

Modellgetriebene Generierung von natürlichsprachlichen Benutzeranleitungen

Seminararbeit
von

Andrei Miclaus

An der Fakultät für Informatik
Institut für Telematik

Erstgutachter:	Dr. Till Riedel
Zweitgutachter:	
Betreuender Mitarbeiter:	Dr. Till Riedel
Zweiter betreuender Mitarbeiter:	Prof. Dr. Michael Beigl

Ich versichere wahrheitsgemäß, die Arbeit selbstständig angefertigt, alle benutzten Hilfsmittel vollständig und genau angegeben und alles kenntlich gemacht zu haben, was aus Arbeiten anderer unverändert oder mit Abänderungen entnommen wurde.

PLACE, DATE

.....
(Andrei Miclaus)

Inhaltsverzeichnis

1	Einleitung	1
2	Installationsunterstützung in Smart-Environments	3
2.1	Studie zur Sensorinstallation für Heimautomatisierungssysteme	3
2.1.1	Verwendete Installationsanleitungen	3
2.1.2	Resultate der Studie	4
2.1.3	Schlussfolgerung der Studie	5
2.2	Proaktive Unterstützung während der Installation	6
2.2.1	Adaptierung an verschiedene Anwendertypen	6
2.2.2	Proaktives System	6
3	Werkzeuge und Ansätze zur Erstellung von Dokumentation und Anleitungen	7
3.1	Intelligente Augmented-Reality-Anleitungen	7
3.1.1	DFKI AR-Handbuch	7
3.1.2	Second-Sighted Glasses	8
3.2	Produktspezifische Dokumentengenerierung	9
3.2.1	DOPLER Metamodell	9
3.2.2	DOPLER Werkzeuge	10
3.2.3	Industrielle Anwendung	11
3.3	Modellgetriebene Generierung von Dokumentation	12
3.3.1	Topcased Gendoc	12
3.3.2	Gendoc Realisierung	13
3.4	Modellbasierte Erstellung von Dokumentation	13
3.4.1	Literate Modeling	13
3.4.2	Automatische Synchronisation für Literate Modeling	14
4	Gegenüberstellung der Ansätze	15
	Literaturverzeichnis	19

1. Einleitung

Heutzutage erlaubt der technologische Fortschritt eine Vielzahl von Geräten mit unterschiedlichen Anwendungs-, Vernetzungs- und Interaktionsmöglichkeiten kostengünstig in der Heimautomatisierung einzusetzen. Als Anwendungsgebiet der Smart-Environments zielt die Heimautomatisierung auf die Komfortsteigerung der Nutzer, durch die Ausstattung des Eigenheims mit intelligenten, vernetzten Geräten (Gadgets genannt). Gadgets sind flexibel, intelligent und größtenteils rekonfigurierbar. Aufgrund, der im System existierenden Interaktionen zwischen Gadgets und der großen Anzahl an Konfigurationsmöglichkeiten, entstehen komplexe Interaktionsmuster, die eine hohe kognitive Last für den Menschen erzeugen. Die Installation eines Heimautomatisierungssystems überfordert daher den Anwender. Installationsanleitungen sollen dabei eine Abhilfe schaffen, wobei laut [BCL04] das konzeptuelle Modell, dass sich in den Gedanken der Nutzer während der Installation bildet, eine wichtige Rolle im Verständnis übernimmt. Wie in [AMS02] aber angedeutet, sind heutige Benutzerhandbücher nicht in der Lage auf die verschiedenen Bedürfnisse oder auf die schon vorhandene Erfahrung der Anwender und den aktuellen Kontext einzugehen.

Die vorliegende Arbeit stellt im ersten Teil Erkenntnisse aus Studien zu Installationsaufgaben vor, um Anforderungen an installationsunterstützende Systeme in Smart-Environments zu sammeln. Im zweiten Teil der Arbeit sind Ansätze und Werkzeuge präsentiert, die zur flexiblen Erstellung von Dokumentationen und Anleitungen geeignet sind. Anschließend findet eine Gegenüberstellung der Ansätze und Werkzeuge, um ihre Eignung bezüglich der Anforderungen festzustellen. Die abgeleiteten Erkenntnisse sollen der Verbesserung von installationsunterstützenden Methoden und Technologien dienen.

2. Installationsunterstützung in Smart-Environments

Als erstes wird eine Studie zur Installation von Heimautomatisierungssystemen und deren Einsichten in den Installationsprozess der Anwender vorgestellt. Danach wird ein Proaktives System vorgestellt, dass aktive Unterstützung während des Installationsprozesses leistet.

2.1 Studie zur Sensorinstallation für Heimautomatisierungssysteme

Um Design-Prinzipien für die Verbesserung der Installation von ubiquitären Systemen zu erstellen, führen [BCL04] eine explorative Studie, zur Installation von Sensoren für ein Heimautomatisierungssystem, durch.

Die Autoren betrachten die Installation von Sensoren als eine Aufgabe mit zwei Dimensionen: *Platzierung* und *Assoziation*. Die Platzierung beeinflusst die Datenqualität beim Auslesen einer physikalischen Größe von einem Sensorknoten. Die Platzierung wird weiterhin in zwei Unterfaktoren aufgeteilt: *Proximität* und *Direktionalität*. Der Proximitätsfaktor gibt an wie nahe der Sensor an sein Messziel angebracht werden muss. Unter dem Direktionalitätsfaktor ist die Sensibilität eines Sensors auf Fehler in seiner Ausrichtung zu verstehen. Eine Kamera ist zum Beispiel stark darauf angewiesen, dass sie die richtige Szene aufnimmt. Ein Mikrofon hingegen hängt nicht so stark von seiner Ausrichtung ab. Die *Assoziation* bezieht sich auf die semantische Verbindung zwischen dem Datenstrom eines Sensors und des gemessenen Objekts in der realen Welt. Eine Eins-zu-Eins Assoziation ist für Menschen die einfachste Variante, weil sie leichter nachvollzogen wird. Die *Präzision* einer Assoziation wird als Funktion der Knotenplatzierung und dessen zu messenden Subjekts beschrieben. Es wird betont, dass die Assoziation eines proximalen Sensors leichter durchzuführen ist als die eines direktionalen Sensors. Das Beobachten von Objekten ist zudem leichter als das Beobachten von Räumen, weil die Reichweite eines Sensors schwer einzugrenzen ist.

2.1.1 Verwendete Installationsanleitungen

Die Teilnehmer der Studie erhalten vor Beginn des Experiments eine Übersicht des Konzeptes, dass hinter dem Heimautomatisierungssystem steht und nur die Platzierung, Ausrichtung und Assoziation der Sensoren bezüglich den von ihnen beobachteten Gegenstände

Tabelle 2.1: Unterschiede zwischen den Dokumentationen

Enthaltene Informationen	Dokumentation A	Dokumentation B	Dokumentation C
Installationsanleitung	✓	✓	✓
Gesammelte Sensordaten	✓	✓	x
Funktion der Sensoren	✓	x	x

wird betrachtet. In der Studie werden keine Aktuatoren oder Interaktionen zwischen den Sensoren betrachtet.

Die Studie ist auf den Umgang der Anwender mit der Installation von Sensoren fokussiert. Folglich erhalten die Teilnehmer verschiedene Anweisungen, um zu vermeiden, dass die Studie die Qualität der Nutzeranleitung bewertet.

In Tabelle 2.1 sind die Unterschiede zwischen den Dokumentationstypen aufgezeigt. Die Namen der Sensoren in Dokumentation A und B, repräsentieren die gemessene Größe (*Strom, Vibration, Bewegung, Bild, Geräusch*). Version C der Dokumentation benutzt nur Farben, um zwischen den Sensoren zu unterscheiden. Abbildung 2.1 zeigt einen Ausschnitt aus Dokumentation C. Man bemerkt, dass die Dokumentation wenig Informationen zum eigentlichen Verhalten und Zweck der anzubringenden Sensoren beinhaltet. Es wird erwähnt, dass es keine statistisch signifikanten Unterschiede, zwischen dem Erfolg der Installation im Zusammenhang mit den verschiedenen Dokumentationen, gab. Drei von fünf Nutzern mit Dokumentation C haben sich aber solche Informationen gewünscht.

Alle Sensoren sind mit Hilfe eines PDAs zu assoziieren, in dem der Sensor und das zu messende Objekt aus dem Katalog eingescannt wurden. Eine richtige Installation wird nach den Positionierungs- und Assoziationsfaktoren bewertet.

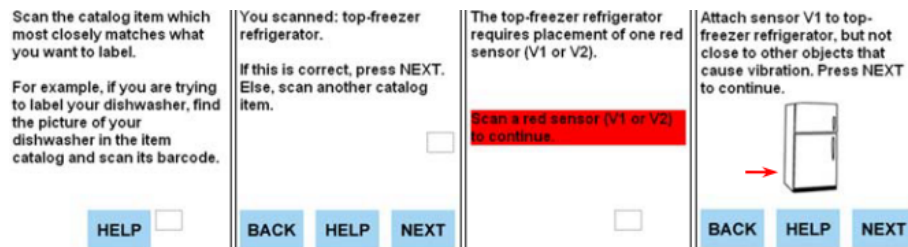


Abbildung 2.1: Ausschnitte aus der Dokumentation C, die auf einem PDA angezeigt wird [BCL04]

2.1.2 Resultate der Studie

Von den 150 Sensorinstallationsaufgaben (15 Teilnehmer x 10 Sensoren) sind 112 (75%) erfolgreich verlaufen. Nur 5 Teilnehmer (30%) haben alle Aufgaben korrekt durchgeführt.

- Das Einlesen mit dem Barcode-Scanner wurde von 13 aus 15 (86%) Teilnehmern korrekt durchgeführt.
- 2 der 15 Teilnehmer konnten keine der Aufgaben erfüllen und haben auch die Assoziation mit Hilfe des Barcodes nicht verstanden. Dieselben konnten weder den Anweisungen auf dem PDA folgen, noch wussten sie wie die Sensoren zu deaktivieren sind.
- Nur 5 aus den 10 Teilnehmern (50%), die Beschreibungen der von den Sensoren aufgenommenen Daten erhalten haben (Dokumentation A und B), konnten den konkreten Anwendungsfall der Sensoren nennen.

Der am meisten fehlerhaft installierte Sensor war der Geräuschsensor, mit 67% Erfolgsrate, gefolgt von dem Bildsensor, mit 70% Erfolgsrate. 4 der 9 fehlerhaften Bildsensorinstallationen folgten aus dem Übersehen oder nicht folgen der Ausrichtungsanweisungen.

Die Autoren motivieren die schlechten Erfolgsraten mit dem Problem der Anwendung von bekannten Technologien in ungewohnten Situationen. Im Vergleich dazu wurde der Stromsensor (ein Steckdosenaufsatz) in 87% der Fälle erfolgreich installiert (korrekte Assoziation 87% und korrekte Platzierung 93%). Neben den dokumentierten Installationen konnte ein Nutzer, mit Hilfe der zusätzlichen Sensorinformationen, einen Sensor erfolgreich an seinen nicht standardkonformen Kühlschrank anbringen.

2.1.3 Schlussfolgerung der Studie

Im Folgenden werden die fünf in [BCL04] erhobenen Design-Prinzipien für die Endnutzerinstallation von Sensoren aufgeführt und erläutert. Die ersten zwei Prinzipien lehnen sich an traditionelle Prinzipien aus der Mensch-Maschine-Interaktion und sind auch durch die Studie bestätigt. Die letzten drei sind Erkenntnisse die aus der Studie gewonnen wurden.

1. **Das konzeptuelle Modell der Benutzer für den Erfolg der Aufgabe nutzen**
Bekannte Technologie, die wie gewohnt benutzt wird unterstützt das Verständnis, da die schon bestehenden konzeptuellen Modelle der Nutzer wiederverwendet werden. Der umgekehrte Effekt entsteht jedoch, wenn vertraute Technologien anders als im Normalfall benutzt werden. Tritt der Fall auf müssen die Technologien im Namen als auch im Aussehen versteckt werden.
2. **Nutzbarkeit der Installation mit den häuslichen Anliegen im Gleichgewicht halten**
Designer und Anwender müssen Faktoren der Umgebung - den Kontext - in Betracht nehmen. Wenn Benutzer aus pragmatischen Gründen die Installation nicht durchführen wollen, treten technische Aspekte in den Hintergrund und die Installation wird abgebrochen.
3. **Die Anwendung von Kameras, Mikrofonen und hochdirektionalen Sensoren vermeiden**
Nicht nur die Sensitiven Daten die solche Sensoren aufnehmen können, sondern auch die verschiedenen Installationsmöglichkeiten in Verbindung mit unterschiedlichen Layouts der Wohnungen können diese Art von Sensoren problematisch machen.
4. **Inkorrekte Installationen sollten erkannt werden und auch Teilinstallationen sollten einen Mehrwert bieten**
Endnutzer wollen vielleicht einige Sensoren, die zum Beispiel sensitive Daten aufnehmen, nicht installieren, obwohl sie einen Mehrwert in der Heimautomatisierung sehen. Kann die Anwendung eine Teilinstallation entdecken und erlauben, so kann trotzdem ein Mehrwert geboten werden.
5. **Der Nutzer sollte über die Datensammlung, -speicherung und -übertragung unterrichtet werden**
Benutzer, die über die aufgenommenen Daten und deren Benutzung informiert sind, gehen mit der Installation besser um. Für die Akzeptanz der Anwendung ist es kritisch, dem Benutzer zu erklären wie die Daten benutzt werden und ob diese das Haus verlassen.

[BCL04] nennen einen weiteren Aspekt, der bei der Installation eines Heimautomatisierungssystems eine Erhöhung der Erfolgsraten ermöglicht. Erkannte Fehlinstallationen, durch die Analyse und den Vergleich von Sensordaten mit ihren Richtwerten, können dem

Anwender gemeldet werden. Dieses Real-Time-Feedback während der Installation kann das konzeptuelle Modell des Nutzers stärken und damit die Installation verbessern.

2.2 Proaktive Unterstützung während der Installation

2.2.1 Adaptierung an verschiedene Anwendertypen

Eine weitere Forschungsgruppe befasst sich mit der Unterstützung der Anwender bei Installationsaufgaben. Laut [AMS02] gibt es zwei Felder in denen Anwender von besseren Anleitungen profitieren:

1. Das Feld der Selbstmontage und -konfiguration aus dem *Do-It-Yourself*-Verbrauchermarkt, wie zum Beispiel bei Heimautomatisierung
2. Das Feld der Anwendungen wo, das Erreichen eines bestimmten Zustandes aus Sicherheitsgründen kritisch ist.

Eine Anleitung muss an das Nutzungsmodell der Anwenders angepasst werden, um ihn nicht über- oder unterzufordern. Die von den Autoren definierten Nutzungsmodelle sind im Folgenden beschrieben:

1. Das *Full-Walk-Through* richtet sich an Anfänger, die wenig Erfahrung und wenig Wissen über die zu erfüllende Aufgabe besitzen. Das Anleitungssystem liefert dieser Gruppe volle Unterstützung bei allen durchzuführenden Schritten.
2. Das *Assistance-on-Demand* richtet sich an Benutzer deren Wissenstand höher als der eines Anfängers ist, aber dennoch kleiner als den eines Experten. Diese Gruppe bevorzugt es eine Installation ohne Anleitung zu starten und bestimmen die Reihenfolge der durchzuführenden Schritte selber. Diese Benutzer sind sich aber bewusst, dass sie jederzeit auf die Anleitung zurückgreifen können.
3. Der *Rescue-from-Trap* Gruppe gehören Domänenexperten an, die nur selten eine Anleitung zur Rate ziehen. Bei dieser Gruppe muss aber das System darauf achten, dass wichtige Sicherheitsregeln beachtet und der Nutzer bei deren Gefährdung gewarnt wird.

Das Wichtige bei einer Anleitung ist es, so erklären die Autoren, den gewünschten Zustand zu erreichen und nicht eine bestimmte Reihenfolge der Schritte zu beachten. Diese Aussage stimmt auch mit der in [BCL04] gemachten Beobachtung, dass ein gutes konzeptuelles Modell, dem Benutzer ermöglicht Zustände des Systems ohne die Befolgung einer bestimmten Reihenfolge der Anweisungen zu erreichen. Die Anleitungen die heutzutage in Gebrauch sind, präsentieren dem Nutzer nur einen eingeschränkten, linearen Lösungsraum. Des weiteren erzeugen sie einen kognitiven Overhead während der Abbildung der Anweisungen von der graphischen Skizze des Lösungsraums und dem Text auf die Aufgabe in der realen Welt.

2.2.2 Proaktives System

Der in [AMS02] beschriebene Ansatz hebt die Limitierungen der heutigen Benutzeranleitungen durch proaktive Unterstützung auf. Sensoren, wie Beschleunigungs- oder Kraftmesser, die an die verschiedenen Bauteile eines aufzubauenden Objektes angebracht sind, melden den Status der Installation an ein Beobachtendes System, dass auf den Fortschritt in der Installation reagiert. Das System identifiziert unvollständig durchgeführte Installationsschritte (Schraube wird gedreht) anhand der Auswertung der Messdaten. Diese wiederum ermöglichen, unter der Verwendung eines Hidden Markov Model, die Identifikation von vollständig durchgeführten Schritten (Schraube wurde fertig gedreht).

Die Autoren weisen darauf hin, dass der so identifizierte Fortschritt der Installation, dem Anwender durch ein geeignetes Medium präsentiert werden muss. Die Autoren nennen dafür zukünftige Augmented-Reality Anwendungen, erwägen aber auch dynamische textuelle Dokumente. Weiterführend können auch die Bauteile, mit Hilfe von Lichtsignalen oder Vibration, die Aufmerksamkeit des Benutzers auf sich ziehen und somit die Installation unterstützen.

3. Werkzeuge und Ansätze zur Erstellung von Dokumentation und Anleitungen

Diese Kapitel stellt Werkzeuge und Ansätze zur Erstellung von Dokumentation und Anleitungen in zwei verschiedenen Medien vor. Auf der einen Seite sind Augmented-Reality-Ansätze präsentiert, auf der Anderen, Ansätze, aus denen textuelle Dokumentationen resultieren.

3.1 Intelligente Augmented-Reality-Anleitungen

3.1.1 DFKI AR-Handbuch

Das Augmented-Reality-Handbuch ist ein am DFKI¹ entwickeltes digitales Handbuch und Anleitungen mit Hilfe eines Head-Mounted-Display (HUD) in das Blickfeld des Benutzers projiziert [Nie13]. Jeder Schritt wird dem Benutzer vorgezeigt bis dieser erfolgreich durchgeführt wird. Die Erstellung von Anleitungsmaterial, das so genannte *Authoring*, für AR-Handbücher ist komplex und leidet unter hohen Zeit- und Geldaufwänden. In der Arbeit von [Nie13] ist dieses Problem mit Hilfe eines *programming by demonstration Ansatzes*, einer Technik aus dem Gebiet der künstlichen Intelligenz, gelöst. Das Authoring-Tool ist in der Lage eine einmal aufgenommene Video-Sequenz automatisch in einzelne Handlungsabläufe mit unterschiedlichen Merkmalen (*statisch, sich wiederholend, nicht wiederholend*) zu zerlegen. Mit einem stochastischen Übergangsmodell werden die Handlungsabläufe wieder zueinander in Verknüpfung gesetzt. Nach diesem Vorverarbeitungsschritt können mit dem Authoring-Tool (vgl. 3.1) neben den automatisch eingeblendeten Elementen auch manuelle Annotierungen, wie Pfeile oder andere graphische Symbole, eingefügt werden, um zusätzliche Informationen darzustellen. Eine in der AR-Brille integrierte Kamera überwacht die Durchführung der Aufgabe vom Nutzer. Die nächsten Schritte in der Anleitung werden die eingeblendeten Informationen mit den Handlungen des Benutzers synchronisiert. Wenn der Nutzer eine Aufgabe mit Hilfe des Systems durchführt (Abbildung 3.2(a)) werden zuerst die notwendigen Schritte für die Teilaufgabe, wie in Abbildung 3.2(b), eingeblendet. Nachdem der Nutzer mit der Ausführung der Teilaufgabe anfängt wird auf das Sichtfeld der Status der Durchführung mit Hilfe einer Farbkodierung angezeigt. Dabei bedeutet in 3.2(c) die grüne Kodierung eine korrekte Durchführung. Dieses System ist leichtgewichtig in dem Sinne, dass keine speziellen Marker benötigt werden und das Erlernen der durchzuführenden Schritte vom System übernommen wird. Die transparenten Einblendungen der

¹Deutsches Forschungszentrum für Künstliche Intelligenz

durchzuführenden Schritte (siehe Abbildung 3.2) werden automatisch von der Software erstellt.

Mit diesem Ansatz können Techniker Referenzabläufe aufnehmen um sicherzustellen, dass alle zukünftigen Aktivitäten in derselben Art und Weise durchgeführt werden.

Das AR-Handbuch soll in Zukunft auch auf Android Geräte laufen. Dies bedeutet, dass auch private Anwender, z.B. bei der Installation von Haushaltsgeräten, davon Gebrauch machen könnten.

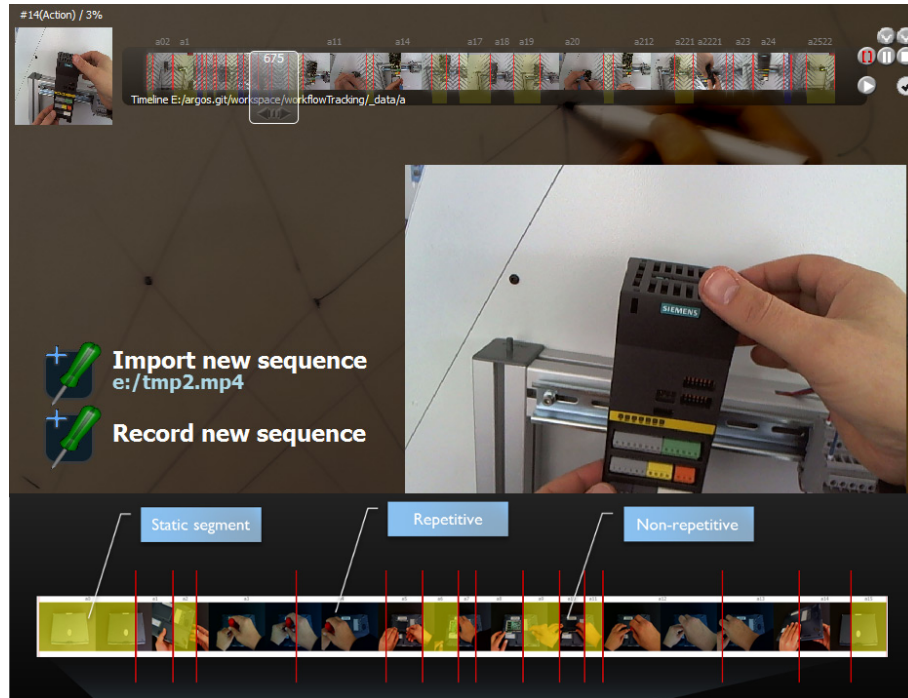


Abbildung 3.1: Authoring für das AR-Handbuch nach dem automatischen Vorverarbeitungsschritt, aus [Nie13]

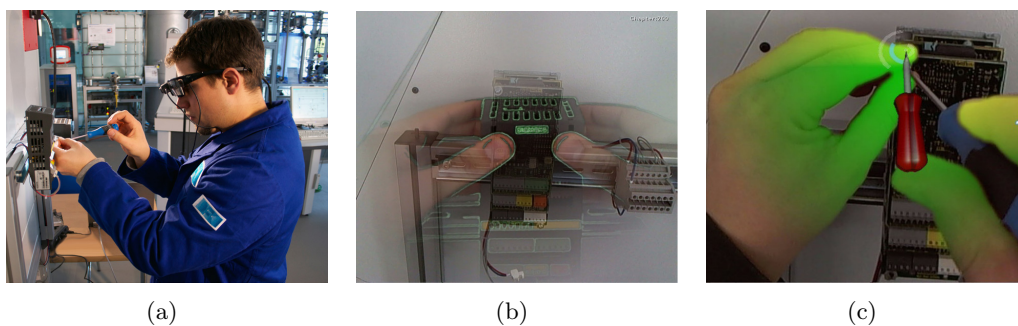


Abbildung 3.2: Das AR-Handbuch System aus [Nie13]

3.1.2 Second-Sighted Glasses

Ein zweiter Ansatz, auch auf AR basierend wird in [YNT11] vorgestellt. Bei dem Second-Sighted Glasses Ansatz handelt es sich um die visuelle Einblendung der verschiedenen Möglichkeiten, die ein Nutzer in der Interaktion mit den Objekten, die ihn umgeben, hat. Das Ziel ist es dem Benutzer zu zeigen mit welchen Aktionen bestimmte Ziele erfüllen. Die möglichen Varianten, auch *Futures* genannt, werden abhängig vom Kontext in dem

sich der Benutzer befindet angezeigt. Diese repräsentieren mögliche Abläufe die vom Kontextsensitiven Systemen definiert werden. Objekte sind mit Hilfe von Markern mit einer vorher definierten Aktion verbunden und lösen eine Mixed-Reality Animation aus, wenn sie in das Blickfeld des Anwenders kommen. Laut den Autoren muss noch weitere Forschung betrieben werden, um die automatische Generierung von Animationen aus den Kontextdefinitionen zu erlauben und um die Zusammenarbeit mit anderen Kontextsensitiven Systemen zu ermöglichen.

AR-Technologien werden noch nicht in der breiten Masse angewendet. Aber Zunehmend mehr Forschung in die Massentauglichkeit von AR-Brillen wird durchgeführt. CastAR [EJ13] zum Beispiel, ist ein AR-System, dass für rund 200 US-Dollar demnächst auf den Markt kommen soll. Kostengünstige Hardware könnte es Ansätzen wie in [Nie13] oder [YNT11] erlauben sehr schnell in den Nutzermarkt einzusteigen und breite Anwendung zu finden.

3.2 Produktspezifische Dokumentengenerierung

Die Erstellung von Benutzeranleitungen für industrielle Anwendungen ist neben der Komplexität der Systeme auch von ihrer Variabilität abhängig. Um die Generierung von Dokumenten als auch von Softwaresystemen aus demselben Variabilitätsmodell zu erlauben, wird in [RHE⁺10] ein Ansatz vorgestellt, der die Generierung der Dokumente mit der Konfiguration des eigentlichen Produktes steuert. Grundlegend bei der Konzipierung des Ansatzes ist die Beobachtung, dass viele Entscheidungen die während der Ableitungsphase des Produktes gemacht werden, sowohl für die Dokumentation als auch für das Softwaresystem relevant sind. Die technische Realisierung des Ansatzes besteht aus den DOPLER Werkzeugen [DRG⁺07, DGRN10] und wurde erfolgreich auf zwei industriellen Produktlinien angewandt.

Der in [RHE⁺10] beschriebene Ansatz ist unabhängig vom Reifegrad, der Variabilitätsmodellierung oder den zu generierenden Dokumenttypen. Bei der iterativen Durchführung der folgenden Schritte, sind Domänenexperten stark eingebunden.

1. Extrahieren und Analysieren der Domänenvariabilität
2. Variabilitäts-Modelle erstellen oder anpassen
3. Variabilitäts-Mechanismus und passenden Generator erstellen
4. Dokumente um Variabilitäts-Information erweitern

Durch die Extraktion der Domänenvariabilität durch Produktlinienexperten und Domänenexperten des Produkt-Managements oder des Vertriebs kann der wesentliche Teil der existierenden Variabilität identifiziert werden (Schritt 1). Mit den so gewonnenen Informationen ist es möglich, Variabilitätsmodelle zu erstellen oder anzupassen (Schritt 2). Die gewählte Modellierungstechnik sollte flexibel sein und fortgeschrittene Automatisierung während der Produktableitung zulassen. Um Variabilitätsinformation in den Dokumenten abzulegen benötigt man einen Mechanismus, wie zum Beispiel ein strukturiertes Dokumentenformat mit einer XML basierten Markupsprache (DocBook) (Schritt 3). Diese Informationen werden von einem Generator benutzt, um die produktspezifischen Dokumente zu erstellen (Schritt 4).

3.2.1 Dopler Metamodell

Die Realisierung des Ansatzes benötigt ein Metamodell, dass zwei Grundlegende Entitäten definiert: *Assets* und *Decisions* (vgl. Abbildung 3.3). Assets sind Artefakte der Produktlinie, wie Dokumente oder Technische Komponenten. Decisions hingegen sind die Entscheidungsbasis, anhand derer die Konfiguration der Assets stattfindet. Diese sind analog zu den

in der Variabilitätsmodellierung eingesetzten Entscheidungspunkten (Variation Points). Die im Metamodell vorhandene Abhängigkeitsbeziehung zwischen Assets modelliert die Funktionsabhängigkeit und den strukturellen Aufbau des Systems. Logische und hierarchische Zusammenhänge zwischen den zu treffenden Entscheidungen (Decisions) werden im Metamodell berücksichtigt und erlauben die Definition von Reihenfolge- und Verhaltensbeziehungen. Die DOPLER Werkzeuge sind in der Lage, Domänen, durch domänen-spezifische Metamodelle, mit beliebiger Granularität und Abhängigkeiten zu modellieren, solange deren Elemente von den Grundentitäten *Asset* und *Decision* erben.

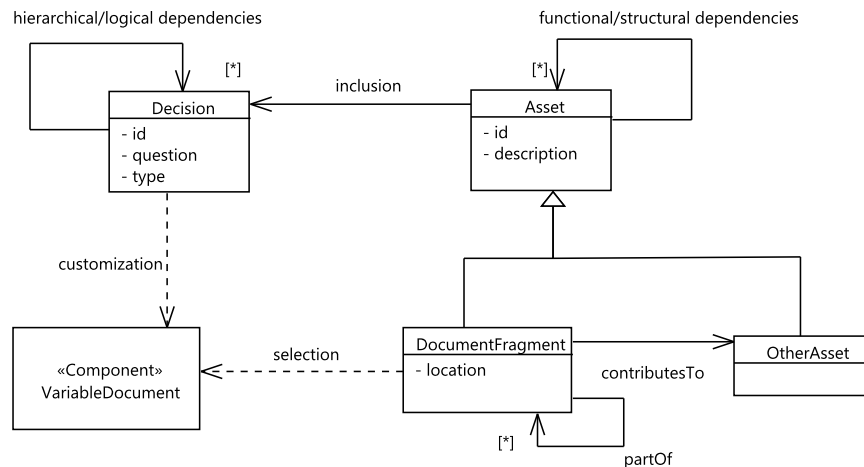


Abbildung 3.3: DOPLER Meta-Model mit Dokumenten aus [RHE⁺10].

Die Inklusion eines Asset in das abgeleitete Produkt geschieht durch seine Verbindung zu der Decision. Domänenexperten beantworten die zu der Decision gehörende Frage und instanzieren dadurch die konkrete Konfiguration. Durch diesen Mechanismus wird die Generierung der Dokumente gesteuert, indem ein Dokumentenfragment ein Asset ist, dass zu anderen Systemkomponenten eine Verbindung hat (siehe Abbildung 3.3). Ein Dokumentenfragment repräsentiert dabei Teile eines Dokumentes wie Kapitel oder Abschnitte. Dokumentfragmente und Komponenten des Systems können dadurch einheitlich behandelt werden. Dokumentfragmente erlauben die Modellierung der grobkörnigen Variabilität. Feinkörnige Anpassungen werden durch einen spezifischen Variabilitätsmechanismus realisiert.

3.2.2 Dopler Werkzeuge

Die DOPLER Werkzeuge benutzen den Profil-Mechanismus des DocBook-Standards², um Elemente und Attribute zu implementieren, die die Variationspunkte in den Dokumenten definieren.

Listing 3.1 zeigt ein Beispiel aus einem DocBook-Dokument mit der DOPLER Erweiterung. Das Kapitel *cooling* wird in die Dokumentation nur aufgenommen, wenn das Dokumentfragment *cooling_chapter* aufgrund einer Decision aufgenommen wurde. Durch den DocBook Mechanismus sind auch Querverweise möglich und Platzhalter können in Verbindung mit dem *doplerdocplaceholder* Element erstellt werden.

Listing 3.1: Das Element *doplerdocplaceholder* mit dem *doplerdoc* Attribut dient als Platzhalter für den Kühlungsmechanismus *cooling_mech* [RHE⁺10]

```
<chapter id="cooling" doplerdoc="cooling_chapter">
```

²www.docbook.org/


```

<para id="cooling_mechanism">
  The caster supports the
  <doplerdocplaceholder doplerdoc="cooling_mech"/>
  mechanism. ...
</para>
</chapter>

```

Der Ablauf zur Generierung von Dokumenten beginnt mit der Erstellung eines Variabilitätsmodells (Abbildung 3.4). Die Antworten auf die Fragen der Decisions werden durch den Nutzer mit Hilfe eines Konfigurationsassistenten (z.B. DOPLER Konfigurations-Wizard) in das Modell eingetragen. Das variable Dokument, das zentrale Artefakt im Prozess der Dokumentengenerierung, entsteht durch das Erweitern existierender Dokumentation mit der im Variabilitätsmodell existierenden Variabilitätsinformation und eine ständige Rückkopplung zu dem Variabilitätsmodell.

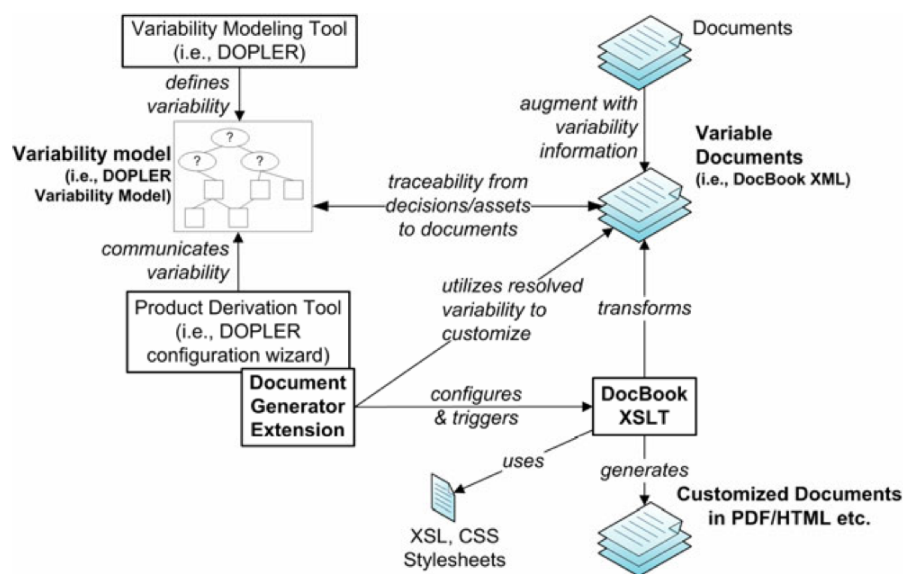


Abbildung 3.4: DOPLER Überblick über den Generierungsprozess der finalen Dokumentation aus dem Variabilitätsmodell in [RHE⁺10].

Der Dokumentengenerator passt mit Hilfe der aufgelösten Variabilität das Dokument-Template an und startet die DocBook-XSLT-Engine, um ein Dokument im gewünschten Ausgabeformat (z.B. PDF) zu erstellen. Formatierungsoptionen sind mit XSL oder CSS realisiert und sind durch Decisions konfigurierbar.

3.2.3 Industrielle Anwendung

Die in den Studien aus [RGD07] eingesetzte Werkzeugkette, besteht aus verschiedenen Ausprägungen des DOPLER Werkzeugs. Das DOPLER Variabilitäts-Modellierungs-Tool und der DOPLER Konfigurations-Wizard, der den Nutzer durch die Produktableitung führt, sind Teil davon. Als Generator dient eine domänenspezifische Erweiterung des DOPLER Konfigurationswerkzeugs.

Im ersten Anwendungsfall, bei der Siemens VAI, wurde auf einer schon bestehenden, automatisierten Software-Entwicklungskette aufgebaut. Ein Großteil der Decisions für die Dokumentengenerierung konnte aus den schon bestehenden Variabilitätsmodellen extrahiert werden. Für den restlichen Teil sind Dokumentenspezifische Decisions in Zusammenarbeit mit Domänenexperten erstellt worden. Ein einziges Entscheidungsmodell steuert die Produktkonfiguration als auch die Erstellung der technischen Benutzerdokumentation.

In der zweiten Industriellen Anwendung, bei der Siemens AG, mussten kundenspezifische Verkaufsdokumente erstellt werden, wobei es keine existierende Variabilitätsmodellierung gab. Zu den unten erwähnten Variabilitätspunkten kamen noch variantenabhängige Berechnungen der Preise und anderer Indikatoren hinzu. Außerdem gab es in diesem Fall mehr Alternativtexte und einige Variationspunkte die einen dokumentenübergreifenden Einfluss hatten.

In beiden Anwendungsfällen wurde mit Hilfe einer Befragung der Domänenexperten die am häufigsten auftretende Variabilität in Dokumenten gesammelt:

- Querverweise und deren Konsistenz
- Optionale Teile
- Platzhalter (z.B Name des Kunden)
- Grammatikalische Variation (Pluralbildung)
- Multi-Media-Objekte (z.B. Bilder)
- Formatierung und Layout (durch XSL oder CSS Style-Sheets realisierbar)

Der in [RHE⁺10] beschriebene Ansatz erlaubt es an den Entwicklungs und Generierungsprozesses von Software die Generierung von Dokumenten einzubinden. Die Austauschbarkeit der Werkzeuge zur Variabilitätsmodellierung und Generierung erlauben eine große Flexibilität bei der Anwendung auf verschiedene Domänen und unterschiedliche technische Gegebenheiten. Besonders nicht-technisches Personal wird durch die so entstandene Werkzeugkette in der Generierung von spezifischen Dokumenten unterstützt.

3.3 Modellgetriebene Generierung von Dokumentation

3.3.1 Topcased Gendoc

Ein weiteres Werkzeug zur automatischen Generierung von Dokumentation ist das in der Topcased Plattform enthaltene Gendoc, das aus Quellmodellen modellgetriebene Erstellung von Dokumentation ermöglicht. Die Model Driven Architecture (MDA), ein Standard zur modellgetriebenen Entwicklung der OMG[MM01], sieht eine Unterteilung in mehreren Modellebenen unterschiedlicher Abstraktion vor. Beginnend mit einem Modell der Domäne werden durch graduelle Verfeinerungen Details zur Bearbeitung von Informationen bis hin zu der Spezifizierung der konkreten technischen Implementierung, um aus den Modellen das fertige System zu generieren. Topcased³ ist ein Open-Source Werkzeug für modellgetriebene Entwicklung und dient dem System- und Software-Engineering von kritischen und embedded Anwendungen [PC06]. Die Plattform unterstützt alle Entwicklungsstufen von der Anforderungsanalyse bis zu der Implementierung. Sie basiert auf der Eclipse Plattform und benutzt das Eclipse Modeling Framework (EMF), um einen modellgetriebenen Arbeitsfluss zu realisieren. Es unterstützt neben UML Modellen auch andere von der OMG standardisierten Modellierungssprachen wie SysML⁴ und ReqIF⁵. Als Teil der Topcased Plattform erlaubt Gendoc die Generierung von textueller Dokumentation aus UML Modellen. Die Generierung wird von Skripten in den Dokumenten-Templates, die Logik und Modellabfragen beinhalten, gesteuert. Um die Modellabfrage zu realisieren sind Acceleo⁶ Skripte und Object Constraint Language (OCL)⁷ Ausdrücke möglich.

³<http://www.topcased.org/> ab 2014 auf <http://polarsys.org/>

⁴<http://www.sysml.org/>

⁵<http://www.omg.org/spec/ReqIF/>

⁶<http://www.eclipse.org/acceleo/>

⁷<http://www.omg.org/spec/OCL/>

3.3.2 Gendoc Realisierung

Acceleo Skripte, die direkt in den Dokumenten-Templates geschrieben werden lesen die Quellmodelle mit Hilfe von OCL aus und füllen das Dokument. Die benutzten Skripte, die in der Acceleo Sprache geschrieben sind, können beliebig Komplexe OCL Ausdrücke wie Schleifen oder Logik enthalten. Die Skripte sind in einer Markup-Ähnlichen Sprache eingebettet, die HTML ähnliche Unterstützung von Bildern und Listen besitzen.

Topcased unterstützt einen Großteil der UML Modelle, namentlich sind das Klassen-, Komponenten- und Deployment-Diagrammes, die die Struktur beschreiben und Sequenz-, Aktivitäts- und Zustandsdiagramme die die dynamischen Teile eines Systems beschreiben. Auch Use-Case Diagramme können modelliert, abgefragt und in die Dokumentation mit eingebunden werden.

Der generative Ansatz verläuft wie in Abbildung 3.5 dargestellt. Die Modelle und die Dokumenten-Templates sind die Eingaben des Systems, als Ausgabe liefert Gendoc das ausgefüllte Dokumenten-Template. Die Abspeicherung der ODT⁸ und DOCX⁹ Formate als XML, erlaubt es Gendoc die Skripte zu extrahieren und Auszuwerten. Da die Skripte in der Modelltransformationssprache Acceleo geschrieben sind können sie direkt auf die Modelle ausgeführt werden. Dieser Ansatz erlaubt es, alle von der EMF unterstützten Modellformate in Gendoc einzusetzen. Gendoc erlaubt es auch im Batch-Modus zu laufen, um automatisierte Werkzeugketten zu realisieren.

Beim Auswerten der Skripte behalten die Ausgaben die Formatierung bei. Dies bedeutet aber auch, dass der Quell-Code im Template-Dokument unterschiedliche, für Code untypische, Formatierungen besitzt, was das bearbeiten erschwert. Da aber die Formatierung des Dokuments beibehalten wird ist die dynamische Verlinkung mithilfe der Dokumenteneigenen Mechanismen möglich. Graphische Elemente, wie konkrete Diagramme können aus dem Quellmodell generiert werden und neben dem Text in das generierte Dokument eingefügt werden.

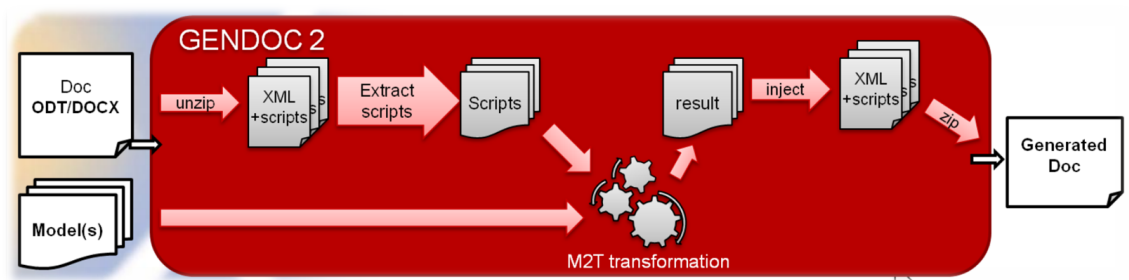


Abbildung 3.5: Architektur des Gendoc Werkzeugs

3.4 Modellbasierte Erstellung von Dokumentation

3.4.1 Literate Modeling

Im Gegensatz zur modellgetriebenen Entwicklung sieht die modellbasierte Entwicklung auch Modelle als Entwicklungsartefakte vor, jedoch werden diese Benutzt zu Dokumentationszwecken benutzt und nicht um daraus das fertige System zu bauen.

Unter *Literate Modeling* versteht man die von [AEQ99] eingeführte modellbasierte Methode, in der das Anforderungsdokument Systemmodelle zusammen mit narrativem Text vorsieht, um ein Gesamtbild des zu entwickelnden Systems zu geben. Das so entstandene

⁸Open Document Text

⁹Microsoft Word Dokumenten Format

Dokument, *Business Context Document (BCD)* genannt, beinhaltet neben dem visuellen Modell auch natürlichsprachliche Beschreibungen der UML Syntax, den Gedankengang der zu Entscheidungen führte, Beispiele zur Illustration bestimmter Aspekte und eine Umschreibung in natürlicher Sprache des Modells. Dieses Dokument schließt die Lücke zwischen den Modellierern und den nicht technischen Stakeholdern und erlauben somit eine bessere Beurteilung der formalisierten Anforderungen. Die Anwendung der Literate-Modeling Methode führt laut [AEQ99] zu einer Verbesserung in der Umsetzung von großen Softwareprojekten, weil das Wissen der Entwickler auch den Entscheidungsträgern zugänglich ist.

3.4.2 Automatische Synchronisation für Literate Modeling

Um die Methode des Literate-Modeling effizient zu nutzen, wird eine gute Werkzeugunterstützung benötigt, die die Konsistenz zwischen Modellen und dem beschreibenden Text gewährleistet. Aus diesem Grund ist in [SCOA12] ein Ansatz beschrieben, um die Synchronisation zwischen den Diagrammelement und den ihnen beschreibenden natürlichsprachlichen Texten durchzuführen. Der dort beschriebene Ansatz basiert auf die Annotierung des Textes und einem Synchronisationsalgorithmus, der natürliche Sprache in Verbindung mit OCL Ausdrücken verarbeitet. Um den Ansatz prototypisch zu testen wurde das Tool LiMonE¹⁰, als Adapter für die Eclipse UML2 Tools und den Papyrus Editor, in einer Masterarbeit, entwickelt. Die semantische Verbindung

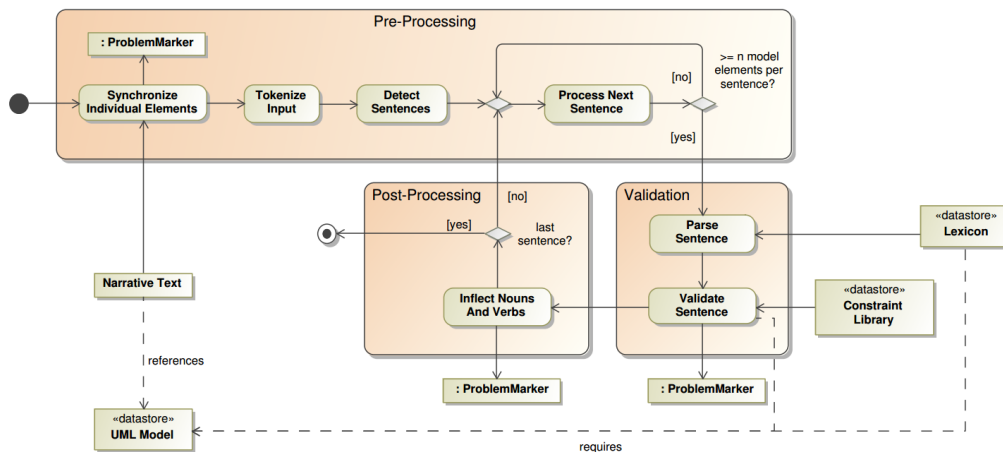


Abbildung 3.6: Der Synchronisierungsprozess als Aktivitätsdiagramm aus [SCOA12]

Um einzelne Elemente zu synchronisieren, werden Textannotationen benutzt, die eine Verbindung zwischen einem im Satz auftretenden Wort und dem Modellelement herstellen. Jeder Modellbeschreibende Satz wird einzeln gegen ein Set von Constraints verifiziert. Dabei wird die grammatikalische Struktur mit den in Modell enthaltenen Informationen überprüft. Für die Analyse der Satzstrukturen wurde ein Parser für natürliche Sprache verwendet¹¹. Die Satz-Constraints werden über logische Auflösung ausgewertet.

Durch die Verbindung der Wörter aus natürlichsprachlichem Text und den dazugehörigen Modellelementen ermöglicht diese System die Konsistenz zwischen der textuellen Dokumentation und den Modellen zu halten. So wird auch die Erstellung der Dokumentation aktiv unterstützt.

¹⁰<http://squam.info/limone>

¹¹Stanford PCFG Parser <http://nlp.stanford.edu/software/lex-parser.shtml>

4. Gegenüberstellung der Ansätze

Bei der Gegenüberstellung der Ansätze gilt der Rahmen der Installationsunterstützung bei Heimautomatisierungssystemen. Faktoren, die die Installationsunterstützung beeinflussen sind in Tabelle 4.1 aufgelistet. Eine einfache Skala von ++ (gute Unterstützung) bis -- (mangelhafte Unterstützung) dient dazu, jeden der Erwähnten Ansätze, zur Erstellung von Dokumentation und Anleitungen, qualitativ einzuschätzen. Ein x kennzeichnet hierbei die Nichterfüllung des Faktors. Gründe für die Bewertung jedes Faktors wird im Folgenden näher betrachtet.

Tabelle 4.1: Vergleich der Ansätze in Bezug auf wichtige Faktoren in der Installationsunterstützung

Faktoren in der Installationsunterstützung	AR-Handbuch	DOPLER	LiMonE	Gendoc
1. Geringer Authoring-Aufwand	++	+	+/-	--
2. Authoring benötigt wenig Expertenwissen	++	--	--	--
3. Anwendung benötigt wenig Expertenwissen	++	+	-	+
4. Verhalten bei Veränderung des Systems	-	++	--	--
5. Endnutzerperspektive(Attraktivität)	++	+/-	-	+/-
6. Zusammenhangsdarstellung	-	+	+	-/+
7. Modellabbildung auf die Realität	++	-	--	--
8. Eignung zur physischen Installation	++	+/-	+/-	+/-
9. Eignung zur Interaktionsbeschreibung	+	++	-	-
10. Erweiterbarkeit auf andere Domänen	++	-	-	--
11. Einbindung zusätzlicher Informationen	+	++	-	-
12. Navigationsunterstützung für Anwender	x	+	+	+

Geringer Authoring-Aufwand für Entwickler Die Synchronisation des Modells mit dem Text in LiMonE bietet beim Authoring volle Modellunterstützung, sodass Konsistente Inhalte schnell Erstellt werden können. In Gendoc müssen Skripte geschrieben werden, die Dank fehlender IDE-Features¹ schwer zu erstellen und warten sind. In

¹Integrated Development Environment

DOPLER werden einfache Annotationen zum Dokument hinzugefügt und ermöglichen sogar deren Auslagerung an nicht-technisches Personal. Für den AR-Ansatz gilt, je weniger Annotationen manuell eingefügt werden, desto kleiner bleibt der Authoring-Aufwand.

Authoring benötigt wenig Expertenwissen In Gendoc wird neben dem Domänenwissen auch Wissen über die Programmierung mit Aceleo und OCL verlangt und eignet sich nur für spezialisiertes Personal. Das Authoring im AR-Ansatz können Nutzer auch ohne Tiefe Kenntnisse über das zu beschreibende System ausführen. In DOPLER und LiMonE müssen fundierte Kenntnisse in den Anforderungen des Systems und seinen Komponenten besitzen.

Anwendung benötigt wenig Expertenwissen Die Anwendung ist beim AR-Ansatz sehr intuitiv. Da bei DOPLER und Gendoc die Dokument-Templates anpassbar sind, können Anleitungen erstellt werden die auch Nutzern mit wenig Vorwissen eine Durchführung der Aufgaben erlauben. LiMonE hingegen ist auf Anwender ausgelegt die entweder Wissen über die Anforderungserhebung oder UML besitzen.

Verhalten bei Veränderung des Systems Da DOPLER mit Produktlinien-Support im Sinne erstellt wurde ist es der Ansatz der am besten mit einem sich ändernden System umgehen kann. LiMonE und Gendoc müssen bei Veränderungen im System alle Stellen in den textuellen Dokumenten manuell geändert werden, was bei vielen Varianten prohibitiv werden kann. Das Selbe gilt auch für den AR-Ansatz, alle veränderten Stellen müssen neu aufgenommen und in die schon existierende Anleitung integriert werden.

Endnutzerperspektive(Attraktivität) Die Attraktivität für den Endanwender ist bei einer AR-Lösung sehr groß, wobei textuelle Darstellungen eher neutral bewertet werden. Die Erstellten Dokumente in LiMonE eignen sich für die Heimautomatisierung nicht sehr, weil die Beschreibung der Systemdetails nicht nötig sind und einen Anwender im Szenario der Heimautomatisierung überfordern würden.

Zusammenhangsdarstellung Der AR-Ansatz muss auf manuelles Authoring der Anleitung zurückgreifen, um Zusammenhänge darzustellen. Dadurch sind die Vorteile des automatischen Authoring nicht mehr gegeben. Außerdem gibt es kein zentrales Repository, dass die Zusammenhangsinformationen beinhaltet. Im Gegensatz dazu, haben die anderen Ansätze den Vorteil eines Modells mit den modellierten Zusammenhängen. In Gendoc aber müssen diese Zusammenhänge mit Hilfe des imperativen Skriptes ausgewertet werden, was einen starken Overhead beim Authoring hervorruft.

Modellabbildung auf die Realität Der AR-Ansatz ermöglicht eine 1:1 Abbildung durch die Einblendung in das visuelle Sichtfeld des Benutzers. DOPLER kann im Gegensatz zu LiMonE und Gendoc, die Modelle besser auf die Realität abbilden, weil die Variabilitätsmodelle mit abstrakteren Konstrukten als UML Modelle arbeiten.

Eignung zur physischen Installation In LiMonE müssen dafür einmalig weitere Diagrammtypen integriert und entsprechende Constraints definiert werden. In DOPLER, LiMonE und GendocIn können Schaubilder oder Textfragmente, die nicht zum Modell gehören, hinzugefügt werden, um zusätzliche Informationen zur Unterstützung der Installation zu liefern. In LiMonE und Gendoc existieren diese zusätzlichen Informationen aber nur in den Dokumententemplates und es gibt keinen expliziten Steuerungsmechanismus dafür. DOPLER ist in der Lage diese zusätzlichen Informationen durch Variabilitätsmodell zu steuern.

Eignung zur Interaktionsbeschreibung Wiederum ist dies in LiMonE nur möglich, wenn das es mit neuen Diagrammtypen erweitert wird. In der AR-Lösung ist dies kaum

möglich, weil Interaktionsmuster in einer Einblendung mit der Realität nur umständlich darzustellen sind und alle Vorteile des Authoring Werkzeug verloren gehen.

Erweiterbarkeit auf andere Domänen In diesem Punkt stellt ist AR-Lösung sehr gut dargestellt, denn sie kann ohne zusätzlichen Aufwand auf unterschiedliche Domänen oder Systeme angewandt werden. Die modellbasierten und modellgetriebenen Verfahren benötigen dazu erstmals ein Model des neuen Systems und die Erstellung von Dokumenten-Templates. In DOPLER existiert sogar eine größere Kopplung zwischen dem System und dem Konfigurationsmechanismus für Dokumente.

Einbindung zusätzlicher Informationen Im AR-Ansatz, LiMonE und Gendoc besteht die Möglichkeit Bilder, Zeichnungen oder zusätzlichen Text einzubinden. Da diese nicht von den Systemen Verwaltet werden, verfügen sie über eine geringe Wiederverwendung und Inkonsistenzen werden nicht automatisch entdeckt. DOPLER erlaubt es verschiedene Informationsquellen in das Variabilitätsmodell einzubinden. Das Modell steuert somit die Verwendung dieser Informationen und bietet volle Unterstützung in der Darstellung.

Navigationsunterstützung für Anwender Der AR-Ansatz bietet keinen Mechanismus für die Navigation durch die virtuelle Anleitung. Die durchzuführenden Schritte müssen linear Abgelaufen werden. Die anderen Ansätze hingegen erlauben die Nutzerführung durch dynamische Links, falls diese in einem elektronischen Format dargestellt werden.

Laut [PBvdL05] ist die Heimautomatisierung eine Komplexe Anwendung, die dank der vielen Interaktions- und Konfigurationsmöglichkeiten einer Produktlinie ähnelt. Die Anpassung an den Kundenwunsch, verschiedene Kombinationen von Heimautomatisierungsprodukten (wie z.B. Heizungs-, Licht-, Fenstersteuerung o.ä.) anzuschaffen, verkörpert die Variabilität im Sinne von Produktlinien. Je nach Konfigurationswunsch sind diese verschiedenen Varianten des Heimautomatisierungsproduktes auszuliefern und in verschiedenen Arten zu installieren und zu warten. Der Kontext des Benutzers spielt dabei eine wesentliche Rolle. Somit muss auch die Variabilität des zu installierenden Systems in Betracht genommen werden.

Die Domäne in denen die verschiedenen Ansätze angewendet werden sind im Folgenden aufgeführt.

AR-Handbuch Unterstützung für Installationen, Wartungsarbeiten oder andere Tätigkeiten, die sichtbare Komponenten enthalten.

Dopler Gedacht für den industriellen Einsatz in Umgebungen mit Produktfamilien. Erstellt Produktbeschreibungsdokumente die zu den ausgelieferten Systemen passen.

LiMonE Wird primär für Anforderungsdokumente angewendet, ist aber noch im Prototypstadium

TopCased Dient primär zur Erzeugung von Architekturbeschreibungen im industriellen Umfeld.

Wie aus Tabelle 4.1 ersichtlich wird, deckt kein Ansatz alle Faktoren ausreichend ab. Eine Installationsunterstützung muss sowohl intuitiv und leicht zu Bedienen als auch das konzeptuelle Modell des Nutzers unterstützen. Sind die Abhängigkeit und Assoziationsinformationen aber nicht formalisiert, so können diese nicht implizit in die Anleitung eingebaut werden und die Wartung ist mit der steigenden Zahl an Varianten nicht mehr durchführbar.

Literaturverzeichnis

- [AEQ99] Jim Arlow, Wolfgang Emmerich und John Quinn: *Literate Modelling - Capturing Business Knowledge with the UML*. In: *UML'98: Selected papers from the First International Workshop on The Unified Modeling Language UML'98*, Seiten 189–199, London, UK, 1999. Springer-Verlag, ISBN 3-540-66252-9. [citeulike-article-id:7705903http://portal.acm.org/citation.cfm?id=760158](http://portal.acm.org/citation.cfm?id=760158).
- [AMS02] Stavros Antifakos, Florian Michahelles und Bernt Schiele: *Proactive Instructions for Furniture Assembly*. In: *Proceedings of the 4th international conference on Ubiquitous Computing*, UbiComp '02, Seiten 351–360, London, UK, UK, 2002. Springer-Verlag, ISBN 3-540-44267-7. <http://dl.acm.org/citation.cfm?id=647988.741481>.
- [BCL04] Chris Beckmann, Sunny Consolvo und Anthony LaMarca: *Some Assembly Required: Supporting End-User Sensor Installation in Domestic Ubiquitous Computing Environments*. In: Nigel Davies, ElizabethD. Mynatt und Itiro Siio (Herausgeber): *UbiComp 2004: Ubiquitous Computing SE - 7*, Band 3205 der Reihe *Lecture Notes in Computer Science*, Seiten 107–124. Springer Berlin Heidelberg, 2004, ISBN 978-3-540-22955-1. http://dx.doi.org/10.1007/978-3-540-30119-6_7.
- [DGRN10] Deepak Dhungana, Paul Grünbacher, Rick Rabiser und Thomas Neumayer: *Structuring the modeling space and supporting evolution in software product line engineering*. *J. Syst. Softw.*, 83(7):1108–1122, Juli 2010, ISSN 0164-1212. <http://dx.doi.org/10.1016/j.jss.2010.02.018>.
- [DRG⁺07] D Dhungana, R Rabiser, P Grünbacher, K Lehner und C Federspiel: *DOPLER: an adaptable tool suite for product line engineering*. 11th International Software Product Line Conference (SPLC 2007), Seiten 10–14, 2007.
- [EJ13] Jeri Ellsworth und Rick Johnson: *Augmented Reality Glasses: Ex-Valve-Employees CastAR*, 2013. <http://www.pcgamesn.com/valve-were-making-augmented-reality-glasses-ex-employees-carrying-project-forward> besucht: 01/08/13.
- [MM01] Joaquin Miller und Jishnu Mukerji: *Model Driven Architecture (MDA)*. Another overview of the MDA, 2001.
- [Nie13] Petersen Niels: *Intelligente Augmented Reality Handbücher Zeigen, wie's geht - Werkerunterstützung für die Fabrik der Zukunft*, 2013. http://av.dfki.de/projects_recent/ar-handbook, besucht: 16.07.2013.
- [PBvdL05] Klaus Pohl, Günter Böckle und Frank J. van der Linden: *Software Product Line Engineering: Foundations, Principles and Techniques*. Springer-Verlag New York, Inc., September 2005, ISBN 3540243720. <http://dl.acm.org/citation.cfm?id=1095605>.

- [PC06] N Pontisso und D Chemouil: *TOPCASED Combining Formal Methods with Model-Driven Engineering*. In: *Automated Software Engineering, 2006. ASE '06. 21st IEEE/ACM International Conference on*, Seiten 359–360, 2006.
- [RGD07] Rick Rabiser, Paul Grunbacher und Deepak Dhungana: *Supporting Product Derivation by Adapting and Augmenting Variability Models*. In: *Proceedings of the 11th International Software Product Line Conference, SPLC '07*, Seiten 141–150, Washington, DC, USA, 2007. IEEE Computer Society, ISBN 0-7695-2888-0. <http://dx.doi.org/10.1109/SPLC.2007.34>.
- [RHE⁺10] Rick Rabiser, Wolfgang Heider, Christoph Elsner, Martin Lehofer, Paul Grünbacher und Christa Schwanninger: *A flexible approach for generating product-specific documents in product lines*. In: *Proceedings of the 14th international conference on Software product lines: going beyond, SPLC'10*, Seiten 47–61, Berlin, Heidelberg, 2010. Springer-Verlag, ISBN 3-642-15578-2, 978-3-642-15578-9. <http://dl.acm.org/citation.cfm?id=1885639.1885645>.
- [SCOA12] Gunnar Schulze, Joanna Chimiak-Opoka und Jim Arlow: *An Approach for Synchronizing UML Models and Narrative Text in Literate Modeling*. In: RobertB. France, Jürgen Kazmeier, Ruth Breu und Colin Atkinson (Herausgeber): *Model Driven Engineering Languages and Systems SE - 38*, Band 7590 der Reihe *Lecture Notes in Computer Science*, Seiten 595–608. Springer Berlin Heidelberg, 2012, ISBN 978-3-642-33665-2. http://dx.doi.org/10.1007/978-3-642-33666-9_38.
- [YNT11] Takuro Yonezawa, Jin Nakazawa und Hideyuki Tokuda: *Retrieving Environmental Smartness with Second-sighted Glasses*. Technischer Bericht ITE Tech. Rep., vol. 35, no. 9, HI2011-9, Keio University, 2011. <http://www.ite.or.jp/ken/paper/000000001ABY/eng/>.