

컴포넌트 성능 측정 결과를 이용한 비교 컴포넌트 추천 기능의 EJB 컴포넌트 레파지토리

이지현⁰ 김우식 신규상
한국전자통신연구원 소프트웨어·컨텐츠기술연구부 컴포넌트공학연구팀
(jihyun⁰, wsk, gsshin)⁰@etri.re.kr

EJB Component Repository Based on Performance Measurement for Recommending Comparative Components

Jihyun Lee⁰ Woosik Kim Gyu-Sang Shin
Component Engineering Team, S/W·Contents Technology Department, ETRI

요 약

컴포넌트 레파지토리는 생성된 컴포넌트를 저장하고 컴포넌트 사용자가 필요한 컴포넌트를 찾아 재사용을 쉽게 할 수 있도록 제안되어 왔으나 대부분의 컴포넌트 레파지토리는 사용자가 찾으려는 컴포넌트의 존재를 쉽게 알 수 없어 컴포넌트 재사용에 중요한 장애가 되고 있다. 본 논문에서는 컴포넌트에 대한 성능 측정 결과를 이용한 레파지토리를 구축한다. 이 컴포넌트 레파지토리는 사용자가 관심 갖는 컴포넌트에 대해 비교할만한 기능과 성능을 갖는 컴포넌트들을 자동으로 추출하여 제공한다. 대형 컴포넌트 레파지토리에서 컴포넌트를 손쉽게 찾지 못하는 경우가 빈번한 점에 비해 본 논문에서는 자동화된 방법에 따라 재사용 컴포넌트를 보다 쉽게 식별할 수 있는 방법을 제공한다.

1. 서론

대형 컴포넌트 레파지토리는 컴포넌트 사용자가 찾고자 하는 컴포넌트의 존재를 손쉽게 알 수 없고, 심지어 관련 컴포넌트가 존재하는지 알 수 없어 컴포넌트 개발자들은 반복적으로 같은 기능을 갖는 함수를 재생산해 왔다[1]. 컴포넌트는 기능과 그 기능에 대한 명세를 갖는 재사용 가능한 소프트웨어 단위이다. 컴포넌트 기반 소프트웨어에서 유지 보수는 변경하려는 컴포넌트에 대치하고자 하는 컴포넌트를 적용함으로써 이루어진다. 하지만, 현재까지의 컴포넌트 레파지토리는 브라우징이나 질의와 같은 기능만으로 컴포넌트 식별을 지원해 왔다[2]. 따라서, 사용자가 찾고자 하는 컴포넌트의 손쉬운 식별과 유사 컴포넌트에 대한 식별은 효과적으로 이루어지고 있지 못하다. 본 논문에서는 컴포넌트의 기능 뿐만 아니라 성능까지 고려한 컴포넌트 레파지토리를 구성하는 방법과 이로부터 사용자에게 비교할 만한 기능과 성능을 갖는 컴포넌트를 자동으로 추출하는 장치 및 방법에 대해 언급한다. 컴포넌트를 자동으로 식별하여 제공하는 장치는 새로운 컴포넌트를 생성하려 할 때 이미 개발된 컴포넌트의 존재를 검사하

고, 대형 비즈니스 어플리케이션을 모델링하기 위한 다양한 용도로 사용될 수 있다.

본 논문의 구성은 다음과 같다. 2장에서 컴포넌트 성능 측정 방법을 제안하고, 3장에서 성능 측정 결과를 가공하여 비교할만한 기능과 성능을 갖는 컴포넌트 패턴을 찾는 방법에 대해 설명한다. 제 4장에서 성능 측정 결과와 명세 정보를 저장할 수 있는 컴포넌트 레파지토리를 제안한다. 5장에서 연관 컴포넌트를 자동으로 추출하여 사용자에게 제공하는 컴포넌트 레파지토리의 활용 방법에 대해 언급한 후 제 6장에서 결론을 맺는다.

2. 컴포넌트 성능 측정

본 논문은 EJB(Enterprise JavaBean) 기반의 컴포넌트를 성능 측정 대상으로 고려한다. EJB 컴포넌트는 하나 이상의 빈(bean)으로 구성되고, 각 빈마다 비즈니스 기능인 홈 인터페이스(home interface)와 원격 인터페이스(remote interface)로 나뉘어 명시된다. 컴포넌트의 성능은 홈/원격 인터페이스에 명시되어 있는 메소드들이 반응하는데 걸리는 시간, CPU 사용 변화율, 메모리 사용 변화율로 나뉘어 측정된다. 그림 1은 컴포넌트 성능 측정 장치

에 대한 구조를 보여 준다.

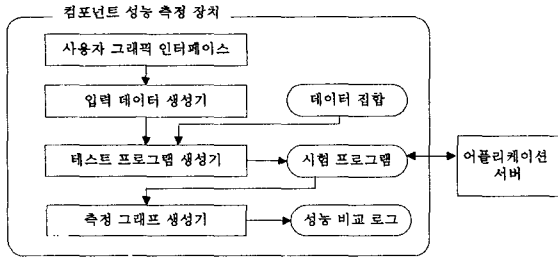


그림 1. 컴포넌트 성능 측정 장치의 구조

사용자 그래픽 인터페이스를 통해 컴포넌트 인터페이스 명세 정보와 어플리케이션 서버 접속 정보를 입력하면 인터페이스에 포함되어 있는 메소드를 분석하고, 메소드의 파라미터에 자동으로 전달될 데이터를 입력 데이터 생성기와 데이터 집합에서 생성한다. 테스트 프로그램 생성기는 컴포넌트의 성능을 측정하기 위한 프로그램을 자동으로 생성하고, 어플리케이션 서버에 전개되어 있는 컴포넌트에 대해 메소드 반응 시간, CPU 사용 변화율, 메모리 사용 변화율을 측정하여 컴포넌트의 성능을 측정한다.

컴포넌트 성능 측정 장치는 성능 측정 결과를 사용자별 로그 파일에 저장하고 측정 그래프 생성기를 통해 사용자에게 성능 측정 결과를 그래프로 보여준다. 측정 그래프 생성기는 성능 측정된 컴포넌트 목록으로부터 컴포넌트들을 선택하고 비교 그래프를 통해 비교 성능 측정을 할 수 있다. 함께 성능 비교된 컴포넌트 리스트와 성능 비교 결과는 사용자의 요구 시 성능 비교 로그에 저장된다.

3. 성능 측정 컴포넌트의 연관성 추출

비교 가능한 추천 후보 컴포넌트를 찾을 때 고려해야 할 컴포넌트의 특성은 기능의 유사성과 성능의 비교 가능성이다. 컴포넌트 비교 성능 측정 결과 로그는 사용자로부터 명확히 입력된 프로파일로써[3][4] 함께 성능 비교된 컴포넌트들간의 연관성을 추출하기 위해 사용된다.

본 논문에서 기능 유사성이 있고 성능의 비교 가능성이 있는 후보 컴포넌트를 추출하기 위해 연관 규칙 탐색 방법을 적용한다. 연관 규칙은 여러 항목에 대해 방대한 데이터가 있을 때 이들 항목 중 한 항목과 다른 항목이 서로 연관되어 있는 지를 찾아 내는 규칙으로 항목 간에 숨겨져

있는 패턴을 추출하기 위해 사용된다. 연관 규칙의 항목 간의 연관 정도는 지지도(support)와 신뢰도(confidence)로 표현된다. 연관 규칙의 적용은 다음과 같다. A 라는 항목과 B 라는 항목이 서로 연관되어 있다고 하면 이 연관성을 이용하여 상품의 판매를 촉진하기 위한 대응책 및 밀접하게 연관된 항목을 분석하는 방법으로 사용 가능하다. 각 컴포넌트에 대해 연관성이 높은 컴포넌트들을 찾기 위해 연관 규칙 탐색의 대표적인 방법인 APRIORI 알고리즘[5]을 컴포넌트 성능 측정 사용자들로부터 얻어진 비교 성능 측정 로그에 적용한다. 다음 그림은 비교 성능 측정 로그로부터 연관 컴포넌트를 추출해 나가는 과정을 보여 준다.

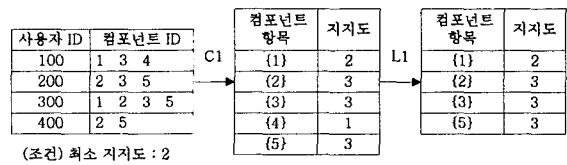


그림 2. 연관 컴포넌트 추출 과정

연관 컴포넌트 추출은 모든 사용자의 비교 성능 측정 로그로부터 컴포넌트 항목을 하나씩 증가시키면서 최소 지지도보다 지지도가 높은 컴포넌트 목록을 더 이상 찾을 수 없을 때까지 반복하여 찾는다.

4. 성능 측정 데이터 및 명세 정보 저장 레파지토리

컴포넌트 레파지토리에 저장되는 정보는 컴포넌트 성능 측정 결과와 컴포넌트 명세 정보로 나뉘어진다. 첫번째, 컴포넌트 성능 측정 결과는 해쉬 구조로 구성되며 하나의 행 구조는 인터페이스 이름을 식별하기 위한 키 값, 메소드의 반응 시간, CPU의 변화량, 메모리의 변화량, 메소드 반응 시간 요약 정보, CPU 변화량 요약 정보, 메모리 변화량 요약 정보를 저장하도록 구성된다. 메소드 반응 시간 항목은 메소드 시그니처와 해당 메소드 반응 시간의 쌍들로 이루어지고, 반응 시간을 기준으로 오름 정렬되어 저장된다. CPU 변화량과 메모리 변화량은 단위 시간마다 변동되는 변화량의 측정값들로 구성된다. 메소드 반응 시간 목록의 요약 정보는 오름 정렬된 메소드 반응 시간 저장 구조로부터 가장 빠른 반응 시간을 보이는 메소드와 가장 느린 반응 시간을 보이는 메소드로 구성되며, 빠르게 찾

위해 메소드 반응 시간 목록의 인덱스로 구성된다. 성능 측정 결과가 같은 경우 상위 컴포넌트를 사용자에게 제공하기 위해 메소드 반응 시간의 평균값과 편차를 계산하여 저장해 둔다. CPU와 메모리 변화량 측정 목록은 CPU와 메모리 별로 변화량이 측정된 총 시간, 가장 높은 변화량, 가장 낮은 변화량, 평균 변화량, 변화량의 편차로 구분되어 저장된다.

컴포넌트 성능 측정 저장 해쉬 구조는 컴포넌트 이름과 인터페이스 이름으로 구성되어 있고, 키 값을 통해 성능 측정된 결과와 요약 정보에 접근할 수 있다. 컴포넌트 성능 측정 저장 해쉬 구조는 컴포넌트 별로 컴포넌트를 구성하고 있는 인터페이스의 성능 측정값과 측정 가공된 요약 정보에 접근할 수 있도록 인덱싱 되고 컴포넌트 레파지토리에 저장된다.

두번째, 컴포넌트 명세 정보는 컴포넌트 성능 측정 장치에 입력된 인터페이스 정보로부터 인터페이스 이름, 인터페이스별 메소드 시그니처 정보를 분석하여 컴포넌트가 위치하고 있는 저장 경로를 키 값으로 컴포넌트 레파지토리에 함께 저장된다.

5. 컴포넌트 추천 방법

비교 성능 로그로부터 추출된 컴포넌트 연관 규칙은 특정 컴포넌트가 성능 측정될 때 사용자에게 자동으로 추출되어 제공된다. 컴포넌트 추천 시 연관성이 높고 사용자가 원하는 성능의 추천 컴포넌트를 선별하기 위해 추출된 연관 컴포넌트 목록 및 성능 측정 결과로부터 계산된 다음의 값을 이용한다.

- 연관 규칙의 지지도와 신뢰도
- 성능 측정 평가 항목의 평균과 편차
- 최소 소요 반응 시간을 보이는 메소드 개수
- 최대 소요 반응 시간을 보이는 메소드 개수

특정 컴포넌트에 대해 연관 컴포넌트를 추천하는 기준은 지지도와 신뢰도가 큰 컴포넌트를 추천하되 같은 지지도와 신뢰도를 갖는 컴포넌트에 대한 순위는 성능 측정 평가 항목의 평균이 높고, 편차가 작으며, 최소 소요 반응 시간을 보이는 메소드 개수가 많고, 최대 소요 반응 시간을 보이는 메소드가 작은 순서로 추천한다.

6. 결론 및 향후 연구

컴포넌트 레파지토리는 컴포넌트의 재사용성을 높일 수 있도록 사용자가 원하는 컴포넌트를 쉽게 찾는 방법이 필요하다. 본 논문에서는 컴포넌트의 기능적인 요소 뿐만 아니라 비 기능 요소인 컴포넌트의 성능을 이용하여 컴포넌트 레파지토리를 구성하고, 함께 성능 비교되는 컴포넌트의 목록을 가공하여 사용자에게 추천할 컴포넌트 패턴을 자동으로 찾아내었다.

이미 개발된 기능과 성능을 갖는 컴포넌트의 재사용성을 높이기 위해서 컴포넌트의 명세 정보, 호환 컴포넌트에 관한 기술 정보, 버전 관리 정보 등이 함께 패키징 되어 유통되어야 하며 레파지토리에 저장되어 관리될 수 있는 표준화된 방법이 함께 고려되어야 한다.

참고문헌

- [1] Rosenbaum, S. and B. DuCastel, "Managing Software Reuse-An Experience Report," In Proceedings of the 17th ICSE, Seattle, Washington, pp. 105-111, April 24-25, 1995.
- [2] Gerhard Fischer and Yunwen Ye, "Personalizing Delivered Information in a Software Reuse Environment," Proceedings of the 8th UM2001, Sonthofen, Germany, pp.178-187, Jul. 13-17, 2001.
- [3] Yunwen Ye and Gerhard Fischer, "Context-Aware Browsing of Large Component Repositories," Proceedings of the 16th ASE, Coronado Island, CA, pp.99-106, Nov. 26-29, 2001.
- [4] Yunwen Ye, "An Active and Adaptive Reuse Repository System," Proceedings of the 34th HICSS, Software Technology Track, Maui, HI, IEEE Press, CD-ROM, 10 pages, Jan. 3-6, 2001.
- [5] R. Agrawal, R. Srikant, "Fast Algorithms for Mining Association Rules," Proc. of the 20th VLDB, Santiago, Chile, Sept 12-15, 1994.