버그리포트를 이용한 정보검색 기반 테스트케이스 우선순위화 기법

안준, 염창선, 김정호, 이은석
성균관대학교 전자전기컴퓨터공학과
e-mail : {ahnjune,klausyoum,jeonghodot,leees}@skku.edu

A Technique for Test Case Prioritization based on IR using Bug Report

June Ahn, Changsun Youm, Jeongho Kim, Eunseok Lee
Dept. of Computer Science and Engineering, Sungkyunkwan University

요 약

비용 효율적인 소프트웨어 유지보수 방안에 대한 기대가 높다. 본 논문에서는 유지보수 비용을 감소시키기 위해 회귀 테스트에 사용되는 테스트케이스를 효과적으로 우선순위화하는 방안을 제안한다. 테스트케이스를 우선순위화하는 방법으로는 코드의 커버리지를 이용해 테스트케이스의 우선순위를 높이는 방법과 모델 기반 테스트케이스 우선순위화 방법 등 여러 가지 방법이 제안되어 왔다.

본 논문에서는 소스코드, 커밋 로그와 버그리포트의 정보를 이용해 정보검색 기반의 테스트케이스 우선순위화 기법을 제안한다. 변경된 소스코드 이력은 새로운 기능의 업데이트 유무를 확인 할 수 있으며, 결함으로 수정된 파일을 추측할 수 있다. 버그 리포트는 소스코드의 결함에 대한 정보를 담고 있다. 제안한 방법의 유효성을 확인하기 위해 오픈소스 프로젝트(Joda-Time, Commons-Lang)를 이용해 실험을 진행하였다. 실험을 통해 소스코드, 커밋 로그와 버그리포트로 테스트케이스 우선순위화 방법의 유효성을 확인했으며, 버그리포트를 적용해 테스트케이스 우선순위화 기법을 이전 연구에 비해 최대 8% 향상된 결과를 확인 할 수 있었다.

1. 서론

비용 효율적인 소프트웨어 유지보수 방안에 대한 기대가 높다. 테스트케이스 우선순위 방법은 전통적으로 커버리지 기반 테스트케이스 우선순위화 방법과 모델 기반 테스트케이스 우선순위화 방법에 대한 활발한 연구가 진행되어져 왔다[1].

Ripon K. Saha[2], 연구에서 제안하는 테스트케이스 우선순위화 방법으로는 소스코드의 변경이력을 이용한 정보검색 기반의 테스트케이스 우선순위 기법을 제안한다. 이전 버전과 새로운 버전 각각의 소스코드와 커버리지 정보를 통해 찾은 정보들을 데이터 마이닝 기법인 TF-IDF (Term Frequency-Inverse Document Frequency)을 사용해 계산하며, 분석된 데이터를 기반으로 변경된 소스코드와 연관된 테스트케이스의 우선순위를 높여 테스트의 효율성 및 정확도를 향상시키는 연구를 제안하고 있다.

본 논문에서는 유지보수 비용을 감소시키기 위해 회귀 테스트(Regression Test)에 사용되는 테스트케이스를 효과적으로 우선순위화하기 위한 방안을 제안한다. 제안하는 테스트케이스 우선순위화 방법은 각 프로젝트의 버전 별 수정 전/후의 소스코드 이력, 테스트케이스, 버그 리포트 및 커밋 로그의 데이터를 수집한다. 이후 수집된 데이터는 노이즈가 많음으로 데이터 품질 향상과 데이터의 일치성을 위해 전처리 과정을 수행한다. 다음으로 텍스트 마이닝 기법인 TF-IDF 기법을 사용해 변경된 소스코드 이력과 버그리포트 정보를 해당 각 프로젝트의 테스트케이스와 비교해 테스트케이스의 우선순위를 결정한다.

본 논문의 구성은 제 2장에서 버그리포트와 정보검색 기반의 테스트케이스 우선순위에 사용되는 관련연구에 대한 배경을 다루며, 제 3장에서 제안한 버그리포트를 이용한 테스트케이스 우선순위화 기법을 소개한다. 제 4장에서 제안한 방법에 대한 실험 환경 및 실험 내용을 설명하며, 제 5 장 관련연구와의 성능 비교를 통해 제안한 알고리즘을 평가한다. 제 6장에서 결론 및 향후 연구를 정리한다.

2. 관련연구

1) 버그리포트

발생된 결함을 해결하고 사후 동일한 결함의 예방 및 해결을 위해 버그리포트를 작성하게 된다[3, 4]. 이러한 버그리포트를 기록하고 관리하는 것이 BTS(Bug Tracking System)이다. BTS는 Mantis, Bugzilla, Trac 등 많은 도구들이 있으며, 해당 도구를 이용해 버그 생성 및 버그가 해결되기까지의 일련의 과정들을 확인할 수 있다.

BTS의 버그리포트에는 다음과 같은 내용들을 작성할 수 있다. 다음 내용을 참고하여 개발자는 버그를 식별하고

버그를 수정한다.

- 버그 ID
- 버그 요약 내용
- 버그 설명
- 스택 트레이스
- 테스트 환경 등

본 논문에서는 소프트웨어 결함과 직접적인 연관이 있는 버그리포트를 이용해 버그와 연관된 테스트케이스의 우선순위를 높이는 방법을 제안한다.

2) 형상 관리 시스템

형상 관리 시스템은 소스코드의 이력을 관리하는 시스템으로 소스코드들을 효율적으로 관리하기 위해 개발된 시스템이다. 상용 시스템으로는 IBM Rational ClearCase, Perforce, PTC Integrity가 있으며, 오픈소스로는 Subversion(SVN), CVS, Git이 있다. 형상관리 시스템에는 개발에 필요한 문서, 소스코드, 개발 도구 등 소프트웨어를 개발하면서 도출되는 결과물들을 모두 포함하고 있으며, 결과물을 업데이트할 때 생성되는 일련의 정보를 커밋 로그(Commit Log)라 한다. 개발자들은 형상 관리 시스템을 통해 결함이 발생했을 때, 개발자들이 결함을 추적하는데 중요한 보조 역할을 한다. 버그리포트에 기반으로 인지한 결함을 형상 관리 시스템의 커밋 로그를 통해 변경된 파일과 코드의 위치를 확인하게 되고, 의심되는 파일을 중심으로 테스트 및 결함을 수정한다.

3) 정보검색 기법(Information Retrieval)

TF-IDF은 정보검색 및 텍스트 마이닝을 위해 문서의 단어들 간의 중요도를 평가하기 위해 만들어진 계산 모델이다. TF-IDF의 점수를 통해 문서들 간의 순위를 결정할 수 있으며, 유사 문서들의 그룹을 찾는 문서군집화를 용이하게 한다. TF-IDF 점수가 클수록 단어가 속해 있는 문서에 특이성이 높다는 것이며, TF-IDF 점수를 통해 질의문과 관련있는 문서를 추출하는 척도로 사용할 수 있다.

$$tf(t,d) = \frac{f(t,d)}{\max f(w,d) : w \in d} \quad (1)$$

$$idf(t,D) = \log \frac{|D|}{|d \in D : t \in d|} \quad (2)$$

$$TFIDFScore(t,d,D) = tf(t,d) \times idf(t,D) \quad (3)$$

식 1 의 TF는 한 문서의 단어 출현 횟수를 나타내는 수식이며, 식 2 의 IDF는 한 단어가 전체 문서에 n개의 문서에 출현하는지를 나타내는 수식이다. 식 3 은 TF-IDF 점수를 구하기 위한 식을 표현하고 있다. TF-IDF의 점수를 구하기 위해 TF(Term Frequency)값과 IDF(Inverse Document Frequency)값을 곱해 문서의 가중치를 구하게 된다. 식 3 에서 TF 값은 문서 내 특정 단어의 출연 빈도를 의미하며, IDF 값은 문서 집합에 포함되어 있는 전체 문서를 특정 단어가 나타난 문서의 빈도로 나눈 것이다. 본 논문에서는 문서들의 집합(소스코드, 버그리포트, 커밋 로그)들의 단어들을 추출해 문서들의 Score 점수를 계산하기 위해 TF-IDF을 사용해 실험한다.

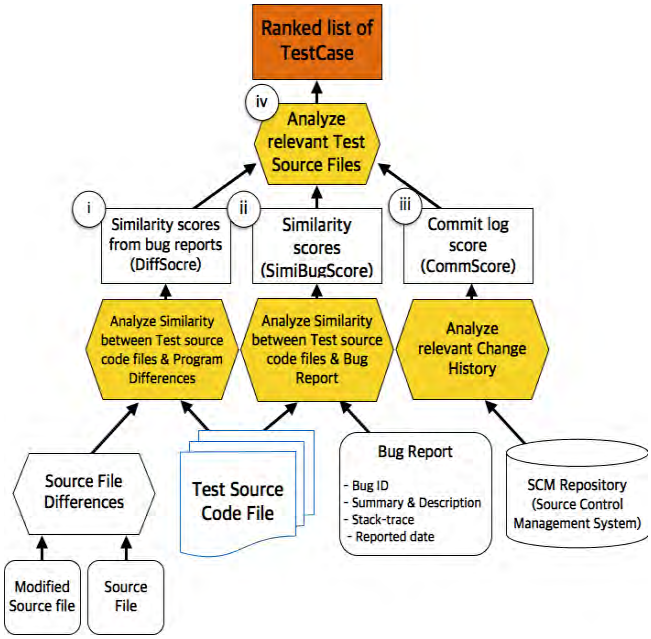
4) 정보검색 기반 테스트케이스 우선순위 기법

정보검색 기반 테스트케이스 우선순위 방법은 소스코드, 테스트케이스 정보를 이용해 테스트케이스를 우선순위화하는 기법이다. 이 기법은 이전 버전의 소스코드의 이력과, 결함으로 인해 수정되거나 새로운 기능이 추가되어 업데이트된 소스코드의 변경된 이력을 추출한다. 질의문으로는 변경된 소스코드 이력이 사용되며, 대상문서로는 테스트케이스들을 정보검색 기법인 TF-IDF를 사용해 점수를 계산한다.

3. 버그리포트를 이용한 정보검색 기반 테스트케이스 우선순위화 방법

본 논문에서는 기존 정보검색 기반의 테스트케이스 우선순위 방법에 버그리포트를 추가한다. 제 2장에서 소개한 버그리포트를 통해 변경된 이력의 실제 결함으로 수정된 파일의 테스트케이스의 우선순위를 향상 시킬 수 있다. 이를 위해 제안하는 방법은 정보검색 기반 테스트케이스 우선순위화를 위해 소스코드의 변경 이력, 버그리포트, 커밋 로그 총 3가지의 팩터를 사용한다. 각 팩터들을 TF-IDF로 계산하기 위해, 각 팩터들에 전처리 작업을 수행한다.

전처리 작업은 총 3 단계의 전처리 단계를 거치게 된다. 정규화(Normalization), 불용어 제거(Stopword Removal)와 어근화(Stemming) 작업을 수행하게 된다. 정규화 작업



(그림 1) 버그리포트를 이용한 정보검색 기반 테스트케이스 우선순위화 기법의 분석 흐름

은 “checkDataType″과 같은 단어를 ”check″, ″Data″, ″Type″으로 나누는 작업을 한다. 불용어 제거 작업은 ″a/an″, ″on″, ″the″와 같은 불필요한 단어들을 제거 한다. 마지막으로 어근화 작업은 각기 다른 단어들 “fishing”, “fished”, “fisher”단어들을 “fish”과 같은 형태로 만든다. 전처리 작업을 완료하게 되면 TF-IDF을 통해 각각의 팩터들의 점수를 계산하게 된다.

그림 1 은 본 논문에서 제안하고 있는 버그리포트를 이용한 정보검색 기반 테스트케이스 우선순위 방법에 대해 설명하고 있다. 버그리포트를 이용한 정보검색기반 테스트케이스 우선순위화 기법은 4단계의 계산과정을 거치게 된다. (i) 소스코드 변경 이력과 테스트케이스를 통해 TF-IDF 기법을 사용해 계산한 값을 DiffScore이다. (ii) 본 논문에서 제안한 버그리포트와 테스트케이스의 가중치를 계산한 값을 BugScore이다. (iii) 커밋 로그의 가중치를 계산한 값을 CommScore라고 한다. (iv) 마지막으로 각각 계산된 DiffScore, BugScore와 CommScore를 통해 최종적으로 테스트케이스 우선순위화에 사용되는 점수를 계산하게 된다.

$$Score = (1-\alpha)\times N(DiffScore) + \alpha\times N(BugScore) + \beta\times N(CommScore) \quad (4)$$

식 4 는 각각 계산된 펙터들을 통해 최종적으로 테스트케이스의 우선순위화를 하기위해 사용하는 계산 수식이다. 각 점수의 영향도를 분석/조절하기 위해 α, β 파라미터를 사용한다. α는 DiffScore와 BugScore의 영향도 값을 조절하며, β는 DiffSocre, BugScore의 조합 값과 CommSocre 간의 영향도를 조절한다. 계산된 값을 통해 테스트케이스의 순위를 매기게 된다.

4. 실험

버그리포트를 이용한 정보검색 기반 테스트케이스 우선순위를 측정하기 위해 오픈소스 프로젝트를 이용해 실험을 한다.

1) 대상 프로젝트

표 1 은 본 논문의 실험에 사용된 오픈소스 프로젝트들로 이전 정보 검색 기반 테스트케이스 우선순위 연구에서 실험에 활용했던 2개의 오픈소스 프로젝트 Joda-Time, Commons-Lang을 대상으로 실험한다.

<표 1> 실험대상 프로젝트

Project	Description	Version	Bugs (개)	Source Files (개)
Joda-Time	날짜와 시간 간격을 정의	2.2-2.3 2.3-2.4 2.4-2.5 2.5-2.6 2.6-2.7 2.7-2.8	7 7 18 7 2 3	157 164 165 165 165 166
Commons-Lang	유틸리티들을 모아놓은 클래스들의 집합	3.1-3.2 3.3-3.4	46 17	83 83

2) 평가항목

제안한 버그리포트를 이용한 정보검색 기반 테스트케이스의 우선순위화에 대한 효과를 측정하기 위해, 발견된 결함의 평균 백분율(APFD : Average Percentage Of Faults Detected)를 사용하였으며, 이를 통해 테스트케이스 우선순위화의 정확도를 측정한다[5, 6, 7].

$$APFD = \frac{T \bullet F_1 + T \bullet F_2 + \ldots T \bullet F_n}{nm} + \frac{1}{2n} \quad (5)$$

식 5 의 T는 n개의 테스트케이스로 이루어지는 테스트 스위트이며, 테스트 스위트는 테스트케이스들을 같은 환경에 따라 구분해 놓은 테스트케이스들의 집합이다. F는 m개의 결함을 포함하는 결함의 집합이다. $T \bullet F_n$는 사전에 정해진 테스트 순서를 따르는 T에 결함을 발견하는 첫 번째 테스트케이스를 의미한다.

5. 실험 결과

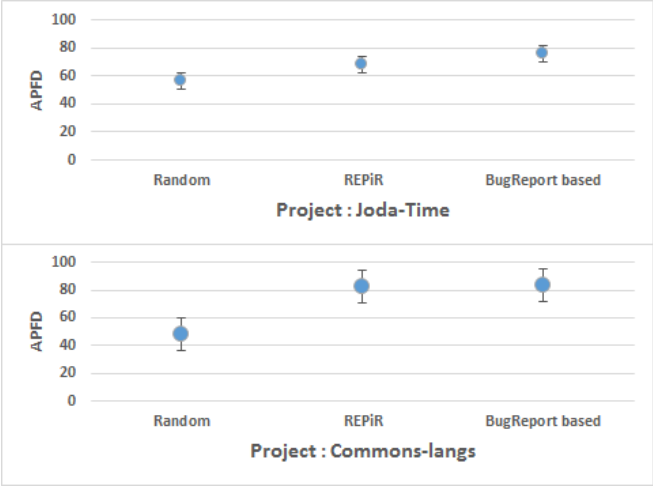
표 2는 연구에서 제안한 테스트케이스 우선순위에 대한 평가 결과를 나타낸다.

1) 버그리포트를 이용한 정보검색 기반 테스트케이스 우선순위 기법의 실험 및 분석

기존 정보 검색 기반의 테스트케이스 우선순위화 기법에서는 버그리포트를 사용한 연구는 없었다. 본 실험으로 버그리포트의 영향을 분석하고 실험을 통해 버그리포트의 유용성을 확인한다. 표 2 에는 3 장에서 설명한 α, β 가중치를 조절해 버그리포트, 소스코드 변경이력의 각각의 최적의 가중치로 평가한다. 그림 2 와 같이 우선순위를 랜덤으로 한 테스트케이스와 비교한 결과, 각 프로젝트 별 APFD 값이 20.0%, 35.5%의 향상된 결과를 나타냈다. 또한 기존 연구 REPiR과 비교한 결과 최대 8% 포인트 향상된 결과를 보였다. Joda-Time 프로젝트 Version 2.2~2.8에 등록된 버그리포트 44개를 이용해 기존 연구 REPiR을 비교한 결과 8% 포인트 향상된 결과를 보였으며, Commons-Lang 프로젝트 Version 3.2~3.4에 등록된 버그리포트 63개를 이용해 테스트케이스 우선순위화의 정확

<표 2> 평가 결과

Project	APFD		
	Random	REPiR	BugReport-based
Joda-Time	56.0%	68.0%	**76.0% VS random +20% VS REPiR 8%**
Commons-Lang	48.0%	82.4%	**83.5% VS Random +35.5% VS REPiR +1.1%**



(그림 2) 버그리포트를 이용한 정보검색기반 테스트케이스 우선순위화 기법의 프로젝트별 효과

도를 1.17% 포인트의 미미한 효과를 확인했다. 본 실험결과를 통해 버그리포트를 통해 회귀 테스트에서 테스트의 효율을 높이고, 테스트 소요시간을 낮추는 효과를 거두게 되었다.

2) 버그리포트의 효과를 높이기 위한 버그리포트 형식 제안

오픈소스 프로젝트인 Joda-Time, Commons-lang에 버그 리포트를 이용한 정보검색 기반 테스트케이스 우선순위화 기법을 적용한 결과, Joda-Time 프로젝트에서는 유의미한 결과를 담고 있었으나, Commons-lang 프로젝트에서는 효과가 미미했다. 이는 Commons-lang 프로젝트에 사용된 버그 리포트의 완성도가 낮은 것이 원인으로 판단된다. 그러므로 개발이 완료되고 추후 유지보수를 효과적으로 하기 위해 버그리포트의 형식을 표 3 과 같이 제안한다. 버그 리포트에 작성되는 내용 중 모든 부분이 중요하지만 “제목”, “내용”, “첨부파일” 3가지 내용은 결함을 해결하는데 있어 필수적인 요소이다. 버그리포트의 제목에는 결함을 명확하게 파악할 수 있는 단어로 간략히 작성하며, 내용 영역에는 결함의 재현방법에 대한 스탭과 스택트레이스 정보등 결함 발생의 원인을 파악할 수 있도록 자세히 기술한다. 첨부파일에는 결함이 발생한 원인을 담고 있는 로그 및 덤프 파일을 첨부해 개발자가 버그리포트를 통해 정확하게 결함을 수정할 수 있도록 하는 버그리포트 형식을 제안한다.

<표 3> 제안하는 버그리포트 형식

Bug ID	#175
Title	[ProjectName] Keyword
Status	Fixed
Platform	JAVA_SE
Reported	15 Sep 2014
Version	2.3
Bug Type	Functional
Description	Fault Description Step 1) Joda-Time Button Select Step 2) DateTime error ... Fail Log, Stack Trace information
Attachment	Crash_LogFile, Dump_File

6. 결론

비용 효율적인 소프트웨어 유지보수 방안을 위한 정보검색 기반 테스트케이스 우선순위 기법에 대한 연구를 한다. 본 논문에서는 버그리포트를 이용한 정보검색 기반 테스트케이스 우선순위화 기법을 제안한다. 이를 통해 결함으로 수정된 소스코드에 대한 테스트케이스 우선순위화 기법의 정확도를 향상시킬 수 있었다. 향후 연구로는 소스코드와 버그 리포트간의 유사도를 더 높여 정보검색 기반 테스트케이스 우선순위화 기법에 대해 연구하고자 한다. 제안한 버그리포트를 이용한 정보검색 기반의 테스트케이스 우선순위화 기법은 테스트케이스의 클래스 수준을 확인하고 있지만 이후 연구로 메소드 수준까지 확인하는 테스트 케이스 우선순위화 기법에 대해 연구할 것이다.

Acknowledgments

이 논문은 2015년도 정부(교육과학기술부)의 재원으로 한국연구재단-차세대정보컴퓨팅개발사업의 지원을 받아 수행된 연구임(No. 2015045358)

참고문헌

[1] Acharya, Arup Abhinna, Prateeva Mahali, and Durga Prasad Mohapatra. "Model Based Test Case Prioritization Using Association Rule Mining". CIDM. pages 429-440, 2015.

[2] Ripon K. Saha, Lingming Zhang, Sarfraz Khurshid and Dewayne E. Perry. "An Information Retrieval Approach for Regression Test Prioritization Based on Program Changes". ICSE. pages 268-279, 2015

[3] Sharma, Anuradha, and Sachin Sharma. "Bug Report Triaging Using Textual, Categorical and Contextual Features Using Latent Dirichlet Allocation". IJIRST. pages 85-96, 2015.

[4] Klaus Changsun Youm, June Ahn, Jeongho Kim, Eunseok Lee. "Bug Localization Based on Code Change Histories and Bug Reports" in press. APSEC, 2015

[5] Manika Tyagi and Sona Malhotra. "An Approach for Test Case Prioritization Based on Three Factors". IJITCS. pages 78-86. 2014.

[6] A. Mohamed Shameem and N. Kanagavalli. "Dependency Detection for Regression Testing using Test Case Prioritization Techniques". IJCA. pages 20-25. 2013.

[7] R. Pradeepa and K. Vimala Devi. "Effectiveness of Testcase Prioritization using APFD Metric : Survey". IJCA. pages 1-4, 2013.