

Anforderungsanalyse in der modellbasierten Entwicklung am Beispiel von AutoFocus  
Bernhard Schätz, Markus Pister, Alexander Wisspeintner  
Fakultät für Informatik, TU München  
Boltzmannstr. 3, 85748 Garching  
Email: {schaetz|pister|wisspein@in.tum.de}

Trotz einer Vielzahl von verschiedenen graphischen Notationen werden die ersten Anforderungen an ein System – gerade im Bereich der eingebetteten Systeme - meist textuell erfasst und mit RE-Werkzeugen wie DOORS [7] oder RequisitePro [5] strukturiert. Solche Werkzeuge erlauben die Strukturierung (meist unter Verwendung von Strukturvorlagen) und Vernetzung von textuellen Anforderungen; darüber hinaus bieten sie auch die Möglichkeit der Verknüpfung dieser Anforderungen zu Elementen der weiteren (graphischen) Modelle des Entwicklungsprozesses an. Dies ermöglicht es, die Verfolgung von Anforderungen um Überdeckungsanalysen zu unterstützen und Änderungskonsequenzen festzustellen. Jedoch mit Einschränkungen: meist werden nur OOAD/UML-Modelle verwendet, die für eingebettete System nur bedingt tauglich sind; aus diesen Modellen werden nur einige „first class citizens“ für die Vernetzung zur Verfügung gestellt; neben Überdeckungs-/Abhängigkeitsanalysen werden kaum tiefere Analysen angeboten.

Hier kann die modellbasierte Anforderungsanalyse einen Beitrag zur strukturierten Erfassung und Überprüfung von Anforderungen liefern. Die modellbasierte Entwicklung [6] beruht auf der Beschreibung des zu entwickelnden Produkts entsprechend den Vorgaben eines expliziten (domänenspezifischen) konzeptuellen Modells, meist unter Verwendung entsprechenden Beschreibungsformalismen. Durch die Hinzunahme von (informellen) Anforderungen in dieses konzeptuelle Modell lassen sich Anforderungsanalyse und frühes Design besser unterstützen und koppeln:

- Bei der Strukturierung der Anforderungen lassen sich Modellelemente ableiten (z.B. Ereignisse eines Sequenzdiagramm aus einer Szenarienbeschreibung) und entsprechende durch das detaillierte Modell identifizierte Unklarheiten präzisieren (z.B. Erzeuger eines Ereignisses).
- Speziell für eingebettete Systeme geeignete, vordefinierte Modellframeworks (z.B. das Vier-Variablen-Modell von SCR [3]) werden verwendet, um durch die Abbildung von textuellen Anforderungen auf deren Modellelemente die strukturierte Entwicklung/Analyse von Anforderungen zu unterstützen, und Unklarheiten zu identifizieren.
- Durch die Verwendung von Domänen-Modellframeworks (ähnlich zu den PID der Model Driven Architecture) kann die Strukturierung und Analyse der textuellen Anforderungen auch speziell auf eine anwendungsdomänenspezifische logische Struktur zugeschnitten werden.

Der zentrale Ansatz bei der modellbasierten Anforderungsanalyse liegt in der Einführung eines gemeinsamen Modells sowohl für Anforderungen als auch Entwurfselemente. Darauf aufbauend folgt die zunehmende Strukturierung der Anforderungen im Entwicklungsprozess (z.B. von unstrukturiertem Text über gegliedertem Text und Tabellen). Damit wird ein gradueller werkzeuggestützter Übergang von der Anforderungsanalyse in den Entwurf ermöglicht und der Bruch zwischen diesen beiden Phasen vermieden. Dieser Übergang wird erreicht durch die schrittweise zunehmende Strukturierung der Anforderungen – ähnlich wie z.B. in [4] - und der Aufnahme von Entwurfsentscheidungen; die zunehmende Präzision der Systembeschreibung und die damit verbundene abnehmende Komplexität erlaubt ein höheres Maß an Analyse- und Generationstechniken zur Unterstützung der Entwicklungsqualität und –effizienz. Dabei fügt sich die modellbasierte Anforderungsanalyse nahtlos in einen strukturierten Entwicklungsprozess ein. So kann beispielsweise die werkzeuggestützte Strukturierung und Klassifizierung mit der strukturierten Analyse der Anforderungen im Rahmen einer Inspektion oder eines Reviews durch einen Domänen- oder Entwurfsexperten kombiniert werden.

Der wesentliche Schritt bei der Einbettung von Anforderungen in einen modellbasierten Entwicklungsprozess ist die Integration der Anforderungen in das Produktmodell [1]. Die Integration erfolgt dabei mittels vier verschiedener Maßnahmen:

- *Domänenspezifische Klassifizierung der Anforderungen*: Einführung von domänenspezifischen Klassen von Anforderungen (z.B. Schnittstellenanforderungen, architektonische Anforderungen, Anforderungen an Benutzermodi, Anwendungsszenarien)
- *Ankopplung an Modellsichten*: Verknüpfung von Anforderungsklassen und anderen Modellelementklassen (z.B. Anforderungen an Benutzermodi zu Kontrollzustände, architektonische Anforderungen zu Komponenten)
- *Einführung von konzeptuellen Korrektheitseigenschaften*: Unterstützung der Analyse der Beziehungen von Anforderungen und weiteren Modellelementen (z.B. Abdeckung von Anforderungen, Übereinstimmung zwischen Hierarchie der architektonischen Anforderungen und der Komponentenhierarchie)
- *Unterstützung von Strukturierungsschritten*: Mechanisierung der zunehmenden Strukturierung (z.B. Identifikation von Komponenten in Architekturanforderungen, von Auslösern in Funktionsanforderungen) oder Generierung (z.B. Kontrollzustände aus Modusanforderungen)

Die durch das Modell vorgegebene Strukturierung und Klassifizierung von Anforderungen und die damit verbundene enge Kopplung mit den weiteren Modellierungselementen entscheidet dabei, welche Integrationsstufe erreicht wird und damit auch, welche Analyse- und Generierungsschritte werkzeuggestützt durchgeführt werden können. Zwei Beispiele dafür sind:

- *Integration unspezifischer Anforderungen*: Diese schwache Integration verwendet ein Produktmodell mit generischen hierarchischen Anforderungen, sowie domänenspezifische Sichten mit engerer Vernetzung der dazugehörigen Modellelemente. Die schwache Kopplung von Anforderungen und anderen Modellelementen erlaubt nur einfache Analysetechniken (z.B. Abdeckungseigenschaften) und generative Schritte (z.B. Dokumentation aus den Sichten).
- *Tiefe Integration domänenspezifischer Anforderungen*: Durch die Verwendung eines Produktmodells mit strukturierten, domänenspezifischen Anforderungen und Sichten wird eine starke Kopplung zwischen den Anforderungen und den übrigen Modellelementen möglich. Dies ermöglicht komplexere Analysetechniken (z.B. Konsistenz der Schnittstellen der architektonischen Anforderungen und der Schnittstellen in den Funktionsanforderungen) und generative Schritte (z.B. Generierung von Sichten aus strukturierten Anforderungen)

Die Machbarkeit und Tauglichkeit der hier beschriebenen Ansätze wird aktuell durch die Entwicklung eines Prototypen untersucht. Aufbauend auf dem Werkzeugprototypen AutoFocus [1] für die Entwicklung reaktiver Systeme wurde eine erste Integration unspezifischer Anforderungen vorgenommen [2]. Aktuelle Arbeiten vertiefen diese Integration durch die Einführung domänenspezifischer Anforderungsklassen.

- [1] Braun, P. Lötzbeyer, H., Schätz, B., Slotosch, O. Consistent Integration of Formal Methods. In: Proc. 6th TACAS'00. Springer, 2000
- [2] Chen, L. Anforderungsanalyse mit AutoFOCUS 2. Bachelor Thesis. TU München, 2003.
- [3] Heitmeyer, C. SCR: A Practical Method for Requirements Specification. In Proc., 17th Digital Avionics System Conference (DASC), 1998.
- [4] IEEE Std 830-1998. IEEE Computer Society, 1998.
- [5] IBM Rational RequisitePro Website [www.rational.com](http://www.rational.com)
- [6] Schätz, B. Pretschner, A. Huber, F. Philipps, J. Model-based Development of Embedded Systems. Technical Report TUMI-0402, Fakultät für Informatik, TU München, 2002
- [7] Telelogic DOORS Website [www.telelogic.com](http://www.telelogic.com)