

## 연구개발 생산성 향상을 위한 태스크 유사도 기반 산출물 재사용 추천 프레임워크

남승우<sup>1</sup>, 혼 다네스<sup>1</sup>, 홍장의<sup>2\*</sup>

<sup>1</sup>충북대학교 컴퓨터과학과 석사과정, <sup>2</sup>충북대학교 컴퓨터과학과 교수

## A reuse recommendation framework of artifacts based on task similarity to improve R&D performance

Seungwoo Nam<sup>1</sup>, Horn Daneth<sup>1</sup>, Jang-Eui Hong<sup>2\*</sup>

<sup>1</sup>The master's course, Department of Computer Science, Chungbuk National University

<sup>2</sup>Professor, Department of Computer Science, Chungbuk National University

**요약** 연구 개발 활동은 다양한 기술 정보의 조사 분석 및 기술 보고서 작성 활동들로 구성된다. 연구 개발 활동이 구체화되면서 이전 단계에 작성된, 또는 이전의 유사 프로젝트에서 작성된 관련 기술 문서를 참조하는 일이 많이 발생한다. 본 논문에서는 연구자가 원하는 이전 산출물의 효율적인 재사용을 가능하게 하는 재사용 추천 프레임워크인 RTRF(research task based reuse recommendation framework)를 제안한다. 제안하는 프레임워크는 기존의 유사어 기반 검색 및 재사용에 추가하여 태스크 유사도를 기반으로, 개발자의 연구와 비슷한 흐름을 가지고 있는 다른 개발자가 재사용한 문서를 추천해주어 개발자에게 필요할 수 있는 정보를 제공한다. 사례연구는 연구자들이 기존 문서를 재사용하여 기술동향보고서를 작성하는 과정에서의 효율성을 보이기 위해 수행하였다. RTRF를 이용하여 재사용을 수행하는 경우, RTRF를 이용하지 않는 경우와 비교했을 때 다른 단계의 문서 및 다른 연구분야의 문서를 더 빈번하게 재사용하는 것을 알 수 있었다. 본 논문에서 제안하는 RTRF는 개발자가 저장소에 저장되어 있는 방대한 양의 R&D 문서들 중에서 원하는 문서를 효율적으로 재사용하는 것에 큰 기여를 한다.

**주제어** : 연구개발 산출물, 재사용, 재사용 추천 프레임워크, 태스크 유사도, 재사용성

**Abstract** Research and development(R&D) activities consist of analytical survey and state-of-the-art report writing for technical information. As R & D activities become more concrete, it often happens that they refer to related technical documents that were created in previous steps or created in previous similar projects. This paper proposes a research-task based reuse recommendation framework(RTRF), which is a reuse recommendation system that enables researchers to efficiently reuse the existing artifacts. In addition to the existing keyword-based retrieval and reuse, the proposed framework also provides reusable information that researchers may need by recommending reusable artifacts based on task similarity; other developers who have a similar task to the researcher's work can recommend reusable documents. A case study was performed to show the researchers' efficiency in the process of writing the technology trend report by reusing existing documents. When reuse is performed using RTRF, it can be seen that documents of different stages or other research fields are reused more frequently than when RTRF is not used. The RTRF may contribute to the efficient reuse of the desired artifacts among huge amount of R&D documents stored in the repository.

**Key Words** : R&D artifacts, reuse, reuse recommendation framework, task similarity, reuseability

\*This research was supported by Next-Generation Information Computing Development Program through the NRF of Korea funded by the Ministry of Science and ICT, Korea (NRF-2014M3C4A7030505).

\*Corresponding Author : Jang-Eui Hong(jehong@chungbuk.ac.kr)

Received December 26, 2018

Revised January 18, 2019

Accepted February 20, 2019

Published February 28, 2019

## 1. 서론

재사용은 본래의 목적 또는 다른 기능수행을 위해 기존의 어떤 것을 다시 사용하는 행위 또는 절차이다[1]. 소프트웨어 개발에서의 재사용은 이미 입증되고 신뢰할 수 있는 소프트웨어 컴포넌트를 사용하여 새로운 소프트웨어 시스템을 개발하는 활동을 말하며, 비용 절감과 품질 보증, 생산성 향상 모두에 있어서 중요하다[2-5].

R&D(Research&Development) 프로젝트에서 소프트웨어 산출물의 종류는 기술조사분석서, 연구계획서, 연구개발보고서, 연구 노트와 같은 개발 절차 산출물과 소프트웨어 요구사항 명세서, 소프트웨어 설계서, 소프트웨어 테스트 설계서, 프로그램 코드와 같은 개발 제품 산출물이 있다[6]. 이러한 연구개발 산출물의 재사용은 소프트웨어 컴포넌트의 기원이 되는 소프트웨어 연구 및 개발 산출물을 다른 곳에서 다시 사용하는 행위로써, 이를 통해 개발자가 소프트웨어 컴포넌트를 재사용하기 전에 기능 및 품질 요구사항을 추적확인 할 수 있다[7,8].

연구개발 산출물의 높은 생산성 및 재사용성을 위해 기술문서의 섹션을 재사용 단위로 모듈화하여 마이크로 컴포넌트(mC, micro Component)로 정의한 연구와 이러한 연구개발 문서의 재사용을 지원하기 위한 프레임워크에 관한 연구가 최근 수행되었다[9,10]. 위와 같은 연구들은 신속하고 높은 효율의 연구개발 문서 작성을 가능하게 만들기 때문에 최종적으로 배포되는 소프트웨어의 품질도 보증할 수 있게 된다.

과거 연구개발 산출물의 재사용을 지원하는 기법 및 도구들이 연구되었지만, 다음과 같은 재사용 어려움이 여전히 존재한다.

- 진행 중인 연구를 나타내는 적절한 키워드 조합이 생각나지 않으면 개발자가 원하는 mC를 최종적으로 얻기 힘들다.
- 방대한 양의 mC가 저장소에 저장되어 있을 때, 원하는 mC를 얻기 위해 많은 시간동안 탐색을 해야 한다.
- 진행 중인 연구와 실질적인 관련은 있지만 연구내용의 유사성이 없는 mC들은 검색 결과로 나타나지 않거나 개발자가 탐색하면서 스스로 판단해야 하는 어려움이 있다.

이와 같은 문제들을 해결하기 위해 본 논문에서는

연구내용의 유사성 외에 연구 태스크(research task)의 유사성 비교를 이용하는 재사용 추천 프레임워크를 제안한다. 제안하는 재사용 추천 프레임워크는 향상된 검색 엔진과 확장 패턴 탐색을 보완하는 역할을 수행할 수 있으며, 개발자에게 폭 넓은 연구방향을 제시할 수 있다.

본 논문의 구성은 다음과 같다. 2장에서는 유사도를 이용하여 개발자에게 문서 및 산출물을 추천해주는 기존의 관련 연구들을 확인한다. 3장에서는 관련연구들이 가지는 한계점을 해결하기 위한 mC 재사용 추천 프레임워크를 제안한다. 4장에서는 제안한 프레임워크의 활용 타당성을 검증할 수 있는 사례연구를 보여주며, 마지막 5장에서는 본 논문에 대한 결론과 향후 연구내용을 제시한다.

## 2. 관련연구

S. Hwang은 연구개발 산출물과 관련된 검색 키워드와 저장소에 있는 mC와의 유사도를 통해, 가장 높은 유사도 값을 가지는 후보 mC를 높은 우선순위로 추천해주는 시스템에 대하여 연구하였다[11,12]. 이 연구는 연구개발 문서의 식별자 또는 노트와 같이 비구조적인 요소들로 문서들 간의 유사성을 측정했던 전 연구와 달리, 문서의 구조적 정보를 포함하여 유사성을 측정하였다. 해당 연구는 도메인 주제, 색인 및 연관 규칙 정보를 결합하여 사용자 질의에 대한 신뢰성 높은 검색을 제공하여 산출물 재사용성 향상에 기여하였다.

D. Kim는 S. Hwang의 연구와 같이 향상된 검색엔진 이용으로 해결할 수 없는 다중 mCs의 재사용 방안을 연구하였다[13-15]. Fig. 1은 연구개발 산출물인 소프트웨어 요구사항명세서(SRS)를 예시로 mC와 mC간의 논리적인 관계를 나타낸다. 탐색의 기준이 되는 mCi는 고유 식별자 외에도 개발단계, 기술유형을 속성으로 가지고 있다. 개발자는 현재 자신이 작성중인 문서의 속성과 탐색을 통해 제공되는 mC의 유형을 비교하면서 재사용 결정을 할 수 있다. 그림 1에 표현된 Previous\_mC와 Next\_mC는 동일 문서 내의 mC 탐색을 의미하며, Preceding\_mC와 Succeeding\_mC는 다른 문서 혹은 다른 버전으로의 탐색을 의미한다.

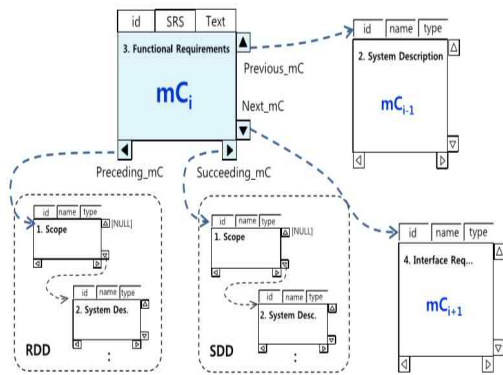


Fig. 1. Logical connections between mCs to provide traceability[15]

S. Deerwester의 연구[16]는 문서를 이루고 있는 문장간의 연관성을 측정하기 위해 LSI(Latent Semantic Indexing)를 사용하여 관련 정도를 나타내었다. LSI를 이용한 접근법은 검색어와 관련된 용어를 기반으로 문서 검색을 향상시키기 위해 용어와 용어의 연관성에서 암시적인 고차원 구조를 활용하는 것이다. D. Shin은 인터넷 정보검색 시스템과 같은 정보검색 시스템을 구축하고, 사용자 질의어 검색에 LSI를 사용하여 관련 문서들을 찾아주는 연구를 수행하였다[17]. 위와 같은 연구에 사용된 LSI는 기준이 되는 문서와 의미적으로 내용이 유사한 문서들을 잘 찾아주는 반면에, 실질적으로 개발자가 필요하지만 의미적으로 내용연관이 없는 문서들을 추천하기 위해 사용되기는 힘들다.

Jaccard 유사도[18]는 비교할 두 이진 벡터들 간의 다른 비트 수를 기반으로 하는 유사성을 측정하는 것이다. S. Niwattanakul의 연구[19]는 정보 검색에 Jaccard 계수를 적용하여 키워드와 색인용어 간의 유사도 측정을 통해, 개발자가 원하는 정보에 대한 접근을 용이하게 하였다. 이는 유사어 존재 및 개발자의 실수로 인한 잘못된 키워드를 이용한 검색에도 높은 안정성을 가지는 정보 검색 엔진을 유지할 수 있다. 하지만 Jaccard 유사도도 LSI와 마찬가지로 문서의 내용을 기반으로 하는 측정법이기에 때문에 비교할 문서의 내용 간 연관성을 도출할 수 없으면 사용될 수 없다.

M. Kersten은 개발자의 생산성 향상을 위해 개발자와 도구의 상호작용을 기반으로 하는 DOI(Degree of Interest) 값을 도입하였다[20]. DOI 값은 산출물과의 상호작용의 빈도 및 상호작용의 최신성의 척도를 기반으

로 한다. 빈도는 산출물을 대상으로 참조하는 상호작용 이벤트의 수를 의미하며, 최신성은 이벤트 스트림에서 산출물과의 첫 번째 상호작용의 위치에 비례하는 값으로 정의된다. 위의 연구에서 현재 수행중인 작업에서 자주 사용된 산출물들은 높은 DOI 값을 가지며, 개발자의 패키지 탐색기 구조의 상단에 위치하게 된다. DOI는 산출물 간의 내용을 기반으로 하는 것이 아닌 작업문맥(task-context)을 기반으로 측정되기 때문에 유사한 작업들을 판별할 수 있는 기준이 될 수 있다.

W. Maalej는 기존의 DOI 값으로 유사도를 측정하는 방식을 더욱 정교화하기 위해 상호작용 이벤트의 지속시간(duration)을 척도 항목에 추가한 FDA 유사도를 제안하였다[21]. F(frequency)와 A(age)는 DOI에서 사용하는 척도와 동일하며 D(duration)은 관심 산출물에 대하여 얼마나 오랫동안 상호작용 했는지를 의미한다. 개발자가 특정 산출물을 작성한 후에, 목록에 있는 다음 작업의 대상이 되는 산출물을 추천 받았을 때, DOI보다 FDA를 기반으로 한 추천 시스템이 개발자 의도에 대하여 더 좋은 적응률을 나타내었다.

작업문맥 기반의 유사도는 의도하는 목적에 따라 쓰이는 방법이 달라진다. 예를 들어, M. Kersten의 연구[20]에서는 개발자가 자주 상호작용한 폴더 및 파일을 상단에 위치시키기 위해 작업문맥 기반의 유사도가 사용된 반면에, W. Maalej의 연구[21]에서는 개발자가 작업 세션에서 한 산출물을 작업한 후에, 목록에 있는 연이어서 작업할 연관 산출물을 추천받기 위해 작업문맥 기반의 유사도가 사용되었다. 하지만 위의 두 연구는 개발자의 생산성 향상을 위한 연구로, 산출물의 재사용성 향상을 위한 방안을 고려하지 않았다. 본 논문에서는 관련 연구에서 사용한 작업문맥 기반의 유사도를 mC 재사용성 향상을 위한 형태로 제시하고 이를 사용하는 프레임워크를 제안한다.

### 3. Research Task와 프레임워크

기존 연구에서 제시된 작업문맥 기반의 유사도를 mC 재사용 추천 프레임워크에 도입하기 위하여 작업의 의미를 연구 태스크로 매핑하였다. 연구 태스크를 나타내는 RT(Research Task)는 R&D 문서 개발자가 새로운 문서를 작성하기 위해 기존 저장소에 있는 mC를 재사용하는 행위를 의미한다. 구체적으로, 개발자는 기존의

산출물을 현재 수행하고 있는 연구개발에 재사용하기 위해 키워드 검색과 확장 패턴을 통한 탐색을 이용한다 [11,15]. 탐색을 이용하여 개발자가 재사용할 mC들을 목록에 하나씩 추가하게 되는데, 이를 현재 수행중인 연구 개발의 방향과 맞는 산출물들을 연관시키는 연구의 한 부분으로 보았다. 제안하는 RT는 식 (1)과 같이 구성되어 있다.

$$RT_i = (id, mCs), i \in N$$

$$mCs = \sum_{i=0}^n mC_i, mC_i \in R_{mC} \quad - (1)$$

식 (1)에서  $N$ 은 자연수 집합을 의미하며,  $id$ 는 개발자가 최종적으로 재사용 mC를 작성중인 문서상에 재사용하기로 결정하였을 때 만들어지는 RT의 고유한 식별자이다. mCs는 개발자가 재사용하기 위해 목록에 추가한 mC들의 집합을 의미하며, 각 mC는 mC 저장소인  $R_{mC}$ 의 원소이다.

RT 정보를 사용하는 RTRF(research task based reuse recommendation framework)는 식 (2)와 같이 정의할 수 있다.

$$RTRF = \{MC, RT, S_{rt}, R_{mC}, H_{rt}, F_{rt}\} \quad - (2)$$

RTRF의 첫 번째 요소인 MC는 문서의 절 단위로 이루어진 R&D 산출물(mC)들의 집합을 의미하며 문서를 작성하는 개발자에 의해 만들어진다. RT는 재사용의 최종 결정에서 생성되는 개별 RT<sub>i</sub>들의 집합이다. S<sub>rt</sub>는 개별 RT<sub>i</sub>의 명세정보를 가지고 있는 집합, 그리고 R<sub>mC</sub>는 mC 저장소를 의미한다. H<sub>rt</sub>는 RT에 속하는 모든 mC들의 과거이력정보 집합을 의미하며, 마지막으로 F<sub>rt</sub>는 재사용을 지원하기 위한 기능들을 의미한다. F<sub>rt</sub>는 식 (3)과 같은 부분요소들로 정의될 수 있다.

$$F_{rt} = \{B_{mC}, R_i, E_{mC}, I_{mC}\} \quad - (3)$$

식 (3)에서 B<sub>mC</sub>는 mC를 생성하는 기능, R<sub>i</sub>는 임포트를 요청하는 기능, E<sub>mC</sub>는 재사용 mC를 추천하는 기능, 마지막으로 I<sub>mC</sub>는 최종 임포트 기능이다. RTRF의 몇 가지 요소들은 D. Kim의 연구[15]를 기반으로 하기

때문에 본 논문에서는 새롭게 추가되거나 확장된 부분만 설명한다. Fig. 2는 본 논문에서 제안하는 RTRF의 개념적 구조를 보여준다.

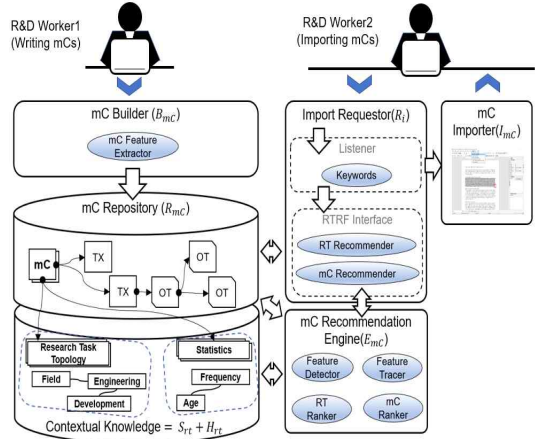


Fig. 2. Conceptual structure of the research task based reuse recommendation framework(RTRF)

RTRF의 사용자 유형은 논리적으로 mC 생성자와 mC 재사용자로 나눌 수 있다. 생성자로부터 mC가 작성되어 입력되면, mC Builder의 mC Feature Extractor는 mC Repository에 mC의 구조 및 내용과 메타정보를 분리하여 저장한다. mC가 가지는 TX는 텍스트 유형, OT는 표 또는 그림과 같은 오브젝트 유형을 의미한다. 각 mC는 Research Task Topology와 Statistics를 메타정보로 가지고 있다. Research Task Topology는 성향이 서로 같은 RT에 해당 mC를 추천하기 위한 정보이다. Fig. 2에서 “Field”는 연구 분야(e.g. Automotive, Avionics, etc.)를, “Engineering”은 소프트웨어 개발 단계(e.g. Requirement, Design, Implementation, Test, others)를, 그리고 “Development”는 mC가 개발과 직접적으로 관련된 내용을 가지고 있는지의 여부를 확인하기 위한 항목이다. Statistics 정보의 구성 항목으로는 “Frequency”와 “Age”가 있다. 이들은 재사용자가 임포트한 mC들을 작성중인 문서의 맥락에 맞게 수정할 때 생기는 이벤트의 빈도수와 시간을 의미한다. 구체적으로, “Frequency”는 재사용자가 수정을 위해 가한 모든 이벤트들의 빈도수를 나타내며 “Age”는 해당 mC에 마지막으로 작용한 이벤트 이후의 초 단위 경과 시간한다. 임포트를 요청하는 재사용자는 Import Requester를

이용하여 원하는 mC를 얻기 위해 키워드 검색을 수행한다. 재사용자는 키워드 검색 후 나오는 결과에서 재사용할 mC를 선택하고, 본인과 유사한 연구문맥을 가지는 다른 사람이 사용한 mC들을 추천받기 위해 RTRF Interface를 이용한다. Fig. 3과 같이 RTRF 사용자 인터페이스의 좌측에는 선택한 mC에 대한 기본정보를 보여주는 mC Info와 자세한 내용을 보여주는 Contents가 있고 우측에는 RT Recommender 기능 수행의 결과물을 보여주는 Recommended RT List와 mC Recommender 기능 수행의 결과물을 보여주는 Recommended mC List로 이루어져 있다. 하단에는 재사용할 것으로 확정된 mC 목록과 목록의 특정 mC 제거, RT Recommend 실행, mC Importer로 전개하는 세 종류의 버튼이 있다.

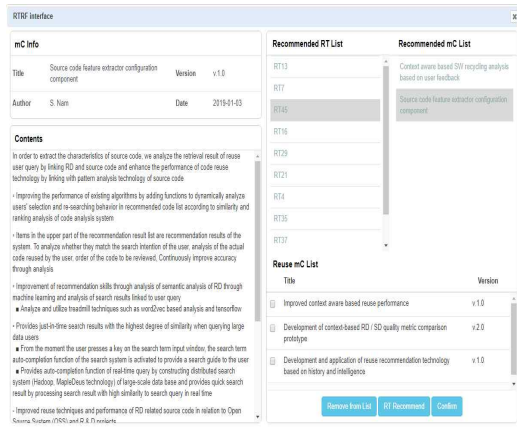


Fig. 3. RTRF User Interface

mC Recommendation Engine은 현재 작성중인 문서의 메타정보를 통해 연구 성향을 파악하는 Feature Detector 기능과 현재 수행중인 연구와 동일한 성향을 가지는 RT를 추적하기 위한 Feature Tracer 기능을 통해 RT와 RT에 속한 mC 후보들을 가져온다. 가져온 RT 및 mC 후보들은 각각 RT Ranker와 mC Ranker를 통해 우선순위를 정하게 된다. 가장 높은 우선순위를 가지는 RT 및 mC는 최상단에 위치하게 된다. 예를 들어, Fig. 3의 오른쪽 상단과 같이 선택된 RT는 후보들 중 세 번째의 우선순위에 있으며 해당 RT에 속하는 두 mC 중에 선택된 mC는 두 번째 우선순위에 있다.

RT Ranker에서 사용되는 RT 유사도 비교와 mC Ranker에서 사용되는 mC와 RT간 연관성 비교는 각각

식 (4), (5)와 같다.

$$Jaccard(RT_i, RT_j) = \frac{|RT_i \cap RT_j|}{|RT_i \cup RT_j|} \quad - (4)$$

$$Relevance(mC_k, RT_l) = \sum_{e \in IN(mC_k, RT_l)} \frac{Age(mC_k, RT_l)}{Freq(mC_k, RT_l)} \quad - (5)$$

식 (4)에서 두 RT의 유사도는 Jaccard indexing을 사용하여 구할 수 있다[22].  $RT_i$ 와  $RT_j$ 의 합집합은  $RT_i$ 에 속하는 mC들과  $RT_j$ 에 속하는 mC들의 중복이 없는 합을 의미하며,  $RT_i$ 와  $RT_j$ 의 교집합은  $RT_i$ 에 속하는 mC들과  $RT_j$ 에 속하는 mC들 중에서 서로 겹치는 것을 의미한다.

식 (5)에서는  $RT_l$ 에 속하는  $mC_k$ 가 다른 mC들보다 얼마나 더  $RT_l$ 와 밀접한지 나타내기 위해 관계성 (Relevance) 모델을 사용했다. 관련성 모델의 값은  $mC_k$ 에 작용하는 모든 상호작용 이벤트에 대하여, 각 이벤트의 빈도수와 마지막 출현 이후의 경과 시간에 대한 정규화 함수를 이용하여 구한다. 본 논문에서는 이벤트 종류를 수정이벤트 하나로 제한하였다.  $mC_k$ 를 임포트한 후 수정횟수가 적고 최근까지 수정하지 않았다면, 해당  $RT_l$ 에 다른 mC들보다 긴밀한 관련이 있다고 본다.  $mC_k$ 가  $RT_l$ 에 거의 그대로 사용되었다면,  $RT_l$ 와 유사한 연구 성향을 가지는 RT에  $mC_k$ 를 높은 우선순위로 추천할 수 있다.

#### 4. 사례연구

사례 연구를 위해 한국전자통신연구원의 웹 사이트에 등록된 R&D문서 및 Google 학술검색에서 수집한 논문 27부를 이용하여 mC들을 생성하였다. 총 생성된 mC의 양은 875개로, 생성된 mC들은 대체로 자율주행 기능을 위한 자동차 임베디드 소프트웨어에 관한 주제를 가지고 있다.

Table 1은 각 문서의 기본정보인 표제 및 제1저자와 메타정보인 'Field' 및 'Engineering', 그리고 생성된 mC의 개수와 수집한 장소를 나타낸다.

Table 1. A list of documents used for case study

no	Title	First Author	Field	Engineering	Number of mCs	Note
1	Trends in Modeling & Simulation Technologies for High-Confidence Software Development	H. Y. Lee	Embedded Software	Other	11	ETRI
2	Independent Object based Situation Awareness for Autonomous Driving in On-Road Environment	S. Noh	Automotive	Other	14	ETRI
3	A Trend of Device Driver Development Tool	C. D. Lim	Embedded Software	Other	19	ETRI
4	Robot Navigation Technology and Its Standardization Trends	W. P. Yu	Robot	Other	9	ETRI
5	Development of Multi-sensor Platform for Context-aware Smart Vehicle(final)	D. S. Kim	Automotive	Other	202	ETRI
6	Technology Trends in Automotive Electronic Control Units	T. M. Han	Automotive	Other	26	ETRI
7	Technology Trend on Embedded Software of Motor Vehicle	J. H. Min	Embedded Software	Other	15	ETRI
8	High Speed V-Link Communication Technology Development for Real Time Control of Automated Driving Vehicle(1year)	H. S. Oh	Automotive	Other	100	ETRI
9	High Speed V-Link Communication Technology Development for Real Time Control of Automated Driving Vehicle(2year)	H. S. Oh	Automotive	Other	96	ETRI
10	State-of-the-Art AI Computing Hardware Platform for Autonomous Vehicles	J. H. Suk	Hardware Platform	Other	11	ETRI
11	Technology Trends of Self-Driving Vehicles	K. H. An	Automotive	Other	11	ETRI
12	Development Trends and Expectation of Three-Dimensional Imager based on LIDAR Technology for Autonomous Smart Car Navigation	G. D. Choi	Automotive	Other	14	ETRI
13	The Current Status of S/W Platform for Advanced Embedded Systems	J. M. Kim	Software Platform	Other	19	ETRI
14	Trends on High-Precision Digital Map for Autonomous Driving Services	J. D. Choi	Automotive	Other	20	ETRI
15	An Approach to Simulate Autonomous Vehicles in Urban Traffic Scenarios	M. C. Figueiredo	Automotive	Other	14	Google
16	An Efficient Method for Testing Autonomous Driving Software against Nondeterministic Influences	P. M. Minnerup	Automotive	Other	85	Google
17	Simulation of city-wide replacement of private car with autonomous taxis in Berlin	J. Bischoff	Automotive	Other	14	Google
18	Development of a Real-Time Driving Simulator for Vehicle System Development and Human Factor Study	S. J. Lee	Automotive	Other	10	Google
19	An integrated architecture for autonomous vehicles simulation	J. L. F. Pereira	Automotive	Other	13	Google
20	Virtual Sensing for Autonomous Vehicle Simulation in Chrono	A. Elmquist	Automotive	Other	16	Google

Table 1. continued

no	Title	First Author	Field	Engineering	Number of mCs	Note
20	Virtual Sensing for Autonomous Vehicle Simulation in Chrono	A. Elmquist	Automotive	Other	16	Google
21	Autonomous Vehicle Simulation Project	S. Lee	Automotive	Other	10	Google
22	Development of Intelligent Vehicle Driving Simulator Based on OpenDS	C. Kim	Automotive	Other	8	Google
23	Basic Simulation Environment for Highly Customized Connected and Autonomous Vehicle Kinematic Scenarios	L. Chai	Automotive	Other	16	Google
24	AutonoVi-Sim: Autonomous Vehicle Simulation Platform with Weather, Sensing, and Traffic Control	A. Best	Automotive	Other	20	Google
25	Simulation Framework for Executing Component and Connector Models of Self-Driving Vehicles	F. Grazioli	Automotive	Other	14	Google
26	CARLA: An Open Urban Driving Simulator	A. Dosovitskiy	Automotive	Other	9	Google
27	A Simulation and Regression Testing Framework For Autonomous Vehicles	C. K. Miller	Automotive	Other	79	Google

Table 1의 목록을 구성하는 문서들은 연구 계획서 및 보고서, 논문들이기 때문에 ‘Engineering’ 항목에 모두 Other 값을 가진다.

#### 4.1 사용자 요구에 부합하는 추천 사례

본 논문에서 제안한 RTRF의 유용성을 확인하기 위해, 기존에 수행했던 연구논문인 ‘Simulation-based Testing of Autonomous Driving Software Using OpenDS’[23]을 기반으로 재사용 mC를 활용하여 자율주행자동차에 대한 기술동향보고서를 작성하였다. 동향 보고서는 3명의 연구자가 위 논문을 읽고 각자 독립적으로 내용을 확장시키면서 보고서를 작성하는 것이다. Table 2는 3명의 연구자가 각자 보고서 작성을 위해 재사용할 문서의 할당 정보를 나타낸다.

Table 2에서 밑줄이 그어진 번호는 다른 연구자들이 재사용한 문서와의 연관관계 형성을 위한 중첩 사용된 문서를 의미한다. 각 연구자는 보고서 작성을 위하여 재사용할 mC들을 결정하고, 선택한 mC들 중에서 특정 내용을 채우기 위해 같이 사용될 수 있는 그룹을 만들었다. 이 그룹은 개별 RT로 저장이 되며, 중복을 제외하고 생성된 1차 RT는 연구자 A는 39개, 연구자 B는 16개, C 연구자는 23개로 총 78개였다. 이 중에서 중첩 사용된

문서에 해당되는 mC들을 두 학생이 서로 다른 mC와 재사용했을 때, 서로의 연구내용이 부분적으로 교집합이 될 수 있도록 포함시킬 mC들을 선택하고 새로운 RT를 생성했다. 추가적으로 중복을 제외하고 14개의 2차 RT가 생성되어, 최종 92개의 RT와 RT에 속하는 246개의 mC들을 도출할 수 있었다.

Table 2. Documents info. used by each member

Member ID	Assigned Reusable documents	Field Ranges
A	1, <u>2</u> , 3, <u>5</u> , <u>6</u> , 8, 9, 11, 12, <u>14</u> , <u>15</u> , <u>16</u>	E/S, Automotive
B	4, 7, <u>14</u> , <u>15</u> , <u>16</u> , 17, 18, 19, 20, <u>21</u> , <u>22</u> , <u>23</u>	Robot, E/S, Automotive
C	<u>2</u> , <u>5</u> , <u>6</u> , 10, 13, <u>21</u> , <u>22</u> , <u>23</u> , 24, 25, 26, 27	H/P, SW/P, Automotive

Fig. 4는 Gephi [24]를 이용하여 생성된 RT와 mC들을 노드로 나타내었고, 단방향 엣지로 RT와 RT에 속하는 mC들의 관계를 나타내었다.

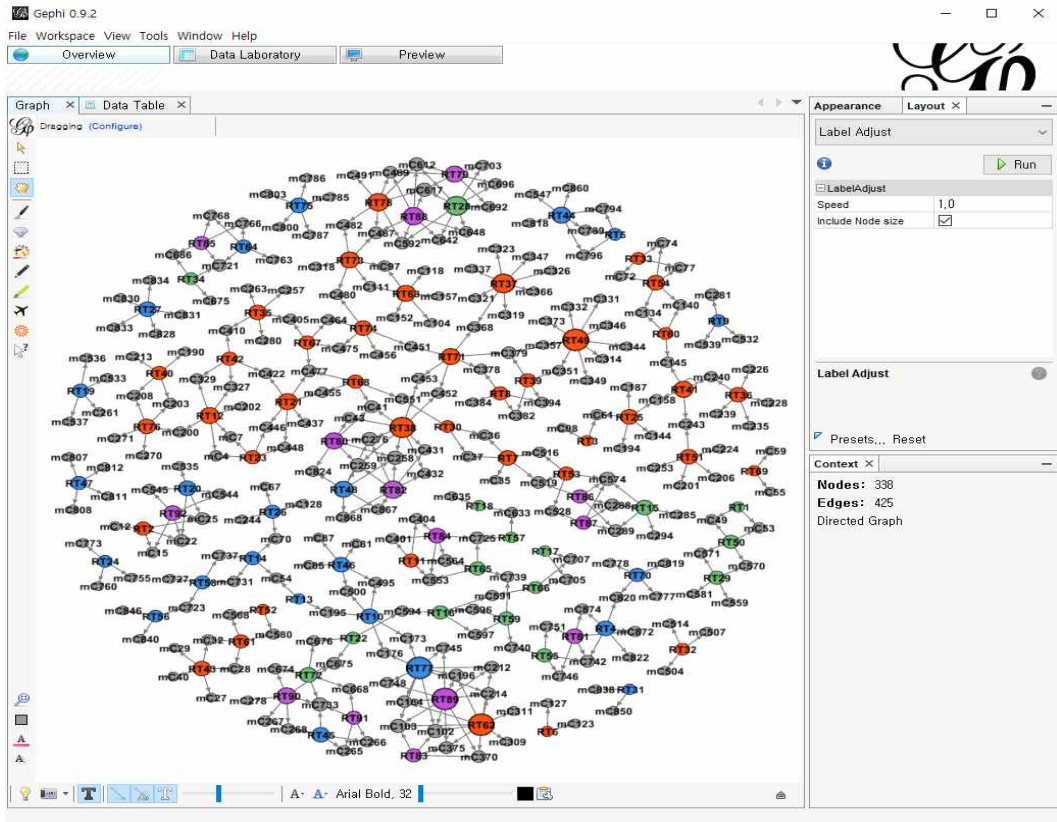


Fig. 4. The generated RTs and mCs with Gephi

그래프의 노드들은 각 연구자들의 RT(붉은색은 A, 초록색은 B, 파란색은 C)의 RT, 보라색은 2차 RT)와 mC를 나타낸다. mC를 많이 재사용한 RT일수록 노드의 크기가 크다. 공통사항으로 3명의 연구자들은 한 번의 재사용 결정을 위해 선택한 mC들은 동일 문서로부터 추출된 경우가 많았으며, 특이사항으로는 연구자 A는 mC의 개수가 많은 3개의 연구 계획서 및 보고서를 재사용했기 때문에 RT 노드의 수, 크기의 수치가 다른 학생들의 RT보다 높다.

2차 RT는 다른 연구자가 재사용한 다른 분야의 문서를 추적을 통해 사용할 수 있게 한다. 예를 들어, 연구자 C가 생성한 RT48(Fig. 4의 중앙에 위치)은 27번째 문서에 해당하는 mC824, mC867, mC868을 사용하며, 'Field' 속성의 값으로 Automotive를 가진다. RT48에서 RT80, RT68, RT21, RT23으로 추적 가능하기 때문에, RT48에서 속하는 mC를 재사용 목록에 추가한 연구자는 1번째

문서에 해당하는 mC4를 추천 받을 수 있다. 즉, 1번째 문서는 연구자 C의 재사용 문서 목록에도 없고 연구분야도 다르지만 RT 개념을 이용하여 추천될 수 있는 것이다.

## 4.2 기존 방법과 제안 방법 비교 사례

'Simulation-based Testing of Autonomous Driving Software Using OpenDS' 논문의 두 저자는 RTRF Interface를 이용하지 않고 확장 패턴 탐색만 이용하여 재사용 mC를 결정하는 경우와 RTRF Interface를 이용하여 결정하는 경우로 사례연구를 진행하였다. 두 저자는 서로 상의하면서 재사용할 mC를 결정했으며, 차례로 RTRF Interface를 이용하지 않는 재사용 mC 결정, RTRF Interface를 이용하는 재사용 mC 결정 절차를 수행하였다.

[I. Using extended pattern search] (1) 두 저자는 먼저 적절한 키워드를 이용하여 확장 패턴 탐색을 시작



할 기준 mC를 선택한다. (2) 선택한 mC에서 같은 문서 내의 mC 혹은 다른 문서의 mC를 확장 패턴을 이용하여 탐색한다. (3) 탐색의 결과로 오는 mC를 재사용 할 것인지 서로 상의한 후에 재사용 목록에 담는다. (4) 앞의 (2)와 (3)의 과정을 반복한다. (5) 더 이상 목록에 담을 mC를 발견하지 않으면 현재 목록에 있는 mC들을 전부 재사용에 이용한다.

**[II. Using extended pattern search with RTRF**

**interface]** (1) 위의 I.(1)과 동일하게 확장 패턴 탐색을 시작할 기준 mC를 선택한다. (2) 앞의 I.(4)의 절차를 따른다. (3) 앞의 I.(5)의 절차를 따른다. (4) 현재 재사용 목록으로 유지되고 있는 mC들의 메타정보를 기반으로 RTRF Interface를 실행한다. (5) 비슷한 연구흐름을 가지고 있는 RT들을 확인한다. (6) 하나의 RT를 선택하고 RT에 속하는 mC들 중에서 연구 내용과 부합하여 재사용될 수 있는 mC들을 목록에 추가한다. (7) 앞의 (2)~(6)의 과정을 반복한다.

위에서 설명한 두 경우의 비교를 위해 사용된 추정 요소를 다음과 같이 정의하였다.

a) 재사용 요청 횟수 NRR(Number of reuse requests) : NRR은 mC들을 재사용하기 위하여 사용자가 발생시킨 재사용 요청 이벤트의 수를 의미한다. 보통 NRR의 값이 클수록 개발자의 문서작성 시간이 단축되었다는 것을 의미하며, 많은 재사용 요청 횟수는 대체로 많은 mC 재사용을 암시하기도 한다.

b) 재사용된 mC들의 평균 ARM(Average of reused mCs) : ARM은 한 번의 재사용 요청이 있을 때에 재사용된 mC들의 평균을 의미한다. 식 (6)은 ARM의 계산식을 나타내며, ARM 값이 클수록 대체로 재사용 효율이 높다고 말할 수 있다.

$$ARM =$$

c) 최종 재사용 결정시간 평균 AFRD(Average of final reuse decision) : AFRD는 한 번의 재사용 요청이 있을 때에 최종 재사용 결정에 사용된 버튼 이벤트 수의 평균을 의미한다. 예를 들어 네 가지의 확장 패턴 탐색 버튼, RT 추천버튼, 추천 mC 선택버튼은 최종 재사용

결정과정에 사용되며, 해당 버튼 이벤트의 수가 많을수록 최종 재사용 결정에 오랜 시간이 걸렸다는 것을 의미한다. 버튼 이벤트 수에 집계되는 버튼의 유형은 DB접속과 관련이 있는 것만 고려하였다. 식 (7)은 AFRD 계산식을 나타내며, AFRD 값이 클수록 대체로 최종 재사용 결정 시간이 오래 걸렸다고 볼 수 있다.

$$AFRD =$$

d) 문서간 연결의 적중률 HRDC(Hit ratio of inter document connection): HRDC는 최종 재사용에 사용된 mC들이 하나의 문서가 아닌 여러 문서에 속한 상황을 의미한다. 본 논문에서 의도하는 재사용은 다른 개발자의 문서, 다른 단계의 산출 문서, 다른 연구분야의 문서 등의 재사용이기 때문에 HRDC를 추정요소에 포함시켰다. 식 (8)은 HRDC의 계산식을 나타내며, HRDC의 값이 클수록 의도한 재사용성이 향상된 것을 의미한다.

$$HRDC =$$

e) 연구분야 간 연결의 적중률 HRFC(Hit ratio of inter research field connection): HRFC는 최종적인 재사용에 사용된 mC들이 하나의 연구분야가 아닌 여러 연구분야의 문서에 속한 상황을 나타낸다. 식 (9)는 HRFC의 계산식을 나타내며, HRFC의 값이 클수록 의도한 재사용성이 향상된 것을 의미한다.

$$HRFC =$$

Table 3은 사례연구의 결과로 오는 추정 요소들의 값을 보여준다.

사례 연구에 참여한 두 저자는 RTRF Interface를 사용하여 재사용 mC들을 선택했을 때가 사용하지 않았을 때보다 한 번의 재사용 요청에서 더 원하는 내용을 얻을 수 있다고 응답했다. 이에 대한 내용은 NRR과 ARM의 값을 통해 알 수 있다.

Table 3. Estimation results from the case study

	Extended Pattern Search	Extended Pattern Search with RTRF
--	-------------------------	-----------------------------------

NRR	9	5
ARM	31/9 = 3.4	42/5 = 8.4
AFRD	114/9 = 12.7	267/5 = 53.4
HRDC	3/9 = 30%	4/5 = 80%
HRFC	1/9 = 10%	2/5 = 40%

세 번째 행의 AFRD 값은 기존의 확장패턴탐색 버튼에서 RTRF Interface 버튼이 추가되었기 때문에 RTRF Interface를 사용하여 탐색한 경우가 더 큰 것을 알 수 있다. 하지만 두 저자는 최종적인 재사용 결정시간이 느려지지만 다양한 mC들을 추천받을 수 있어 좋다고 하였고, 추천 mC가 본인들의 연구와 비슷한 흐름을 가지고 있기 때문에 mC 하나하나 그냥 지나칠 수 없어서 놀라보게 된다고 응답하였다.

2배 이상 차이가 나는 AFRD는 최종 재사용 결정시간의 실제 차이를 의미하지 않았다. HRDC와 HRFC는 본 연구의 핵심 추정요소로 다른 문서 및 다른 연구분야이지만 개발자가 필요할 수도 있는 정보를 개발자에게 얼마나 잘 도달해줄 수 있는지에 대한 지표이다. HRDC와 HRFC의 값은 제안하는 RTRF를 이용했을 때가 이용하지 않았을 때보다 월등히 높았다.

## 5. 결론

재사용은 개발자가 이미 품질이 보증된 누군가의 컴포넌트를 사용함으로써, 효율적인 소프트웨어 시스템의 개발을 가능하게 하는 활동이다. 기존의 소프트웨어 컴포넌트가 기능 및 품질요구사항에 맞아야 재사용되었던 것에 반해, R&D문서는 다른 개발자, 다른 단계 및 다른 연구분야 일지라도 개발자가 수행중인 연구흐름과 유사하면 재사용될 수 있다. 본 논문은 R&D문서 단위의 크기축소가 재사용성 향상에 어느 정도 한계가 있음을 드러냈으며 근본적인 재사용성 향상에 기여할 수 있는 RT 개념 및 RT를 사용하는 프레임워크를 제안하였다. 또, 사례연구를 통해 제안하는 프레임워크 사용의 타당성을 검증하였다. 향후연구로는 변경되는 mC에 대한 버전관리를 고려한 RT유사도 및 관계모델 계산법을 확장할 것이다.

## REFERENCES

- [1] Wikipedia contributors. (2018). *Reuse*. Wikipedia. <https://en.wikipedia.org/w/index.php?title=Reuse&oldid=73286698>
- [2] D. K. Kim, E. Song, J. Ryoo & Y. R. Reddy. (2017). Special issue on software reuse. *Software Practice and Experience*, 47(7), 941-942.
- [3] J. S. Park, J. J. Kwon, J. E. Hong & M. C. (2013). Software Architecture Recovery for Android Application Reuse. *Journal of Convergence for Information Technology*, 3(2), 9-17.
- [4] Y. Choi & J. E. Hong. (2017). Designing Software Architecture for Reusing Open Source Software. *Journal of Convergence for Information Technology*, 7(2), 67-76.
- [5] J. H. Kim. (2014). Support of Reuse in Backlog Refinement with Backlog Factoring. *Journal of Digital Convergence*, 12(12), 337-343.
- [6] Amarmend, E. C. Lee, J. W. Lee & B. Lee. (2016). Describing Activities to Verify Artifacts(Documents and Program) in Software R&D. *Journal of Internet Computing and Services(JICS)*, 17(2), 39-47.
- [7] J. P. Kim, D. H. Kim & J. E. Hong. (2011). Techniques to Support Low-Power Characteristics in Embedded Software Development Process. *Journal of Convergence for Information Technology*, 1(1), 55-65.
- [8] S. K. Kim & J. E. Hong. (2016). Application of Safety Analysis and Management in Software Development Process. *Journal of Convergence for Information Technology*, 6(1), 7-15.
- [9] S. Kim, et al., (2012). Toward Offline Contents Based Software R&D Support System. *Advances in Computer Science and Ubiquitous Computing*, 1097-1101.
- [10] D. Kim & J. E. Hong. (2018). Improving software artifacts reusability based on context-aware reuse technique. *Journal of Theoretical and Applied Information Technology*, 96(2), 523-533.
- [11] S. Hwang, K. Seo, W. Ryu & Y. Nam. (2017). System for the Researcher Map to Promote Convergence Research. *Advances in Computer Science and Ubiquitous*, 1168-1173.
- [12] S. Hwang, Y. Lee & Y. Nam. (2018). System for extracting domain topic using link analysis and searching for relevant features. *Journal of Ambient Intelligence and Humanized Computing*, 1-13.
- [13] D. Kim, S. K. Kim, W. Jung & J. E. Hong. (2016). A Context-Aware Architecture Pattern to Enhance the Flexibility of Software Artifacts Reuse. *Advances in Computer Science and Ubiquitous Computing*, 654-659.
- [14] D. Kim & J. E. Hong. (2018). The microComponent and Its Extension Patterns for Flexible Reuse of Software Artifacts. *Advances in Computer Science and Ubiquitous*, 1084-1090.

[15] D. Kim, S. Nam & J. E. Hong. (2018). A dynamic control technique to enhance the flexibility of software artifact reuse in large-scale repository. *The Journal of Supercomputing*, 1-31.

[16] S. Deerwester, S. T. Dumais, G. W. Furnas, T. K. Landauer & R. Harshman. (1990). Indexing by latent semantic analysis. *Journal of the American Society for Information Science*, 41(6), 391-407.

[17] D. Shin. (2000). *A Study on Content-Based Information Retrieval System using LSA*. Master dissertation. Seoul National University, Seoul.

[18] P. N. Tan. (2006). *Introduction to Data Mining*. Boston: Pearson Addison-Wesley.

[19] S. Niwattanakul, J. Singthongchai, E. Naenudorn & S. Wanapu. (2013). Using of Jaccard Coefficient for Keywords Similarity. *Proceedings of the International MultiConference of Engineers and Computer Scientist 2013*, 1-5.

[20] M. Kersten, Mik, G. Murphy & G. C. (2006). Using task context to improve programmer productivity. *Proceedings of the 14<sup>th</sup> ACM SIGSOFT International Symposium on Foundations of Software Engineering*, 1-11.

[21] W. Maalej, M. Ellmann & R. Robbes. (2017). Using contexts similarity to predict relationships between tasks. *The Journal of Systems and Software*, 128, 267-284.

[22] Wikipedia contributors. (2018, December 2). *Jaccard index*. In Wikipedia, The Free Encyclopedia. Retrieved 12:39, January 6, 2019, from [https://en.wikipedia.org/w/index.php?title=Jaccard\\_index&oldid=871576429](https://en.wikipedia.org/w/index.php?title=Jaccard_index&oldid=871576429)

[23] C. E. Lee, Y. Yun & J. E. Hong. (2018). Simulation-based Testing of Autonomous Driving Software Using OpenDS. *Proceedings Convergence Society for SMB*, 25(2), 541-543.

[24] Wikipedia contributors. (2018). *Gephi*. Wikipedia. <https://en.wikipedia.org/w/index.php?title=Gephi&oldid=875139913>



혼 다니스 (Horn, Daneth)

[학생회원]

- 관심분야 : Software Quality, Software Safety, Cyber Physical Systems
- E-Mail : swnam@selab.cbnu.ac.kr



홍 장 의 (Hong, Jang-Eui)

[종신회원]

- 2015년 2월 : Univ of Royal, Phnom Penh 컴퓨터학과(학사)
- 2017년 9월 ~ 현재 : 충북대학교 대학원 컴퓨터학과(석사과정)

- 관심분야 : Software Quality, Software Safety, Cyber Physical Systems
- E-Mail : swnam@selab.cbnu.ac.kr



- 2001년 2월 : KAIST 전산학과(박사)
- 2003년 3월 : 국방과학연구소 선임연구원
- 2004년 3월 : ㈜솔루션링크 기술연구소장
- 2004년 3월 ~ 현재 : 충북대학교 소프트웨어학과 교수
- 관심분야 : Software Quality, Software Safety, Model-Based SW Engineering, Cyber Physical Systems
- E-Mail : jehong@chungbuk.ac.kr

남 승 우(Nam, Seungwoo)

[학생회원]

- 2017년 2월 : 충북대학교 컴퓨터공학과(학사)
- 2017년 3월 ~ 현재 : 충북대학교 대학원 컴퓨터학과(석사과정)