

Aufbau und Konsolidierung einer Konzepthierarchie für Anforderungsbeschrei- bungen aus unterschiedlichen Wissensquellen

Dokumententart: Exposé für eine Bachelorarbeit
Autor: Maximilian Wessendorf
Matrikel-Nr.: 1928241
Studiengang: Informationswirtschaft Bachelor
Betreuer: Tobias Hey
Datum: 28. August 2019

1 Motivation

In den letzten fünfzig Jahren ist Software nicht nur immer weiter in alle Bereiche unseres Lebens vorgedrungen, sondern auch immer komplexer geworden [Dvo09]. Hierdurch entstehen Bedürfnisse für neue technische Lösungen, die Softwareingenieure zum Beispiel bei der Wartung von Software oder der Auswirkungsanalyse von Anforderungsänderungen unterstützen. Die automatische Verknüpfung von Anforderungen und Quelltext ist eine Methode, die dabei helfen soll [ACC⁺02].

Es gibt bereits Ansätze, die derartige Verknüpfungen erfassen können. Diese setzen allerdings eine entsprechende Benennung der Konzepte im Anforderungsdokument und im Quelltext voraus, um eine syntaktische Verbindung herstellen zu können. In der Realität liegt eine derart systematisch durchdachte Benennung allerdings möglicherweise nicht vor. Ein Grund hierfür kann sein, dass unterschiedliche Abstraktionsebenen für Konzepte in der Anforderung und dem Quelltext gewählt werden. Daher braucht es neue Methoden, die auch ohne syntaktische Verbundenheit, solche Verknüpfungen, über mehrere Abstraktionsebenen hinweg, herstellen können. Eine Möglichkeit ist die Einbeziehung von Weltwissen, um ein explizites Verständnis der Anforderungen zu erlangen. Abbildung 1 zeigt ein Beispiel für eine Anforderung und einen Quelltext-Ausschnitt bei dem zunächst nicht ausreichend Informationen in Anforderung und Quelltext vorhanden sind, um eine Verbindung

zu erfassen. Nach Hinzufügen von Weltwissen aus WordNet [FM98] und Wikipedia [VK14], bieten die hinzugefügten Konzepte, der nächst höheren Abstraktionsebene, nun die Möglichkeit eine syntaktische Verknüpfung von der Anforderung zum Quelltext herzustellen. Das Hinzufügen von Weltwissen erlaubt also, dass auch dann eine automatische Verknüpfung von Anforderungen und Quelltext erreicht werden kann, wenn in der Entwurfsphase und der Implementierungsphase des Softwareentwicklungsprozesses unterschiedliche Detailgrade verwendet werden.

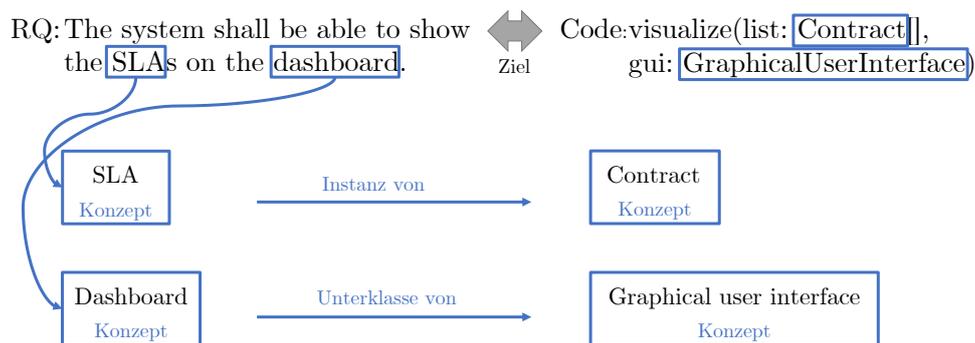


Abbildung 1: Beispiel für eine Anforderung und einen Quelltext-Ausschnitt, die durch Weltwissen miteinander verknüpft werden können.

2 INDIRECT: INtent-DrIven REquirements-to-Code Traceability

Das Projekt INDIRECT zielt darauf ab, die Präzision der automatischen Verknüpfung zwischen Anforderungen und Quelltext signifikant zu erhöhen. Dafür wird ein neuer Ansatz vorgeschlagen, bei dem explizite Absichtsmodelle für die in natürlicher Sprache verfassten Anforderungen sowie für den Quelltext erzeugt werden. Absichtsmodelle sind Graphrepräsentationen, die die Bedeutung von Anforderung und Quelltext repräsentieren. Die Verknüpfung der Absichtsmodelle von Anforderung und dem entsprechenden Teil im Quelltext wird daraufhin durch übereinstimmende Muster in den Graphstrukturen der beiden Modelle gewonnen.

INDIRECT verwendet die agentenbasierte Architektur PARSE (ehemals ProNat) [WT15]. Dabei dient ein zentraler Datenspeicher als Schnittstelle für verschiedene Agenten, die die Daten kontinuierlich ergänzen. Neue Eingaben werden hierbei zunächst mit seichter Sprachverarbeitung vorbereitet. Die Anforderungssätze werden dabei in die einzelnen Wörter (Token) zerlegt und diese um erste Zusatzinformationen, wie ihre Wortart, ergänzt.

Diese Token werden dann in einen Graphen überführt. Die verschiedenen Agenten ergänzen dann diesen Graphen entsprechend ihrer Aufgabe. Einer dieser Agenten ist der Konzeptualisierer, der Kontextwissen aus WordNet zum Graph hinzufügt. Dadurch können Zusammenhänge zwischen Anforderungen und Quelltext erfasst werden, die durch einen rein syntaktischen Vergleich nicht erkannt worden wären.

3 Zielsetzung

Ziel dieser Arbeit ist es, die Vollständigkeit und die Qualität, des zu den Absichtsmodellen hinzugefügten Weltwissens, zu verbessern. Der Konzeptualisierer arbeitet bisher ausschließlich mit der lexikalischen Datenbank WordNet, welche für viele Konzepte aus dem Bereich der Softwareentwicklung nur wenige bis keine Informationen enthält. INDIRECT soll deshalb so erweitert werden, dass beliebige Wissensquellen modular zum System hinzugefügt werden können. Es soll dabei für in Anforderungen auftretende Konzepte eine Konzepthierarchie aus den verschiedenen angeschlossenen Wissensquellen konsolidiert werden.

3.1 Beispiel

Abbildung 2 zeigt ein Beispiel für jeweils einen Teil der Konzepthierarchie, für das Konzept Contract, aus den beiden Wissensquellen WordNet und Wikipedia. Ziel ist es, diese beiden Hierarchien zu einer zu konsolidieren. Auf der rechten Seite ist dargestellt, wie so eine konsolidierte Konzepthierarchie aussehen könnte.

4 Vorgehen

Um das beschriebene Ziel zu erreichen, sollen zunächst mögliche Wissensquellen auf ihre Eignung untersucht werden. Daraufhin werden Agenten für INDIRECT entwickelt, die die Informationen aus den ausgewählten Wissensquellen extrahieren können. Hierbei wird es möglich sein Parameter wie das Einstiegskonzept, die gewünschte Suchtiefe und eventuelle Beschränkungen auf bestimmte Relationen, flexibel zu bestimmen. Des Weiteren wird eine Strategie entwickelt, mit der die Ergebnisse aus den verschiedenen Wissensquellen zu einer Konzepthierarchie konsolidiert werden können. Zunächst werden hierbei regelbasierte Strategien untersucht. Ein Beispiel hierfür sind die im vorangegangenen Beispiel verwendeten Regeln, nach denen ähnliche Konzepte zusammengeführt und an die speziellste Stelle eines der Ausgangskonzepte platziert wurden (Regel 1) und Speziellere Konzepte zwischen ihre Oberklasse und deren Unterklasse eingefügt wurden (Regel 2). Dadurch können Probleme wie strukturelle Unterschiede der Wissensquellen

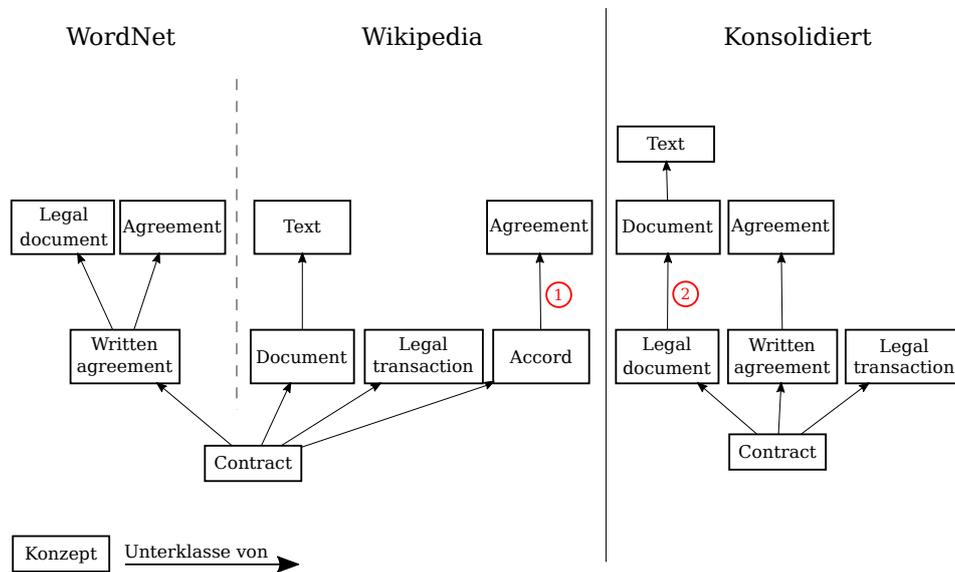


Abbildung 2: Ausschnitte aus den Konzepthierarchien von WordNet und Wikipedia und eine mögliche Konzepthierarchie nach Konsolidierung.

und unterschiedliche Granularität der Abstraktionsebenen überwunden werden [PGJ99]. Eine semantische Unterscheidung syntaktisch gleicher Begriffe wird hierbei, wie auch zu Beginn, bei der Auswahl des Einstiegskonzepts, eine wichtige Rolle spielen. Letztlich wird ein Agent für INDIRECT entwickelt, der eine oder mehrere Integrationsstrategien, auf Basis des durch die Agenten der Wissensquellen verfügbar gemachten Wissens, umsetzt.

5 Evaluation

Die Ergebnisse der Arbeit sollen in zwei Schritten evaluiert werden. Zunächst soll untersucht werden, welchen Vorteil der Ansatz, mehrere Wissensquellen zu kombinieren, unter Verwendung der verfügbaren Wissensquellen bringt (Nützlichkeit). Hierzu soll berechnet werden, welchen Anteil an Konzepten aus einem Anforderungsdatensatz dem Konzeptualisierungsmodul vor und nach der Arbeit bekannt sind. Im zweiten Teil soll überprüft werden, ob die erlangten Resultate dazu geeignet sind die Qualität der INDIRECT Ergebnisse zu verbessern. Dafür soll händisch eine Auswahl an Musterlösungen (Gold-Standard) erstellt werden, gegen die die Ergebnisse einer oder mehrerer Integrationsstrategien evaluiert werden können.

Literatur

- [ACC⁺02] ANTONIOL, G. ; CANFORA, G. ; CASAZZA, G. ; DE LUCIA, A. ; MERLO, E.: Recovering Traceability Links between Code and Documentation. In: *IEEE Transactions on Software Engineering* 28 (2002), Oktober, Nr. 10, S. 970–983. <http://dx.doi.org/10.1109/TSE.2002.1041053>. – DOI 10.1109/TSE.2002.1041053. – ISSN 0098–5589
- [Dvo09] DVORAK, Daniel: NASA Study on Flight Software Complexity. In: *AIAA Infotech@Aerospace Conference*. Seattle, Washington : American Institute of Aeronautics and Astronautics, April 2009. – ISBN 978–1–60086–979–2
- [FM98] FELLBAUM, C. ; MILLER, G.A.: *WordNet: An Electronic Lexical Database*. MIT Press, 1998 (Language, speech, and communication). <https://books.google.de/books?id=Rehu800zMIMC>. – ISBN 9780262061971
- [PGJ99] PRASENJIT MITRA ; GIO WIEDERHOLD ; JAN JANNINK: *Semi-automatic Integration of Knowledge Sources*, 1999
- [VK14] VRANDEČIĆ, Denny ; KRÖTZSCH, Markus: Wikidata: a free collaborative knowledgebase. In: *Communications of the ACM* 57 (2014), September, Nr. 10, 78–85. <http://dx.doi.org/10.1145/2629489>. – DOI 10.1145/2629489. – ISSN 00010782
- [WT15] WEIGELT, Sebastian ; TICHY, Walter F.: Poster: ProNat: An Agent-Based System Design for Programming in Spoken Natural Language. In: *2015 IEEE/ACM 37th IEEE International Conference on Software Engineering*. Florence, Italy : IEEE, Mai 2015. – ISBN 978–1–4799–1934–5, S. 819–820