페트리 넷을 이용한 EJB기반의 문제 학습 시스템

정화영

예원대학교 정보경영학부

EJB Based Examination Studying System using Petri Net

Hwa-Young Jeong

Yewon University

E-mail : jmichael@hanmir.com

요 약

최근, 컴포넌트 기술의 발전에 따라 웹 기반 교육 시스템에서도 이를 응용하여 다양한 교육 컨텐츠의 제공 및 활용하려는 연구가 진행되고 있다. 또한, EJB는 웹 기반 컴포넌트 모델로서 웹 기반 응용 시스템부분에서 많은 관심을 받고 있다. 그러나, 컴포넌트 기반 웹-코스웨어는 이용 가능한 상용 컴포넌트와 인식부족으로 아직까지 실용화되지 못하고 있는 실정이며 분석에 관한 정형화도 이루어지지 않고 있다.

따라서, 본 연구에서는 EJB 컴포넌트를 이용한 웹 코스웨어를 분석 및 구현을 하였다. 컴포넌트들 사이의 메시지 흐름에 대한 정형적인 명세 분석을 위하여 페트리 넷을 이용하였으며, 각 기능에 따라 무상태 세션 빈의 형식으로 EJB 시스템을 구현하였다.

I. 서 론

웹에 제시된 학습자료는 필요나 목적에 따라 쉽게 수정이 가능하므로 짧은 시간에 적은 노력으로 최대한의 효과를 높일 수 있다[1]. 이에 따라 다양하고 빠르게 변화되는 컨텐츠의 효율적인 대응 및 적용을 위한 소프트웨어 개발기법은 현재까지 크고 복잡한 소프트웨어를 개발하기 위해서 객체지향방법론들이 널리 적용되었으나[2], 이미 존재하는 소프트웨어 컴포넌트를 조립함으로서 시스템을 개발하는 방법인 CBSD[3]를 도입하려는 개발기법의 변화가 일고 있다. 또한, 웹 기반으로는 서버측 컴포넌트 모델인 EJB가 이용되고있다[4]. 이에 따라, 웹 기반 교육 시스템의 효율적인 개발을 위해서는 웹 컴포넌트를 이용한 많은 연구 및 사례들이 필요하다.

따라서, 본 연구에서는 EJB를 조립, 합성한 문제풀이 시스템을 설계 및 구현하고자 한다. 또한, 조립된 컴포넌트들의 메시지 흐름을 정형적으로 명세하기 위하여 페트리 넷[5]을 이용하였다.

II. 관련연구

2.1 컴포넌트 기반 소프트웨어 개발기법

컴포넌트는 응집력을 갖는 소프트웨어 구현으로서, 독립적으로 개발되고, 인터페이스에 대한 명시적이고 잘 정의된 스펙을 갖는다. 컴포넌트는 그 자체를 수정하지 않고 다른 컴포넌트들과 조립될 수 있고, 그의 특성을 맞출 수 있다[6].

따라서, 컴포넌트 모델은 컴포넌트의 기본적인 아키텍처, 컴포넌트의 인터페이스, 그리고 컴포넌트와 컨테이너간의 상호작용을 위한 메커니즘을 정의한다. 이와 같이 컴포넌트 모델은 재 사용할 수 있는 컴포넌트를 지원하기 위한 환경을 정의한다. 이에 관한 프로세스는 [그림 1]과 같다.

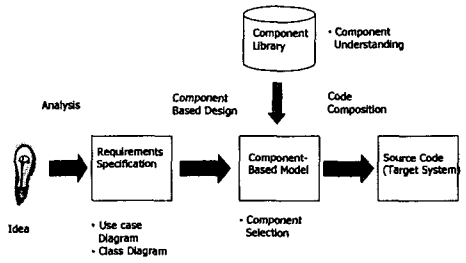

그림 1. CBSD의 프로세스

2.2 페트리 넷

페트리 넷의 일반적인 특성은 다음과 같다[7].

첫째, 시스템의 자료분할 행위와 동시성, 동기성을 따른다.

둘째, 작업수행 분석을 위한 결과를 나타낸다.

셋째, 객체지향 소프트웨어 구조에서 자동적인 행위분석의 의미로 사용된다.

즉, 페트리 넷은 동시성, 분산, 동기성, 병렬성, 결정적과 비결정적을 포함하는 다양한 시스템을 위한 정형적인 명세도구이다. 기본 페트리 넷 구조는 Place와 Transition의 유한집합, arc들의 유한집합, 초기 마킹을 정의하는 토큰의 집합들로 구성된다. arc들은 입력과 출력의 기능을 갖고 있으며, 기능들은 Place에서 Transition으로, Transition에서 Place로의 토큰에 대한 흐름을 나타낸다.

페트리 넷은 (P,T,Ai,Ao,m_0)과 같이 나타낼 수 있다. P는 Place집합, T는 Transition집합, Ai은 입력 Incidence Matrix, Ao은 출력 Incidence Matrix을 나타낸다. 또한, m_0은 초기 마킹을 나타낸다. 이에 따라, Place와 Transition의 입출력 관계는 다음과 같이 나타낸다.

$$\bullet p = \{t \mid a(t,p) \neq 0, t \in T\}, \quad p\bullet = \{t \mid a(p,t) \neq 0, t \in T\}$$
$$\bullet t = \{p \mid a(p,t) \neq 0, p \in P\}, \quad t\bullet = \{p \mid a(t,p) \neq 0, p \in P\}$$

또한, 입력 Place에 대한 Transition의 인에이블 규칙은 다음과 같다.

$$\forall p \in \bullet t, a(p,t) \leq m(p)$$

Transition t가 인에이블된 후에 대한 점화규칙은 다음과 같다.

$$\forall p \in \bullet t, m'(p) = m(p) - a(p,t)$$
$$\forall p \in t\bullet, m'(p) = m(p) + a(t,p)$$

위 식에서, 현재의 마킹 m은 새로운 마킹 m'으로 변화되면서 Transition t이전의 Place 토큰이 연결된 arc수만큼 감소되고, t이후의 Place 토큰은 연결된 arc수만큼 가산됨을 알 수 있다.

III. EJB 컴포넌트 기반의 문제학습 웹-코스웨어 분석 및 구현

본 시스템 구성은 그림 3과 같다. 메인 서버에서의 프리젠테이션 로직을 위한 HTML, JSP, 서블릿을 이용하였으며, 컴포넌트를 위한 EJB서버를 별도로 두었다. 데이터베이스는 LoginEJB는 회원인증DB를, 문제풀이EJB는 문제관련DB를, 해답풀이EJB는 해답풀이DB를, 전체결과EJB는 전체결과DB를 핸들링 하도록 한다.

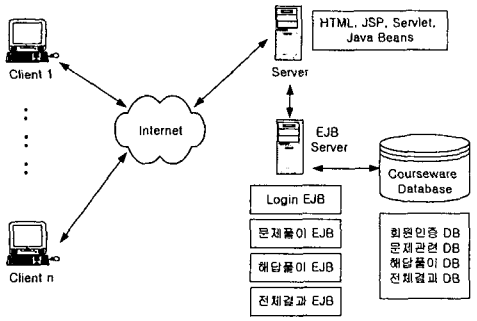

그림 2. EJB를 활용한 문제풀이 시스템

로그인 이후에 문제풀이를 할 수 있으며 해답을 요구할 수 있다. 또한, 전체결과를 요청하여 이를 확인할 수 있다. 이에 대한, 각 컴포넌트 구성은 다음 그림 3과 같이 나타낸다.

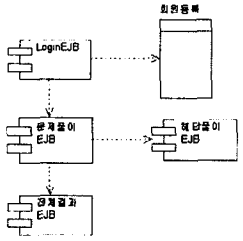

그림 3. Component Diagram

각 프로세스에 대한 사용자 요구사항은 다음 그림 4와 같다.

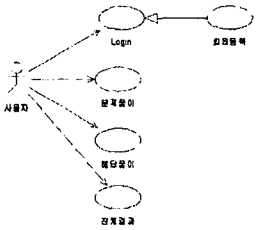

그림 4. Usecase Diagram

이를 기반으로 한 클래스는 다음 그림 5와 같이 구성하였다. 이는, 각 컴포넌트에 있는 리모트 메소드들을 모두 풀어놓은 상태로서 각 컴포넌트 내의 소속 클래스들 사이의 관계를 나타낸다.

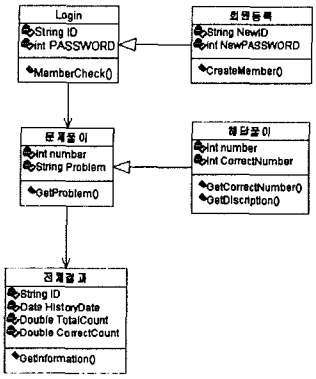

그림 5. Class Diagram

이에 따른 프로세스 순서도는 다음 그림 6과 같이 구현하였다.

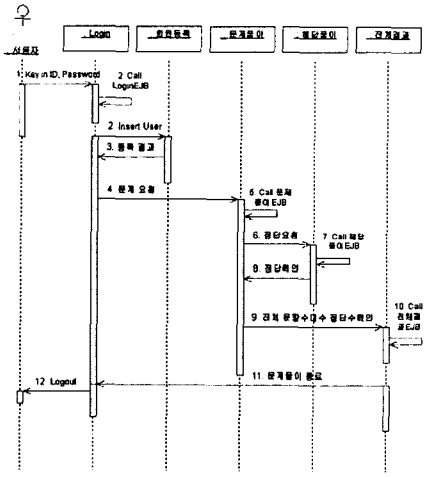

그림 6. Sequence Diagram

즉, 컴포넌트에 해당하는 Login, 문제풀이, 해답풀이, 전체결과 부분은 각자 자신의 EJB를 호출하도록 하였다. 위 순서에 따라 상태의 흐름을 구성한 부분은 그림 7과 같다.

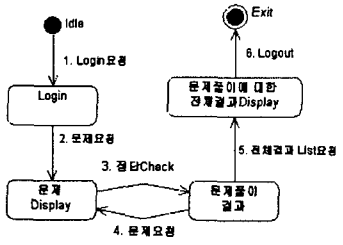

그림 7. Statechart Diagram

본 시스템의 구성에 있어서 각 단계별 흐름을 정형적으로 파악하기 위하여, 다음 그림 8과 같이 전체 시스템을 페트리 넷으로 재구성하였다.

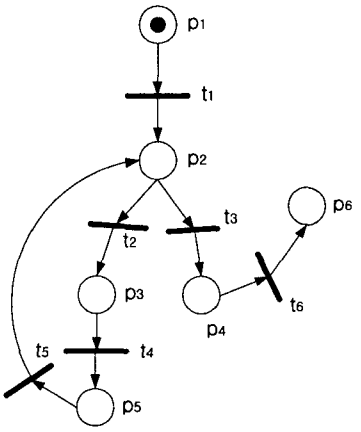

그림 8. 문제풀이 시스템의 페트리 넷

Place p_1은 Login전의 유휴상태를 나타내며, p_2는 LoginEJB, p_3는 문제풀이EJB, p_4는 전체결과EJB, p_5는 해답풀이EJB, p_6는 Logout된 Exit상태를 나타낸다. 또한, Transition t_1은 Login요청 메시지를 나타내며, t_2는 문제풀이 요청 메시지, t_3는 전체 결과보기 요청 메시지, t_4는 정답요청 메시지, t_5는 다음 문제 요청 메시지, t_6는 Logout 요청 메시지를 나타낸다.

초기 마킹m_0는 m_0=(1,0,0,0,0,0)이다. 따라서, Place p_1에 대한 토큰은 $m(p_1)=1$이다. p_1에서 t_1로의 arc수 $a(p_1,t_1)=1$이므로 Transition t_1에 관한 인에이블 규칙이 성립되며, 점화규칙에 의하여 $m(p_1)=0$이 되며 $m(p_2)=1$이 된다. Place p_1에서는 $\{t_2, t_3\} \in p_1$이 되므로 Transition t_2와 t_3가 동시에 발화하지 못하며 t_2나 t_3중 하나가 발화하게 된다. 따라서 Login이후 문제풀이를 요청하든지 아니면 전체결과를 택하게된다. p_5의 문제풀이까지 끝나게 되면 다음 문제요청 메시지인 t_5가 발화되면서 초기 Login상태로 되돌아간다. 따라서, 각 Place에 위치한 EJB의 흐름경로는 이상 없이 운용될 수 있다. 위 과정을 통하여 구현된 결과는 다음과 같다. 즉, 그림 9는 초기 Login화면을, 그림 10은 문제를 나타내고, 그림 11, 12에서 정답과 결과를 나타낸다.

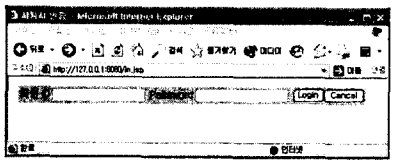

그림 9. 초기 Login 화면

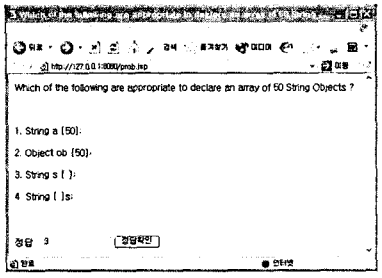

그림 10. 문제풀이 화면

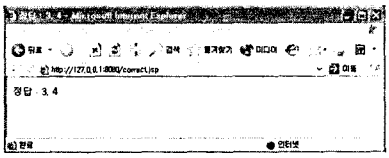

그림 11. 해답풀이 화면

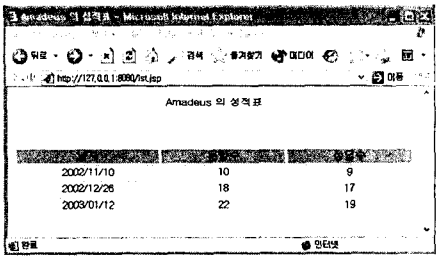

그림 12. 전체결과 화면

IV. 결론 및 향후 연구과제

본 연구에서는 EJB 컴포넌트 기반의 문제풀이 시스템을 구현하였으며, 각 컴포넌트의 흐름을 정형적으로 명세 하고자 페트리 넷을 이용하였다. 각 기능단위를 독립적인 운용이 가능한 EJB컴포넌트로 구현함으로서 CBSD에서 갖는 장점인 시스템 개발의 효율성 및 대용성, 재 사용성을 높일 수 있었다. 또한, 페트리 넷을 이용함으로서, 컴포넌트 조립 및 합성 시에 간과하기 쉬운 컴포넌트 사이의 메시지 흐름을 정형적으로 분석할 수 있었으며, 완성된 시스템의 오류율을 줄일 수 있었다.

그러나, 본 시스템은 컴포넌트의 효율적인 운영관리를 위하여 메인 서버와 EJB컴포넌트 서버를 분리하였으나, 이들 사이의 네트워크 부하가 충분히 고려되지 않았다. 또한, 컴포넌트를 이용한 체계적인 웹 기반 교육 시스템을 구축하기 위해서는 다양한 교육 컨텐츠와 이를 핸들링 하는 컴포넌트들을 구체적으로 분류하고, 이를 효과적으로 구현하기 위한 아키텍처기반 컴포넌트 합성 기법이 적용되어야한다.

참고문헌

[1] 이재무, "개인차를 고려한 웹 기반 코스웨어 개발", 한국컴퓨터산업교육학회 논문지 VOL. 02 NO. 12, 2001.

[2] 김영호, 김영곤, 배두환, 김민경, 유병규, "객체지향 개발방법의 체계적인 구성", 정보과학회논문지 제 27 권 제 5 호, 2000.

[3] SEI in Carnegie Mellon University, "Component-Based Software Development/CO TS Integration", http://www.sei.cmu.edu/str-/descriptions/cbsd_body.html, 2001.

[4] Sun Microsystems Inc, "Enterprise Java Beans Specifications", at URL:http://java.sun.com, 2001.

[5] James L. Peterson, "Petri Nets", ACM Computing Surveys 9(3):223-252, September, 1997.

[6] Desmond Francis D'Souza and Alan Cameron Wills, "Object, Component, and Frameworks with UML : The Catalysis Approach", Addison-Wesley Object Technology Series, 1998.

[7] David, R. and Alla, H., "Petri Nets for Modeling of Dynamic Systems: A Survey," Automatica, vol. 30, no. 2, pp. 175-202, 1994.