

농공학 분야 프로그램들의 통합 및 재사용성 향상을 위한 방안의 모색

A Study about Alternative Program Paradigm for Improving Integration and Reusability of Agricultural Engineering Programs

김태곤, 이호재, 이정재 (서울대학교)

Abstract

Many computational models using computer program in agricultural engineering were proposed and introduced until now. These programs were developed based on procedure and structural program paradigm, so it's hard to reuse because programs become bigger and complex. In this study proposals a program paradigm that is easy to combine and reuse. Embedded engineering techniques are Java, JavaBean model, object-oriented programming.

I. 서론

1. 연구의 배경

농공학 분야의 많은 컴퓨터 프로그램들을 이용한 모델들이 포트란 기반의 절차적 방식으로 제작되어 이용되어 왔으며, 그 이후 C를 이용하여 구조적 방식으로도 많은 프로그램이 만들어졌다. 하지만 이와 같이 절차적, 구조적 방식의 프로그램은 데이터와 함수가 분리되어 있어 데이터구조가 조금만 바뀌어도 이용하는 함수가 모두 바뀌어야 하기 때문에 기존 프로그램을 수정, 보완하는 등 유지보수가 매우 어려운 단점이 존재하였다. 이에 C++, Java 등의 언어를 이용하여 객체 지향적 개발 방식이 제안되었으나, 새로운 언어 및 개발방식을 익히는 것과 기존 프로그램을 이해해야 하는 이중의 수고로 인하여 이전에 개발된 프로그램을 재이용하는 것이 쉽지가 않다. 본 연구에서는 보다 쉬운 방법으로 프로그램을 개발할 수 있으며, 데이터베이스를 이용한 자료 통합, 일방향 자료 전달을 이용한 시스템의 복잡도 해소, 추상화된 컴포넌트를 이용한 시스템의 진화성을 도모하기 위한 시스템의 구상을 목표로 한다.

2. 연구의 방향

많은 과학이 자연으로부터 문제 해결 방법을 배워왔다. 인체는 작은 세포들이 유기적으로 결합되어 구성된 매우 복잡한 시스템이다. 독립적인 세포들이 결합하여 하나의 생명체를 구성하는 것은 객체나 컴포넌트를 이용하여 소프트웨어 시스템을 구성하는 것과 유사하다. 세포는 배아줄기세포로부터 모든 세포로 발전하여 모든 시스템을 구성하게 된다. 소프트웨어에서도 배아줄기세포와 같이 추상화된 컴포넌트를 만들어 낼 수 있다면, 추상화된 컴포넌트를 이용하여 발전한 컴포넌트를 만들어 인체와 같이 복잡한 시스템을 구성할 수 있을 것으로 생각된다. 따라서 본 연구에서는 소프트웨어의 기작을 분석하여 추상화된 컴포넌트에 대해 고찰해보고자 한다.

3. 연구사

Jones(2001)에 의해 이해할 수 있는 농업 모델의 개발에 있어 모듈성의 중요성이 제시되었으며, Hatton(1998)은 객체 지향 프로그래밍 패러다임이 소프트웨어의 모듈성을 증가시키는 기술이긴 하지만 많은 경우 객체지향 언어 자체가 높은 모듈성을 제공하지는 못하여 재사용의 이익 또한 크지 않다고 하였다. Szyperski(2002)는 이전의 객체 지향 기법에서 보다 추상화된 컴포넌트를 이

용하여 시스템을 구성하는 컴포넌트 기반의 개발방식을 제안하였고, 컴포넌트는 객체와 유사하면서 보다 추상화되고 규약에 의한 외부 인터페이스를 유지함으로써, 소프트웨어 개발에 있어 재사용성을 높인다. 컴포넌트 기반의 개발 패러다임 외에 이대희(2000)은 관계형 데이터베이스를 통한 자료 통합 방식을 제안하여 소프트웨어의 모듈성을 높일 수 있는 방안을 제시하였다. 이호재(2003)은 컴포넌트 기반의 JavaBean™ 기술을 이용하여 재사용 가능하면서 복잡도를 감소하기 위해 일방향 정보 전달 기법을 제안하고, 이용한 Generic Agricultural Systems Simulators(GASS)를 개발하였다.

II. 컴포넌트의 구성

1. 자료의 통합

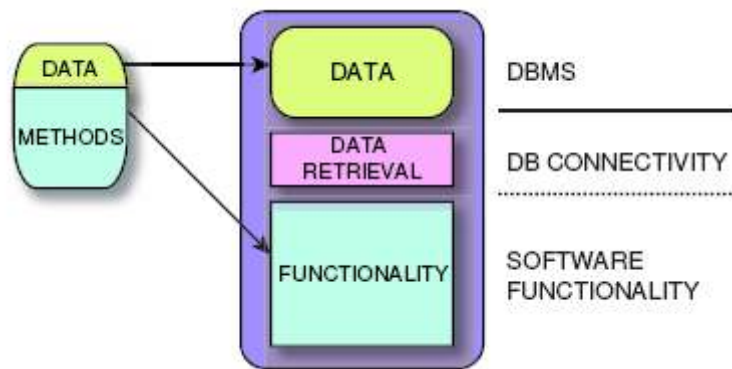


Fig. 1. Concept of Objective Software Module

객체 지향 패러다임은 데이터와 메소드로 구성되는 객체들의 집합으로 프로그래밍의 모듈성을 높이도록 하였다. 하지만 객체끼리 통신할 때 객체 각각의 데이터는 재사용되기 힘들고 통합하기 힘든 점이 존재한다. 따라서 Fig. 1과 같이 이대희(2000)이 제안한 객체적 소프트웨어 모듈을 이용하여 데이터는 데이터베이스를 통하여 관리하고, 컴포넌트에서는 기능과 데이터베이스를 연결해주는 연결고리 부분으로 구성할 수 있다.

2. 시스템의 구성

본 연구에서 구상하고 있는 모델은 추상화된 하나의 컴포넌트를 이용하여 속성을 지정함으로써 구체화된 컴포넌트를 만들어 낼 수 있고, 이 구체화된 컴포넌트를 실제로 인스턴스화하여 얻어진 인스턴스를 연결시켜 전체 시스템을 구성할 수 있도록 한다. 속성을 정의하는 데 있어 데이터는 모두 데이터베이스를 이용하여 통합 관리되어지며, 이를 통해 자료에 독립적인 모델 구성이 가능하도록 한다. 또한 만들어진 인스턴스 간의 관계를 정의하는 데 있어 GASS의 인접성을 이용한 일방향 정보 전달 기법(Unidirectional Information Fetch)을 적용하여 컴포넌트 간의 관계를 기하학적인 인접관계로 환원함으로써, 시스템을 직관적으로 구성할 수 있으며, 시스템을 구성하는 모델 간의 결합에 의한 복잡도를 낮추도록 한다.

3. 컴포넌트의 구성

본 연구에서 개발하고자 하는 컴포넌트는 크게 데이터, 속성, 해석엔진, 정보 통신로 구성되어질 것으로 생각되어진다. 데이터는 데이터베이스를 이용하여 관리되어지며, 속성은 범용 추상 컴포넌트로부터 구체화된 컴포넌트를 만들어내기 위한 특성을 지정한다. 지정된 속성을 이용하여 이미

구축된 프로그램 데이터베이스에서 자료를 가져와서 해석하여 구체화된 컴포넌트로 바뀌게 되고, 이들 컴포넌트를 인스턴스화한 인스턴스 간의 정해진 통신 규약을 통하여 정보를 주고 받을 수 있는 부분이 필요할 것으로 사료된다.

4. 복잡도와 컴포넌트 간의 통신

컴포넌트 기반의 개발에 있어 컴포넌트 간의 결합은 잘 정의된 인터페이스 간의 결합으로 정의할 수 있다. 하지만 인터페이스 간의 결합은 결합하고자 모델을 서로 연결하고 관리해 줄 수 있는 메인 모듈이 존재하여 관리해야만 한다. 메인 모듈에서는 각각의 컴포넌트 간의 통신을 제어하여야 하기 때문에 컴포넌트의 수가 증가함에 따라 컴포넌트의 수의 제공에 비례하여 증가하게 된다. 하지만 일방향 정보 전달 기법을 적용하면 컴포넌트 간의 결합에 있어 복잡도는 컴포넌트 수에 비례하기 때문에 큰 시스템을 개발하는 데 있어 복잡도를 낮추는 데 기여할 수 있다.

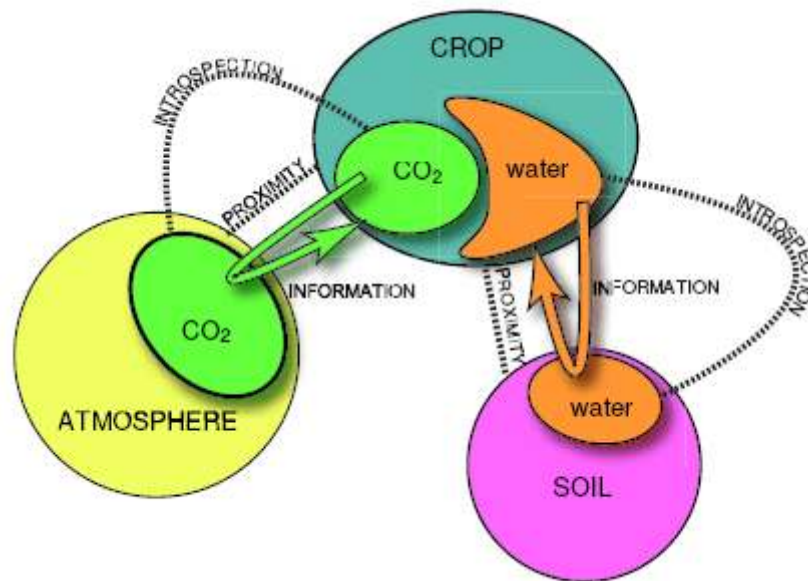


Fig. 2. Concept of GASS Component

Fig. 2는 작물이 대기와 토지로부터 이산화탄소와 물을 가져와 성장하는 모델의 예로써 GASS 컴포넌트간의 일방향적 정보 전달 방법을 묘사하고 있다. 컴포넌트 사이에는 기하학적 인접모델을 이용한 모델간의 정보 전달 형태를 지니고 있다. 이는 각 컴포넌트는 인접한 주변의 모든 컴포넌트를 찾고, 서로 간의 모델을 관찰하여 필요한 정보가 있을 경우 그 정보를 가져오는 형태를 취하고 있다.

III. 결론

단일 프로그램으로 개발되어 왔던 많은 프로그램들의 재이용은 지식의 재이용이라는 측면에서 중요한 문제라고 할 수 있다. 하지만 통합 자료 및 통합 프레임워크가 없는 상황에서는 I/O 자료를 변환하여 순차적으로 결합하거나, 통합하고자 하는 프로그램들 간의 인터페이스를 모두 이해하고 결합해야 하는 등의 문제로 인하여 거의 불가능하다. 본 연구에서 제안하는 프로그램은 이러한 상황에서 데이터베이스를 이용한 통합 자료 기반과 추상화된 인터페이스를 통하여 모델 간의 결합을 유연하게 함으로써, 기존 프로그램의 통합에 있어 생기는 문제점을 해결하고자 하였다. 또한 한발 더 나아가 범용 추상화 컴포넌트를 이용하여 복잡한 시스템을 유기적으로 구성해 보고자 한

다.

본 연구는 농림기술관리센터 '농업시설의 계획·설계를 위한 CAD와 GIS 자료구조 통합 시스템 개발'(과제번호: 203103-03-3)의 연구비 지원으로 수행되었습니다.

참고문헌

1. Christopher R. Myers, Software systems as complex networks: structure, function, and evolvability of software collaboration graphs, *Physical review*, 68, 046116, 2003.
2. Clemens Szyperski, Dominik Gruntz, Stephan Murer, *Component software: beyond object-oriented programming*, 2nd edition, pp.35-82, Addison Wesley, 2002.
3. Hatton, L., Does oo sync with how we think?, *IEEE Software*, 15(3), 46-54, 1998.
4. John Cheesman, John Daniels, *UML Components: a simple process for specifying component-based software*, Addison-Wesley, 2001.
5. Jones, J., B. Keating, and C. Porter, Approaches to modular model development, *Agricultural Systems*, pp. 421-443, 2001.
6. Ken Arnold and James Gosling. *The Java™ programming language*. Addison-Wesley Publishing Company, Inc, 1996.
7. Lee Dae Hee, A study on the structural design platform based on database, Master's thesis, Seoul National University, 2000.
8. Wojtek Kozacrynski and G. Booch, "Component-based software engineering", *IEEE Software*, Vol.15, No.5, pp.34-36, Sept./Oct., 1998.
9. Yi Ho Jae. Simplifying complexity and emerging self-organization of integrated agricultural system models. PhD thesis, Seoul National University, 2003.