 기계나 조립식 구조물 같이 부품을 조합하여 만들 수 있었으면 하는 바람을 한번 짚은 해 보았을 것이다. 이와 같은 것이 바로 객체지향 기법에 의하여 프로그램에도 가능해질 것이다. 이제는 우리의 사고를 바꾸고 지식을 습득해야 한다. 새로운 소프트웨어의 개발 기법인 객체지향은 가장 먼저 현실세계를 객체화하여 컴퓨터 안에서 구현하는 사고의 전환을 요구하기 때문이다.

각자 개발한 객체를 공유하고 발전시켜 나가면서 다른 사람의 객체를 부품으로 나의 프로그램을 개발할 수 있음으로 해서 오는 시너지 효과를 느꼈으면 하는 바람을 다시 한번 가져본다.

#### 참고문헌

1. 김영오, 1998. 수자원관리를 위한 Object-Oriented Simulation Software들의 소개, 한국수자원학회지, 제31권 제6호, pp.21~25
2. 정하우, 1997. 성주농조 성주지구 집중 용수 관리 시스템(I), 성주농조, 농지개량조합연합회
3. Booch, Grady, 1991. Object Oriented Design with Application, The Benjamin/ Cummings Publishing Company, Inc.
4. John, A. Hinckley, Jr., 1996. Object-GAWSER, Object-Oriented Guelph All-Weather Storm-Event Runoff Model, US Army Corps of Engineers, Special Report 96-4
5. Kutija, V., 1998. Use of Object Oriented Programming in modelling of flow in open channel networks, Hydroinformatics '98, Babovic & Larsen(eds), pp. 633~640
6. Larman, Craig., 1998. Applying UML and Patterns, An introduction to Object-Oriented Analysis and Design, Prentice Hall