

# FCA기반 클래스계층구조 설계를 위한 BlueJ의 확장

서정혁\*, 황석형\*\*, 양해솔\*\*\*

\*전문대학교 일반대학원 전자계산학과

\*\*전문대학교 컴퓨터정보학부

\*\*\*호서대학교 벤처전문대학원

e-mail:\*inerbi24@sunmoon.ac.kr, \*\*shwang@sunmoon.ac.kr

## Extension of BlueJ for Class Hierarchy Construction based on the Formal Concept Analysis

Jeong-Hyeok Seo\*, Suk-Hyung Hwang\*\*, Hae-Sool Yang\*\*\*

\*Dept of Computer Science, Sunmoon University

\*\*Division of Computer & Information Science, Sunmoon University

\*\*\*Graduate School of Venture, Hoseo University

### 요 약

객체지향 프로그램에 있어 클래스계층구조는 프로그램의 뼈대가 된다. 따라서 이러한 클래스계층구조를 얼마나 잘 만드느냐에 따라 프로그램의 품질이 좌우된다. 그러나 좋은 품질의 클래스계층구조를 구축하는 작업은 객체지향 초보자에게는 쉬운 일이 아니다.

본 논문에서는 FCA(Formal Concept Analysis)기법을 이용하여 클래스계층구조 설계 도구를 BlueJ의 확장기능으로 구현하였다. 본 연구결과는 객체지향 프로그래밍 초보자들이 클래스계층구조를 보다 수월하게 설계함으로써 좀 더 좋은 프로그램을 작성 할 수 있는 지원도구로서 제공될 수 있다.

### 1. 서론

클래스계층구조는 객체지향 소프트웨어 시스템의 핵심이 되며 재사용성을 갖춘 객체지향 아키텍처의 근간을 이루고 있다. 따라서 클래스계층구조를 구축하는 것은 객체지향 소프트웨어 개발공정에 있어 중요한 작업일 수밖에 없다. 그러나 클래스계층구조를 만들고 유지 보수하는 것이 쉬운 일이 아니기 때문에 많은 객체지향 프로그램 개발자나 설계자들은 클래스계층구조 설계를 도울 수 있는 도구나 방법론 등에 지대한 관심을 가지고 있다.

지금까지의 관련 연구들[1][2]에서는 FCA라는 수학적 기법을 사용하여 객체지향 클래스계층구조 설계에 적용시키기 위한 이론적 토대를 제공하고 있다. FCA는 lattice 이론을 바탕으로 데이터를 분석하는 수학적 기법이며 이 기법을 사용함으로써 좀 더 잘 정의된 클래스 계층구조를 설계 할 수 있다.

한편, 객체지향 프로그램 초보자들의 프로그래밍

을 수월하게 수행할 수 있도록 지원해주는 여러 가지 도구들 중에 BlueJ라는 도구가 있다. BlueJ는 모델구동형기법을 사용하는 자바통합개발환경(IDE)으로 객체지향 초보자들도 객체지향 프로그래밍을 쉽게 할 수 있는 장점을 가지고 있다[3].

본 연구에서는 객체지향 프로그래밍 초보자들이 FCA기법을 이용해 클래스계층구조를 쉽게 설계할 수 있는 도구(FCAWizard)를 구축하였다. FCAWizard는 BlueJ를 기반으로 하는 플러그인 형태의 확장기능으로 구현되었으며, 객체지향 초보자들은 BlueJ를 사용함으로써 잘 정의된 클래스계층구조를 갖춘 좋은 품질의 프로그램을 수월하게 작성 할 수 있는 프로그래밍 개발환경이 제공된다.

본 논문의 구성은 다음과 같다. 2장에서는 클래스계층구조설계와 관련된 기존의 연구들에 대해서 소개한다. 3장에서는 FCA의 이론에 대해서 설명한다.

4장에서는 자바통합개발환경인 BlueJ에 대해서 소개한다. 5장에서는 BlueJ 확장기능으로서 클래스계층구조 설계도구인 FCAWizard에 대해서 설명한 후 6장에서 결론 및 향후 계획에 대해서 기술한다.

2. 관련연구

클래스계층구조 설계에 있어 기존의 여러 연구들 [1][2][4]을 살펴보면 다양한 관점에서 좋은 품질의 클래스계층구조 설계를 위한 이론적 설계지침 등을 제안하고 있다. [4]의 연구에서는 정당성과 확장성의 측면을 고려한 클래스계층구조 설계지침을 제안하였다. 즉, 적당한 클래스계층구조를 만들기 위해서는 계층구조 내에 싸이클이 존재하면 안되며 컴퍼먼트들의 이름은 고유해야 하며 상위 클래스는 추상클래스로 하위 클래스는 구상클래스로 정의되어야 한다. 위의 조건을 만족하는 클래스계층구조는 잘 정의된 클래스계층구조라고 한다.

한편, [1][2]의 연구에서는 클래스계층구조를 설계하는데 있어 FCA(Formal Concept Analysis)라는 수학적 기법을 이용하였다. [1]에서는 정적요소인 객체들이 가지는 속성들 중 공통적인 속성들을 최대한 추출하여 추상화시킴으로서 계층구조를 만들었고, [2]에서는 동적인 의존관계를 고려한 계층구조 구축 방법을 제안하고 있다. 이와 같은 연구[1, 2]는 문제 영역내의 객체 및 객체의 특성(속성, 동작)등에 관한 정보를 토대로 FCA를 이용하여 Concept Lattice로 구조화하고, 이러한 Concept Lattice를 기반으로 설계자에게 클래스계층구조를 구축하기 위한 이론적인 토대를 제공하고 있다.

3. FCA(Formal Concept Analysis)

FCA는 객체들, 속성들 그리고 객체와 속성간의 이항관계로부터 Concept을 추출하여 Lattice 구조로 계층화시키기 위한 수학적 기법이다. FCA기법을 이해하기 위한 몇 가지 기본사항들은 다음과 같다.

Formal Context는 세 개의 구성요소(O, A, R)로 이루어지며, O는 객체들의 집합, A는 속성들의 집합, R은  $R \subseteq O \times A$ 의 이항관계이다.

| 객체 \ 속성 | 소속학과 | 지도교수 | 봉급 | 지도학생 | 학위 |
|---------|------|------|----|------|----|
| 교수      | x    |      | x  | x    | x  |
| 직원      |      |      | x  |      |    |
| 학부생     | x    | x    |    |      |    |
| 대학원생    | x    | x    |    |      | x  |

표 1 Context Table의 예

Context는 위의 표1에서와 같이 표로 나타낼 수 있으며 각각의 행들은 객체들을, 각각의 열들은 속성들을 나타낸다. 또한 x로 표시된 부분은 객체들과 속성들 사이의 이항관계가 존재함을 표시한다.

Formal Concept은 주어진 X와  $Y (X \subseteq O, Y \subseteq A)$ 가  $intent(X) = Y \wedge extent(Y) = X$ 를 만족할 때, (X, Y)를 Formal Concept이라고 정의한다[5]. 단,

$$intent(X) = \{a \in A \mid \forall o \in X : (o, a) \in R\},$$

$$extent(Y) = \{o \in O \mid \forall a \in Y : (o, a) \in R\}.$$

예를 들어,  $X = \{\text{교수, 직원}\}$ 이고  $Y = \{\text{봉급}\}$ 이라고 할 때, 객체들의 집합 X가 공통적으로 가지는 속성들은  $intent(X) = \{\text{봉급}\}$ 이며 속성들의 집합 Y를 공통적으로 가지는 객체들은  $extent(Y) = \{\text{교수, 직원}\}$ 이 되므로  $(\{\text{교수, 교직원}\}, \{\text{봉급}\})$ 은 Formal Concept이 된다. 따라서 Context Table에서 객체들과 속성들 간에 이항관계의 최대사각형으로 더 많은 Concept들을 추출해 낼 수 있다.

이와 같이 추출된 Concept들 간에는 일종의 순서 관계(Superconcept-Subconcept 관계)가 존재한다. 즉, 주어진 두 개의 Concept  $C_1 = (X_1, Y_1), C_2 = (X_2, Y_2)$ 가  $(X_1, Y_1) \leq (X_2, Y_2) \Leftrightarrow (X_1 \subseteq X_2) \wedge (Y_1 \supseteq Y_2)$ 를 만족할 때  $C_1$ 은  $C_2$ 의 Subconcept, 또는  $C_2$ 는  $C_1$ 의 Superconcept이라고 정의한다[5].

예를 들어,  $C_2 = (\{\text{교수, 직원}\}, \{\text{봉급}\})$  이고  $C_1 = (\{\text{교수}\}, \{\text{소속학과, 봉급, 지도학생, 학위}\})$ 라 할 때,  $\{\text{교수}\} \subseteq \{\text{교수, 직원}\}$ 이고  $\{\text{봉급}\} \supseteq \{\text{소속학과, 봉급, 지도학생, 학위}\}$ 이므로,  $C_1$ 은  $C_2$ 의 Subconcept이 된다.

이상의 제반정의들을 토대로, Concept Lattice는 다음과 같이 정의된다. Formal Context  $C = (O, A, R)$  이고  $B(C)$  가 모든 컨셉들의 집합일 때,  $Concept\ Lattice = (B(C), \leq, \wedge, \vee, (extent(X), A), (Y, intent(O)))$

이와 같은 Concept Lattice는 그림 1과 같이 Hasse Diagram으로 표현된다.

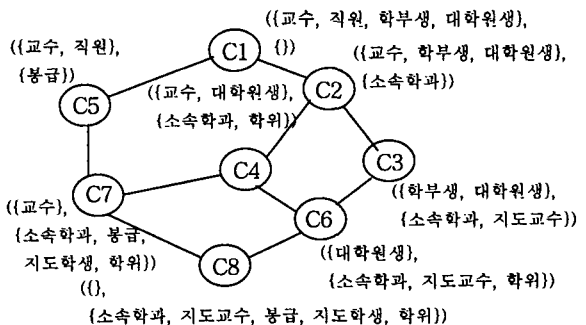


그림 2 Concept Lattice를 Hasse Diagram으로 표현한 예

#### 4. BlueJ

BlueJ는 객체지향 초보자들을 대상으로 하는 자바 통합개발환경이다[3]. 모델구동형 아키텍처를 사용하여 자바를 공부하는 객체지향 초보자들이 코드보다는 모델을 먼저 만들고 그에 따른 구현을 할 수 있도록 도와준다. 또한 Extension API를 통하여 여러 가지 확장이 가능하도록 함으로써 개발환경의 유연성을 제공한다[6]. 따라서, 누구든지 필요한 기능이 나 새로운 기능을 추가 가능하다. 특히 BlueJ는 호주에서 자바언어 초보자들의 교육을 위한 도구로서 34%에 달하는 높은 사용률을 보이고 있다.

따라서, 본 논문에서는 객체지향 프로그래밍 초보자를 위한 프로그램 설계 및 클래스계층구조 구축을 지원하기 위하여 FCA 이론을 바탕으로 한 클래스계층구조 설계도구인 FCA Wizard를 설계/구현하여 BlueJ의 확장기능으로 구축하였다.

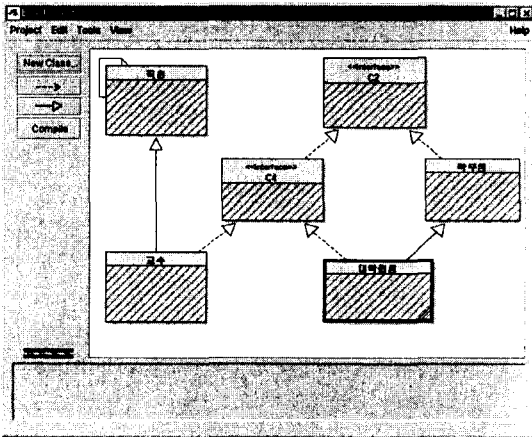


그림 3 BlueJ의 실행 화면

#### 5. 클래스계층구조 설계도구 : FCAWizard

FCAWizard는 클래스계층구조 설계도구로 BlueJ를 사용하는 객체지향 초보자들이 클래스계층구조를 좀 더 쉽게 설계할 수 있도록 지원하는 도구이다. 따라서 BlueJ를 사용하는 객체지향 초보자들은 잘 정의된 클래스계층구조를 갖춘 좋은 품질의 객체지향 프로그램을 만들 수 있다.

FCAWizard는 다음과 같은 절차에 따라서 사용된다. 첫 번째 단계는 객체들과 속성들을 입력하는 단계로 클래스계층구조를 만들고자 하는 객체들과 객체들이 가질 수 있는 속성을 입력하는 단계이다(그림 3).

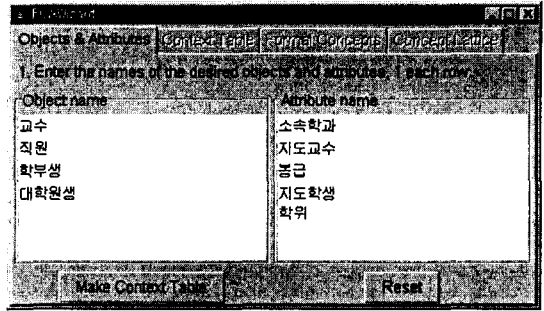


그림 4 객체 및 속성 입력

두 번째 단계는 객체들과 속성들 간의 이항관계를 체크하는 단계로 각각의 객체들이 어떠한 속성을 가지고 있는지를 테이블형식의 표(Context Table)에 체크해줌으로써 설정할 수 있다(그림 4).

그림 5 Context의 이항관계 표시

세 번째 단계(그림 5)는 Concept들을 표시하는 단계로 두 번째 단계에서 일어난 데이터(객체들의 집합, 속성들의 집합, 객체와 속성간의 이항관계)들을 이용하여 Concept들을 추출해 표 형식으로 보여준다.

| Concepts | Extension           | Intension                  |
|----------|---------------------|----------------------------|
| C1       | {교수, 직원, 학부생, 대학원생} | 0                          |
| C2       | {교수, 학부생, 대학원생}     | {소속학과}                     |
| C3       | {학부생, 대학원생}         | {소속학과, 지도교수}               |
| C4       | {교수, 대학원생}          | {소속학과, 학위}                 |
| C5       | {교수, 직원}            | {등급}                       |
| C6       | {대학원생}              | {소속학과, 지도교수, 학위}           |
| C7       | {교수}                | {소속학과, 등급, 지도학생, 학위}       |
| C8       | 0                   | {소속학과, 지도교수, 등급, 지도학생, 학위} |

그림 6 Concept 추출 결과 표시

네 번째 단계는 세 번째 단계를 토대로 추출된 Concept들을 Superconcept- Subconcept 관계를 사용하여 Hasse Diagram으로 표현해 주는 단계로 Concept들의 모든 객체들과 속성들을 표현해주는

Concept Lattice(그림 6)와 상속관계에 있어 불필요한 정보들을 제거한 Inheritance Concept Lattice(그림 7)의 두 가지 표현방법이 있다.

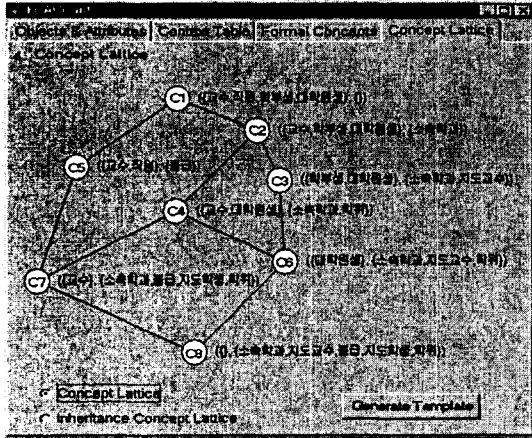


그림 7 Concept Lattice 표시

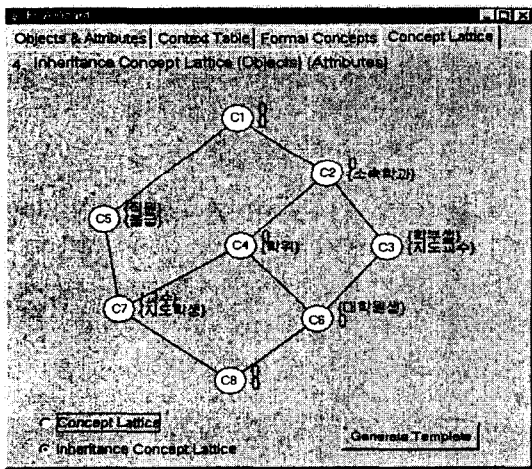


그림 8 Inheritance Concept Lattice 표시

네 번째 단계에서 만들어진 Lattice로부터 불필요한 요소들을 제거한 형태의 클래스계층구조를 토대로 프로그램상의 클래스 템플릿을 만들어 BlueJ에서 로드하게 되면 그림 2와 같은 클래스계층구조로 표현된다.

6. 결론 및 향후 연구 계획

본 논문에서는 클래스계층구조를 설계하는데 있어 FCA기법을 이용하여 객체들이 공통적으로 가지는 속성을 최대한 추출하여 추상화시킴으로서 잘 정의된 클

래스계층구조를 설계할 수 있는 지원도구를 BlueJ의 확장기능으로서 구현하였다. 특히, FCAWizard에 의해 생성된 Concept Lattice에서는 문제 영역으로부터 클래스 및 클래스계층구조를 생성/구축하기위한 추상화 작업을 할 때 생성 가능한 모든 클래스 추상화 형태를 확보할 수 있다. 따라서 클래스계층구조 설계도구인 FCAWizard를 사용하면 공통속성을 최대한 추출한 형태의 품질 좋은 클래스계층구조를 수월하게 구축할 수 있다.

향후 연구 과제로는 클래스계층구조를 설계하는데 있어 지금까지는 계승관계에만 초점을 맞추고 있지만 더 나아가 클래스들의 집약, 합성, 연관 관계에 대한 연구들도 행해져야 할 것이다. 또한 클래스계층구조 내부의 동적인 의존 관계를 고려한 클래스계층구조의 구축 개발환경에 관하여 연구해 나갈 계획이다.

참고문헌

[1] Suk-Hyung Hwang, "Towards a Semi-Automatic Construction of Object Oriented Class Hierarchies using Formal Concept Analysis", *Proceeds of the ACIS ICIS 2004*, pp 381-387, Aug 2004.

[2] Gabriela Arevalo, "Understanding Behavioral Dependencies in Class Hierarchies using Concept Analysis", *Proceedings of LMO 2003: Languages et Modeles à Objets*, Hermes, Paris, pp. 47-59. January 2003

[3] BlueJ Web Site([www.bluej.org](http://www.bluej.org))

[4] Suk-Hyung Hwang, "An approach for construction of valid and extensible class hierarchy structures", *Proceeds of KSEJW-2003*, pp 61-68, August 2003.

[5] Frank Buchli, "A Short FCA Primer", *The summary of the Appendix of the Mater Thesis of Frank Buchli*, University of Bern, 2003

[6] BlueJ Extension API(<http://www.bluej.org/doc/extensionsAPI/>)