

Projektszám: 94öu8		HUF 53 000 EUR 4 160
Pályázó neve: Dr. Vidács László		Intézménye: MTA-SZTE
Projektpartner neve: Prof. Dr. Martin Pinzger		Intézménye: AAU
Pályázat címe: Forráskód és teszt adatok bányászata a szoftver evolúció támogatására		

A projekt jellege: (kérjük bejelölni)

- Workshop, konferencia
- Publikáció, tananyag
- Kutatási együttműködés
- Oktatási program

Beszámoló/Eredmények

A project során kitűzött célunk újító módszerek és eszközök tervezése és megvalósítása volt a program és teszt kód együttes evolúciójának elemzése területén. A projektet az előzetes kutatási terv szerint valósítottuk meg:

1. Módszerek és eszközök az együttes evolúció elemzéséhez

Különböző szoftver-eszközöket terveztünk és implementáltunk a teszt és program kód együttes evolúciójának elemzéséhez Java programnyelvhez. Mind statikus, mind dinamikus elemzési módszereket alkalmaztunk. A tesztek lefutásának részletes rögzítéséhez dinamikus elemzést alkalmaztunk. A teszt és program osztályok közötti kapcsolatokat tesztenkénti lefedettség méréssel rögzítettük metódus szinten. Statikus elemzéssel a Java osztályok részletes változtatásait nyertük ki.

2. Evolúciós minták bányászata

Az első elemzési fázis eredményei alapján a program és teszt kód együttes változásaiból ún. tranzakciókat készítettünk. A tranzakciókon asszociációs szabályok kinyerésével evolúciós mintákat kerestünk. Sikeresen ismételtük meg Marsavina és társainak munkáját a commons-lang nevű programon. Az előzmény cikkhez nagyon hasonló evolúciós mintákat találtunk, azonban a feltárt kapcsolatok gyengébbek voltak a korábbiaknál.

2. Új irányok és eredmények

A változások összetételét elemezve együttes evolúció szempontjából számos lehetőséget találtunk a program és teszt kód összekapcsolásának továbbfejlesztésére. A szoftver fejlesztői által elvégzett módosítások sok esetben egy összetettebb cél érdekében történnek, pl. egy hiba kijavítását célozzák meg. Ezen célok direkt módok nem jelennek meg a verziókezelő rendszerben, de nyilván vannak tartva a projekt menedzsment eszközökben. Első kísérleteink a magasabb szintű csoportosítás elemzésére sikeres volt, és új kutatási lehetőségeket nyitott meg előttünk. Egy másik ígéretes kutatási irány az együttes evolúció vizsgálata hibás verziók esetén. Kísérleteink során az építés vagy a teszt során hibát tartalmazó verziókat elemeztük a fenti evolúciós mintakereső algoritmusokkal. Több érdekes mintát is felfedeztünk a fejlesztők viselkedéséről hibás verziók esetén, melyek részletes vizsgálata szintén egy jövőbeli kutatási irány lesz.

A projekt eredményei (a kutatási terv szerint)

1. Módszerek, eszközök és adatok program és teszt kód együttes evolúciójának elemzéséhez
Elemző eszközöket (programokat) fejlesztettünk bash, C++ és R nyelveken, melyekkel elértük, hogy az elemzési folyamat nagy része automatizált. A szükségesnél jóval több és mélyebb adatot nyertünk ki, előmozdítva további kapcsolódó kutatások elvégzését.

2. Legalább 2 tudományos publikáció előkészítése

Az eredményeket egy kutatási riportban foglaltuk össze, melyben beszámolunk az előzmény cikk replikációjáról és a módszerek továbbfejlesztéséről. Emellett több ígéretes kutatási irányban vannak eredményeink, melyeket a közeljövőben részletesen szeretnénk kidolgozni közös kutatási munka keretében.

Publikációs jegyzék:

Publikáció:

Projektnummer: 94öu8		HUF 53 000 EUR 4 160
Antragsteller: Dr. Vidács László		Institut: HAS-USZ
Projektpartner: Prof. Dr. Martin Pinzger		Institut: AAU
Titel: Mining von Code und Tests zur Unterstützung der Evolution von Softwaresystemen		

Art der Förderung:

- Workshop, Konferenz
- Publikation, Lehrmaterial
- Forschungsprojekt
- Unterrichtsprojekt

Bericht

Das Ziel dieses Forschungsprojekts war der Entwurf und die Implementierung von Methoden und Tools zur Verbesserung des aktuellen Wissensstands im Bereich der Analyse der Co-Evolution von Produkt- und Test-Code. Das Projekt umfasste folgende Schritte:

1. Methoden und Tools für die Co-Änderungsanalyse

Wir entwarfen Methoden und implementierten verschiedene Tools für die detaillierte Analyse der Co-Evolution von Produkt- und Test-Code in Java Programmen. Hierzu verwendeten wir sowohl statische wie auch dynamische Analysemethoden. Dynamische Analyse wurde verwendet, um das detaillierte Verhalten von Test-Klassen aufzuzeichnen. Für jeden ausgeführten Testfall zeichneten wir den Überdeckungsgrad auf Methoden-Ebene auf, um die Verbindung zwischen Produkt- und Test-Klassen herzustellen. Statische Analyse wurde verwendet, um die fein-granularen Änderungen für jede Version einer Java-Klasse der untersuchten Softwaresysteme zu extrahieren.

2. Mining von Mustern der Co-Evolution

Die Resultate der ersten Phase wurden verwendet, um sogenannte Co-Change Transaktionen für Produkt- und Test-Klassen zu berechnen. Diese Transaktionen wurden verwendet, um mittels Association Rule Mining Muster für die Co-Evolution zwischen Produkt- und Test-Klassen zu finden. Wir konnten mit unserer Methode die Originalstudie von Marsavina et al. für ein Zielsystem, nämlich commons-lang, erfolgreich replizieren. Unserer Studie ergab ähnliche Muster der Co-Evolution, jedoch zeigten diese im Vergleich zur Studie von Marsavina et al. schwächere Verbindungen.

3. Weitere Resultate und neue Forschungsrichtungen

Wir analysierten die Co-Evolution von Produkt- und Test-Code auf Basis von Commits. Während unserer Studie fanden wir einige Verbesserungsmöglichkeiten dieses Ansatzes. Eine davon ist, dass Änderungen meist durchgeführt werden, um ein übergeordnetes Ziel, wie zum Beispiel die Behebung eines Fehlers, zu erreichen. Derartige Ziele werden selten in Versionierungssystemen festgehalten, dafür aber in anderen Systemen, wie zum Beispiel Issue Trackers. Wir extrahierten diese Informationen von einem Issue Tracker und es zeigte sich, dass Fehlerberichte helfen, Änderungen in Produkt- und Test-Klassen besser zu gruppieren. Dieser Ansatz eröffnet eine neue Richtung zur Verbesserung unserer Studie, die wir in Zukunft untersuchen werden. Eine weitere Verbesserungsmöglichkeit ergab sich durch die Analyse der Co-Evolution im Falle von Build-Fehlern. Wir untersuchten die Co-Evolution in Programmversionen, in denen der Build oder die Tests fehlschlügen, und fanden einige interessante Muster, die in zukünftiger Forschung weiter untersucht werden sollten.

Zusammenfassung der Resultate

1. Methoden, Tools und Daten für die Analyse der Co-Evolution von Produkt- und Test-Code. Die Tools wurden mittels bash/C++/R implementiert und ermöglichen die Automatisierung des Analyseprozesses. Wir haben zusätzliche Daten von Systemen und Versionen extrahiert, die für zukünftige Forschung zur Verfügung stehen.

2. Entwürfe von mindesten 2 wissenschaftlichen Artikeln, die wir bei renommierten Konferenzen/Journalen einreichen möchten. Als Grundlage dafür haben wir einen technischen Bericht erstellt, der die Originalstudie von Marsavina et al. repliziert und erweitert.

Unsere Resultate zeigen einige vielversprechende Möglichkeiten auf, die Analyse der Co-Evolution von Produkt- und Test-Code zu verbessern. Diese Möglichkeiten planen wir in einer zukünftigen Zusammenarbeit der beiden Projektpartner weiter zu untersuchen

Publikationsliste:

Publikationsverzeichnis:

Abschlußbericht

Weitere Fragen zu den Ergebnissen:

1. Nutzung und Verbreitung der Ergebnisse:

Welchen konkreten Nutzen konnten Sie und Ihr Kooperationspartner aus dem Projekt gewinnen. Bitte denken Sie insbesondere an Publikationen, Experimente, gemeinsame Seminare, Sommerschools und/oder an eine anderweitige Umsetzung in die Praxis.

In diesem Projekt entwickelten wir eine Anzahl von Analysetools, um die Datenaufbereitung und -analyse zu automatisieren. Mit diesen Tools war es uns möglich, eine große Menge von Daten zu extrahieren, die wir für unsere zukünftigen Experimente verwenden werden. Die Resultate dieser Experimente planen wir dann, in 1-2 wissenschaftliche Publikationen zu veröffentlichen, wovon die erste Publikation innerhalb der kommenden 6 Monate erfolgen soll.

2. Durchführung:

Welche konkrete Änderungen gegenüber der Planung ergaben sich hinsichtlich Inhalte und Mitarbeit/Anzahl der Teilnehmer während des Projektverlaufes?

Keine.

3. Bewertung:

Bitte führen Sie besonders positive, aber auch negative Beobachtungen und Erfahrungen an. Ev. langfristige Auswirkungen Ihres Projektes?

Das Projekt ermöglichte eine erste Zusammenarbeit der beiden Forschungsgruppen an der Alpen-Adria Universität Klagenfurt und Ungarische Akademie der Wissenschaften & Universität Szeged, welche die Basis für zukünftige Zusammenarbeiten bildet.

4. Perspektiven:

Hat sich eine Fortführung der Kooperation ergeben?

- a. Welche geplante Fortführung gibt es?
- b. Welche konkrete Fortführung gibt es?

ad a) Wir planen die Erstellung einer dauerhaften Zusammenarbeit in der Forschung und in der Lehre zwischen den beiden Universitäten. Bezüglich Lehre planen wir, ein Erasmus+ oder Ceepus Programm einzurichten, das den Austausch von Studierenden und Lehrenden unserer Universitäten ermöglicht.

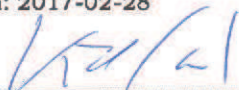
ad b) Wir arbeiten derzeit an einer gemeinsamen wissenschaftlichen Publikation, in der wir die ersten Resultate dieses Forschungsprojekts veröffentlichen wollen.

5. Verbesserungsvorschläge:

Nenne Sie uns, Bitte, Verbesserungsvorschläge, wie Sie Ihre Arbeit oder wie wir unseren Service besser gestalten könnten?

Ihr Service funktionierte sehr gut und wir möchten uns gerne nochmals bedanken, dass die Stiftung Aktion Österreich-Ungarn unser Projekt finanziert hat.

Datum: 2017-02-28



Antragsteller (Unterschrift)



Projektpartner (Unterschrift)