

# 소프트웨어 재사용에서 검색 피드백을 위한 유저 프로필 정의에 관한 연구

김귀정\*

\*건양대학교 컴퓨터학과

e-mail:gjkim@konyang.ac.kr

## A Study on Definition of User Profile for Retrieval Feedback in Software Reuse

Gui-Jug Kim\*

\*Dept of Computer Science, KonYang University

### 요 약

본 연구는 소프트웨어 재사용을 효과적으로 수행하기 위해 사용자 검색 피드백을 지원할 수 있는 유저 프로필을 정의하였다. 컴포넌트 검색을 위해 퍼지 함수를 이용한 신뢰값을 사용하였으며, 사용자 집단의 요구에 능동적으로 반응할 수 있도록 퍼지 함수를 변화시켜 컴포넌트의 검색 우선순위를 변경시키는 방법을 사용하였다. 본 연구에서는 사용자 피드백을 반영하는 퍼지화 함수에 유저 프로필을 적용하여 사용자 등급에 따라 사용자 의견이 시스템에 반영되는 정도를 다르게 해주었다. 본 연구의 사용자 검색 피드백 방법은 퍼지 기법을 적용한 퍼지화 함수와 유저 프로필을 이용하여 시스템을 장기간에 걸쳐 서서히 변화시킬 수 있도록 하였다.

### 1. 서론

본 논문은 컴포넌트의 효율적인 재사용을 위해 후보 컴포넌트들을 우선 순위로 검색하는데 있어서 사용자 등급에 따라 사용자 의견이 시스템에 반영되는 정도를 다르게 하는 방법을 제안하였다. 컴포넌트를 검색하기 위하여 신뢰값을 사용하는데, 신뢰값 계산은 컴포넌트와 질의 사이의 유사도와 퍼지 함수를 이용한다[1,4]. 신뢰값은 컴포넌트와 질의의 일치 정도를 퍼지로 표현하며, 퍼지화 함수는 사용자의 등급을 나타내는 유저 프로필을 이용한다. 즉, 사용자 검색 환경과 사용자 등급, 권한에 따라 컴포넌트의 검색순위 재구성이 이루어지게 된다. 본 논문에서는 사용자 등급의 척도로써 사용자가 가지고 있는 기술을 유저 프로필로 정의하였다. 이 프로필을 벡터값으로 표현하였고, 각 벡터 요소를 사용자의 도메인 기술(domain skill), 작업 기술(task skill), 그리고 전략 기술(strategy skill)로 정의하였다. 초기 퍼지화 함수는 삼각형 모양으로 설정하였으며, 퍼지화 함수는 어떤 등급의 사용자가 어떤 컴포넌트를

선택하느냐에 따라서 그 모양이 수정될 수 있도록 하였다[2]. 이 방법은 단일 질의에 대한 최적화보다는 시스템의 전반적인 향상을 목적으로 한다. 본 연구에서 제안한 검색 피드백 방법은 퍼지 기법을 적용한 퍼지화 함수와 유저 프로필을 이용하여 시스템을 장기간에 걸쳐 서서히 변화시킬 수 있도록 하였다.

본 논문은 다음과 같이 구성된다. 2장에서는 검색 피드백에 대해 설명하고, 3장에서는 유저 프로필을 정의한다. 4장에서는 검색방법과 컴포넌트 검색의 실험결과에 대해 설명하고, 끝으로 5장에서 결론을 맺는다.

### 2. 검색 피드백

검색 피드백은 사용자 집단의 요구에 적응적으로 반응하기 위하여 시스템을 장기간에 걸쳐 서서히 변화시킴으로써 가능하다. 이 방법은 단일 질의에 대한 최적화보다는 시스템의 전반적인 향상을 목적으로 한다. 본 연구에서의 피드백은 클래스 가중치를

변화시키는 것이 아니라 퍼지화 함수의 모양(기술기)을 변화시킴으로써 이루어진다. 각 컴포넌트는 제 자신 고유의 퍼지화 함수를 가지고 있으며, 사용자의 지속적이고 장기적인 컴포넌트 선택여부에 따라 함수값이 변하게 된다[2,3]. 즉, 사용자 검색 환경에 따른 컴포넌트의 검색순위 재구성이 이루어지게 된다. 본 연구에서는 초기 퍼지화 함수를 삼각형 모양의 함수로 설정하였다. 이는 모든 컴포넌트에 초기 함수로 설정된다. 퍼지화 함수는 사용자가 어떤 컴포넌트를 선택하느냐에 따라서 그 모양이 수정될 수 있도록 하였다.

퍼지화 함수의 수정 방법은 다음과 같다. 우선순위에 따라 검색된 컴포넌트 중 사용자가 첫 번째 컴포넌트를 선택하지 않고 k번째 컴포넌트를 선택했을 때, 첫 번째부터 k번째 컴포넌트의 퍼지화 함수는 아래와 같은 식에 의해 변환된다.

$$D_{new}(t) = (1 - \beta)D_{old}(t) + \beta D_{corr}(t)$$

여기에서, 첫 번째부터 (k-1)번째 컴포넌트의 경우에  $D_{corr}(t)$ 는 다음과 같이 수정된다.

$$D_{corr}(t) = (1 + \gamma)t - \gamma, \quad t \leq 0.5$$

$$D_{corr}(t) = 2(1 + \gamma)(1 - t) - \gamma, \quad t > 0.5$$

그리고, 사용자가 선택한 k번째 컴포넌트의  $D_{corr}(t)$ 는 다음과 같이 수정된다.

$$D_{corr}(t) = 2(1 - \gamma)t + \gamma, \quad t \leq 0.5$$

$$D_{corr}(t) = (1 - \gamma)(1 - t) + \gamma, \quad t > 0.5$$

파라메타  $\gamma$ 은 애플리케이션 엔지니어에 의해 조절되며, 시스템의 변화 정도를 나타낸다.  $\gamma$ 의 값이 크면 클수록, 퍼지화 함수의 모양이 더 급격히 변하게 되어 사용자 피드백의 결과가 시스템에 더 빨리 반영되게 된다. 퍼지화 함수  $D_{new}(t)$ 에서의 파라메타  $\beta$ 는 사용자 등급에 따라 시스템 반영 정도를 다르게 해주는 역할을 한다. 함수 변화는 컴포넌트의 지속적이고 장기적인 선택의 결과로써 이루어지기 때문에  $\beta \ll 1$ 이며, 사용자 피드백에 의한 퍼지화 함수 모양의 수정은 매우 서서히 이루어지게 된다.

### 3. 유저 프로필의 정의

사용자 등급을 표현하는  $\beta$ 의 값을 정의하기 위해서는 사용자의 개념을 좀더 상세히 세분화할 필요가 있다. 소프트웨어 개발 환경에서 엔지니어와 개발자는 모두 각각의 고유한 위치와 책임이 주워지게 되며, 이에 따라 재사용 가능한 컴포넌트의 선택이 달라질 수 있다. 이에 본 논문에서는 사용자 등급의

척도로써 사용자가 가지고 있는 기술을 사용자 프로필로 나타내었다[3]. 이 프로필은 벡터값  $(a_1, a_2, a_3)$ 으로 표현되고 0에서 1사이의 값을 갖는다. 각 벡터 요소는 사용자의 도메인 기술(domain skill), 작업 기술(task skill), 그리고 전략 기술(strategy skill)을 나타낸다. 이 사용자 프로필은 고정적인 것이 아니라 현재 작업하고 있는 소프트웨어 도메인에 따라 달라진다. 작업 기술은 사용자가 참가하고 있는 개발 그룹의 수를 말하며, 도메인 기술은 특정 응용 도메인에 참가한 사용자의 프로젝트 수를 말한다. 또한, 전략 기술은 팀 내에서 차지하는 사용자의 책임 정도치를 나타낸다. 시스템에 접속할 때마다 사용자 프로필이 매번 계산되어지고, 이는 피드백을 위한 퍼지화 함수  $D_{new}(t)$ 에서의 파라메타  $\beta$ 를 계산하기 위해 사용된다. 이 사용자 프로필에 각 기술 가중치(WA)를 적용함으로써 사용자 등급  $\beta$ 값을 구할 수 있다. 다음은 파라메타  $\beta$ 를 구하는 식이다.

$$\beta = H \sum_{i=1}^3 w_i a_i$$

제약조건은  $\sum_{i=1}^3 w_i = 1$ 이고,  $H$ 는  $\beta \ll 1$ 를 만족하는 최대값이다. 본 논문에서는  $H=0.1$ 로 하였다.

예를 들어, 6명의 전문가와 12명의 초급자, 그리고 이들을 관리하는 2명의 매니저로 이루어진 개발팀이 있다고 가정하자. 이때, 매니저는 높은 전략 기술과 도메인 기술을 가지고 있지만 작업 기술에는 제한이 있을 것이다. 그러므로 프로필을 벡터값  $(a_1(\text{도메인 기술}), a_2(\text{작업 기술}), a_3(\text{전략 기술}))$ 으로 표현할 때, 매니저의 프로필은  $(a_1=0.8, a_2=0.1, a_3=0.8)$ 로 표현될 수 있다. 반면, 초급 프로그래머는 좋은 작업 기술은 있지만, 도메인과 전략에 대한 주요 지식은 많이 부족할 것이다. 그러므로 초급자의 프로필은  $(a_1=0.1, a_2=0.8, a_3=0.1)$ 로 표현될 수 있다. 또한 고급 엔지니어는 도메인 기술이 가장 뛰어나고 적당한 작업 능력을 가지고 있으며, 전략에 대한 지식은 부족할 것이다. 그러므로 고급자의 프로필은  $(a_1=0.8, a_2=0.6, a_3=0.2)$ 로 표현될 수 있다.

가중치  $w_i$ 는 시스템을 사용하는 개발팀의 각 엔지니어들이 피드백 과정에 어느 정도 역할을 담당할지 결정한다. 매니저가 선택한 컴포넌트와 초급 엔지니어가 선택한 컴포넌트가 재사용에서 같은 정도의 중요성을 갖는다고 볼 수는 없을 것이다. 이를 위해 사용자 프로필에 기술 가중치를 적용하는 것이다.

이와 같은 방법은 재사용 개발환경에서 어떤 컴포넌트가 정책적으로 가장 유용할 것인가를 잘 알고 있는 매니저의 의견을 충분히 반영하고자 하는데 그 목적이 있다. 본 연구에서는 사용자 프로필 벡터의 가중치 값을  $\omega_1=0.3, \omega_2=0.2, \omega_3=0.4$ 와 같이 설정하였다. 전략 기술에 가장 큰 가중치를 부여하여 매니저의 의견이 시스템 피드백에 가장 잘 반영될 수 있도록 하였다. 사용자 프로필에 위의 기술 가중치를 적용하여 사용자 등급  $\beta$ 값을 구하는 식은 다음과 같다.

$$\beta = H \sum_{i=1}^3 w_i \alpha_i$$

$$= 0.1 \times ((0.8, 0.1, 0.8) \times (0.3, 0.2, 0.4))$$

$$= 0.1 \times 0.58$$

$$= 0.058$$

고급자에 대한 사용자 등급  $\beta$

$$= 0.1 \times ((0.8, 0.6, 0.2) \times (0.3, 0.2, 0.4))$$

$$= 0.1 \times 0.44$$

$$= 0.044$$

초급자에 대한 사용자 등급  $\beta$

$$= 0.1 \times ((0.1, 0.8, 0.1) \times (0.3, 0.2, 0.4))$$

$$= 0.1 \times 0.23$$

$$= 0.023$$

4. 컴포넌트 검색

컴포넌트를 검색하기 위하여 질의와 컴포넌트들의 유사도를 계산한다[5]. 먼저 질의와 컴포넌트간의 동치관계를 계산한다. 이 식은 질의에 나타난 클래스와 컴포넌트에 있는 각 클래스 간의 유의값을 반환한다.

$$Eq(Query(u), Comp(v)) = SYNON(Query(u), Comp(v))$$

$u$  : 질의에 있는 질의어 갯수  
 $v$  : 컴포넌트에 있는 클래스 갯수

다음은 질의와 컴포넌트간의 함축관계를 계산한다. 함축관계식에 의해서, 질의에 설정한 질의 중요도가 함축관계에 의해 계산되어진 값보다 작거나 같을 경우 질의와 컴포넌트의 각 클래스에 대한 교환이 이루어진다.

$$Imp(Query(u), Comp(v)) = \max(u(Query(u)), u(Comp(v))) [Eq(u, v)]$$

질의와 컴포넌트 클래스의 만족도를 계산한다. 만족도(satisfaction value)는 질의어와 컴포넌트의 각 클래스가 얼마나 호환성이 있는가를 말해준다.

$$Sat(Query, Comp(v)) = \frac{[\sum_{u=1}^U Imp(u, v) \times Eq(u, v)]}{U}$$

마지막으로 질의와 컴포넌트간의 유사도를 계산한

다. 위에서 만들어진 만족집합에 컴포넌트의 가중치 벡터를 적용함으로써 최종적인 질의와 컴포넌트 간의 유사도를 계산한다.

$$Sim(Query, Comp) = Sat \times W$$

검색된 모든 후보 컴포넌트에 대해서 위와 같은 방법으로 유사도를 계산하여 우선순위 검색이 가능하도록 하였다. 이렇게 계산된 유사도와 퍼지화 함수를 이용하여 검색 피드백 기능을 수행하는 신뢰값(confidence value:CV)을 계산한다. 신뢰값은 사용자에 의해 주어진 질의에 대한 최적의 컴포넌트를 검색해주며, 또한 사용자 피드백을 반영한다. 퍼지화 함수를 이용한 신뢰값(CV)은 다음과 같이 구해진다.

$$CV = L \sum_{i=1}^k D_i CV_i$$

$CV_1, CV_2, \dots, CV_k$ 는 유사집합(Sim)의 요소를 오름차순으로 정렬한 값이다. 여기에서  $L$ 은 지나치게 작은 값이 나올 경우 그 값을 증폭시키는 역할을 한다. 예로서  $L=10, K=3$ 이고 유사도가  $Sim=(0.016, 0.043, 0.05)$ 일 경우, 신뢰값(CV)은 다음과 같이 계산되어 진다.

$$CV = 10 \times (D_1 CV_1 + D_2 CV_2 + D_3 CV_3)$$

$$= 10 \times (0.5 \times 0.016 + 1 \times 0.043 + 0.5 \times 0.05)$$

$$= 0.76$$

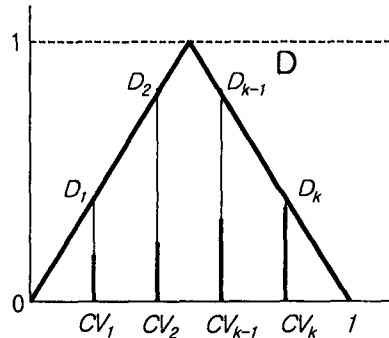


그림 1. 삼각 퍼지화 함수

본 연구에서는 제안한 사용자 검색 피드백을 위한 유저 프로필 정의의 효율성을 다음과 같이 실험하였다. “Document” 컴포넌트, “OleDocument” 컴포넌트, “ClientDocument” 컴포넌트에 대하여 같은 질의를 100번 수행하고 그 CV값의 변화를 실험하였다. 사용자 등급은 매니저, 전문가, 초급자로 나누었으며, 각 등급에 따른 기술 벡터값  $\{a_1, a_2, a_3\}$ 과 이에 따른  $\beta$ 는 표 1과 같다. 사용자 등급이 높을수록,  $\beta$

값이 커져 시스템 반영정도가 높아짐을 알 수 있다.

구분 \ 사용자	매니저	전문가	초급자
사용자기술값 $\alpha$	$\alpha_1=0.8$ $\alpha_2=0.1$ $\alpha_3=0.8$	$\alpha_1=0.8$ $\alpha_2=0.6$ $\alpha_3=0.2$	$\alpha_1=0.1$ $\alpha_2=0.8$ $\alpha_3=0.1$
가중치 $w$	$w_1=0.3$ $w_2=0.2$ $w_3=0.4$	$w_1=0.3$ $w_2=0.2$ $w_3=0.4$	$w_1=0.3$ $w_2=0.2$ $w_3=0.4$
$H$	0.1	0.1	0.1
$\beta$	0.058	0.044	0.023

표 1. 사용자 등급 값

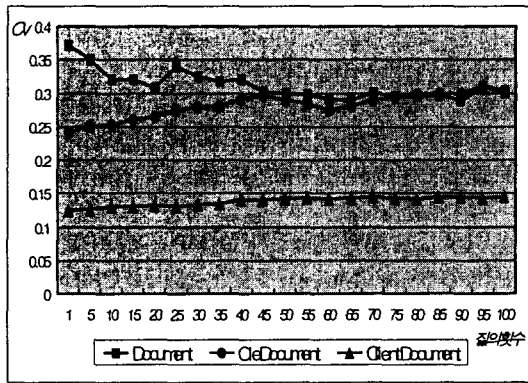


그림 2. 사용자 피드백에 따른 검색

실험에서 100번 질의 중 38번은 매니저에 의해 수행되었으며, 30번은 전문가에 의해 수행되었고 32번은 초급자에 의해 수행되었다. 질의에 대해 사용자에게 의해 선택된 컴포넌트의 비율은 각각 "Document" 컴포넌트가 10%, "OleDocument" 컴포넌트가 80%, 그리고 "ClientDocument" 컴포넌트가 10%를 차지하였다. 그림 2는 각 질의에 대한 컴포넌트 선택에 따른 CV의 변화를 나타낸 것이다. 사용자가 "OleDocument" 컴포넌트를 많이 선택함에 따라 "OleDocument" 컴포넌트의 CV값이 점차로 증가함을 볼 수 있다. 이는 초기 CV값은 "Document" 컴포넌트가 더 높지만, 사용자가 "OleDocument" 컴포넌트를 많이 선택함으로써 검색 우선순위가 더 높아짐을 의미한다.

### 5. 결론

본 연구는 컴포넌트 재사용 환경에서 사용자 검색 피드백을 지원하기 위하여 유저 프로필을 이용하여 개발팀에 참여하는 사용자를 여러 등급으로 분류함으로써 사용자 피드백이 시스템에 미치는 반영정도를 차별화하는 방법을 제안하였다. 유저 프로필은 벡터값으로 표현되고, 각 벡터 요소는 도메인 기술, 작업 기술, 그리고 전략 기술로 정의하였다. 시스템에 접속할 때마다 유저 프로필이 매번 계산되어지고, 이는 피드백을 위한 퍼지화 함수를 계산하기 위해 사용된다. 퍼지 함수는 사용자 요구에 맞는 컴포넌트를 우선적으로 검색해 줄 수 있도록 시스템을 적응적으로 변화시키는 역할을 수행한다. 따라서 본 논문의 유저 프로필을 이용한 검색 피드백 방법은 사용자 집단의 요구에 능동적으로 반응하여 효율적인 재사용 환경을 제공할 수 있다.

### 참고문헌

- [1] A. M. Zaremski, J. M. Wing, "Signature Matching: A Tool for Using Software Libraries," ACM Transaction Software Engineering and Methodology, Vol. 4, No. 2, Apr. 1995. 146-170.
- [2] E. Damini, M.G.Fugini, "Fuzzy Techniques for Software Reuse", In Proceedings of ACM SIG-APP Conference on Applied Computing, Feb. 1996, 552-557.
- [3] E. Damini, M.G.Fugini, C. Bellettini, "A Hierarchy-Aware Approach to Faceted Classification of Object-Oriented Components", The ACM Transaction on Software Engineering and Methodology, Vol.8, No.4, Oct. 1999, 425-472.
- [4] 김귀정, 한정수, 송영재, "컴포넌트 검색을 지원하는 퍼지 기반 시소러스 구축," 한국정보처리학회 논문지, 제10-D권 제5호, pp. 753-762, 8. 2003.
- [5] M. Moormann Zaremski and J. M. Wing, "Signature matching : A tool using software librarsy," ACM Transaction on Software Engineering and Methodology, Vol.4, No.2, pp.146-170, Apr. 1995.