

일련 자료 처리를 위한 시각적 객체 기반 모델러 개발

Development of a Graphical Modeler for Manipulating Series Data Based on Object-Oriented Technique

김태곤* · 이정재**,*†

Kim, Taegon* · Lee, Jeong-Jae**,*†

ABSTRACT

Many researcher uses simulation techniques for understanding the phenomenon and expecting the response. Simulation techniques have many advantages which is less time-consuming, easy to control constraints, and cheaper than experimental study. However Researches usually are not good at programming the model, it's hard to implement simulation model using computer language. They use spreadsheet program like Microsoft excel™ Although a spreadsheet program is good for modeling, it's cumbersome to describe expressions which consist of not variable name but location indicator. This study suggests the program which helps researcher developing model using graphical interface and variable name. For verifying usability, the model which is implemented by developed modeler were compared with by spreadsheet program.

Keywords: graphical user interface; modeler; object oriented programming; simulation

1. 서 론

농공학은 자연을 대상으로 한 연구로 다양한 요인들이 복합적으로 작용하며, 그 인과관계를 명확히 하기 쉽지 않기 때문에 현장 실험만을 통해서 현상을 파악하는 것이 어렵다. 이에 많은 연구들이 현상에 대한 모델을 개발하여 모의 실험함으로써 현장 실험이 갖는 시간적, 공간적, 비용적 제약을 해결하고 있다. 모델에 사용되는 데이터는 시간의 흐름에 따라 변화양상을 모의하는 경우가 많고, 이에 따라 시계열 자료를 주 데이터로 이용하며 공간적 분포에 따라 변화 양상을 모의하는 경우에는 공간계열 자료를 이용하며, 두 가지 타입의 계열 데이터를 혼합하여 사용하는 모델 역시 다수 존재한다. 많은 수의 모델들이 이와 같이 계열 데이터를 이용하여 연산을 기본으로 수

행하고 있다.

시뮬레이션 모델을 구현하기 위해서 C, FORTRAN 등의 프로그래밍 언어를 이용하고 있으나, 모형을 개발하는데 많은 노력과 시간이 필요하고, 프로그래머 이외의 사람이 모형을 사용하고자 할 때, 프로그램의 수정 또는 보완이 용이하지 않으며, 모형의 수행 결과를 일반인에게 효과적으로 전달하기 어렵다는 단점을 갖는다 (Palmer et al., 1993).

간단한 모델의 경우, 프로그래밍 언어보다는 스프레드시트 프로그램을 이용하는 경우가 많은데, 스프레드시트 프로그램은 이용이 간단하고, 수식 표현이 쉽기 때문에 모델을 구현하는데 편리하기 때문이다 (Jones et al., 2003). 하지만 Lotus 1-2-3, 마이크로소프트의 Excel™ 등과 같은 스프레드시트 프로그램은 수식을 텍스트로 표현하도록 하였으며, 셀의 위치를 피연산자로 이용함으로써 사용자가 수식을 읽기 쉽지 않고, 작성하는 데 오류를 범하기 쉬울 뿐만 아니라 수식에 포함된 오류를 자동으로 찾아낼 수도 없다 (Smedley et al., 1996). 이는 기존 모델을 타 지역에 적용하거나 기존 모델을 파라미터만 보정해서 사용하는 경우가 많은 농공학 분야에 있어서 스프레드시트 모델의 수정 및 적용이 갖는 단점으로 판단된다.

본 연구에서는 다양한 시뮬레이션 모델에서 일반적으로 활용

* 서울대학교 생태조경·지역시스템공학부 대학원
** 서울대학교 조경·지역시스템공학부 교수, 서울대학교 농업생명과학연구원
† Corresponding author. Tel.: +82-2-880-4581
Fax: +82-2-873-2087
E-mail address: ljj@snu.ac.kr

2009년 1월 19일 투고
2009년 3월 19일 심사완료
2009년 3월 28일 게재확정

되는 일련 자료를 처리하는 데 있어서 연구자의 이해도를 높이고 오류를 줄일 수 있는 방안 및 도구로서 그래픽 모델러를 제안한다. 더불어 모델러를 구현하는 데 있어서 모델의 재사용성을 높이고, 모델러의 확장성을 도모하기 위해 객체 지향 패러다임을 도입한 객체 기반 모델러를 개발하였다. 기존 스프레드시트 프로그램으로 구현한 모델과 본 연구에서 개발한 모델러로 구현한 모델을 비교함으로써 시각적 객체 기반 모델러의 효용성에 대해 살펴본다.

II. 시뮬레이션 모델

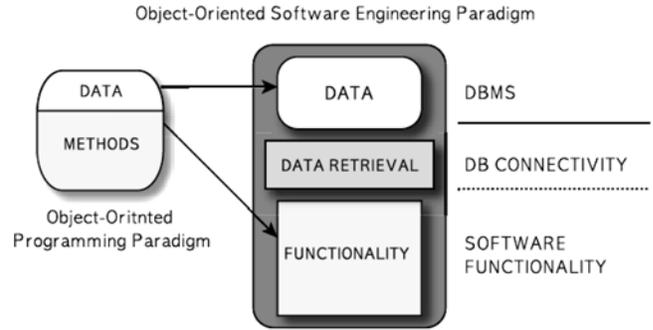
1. 시뮬레이션 모델

농공학 분야에서는 자연을 대상으로 한 다양한 연구가 진행되고 있으며, 수문학, 수리학, 관개배수학, 농업시스템공학 등의 다양한 학문들이 존재하고 있다. 이들 학문들은 자연의 현상을 관찰하고, 공학적 도구를 이용하여 분석함으로써 기작을 이해하고, 예측을 목적으로 한다. 시간의 흐름에 따른 변화를 관찰하고, 과거 데이터를 이용하여 미래를 예측하는 형태의 모델링 기법을 주로 이용하고 있다. 이러한 시뮬레이션 모델은 함수형태로 표현되며, 입력 데이터로는 시계열 데이터를 주로 이용하고 있다. 용수이용을 위하여 관개량, 담수심, 유효유량을 산정하는 데 있어서 물수지 방정식을 기본 모델로 설정하고, 일별 강수량, 증발산량, 침투량 등의 시계열 데이터를 이용한 연구가 있었으며 (Ryoo and Hwang, 2007), 논 오염부하를 구하기 위한 단순회귀모형으로서 선형방정식을 수학적식으로 구성하고, 강우량, 강수일수 등의 이벤트단위의 시계열 자료를 활용하기도 하였다 (Choi et al., 2007). 그 외에도 많은 수문 모형과 수리 모형에서 유사한 형태를 취하고 있다 (Jung et al., 2007; Kim et al., 2006; Lee et al., 2006; Yoo, 2006).

2. 객체형 소프트웨어 모듈

이 (2003)는 소프트웨어의 재사용성을 향상을 위해 통합 데이터베이스를 기반으로 한 객체형 소프트웨어 모듈 개념을 제안하였다. 이는 소프트웨어 모듈에서 사용하는 모든 데이터를 관계형 데이터베이스에 저장하고, 이 데이터를 소프트웨어의 기능과 데이터베이스 연결의 결합으로 캡슐화 하였다. 객체적 소프트웨어 모듈은 데이터의 관리와 이용을 분리함으로써 소프트웨어 모듈의 재활용성을 높일 수 있으며, Fig. 1과 같이 도출할 수 있다.

객체형 소프트웨어 모듈 스키마는 소프트웨어에 입력되는 데이터를 데이터베이스에 저장하고, 이를 이용하고 제어하는



Source : Simplifying Complexity and Emerging Self-Organization of Integrated Agricultural System Models(Hojae Yi, 2003)

Fig. 1 Concept of objective software module

데이터베이스 연결 부분을 담당하는 부분을 사용자에게 제공하여, 단일한 입출력 형태를 갖도록 구조적으로 제안한다. 이와 같이 데이터의 관리를 일원화하고, 이를 지원하는 프레임의 제공함으로써 사용자가 소프트웨어를 개발하는 데 있어서 기능 구현에 보다 집중할 수 있다. 장기적으로 소프트웨어 모듈을 개발하는 데 공통되는 데이터는 데이터베이스에서 관리되고 독립된 고유 기능 위주로 구현하도록 안내하기 때문에 소프트웨어의 모듈성이 증가될 수 있다.

본 연구에서는 객체적 소프트웨어 모듈을 적용하고 Java 언어를 통해 모델러를 구현하였다. Java는 객체적 소프트웨어 모듈 개념을 적용하는 데 있어서 다음과 같은 유익한 특징을 가지고 있다. 자바는 처음부터 객체지향 프로그래밍 언어로 개발되어 객체지향 패러다임을 구현하는 데 있어서 제약이 적으며, 데이터를 효율적으로 이용할 수 있는 데이터베이스를 활용할 경우에 객체지향 개념을 도입한 JDBC (Java Database Connectivity) 라이브러리를 이용하면 다양한 데이터베이스를 프로그램의 수정을 최소화하면서 손쉽게 활용할 수 있다. 많은 연구자들에 의해 자연 과학이나 농업 시스템 모델링을 구현하기에 뛰어난 언어로 알려져 있기도 하다 (Fox and Furmanski, 1997; Chen and Cowie, 1997; Budimlic and Kennedy, 1997; Broy et al., 1998).

III. 모델러의 구성

객체 지향 패러다임을 이용하여 모델을 구성할 때는 실세계에 존재하는 모든 사물이나 현상을 객체로 생각하고 프로그래밍 한다. 농업 시스템과 같이 복잡한 시스템을 모델링하기 위해서는 체계적으로 시스템을 분석하여 객체로 분석하고 구현해야 한다. 농업 시설물 설계에 있어서 구성요소로 분류하여 객체로 표현하거나 (Bae, 2001), 유역에서 유출을 모의하기 위

해 유역을 구성하는 요소를 객체로 치환하여 분석하기도 하였다 (Jun Wang et al., 2005). 하지만 연구자에 따라 시스템을 분석하는 방법이나 견해의 차이가 존재하기 때문에 시스템의 체계를 다양하게 정의하게 된다. 따라서 다양한 요구와 체계를 만족하기 위해서는 모델을 구성하는 객체를 보다 추상화할 필요가 존재한다. 연구자는 현상에 대해서 관찰 및 사례연구를 통해 수학적 모델을 가정하고 이를 검증하는 과정을 통하여 실 세계를 추상화하여 객체로 환원한다. 본 연구에서 개발한 모델러에서는 기존의 구조적 프로그램의 서브루틴을 객체화하여 구현할 수 있도록 함으로 하여, 기존의 구조적 프로그래밍 기법을 그대로 활용되 순서도를 그리는 것과 같이 프로그램을 작성할 수 있도록 설계하였다.

프로그램화되는 모델을 살펴보면, 데이터 입력, 데이터 처리·연산, 데이터 출력으로 크게 나눌 수 있으며, 입출력은 데이터의 저장 및 검색 기능으로 통합할 수 있다. 따라서 모델은 데이터를 가공하는 부분과 데이터를 관리하는 부분으로 나눌 수 있다. 본 연구에서는 데이터를 가공하는 부분을 알고리즘 객체로, 데이터를 관리하는 부분을 데이터 객체로 정의하고, 이 둘을 조합하여 모델을 개발할 수 있도록 하였다.

1. 데이터 객체

모델에서 데이터를 저장하고 검색하는 기능을 주로 담당하는 객체로서 모델의 입력을 담당한다. 다양한 형태의 입력 데이터를 일관된 형태로 통일시켜 줌으로써, 데이터를 처리하는 알고리즘 객체에서는 데이터의 유효성이나 적합성을 고려하지 않고 모델을 구현할 수 있도록 도와주는 역할을 수행한다. 데이터 객체는 현재 고정 값, 텍스트 파일 (Plain text), 데이터베이스, CSV (Comma Separated Value) 파일 등을 지원하며, 추가적으로 마이크로소프트의 ExcelTM 파일, XML (Extensible Markup Language) 등 다양한 형태의 파일을 지원할 수 있도록 설계하였다. 입력된 데이터는 효율적인 관리를 위하여 데이터베이스에 값을 저장하고, 데이터 객체에서는 데이터베이스와의 연결 정보를 관리함으로써 데이터베이스에서 제공하는 동시성, 무결성 등의 다양한 장점을 프로그램으로 구현하였다.

2. 알고리즘 객체

모델은 수학적 도구를 통해 분석한 결과로서 수식을 이용하여 표현하게 되는데, 알고리즘 객체는 수식을 이해하고, 계산할 수 있도록 개발하였다. 수식은 모델러에서 데이터 객체 및 알고리즘 객체에 지정한 변수명을 이용하여 표현 가능하도록 하여 사용자가 이해하기 쉽도록 구현하였다. 구현하는 데 있어

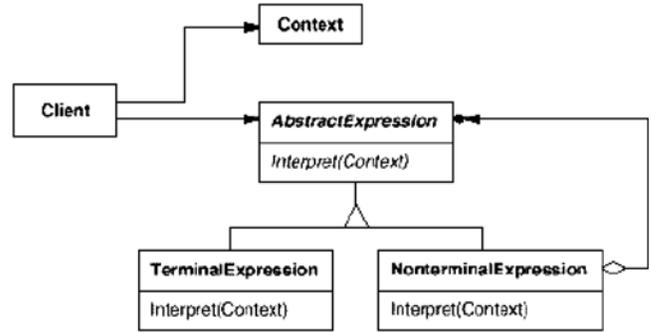


Fig. 2 Class diagram of interpreter pattern

서 수식은 프로그램이 작성되는 시점(컴파일 시점)에 결정되는 것이 아니라 모델러가 개발되어 사용되는 시점(런타임 시점)에 연구자가 입력한 수식을 해석해서 계산해야하므로 파싱이라고 불리는 수식을 해석하는 기법이 필요하다. 본 연구에서는 Fig. 2와 같이 디자인 패턴에서 제안하는 인터프리터 패턴을 활용하여 수식 해석기를 구현하였다. 인터프리터 패턴은 사용자가 입력한 수식을 분석하여, 더 작은 수식으로 나눈다. 수식은 또 다른 수식으로 구성된 비종단 표현식 (Nonterminal Expression) 과 특정 값을 갖는 종단 표현식 (Terminal Expression)으로 구성되어, 입력받은 수식을 재귀적으로 해석하여 최종적으로 종단 표현식인 데이터 객체 또는 해석이 끝난 알고리즘 객체의 값을 이용하여 해석된 값을 구해낼 수 있게 된다.

수식에서 사용되는 데이터의 이름은 연결된 데이터 객체 및 알고리즘 객체의 변수이름을 나열해줌으로써 사용자가 손쉽게 선택할 수 있도록 하였다. 수식에 사용되는 변수이름은 일련 자료를 효과적으로 표시하기 위해서 식 1과 같은 모델이 주어졌을 때, T_i 는 ' $T[0]$ '로, T_{i-1} 은 ' $T[-1]$ '로 표시하도록 설계하였으며, ' $T[0]$ '는 다시 ' $[0]$ '를 생략하고 T 로 작성할 수 있도록 구성하였다. 이에 따라 식 1을 개발된 모델러를 이용하여 수식을 작성하면 식 2와 같이 표현된다.

$$T_i = aT_{i-1} + \beta \tag{1}$$

$$T = a * T[-1] + b \tag{2}$$

기존의 구조적 프로그래밍 방식에서는 모델을 개발하는 연구자가 수식을 구현하는 과정에서 데이터를 저장하기 위한 메모리를 준비하고, 메모리에 적절하게 데이터를 저장하여야 하며, 데이터를 이용할 때 데이터 범위가 적절한지 검사하는 등의 자료 연산 과정 외의 제어 과정을 구현하여야 한다. 본 연구에서는 제어 과정을 데이터 객체로 분리함으로써 알고리즘 객체에서는 자료 연산 과정에 집중하여 구현할 수 있도록 도와준다.

3. 객체의 연결

데이터의 흐름을 시각적으로 표현함과 동시에 데이터의 유효 범위(scope)를 결정해 주기 위해서 알고리즘 객체와 알고리즘 객체 사이 또는 데이터 객체와 알고리즘 객체 사이를 선으로 연결해 주도록 모델러를 설계하였다. 객체를 화면상에 위치시키고, 이들 객체를 화살표로 연결함으로써 모델의 흐름을 시각적으로 판단할 수 있다. 연결한 선을 따라서 데이터가 이동하게 되며, 알고리즘 객체에서는 객체 쪽으로 들어오는 데이터만을 사용할 수 있게 함으로써 데이터의 유효 범위를 한정시켜주었다. 데이터의 유효 범위를 한정하는 이유는 데이터의 유효 범위가 넓어지면서 사용해서는 안 되는 곳에서 데이터를 사용하거나, 의도하지 않게 변수에 다른 값이 들어가서 프로그램이 논리적으로 오작동할 수 있기 때문이다 (Steve McConnell, 2004).

IV. 모델러의 개발

통합 데이터베이스를 이용하기 위해서 객체형 소프트웨어 모델 개념을 도입하여 모델러를 개발하였다. 모델러는 Java를 이용하여 개발하였으며, 시각적으로 표현하기 위해 스윙 (swing) 라이브러리를 이용하였다. 스윙 라이브러리는 순수하게 Java언어만으로 작성된 그래픽 라이브러리로서 기본으로 제공되는 AWT (Abstract Widget Toolkit) 라이브러리에 비해 빠른 속도와 다양한 기능을 제공하고 있으며, 한번 작성한 코드를 JVM (Java Virtual Machine)이 깔린 곳이면 어디든 쓸 수 있는 Java의 장점을 그대로 지니고 있다 (Bruce Eckel, 2003).

스윙 라이브러리를 이용하여 구현한 모델러는 Fig. 3과 같으며, 노란색의 둥근 사각형이 데이터 객체, 파란색의 둥근 사각형이 알고리즘 객체를 나타내며, 각 객체는 빨간 화살표를 통해 연결되어 있음을 확인할 수 있다. 개발된 모델러의 메뉴를 살펴보면, 새 모델 만들기, 모델 불러오기, 저장하기, 새 이름으로 저장하기, 모델러 단기로 구성된 파일 메뉴가 있으며, 모델을 구성하기 위한 메뉴로는 데이터 객체 만들기, 알고리즘 객체 만들기, 객체 연결하기 메뉴가 존재한다. 객체를 적절히 배치하고, 선으로 연결시켜 데이터의 흐름을 표시할 수 있도록 화면을 구성하였다.

데이터 객체와 알고리즘 객체에 대한 상세 정보를 모델링하기 위해서는 객체를 더블 클릭하면 객체의 속성 정보를 관리할 수 있는 다이얼로그가 나타난다. 데이터 객체에서는 Fig. 4와 같이 도시화되는 데, 어떤 데이터 형태를 처리할 것인지 결정할 수 있는 타입 선택 창과 실제 데이터의 위치 및 값을 입력할 수 있는 데이터 입력창으로 구성되며, 상단에는 모델러 내에서 사용할 데이터의 이름을 정의해 줄 수 있는 칸이 주어

져있다. 이 때 사용하는 이름은 공백이 포함되어 있지 않다면 영문자뿐만 아니라 한글도 사용 가능하도록 구현하였다. 알고리즘 객체의 속성 에디터는 Fig. 5와 같이 표시되며, 데이터 객체와 동일하게 참조할 객체의 이름을 입력해주고, 조건식에 따라 실행될 함수식을 정의함으로써 모델의 수식을 서술할 수 있도록 하였다. 수식에 이용되는 이름은 앞서 데이터객체와 다른 알고리즘 객체의 이름으로 사용된다.

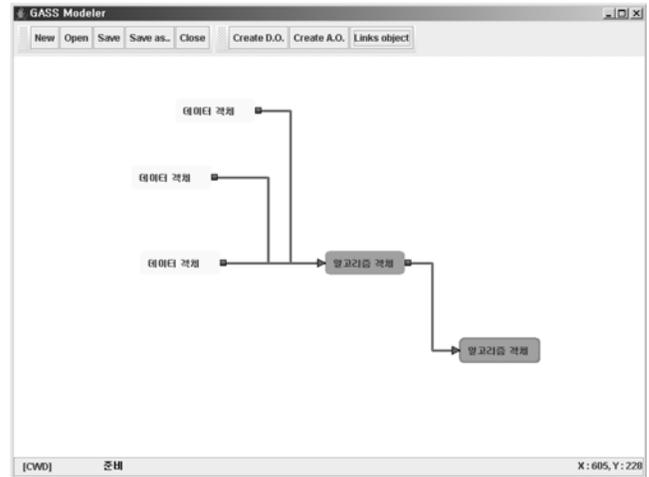


Fig. 3 Screenshot of modeler frame

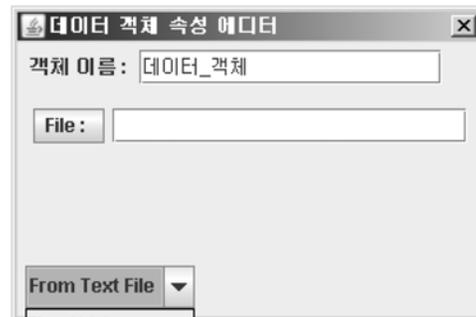


Fig. 4 Attribute editor of data objects

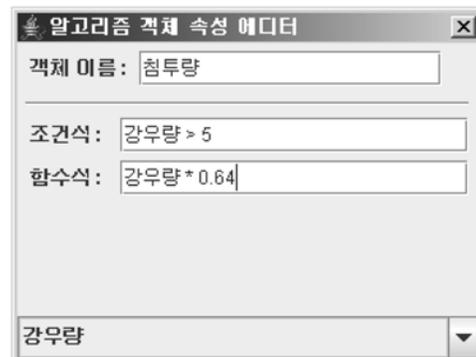


Fig. 5 Attribute editor of algorithm objects

V. 모델러의 적용

1. 잠재증발산량 모형 (FAO Modified Penman)

잠재증발산량은 관개배수 분야에서 소비수량을 계산하기 위해 가장 기본적인 데이터로서 직접 증발산량을 산정하기 어렵기 때문에 모델을 통해 간접적으로 구하여 이용한다. 보편적으로 사용되는 FAO Modified Penman법은 기온, 풍속, 습도 등의 기상 인자들을 이용하여 기준 작물인 알과파의 증발산량을 모의하여 실제 재배되고 있는 작물에 따라 작물계수를 곱하여 사용하고 있다. FAO Modified Penman법의 식은 아래 식 3과 같이 표현되며 (Allen et al., 1998), 이를 스프레드 프로그램인 마이크로소프트사의 엑셀을 이용하여 구한 결과는 Fig. 6과 같이 도시된다.

$$\lambda ET = \frac{\Delta (R_n - G) + \rho_a \cdot c_p \cdot \frac{e_s - e_a}{r_a}}{\Delta + \gamma \left[1 + \frac{r_s}{r_a} \right]} \quad (3)$$

이를 본 연구에서 개발한 모델러를 이용하여 구현한 결과는 Fig. 7과 같이 도시화되는데 동일한 모델을 구현하였기 때문에 동일한 결과를 보여줌을 확인하였으며, 데이터의 흐름을 시각화하여 보여주기 때문에 앞서 엑셀을 통해 구한 결과와 비교해 볼 때 훨씬 이해하기가 쉽다는 것을 확인할 수 있다. 또한 Table 1에서 볼 수 있듯이 모델을 구현하기 위해 사용한 모델의 수식을 살펴보면, 시트의 위치를 나타내는 엑셀에 비해, 한글 및 영문자를 이용하여 데이터의 이름을 쓴 모델러의 수식이 작성하고 이해하기에 유리한 것으로 판단된다.



Fig. 6 FAO Modified Penman method using Excel™

2. 수문학적 홍수추적 모형 (Muskingum Method)

홍수 추적은 홍수 시 발생한 유량이 하천의 임의구간을 통과하는 동안 구간내의 하도저류량의 크기 변화에 따른 통수량 변화 및 통과 시간을 밝혀내는 과정으로서 홍수에 대비하기 위한 중요한 부분이다. 다양한 홍수추적 모형 중 비교적 간단히 모형을 구성할 수 있는 Muskingum 방법은 하도저류량이 구간의 유입량과 유출량에 의해 주로 결정되는 데 착안하여 개발한 모델로서 식 4와 같이 표현된다 (Yoon, 2008).

$$Q_t = C_0 I_t + C_1 I_{t-1} + C_2 Q_{t-1} \quad (4)$$

앞서 잠재증발산량 모형과 마찬가지로 유출모형의 경우에도 엑셀과 모델러로 구현한 모델을 각각 비교하여 Fig. 8, Fig. 9와 같이 도시화하였다. Muskingum 모형 역시 두 프로그램을 이용한 결과가 똑같음을 확인하였으며, Table 2는 두 프로그램에 의해 구현된 모델식을 비교하였다. 스프레드시트 프로그램을 이용하는 경우에는 각 요소의 이름이 시트의 위치를 나타

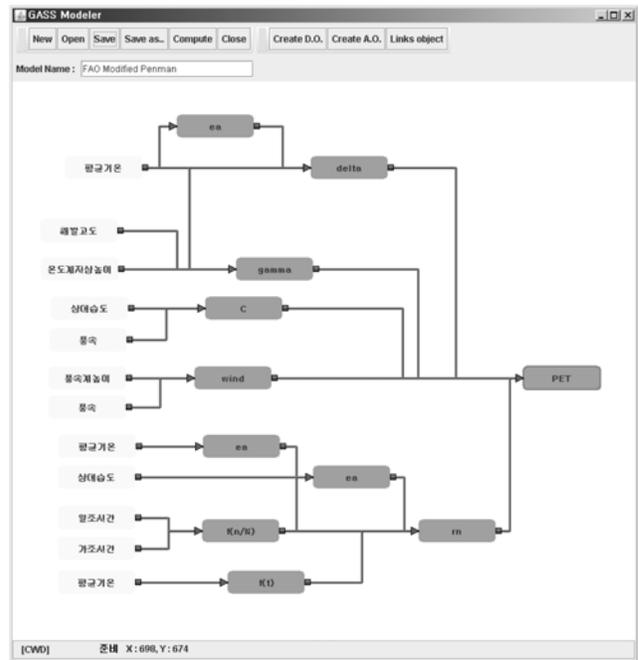


Fig. 7 FAO Modified Penman method using the modeler

Table 1 a Comparison of model expression using each modeler (Penman)

Modeler Program	Model Expression
Spreadsheet	= G9 * ((L9/(L9 + M9)) * U9 + (M9/(M9 + L9)) * O9 * (K9 - Q9))
Developed Modeler	= 보정계수 * ((포화변화율/(포화변화율 + 건습계)) * 순일사 + (건습계/(건습계 + 포화변화율)) * 풍속 * (포화수증기 - 현재방사))
Developed Modeler (using simple name)	= C * ((delta/(delta + gamma)) * rn + (gamma/(delta + gamma)) * wind * (ea - ed))

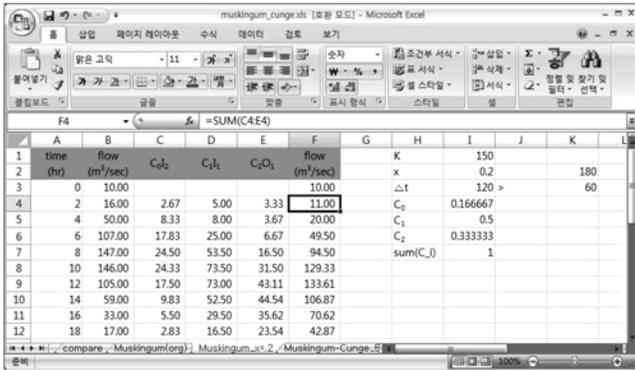


Fig. 8 Muskingum method using Excel™

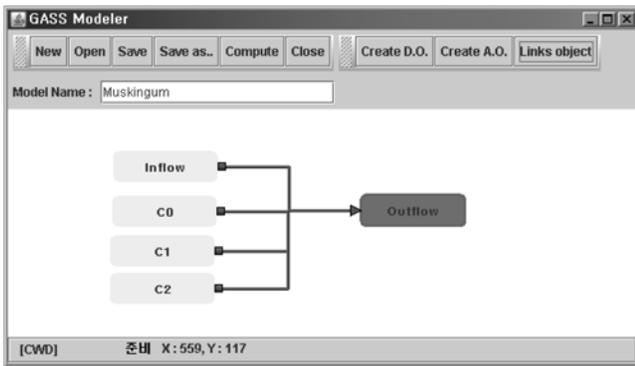


Fig. 9 Muskingum method using the modeler

Table 2 a Comparison of model expression using each modeler (Muskingum)

Modeler Program	Model Expression
Spreadsheet	= \$I\$4 * B4 + \$I\$5 * B3 + \$I\$6 * F3
Developed Modeler	= C0 * Inflow + C1 * Inflow[-1] + C2 * Outflow[-1]

내고 있기 때문에 서술한 수식만으로는 이해하기가 쉽지 않다. 엑셀 프로그램에서는 각 인자를 다양한 색으로 표현하고 각 인자가 나타내는 위치를 같은 색으로 표현함으로써 이해도를 높여려는 시도를 하였으나, 인자가 많은 경우에는 크게 효용성이 떨어진다. 반면 모델러를 이용한 표현식은 수학적식에서 사용하던 변수명과 유사하게 표현함으로써 인하여, 모델식을 프로그램 상에 그대로 표현할 수 있어 보다 수식을 읽기가 수월함을 알 수 있다.

VI. 결 론

본 연구에서는 일련의 데이터를 이용하여 모의하고자 하는 수학 모델에 대한 구현을 쉽게 하기 위해 시각적 객체 기반 모델러를 개발하였다. 잠재증발산량 모형과 홍수추적 모형을 Excel™

과 본 연구에서 개발한 모델러를 이용하여 각각 구현하여 그 사용성을 살펴보았다. 모델을 시각적으로 표현함으로써 데이터의 흐름을 쉽게 파악할 수 있게 함으로서 모델 작성의 오류를 줄이고, 기존 모형을 파악하는 데 있어서 스프레드시트에 비해 쉽게 이해할 수 있다고 판단된다. 모델 개발에 있어서 프로그래밍 언어를 따로 배우지 않고, 직관적으로 활용할 수 있어 빠르게 모델을 개발할 수 있을 것으로 기대된다.

모델러를 구현하는 데 있어서 객체형 소프트웨어 모듈 스키마를 적용하여 구현한 데이터 객체는 다양한 형태의 자료를 입력받을 수 있으며, 입력된 데이터는 데이터베이스로 저장되게 됨으로써 모델러 상에서는 동일한 형태의 데이터로 처리할 수 있도록 구현하였다. 따라서 추후 모델러에 새로운 데이터 포맷을 지원하기 위해 모델러를 개량하더라도 기존 모델은 전혀 수정 없이 모델러를 수정할 수 있을 것으로 기대된다.

제안하는 모델러가 단순히 스프레드시트 프로그램으로 해결할 수 있는 문제를 모두 대체할 수는 없겠지만, 다소 복잡한 모델을 구현하거나, 기존 모델을 활용하여 개선된 모델을 개발하는 등의 경우에 경쟁력을 지니고 있다고 판단된다. 덧붙여 단순히 나열되어 있는 일련 자료뿐만 아니라 테이블 형태의 일련 자료나 연관 관계가 있는 데이터를 이용할 수 있도록 모델러를 개선함으로써 모델러의 활용성을 더욱 높일 수 있을 것으로 기대된다. 향후 개발된 모델러는 단일 모형을 작성하는 데 그치는 것이 아니라, 통합 시뮬레이션 환경에서 활용할 수 있는 서브시스템을 개발하는 데 사용될 수 있도록 개량함으로써, 이 (2003)가 개발하고, 서 등 (2005)이 활용한 GASS와 같은 범용 시스템 시뮬레이터의 구성요소를 개발하는 도구로서 활용될 수 있을 것으로 기대된다.

REFERENCES

1. Allen, R. G., L. S. Pereira, D. Raes, and M. Smith, 1998. Crop evapotranspiration, Guidelines for computing crop water requirements. FAO Irrigation and Drainage Paper 56. Food and Agriculture Organization of the United Nations(FAO), Via delle Terme di Caracalla, 00100 Rome, Italy. ISBN: 92-5-104219-5
2. Bae, Y. J., 2001. Implementation and Improvement of the P-C Model in Intergrated irrigation Facility Design System. Master's thesis, Seoul National University. (in Korean)
3. Broy, M., A. Deimel, J. Henn, K. Koskimies, F. Plasil, G. Pomberger, W. Pree, M. Stal, and C. Szyperski,

1998. What characterizes a software component?. *Software - Concepts & Tools* 19: 49-56.
4. Bruce Eckel, 2003. *Thinking in Java 3/E*. Prentice Hall.
 5. Budimlic, Z. and K. Kennedy, 1997. Optimizing java: Theory and Practice. *Concurrency: Practice and Experience* 9(6): 445-463.
 6. Chen, M. and J. Cowie, 1997. Java's role in distributed collaboration. *Concurrency: Practice and Experience* 9(6): 509-519.
 7. Choi, Woo-Jung, Jin Hyeob Kwak, Jae Woon Jung, Kwang-Sik Yoon, Nam Ik Chang, and Yu Jeong Huh, 2007. Application of Simple Regression Models for Pollutants Load Estimation of Paddy to Yeongsan and Seomjin River Watersheds. *Journal of Korean Society of Agricultural Engineering* 49(1): 89-97 (in Korean).
 8. Fox, G. C. and W. Furmanski, 1997. Java for Parallel Computing and as a General Language for Scientific and Engineering Simulation and Modeling. *Concurrency: Practice and Experience* 9(6): 415-425.
 9. Gamma, Erich, Richard Helm, Ralph Johnson, and John Vlissides, 1995. *Design Patterns*. Addison Wesley.
 10. Jones, Simon Peyton, Alan Blackwell, and Margaret Burnett, 2003. A user-centred approach to functions in Excel. *ACM SIGPLAN Notices* 38(9): 165-176.
 11. Jun Wang, James M. Hassett, and Theodore A. Endreny, 2005. An object oriented approach to the description and simulation of watershed scale hydrologic processes. *Computers & Geosciences*. 31: 425-435.
 12. Jung, Kwang-Wook, Chun G. Yoon, Jae-Ho Jang, and Hyung-Chul Kim, 2007. Quantitative Estimation of Pollution Loading from Hwaseong Watershed using BASINS/HSPF. *Journal of Korean Society of Agricultural Engineering* 49(2): 61-74. (in Korean)
 13. Kim, Ok-kyoung, Jin-yong Choi, Min-won Jang, Seung-hwan Yoo, Won-ho Nam, Joo-heon Lee, and Jae-kyoung Noh, 2006. Watershed Scale Drought Assessment using Soil Moisture Index. *Journal of Korean Society of Agricultural Engineering* 48(6): 3-13. (in Korean)
 14. Kim, Tai Cheol, Jae Kyoung Noh, and Seung Ki Park, 1991. Daily Streamflow Model based on the Soil Water Storage in the Watershed. *Journal of Korean Society of Agricultural Engineering* 33(4): 61-72. (in Korean)
 15. Lee, D. H., 2000. A study on the structural design platform based on database. Master's thesis, Seoul National University. (in Korean)
 16. Lee, Yong-Jig, Sun Joo Kim, Phil-Shik Kim, Uk Jong Joo, and Yong Seok Yang, 2006. Study on the Effective Calculation Method of Irrigation Water in a Paddy Fields Area. *Journal of Korean Society of Agricultural Engineering* 48(3): 11-20. (in Korean)
 17. Palmer, Richard N., Allison M. Keyes, and Selene Fisher. 1993. Empowering stakeholders through simulation in water resources planning. *WATER RESOURCE PLANNING MANAGE URBAN WATER RESOURCE*. 451-454. NEW YORK, NY (USA): ASCE.
 18. Ryoo, Kyong Sik and Man Ha Hwang, 2007. Optimal Estimation of Water Use in the Large-Scale Basin. *Journal of Korean Society of Agricultural Engineering* 49(3): 3-10. (in Korean)
 19. Smedley, Trevor J., Philip T. Cox, and Shannon L. Byrne, 1996. Expanding the utility of spreadsheets through the integration of visual programming and user interface objects. *Proceedings of the workshop on Advanced visual interfaces*, 148-155.
 20. Steve McConnell, 2004. *Code Complete 2/E*. Microsoft Press.
 21. Suh, Kyo, JeongJae Lee, Taegon Kim, and Hojae Yi, 2005. System Simulation of Agricultural Marketing Based on GASS. *Journal of Korean Society of Agricultural Engineering* 47(4): 3-12. (in Korean)
 22. Yi, Ho Jae, 2003. Simplifying Complexity and Emerging Self-Organization of Integrated Agricultural System Models. Ph.D. diss., Seoul National University.
 23. Yoo, Seung-Hwan, Jin-Yong Choi, and Min-Won Jang, 2006. Estimation of Paddy Rice Crop Coefficients for FAO Penman-Monteith and Modified Penman Method. *Journal of Korean Society of Agricultural Engineering* 48(1): 13-23. (in Korean)
 24. Yoon, Yong Nam, 2008. *Hydrology*. Cheong-moon-gak. (in Korean)