Entkopplung einer Komponente in ProB

Silas A. Kraume

Exposé zur Bachelorarbeit

Softwaretechnik und Programmiersprachen
Heinrich Heine Universität Düsseldorf, Deutschland
6. Juli 2024

- Erstgutachter: -
- Zweitgutachter: -
- Gewünschter Starttermin: Ende Juli 2024

1 Hintergrund

Der an der HHU am Lehrstuhl der Softwaretechnik und Programmiersprachen entwickelte Animator, Constraint-Solver und Model-Checker ProB (Leuschel und Butler 2008) für Modelle der B-Methode, unterstützt die automatische Animation vieler B Spezifikationen und kann verwendet werden um diese systematisch nach Fehlern abzusuchen. Als solcher findet ProB bereits in vielen Systemen Verwendung zur Datenvalidierung und Validierung komplexer Eigenschaften für sicherheitskritische Systeme. ProB wird bereits von mehreren Unternehmen verwendet und ist November 2022 mit dem 'AlainColmerauer Prize' ausgezeichnet worden.

2 Problembeschreibung

Um Prädikate zu lösen ist in ProB der von Microsoft Research entwickelte Z3 Solver(C++) (Moura und Bjørner 2008) in einer Programmerweiterung als Backend eingebunden. Dieser stellt jedoch insofern ein Problem dar, dass sporadisch Speicherlecks und Segmentation Faults auftreten. Zudem werden aktuell mehrere Prädikate sequentiell gelöst, wodurch es gegebenenfalls zum Erreichen einer Performance Grenze kommen kann. Ersteres stellt insbesondere bezüglich der Hintergrundinformationen, dass ProB zur Datenvalidierung für sicherheitskritische Systeme verwendet wird, eine ernstzunehmende Problematik dar.

3 Ziele

In dieser Bachelorarbeit soll der Z3 Solver in einen seperaten (C++) Prozess ausgelagert werden, sodass ProB und Z3 entkoppelt sind. Es wird also die bestehende Vorgehensweise, die Prädikate im C-Interface zusammenzubauen und zu lösen, verworfen. Stattdessen wird hiermit ein System eingeführt, bei dem die Prädikate an den Z3 Prozess gesendet werden, wo diese gelöst und zurückgeschickt werden. Als Technologie zur Kommunikation zwischen den zwei Prozessen wird die ZeroMQ-Bibliothek (Hintjens 2013) verwendet, um eine direkte IPC-Protokoll Verbindung zu legen. ZeroMQ (auch ØMQ, 0MQ oder ZMQ) ist eine Nachrichtenaustauschbibliothek, die speziell für verteilte Systeme entwickelt wurde, und zeichnet sich durch ihre geringe Latenz aus.

Diese Arbeit wird mit dem Interesse der Erweiterbarkeit vollrichtet, sodass zukünftig die Option besteht, gegebenenfalls mehrere Instanzen des Z3 Prozesses zu starten und das Lösen der Prädikate zu parallelisieren.

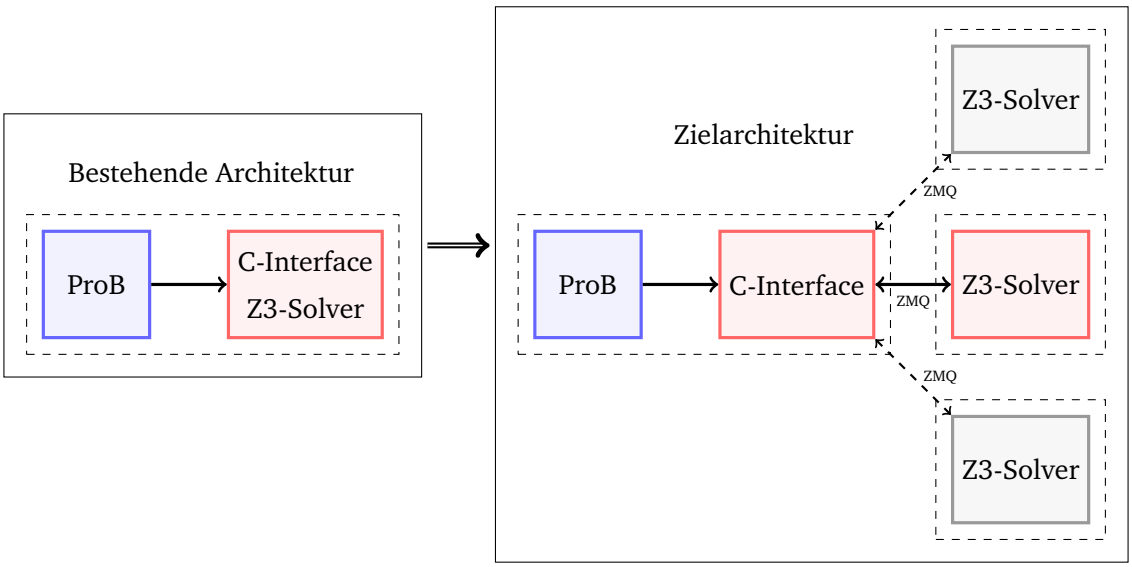

Abbildung 1: Geplante Architekturänderung (Die Komponenten-Entkopplung ist in Rot markiert. Gestrichelte Boxen zeigen die verschiedenen Prozesse an.)

Zuletzt besteht die Möglichkeit diese Arbeit insofern zu erweitern, dass die Leistungsfähigkeit der Entkopplung, durch etwaige Benchmarks und Tests, evaluiert wird. In diesem Fall wird der Einfluss auf die Gesamtperformance der Programmerweiterung hinsichtlich dem Vergleich zur vorherigen Vorgehensweise analysiert.

4 Minimalanforderungen

- Trennung von ProB und Z3 mittels ZeroMQ

5 Erweiterungen

- Evaluation des Performance-Overheads
- ggf. alternative Lösungsansätze

6 Ressourcen

Zur Ausführung des ProB Codes sind die Lizenzinformationen von SICStus Prolog 4.8 notwendig. Diese werden von der Betreuung vorliegender Bachelorarbeit bereitgestellt. Ein Zugang zu einem Versionskontrollsystem wie Git ist hilfreich bis notwendig, um den Fortschritt der Arbeit zu protokollieren und zu erleichtern. Entsprechend wird voraussichtlich GitLab als Git-Repository-Manager verwendet.

7 Hindernisse und Schwierigkeiten

Unter den wichtigsten Fertigkeiten, um diese Bachelorarbeit zu absolvieren, ist eine moderate bis gute Kenntnis in der Programmiersprache C++. Dies stellt zunächst ein Hinderniss dar, da diese Sprache noch unvertraut ist. Entsprechend muss das Erlernen von C++ im Zeitplan berücksichtigt werden.

8 Ungefährer Zeitplan

Eine grobe Herangehensweise im Entkoppeln der Z3 Komponente lautet wie folgt:

Aufstellen einer ZeroMQ Kommunikation zwischen einem simplen Client-Server Modell:	bis Woche 1
Das Client Modell in das C-Interface von ProB integrieren:	bis Woche 1
Implementieren einer einzelnen Funktion im Server Prozess:	bis Woche 3
den Server ausbauen um das Erweitern weiterer Funktion zu erleichtern:	bis Woche 4-5
nacheinander alle Funktionen in die neue Architektur transferieren:	bis Woche 8-10
(Hier sind die meisten Probleme und EdgeCases zu erwarten.)	
Performance Analyse:	bis Woche 12

9 Aufsicht

TBD

Literatur

[1] Michael Leuschel und Michael J. Butler. “ProB: an automated analysis toolset for the B method”. In: *STTT* 10.2 (2008), S. 185–203.

[2] Leonardo de Moura und Nikolaj Bjørner. “Z3: An Efficient SMT Solver”. In: *Tools and Algorithms for the Construction and Analysis of Systems*. Hrsg. von C. R. Ramakrishnan und Jakob Rehof. Berlin, Heidelberg: Springer Berlin Heidelberg, 2008, S. 337–340. ISBN: 978-3-540-78800-3.

[3] Pieter Hintjens. *ZeroMQ: messaging for many applications*. " O’Reilly Media, Inc.", 2013.