

Extraktion von gemeinsamen und variablen Anforderungen aus Benutzerdokumentationen¹

Isabel John

Isabel.John@iese.fraunhofer.de

Fraunhofer IESE, Sauerwiesen 6, 67661 Kaiserslautern

Zusammenfassung: Dieser Artikel beschreibt einen Ansatz zur Extraktion von Anforderungen an eine Produktlinie von Systemen basierend auf der Benutzerdokumentation existierender Systeme. Domänenexperten sind oft der Flaschenhals bei der Produktlinienentwicklung, daher ist das primäre Ziel des Ansatz ist die Entlastung dieser Experten und somit die Unterstützung des Produktlinien Technologie Transfers in die Praxis.

1 Einleitung

Existierende Software wird oft von Dokumentation begleitet. Diese Dokumentation ist eine wichtige Informationsquelle beim Requirements Engineering für neue Systeme, wird jedoch bisher oft nicht systematisch genutzt. Gerade beim Produktlinienengineering ist das Ziel oft, auf existierenden Systemen (die häufig keine Requirements Dokumente besitzen) aufzubauen, es sollen also gezielt existierende Systeme in die Produktlinie integriert werden. Die Erhebung neuer Anforderungen ist im Requirements Engineering für Einzelsysteme und für Produktlinien ein interaktiver Prozess unter starker Einbeziehung von Domänenexperten die oft auch eine zentrale Rolle in der Entwicklung spielen. Die starke Einbindung und nicht vorhandene Zeit der Experten haben sich bei Produktlinieneinführungen oft als Flaschenhals erwiesen. Durch das gezielte Nutzen von existierender Dokumentation können Experten entlastet und existierende Systeme reibungsloser integriert werden. Mit Hilfe des CaVE-Ansatz (C_{ommonality} and V_{ariability} E_{xtraction}) [1] den wir im folgenden vorstellen, können strukturiert Anforderungsfragmente in Dokumentationen existierender System gefunden und als gemeinsame und variable Anforderungen an eine neue Produktlinie integriert werden. Die existierenden Anforderungen können also, zumindest als Fragmente, aus der Dokumentation gewonnen werden, den Experten bleibt somit Zeit innovative Funktionalität für die neuen Produkte der Produktlinie zu entwickeln.

Der CaVE Ansatz besteht aus Extraktionspattern, einem Metamodell und einem Extraktionsprozess. Im folgenden werden wir nur auf den Extraktionsprozess eingehen, eine Beschreibung der anderen Komponenten findet sich in [1]

2 Der Extraktionsprozess

Wie bei der Suche von Anforderungsinformation in

Benutzerdokumenten vorzugehen ist, beschreibt ein dreistufiger Extraktionsprozess. Im ersten Schritt wählt der Produktlinienanalyst die Ausgangsdokumente aus und zerteilt sie in handhabbare Stücke, so dass die Extraktion der Anforderungen aus den Dokumenten so effektiv und effizient wie möglich erfolgen kann. Weiterhin wählt er aus einer Sammlung von Extraktionspattern jene aus, die für diese Dokumentationsteile geeignet sind. Diese Extraktionspattern, die bei der eigentlichen Suche verwendet werden, beschreiben wo man typischerweise in der Benutzerdokumentation welche Anforderungselemente findet. Im zweiten Schritt identifiziert der Produktlinienanalyst mit Hilfe der Extraktionspattern die Elemente in den Dokumentationsteilen, die dann in gemeinsame und variable Anforderungsfragmente überführt werden. Erst im dritten Schritt wird der Domänenexperte involviert. Der Produktlinienanalyst fügt die in den Benutzerdokumenten identifizierten Elemente zu einem vorläufigen, partiellen Produktlinienmodell zusammen und legt es dem Domänenexperten zur Begutachtung vor. Dieser überprüft die Korrektheit und Vollständigkeit des Produktlinienmodells aus seinem Experten-Blickwinkel. Dabei führt er gegebenenfalls Änderungen und Ergänzungen durch. Damit entsteht ein partielles Produktlinienmodell aber auch Information die für die Produktlinienplanung genutzt werden kann. Diese Information kann nun weiter detailliert und gegebenenfalls durch andere Spezifikationen ergänzt werden für die keine Elemente extrahiert werden konnten. Abbildung 1 gibt eine Übersicht über die verschiedenen Prozessschritte und deren Ein- und Ausgaben.

2.1 Vorbereitung

In der Vorbereitungsphase wird die existierende Dokumentation gesichtet, es werden 2-4 typische Dokumentationen ausgewählt. Die ausgewählten Dokumentationen werden in Kapitel bzw. Teile (z.B. Inhalt, Kapitel, Glossar) aufgespalten so dass zusammenhängende Teile verschiedener Dokumentationen parallel verglichen werden können. Ausgehend von der ersten Sichtung der Teile wird sich für das weitere Vorgehen entschieden: Unterscheiden sich die verschiedenen Dokumentationen stark voneinander (z.B. mehr als 30% Unterschiede), wird die eigentliche Analyse zuerst mit einer einzelnen Dokumentation durchgeführt, danach werden die Unterschiede zu den anderen Systemen identifiziert. Unterscheiden sich die Dokumentationen nur wenig kann im weiteren Vorgehen parallel verglichen werden.

¹. Zum Teil gefördert vom Eureka S! 2023 Program, ITEA project ip00004, Café.

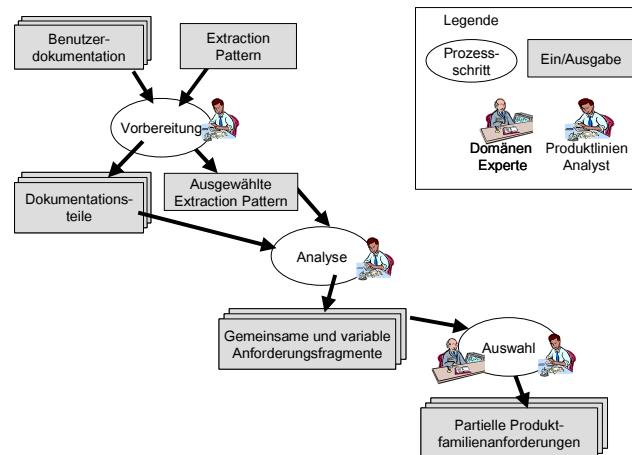


Abbildung 1 Übersicht des Extraktionsprozess

Zusätzlich werden die anzuwendenden Pattern anhand verschiedener Kriterien (z.B. Korrektheit der Pattern, angestrebte Resultate) ausgewählt.

2.2 Extraktion

In der Extraktionsphase werden Anforderungselemente wie z.B. Features [2], Elemente von textuellen Use Cases [3], Funktionale Anforderungen mit Hilfe der ausgewählten Extraktionspattern im Text gefunden. Zusätzlich werden mit Hilfe von Pattern Gemeinsamkeiten und Unterschiede identifiziert. Die identifizierten Elemente werden im Text markiert. In der Analysephase werden Anforderungselemente mit Hilfe der ausgewählten Extraktionspattern im Text gesucht. Mögliche Anforderungselemente können z.B. Use Case Elemente wie Use Case Namen, Use Case Schritte etc., textuelle Anforderungen oder Features sein. Features sind hier für den Benutzer wichtige funktionale oder nicht funktionale Eigenschaften des Systems, Features werden beschrieben durch einen Namen und mögliche Werte (z.B. könnte das Feature „Schiebedach“ bei einer Produktlinie von PKW die Werte „automatisch“ „manuell“ und „keins“ annehmen). Aufbauend auf den identifizierten Anforderungselementen werden mit Hilfe von Pattern Gemeinsamkeiten und Unterschiede identifiziert. Der Produktlinienanalyst markiert die identifizierten Elemente im Text und vergibt optional eine Kennzeichnung für das gefundene Element.

2.3 Nachbereitung

In der Nachbereitung werden die identifizierten Elemente zunächst zu vorläufigen Produktlinienartefakten (z.B. Use Cases, Feature Listen, Anforderungen) zusammengesetzt. Hier setzt der Produktlinienanalyst die identifizierten Elemente zunächst zu vorläufigen Produktlinienartefakten (z.B. Use Cases, Feature Listen, textuelle Anforderungen) zusammen. Der Produktlinienanalyst legt dieses vorläufige partielle Produktlinienmodell dem Domänenexperten zur Auswahl, Änderung und zum

Hinzufügen von neuen Artefakten vor. Der Experte ist also nur in einem Teilschritt des Prozess involviert. Resultat des Ansatzes ist ein partielles Produktlinienmodell welches dann in einem folgenden Modellierungsschritt weiter analysiert, vervollständigt und verifiziert werden kann. Die erzeugten Artefakte können direkt als Eingabe für verschiedene Produktlinienmodellierungstechniken verwendet werden.

3 Zusammenfassung

Mit Hilfe des hier vorgestellten CaVE Ansatz der auf Basis von Extraktionspattern Anforderungsfragmente in Benutzerdokumentation findet, können Anforderungen in beliebigen Notationen gefunden und in eine Produktlinie integriert werden. Dieser dokumentationsbasierte Ansatz entlastet die Domänenexperten. Der Ansatz wurde in industriellen Fallstudien erfolgreich angewendet. Momentan wird eine Toolunterstützung des Ansatz realisiert, die mit Hilfe von Information Retrieval Methoden [4] relevante Dokumententeile findet und eine Suche mit Hilfe der Pattern durchführt.

Referenzen

- [1] I. John, J. Dörr, "Elicitation of Requirements from User Documentation", Proceedings of REFSQ '03, Klagenfurt, June 2003
- [2] K. Kang, S. Cohen, J. Hess, W. Novak, and S. Peterson. Feature-Oriented Domain Analysis (FODA) Feasibility Study. Technical Report CMU/SEI-90-TR-21, Software Engineering Institute, Carnegie Mellon University, November 1990
- [3] A. Cockburn. Writing Effective Use Cases. Addison Wesley, 2001.
- [4] R. Baeza-Yates, B. Ribeiro-Neto. "Modern information retrieval". Addison-Wesley, 1999