

Papiergestützte Service-Modellierung für Endbenutzer

**Konzeption und Entwicklung eines papierbasierten
Interaktionskonzeptes zur endbenutzergerechten Unterstützung
der visuellen Orchestrierung von Web Services**

Diplomarbeit

zur Erlangung des akademischen Grades
„Diplom-Wirtschaftsinformatiker“

Bearbeiter: Björn Borggräfe
Matrikelnummer: 681571

Betreuer: Dipl.-Wirt.-Inform. Christian Dörner

Erstprüfer: Prof. Dr. Volkmar Pipek

Zweitprüfer: Prof. Dr. Volker Wulf

Einreichung: 03. September 2008

Erklärung

Hiermit versichere ich, dass ich die vorliegende Arbeit selbständig verfasst und keine anderen als die angegebenen Quellen und Hilfsmittel benutzt habe, insbesondere keine anderen als die angegebenen Informationen aus dem Internet.

Diejenigen Paragraphen der für mich gültigen Prüfungsordnung, welche etwaige Betrugsversuche betreffen, habe ich zur Kenntnis genommen.

Der Speicherung meiner Diplomarbeit zum Zweck der Plagiatsprüfung stimme ich zu. Ich versichere, dass die elektronische Version mit der gedruckten Version inhaltlich übereinstimmt.

Siegen, 03. September 2008

Björn Borggräfe

Abstract

Die Möglichkeit zur flexiblen Anpassung der eigenen Prozesse an sich verändernde Anforderungen ist ein wichtiges Ziel für Unternehmen, das auf technischer Ebene häufig durch die Konzepte des Geschäftsprozessmanagements und der serviceorientierten Architekturen angestrebt wird. Auf organisatorischer Ebene spielt für eine flexible Prozesserstellung vor allem die Einbeziehung der Fachanwender (i.F. Endbenutzer) eine wichtige Rolle. Bestehende (endbenutzergerechte) Modellierungssysteme bieten jedoch kaum Unterstützung für kollaborativer Modellierungstätigkeiten, für die Abbildung informeller Aspekte und für eine Erstellung komplexerer Prozesse. Dies ermöglicht es Endbenutzern nicht sich aktiv an der Prozesserstellung zu beteiligen und bedingt sich zum Teil durch Einschränkungen graphischer Benutzerschnittstellen. Neuartige Eingabekonzepte auf Basis von Papier und Skizzen besitzen dagegen, aufgrund von Flexibilitätsvorteilen, einer natürlichen Nutzung und einer guten Unterstützung kreativer Aspekte, Vorteile für kollaborative und kreative Nutzungsprozesse. Motiviert durch die Chancen eines Papierinterfaces nimmt sich die Arbeit der Forschungslücke eines fehlenden Systems für eine endbenutzergerechte Prozessmodellierung an, indem sie ein Konzept aufzeigt, das es Endbenutzern durch eine semi-formale Modellierung auf einem Papierinterface ermöglicht, sich aktiv in die erste Phase der Prozesserstellung einzubringen. Ein Kernaspekt ist dabei die Integration endbenutzerorientierter, papierbasierter und serviceorientierter Gestaltungsaspekte in einem geeigneten Interaktions- und Modellierungskonzept. Zudem liefert die Arbeit wichtige Hinweise auf das Potential des natürlichen Interaktionsmediums Papier für den Bereich des End-User Developments und zeigt Chancen und Herausforderungen eines neuen Nutzungsfeld (der Modellierung) für papierbasierte Anwendungen auf.

Inhaltsverzeichnis

Erklärung	I
Abstract	II
Inhaltsverzeichnis	III
1 Einleitung	1
1.1 Motivation und Forschungsfrage.....	1
1.2 Ziele	3
1.3 Aufbau der Arbeit.....	4
2 Stand der Forschung	6
2.1 Prozessmodellierung in serviceorientierten Architekturen.....	6
2.1.1 Grundlagen des Geschäftsprozessmanagements	6
2.1.2 Grundlagen serviceorientierter Architekturen	8
2.1.3 Serviceorientierte Prozessmodellierung	13
2.2 Softwareanpassung durch Endbenutzer.....	16
2.2.1 Grundlagen des End-User Development	17
2.2.2 Gestaltung von EUD-Systemen.....	19
2.2.3 EUD in SOA.....	25
2.3 Papier- und skizzenbasierte Modellierung	28
2.3.1 Grundlagen papierbasierter Systeme	28
2.3.2 Chancen und Herausforderungen papierbasierter Systeme	30
2.3.3 Grundlagen papier- und skizzenbasierter Modellierung	32
2.3.4 Interaktive Papier-Anwendungen	33
2.3.5 Übersicht elektronischer skizzenbasierter Designwerkzeuge.....	35
2.3.6 Interaktionskonzepte papierbasierter Sketching-Anwendungen	37
2.4 Zusammenfassung und Fokus	39
3 Vorstudien: Modellierungspraxis	41
3.1 Fallstudie: BPM in SOA in der Praxis	41
3.1.1 Konzeption und Methodik.....	41
3.1.2 Untersuchungsgegenstand	42
3.1.3 Ergebnisse und Auswertung	43
3.1.4 Bewertung und Fazit.....	48
3.2 Design-Workshop: Kollaboratives BPM durch Endbenutzer	49
3.2.1 Methodik.....	50
3.2.2 Konzeption	50
3.2.3 Ablauf	51

3.2.4	Ergebnisse.....	52
3.2.5	Bewertung und Fazit.....	53
4	Konzeption einer papierbasierten Servicemodellierung.....	54
4.1	Gestaltungsrahmen	54
4.1.1	Positionierung und Abgrenzung des Konzeptes.....	54
4.1.2	Idealtypischer Modellierungsablauf	55
4.1.3	Rahmenbedingungen und Einflussfaktoren.....	56
4.1.4	Diskussion kognitiver Gestaltungsdimensionen.....	57
4.2	Anforderungen an die Gestaltung von PaSeMod	60
4.2.1	Allgemeine Anforderungen	60
4.2.2	Anforderungen an das Modellierungskonzept.....	62
4.2.3	Anforderungen an die Prozessnotation.....	64
4.2.4	Anforderungen an die Feedbackgestaltung	66
4.2.5	Übersicht der Anforderungen	67
4.3	Aufbau der Grundkonzeptes.....	67
4.4	Papierbasiertes Interaktionskonzept	69
4.4.1	Papierbasiertes Grundkonzept	69
4.4.2	Differenzierung der Stiftmodi	72
4.4.3	Visualisierung des Prozessablaufes	73
4.4.4	Visualisierung der Prozessdetails	76
4.4.5	Erweitertes Interface.....	77
4.4.6	Beispielhafter Ablauf der Modellierung.....	78
4.5	Feedbackkonzept	80
4.5.1	Feedbackarten.....	80
4.5.2	Identifikation von benötigtem und sinnvollem Feedback	81
4.5.3	Gestaltung des Feedbacks.....	82
4.5.4	Gestaltung des visuellen Feedbacks	84
4.6	Weiterverarbeitung	84
4.7	Zusammenfassung des PaSeMod-Konzeptes	84
5	Realisierung von PaSeMod	87
5.1	Architektur des Systems	87
5.1.1	Architekturkonzept.....	87
5.1.2	Technische Anforderungen	90
5.2	Technologieauswahl und verwendete Technologien.....	90
5.2.1	Interaktive Papiertechnologie.....	91
5.2.2	Interaktives Papier- und Sketching-Framework	92
5.2.3	Sonstige Technologien	94

5.3	Implementierung von PaSeMod	95
5.3.1	Einschränkung der zu realisierenden Systemaspekte	95
5.3.2	Realisiertes PaSeMod-System	96
5.4	Zusammenfassung der PaSeMod-Realisierung	100
6	Evaluation von PaSeMod	102
6.1	Ziel	102
6.2	Methodik	102
6.3	Konzeption des Workshops	103
6.3.1	Teilnehmer	104
6.3.2	Untersuchungsrahmen	104
6.3.3	Ablauf des Workshops	105
6.4	Ergebnisse	106
6.4.1	Technische Hindernisse	107
6.4.2	Benutzbarkeit	107
6.4.3	Brauchbarkeit	111
6.5	Zusammenfassung und Bewertung	113
7	Zusammenfassung und Ausblick	114
	Danksagung	VI
	Literaturverzeichnis	VII
	Abbildungsverzeichnis	XX
	Abkürzungsverzeichnis	XXII
	Tabellenverzeichnis	XXIII
	Anhang	XXIV

1 Einleitung

Die grundlegende Motivation für die Arbeit ist die Schaffung von Methoden und Konzepten, die Anwender von Informationstechnologie dazu befähigen, ihre Systeme mit wenig Aufwand an persönliche Anforderungen und Bedürfnisse anzupassen. Dazu soll vor allem Anwendern ohne ausgeprägtes technisches Wissen ein guter Zugang zu der benötigten Funktionalität und ein weicher Einstieg in Anpassungsprozesse ermöglicht werden. Vor diesem Hintergrund erscheint als spannende Fragestellung, wie die in der Welt der meisten Nutzer allgegenwärtigen, natürlichen Medien Stift und Papier dies geeignet unterstützen können. Im Folgenden wird zu Beginn die zentrale Fragestellung der Arbeit motiviert und beschrieben. Anschließend werde die Ziele und der Aufbau der Arbeit vorgestellt.

1.1 Motivation und Forschungsfrage

Ein wichtiger Aspekt für Unternehmen ist die Möglichkeit, die eigenen Geschäftsprozesse und Systeme flexible an dynamische Entwicklungen und sich verändernde Umweltbedingungen anpassen zu können (vgl. Wulf [172]). Um diese Flexibilität auf organisatorischer und technischer Ebene zu erreichen, kommt der Verbindung der Konzepte des *Geschäftsprozessmanagements* (im folgenden *BPM* für *Business Process Management*) und der *serviceorientierten Architekturen (SOA)* eine große Bedeutung zu. Im Fokus von BPM steht die Betrachtung der Geschäftsprozesse eines Unternehmens und speziell die Modellierung und Ausführung von Geschäftsprozessen als eine zeitlich-logische Abfolge von Aktivitäten. Durch die Möglichkeit, einzelne Aktivitäten (die durch sogenannte „Services“ repräsentiert werden) auf technischer Ebene flexibel und dynamisch zu Prozessen zusammenstellen zu können („lose Kopplung“ von Services), bietet SOA eine geeignete technische Basis für ein flexibles BPM.

Neben diesen technischen Aspekten spielt für die Flexibilität auf organisatorischer Ebene die Einbeziehung der Anwender (= „Endbenutzer“), die in den Prozessen arbeiten und daher das entsprechende Detailwissen über die Prozessabläufe besitzen, eine bedeutende Rolle. Professionelle Modellierer und Entwickler, die nicht aus den Anwendungsbereichen stammen, verfügen meist nicht über das für die Prozessmodellierung nötige domänenspezifische Fachwissen. Daher sind, wie Brahe und Schmidt [23] feststellen, bei der Prozesserstellung Austauschprozesse zwischen professionellen Entwicklern und Endbenutzern wichtig. Darüber hinaus können Endbenutzer, wenn sie selbst die Möglichkeit besitzen Prozess zu erstellen, flexibler und schneller an individuelle Informationen gelangen (vgl. Han et al. [63]), da unter anderem ein hoher Kommunikationsaufwand zwischen Endbenutzern und Entwicklern vermieden wird (vgl. Wulf und Jarke [173]).

Allgemein erscheint das Grundkonzept von BPM und SOA, durch die Möglichkeit Prozesse flexibel zusammenzustellen, sowie einem Fokus auf Geschäftsprozesse (und damit die Anwendungsdomäne der Nutzer), geeignet für Endbenutzer. In der Praxis gestaltet sich die Einbeziehung von Endbenutzern jedoch schwierig (vgl. [23]). So sind, wie verschiedene Untersuchungen zeigen (z.B. [63, 104, 161]), professionelle Modellierungswerkzeuge, aufgrund ihrer Konzeption für professionelle Entwickler und einer daraus resultierenden Komplexität und schlechten Erlernbarkeit, für Endbenutzer ungeeignet. Dies resultiert darin, dass Endbenutzer meist nur zu Beginn einer Prozessentwicklung beteiligt sind und nicht aktiv an der Prozessmodellierung teilnehmen. Damit auch Endbenutzer sich aktiv einbringen können, werden spezifische Werkzeuge benötigt, welche die Anforderungen der Endbenutzer berücksichtigen. Um dies zu erreichen, wurden vor allem unter dem Begriff des *End-User Developments (EUD)* (vgl. Lieberman et al. [103]) verschiedene Methoden entwickelt, die auch Endbenutzern zu einem gewissen Grad eine Entwicklung, zum Beispiel von Prozessen, ermöglichen sollen. Unter Berücksichtigung dieser Methoden, wurden verschiedene endbenutzergerechte Modellierungswerkzeuge entwickelt. Beispielhaft genannt seien hier die Arbeiten von Liu et al. [104], Oinn et al. [130] und Paczynski [135].

Diese endbenutzergerechten Modellierungswerkzeuge bieten eine gute Unterstützung für einfache Prozesse mit wenigen Elementen und für eine Modellierung durch einzelne Benutzer. In der Praxis finden sich jedoch auch vielfach Szenarien, in denen eine gemeinsame (= kollaborative) Modellierung von mehreren Benutzern stattfindet. So werden zum einen Prozesse oft gemeinsam von Endbenutzern und professionellen Modellierern erstellt (vgl. Brahe und Schmidt [23] sowie Loser [106]), zum anderen führen Endbenutzer Entwicklungsaktivitäten häufig gemeinsam mit anderen Endbenutzern durch (vgl. Nardi [121]). Endbenutzergerechte Modellierungswerkzeuge sind aufgrund der beschränkten Modellierungsfläche eines Bildschirms wenig geeignet für eine kollaborative Modellierung, sowie für die Abbildung komplexer Prozesse. Auch unterstützen diese Systeme keine Darstellung informeller Aspekte. Dies ist jedoch ein wichtiger Aspekt, vor allem einer frühen Phasen der Prozessmodellierung (Damm et al. [36]), und eine Grundvoraussetzung für ein „unbeschränktes Modellierungssystem“¹, wie es Jarzabek et al. [78] fordern. Durch eine informelle Modellierung lassen sich Prozessaspekte ausdrücken, für die keine formale Darstellung bekannt ist. Zudem kann die Notation einfach um eigene Elemente ergänzt werden.

Eine Alternative zur Modellierung am Rechner ist die Nutzung von *papierbasierten Benutzerinterfaces (PBUIs)*, die auf einer interaktiven Stift-und-Papier-Technologie aufbauen.

¹ Ein „unbeschränktes Modellierungssystem“ ist nach Jarzabek et al. nicht ausschließlich an vordefinierte Modellierungselemente gebunden, sondern ermöglicht eine „unbeschränkte“ Modellierung, z.B. durch die Kombination von formalen Elementen mit informellen Skizzen.

Allgemein beginnen Nutzer, wenn sie mit einem Problem (z.B. der Modellierung eines Prozesses) konfrontiert werden, häufig mit einer ersten Lösungsskizze auf Papier (siehe Snyder [158]). Auch im Bereich des partizipativen Systemdesigns wird Papier, aufgrund seiner Flexibilität und Natürlichkeit, in Form von Mock-Ups vielfach eingesetzt (vgl. Bødker et al. [18]). Eine Modellierung durch Skizzen auf Papier ist gegenüber einer bildschirmbasierten Modellierung nicht an vordefinierte Elemente gebunden, sondern ermöglicht ein einfaches Skizzieren informeller Elemente. Durch die Möglichkeit von Papier, Ideen schnell und einfach artikulieren zu können (vgl. Dai [35]) und mit mehreren Stiften parallel skizzieren zu können, ist Papier auch für eine kollaborative Modellierung gut geeignet. Die Modellierung auf großformatigem Papier ermöglicht es zudem auch umfangreichere Prozesse zusammenhängend darzustellen. Ein weiterer Vorteil einer Modellierung mit Stift und Papier ist der natürliche Umgang mit dem Eingabemedium Stift, so dass Nutzer weniger Aufmerksamkeit auf die Interaktion legen müssen und sich stattdessen auf die Modellierung konzentrieren können (vgl. Dai, Le und Oviatt et al. [35, 94, 134]). Auch besitzt das Medium Papier eine breite Akzeptanz bei den Nutzern (vgl. Sellen und Harper [152] sowie Wellner [168]). Durch neue Technologien, wie zum Beispiel Anoto [10], bieten PBUIs ein hohes Maß an Flexibilität und Endbenutzern einen leichten Zugang zur Technologie. Allgemein ermöglichen PBUIs die Verbindung der Vorteile des natürlichen Werkzeuges Stift mit den Verarbeitungsmöglichkeiten moderner Computer (vgl. [168]).

Bei der Betrachtung bestehender endbenutzerorientierter Modellierungsumgebungen ist zu erkennen, dass bei der Gestaltung dieser Systeme bisher wenig Augenmerk auf kollaborative und informelle Modellierungsaspekte gelegt wurde. Zudem zeigen verschiedene Untersuchungen zur Nutzung von PBUIs und zur Eingabe über Skizzen, dass diese, aufgrund von Flexibilitätsvorteilen, einer natürlichen Nutzung und einer guten Unterstützung kreativer Aspekte, signifikante Vorteile für kollaborative und kreative Nutzungsszenarien bieten können. Aufgrund des Fehlens kollaborativer und informeller (endbenutzergerechter) Modellierungssysteme einerseits und der beschriebenen Vorteile von PBUIs andererseits, erscheint eine Kombination dieser beiden Gebiete in einem papierbasierten System für die skizzenbasierte Modellierung durch Endbenutzer vielversprechend.

1.2 Ziele

Das primäre Ziel der Arbeit ist daher die Entwicklung, Umsetzung und Evaluation eines Konzeptes das durch die Verwendung papierbasierter Interfaces Endbenutzer, besonders in kollaborativen Szenarien, bei der Modellierung von Prozessen bzw. der Orchestrierung von Web Services (der verbreitetste Implementierungsform einer SOA) unterstützt.

Der Schwerpunkt der Arbeit liegt auf der Entwicklung eines geeigneten Interaktionskonzeptes für eine papierbasierte Modellierung. Dabei wird von einem weiter gefassten

Interaktionsbegriff ausgegangen. So spielt neben dem „eigentlichen“ *Interaktionskonzept* (z.B. die Gestaltung der Papierinterfaces und eines geeigneten Feedbacks) das *Modellierungskonzept* (z.B. die verwendete Notation), also die Interaktion der Nutzer mit den Prozessinformationen, eine wichtige Rolle. Zu berücksichtigen sind für das Konzept insbesondere die spezifischen Eigenschaften von Papierinterfaces, die Anforderungen an endbenutzergerechte Entwicklungssysteme und die Aspekte serviceorientierter Architekturen (als Oberkonzept der Web Service Orchestrierung). Da der Fokus der Arbeit auf der Gestaltung der papierbasierten Interaktion liegt, werden mögliche computerbasierte Weiterverarbeitungsaspekte nur am Rande betrachtet.

Um das System konzipieren zu können, ist ein Verständnis der derzeitigen Modellierungs- und Orchestrierungspraxis, sowie eine Analyse papierbasierter Gestaltungsansätze wichtig. Aufbauend auf diesen Erkenntnissen und auf Erfahrungen aus der Literatur sollen wesentliche Aspekte identifiziert und ein Konzept einer *papierbasierten, serviceorientierten Modellierung* für Endbenutzer (abgekürzt *PaSeMod*) erstellt werden. Abschließend soll durch die Analyse der Brauchbarkeit und der Benutzbarkeit untersucht werden, wie geeignet das entwickelte Konzept für die endbenutzerorientierte Modellierung ist. Für die Evaluation soll ein auf dem Konzept aufbauendes System erstellt werden.

1.3 Aufbau der Arbeit

Einen ersten Überblick über den Aufbau der Arbeit zeigt Abbildung 1. Im Anschluss werden die einzelnen Abschnitte kurz erläutert.

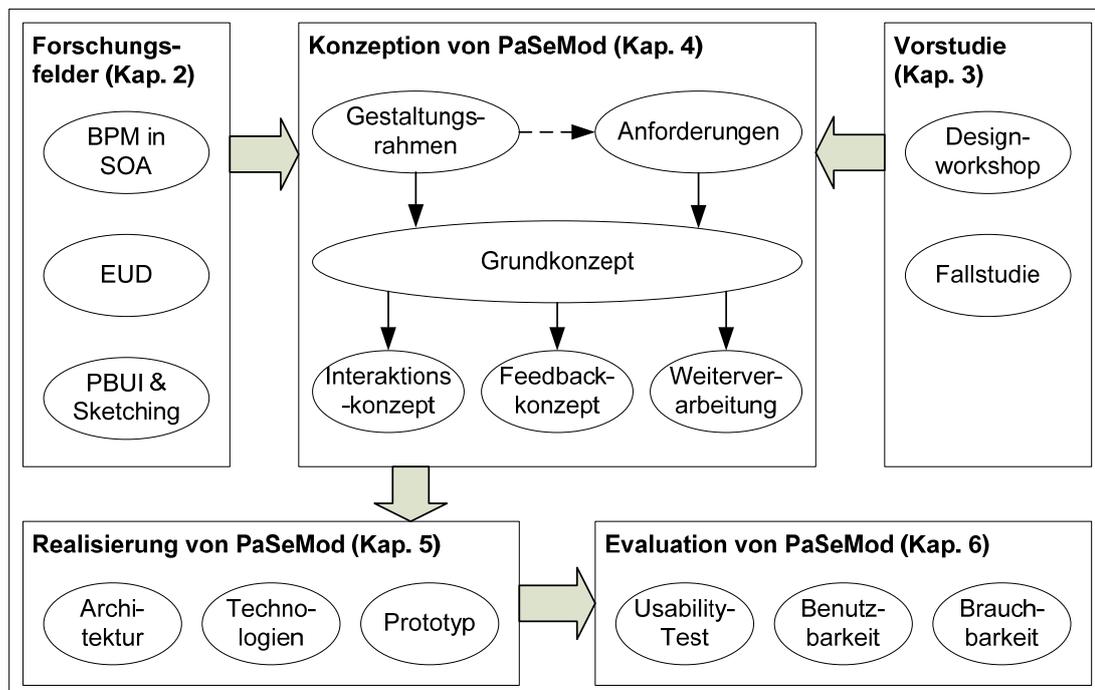


Abbildung 1: Mentales Modell der Arbeit

Die Arbeit ist primär in den drei Forschungsbereichen „Geschäftsprozessmanagement in serviceorientierten Architekturen“, „End-User Development“, sowie „papier- und skizzenbasierte Systeme“ angesiedelt. **Kapitel 2** führt in diese drei Forschungsfelder ein und beschreibt den Stand der Forschung. Im Bereich *BPM in SOA* wird neben der Einführung in diese beiden Begrifflichkeiten die serviceorientierte Modellierungspraxis vorgestellt. Im *EUD*-Teil werden allgemeine EUD-Gestaltungsansätze und bestehende Forschungsansätze im Bereich von EUD in SOA behandelt. Der Bereich der *papier- und skizzenbasierten Systeme* stellt die alternativen Eingabekonzepte über interaktives Papier und über handgezeichnete Skizzen vor. Im Abschluss des Kapitels werden die Ergebnisse zusammengefasst und der konkrete Forschungsbedarf aufgezeigt.

Kapitel 3 beschreibt die Planung, Durchführung und die Ergebnisse einer zweistufigen Voruntersuchung, die durchgeführt wurde, um einen Einblick in die Modellierungspraxis zu erhalten und um Designentscheidungen für das Konzept treffen zu können. Im ersten Abschnitt wird eine interviewbasierte *Fallstudie* in einem Unternehmen des Finanzsektors vorgestellt. Im Anschluss wird ein *Design-Workshop* mit Endbenutzern beschrieben, in dem Gestaltungsaspekte einer kollaborativen, papierbasierten Modellierung untersucht wurde.

Kapitel 4 bildet den Kern der vorliegenden Arbeit. Dort wird, aufbauend auf der Literatur und der Vorstudie, ein System zur papierbasierten, serviceorientierten Modellierung konzipiert. Dazu werden zu Beginn der *Gestaltungsrahmen* abgesteckt und *Anforderungen* an ein solches System identifiziert. Aufbauend auf diesen Punkten wird das *PaSeMod-Grundkonzept* entwickelt. Anschließend werden das *papierbasierte Interaktionskonzept* und das *Feedbackkonzept* herausgearbeitet, sowie *Weiterverarbeitungsaspekte* für die erstellten Prozessmodelle diskutiert. Abschließend wird das Gesamtkonzept zusammengefasst.

In **Kapitel 5** wird die Realisierung des PaSeMod-Konzeptes beschrieben. Zuerst werden die *grundlegende Architektur* des Systems und die *verwendeten Technologien* vorgestellt. Im Anschluss wird die *prototypische Umsetzung* von PaSeMod beschrieben. Neben konkreten Realisierungsaspekten werden hier auch notwendige Einschränkungen und Abweichungen vom Konzept diskutiert.

Den Abschluss der Arbeit bildet eine kurze Evaluation von PaSeMod. Diese liefert ein erstes Feedback zu dem entwickelten PaSeMod-Konzept bzw. dessen Realisierung. In **Kapitel 6** wird ein mit zwei Probanden aus der Praxis durchgeführter *Usability-Test* beschrieben. Anhand der Ergebnisse werden erste Erkenntnisse für die *Benutzbarkeit* und die *Brauchbarkeit* von PaSeMod vorgestellt und diskutiert.

Am Ende erfolgt eine Zusammenfassung der Arbeit und es wird ein Ausblick auf offene Forschungs- und Realisierungsfragen gegeben.

2 Stand der Forschung

In diesem Kapitel werden die drei zentralen Forschungsbereiche der Arbeit vorgestellt. Kapitel 2.1 beschreibt, aufbauend auf den Grundlagen des Geschäftsprozessmanagements und der serviceorientierten Architekturen, aktuelle Ansätze und Herangehensweisen einer serviceorientierten Prozessmodellierung. Anschließend gibt Kapitel 2.2 einen Überblick über den Forschungsbereich des End-User Developments. Dazu werden relevante Ansätze für die Gestaltung endbenutzerorientierter Systeme diskutiert und der bestehende Forschungsstand an der Schnittstelle des End-User Developments und der serviceorientierten Architekturen beschrieben. Im Kapitel 2.3 werden Grundlagen, Gestaltungsaspekte und bestehende Ansätze papier- und skizzenbasierter Anwendungen vorgestellt. Abschließend werden die drei Bereiche in Kapitel 2.4 zusammengeführt und der daraus resultierende Forschungsbedarf beschrieben.

2.1 Prozessmodellierung in serviceorientierten Architekturen

Durch Einflussgrößen wie steigende Produktkomplexität, zunehmende Marktdynamik oder die Individualisierung von Produkten und Lösungen, müssen Unternehmen in immer stärkerem Maße flexibel auf sich ändernde Anforderungen und Umweltbedingungen reagieren können (vgl. Scheer [149] und Wulf [172]). Nach Goranson [51] steht diese nötige Flexibilität und Agilität zumindest gleichrangig neben bestehenden Rationalisierungs- und Automatisierungsbestrebungen. Um beiden Anforderungen gerecht zu werden, benötigen Unternehmen dynamische Lösungen zur Anpassung ihrer Prozessabläufe und Systeme [149]. In diesem Zusammenhang kommt besonders der Verbindung der Konzepte des *Geschäftsprozessmanagements* und der *serviceorientierten Architekturen* eine große Bedeutung zu.

2.1.1 Grundlagen des Geschäftsprozessmanagements

BPM ist ein allgemeines Managementthema, das auf strategischer und operativer Ebene die Prozessorientierung des gesamten Unternehmens sowie einzelner Geschäftsbereiche fokussiert. Eine mögliche Definition von Gaitanides et al. [49] sieht das Prozessmanagement als Menge:

„[...] planerischer, organisatorischer und kontrollierender Maßnahmen zur zielorientierten Steuerung der Wertschöpfungskette eines Unternehmens hinsichtlich Qualität, Zeit, Kosten und Kundenzufriedenheit.“ (Seite 3)

Verschiedene Untersuchungen dokumentieren sowohl auf geschäftlicher als auch auf technischer Ebene den hohen Stellenwert von BPM für die Unternehmensplanung und trotz

oder gerade wegen anhaltendem Kostendruck auch ein starkes Interesse an Investitionen in diesem Bereich (siehe Gadatsch [48] sowie Schmelzer und Sesselmann [151]).

Kernpunkt von BPM ist die Betrachtung von Geschäftsprozessen in Unternehmen. Trotz einer fehlenden einheitlichen Definition dieser (vgl. [48]), kann allgemein festgehalten werden, dass ein Geschäftsprozess eine zeitlich-logische Abfolge von Aktivitäten darstellt, die arbeitsteilig von ein oder mehreren Personen (und zum Teil auch automatisiert) zielgerichtet durchgeführt werden können, um gemäß den Unternehmenszielen verschiedene Leistungen zu erstellen.

Die Ausrichtung von BPM auf die Unternehmensprozesse ist kein neues Phänomen, sondern folgt dem *Total Quality Management*-Ansatz aus den 80er Jahren und dem Anfang der 90er Jahre weit verbreiteten Paradigma des *Business Process Reengineerings (BPR)*. Gegenüber dem Ansatz des häufig gescheiterten BPR (siehe z.B. Krafzig et al. [88]) zielt BPM jedoch nicht mehr auf eine radikale Neuausrichtung der Prozesse und Abläufe, sondern vielmehr auf eine inkrementelle Veränderung sowie eine evolutionäre Prozessoptimierung, ab.

Durch den starken Prozessfokus spielt die Informationstechnologie (IT) eine bedeutende Rolle für die operative Umsetzung der Prozessabläufe, sowie für die Unterstützung der organisatorischen Gestaltung des BPM. Ziel der IT im Bereich von BPM ist die Erstellung, Veränderung, Neukonfiguration und Optimierung von Prozessdefinitionen mit grafischen Werkzeugen, die auch von technisch weniger begabten Geschäftsanalysten benutzt werden können [88]. Im Gegensatz zur Geschäftssicht liegt der Fokus der IT-Sicht auf der fachlich-konzeptionellen und auf der operativen Ebene [48]. Dabei stehen Begriffe, wie Workflow-Management, Prozessmodellierung und Web Services im Vordergrund. In der Arbeit wird diese IT-orientierte Sicht von BPM zu Grunde gelegt, die van der Aalst et al. [166] wie folgt definieren:

„Supporting business processes using methods, techniques, and software to design, enact, control, and analyze operational processes involving humans, organizations, applications, documents and other sources of information.“ (Seite 1)

Praktisch wird BPM in der IT häufig als „next step“ nach der Workflow-Bewegung der 90er Jahre gesehen (vgl. [166]) und baut auf den Konzepten des *Workflow-Managements (WFM)* auf. Oft wird der Begriff BPM mittlerweile auch an Stelle von WFM verwendet (z.B in [23]) und viele Workflow-Hersteller positionieren ihre WFM-Systeme als BPM-Systeme [166]. In Abgrenzung zu WFM hat BPM jedoch den Anspruch eines gemeinsamen Sprachgebrauchs mit den Fachabteilungen (vgl. Grille [55]) und unterstützt, im Gegensatz zu WMF, nicht nur die einmalige Erstellung, Konfiguration und Ausführung von Prozessen, sondern einen Gesamtlebenszyklus aus Diagnose, Prozessdesign, Systemkonfiguration und Prozessdurchführung (siehe Abbildung 2). Ein Kernaspekt bleibt jedoch die Ausführung von

Prozessabläufen als Workflows (sowohl menschlich als auch automatisiert). Typischerweise werden diese Prozesse in BPM-Systemen als zeitlich-logische Abfolge von Teilaktivitäten grafisch modelliert und blockorientiert gespeichert, so dass sie durch ein Prozesssystem dynamisch ausgeführt werden können.

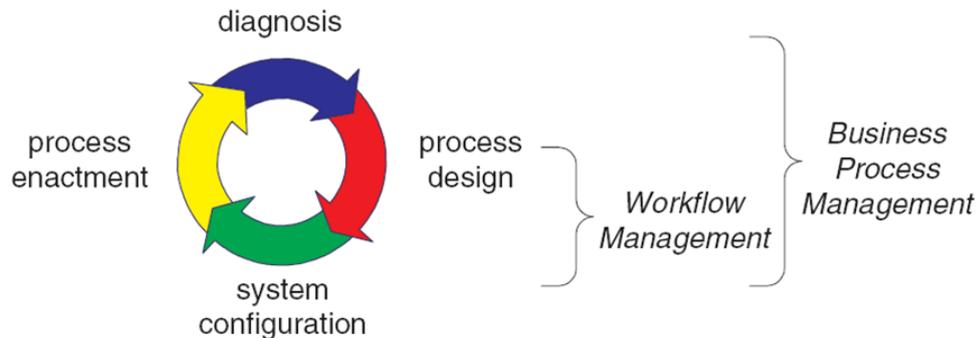


Abbildung 2: BPM und WFM Lebenszyklus [166]

2.1.2 Grundlagen serviceorientierter Architekturen

Um die von BPM geforderte Flexibilität der Unternehmensprozesse auf IT-Ebene zu gewährleisten, ist eine Systemarchitektur nötig, die eine einfache, flexible und dynamische Zusammenstellung, Anpassung und Ausführung von Anwendungen und Ressourcen ermöglicht. SOA bietet ein Konzept einer abstrakten Systemarchitektur,

„[...] die vielfältige, verschiedene und eventuell inkompatible Methoden oder Applikationen als wieder verwendbare und offen zugreifbare Dienste repräsentiert und dadurch eine plattform- und sprachenunabhängige Nutzung und Wiederverwendung ermöglicht.“ (Melzer [112], Seite 11)

Durch die Möglichkeit einer losen Kopplung und Bindung dieser Dienste zur Laufzeit stellt SOA eine geeignete technische Basis für aktuelle BPM-Bestrebungen dar. Eines der Hauptziele von SOA sind zudem Dienste zu denen eine direkte Entsprechung von Funktionen auf Geschäftsseite existiert. Durch diese Entsprechung von Technik- und Geschäftsebene besitzt SOA den Anspruch, als direkte Schnittstelle und „Diskussionsgrundlage“ zu fungieren und für beide Seiten eine dauerhafte Wertschöpfung zu bieten (vgl. Krafzig et al. [88]).

SOA ist aber kein gänzlich neues Konzept, sondern stellt eine konsequente Weiterentwicklung bestehender Ansätze der komponentenbasierten Entwicklung und verteilter Internet-Architekturen, insbesondere der Ansätze *RPC (Remote Procedure Calls)* und *CORBA (Common Object Request Broker Architecture)*, dar (siehe [88]). Kernaspekte von SOA sind die lose Kopplung von Diensten, die Kapselung von Diensten, verteilte Dienste und ein Punkt, an dem frühere serviceorientierte Ansätze wie CORBA unter anderem gescheitert sind: die Verwendung offener und industrieweit akzeptierter Standards, allen voran XML (Extensible Markup Language) als Basis für den Austausch von Informationen

und Daten (siehe Kloppmann et al. [83]). Anders als zum Teil fälschlicherweise angenommen, ist SOA selbst jedoch keine konkrete Technik, sondern eine abstrakte Software-Architektur. Die aktuell am weitesten fortgeschrittene und verbreitete Umsetzung stellt die *Web Service (WS) Technologie* dar.

Bestandteile einer SOA

Im Mittelpunkt einer SOA stehen die Services und ihre verteilte Nutzung. Dabei wird mit dem Begriff Service in der Regel ein *Business Service* assoziiert, der die Ausführung bestimmter Geschäftsaktivitäten ermöglicht. Allgemein sind Services als grobkörnige Komponenten zu verstehen, welche die Ausführungsdetails „verstecken“, indem sie die Geschäftskonzepte auf einer höheren Ebene kapseln, und nach außen für Nutzer eine Art „Blackbox“ darstellen. Diese Kapselung erfolgt auf mehreren Ebenen, so dass sich im SOA-Umfeld vier Service-Typen differenzieren lassen: *Basis-Services*, *Vermittlungs-Services*, *Prozesszentrierte Services* und öffentliche *Unternehmens-Services*. Besonders hervorzuheben sind Basis- und Prozesszentrierte Services. Basis-Services bilden das Fundament einer jeden SOA, indem sie kleine Kernkomponenten abbilden. Sie besitzen eine geringe Änderungshäufigkeit und eine hohe Wiederverwendbarkeit. Prozesszentrierte Services dagegen kapseln die Prozesslogik und damit das Wissen über Geschäftsprozesse in Unternehmen, sowie die vorhandene Prozesskomplexität. Dementsprechend unterliegen sie häufigen Änderungen und besitzen eine geringe Wiederverwendbarkeit. Sie sind im Gegensatz zu Basis-Services zwar kein zwingender Bestandteil einer SOA, stellen aber die fortgeschrittenste Form einer SOA dar (vgl. [88]).

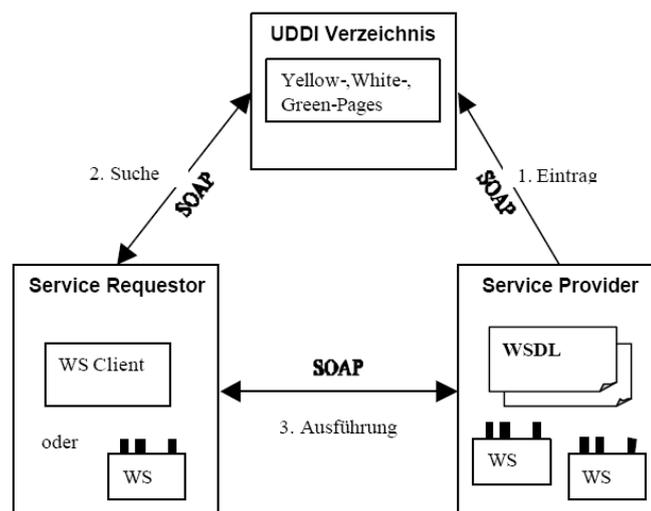


Abbildung 3: Serviceorientierte Architekturen am Beispiel von Web Services („SOA-Dreieck“) [143]

Bei der Service-Nutzung lassen sich drei generelle Rollen und damit verbundene Hauptfunktionen einer SOA unterscheiden (siehe Abbildung 3): Ein *Service Provider* stellt

Services zur Verfügung, die über Schnittstellenbeschreibungen, im Fall von Web Services im *WSDL*-Format, dokumentiert werden. Verfügbare Services werden auf Basis der Schnittstellenbeschreibung in einem zentralen Verzeichnis, dem *Service Repository* (z.B. *UDDI* im Web Service Umfeld), eingetragen. Ein *Service Requestor* kann dort anhand von spezifischen Beschreibungen einzelne Services suchen. Er erhält einen Verweis auf den entsprechenden Service beim Provider und kann diesen dann direkt ansprechen. Der gesamte Nachrichtenaustausch erfolgt über einen einheitlichen Kommunikationsstandard auf Basis von XML. Bei Web Services wird meist das *SOAP*-Protokoll eingesetzt.

Weitere wichtige Bestandteile für den Einsatz einer SOA sind, neben den Services und den Service Repositories, spezifische *Anwendungs-Frontends* und der *Enterprise Service-Bus*. Die Anwendungs-Frontends (WS oder WS-Client auf Seiten des Service Requestors in Abbildung 3) initiieren die Prozesse und verarbeiten die Ergebnisse. Sie bildet die aktive Komponente einer SOA, die als Schnittstelle zwischen den Nutzern und den Prozessen den eigentlichen Nutzen schafft. Der Enterprise Service-Bus [30] ist die Middleware-Infrastruktur einer SOA und sorgt hauptsächlich für die Kommunikation der einzelnen Services (und der Anwendungs-Frontends) untereinander. Er stellt über die Unterstützung heterogener Techniken die hohe Kompatibilität von SOA sicher.

Kompositionen von Services

Über das vorgestellte SOA-Dreieck (Abbildung 3) lassen sich verteilte Services finden und ausführen. SOA ist jedoch mehr, als nur eine Menge von Services, die miteinander interagieren können. Um SOA als technische Basis für BPM zu nutzen, werden Methoden benötigt, die eine Komposition mehrerer Services zu neuen Anwendungen (z.B. einem neuen Geschäftsprozess) ermöglichen. Diese Zusammenstellung von vorhandenen Komponenten wird als *Programmierung im Großen* bezeichnet, in Abgrenzung zur typischen Implementierung von Kernfunktionalitäten (z.B. von Basis-Services) auf technischer Ebene (*Programmierung im Kleinen*). Die eigentliche Leistung bei der Entwicklung der Systeme ist damit nicht mehr die Programmierung, sondern die Modellierung der Geschäftsabläufe und Prozesse (Melzer [112]). In Bezug auf die verschiedenen Service-Klassen, findet sich die Kerngeschäftslogik in den Basis-Services, die Dialogsteuerung in den Anwendungs-Frontends und die Prozesslogik in den Prozesszentrierten Services. Komponierte Services können somit selbst wieder als Service verfügbar gemacht werden.

Wenn über die Komposition von Web Services gesprochen wird, fallen häufig die Begriffe *Choreographie* (auch Koordination) und *Orchestrierung*. Die Choreographie befasst sich mit der Art und Weise, wie verschiedene Partner (Prozesse oder Benutzer) miteinander interagieren und beschreibt die Kommunikationen zwischen einzelnen Partnern durch gültige Nachrichtenabfolgen. Die Orchestrierung dagegen beschreibt einen Geschäftsprozess durch

eine Abfolge verschiedener Service-Aufrufe und -Instantiierungen (inklusive des Nachrichtenaustausches) und an die Aufrufe geknüpfte Ausführungsbedingungen (Reichert und Stoll [143]). In Abgrenzung zur allgemeinen Prozessmodellierung, die auch semi-formale Aspekte unterstützt (Thomas et al. [164]), werden bei der Orchestrierung konkrete und existierende Aktivitäten so zusammengestellt, dass diese direkt ausgeführt werden können. Formal gesehen ist das Ergebnis einer Orchestrierung ebenso ein Prozessmodell, welches lediglich statt diagrammbasiert, skriptbasiert beschrieben wird und so eine präzisere Modellspezifikation ermöglicht (Gadatsch [48]). Für die Orchestrierung von Web Services haben sich eine Vielzahl an Prozessdefinitionssprachen, wie zum Beispiel BPEL, BPML, WFML und WS-CDL, herausgebildet. In den letzten Jahren fand eine Fokussierung statt, so dass sich die *Business Process Execution Language (BPEL)* [11] als de facto Standard für die Orchestrierung von Geschäftsprozessen auf Basis von Web Services herausgebildet hat (Ouyang et al. [132]). BPEL ist, wie viele andere Prozessspezifikationsprachen auch, eine XML-basierte Erweiterung imperativer Programmiersprachen mit speziellen Aspekten für die Web Service Programmierung. BPEL zeichnet sich durch eine blockorientierte Struktur, eine strikte Syntax und durch das Nichtvorhandensein einer spezifizierten grafischen Notation aus.

Ursprünglich unterstützte BPEL ausschließlich die Computer-zu-Computer Interaktion. Daher konnten mit BPEL lange Zeit nur automatische Prozesse, in die keine Nutzer involviert sind, orchestriert werden. Auch die 2007 veröffentlichte BPEL 2.0 Spezifikation enthält keine Möglichkeit zur Beschreibung von Prozessschritten, die durch eine Benutzerinteraktion (z.B. über eine GUI) ausgeführt werden können. Da Benutzerinteraktionen jedoch einen elementaren Aspekt von Geschäftsprozessen darstellen (Kloppmann et al. [84]), besitzen viele BPM-Systeme individuelle Erweiterungen, die eine Ausführung von Aufgaben durch Personen ergänzen (siehe Brahe [21]). Der endgültige Schulterschluss der SOA-Gemeinde mit der Workflow-Fraktion erscheint möglich durch die Erweiterung von BPEL durch die BPEL4People-Spezifikation [3] und die ergänzende Spezifikation WS-HumanTask [2], die gemeinsam die Implementierung menschlicher Aktivitäten ermöglichen. Beide Spezifikationen wurden bisher jedoch noch nicht als offizieller Standard anerkannt [1].

BPM und SOA in der Praxis

Auf Seite 8 wurde bereits darauf eingegangen, warum ein starkes Interesse an der Verbindung von BPM und SOA existiert und warum diese Verbindung sinnvoll erscheint. Besonders sei noch einmal die gemeinsame Betrachtungsebene der Geschäftsprozesse auf IT- und Geschäftsseite, sowie die Möglichkeit zur flexiblen und dynamischen Zusammenstellung von Geschäftsprozessen hervorgehoben. Technisch gesehen stellt SOA die Back End-Funktionalität für die Implementierung der Prozessfunktionalität eines BPM-Systems bereit, welches auf die Prozess-Schicht der SOA zugreift. So sind Geschäftsprozessabläufe als eine

Abfolge aus Service-Aufrufen und menschlichen Aktivitäten (über User Interfaces) zu sehen (siehe Abbildung 4). Trotz großem Interesse in der Verbindung dieser beiden Konzepte, sowohl in der Wissenschaft als auch in der Praxis (siehe [23, 165]), existieren bisher kaum dokumentierte, empirische Forschungsergebnisse zur Realisierung von BPM in SOA in der Praxis.²

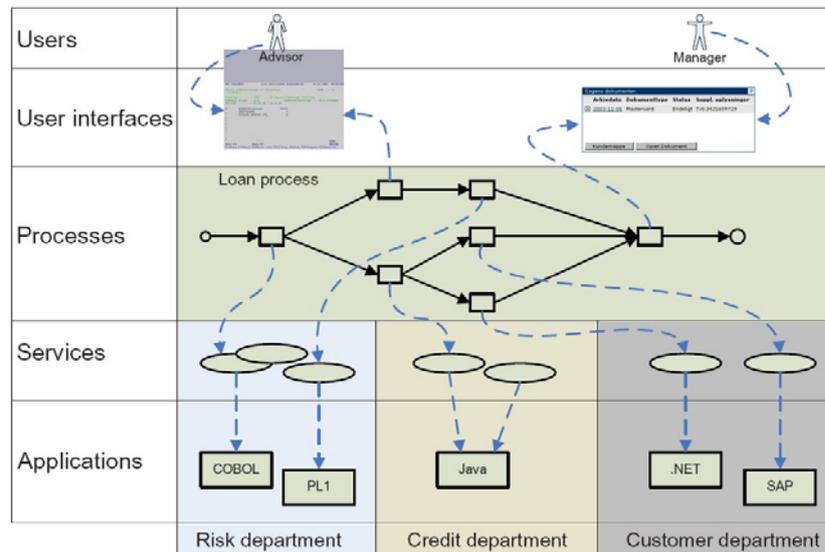


Abbildung 4: Geschäftsprozesse als Komposition von Services und menschlichen Aktivitäten [23]

Ein gutes Beispiel ist jedoch die Studie von Brahe und Schmidt [23]. Sie untersuchen die Implementierung von BPM in einer SOA-Umgebung im Finanzsektor. In einer Analysephase definiert ein *Business Developer* gemeinsam mit Nutzern aus den Fachabteilungen auf einer abstrakten Ebene die Anforderungen an einen Prozess. Im nächsten Schritt erstellt ein *Solution Architect* ein so genanntes *Solution Model*, in dem alle Aufgaben des vorher definierten Prozesses beschrieben werden. Dieses Modell wird anschließend von einem *Process Developer* verwendet, um den Prozess zu implementieren. Das bei Brahe und Schmidt beobachtete Vorgehensmodell ähnelt dem idealtypischen Ablauf einer integrierten Prozesserstellung in SOA, wie ihn Thomas et al. [164] beschreiben. Nach Thomas et al. erfolgt zu Beginn der Designphase die Modellierung des Informationsmodells der Geschäftsprozesse. Wie zum Beispiel in ARIS (Architektur integrierter Informationssysteme) [148], wird dabei im Sinne einer ganzheitlichen Prozessmodellierung meist nicht ausschließlich der Prozessablauf, sondern auch Dimensionen wie Funktionen, Daten und die Benutzerschnittstelle beschrieben. Die Prozessbeschreibung erfolgt entweder textuell oder anhand einer grafischen (Prozess-)Modellierungssprache. Abhängig von der verwendeten Prozessbeschreibung muss gegebenenfalls im nächsten Schritt aus der Prozessbeschreibung

² Eine Vielzahl an Untersuchungen findet sich jedoch zur technischen Einführung einer SOA wie auch zu wirtschaftlich-strategischen Aspekten von BPM.

ein Anwendungsmodell erstellt werden, in dem ausschließlich Informationen der Konfiguration des Prozesssystems abgebildet sind. Um durch ein BPM-System ausführbar zu sein, muss das grafenorientierte Prozessmodell abschließend in eine blockorientierte Prozessdefinitionssprache überführt werden. Bei der Web Service Technologie erfolgt die Modellierung und die Konfiguration meist über die grafische *Business Process Modeling Notation (BPMN)* [128], die für die Ausführung dann automatisch oder manuell in die Prozessdefinitionssprache BPEL transformiert wird. Die Trennung von Prozessmodellierung und Prozessspezifikation ist nötig, da die Entwicklung von Prozessen auf Technischebene für die Anwender in Fachabteilungen nicht geeignet ist. Diese sind es eher gewohnt ihre Vorstellung vom Verhalten neuer Anwendung über Prozessmodelle zu beschreiben, die in Werkzeugen wie Microsoft Visio oder IDS Scheer ARIS erstellt werden (Grille [55]). Die Verwendung verschiedener Entwurfsebenen und damit verschiedener Notationen und Programmiersprachen verlangt jedoch nach einer geeigneten Zuordnung der verschiedenen Modellebenen (Thomas et al. [164]), was häufig durch eine nicht vollständige automatische Transformation und einen fehlenden oder unzureichenden Abgleich der Modelle eingeschränkt ist (vgl. Borggräfe [19] und Brahe [21]). Um diese Probleme zu vermeiden, sind in verschiedenen Forschungsansätzen Werkzeuge entstanden, die auch Fachanwendern die direkte Orchestrierung kleiner Prozesse ermöglichen sollen (siehe Kapitel 2.2.3).

2.1.3 Serviceorientierte Prozessmodellierung

“Modeling is a complex activity of abstracting information and knowledge from a particular domain in order to achieve a model containing the essentials from the perspective of the modelers and their given goals.” (Kaindl und Carroll [80], Seite 28)

Geschäftsprozessmodelle bilden einen relevanten Ausschnitt aktueller oder zukünftiger Realität ab und bilden die Grundlage für die Analyse, Optimierung, Ausführung und Kontrolle der Prozesse. Dementsprechend existiert eine Vielzahl an Modellierungssprachen und Werkzeugen, welche die Modellierung für verschiedene Nutzergruppen und Kontexte unterstützen. Wie bereits dargestellt, werden Prozesse in der Regel nicht direkt in BPEL zusammengestellt, sondern erst grafisch modelliert und die Modelle anschließend nach BPEL überführt. Aufgrund der fehlenden grafischen Notation von BPEL wurden auch verschiedene proprietäre grafische Erweiterungen und Werkzeuge entwickelt, die eine grafische Prozessmodellierung „direkt“ in BPEL ermöglichen. Im Gegensatz zu den typischerweise grafenbasierten Modellierungsnotationen besitzen BPEL-Visualisierungen aufgrund der engen Bindung an BPEL jedoch eine Blockstruktur und bieten daher weniger flexible Modellierungsmöglichkeiten. Eine weitergehende Diskussion von Modellierungsaspekten für

die Softwareneinführung und -anpassung mit besonderem Fokus auf sozio-technische Aspekte findet sich bei Loser [106].

Serviceorientierte Modellierungswerkzeuge

Nach Dörner et al. [41] lassen sich bestehende Modellierungssysteme in die zwei Kategorien „Professionelle Werkzeuge“ und „Web 2.0 Werkzeuge“ unterteilen. Bekannte Beispiele professioneller Modellierungswerkzeuge sind *WebSphere Business Modeler*, *IDS Scheer ARIS Business Architect*, *Oracle BPEL Process Manager* und *SAP NetWeaver Composition Environment*. Diese Systeme sind meist eingebettet in eine ganzheitliche BPM-Lösung mit Schnittstellen zu anderen BPM-Bereichen (z.B. Design, Ausführung und Analyse). Dementsprechend besitzen sie eine hohe Funktionalität, vielfältige Verbindungen mit anderen Komponenten, aber auch eine hohe Komplexität. Zielgruppe für diese Systeme sind daher Business Analysten und Entwickler mit fortgeschrittenem Modellierungswissen. Web 2.0-basierte Werkzeuge ermöglichen die Zusammenstellung verschiedener Web-Inhalte als Prozessablauf in sogenannten Mashups. Beispiele für Mashup-Systeme sind *Yahoo! Pipes*, *MS Popfly* und das endbenutzerorientierte System *MARMITE* [171]. Eine Sonderrolle nimmt das Werkzeug *Serena Business Mashups* [153] ein, welches das Mashup-Konzept auf die Erstellung SOA-basierter Prozesse überträgt und so auch technisch erfahrenen Fachanwendern die Modellierung ausführbarer Prozesse ermöglichen soll. Neben diesen beiden Gruppen spielen für die wissenschaftliche Betrachtung zudem zunehmend skizzenbasierte Modellierungssysteme eine Rolle. Auf diese Systeme wird in Kapitel 2.3.5 detaillierter eingegangen.

Serviceorientierte Modellierungssprachen

Prozessmodelle werden typischerweise auf Basis eines oder mehrerer Diagramme beschrieben. Ein Diagramm besteht aus einer Zusammenstellung verschiedener, meist grafischer und textueller Notationselemente, deren Syntax und Semantik informell beschrieben oder formal spezifiziert ist. Grundlegend lassen sich diagrammbasierte Modellierungsnotationen nach Gadatsch [48] in datenflussorientierte, in objektorientierte und in kontrollflussorientierte Ansätze unterteilen. Datenflussorientierte Notationen bilden die Verarbeitung und Verknüpfung von Eingabe- und Ergebnisdaten verschiedener Teilaufgaben ab und werden in der Modellierungspraxis immer seltener eingesetzt (vgl. [48]). Ein typisches Anwendungsfeld datenflussorientierter Notationen sind Mashup-Systeme, bei denen die Verarbeitung der Daten von verschiedenen Webseiten im Vordergrund steht. Im Bereich der Service-Orchestrierung findet sich ein datenflussorientierter Ansatz in der Notation des endbenutzerorientierten Orchestrierungssystems *SiSeOr* [135]. Objektorientierte Ansätze werden vor allem für grobe Prozessbeschreibungen auf hohem Abstraktionsniveau (z.B. über Aktivitätsdiagramme), sowie für Beschreibungen von Unternehmenszusammenhängen (z.B.

über Use-Case Diagramme) verwendet. Die größte Verbreitung für die Modellierung von Geschäftsprozessen besitzen kontrollflussorientierte Sprachen, die den Prozess als zeitlich-logischen Ablauf von Aktivitäten und Bedingungen darstellen und so nah an der tatsächlichen Prozessausführung sind. Einige Sprachen unterstützen zudem die Modellierung sowohl kontrollfluss-, als auch datenflussorientierter Aspekte innerhalb einer oder verschiedener Sichten (z.B. bei Pautasso und Alonso [138]). Im Folgenden werden die in der Prozessmodellierung weit verbreiteten Notationen *BPMN*, *eEPK* (*erweiterte Ereignisgesteuerte Prozessketten*) und *UML* (*Unified Modeling Language*) kurz vorgestellt.

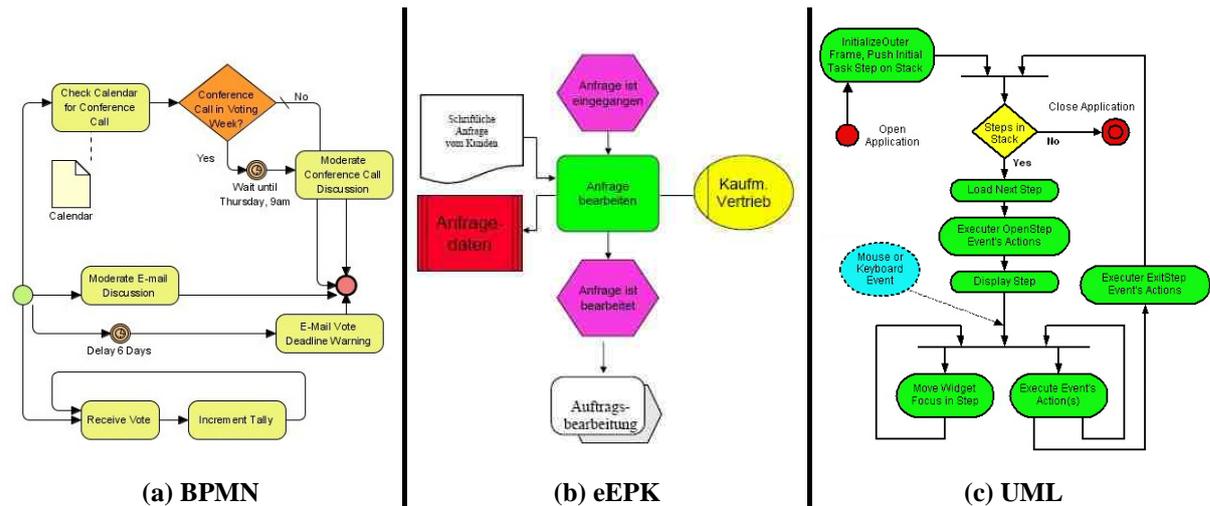


Abbildung 5: Beispiele für die Modellierungsnotationen (a) BPMN, (b) eEPK und (c) UML (Aktivitätsdiagramm)

BPMN [128] (Abbildung 5a) wurde ursprünglich von der *Business Process Management Initiative* entwickelt und besitzt den Anspruch einer einfach verständlichen, flexiblen und erweiterbaren Notation, die nicht auf eine spezifische Nutzergruppe fokussiert ist. Stattdessen soll sie eine Schnittstelle zwischen IT- und Geschäftsorganisation bilden, um den Umfang und die Funktionalität von Prozessen und Anwendungen zu diskutieren (vgl. Krafzig et al [88]). BPMN teilt die Notationselemente in vier Kategorien ein, die abhängig von der Modellierungskomplexität verwendet werden können. So ist eine einfache Modellierung bereits mit wenigen Elementen möglich. Um die Lücke zwischen Prozessmodell und Prozessspezifikation zu schließen, beinhaltet die BPMN-Spezifikation bereits Vorschläge, die eine direkte Transformation von BPMN-Modellen nach BPEL ermöglichen sollen. Oft wird BPMN daher auch als grafische Erweiterung von BPEL gesehen. Vor allem die konzeptionelle Nähe von BPMN zur Prozessspezifikation hat dazu geführt, dass BPMN eine große Verbreitung bei der Analyse von Geschäftsprozessen gefunden hat (vgl. Ouyang et al. [131]) und von vielen BPM-Softwareanbietern unterstützt wird.³

³ Eine Auflistung der Systeme, die BPM einsetzen findet sich unter http://www.bpmn.org/BPMN_Supporters.htm.

Die eEPK-Notation [96] (Abbildung 5b) wurde Anfang der 90er Jahre als eine der ersten Notationen zur Darstellung des zeitlich-logischen Ablaufes von Funktionen auf Geschäftsseite entwickelt. Die Modelle sind oft eingebettet in eine ganzheitliche Modellierung nach dem ARIS-Konzept und haben unter anderem durch die zentrale Integration in SAP R/3 eine große Verbreitung in der Wirtschaftspraxis gefunden (Gadatsch [48]). Das Grundkonzept von eEPK sieht einen Geschäftsprozess als eine wechselseitige Abfolge aus Ereignissen und Funktionen, verbunden durch gerichtete Kanten und logische Operatoren. Ebenso wie bei BPMN ist eine Modellierung auf unterschiedlichen Komplexitätsstufen möglich.

UML (Abbildung 5c) ist ein weit verbreiteter Ansatz für die objektorientierte Entwicklung von Informationssystemen. Die UML setzt sich aus einer Vielzahl verschiedener, zum Teil eng verwandter, Diagrammarten für verschiedene Aufgabengebiete zusammen. Mit dem Argument, dieselben Konzepte sowohl für die Entwicklung betrieblicher Systeme, als auch für die Modellierung geschäftlicher Aspekte einzusetzen, um so eine bessere Kommunikation zwischen IT- und Geschäftsseite sowie eine höhere Qualität der Spezifikation zu erreichen, gibt es Bestrebungen UML auch für die Geschäftsprozessmodellierung zu verwenden (Brücher und Endl [26]). Die UML 2.0 Spezifikation [129] erweitert die Notation (insbesondere das Aktivitätsdiagramm) daher um entsprechende Konzepte zur Prozessmodellierung. Da UML jedoch für die Entwicklung von Softwaresystemen konzipiert wurde, fehlen ihr geschäftsnahe Konstrukte zur Beschreibung und Detaillierung von Geschäftsprozessen (vgl. Gadatsch [48]). Zudem zeichnet sich UML durch einen großen Sprachumfang aus und erfordert somit einen hohen Aufwand zum Erlernen der Notation. Die Grundelemente ähneln in ihrer Gestaltung den Elementen von BPMN und eEPK.

Neben diesen drei weit verbreiteten Modellierungsmethoden haben einzelne BPM-Hersteller verschiedene proprietäre Erweiterungen und Adaptionen vorhandener Notationen erstellt. Ein Beispiel hierfür ist die *Solution Process Flow*-Notation [85], die im *Business Modeler* von IBM verwendet wird und aus eine Mischung von BPMN, UML und eEPK besteht.

2.2 Softwareanpassung durch Endbenutzer

Im vorangegangenen Kapitel wurde die Bedeutung flexibler Anpassungsmöglichkeiten für Unternehmen beschrieben und aufgezeigt, wie serviceorientierte Architekturen diese Flexibilität auf technischer Ebene unterstützen können. Auf organisatorischer Ebene soll die flexible Anpassung von Systemen durch die Einbeziehung der Anwender in die Entwicklung des Systems, zum Beispiel durch den Einsatz von BPMN, erreicht werden. Wie Brahe und Schmidt [23] zeigen, gestaltet sich diese Einbeziehung in der Praxis jedoch oft schwierig. Nach Lieberman et al. [102] kann Flexibilität zudem nur erreicht werden, wenn Anwender befähigt werden ihr System kontinuierlich selbstständig an ihre Bedürfnisse anzupassen. Dies vermeidet einen hohen Kommunikationsaufwand zwischen Anwendern und Entwicklern (und

daraus eventuell entstehende Kommunikationslücken) und steigert somit zusätzlich die Effektivität (Wulf und Jarke [173]). Um eine Anpassbarkeit durch Endbenutzer zu erzielen, werden unter dem Begriff *End-User Development (EUD)* Konzepte betrachtet, die nicht mehr nur einfach benutzbare, sondern auch einfach zu entwickelnde Systeme ermöglichen [102]. Zu Beginn dieses Kapitels werden die Grundlagen und die Entwicklung von EUD beschrieben. Im Anschluss werden relevante EUD-Ansätze und -Konzepte aufgezeigt und abschließend die bisherige Forschung zu EUD in SOA betrachtet.

2.2.1 Grundlagen des End-User Development

Ein Kernaspekt von EUD ist es Endbenutzer zu befähigen, ihre Systeme aktiv zu kontrollieren und sie ohne Hilfe professioneller Entwickler oder IT-Abteilungen zu entwickeln. Nach Lieberman et al. [102] definiert sich EUD als:

„[...] a set of methods, techniques, and tools that allow users of software systems, who are acting as non-professional software developers, at some point to create, modify, or extend a software artifact.“ (Seite 2)

Endbenutzer zur Anpassung ihrer Systeme zu befähigen wird umso wichtiger, als dass Entwickler nicht länger alle Anforderungen von Endbenutzern antizipieren und mit typischen Anpassungsprozessen nicht flexibel genug auf dynamische Veränderungen reagieren können (Repenning und Ioannidou [144]). Da Endbenutzer in der Regel jedoch keine Erfahrungen mit der Anpassung von Systemen haben [102] und meist nur eine begrenzte Zeit und Motivation besitzen, um die nötigen Methoden für die Anpassung der Systeme zu erlernen (Nardi [121]), sind professionelle Entwicklungsmethoden meist zu komplex und können nicht direkt auf Endbenutzer übertragen werden. Für EUD-Systeme müssen vielmehr endbenutzergerechte Konzepte und Methoden entwickelt und zur Verfügung gestellt werden. Brooks weist jedoch darauf hin, dass diese Methoden lediglich unbeabsichtigte Schwierigkeiten, aber nicht die der Softwareentwicklung inhärente Komplexität überwinden können [25].

Entwicklung von EUD

Der Begriff EUD selbst ist noch relativ jung, jedoch ist EUD kein neues Forschungsfeld. Es baut vielmehr auf einer Vielzahl bestehender, endbenutzerorientierter Ansätze, wie dem *End-User Computing* und dem *Tailoring*, sowie auf Aspekten der partizipativen Softwareentwicklung, auf. Bereits früh wurde mit Methoden wie der objektorientierten Programmierung und Programmiersprachen wie Smalltalk oder BASIC versucht die Kommunikation zwischen Benutzer und System zu vereinfachen und so dem Benutzer Veränderungen des Systems zu ermöglichen (vgl. Hale [60]).

Ein erster Ansatz des *End-User Programmings* ist ein Mitte der 70er Jahre vorgestelltes *Programming by Example System (Pygmalion)* von Smith [157]. Seit Mitte der 80er Jahre

wurden unter dem Begriff *End-User Computing* (und zum Teil auch *End-User Programming*) Methoden der Aneignung und Nutzung von Informationstechnologie populär (Brancheau und Brown [24]). Konzepte wie *Visuelle Programmierung (VP)*, *Programming by Demonstration* und *Natural Language Programming* sollen Personen außerhalb von IT-Abteilungen die Entwicklung von Software, mit dem Ziel der Unterstützung organisatorischer Aufgaben, ermöglichen. Im Vordergrund stehen die Unterstützung der Programmierung und programmiernaher Tätigkeiten (z.B. Debugging), sowie die Anpassung über Parameter und einfache Skripte. Auf Basis der Erkenntnis, dass die Anpassung von Systemen an die Bedürfnisse der Endbenutzer zwar auch die Programmierung einschließt, sich jedoch nicht auf diese beschränkt, weitet EUD den Fokus auf Entwicklungstätigkeiten allgemein aus [144].

Ein weiteres eng verknüpft Konzept von EUD stellt das zu Beginn der 90er Jahre von Henderson und Kyng [66] geprägte *Tailoring* dar. Tailoring definiert sich nach Henderson und Kyng als Aktivität zur Modifikation einer Computeranwendung im Nutzungskontext, die nach dem ursprünglichen Design und der Implementierung des Systems stattfindet. Einen der ersten wissenschaftlichen Prototyp stellt das *Buttons*-System von MacLean et al. dar [109]. Weitere Tailoring-Ansätze finden sich im Bereich komponenten- und servicebasierter Systeme (Alda et al. [4] und Wulf et al. [174]), die in Kapitel 2.2.3 angesprochen werden.

Im Kontrast zu dem hauptsächlich in Europa geprägten Begriff EUD hat sich in den letzten Jahren im amerikanischen Raum der Begriff *End-User Software Engineering (EUSE)* [28] herausgebildet. EUSE fordert, dass Endbenutzer in allen Bereichen des Softwarelebenszyklus, also auch bei dem Testen und der Wartung, unterstützt werden.

Definition von Endbenutzern

Endbenutzer sind allgemein Anwender von Informationstechnologie, die diese nutzen, um ihre regulären Aufgaben zu bewältigen. Dementsprechend findet sich in der Praxis ein breites Spektrum verschiedener Typen von Endbenutzern und in der Literatur zu EUD verschiedene Klassifizierungsansätze. Fischer [44] beschreibt ein Kontinuum an Endbenutzern anhand unterschiedlicher Affinität zur Informationstechnologie. Dieses Kontinuum reicht von *passive consumer* über *active consumer*, *end-user*, *power user* und *domain designer* bis hin zu *meta-designer* (Abbildung 6). Nardi und Miller [122] klassifizieren Endbenutzer anhand ihres technischen Wissens in *end-user* (auch *non-programmer*), *local developer* und *professional programmer*. *Non-programmer* besitzen keinerlei Programmierausbildung und zeigen ein geringes Eigeninteresse gegenüber Computern. *Local developer* sind Domänenexperten, die bereits fortgeschrittene Kenntnisse von Computern und Computersystemen besitzen und in gewisser Weise einen Ansprechpartner für *non-programmer* darstellen. Professionelle Entwickler werden unter *professional programmer* eingeordnet. Eine ähnliche Klassifizierung findet sich auch bei MacLean [109], der statt *local developer* die Bezeichnung *tinkerer*

verwendet. Scaffidi et al. [147] wählen einen anderen Ansatz und differenzieren anhand verschiedener Softwarenutzungspraxen und der Art wie Endbenutzer mit Abstraktionen umgehen. Burnett et al. [29] zeigen schließlich, dass auch geschlechtsspezifische Aspekte bei der Nutzung von Softwareentwicklungswerkzeugen durch Endbenutzer eine Rolle spielen.

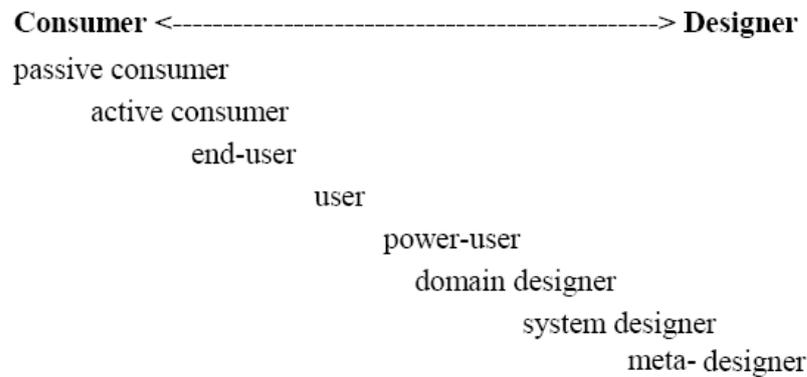


Abbildung 6: Kontinuum von Endbenutzern nach Fischer [46]

Die Existenz verschiedener Typen von Endbenutzern zeigt, dass bei der Entwicklung von Systemen geeignete EUD-Methoden für die anzusprechende Nutzergruppe ausgewählt werden müssen. In kollaborativen Szenarien wird dies jedoch erschwert durch die Beteiligung verschiedener Nutzertypen mit unterschiedlichen Kenntnissen.

2.2.2 Gestaltung von EUD-Systemen

In der Vergangenheit wurden verschiedene Anforderungen, Gestaltungsansätze und Lösungsansätze für EUD-Systeme identifiziert und entwickelt. Im Folgenden werden die relevantesten Aspekte vorgestellt.

Basisanforderungen an EUD-Systeme

In der Literatur zu EUD finden sich verschiedene Anforderungen an Systeme die Endbenutzern helfen sollen effektiver zu arbeiten. Allem voran sei die bereits angesprochene Gestaltung und Auswahl geeigneter Methoden für die adressierte Gruppe von Endbenutzer genannt. Repenning und Ioannidou [144] fordern von EUD-Systemen einen *flow of learning*, der sorgfältig die Fähigkeiten und Herausforderungen für die Nutzer ausbalanciert und Endbenutzer schrittweise befähigt die benötigten Fähigkeiten zur Bewältigung von Entwicklungsherausforderung zu erwerben. Dies soll den von MacLean [109] beschriebenen *customization gulf* vermeiden, so dass Endbenutzer nicht durch Sprünge der Anpassungskomplexität überfordert und demotiviert werden. Mit einem Fokus auf das Tailoring komponentenbasierter EUD-Systeme fordern Wulf et al. [174] ein Framework für das Re-Design der Software bereitzustellen, endbenutzerorientierte Konzepte und Interfaces zu verwenden und eine starke Kongruenz zwischen Architektur- und Interface-Konzepten

anzustreben. Dittrich et al. [38] weisen darauf hin, dass es nicht ausreicht nur die direkte Anpassung zu unterstützen. Stattdessen müssen EUD-Aspekte auch in bestehende Entwicklungsprozesse eingebunden und Wartungs- und Wiederverwendungsaspekte berücksichtigt werden. Dies führt dazu, dass die Anforderungen an ein EUD-System nicht nur durch den Nutzungskontext, sondern auch durch bestehende Techniken und Entwicklungsprozesse beeinflusst werden. Zudem sollen auch geeignete Konzepte für das Testen und die Wiederverwendung von Softwareanpassungen angeboten werden (Burnett et al. [28]). Da Endbenutzer Entwicklungs- und Anpassungsaktivitäten typischerweise kollaborativ durchführen (vgl. [121]) und die gemeinsame Anpassung mehrerer Nutzer mit unterschiedlichen Erfahrungsniveaus zu intra- und inter-organisatorischen Delegationsprozessen führen kann (vgl. Dörner et al. [40]), soll auch eine kollaborative Erstellung und Nutzung von Artefakten geeignet unterstützt werden (vgl. [38, 174]).

Gestaltungsgrundsätze und Meta-Level Ansätze

Im folgenden Abschnitt werden grundsätzliche Gestaltungsansätze für EUD-Systeme beschrieben. Weitere, bereits sehr konkrete Designrichtlinien für die Gestaltung von EUD-Systemen finden sich bei Repenning und Ioannidou [144].

Gentle Slope of Complexity: Das Konzept des *Gentle Slope of Complexity* geht auf MacLean et al. [109] zurück. Es stellt an EUD-Systeme die Forderung eine große Bandbreite an Anpassungsmethoden mit einer jeweils geringen Abstufung der Anpassungsmächtigkeit und Anpassungskomplexität anzubieten. Damit wird der beschriebenen Forderung des *flow of learnings* entsprochen, da die zusätzlich erforderten Fähigkeiten für die nächst komplexere Anpassung immer nur ein kleines Stück höher ist. Dies ermöglicht einen graduellen Lernprozess der Endbenutzer. Dementsprechend müssen für verschiedene Fähigkeits- und Komplexitätslevel geeignete Methoden entwickelt oder ausgewählt werden. Spahn et al. [161] bekräftigen die Forderung nach einer Reihe von Anpassungsmethoden und geben Empfehlungen für die Gestaltung eines idealtypischen Komplexitätsanstiegs (Abbildung 7).

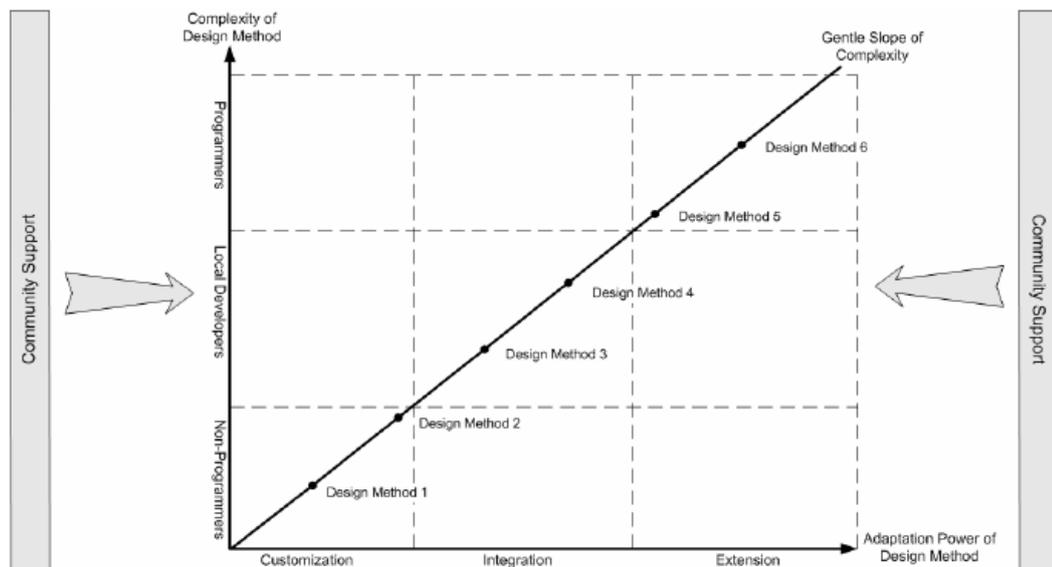


Abbildung 7: Gestaltung eines EUD-Systems mit einem idealisierten Gentle Slope of Complexity [161]

Meta-Design: Ein Kernaspekt von EUD ist die Erstellung von Systemen, die es Endbenutzern erlauben ihre eigenen Systeme anzupassen und so selbst als Designer zu agieren. EUD ist damit als zweistufiger Prozess zu sehen. Im ersten Schritt wird die Design-Umgebung erstellt. Diese wird im zweiten Schritt dann genutzt, um die entsprechenden Anwendungen anzupassen. Dieses als Meta-Design benannte Konzept geht vor allem auf Fischer zurück [45, 46]. Fischer schlägt zudem das *SER-Modell* als inkrementelles Vorgehensmodell für das Meta-Design vor. Es beschreibt wie Meta-Design Systeme verstanden werden können und wie offene Systeme sich über die Zeit entwickeln. Andere Ansätze für die Einbeziehung von Nutzern in den Designprozess sind das *User-Centered Design* und im stärkeren Maße das *Participatory Design* (siehe [116, 117]), sowie das Konzept der *Software Shaping Workshops* von Costabile [33].

Aufgaben- und Domänenorientierung: Eine Schwierigkeit bei der Nutzung typischer Programmiersprachen ist die mentale Umsetzung domänenspezifischer Probleme auf die generischen Konzepte der Programmiersprachen. Aufgabenspezifische Programmiersprachen und domänenorientierte Konzepte sollen helfen die konzeptuelle Distanz zwischen der Semantik der Anwendungsdomäne und des Softwareartefakts signifikant zu reduzieren (siehe Fischer [43] und Nardi [121]). Dies verringert einerseits die Herausforderung zur Nutzung des Systems und so auch den Lernaufwand für die Endbenutzer. Andererseits schränkt die Fokussierung auf bestimmte Aufgaben oder Domänen jedoch die Generalität und damit eine breite Einsetzbarkeit des Systems ein (vgl. Repenning und Ioannidou [144]). Zudem müssen Notationen für die verschiedenen Aufgaben oder Domänen erst erstellt werden. Ein alternativer Ansatz ist daher die Gestaltung Meta-domänenspezifischer Konzepte (vgl. [144]).

Usability: Das Usability-Engineering ist ein allgemeiner Ansatz der HCI (Human Computer Interaction), der darauf abzielt Systeme zu entwerfen, die einfach zu benutzen, einfach zu erlernen und flexibel sind, sowie eine positive Einstellung beim Benutzer erzeugen (Benyon et al. [14]). Viele dieser Aspekte sind wichtige Anforderungen an EUD-Systeme. Im Fokus der Usability steht meist die Gebrauchstauglichkeit nach ISO-Norm 9241-11. Ein häufig eingesetztes Analysemittel für die Gestaltung von EUD-Programmierungsumgebungen ist zudem das *Cognitive Dimensions (CD) Framework* von Green et al. [53, 54]. Das Framework ist ein allgemeiner theoretischer Ansatz zur Analyse und Diskussion der Usability von Entwicklungsumgebungen, mit besonderem Fokus auf eine visuelle Programmierung. Es beinhaltet keine Methoden zur exakten Berechnung der Usability, sondern dient als „Diskussionswerkzeug“. Als Diskussionsgrundlage dienen zwölf kognitive Dimensionen, die sich auf strukturelle Eigenschaften der Systeme beziehen. Ein System wird dabei als Gesamtkonzept, bestehend aus Notation, dem Werkzeug und dem Medium, gesehen. Beispielphaft seien mit Bezug auf EUD die Dimensionen *Closeness of Mapping* (wie domänenorientiert ist das System), *Imposed Guess-ahead* (muss die Programmstruktur im Voraus geplant werden) und *Visibility* (was ist ohne kognitiven Aufwand erfassbar) genannt. Obwohl die CDs relativ unabhängige Aspekte beschreiben, existieren über die Auswirkungen auf das Gesamtsystem Zielkonflikte zwischen einzelnen CDs (z.B. *Closeness of Mapping* vs. hohe Abstraktion). Daher ist es nicht möglich das „perfekte“ System zu entwerfen. Vielmehr ist es nötig die Zielkonflikte hinsichtlich der Einsatzdomäne sorgfältig abzuwägen. Nach der ersten Veröffentlichung wurden verschiedene Erweiterungen für das CD-Framework entwickelt (siehe Blackwell [17]).

Unterstützung von Kollaboration: Bereits früh wurde in empirischen Untersuchungen die hohe Bedeutung kollaborativer Aktivitäten im Bereich EUD, sowohl zwischen verschiedenen Endbenutzern, als auch zwischen Endbenutzern und Entwicklern, festgestellt (siehe Mackay [107] und Nardi [121]). Eine weitergehende Betrachtung von Kollaborationsaspekten bei Endbenutzern, mit Fokus auf Tailoring-Aktivitäten, findet sich bei Pipek und Kahler [139]. Die Kollaboration findet statt und kann unterstützt werden durch die Wiederverwendung von Software-Artefakten und durch die direkte Kommunikation über face-to-face oder Computer-Mediation. Da in kollaborativen Szenarien häufig Endbenutzer mit verschiedenen Erfahrungsstufen involviert sind, spielen zum Beispiel intra- und interorganisationale Delegationsmuster eine große Rolle (vgl. Dörner et al. [40]). Aufbauend auf dem theoretischen Konzept der *Situated Action Theory* von Suchman [163], bietet Hundhausen [74] ein Framework zur Analyse und Diskussion der Unterstützung von Kommunikation in endbenutzerorientierten Visualisierungsumgebungen. Hundhausen erweitert dazu das Konzept der CDs um die sechs kommunikativen Dimensionen *Programming Saliency*, *Provisionality*, *Story Content*, *Modifiability*, *Controllability* und *Referencability*.

Lösungsansätze für EUD-Systeme

Forschungsansätze im Bereich EUD umfassen ein breites Feld an Methoden von der Anpassung durch Parameter über die Kontrolle komplexer Geräte bis hin zur Erstellung von Code. Grundlegend lassen sich diese anhand ihrer Mächtigkeit und Komplexität differenzieren in Parametrisierungs- oder Anpassungsansätze und in Methoden zur Programmerstellung oder -modifikation (Lieberman et al. [102]). Eine gute Übersicht aktueller Ansätze bietet das Buch *End-User Development* [103] und das *Deutsche End User Development Wiki*⁴. Eine umfassende Taxonomie aller EUD-Ansätze fehlt bisher jedoch. Einen vielversprechenden Ansatz zur Kategorisierung von EUD-Methoden nach ihrer Komplexität und ihrer inhärenten Anpassungsmächtigkeit liefern Spahn et al. [161].

Visuelle Programmierung (VP): Die visuelle Programmierung ist ein häufig adressiertes Gestaltungsparadigma des EUD. VP-Umgebungen sollen Endbenutzern eine schnelle und einfache Erstellung grafischer Repräsentationen (sowohl statische als auch dynamische) von Programmlogik und/oder Benutzerschnittstellen ermöglichen. Die Programmierung erfolgt primär visuell durch die Zusammenstellung von Piktogrammen, die Daten und Operationen repräsentieren, oder basierend auf Diagramm-Repräsentationen⁵ (vgl. Schiffer [150]). Um eine natürlichere Kommunikation zwischen Endbenutzern und dem System zu erreichen, sind die visuellen Repräsentationen in EUD-Systemen häufig an der Anwendungsdomäne orientiert (vgl. Mehandjiev und Bottaci [111]). VP ist jedoch nicht ausschließlich auf visuelle Elemente beschränkt, sondern besteht in der Regel aus einer Kombination visueller und verbaler Inhalte. Verbale Beschreibungen werden zum Beispiel zur Darstellung dichter Informationen (Kommentare oder Erläuterungen zu wichtigen Programmstellen) verwendet. Eine populäre VP-Umgebung ist das kommerzielle System *LabVIEW* [32]. Dieses ermöglicht die visuelle Erstellung des Datenflusses und der Benutzerschnittstelle von virtuellen Instrumenten. Weitere Beispiele endbenutzerorientierter VP-Ansätze sind *AgentSheets* [145] und *DENIM* [123]. *AgentSheets* ist eine Plattform mit der Lehrer und Studenten eigene grafische (wissenschaftliche) Simulationen erstellen können. *DENIM* ermöglicht Designern schnell und einfach interaktive Prototypen von Webseiten zu skizzieren (siehe Kapitel 2.3.5). VP bietet eine Reihe von Potentialen, insbesondere bei der Vermittlung von Ideen und der Darstellung komplexer Zusammenhänge. Endbenutzern kommt vor allem die effektive kognitive Verarbeitung visueller Informationen, ein erhöhter Aufmerksamkeitswert, ein großes Motivations- und Lernpotential, sowie eine Abschwächung syntaktischer Strukturen zu Gute [150]. Studien zeigen zudem, dass visuelle Sprachen im starken Maße die

⁴ <http://www.enduserdevelopment.de/>

⁵ Piktogramme beschreiben sich durch ihren ikonischen Charakter selbst und sind so direkt verständlich, Diagrammelemente besitzen dagegen eine „Codierung“. Für das Verständnis der Diagramme ist daher ein Vorwissen über die Semantik der Elemente notwendig.

Kommunikation zwischen Entwicklern und Anwendern verbessern (vgl. [12, 114]). Jedoch besitzen visuelle Programmiersprachen (auch jenseits der von Dijkstra befürchteten „Infantilisierung der Computerwissenschaft“ [37]) eine Reihe von Nachteilen gegenüber typischen, verbalen Programmiersprachen. Hervorzuheben sind eine geringe Darstellungsdichte, eine hohe Resistenz gegenüber Veränderungen und eine schlechte Formalisierbarkeit. Vor allem die unökonomische Nutzung der Darstellungsfläche und eine schlechte Skalierbarkeit schränken die Nutzung für komplexe Probleme ein [150]. Für den Einsatz und die Gestaltung visueller Programmierumgebungen sind daher die einzelnen Vor- und Nachteile genau abzuwägen.

Programming by Example (PbE) (auch: Programming by Demonstration): Bei PbE findet keine direkte Programmierung statt. Der Benutzer demonstriert dem System stattdessen benötigte Interaktionen an einer begrenzten Menge an Objekten. Das System versucht diese dann zu generalisieren und automatische Routinen, die über das einfache Wiederholen der Aktionen hinausgehen, zu erstellen (siehe Lieberman [100]). Ein Beispiel ist das *AgentTalk* System [145], bei dem Simulationsabläufe mittels PbE erstellt werden können. Eine detaillierte Einführung und Übersicht über PbE geben Cypher [34] und Liebermann [100].

Natural Programming (NP) & Natural Language Programming (NLP): Ziel von NP ist die Entwicklung von Programmiersprachen und -umgebungen, die möglichst natürlich oder zumindest näher an der Art und Weise sind, wie Benutzer über ihre Aufgaben denken (Myers et al. [118]). Im Optimalfall sollen die Sprachen und Entwicklungsumgebungen so arbeiten, wie Nicht-Programmierer es erwarten. Um dies zu erreichen und entsprechende Werkzeuge erstellen zu können, ist ein genaues Verständnis der anzusprechenden Nutzer und der Anwendungsdomäne nötig. Ein zentraler Punkt für NP ist die Nähe der Notation zu dem mentalen Modell der Benutzer. Je ähnlicher sich beide sind, desto einfacher können Endbenutzer ihre Ideen so ausdrücken, wie sie über diese denken. Die natürlichere Programmierung geht jedoch auf Kosten der Effektivität und Formalisierbarkeit. Gute Beispiele natürlicher System sind *HANDS*, eine eventbasierte Programmierumgebung für Kinder, und als ein Teilaspekt das *Whyline-Konzept*, welches ein natürlicheres Debugging basierend auf Fragen ermöglicht [118, 136]. Einen Schritt weiter geht das Konzept von NLP, welches nach Lieberman und Liu [101] darauf abzielt natürliche Sprache als Kommunikationsmedium zwischen Mensch und Computer zu verwenden, um gemeinsam ein Programm zu konstruieren.

Weitere Ansätze zur Gestaltung endbenutzergerechter Programmiertechniken sind *Form-based Programming*, *Programming by Example Modification* und *Scripting* (z.B. über VBA) (siehe Nardi [121]). Zwei verbreitete Ansätze zur Anpassung von Software durch End-User sind *Interface Customization* und *Parametrisierung* (siehe Spahn et al. [161]). Ein Beispiel, das unter anderem diese beiden Konzepte einsetzt, ist das *Buttons*-System von MacLean

[109]. Eine Betrachtung unterstützender Ansätze für ein ganzheitliches Software-Engineering finden sich bei Burnett et al. [28].

2.2.3 EUD in SOA

Durch die Möglichkeit, Anwendungen flexibel zusammenzustellen, einzelne Komponenten während der Nutzung zu ersetzen, sowie dem Fokus auf Geschäftsprozesse (und damit die Anwendungsdomäne der Nutzer), bieten serviceorientierte Architekturen eine geeignete Grundlage für die Entwicklung anpassbarer EUD-Systeme. Da SOA ein relativ neues Feld ist, existieren bisher jedoch erst wenige Untersuchungen zur Verbindung von EUD und SOA. Ein zu SOA ähnliches und im Bereich EUD schon länger untersuchtes Konzept flexibler Infrastrukturen sind *komponentenbasierte Systeme*. Ähnlich wie in SOA lassen sich Systeme durch die Zusammenstellung von gekapselten Komponenten anpassen, ohne dass die spezifischen Implementierungsdetails der einzelnen Komponenten dem Benutzer bekannt sein müssen (vgl. Mørch et al. [115]). Besonders hervorzuheben ist die Forschung rund um die auf dem Softwarekomponentenmodell *FlexiBeans* aufbauende, komponentenbasierte Tailoring-Plattform *FreEvolve*. In deren Kontext wurde eine Vielzahl an EUD-Konzepten und Visualisierungsansätzen entwickelt und erprobt. Ein detaillierte Übersicht bestehender Forschung zu Komponentenbasierten Systemen findet sich bei Wulf et al. [174].

Ansätze und Herausforderungen von EUD in SOA

Nach Beobachtungen von Brahe und Schmidt [23] besteht bei der Gestaltung von serviceorientierten Prozessen in der Praxis die Notwendigkeit zur Einbeziehung von Nutzern. Durch eine kollaborative Prozesserstellung von Endbenutzern und Entwicklern können nötige Rückfragen verringert und ein wechselseitiger Lernprozess angestoßen werden. Andere Forscher gehen einen Schritt weiter und fordern Systeme, mit denen Endbenutzer die Möglichkeit haben selbstständig geeignet zu orchestrieren (z.B. Han et al. [63] und Oinn et al. [130]). Bei Bedarf sollen sie selbstständig Prozesse erstellen und damit flexibler und schneller an personalisierte Informationen gelangen können. Für die Erstellung solcher Systeme treffen im Allgemeinen die in Kapitel 2.2.2 vorgestellten EUD-Kriterien zu. Da SOA jedoch für professionelle Entwickler und ohne Berücksichtigung von Endbenutzern und ihren Anforderungen konzipiert wurde, haben serviceorientierte EUD-Systeme besonderen Herausforderungen Rechnung zu tragen. Eine allgemeine Betrachtung organisatorischer Probleme von SOA in der Praxis, die auch die Rolle von Endbenutzern berücksichtigt, findet sich bei Brahe und Schmidt [23]. Eine Identifikation EUD-spezifischer Probleme und erster Lösungsansätze liefern Dörner et al. [41]. Identifizierte Probleme sind vor allem eine zu hohe Komplexität bestehender Modellierungsinterfaces (und daraus resultierend eine schlechte Erlernbarkeit), ungeeignete Mechanismen für das Auffinden und Verstehen von Services, sowie eine für Endbenutzer ungeeignete Granularität existierender Services. Auch bieten

EUD-Werkzeuge den Nutzern keine Möglichkeit für sie unbekannte Konzepte, zum Beispiel durch eine informelle Modellierung, auszudrücken.

Modellierung und Orchestrierung durch Endbenutzer

Wie Untersuchungen zeigen, sind aktuelle Modellierungs- und Orchestrierungswerkzeuge, aufgrund ihrer Konzeption für professionelle Entwickler, für Endbenutzer ungeeignet. Sie besitzen in der Regel eine hohe Komplexität und Formalität und erfordern einen hohen Einarbeitungsaufwand. Dies ermöglicht es Endbenutzern nicht, das Konzept einer serviceorientierten Prozesserstellung einfach zu verstehen und anzuwenden (vgl. [63, 104]). Darauf aufbauend haben sich verschiedene Forschungsansätze mit dem Ziel der Gestaltung endbenutzergerechter Modellierungsumgebungen gebildet. Der Fokus der Ansätze liegt, neben der Auswahl, Filterung und Gruppierung geeigneter Services, auch auf der Gestaltung geeigneter Notationen (siehe Han et al. [63]). Eine Sonderrolle nimmt die *End-User Service Orchestration Platform (EUSOP)* von Hofmann et al. [70] ein. Sie adressiert technische Einschränkungen der Web Service Architektur und stellt ein Basis-Framework für grafische Orchestrierungsansätze bereit. EUSOP bietet architektonische Erweiterungen, wie die Speicherung zusätzlicher Meta-Daten, erweiterte WSDL-Dokumentationen und eine Schnittstelle für grafische Kompositionswerkzeuge. Ein konzeptioneller Ansatz zur Gestaltung endbenutzergerechter Services ist das Konzept der *End User Visible Services* [175]. Diese kapseln normale Services in für Endbenutzer relevante Geschäftsaufgaben und bilden so eine Zwischenschicht zwischen realen Services und der Welt der Benutzer.

Übersicht endbenutzerorientierter Modellierungs- und Orchestrierungssysteme

Ein erster Ansatz zur Gestaltung einer Orchestrierung durch Endbenutzer ist die, als endbenutzergerechte Alternative zu BPEL entwickelte, textuelle *Simple Service Composition Language (SSCL)* [50]. Die Mehrheit der entwickelten Ansätze baut jedoch auf einer visuellen Komposition auf der Basis von „Kästchen“ (= Services oder Aktivitäten) und „Linien“ (= Prozessfluss) auf.⁶ Zum Teil sind die Kästchen zudem strukturiert und die Elemente mit textuellen Informationen angereichert. Eine Ausnahme stellt die Notation des *VINCA*-Systems [64] dar. Dort wird der Prozessablauf als Baumstruktur visualisiert. Eine Übersicht existierender Modellierungs- und Orchestrierungssysteme für Endbenutzer findet sich in Tabelle 1. Unter den Ansätzen finden sich sowohl domänenunabhängige, als auch domänenspezifische Werkzeuge. Eine Sonderrolle nimmt der *Simple Service Orchestration (SiSeOr)* Ansatz [135] ein. Durch einen Austausch der Visualisierungssymbole lässt sich das System an verschiedene Domänen anpassen. Der ganzheitliche *End User Service Composition*

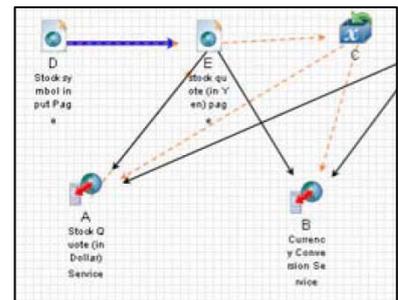
⁶ In Folgenden wird dieser Visualisierungsansatz als *Boxes-and-Wires* Konzept bezeichnet.

Ansatz [104] unterstützt als einziges System eine Modellierung in verschiedenen Sichten (= Darstellungsformen) und damit unterschiedliche Komplexitätsstufen (siehe Konzept der *Multiple Representations* von Mørch und Mehandjiev [114]). Allgemein sind die vorgestellten Ansätze für eine einfache Modellierung kleiner Prozesse konzipiert. Ebenso wie die professionellen Umgebungen, besitzen sie jedoch keine Möglichkeit unbekannte Aspekte, zum Beispiel durch informelle Anmerkungen, auszudrücken. Auch erfordert ihre Nutzung bereits sehr konkreter Vorstellungen und Kenntnisse über die verwendeten Services.

Meta-Domänen Ansätze

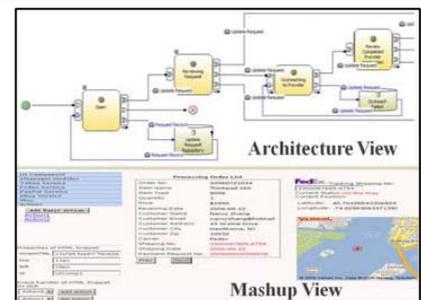
ADIEU [154]

ADIEU ist ein von IBM entwickeltes assistentenbasiertes Werkzeug zur ad hoc Komposition von Services und der Erstellung dazugehöriger webbasierter GUIs. Das System verwendet viele technische Begriffe und besitzt eine Vielzahl verschiedener Verbindungselemente.



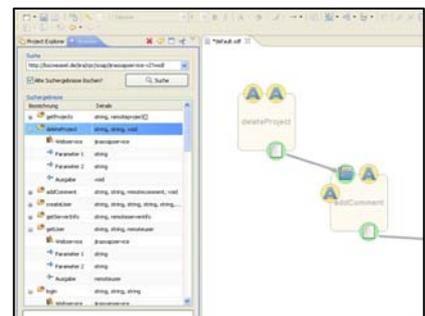
End User Service Composition [104]

Das Konzept von Liu et al. ermöglicht die Komposition von Web Service in einem zweistufigen Ansatz. In einer Mashup-Sicht können die Services auf Basis ihrer visuellen Repräsentation zusammengestellt werden. Eine detailliertere und komplexere Modellierung ist in einer Architektur-Sicht auf Basis von BPMN möglich.



SiSeOr [135]

SiSeOr ist ein grafischer BPEL-Editor, der unter anderem durch die Berücksichtigung domänenspezifischer, kollaborativer und kognitiver Faktoren für Endbenutzer mit geringem Lern- und Nutzungsaufwand einsetzbar ist. Er besitzt ein feingranular anpassbares, grafisches Interface mit der Möglichkeit austauschbarer domänenspezifischer Visualisierungskonzepte. Im Gegensatz zu den anderen Systemen wird bei SiSeOr primär der Datenfluss modelliert.



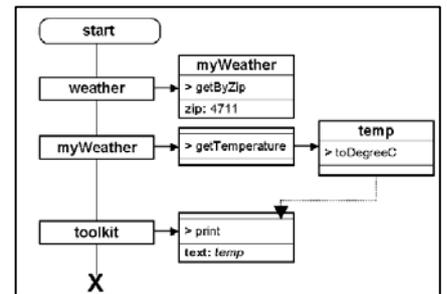
VINCA [64]

VINCA ist ein Werkzeug zur schnellen Komposition personalisierter, webbasierter Services durch Endbenutzer, um kurzfristige Anforderungen selbstständig lösen zu können. Der Prozessfluss wird, im Gegensatz zu den anderen Ansätzen, als Baumsstruktur visualisiert.

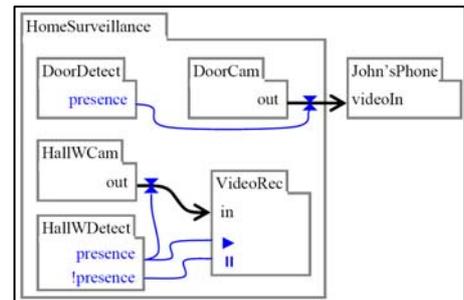


WIDE [68]

WIDE ist ein webbasiertes Modellierungssystem das sich an web-affine Endbenutzer richtet. Die Notation besitzt eine stark formalisierte Syntax, die an BPEL angelehnt ist. Die Visualisierung ähnelt einer Mischung aus UML-Klassen- und Ablaufdiagrammen.

**Domänenspezifische Ansätze****uDesign [160]**

Das Konzept von uDesign ermöglicht Endbenutzern in der, durch hohe Personalisierung und dynamische Anforderungen geprägten Domäne der „Smart Spaces“, Systeme für die Überwachung und Kontrolle dieser zu entwerfen und weiterzuentwickeln. Die Visualisierung baut auf dem Konzept der Boxes-and-Wires auf.

**Sucfl [130]**

Sucfl ist ein domänenspezifisches Werkzeug, das „e-Scientists“ die Möglichkeit gibt, das Internet aus einer serviceorientierten Sicht zu sehen und individuelle Workflows zu erstellen. Dazu können Wissenschaftler verfügbare Services suchen, Workflows durch die Zusammenstellung von Web Services erstellen und diese abschließend testen und ausführen. Die Modellierung des Prozessflusses erfolgt grafisch, die Zuordnung von Parametern durch textuelle Ergänzungen.

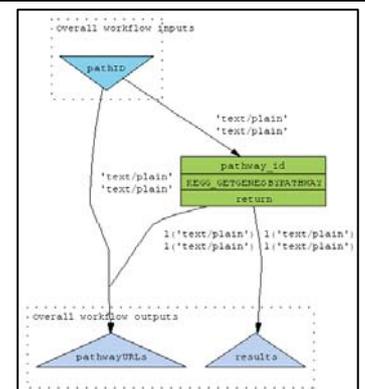


Tabelle 1: Übersicht endbenutzerorientierter Modellierungs- und Orchestrierungssysteme

2.3 Papier- und skizzenbasierte Modellierung

Ein Aspekt von EUD und HCI allgemein ist die Gestaltung geeigneter Benutzerschnittstellen. Eine Alternative zu „typischen“ grafischen Benutzerschnittstellen und der Eingabe über Tastatur und Maus sind greifbare („tangible“), papierbasierte Interfaces und skizzenbasierte Anwendungen. Im Folgenden werden Grundlagen, existierende Ansätze und Gestaltungsaspekte papier- und skizzenbasierter Systeme beschrieben.

2.3.1 Grundlagen papierbasierter Systeme

Die Grundidee zur Verwendung papierbasierter System beschrieb bereits 1991 Weiser in seinem Artikel „The Computer for the 21st Century“ [167]:

“At breakfast Sal reads the news. She still prefers the paper form, as do most people. She spots an interesting quote from a columnist in the business section. She wipes her

pen over the newspaper's name, date, section, and page number and then circles the quote. The pen sends a message to the paper, which transmits the quote to her office.”
(Seite 7)

Entstehung und Entwicklung stift- und papierbasierter Systeme

Stift und Papier haben eine lange Tradition in der Aufzeichnung von Informationen und Ideen. Sie sind neben Sprache und Gesten eines der natürlichsten Medien zur Kommunikation mit anderen Menschen. In der Kommunikation mit Computern sind Nutzer jedoch meist an die *WIMP*-Interaktionsmetapher (*windows, icon, menu, pointer*) gebunden. Daher erfordert die Interaktion mit Systemen die Nutzung formaler Sprachen und die Steuerung über Tastatur und Maus (Le et al. [94]). Besonders in kreativen Szenarien werden wichtige Denkprozesse durch exzessiven Wechsel zwischen Menüs, Aktionen mit Knöpfen und nötige Tastatureingabe häufig unterbrochen (Dai [35]). Als Alternative sind in den letzten Jahren zunehmend greifbare Interaktionssysteme (siehe z.B. Iacucci et al. [76]), insbesondere auf Basis einer stiftbasierten Eingabe, populär geworden. Kernaspekt dieser Systeme ist die „informelle Natur“ und der natürliche Umgang der Nutzer mit den Interaktionsmedien (z.B. Stiften). Dies erlaubt es Nutzern sich auf den Inhalt der Interaktion und nicht auf den Interaktionsablauf selbst zu konzentrieren (Subrahmonia und Zimmerman [162]). Noch einen Schritt weiter gehen Systeme, die Gebrauch von aufkommenden, papierbasierten Nutzerinterfacekonzepten (PBUI) machen. PBUIs ermöglichen nicht nur eine natürlichere Nutzung als zum Beispiel stiftbasierte Tablet-Interfaces (vgl. [42, 94]), sondern verknüpfen physische und digitale Artefakte, um die Nutzungstradition von Papier mit den Verarbeitungsmöglichkeiten moderner Computer zu kombinieren und Papier um interaktive Möglichkeiten anzureichern (= interaktives Papier). Ziel ist daher nicht das digitale System dem physischen nachzuempfinden und papierbasierte Arbeitsweisen zu digitalisieren, sondern, dass sich beide Systeme gegenseitig anreichern und komplementär ergänzen [35]. Diese Idee wird auch gestützt von Sellen und Harper, die in ihrem Standardwerk „The Myth of the paperless Office“ [152] proklamieren, dass, entgegen dem „Mythos“ des papierlosen Büros, Papier weiterhin eine große Rolle in Büroumgebungen spielen wird, vor allem als temporäres Medium und zunehmend in Verbindung mit elektronischen Werkzeugen.

Interaktive Papiertechnologien

Erste Ansätze zur Integration der physischen Welt mit einem digitalen Schreibtisch gehen zurück auf den *Digital Desk* von Wellner [168]. Bei Wellners System, aber auch bei neueren Ansätzen (z.B. Holman et al. [71]), erfolgt die Digitalisierung der Papierinteraktion über Kameras und entsprechende Auswertungsalgorithmen. Dies hat den Vorteil, dass nicht nur die Interaktion mit dem Stift, sondern auch Papierbewegungen und Handgesten erfasst werden. Jedoch schränkt ein solches System, zum Beispiel durch den nötigen Aufbau von Kameras,

viele Eigenschaften von Papier, allem voran die flexible Nutzbarkeit sowie Kostenvorteile, ein. Die meisten aktuellen PBUI-Systeme verwenden daher neuartige Digitalstifttechnologien, bei der die Eingabe direkt durch einen Stift erfasst wird. Im Gegensatz zu dem Kameraansatz kann jedoch nur die direkte Stifteingabe erfasst werden. In den letzten Jahren haben sich verschiedenste Digitalstifttechnologien entwickelt. Eine detaillierte Übersicht technischer Ansätze findet sich bei Le, Signer und Subrahmonia [94, 155, 162]. Hervorzuheben für den Bereich interaktiver Papieranwendungen sind Stifte auf Basis von Hochfrequenztechnologie (z.B. EPOS⁷), von Bewegungserkennung und von optischer Mustererkennung. Unter diesen kommt letzteren eine besondere Bedeutung zu, da sie nicht nur die relative Erfassung von Strichen, sondern die absolute Bestimmung der Stiftposition auf dem Papier ermöglichen.⁸ Dies erfordert jedoch eine eindeutige Kennzeichnung jedes Papiers über ein eindeutig aufgedrucktes Muster. Bekanntestes Beispiel ist die *Anoto-Technologie* [6] (Abbildung 8). Eine Einschränkung aktueller interaktiver Papiere ist die fehlende Möglichkeit einmal gezeichnete Striche wieder zu entfernen. Auch die nahe liegende Möglichkeit mit Papier zu „basteln“ kann mit aktueller Digitalstifttechnologie nicht erfasst werden.

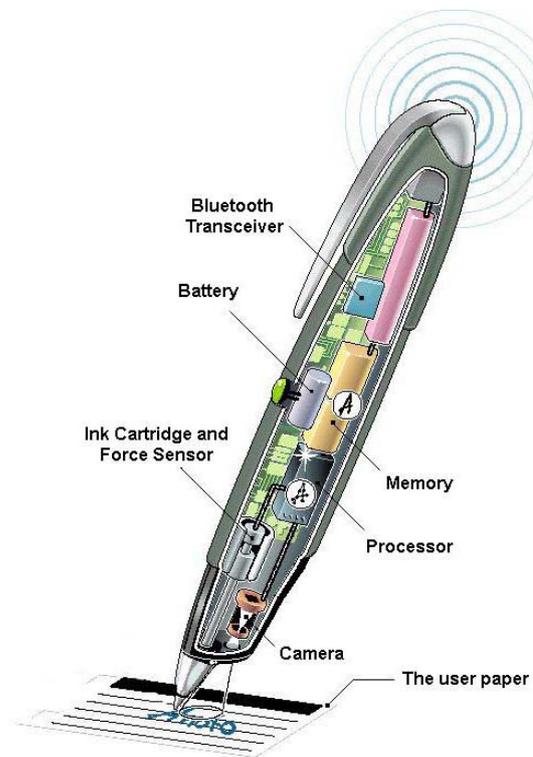


Abbildung 8: Anoto Digitalstift [6]

2.3.2 Chancen und Herausforderungen papierbasierter Systeme

Trotz vielfältiger Bemühungen um das „papierlose Büro“ und der damit verbundenen Erwartung der Ablösung von Papier durch elektronische Geräte (z.B. Gwizdka et al [59]), spielt Papier in der alltäglichen Arbeitspraxis weiterhin eine wichtige Rolle (siehe Sellen und Harper [152]). Von entscheidender Bedeutung für die fortwährende Papiernutzung sind die *Affordances*⁹ von Papier. Diese heben sich deutlich von den Eigenschaften typischer Computersysteme ab. Papier ist vor allem einfach greifbar, bewegbar, transportierbar und

⁷ <http://www.epos-ps.com/>

⁸ Im Folgenden wird dies als *Location-Awareness* bezeichnet.

⁹ Der Begriff der *Affordances* wurde für den Bereich HCI von Donald Norman durch sein Werk „The Design of Everyday Things“ geprägt. *Affordances* beschreiben, welche Nutzungsweisen ein Medium, hier Papier, ermöglicht und nahe legt.

annotierbar. Des Weiteren ermöglicht Papier ununterbrochenes kreatives Denken und Skizzieren, besitzt nur geringe formale Nutzungsanforderungen und erfordert einen geringen kognitiven „Workload“ (Oviatt [133]). So liegt ein Schwerpunkt der Papiernutzung vor allem in spontanen, kollaborativen und kreativen Szenarien. Zum Beispiel ist zu beobachten, dass bei dem kreativen Prozess der Erstellung von Modell- und Designentwürfen, häufig erste Skizzen und Entwürfe auf Papier erstellt werden (vgl. Landay und Myers [93]). Durch den informellen Charakter von Stift und Papier und der daraus resultierenden Möglichkeit das eigene mentale Modell ähnlich wiederzugeben, können schnell und ohne großen Aufwand erste Modellskizzen erstellt werden (Nakagawa et al. [120]). Kollaboratives Arbeiten wird durch die allgemeine Tendenz von Papier zur gemeinsamen Nutzung, besonders von großformatigen Papieren, unterstützt (vgl. Dai [35]). Auf den in diesem Zusammenhang wichtigen Aspekt der Erstellung von Modellen auf Basis von Skizzen wird in Kapitel 2.3.3 und 2.3.5 genauer eingegangen. Auch im Bereich des Interface-Designs, im Besonderen im Bereich des partizipativen Systemdesigns, hat sich gezeigt, dass Papier Mock-Ups und Prototypen, die verwendet werden um „Computer zu spielen“, zu einem effektiverem Design beitragen (Bødker et al. [18]). Weitere Vorteile von Papier sind seine Allgegenwärtigkeit und Persönlichkeit (Grasso et al. [52]), die Vertrautheit der Nutzer mit dem Medium (Wellner [168]), die Möglichkeit es von verschiedenen Perspektiven zu Betrachten (Signer und Norrie [156]), sowie die kostengünstige Produktion größerer Interfaces (Yeh et al. [176]). Basierend auf den Affordances von Papier ermöglichen PBUIs sowohl die Anreicherung bestehender papierbasierter Arbeitspraxis als auch die Schaffung gänzlich neuer Anwendungsfelder. Dabei ist sowohl eine ausschließliche Erfassung und nachträgliche Auswertung digitalisierter Informationen, als auch die Gestaltung interaktiver Anwendungen, bei denen eine fortlaufende Analyse der Skizzen über einen verbundenen Computer erfolgt, möglich.¹⁰ PBUIs mit einer Location-Awareness Funktionalität ermöglichen zudem Affordances, die über eine traditionelle Papiernutzung hinausgehen. Es können beispielsweise digitale Beziehungen zwischen verschiedene Papiere erstellt werden, indem diese mit einem Strich verbunden werden. Verschiedene Papiere mit identischen Bereichsinformationen können zudem als „Papiertypen“ betrachtet werden, denen eine bestimmte Semantik zugeordnet ist.

Jedoch besitzt Papier auch eine Reihe von Einschränkungen. Sellen und Harper heben insbesondere symbolische, kostenbezogene und interaktionsbezogene Probleme hervor [152]. So eignet sich Papier aufgrund unökonomischer Aufbewahrungsmöglichkeiten zum Beispiel nur bedingt für die langfristige Verwahrung von Informationen. PBUIs müssen vor allem mit der mangelnden Interaktivität (insb. ein fehlendes dynamisches Feedback) von Papier umgehen. Insgesamt charakterisiert sich Papier durch drei Hauptprobleme:

¹⁰ Im Folgenden wird in diesem Kontext von einer online- und einer offline-Nutzung gesprochen.

1. *Paper ist statisch,*
2. es stellt *kein Feedback außer den erstellten Strichen* zur Verfügung und,
3. mit bestehender Technik können *einmal erstellte Skizzen nicht wieder entfernt* werden.

Besonders aufgrund der letzten zwei Punkte ist eine verlässliche Verarbeitung durch das System unabdingbar. Weitere Herausforderungen sind die Gestaltung eines geeigneten Interaktionssystems mit einer ausreichenden Mächtigkeit, das bestehende papierbasierte Interaktionskonzepte berücksichtigt und den Papiernutzungsprozess nicht unnötig unterbricht (Liao et al. [98]). Die Erstellung von PBUIs erfordert somit eine genaue Betrachtung der Affordances von interaktivem Papier und seiner digitalen Entsprechung, um diese dann komplementär zu kombinieren. Da Interaktion oft verbunden ist mit den sozialen Hintergründen und der Arbeitsumgebung der Menschen (vgl. [152]), ist für die Gestaltung von PBUIs zudem ein genaues Verständnis papierbasierter Arbeitsweisen und des jeweiligen Anwendungskontextes wichtig. Tabelle 2 fasst die wichtigsten Chancen und Herausforderungen papierbasierter Anwendungen zusammen.

Chancen PBUIs	Herausforderungen PBUIs
<ul style="list-style-type: none"> • Vorteile von Papier in informellen, kollaborativen und kreativen Szenarien. • Hohe Nutzerakzeptanz durch Natürlichkeit und Vertrautheit. • Anreicherung bestehender Papiernutzungspraxis sowie neuartige Nutzungsmöglichkeiten von Papier. • Online und offline Nutzung von Papier. • Neue Nutzungsmöglichkeiten durch „Location-Awareness“ Technologien. 	<ul style="list-style-type: none"> • Kombination der digitalen und der physischen Welt, so dass sich die Affordances komplementär ergänzen. • Fehlendes dynamisches Feedback. • Schlechte Veränderbarkeit einmal erstellter Papierskizzen. • Gestaltung eines verlässlichen Systems. • Integration bestehender papierbasierter Interaktionsaspekte und Arbeitspraxis. • Gestaltung eines geeigneten Interaktionskonzeptes.

Tabelle 2: Übersicht der Chancen und Herausforderungen papierbasierter Anwendungen

2.3.3 Grundlagen papier- und skizzenbasierter Modellierung

Verschiedene HCI-Studien zeigen, dass Nutzer es bevorzugen Design- und Modellentwürfe erst mit Stift und Papier zu skizzieren¹¹ und die handgezeichneten Entwürfe dann in computerbasierte Systeme zu übertragen (vgl. [36, 69, 92, 176]). Dies ist auf eine Reihe von Vorteilen von Skizzen gegenüber einer formalen, computerbasierten Modellierung zurückzuführen. Empirische Untersuchungen von Modellierungswerkzeugen haben ergeben, dass diese eine informelle Modellierung stark beschränken (siehe Chen et al. [31] und Damm

¹¹ Mit Skizzieren ist der Prozess des schnellen Zeichnens von Entwürfen gemeint.

et al. [36]). Skizzen sind dagegen nicht auf vorgegebene Elemente beschränkt, sondern können einfach um informelle oder eigene Elemente erweitert werden. Auch führt die unfertige und „rohe“ Erscheinung handgezeichneter Entwürfe zu einem stärkeren Fokus einer Diskussion auf den Inhalt und nicht auf die vordergründige Gestaltung des Entwurfs (Black [16]). Besonders unerfahrene Nutzer tendieren bei rohen Entwürfen eher dazu substantielle Veränderungen beizutragen, als bei computergenerierten Modellen, die den Eindruck eines fortgeschrittenen Entwurfes vermitteln (Winograd [169]). Eine andere Untersuchung zeigt, dass ältere Designer ihre Entwürfe auf Papier erstellen und diese jüngeren Mitarbeiter zu Digitalisierung weiterreichen, da ihnen aktuelle Designsysteme zu komplex sind (siehe Yeh et al. [177]). Auch können Diagramme schneller skizziert als mit Maus und Tastatur erstellt werden (siehe Yeung et al. [179]). Um die beschriebenen Vorteile von Skizzen für die Modellierung technisch zu unterstützen, müssen Systeme die Skizzen automatisch und zuverlässig zu erkennen und sie für eine weitere Verarbeitung zugänglich machen.

2.3.4 Interaktive Papier-Anwendungen

Im Folgenden werden ausgewählte papierbasierte Systeme und Ansätze vorgestellt. Der Schwerpunkt der Auswahl liegt in der Unterstützung kreativer und kollaborativer Szenarien und der Verknüpfung formeller und informeller Informationen. Aktuelle papierbasierte Ansätze finden sich vorrangig in den Bereichen Bearbeitung papierbasierter Formulare, Digitalisierung handschriftlicher Notizen, Anreicherung von Papierdokumenten mit digitaler Bedeutung, Steuerung von Applikationen und Anwendungen und Unterstützung von papierbasierten Skizzen (siehe [20]).

Bearbeitung papierbasierter Formulare

Die digitale Erfassung papierbasierter Formulare ist das typische Anwendungsfeld kommerzieller PBUIs. Ein populäres Beispiel, welches jedoch an Sicherheitsbedenken scheiterte, ist die Einführung von Wahlzetteln auf interaktivem Papier für die Bürgerschaftswahl in Hamburg [13]. Die Wahlberechtigten würden wie gewohnt ihre Stimme auf einem Papierwahlzettel abgeben, dabei jedoch einen speziellen Stift verwenden. Der Stift würde die Stimme über ein auf dem Wahlzettel aufgedrucktes Muster erkennen und das Ergebnis zur automatischen Auswertung direkt an einen Wahlcomputer senden.

Digitalisierung handschriftlicher Notizen

Zu den Standardfunktionen aktueller Digitalstifte gehört mittlerweile oft ein System zur Digitalisierung handschriftlicher Notizen auf Basis einer Schrifterkennung¹². Hervorzuheben

¹² Beispielsweise wird bei Logitechs Digitalstift *io2 pen* die Handschrifterkennungssoftware *MyNotes* als Demo-Version mitgeliefert.

ist in diesem Zusammenhang das *LiveScribe*-Konzept [105]. Dort werden nicht nur handschriftliche Notizen digital erfasst, sondern der Stift zeichnet während des Schreibens zusätzliche Audiodaten auf und verknüpft diese zeitlich mit den Notizen. Durch Markieren ausgewählter Notizen kann eine zugeordnete Audioaufzeichnung abgerufen werden.

Anreicherung von Papierdokumenten mit digitaler Bedeutung

Im Bereich der Anreicherung von Papier mit einer digitalen Bedeutung, finden sich die meisten Forschungsansätze. Deren Fokus liegt meist auf Dokumentenarbeit. Ein sehr umfassender Ansatz ist das Konzept der *Paper Augmented Digital Documents (PADD)* [58] in Verbindung mit dem Interaktionskonzept *PaperCraft* [98]. In PADD können Nutzer Dokumente alternativ an einem Computerbildschirm oder mit einem Digitalstift auf Papier bearbeiten. Wenn der Stift mit dem System synchronisiert wird, erfolgt ein automatischer Abgleich der Dokumente. Über das Interaktionskonzept *PaperCraft* können, durch definierte Gesten, Änderungen am Dokument vorgenommen werden. Das System *Print-n-Link* [127] unterstützt den Nutzer beim Lesen gedruckter wissenschaftlicher Publikationen. Abhängig vom jeweiligen Nutzungskontext (online oder offline) stehen verschiedene Interaktionsmöglichkeiten zur Verfügung. Durch das Antippen von Literaturreferenzen im Text kann der Nutzer über ein Bluetooth-Headset oder einen Bildschirm Informationen zu dem Autor erhalten oder eine Kopie der markierten Referenz anfordern. Weitere Ansätze sind die Anreicherung papierbasierter Touristeninformationen um digitale Funktionen [52, 125] und die Verknüpfung papierbasierter und digitaler Terminkalender [5].

Steuerung von Applikationen und Anwendungen

Durch die Verfügbarkeit von Digitalstiften mit Location-Awareness sind verstärkt Ansätze entstanden, die eine Steuerung von Anwendungen über Stift und Papier möglich machen. In dem System *PaperPoint* [156] können über selbst zusammenstellbare Navigationsinterfaces PowerPoint-Präsentation gesteuert werden. Zusätzlich bietet das System die Möglichkeit interaktive physische Repräsentationen der Präsentationsfolien zu erstellen. Wenn diese mit einem digitalen Stift annotiert werden, springt die Präsentation zu der entsprechenden Folie und blendet die erstellten Annotationen digital ein. Andere Ansätze ermöglichen die Steuerung eines Meeting-Raums (*POEMS* [73]), eines Media-Centers (*pRemote* [67]), eines Videorecorders [86] oder vernetzter Haushaltsanwendungen allgemein [87].

Unterstützung von papierbasiertem Sketching

Bisher war die Erstellung stiftbasierter Skizzen ausschließlich auf elektronische Sketching-Systeme beschränkt (siehe Kapitel 2.3.5). Aufkommende Ansätze beschäftigen sich mit der Gestaltung von PBUIs für das Design auf Basis von Skizzen, vorrangig in den Bereichen CAD und Interface-Design. So bietet *ModelCraft* [159] die Möglichkeit zur Modellierung von

3D-Objekten mit interaktivem Papier. Dazu wird aus interaktivem Papier ein 3D-Modell gebastelt, auf dem dann Veränderungen und Verhalten des Modells eingezeichnet werden können. Die Annotationen auf dem Papiermodell werden anschließend in einem CAD-Programm auf das Ursprungsmodell angewendet. Ein anderer Ansatz im CAD-Umfeld ist die Programmierung industrieller Roboter durch Annotation auf einem gedruckten CAD-Modell [140]. Das von Fly Fusion für Kinder entwickelte, kommerzielle *Fly Pentop* Lernwerkzeug [95] ermöglicht die Modellierung eigener Papierinterfaces für Spiele und Lernanwendungen. Dazu können Kinder sowohl vorgedruckte Interfaces mit selbstgezeichneten Elementen erweitern, als auch für vorgegebene Anwendungen eigene Interfaces zeichnen. Im Bereich papierbasierter Modellierung von Diagrammen finden sich bisher keine Ansätze. Ein allgemeines Sketching-System mit Fokus auf kollaboratives Brainstorming ist *Diamond's Edge* [15]. In *Diamond's Edge* sitzen alle Teilnehmer gemeinsam an einen Tisch mit einer interaktiven Digitalanzeige¹³. Jeder Teilnehmer hat zudem einen privaten interaktiven Skizzenblock vor sich liegen. Auf dem privaten Block können Skizzen erstellt, diese dann digital auf den kollaborativen Tisch übertragen und dort bearbeitet werden. Weitere skizzenbasierte Systeme, die interaktive Papiertechnologie verwenden, sind der „Shared Design Space“ *Office of Tomorrow* und das darauf aufbauende Grafikbearbeitungssystem *Conoto* [61]. Jedoch findet bei beiden keine echte Papiernutzung statt, sondern das interaktive Papier wird als preisgünstige Alternative zur Tablet-Technologie verwendet.

2.3.5 Übersicht elektronischer skizzenbasierter Designwerkzeuge

Eng verbunden mit PBUIs ist die Forschung zu elektronischen Sketching-Systemen auf Basis von „low-fidelity“-Visualisierungen, die zu einem großen Teil auf E-Whiteboard und Tablet-PC Technologie aufsetzen. Verschiedene Ansätze untersuchen die Anreicherung von Modellierungswerkzeugen um skizzenbasierte Funktionen, sowie die Erstellung von Werkzeugen, die es Endbenutzern ermöglichen Interfaces zu skizzieren und sogenannte „low-fidelity“ Prototypen zu verwalten. Der Fokus der Ansätze liegt vor allem auf der Gestaltung von Eingabekonzepten, der Erkennung von Gesten und der Visualisierung und Formalisierung skizzierter Prototypen. Dabei stehen jedoch meist technische Aspekte im Vordergrund. Elektronische Sketching-Systeme werden auch als „informelle“ Designwerkzeuge bezeichnet, da sie die Mehrdeutigkeit und Informalität von Skizzen unterstützen (vgl. Yeung et al. [179]). Sie grenzen sich von PBUIs insofern ab, als das sie im Gegensatz zu Papier ein direktes Feedback unterstützen und so einfacher dynamische Interaktionskonzepte anbieten können. Es gehen jedoch viele Vorteile von Papier (z.B. Kostenvorteile, Haptic und Mobilität) verloren.

¹³ Der digitale Tisch basiert auf einer Plexiglasscheibe mit darunterliegendem interaktive Papier sowie einer Bildprojektion von oben.

Skizzenbasierte Modellierungswerkzeuge

Der Fokus skizzenbasierter Modellierungswerkzeuge liegt beinahe ausschließlich auf der Anreicherung von UML-Modellen für professionelle Entwickler. Einer der frühesten Ansätze ist *Knight* [36]. *Knight* ermöglicht die kollaborative Modellierung von UML-Klassendiagrammen auf einem großen elektronischen Whiteboard. Über Gesten, die mit einem Stift erstellt werden, können Nutzer sowohl formelle als auch informelle Elemente skizzieren und miteinander kombinieren. Über einen speziellen „Freihand“-Modus können bewusst informelle Annotationen gezeichnet werden. Um eine intuitive Interaktion zu ermöglichen, verwendet *Knight* „compound gestures“ und „eager recognition“. „Compound gestures“ bedeutet, dass zeitlich und räumlich nahestehende Gesten kombiniert werden, um ein Element zu erstellen. Dies ermöglicht einen iterativen Aufbau des Modells. „Eager recognition“ versucht Gesten während des Zeichnens zu erkennen und dem Benutzer ein direktes Feedback zu geben. Formale Aspekte werden bei *Knight* direkt erkannt und durch formale, computergezeichnete Repräsentationen ersetzt. Ein ähnliches kommerzielles System bietet Ideogram mit *Pervasive UML* [77]. Das System *Tahuti* [62] bietet darüber hinaus die Möglichkeit einer integrierten Erkennung sowohl von Schrift, als auch von Zeichnungen. *SUMLOW* [31] ist ein weiteres E-Whiteboard-basiertes System zur kollaborativen Erstellung verschiedener UML-Modelle (Klassen-, Use-Case- und Sequenzdiagramme) durch Skizzen. Im Gegensatz zu Systemen wie *Knight* und *Tahuti* wird jedoch bei formal erkannten Elementen das ursprüngliche handgezeichnete Aussehen soweit wie möglich erhalten. Zudem erfolgt, anders als bei *Knight*, eine progressive und keine sofortige Formalisierung der Elemente („lazy recognition“). Einen ersten stark prototypischen Ansatz zur Modellierung von Geschäftsprozessen über das Skizzieren von UML-Aktivitätsdiagrammen liefern Donaldson und Williamson [39]. Die Erweiterung *MaramaSketch* [56] bietet durch seine Integration in das Meta-Diagramm-Werkzeug *Marama* [57] Unterstützung für beliebige, mit *Marama* erstellte Diagrammart. Da die Formen der zu erkennenden Elemente nicht vorab bekannt sind, ist jedoch vor jeder Nutzung von *MaramaSketch* für eine neu erstellte Diagrammart ein Training der Erkennungsroutine nötig.

Skizzenbasiertes Interfacedesign

Einer der frühesten und bekanntesten Ansätze, *SILK* [93], ermöglicht Interfacedesignern mit einem elektronischen Stift schnell Interfaceentwürfe zu skizzieren. Während des Zeichnens erkennt *SILK* automatisch einzelne Interfaceelemente und ermöglicht ein interaktives Testen der erkannten Interfaces. Ein ähnlicher Ansatz für den Bereich des Web-Interface Designs ist *DENIM* [72]. Weitere Ansätze, die eine schnelle Erstellung von Interfaceprototypen auf Basis von Skizzen ermöglichen sind *FreeForm* [142] und *PatchWork* [81].

2.3.6 Interaktionskonzepte papierbasierter Sketching-Anwendungen

In Kapitel 2.3.2 wurde auf den Unterschied von PBUIs zu typischen GUIs eingegangen und beschrieben, dass die Gestaltung eines geeigneten Interaktionskonzeptes (Eingabe und Feedback) eine der Kernherausforderungen papierbasierter Anwendungen ist. Auf einige Gestaltungsansätze wurde bereits bei der Beschreibung papier- und skizzenbasierter Systeme in Kapitel 2.3.4 und 2.3.5 eingegangen. Im Folgenden werden bestehende Interaktionsansätze aus beiden Bereichen, soweit anwendbar auf papierbasierte Systeme, einzeln dargestellt. Meist findet sich indessen eine multimodale Kombination mehrerer Ansätze.

Eingabegestaltung

In den meisten PBUIs erfolgt die Eingabe ausschließlich über einen Stift auf einem interaktiven Papier der Größe DIN A4 oder kleiner. Ein Vorteil von Papier ist jedoch, dass auch großformatige Papier-Interfaces kostengünstig produziert und flexibel genutzt werden können. Großformatige Papiere unterstützen auch gut kollaborative Aspekte durch die Möglichkeit mit mehreren Stiften gleichzeitig zu skizzieren. Erste Untersuchungen zur Eingabe auf großformatigem Papier, mit Fokus auf den visuellen Kontext, die Mobilität, die Kollaboration und auf spezifische Interaktionsformen, finden sich im Konzept *Interactive GigaPixel Prints* [176, 177], sowie in den Systemen *Office of Tomorrow* [61] und *Diamond's Edge* [15]. Ein weiteres Eingabekonzept sind kleine greifbarer („tangible“) Eingabeobjekte, insbesondere Post-Its. Dieses Konzept der greifbaren Eingabeobjekte findet sich bei [61, 79, 82, 110]. Greifbare Eingabeobjekte können über physische oder digitale Verbindungsgesten einem Objekt zugeordnet werden. Durch Antippen oder Beschriften der Eingabeobjekte können dann „ferngesteuert“ verschiedenen Funktionen und Modi ausgelöst werden. Andere Ansätze erforschen die Verknüpfung einer papierbasierten Eingabe mit natürlicher Spracherkennung [110] oder die Visualisierung nicht-sprachlicher vokaler Eingabe [65].

Da Papier kein dynamisches Feedback bietet, ist eine Auswahl von Befehlen und Modi nicht wie in elektronischen Stiftsystemen über ein interaktives Kontextmenü möglich. Typische Ansätze von PBUIs sind Papiergesten und greifbare Eingabeobjekte. Erste Untersuchungen zur Gestaltung alternativer Befehlseingabekonzepte, zum Beispiel über integrierte Knöpfe am Stift oder die Verwendung mehrerer Stifte, finden sich bei Li et al. und Liao et al. [97, 98].

Ein Kernaspekt der Eingabegestaltung PBUIs ist nach Frankish et al eine gute Handschrift- und Gestenerkennung [47]. Bestehende Ansätze unterscheiden sich dahingehend, ob sie die in Kapitel 2.3.5 bereits beschriebenen Konzepte der „compound gestures“ und der „eager“ oder der „lazy recognition“ unterstützen und ob ein Training der Erkennungsroutine nötig ist. In Systemen, in denen eine integrierte Erfassung von Text- und Gestenelemente nötig ist (z.B. diagrammbasierte Sketching-Systeme), ergibt sich zudem die Schwierigkeit Text- und Diagrammskizzen zu differenzieren (Patel et al. [137]). Die am weitesten verbreitete

Erkennungsroutine für Gesten ist der *Rubines*-Algorithmus [146]. Ein einfacher und schneller Ansatz für prototypische Umgebungen ist der *\$I-Recognizer* [170].

Feedbackgestaltung

Feedback kann in papierbasierten Anwendungen entweder direkt über den Stift (stiftbezogenes Feedback) oder über zusätzliche Feedbackgeräte durch einen mit dem Stift verbundenen Computer (allgemeines Feedback) erfolgen. Bei der Nutzung von allgemeinem Feedback ist der Trade-off zwischen der Flexibilität von Papier und der Mächtigkeit zusätzlicher Feedbackmechanismen zu beachten (siehe Yeh et al. [177]). Grundlegend unterscheidet sich mögliches Feedback (in beiden Bereichen) in die Modalitäten auditiv (Sprache und Ton), visuell und haptisch. In welcher Ausprägung welches Feedback nötig ist, hängt von der jeweiligen Anwendung und dem Anwendungskontext, sowie von Aspekten wie der Zuverlässigkeit des Systems (vgl. Liao et al. [98]), ab. Zudem kann das verfügbare Feedback je nach Kontext variieren (siehe z.B. *Print-n-link* [127]).

		Feedback type		
		Discovery	Status	Task
Modality	Visual	Command direction	Mode, Recognition	Page-level location
	Tactile	Boundary warning	Ambiguity detection	
	Sound	Boundary crossing		
	Voice	Command name		Document-level location

Tabelle 3: Gestaltung von stiftbasiertem Feedback [99]

Stiftbezogene Feedbackmechanismen (z.B. Vibration und LEDs) sind in verschiedenen kommerziellen Digitalstiften bereits integriert, jedoch fehlt die Möglichkeit diese über entsprechende Softwareschnittstellen anzusteuern. Eine Ausnahme bildet das beschriebene *Fly Pentop* System [95], welches über ein integriertes Sprachinterface Text auf dem zugehörigen Stift abspielen kann. Wie Vibration und LEDs dennoch als stiftbezogenes Feedback eingesetzt werden können, zeigen Liao et al. [99] prototypisch. Zudem kategorisieren sie mögliches Feedback für PBUIs („feedback types“) und untersuchen, wie diese geeignet durch verschiedene stiftbezogene Feedbackarten („modalities“) unterstützt werden können (siehe Tabelle 3).

Allgemeines Feedback ist bei PBUIs sinnvoll, wenn Wiedergabe komplexer Informationen stattfinden soll. Die meistens Systeme, insbesondere im Bereich Skizzen- und

Dokumentenbearbeitung, verwenden Bildschirme nahestehender Computer um erstellte Skizzen und Annotationen zu visualisieren. Ein weitergehendes Visualisierungskonzept ist die Verknüpfung von Papierskizzen mit einer Digitalanzeige durch die Überlagerung des Papiers mit einem Beamer oder durch die Verwendung von Tabletop-Konzepten (siehe [15, 61, 79]).

Ein Feedbackaspekt skizzenbasierter Designwerkzeuge ist die Visualisierung formal erkannter Skizzenbestandteile. Die Spanne der Visualisierungsansätze reicht von der bloßen Wiedergabe der gezeichneten Skizzen, zum Teil mit einer zusätzlichen Kennzeichnung formal erkannter Elemente (z.B. bei *SUMLOW* und *FreeForm*), bis zu einer vollständigen Ersetzung der Skizzen durch computergezeichnete Repräsentationen (z.B. bei *Knight* und *Tahuti*). Hundhausen benennt diese beiden Ansätze als *high* und *low fidelity* Visualisierung und beschreibt zwischen diesen beiden Extremen ein Kontinuum verschiedener Formalisierungsgrade [75]. Der Prozess der Transformation handgezeichneter Skizzen in formalere und gleichmäßigere Darstellung wird als *Beautification* bezeichnet [179].

2.4 Zusammenfassung und Fokus

In Kapitel 2.1 wurde die Bedeutung eines flexiblen Managements von Geschäftsprozessen beschrieben (Kapitel 2.1.1) und dargestellt, wie serviceorientierte Architekturen diese Flexibilität technisch unterstützen können (Kapitel 2.1.2). Für professionelle Modellierer und Entwickler existieren daher, wie in Kapitel 2.1.3 gezeigt, im Kontext von SOA verschiedene Modellierungsnotationen (z.B. BPMN und UML) und Modellierungswerkzeuge (z.B. WebSphere Business Modeler und ARIS Business Architect). Wie in der Motivation bereits angedeutet (Kapitel 1.1), fanden bei der Konzeption dieser Werkzeuge endbenutzerorientierte Gestaltungsaspekte, wie die Gestaltung endbenutzergerechter Interfaces, eine Einbettung in einen „Gentle Slope of Complexity“ oder eine Meta-Domänenorientierung (siehe Kapitel 2.2.2), keine Berücksichtigung. Kapitel 2.2.3 zeigte Forschungsansätze zur Prozessmodellierung durch Endbenutzer (z.B. EUSC, SiSeOr oder VINCA). Diese bieten eine gute Unterstützung für einfache und wenig umfangreiche Prozesse mit wenigen Elementen sowie für eine Modellierung durch einzelne Benutzer. Kollaborative Entwicklungs- bzw. Modellierungstätigkeiten, wie sie typischerweise bei Endbenutzern zu beobachten sind (siehe Kapitel 2.2.2), werden jedoch weder von endbenutzerorientierten, noch von professionellen Orchestrierungssystemen unterstützt. Auch informelle Aspekte, wie sie zu Beginn einer Prozessmodellierung zu finden sind (siehe Kapitel 2.1.2), lassen sich, aufgrund des starken Fokus der Systeme auf die Orchestrierung, nicht abbilden. Schlussendlich können auch komplexere Prozesse nur sehr bedingt modelliert werden.

Kapitel 2.3.2 und 2.3.3 zeigten, dass die Nutzung von Papier und Skizzen, aufgrund von Flexibilitätsvorteilen, einer natürlichen Nutzung und einer guten Unterstützung kreativer Aspekte, Vorteile für kollaborative und kreative Nutzungsprozesse bietet. Jedoch fanden sich

im Bereich papierbasierter Systeme keine Ansätze bezüglich des Einsatzes von Papier für Modellierungsaspekte oder für das End-User Development (siehe Kapitel 2.3.4). Kapitel 2.3.5 zeigte, dass es im Bereich elektronischer Skizzensysteme losgelöst voneinander sowohl Ansätze zur professionellen Modellierung durch Skizzen, als auch zur Unterstützung von Endbenutzern bei Designaktivitäten gibt. Jedoch fanden sich auch hier keine Ansätze einer skizzenbasierten Modellierung durch Endbenutzer. In Bezug auf die in Kapitel 1.2 beschriebenen Ziele stellt sich daher als primäre Forschungsfrage:

Wie kann eine papierbasierte und endbenutzerorientierte (kollaborative) Modellierung von Geschäftsprozessen in serviceorientierten Architekturen gestaltet werden?

Wie Kapitel 2.3.2 und 2.3.6 zeigten, bedürfen papierbasierte Interfaces grundlegend anderer Interaktionskonzepte als „typische“ Applikationen. So besitzt Papier zum Beispiel keine direkte Möglichkeit interaktive Inhalte anzuzeigen und auch das Entfernen einmal erstellter Skizzen ist oft nicht möglich. Interaktionskonzepte bestehender (endbenutzergerechter) Modellierungssysteme (Kapitel 2.1.3 und 2.2.3), wie zum Beispiel interaktive Assistenten oder ein Syntax-Highlighting, lassen sich daher nicht oder nur begrenzt übernehmen. Der Schwerpunkt der Arbeit liegt daher auf folgendem Teilaspekt:

Wie kann ein Interaktionskonzept (insb. die Gestaltung der Papierinterfaces und des Feedbacks) für eine endbenutzergerechte Modellierung, unter Berücksichtigung bestehender Interaktionsansätze und der Chancen und Herausforderungen von interaktivem Papier, gestaltet werden?

Ein Kernaspekt eines Modellierungswerkzeuges, der auch die Interaktion maßgeblich beeinflusst, ist das verwendete Modellierungskonzept, insbesondere die Notation und deren Metaphorik und Symbolik. Dabei ist besonders zu berücksichtigen, dass die Affordances von interaktivem Papier, wie in Kapitel 2.3.2 beschrieben, spezifische Anforderungen an eine Modellierungsnotation stellen. So legt Papier unter anderem eine informelle Nutzung, nicht jedoch eine hohe Veränderbarkeit nahe. Daher sollten Modellierungselemente zum Beispiel einfach zu zeichnen sein und sich klar voneinander unterscheiden. Da sich zeigte, dass zwar skizzenbasierte (z.B. KNIGHT, siehe Kapitel 2.3.5), aber bisher keine rein papierbasierten Modellierungsansätze existieren, stellt sich als weiterer wichtiger Teilaspekt die Frage:

Wie kann ein Modellierungskonzept (insb. eine Modellierungsnotation) für eine papierbasierte, endbenutzerorientierte Modellierung im Kontext von BPM und SOA gestaltet werden?

3 Vorstudien: Modellierungspraxis

In der Literatur finden sich zwar mehrere Ansätze einer Service-Orchestrierung durch Endbenutzer (z.B. Scufl oder VINCA, siehe Kapitel 2.2.3), jedoch beinhalten diese keine konkreten Studien, wie eine (kollaborative) serviceorientierte Modellierung mit Endbenutzerbeteiligung in der Praxis stattfindet. Auch existieren bisher kaum dokumentierte empirische Ergebnisse zur Einführung von BPM mit SOA in der Praxis (siehe Kapitel 2.1.2). Um einen Einblick in die konkrete Orchestrierungs- bzw. Modellierungspraxis zu erhalten und um Designentscheidungen für die zu entwickelnde Lösung treffen zu können, wurde eine zweistufige Voruntersuchung mit Mitarbeitern aus zwei Unternehmen durchgeführt. Im ersten Schritt wurde in einer Fallstudie analysiert, wie in der Praxis die Erstellung von Prozessen im Kontext serviceorientierter Architekturen stattfindet und inwiefern Endbenutzer in diese einbezogen werden. Im zweiten Schritt wurde ein Prozessdesign-Workshop mit Endbenutzern aus einem Unternehmen durchgeführt. Dort wurden die Akzeptanz und das Verständnis einer serviceorientierten Prozessmodellierung sowie Gestaltungsansätze einer kollaborativen papierbasierten Modellierung untersucht.

3.1 Fallstudie: BPM in SOA in der Praxis

Untersuchungen zu BPM in der Praxis gestalten sich schwierig, da kaum dokumentierte Berichte über den organisatorischen Ablauf der Prozessmodellierung im SOA-Kontext existieren. Um ein genaues Verständnis zu erhalten, wie im SOA-Kontext Geschäftsprozesse in der Praxis erstellt und umgesetzt werden, welche Rolle Endbenutzer dabei einnehmen und wie ein solcher Prozess unterstützt werden kann, wurde zu Beginn eine Fallstudie in einem Unternehmen aus dem Finanzsektor durchgeführt. Neben der Analyse der aktuellen Arbeitspraxis wurde untersucht, auf welche Resonanz die Idee und erste Gestaltungsansätze eines papierbasierten Modellierungsinterfaces in der Praxis stoßen und welche Chancen und Herausforderungen für die Gestaltung eines solchen Systems gesehen werden.

3.1.1 Konzeption und Methodik

Die Durchführung der Fallstudie erfolgte auf Basis qualitativer Forschungsmethodiken (Kvale [91]). Als Untersuchungsobjekt wurde die Danske Bank, ein Unternehmen des Finanzsektors, gewählt, zu dem im Bereich BPM und SOA bereits mehrere Studien existierten (siehe [21-23]). Aufbauend auf diesen ersten Erkenntnissen wurde in einem semi-strukturierten Interview die Modellierungspraxis genauer analysiert. Als Interviewpartner wurde ein Entwickler der Workflow-Abteilung ausgewählt, der durch die organisatorische Betreuung der Workflow-Entwicklung an der Schnittstelle zwischen Anwendern und Entwicklern stand und zudem ein gutes Organisationswissen besaß (siehe weiter 3.1.2). Aufgrund der Entfernung des Unternehmens fand das Interview in einem Telefongespräch statt. Die Form

des semi-strukturierten Interviews wurde gewählt, um einen detaillierten Einblick in die Arbeitswelt des Teilnehmers zu erlangen und ihn dabei nicht zu stark zu beeinflussen. Der Interviewleitfaden (siehe Anhang A) orientierte sich an den Gestaltungsvorschlägen von Merton und Kendall¹⁴ [113]:

- Nicht richtungsweisend (Keine Beeinflussung des Befragten)
- Ausführliche Darstellung der persönlichen Meinung des Befragten bezüglich der Interviewthemen
- Erfassen einer breiten Palette an Facetten des Interviewthemas
- Emotionale Tiefe der Interviewstatements und -situationen

Um eine tiefergehende wissenschaftliche Fundierung zu erhalten, war ursprünglich die Durchführung weiterer Fallstudien geplant. Dies scheiterte jedoch an der Verfügbarkeit geeigneter Untersuchungspartner. So konnte eine weitere vorbereitete Untersuchung letztlich nicht stattfinden, da keine Kooperationsbereitschaft der zuständigen Fachverantwortlichen bestand. Allgemein erwies es sich als schwierig, geeignete Untersuchungspartner zu finden. Dies begründet sich vor allem in der bisher „geringe Umsetzung von BPM mit SOA in der Praxis“ (vgl. Krafzig et al. [88]).

3.1.2 Untersuchungsgegenstand

Der Interviewte arbeitete zum Zeitpunkt des Interviews seit drei Jahren als *Infrastructure Developer* und als *Solution Architect* in der Workflow-Abteilung (eine Unterabteilung der IT) bei der Danske Bank. Seine primäre Aufgabe war die Unterstützung der Entwicklung und der organisatorischen Umsetzung von Workflows. Als Infrastructure Developer entwickelte er neue Funktionalität und Erweiterungen für die Prozessausführungsumgebung und das Modellierungswerkzeug. Als Solution Designer erstellte und verwaltete er Anforderungen für Geschäftslösungen gemäß eines spezifizierten *Business Modells* (siehe Kapitel 3.1.3). Zusätzlich leistete er anderen Projekten Unterstützung bei deren Entwicklung von Workflows.

Die Danske Bank ist ein aktienelistetes Unternehmen aus dem Banken und Versicherungssektor mit circa 20.000 Mitarbeiter. Sie betreibt bereits seit 2001 eine serviceorientierte Entwicklung und alle neuen Projekte werden – soweit sie einen Prozessaspekt besitzen – prozessorientiert realisiert. Bis Anfang 2008 wurden circa 2000 einzelne Services sowie circa 80 ausführbare Prozesse erstellt. Diese teilen sich in rund 60% einfache und in jeweils 20% mittelkomplexe und komplexe Prozesse auf. Diese entwickelten Prozesse deckten jedoch erst einen kleinen Teil der betrieblichen Prozessabläufe ab. So schätzte der Interviewte, dass zwei Jahren später bis zu 200 ausführbare Prozesse existieren werden. Zurzeit des Interviews wurden im Schnitt alle drei Monate bis zu fünf neue Prozesse

¹⁴ Übersetzung aus dem Englischen durch den Autor.

erstellt. Zudem werden bestehende Prozesse laufend iterativ weiterentwickelt. Dies ist nötig, da aufgrund zeitlicher, technischer (fehlende Services) und kostentechnischer Restriktionen oft nur ein Teil der idealisiert beschriebenen Prozesse direkt umgesetzt werden kann. Verbleibende Aktivitäten werden manuell im Back-Office erledigt und im Laufe der Zeit schrittweise automatisiert. Ein weiterer häufiger Grund für die Anpassung von Prozessen sind neue oder veränderte fachliche Anforderungen.

3.1.3 Ergebnisse und Auswertung

Das auf Englisch geführte Interview wurde auf Basis von Stichpunkten transkribiert und unter Berücksichtigung der bereits existierenden Erkenntnisse anschließend konsolidiert und analysiert. Im Fokus der Auswertung standen folgende Fragestellungen, die anschließend in eben dieser Reihenfolge beantwortet werden:

- Welches allgemeine Prozessverständnis existiert im Unternehmen?
- Welche Personen sind in welcher Form am BPM-Entwicklungsprozess beteiligt?
- Gibt es ein Vorgehensmodell, und aus welchen Elementen setzt sich eine BPM-Lösung zusammen?
- Wie sieht der Ablauf der Prozesserstellung (inklusive Modellierung) im Detail aus?
- Wie sind Endbenutzer in den Gesamtprozess einbezogen?
- Wie schätzt der Befragte die Verwendung eines papierbasierten Modellierungsinterfaces in der Praxis ein?

Prozessverständnis

In der Danske Bank existiert, unabhängig von den bestehenden, ausführbaren Prozessen, eine starke Prozessorientierung. Dies äußert sich dahingehend, dass alle Prozesse im Unternehmen, auch nicht implementierte, modelliert werden sollen. Der Fokus liegt jedoch ausschließlich auf geschäftsrelevanten Abläufen. „Die kleinen alltäglichen Dinge“ werden nur unterstützt, wenn sie auch für den Geschäftsablauf relevant sind. Ein Prozess in der Danske Bank besteht grundlegend aus *Aktivitäten (Tasks)*, die kontextabhängig ausgeführt werden und in einer bestimmten Reihenfolge miteinander verknüpft sind. Die modellierten Aktivitäten lassen sich unterscheiden in *automatisch* ausgeführte, *manuell* ausgeführt und *user experienced* Aktivitäten. Letztere sind manuelle Aktivitäten, die durch ein grafisches Benutzerinterface unterstützt werden. Eine Aktivität stellt jeweils eine abgeschlossene Handlung dar. Das bedeutet, einzelne Schritte einer abgeschlossenen Handlung werden nicht im Prozessmodell dargestellt, sondern als Inhalt der Aktivität beschrieben. Es existieren zudem zwei verschiedene Prozessstufen: Während der Spezifikation wird ein logisches Prozessmodell erstellt, dem technische Details, wie etwa eine Fehlerbehandlung, zur Ausführbarkeit fehlen. Dieses als *Solution Process Flow (SPF)* (Abbildung 9) visualisierte Modell, muss für die

Prozessimplementierung in ein physisches Prozessmodell, das die Grundlage für die Ausführung darstellt und in BPEL vorliegt, transformiert werden.

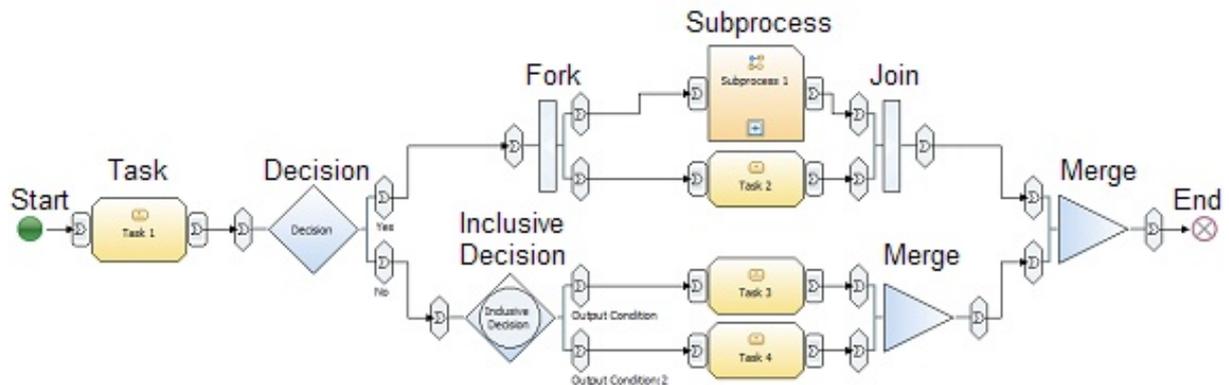


Abbildung 9: SPF-Notation [85]

Am Entwicklungsprozess beteiligte Personen

An der Prozessentwicklung sind vor allem vier Rollen beteiligt:

- *Business User*
- *Business Developer*
- *Solution Architect*
- *Technical Architect*

Um diese im Feld EUD einordnen zu können, werden die Rollen den von Nardi vorgeschlagenen Endbenutzertypen ([122], siehe S. 18) zugeordnet. Business User sind Endbenutzer aus den Fachabteilungen und entsprechen somit in der Regel dem End-User von Nardi. Business Developer sind Fachvertreter in den IT-Abteilungen, die ein hohes domänenspezifisches Wissen besitzen. Generell soll ein Business Developer daher aus der Anwendungsdomäne kommen und dann in die IT wechseln, um dort die Anwender zu vertreten und das Wissen aus der Anwendungsdomäne in die IT-Abteilung einzubringen. Sie helfen der IT bei der Analyse der Anforderungen und der Prozessmodellierung und stellen bei „Rückfragen“ den Kontakt zu den Business Usern her und fungieren so als Schnittstelle. Der Business Developer ist somit eine Art „erfahrener Benutzer“ und entspricht Nardis Typ des Local Developers. Ein Solution Architect begleitet den gesamten Prozess von der Analyse bis zur abgeschlossenen Spezifikation aus Sicht der IT und ist für die fachliche Korrektheit der Lösung verantwortlich. Er sorgt vor allem für die Definition der Anforderung und die Erstellung der Modelle und ist als Professional Programmer einzuordnen. Zudem ist er eine Art „Coach“ für den Business Developer, mit dem er gemeinsam die Anforderungen erfasst und ihn in die Prozessmodellierung einführt. Der Technical Architect ist eine Rolle, die in naher Zukunft geschaffen werden soll, um eine Trennung der fachlichen und der technischen

Prozessverantwortung zu erreichen. Seine Aufgabe soll die technische Korrektheit der Lösung sein. Demnach muss er ebenfalls als Professional Programmer eingeordnet werden.

Entwicklungsmodell und Prozessbestandteile

Für die Prozessentwicklung existiert im Unternehmen ein Vorgehensmodell (*Development Model*), das detailliert beschreibt, wie die Entwicklung von Geschäftsprozessen von der Idee bis zur fertigen Lösung abläuft. Das Vorgehensmodell unterteilt die zu entwickelnde Lösung in die vier Bereiche *Prozess*, *Funktionalität*, *Front-End* und *Daten*, und beschreibt deren Aufbau und Bedeutung. Dies entspricht einem ganzheitlichen Modellierungsansatz (siehe Kapitel 2.1.3). Gemäß dem verfolgten Entwicklungsparadigma „*erst die Anforderungen, dann die Entwicklung*“ steht die Analyse der Anforderungen immer an erster Stelle. Daher müssen zu Beginn alle Anforderungen analysiert und erfasst werden. Die informelle Beschreibung der Lösung erfolgt anhand der Anforderungen in mehreren *System Use-Cases (SUC)*. Diese beschreiben jeweils die Anforderungen einzelner Aktivitäten und Lösungselemente. Je nach Grad der Spezifizierung verknüpfen sie zudem die erstellten Modelle der verschiedenen Lösungsbereiche (Prozess, Interface, Funktionalität, Daten) untereinander und mit den jeweiligen Anforderungen. Elemente eines SUCs sind Name, Typ und informelle Beschreibung der Aktivität, sowie dessen Operationen, Attribute, Vorbedingungen und Ergebnisse (siehe Anhang B). Detailliertere SUCs dienen zudem als Grundlage für die Erstellung neuer Services. Die verschiedenen SUCs eines Geschäftsprozesses werden in einem einfachen Textdokument, der *Process Solution Description (PSD)*, zusammengefasst. Diese enthält alle Elemente der Prozesslösung, sowie deren Anforderungen.

Am Anfang der formalen Spezifikation steht die Modellierung eines formalen SPF-Prozessmodells. Dieses enthält detaillierte Informationen über den Prozessablauf, wie Aktivitäten, Verbindungen und Nachrichtenaustausch. Darauf aufbauend werden die benötigte Funktionalität bestimmt, User-Interfaces erstellt und die drei Bereiche miteinander verknüpft. Nachgelagert werden Datenaspekte, wie etwa die Speicherung in Tabellen, behandelt. Im Gegensatz zu integrierten, ganzheitlichen Modellierungsansätzen, wie zum Beispiel ARIS [148], werden die Modelle nicht in einem System integriert. Stattdessen existieren für jeden der einzelnen Lösungsbereiche individuelle Modelle, die lediglich textuell in der PSD zusammengeführt werden. Die SPF-Prozesse werden im *WebSphere Business Modeler*, die Funktionalität im *Rational Software Modeler* und Interfaces in *Giminy*¹⁵ modelliert.

¹⁵ Ein von der Danske Bank selbstentwickeltes Werkzeug.

Detaillierter Ablauf der Prozesserstellung

Die Erstellung von Prozessen lässt sich grob in die Phasen Analyse, Spezifikation und Implementierung einteilen. Der vereinfachte Ablauf, sowie die involvierten Personen und resultierenden Dokumente sind als eEPK vereinfacht in Abbildung 10 dargestellt.

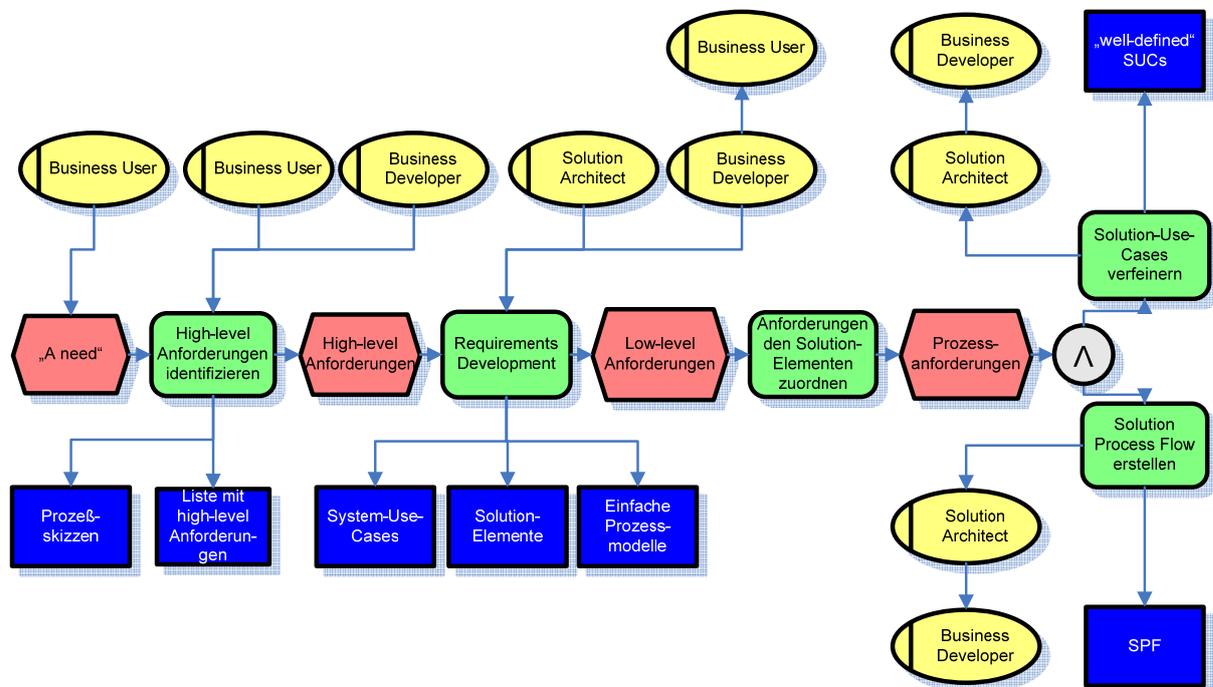


Abbildung 10: Identifizierter Ablauf der Erstellung von BPM-Lösungen

In der *Analysephase* werden alle Anforderungen der Lösung spezifiziert. Ausgehend von einem „Bedarf“ der Anwender werden zu Beginn der Prozesserstellung gemeinsam von einem Business Developer und von Business Usern strukturierte und messbare, abstrakte Anforderungen definiert und bereits grob die einzelnen Schritte des Prozesses beschrieben. In dieser Phase stehen informelle Dokumente (informelle Prozessskizzen und Anforderungsbeschreibungen) und nicht Modelle im Vordergrund. Ergebnis dieser Phase ist eine Liste mit den ermittelten abstrakten Anforderungen. Im *Requirement Development*, dem zweiten Teil der Analysephase, werden diese Anforderungen durch den Business Developer und den Solution Designer in kleinere spezifischere Anforderungen heruntergebrochen und der Prozessaspekt ausgeformt. Die spezifizierten Anforderungen der einzelnen Aktivitäten und Elemente werden jeweils in einzelnen SUC beschrieben und zum Teil bereits konkrete Services „erwähnt“ (= informell referenziert). Falls weitere fachliche Informationen nötig sind, werden die Business User über den Business Developer in den Prozess der Anforderungsdefinition miteinbezogen. Die erstellten low-level Anforderungen werden dann den einzelnen Elementen der Lösung (Aktivitäten, Interfaces, etc.) zugeordnet. Das Ergebnis der Anforderungsanalyse ist eine komplette Beschreibung aller Anforderungen und Lösungselemente des Geschäftsprozesses in einer PSD.

In der Spezifikationsphase werden die ermittelten Anforderungen in entsprechende Modelle überführt. Während der Spezifikationsphase erstellt der Solution Architekt zuerst aus den bestehenden SUCs „wohldefinierte“ SUCs, indem konkrete Services referenziert und technische Informationen (z.B. Übergabeparameter für die Service) ergänzt werden. Anschließend werden die definierten Anforderungen der Lösung manuell in ein SPF-Modell transformiert. Die einzelnen Aktivitäten eines SPF-Modells werden jeweils mit den zugehörigen SUCs verknüpft. Ziel dieser Phase ist die vollständige Spezifikation der zu entwickelnden Lösungen aus fachlicher Sicht, so dass nach dieser Phase keine fachlichen Benutzer mehr in die Prozessentwicklung eingebunden sein müssen. Wenn der Solution Architekt feststellt, dass die Spezifikation, beispielsweise aufgrund fehlender Daten, unvollständig ist, passt er das Modell, soweit ihm möglich, an und stimmt das veränderte Modell mit dem Business Developer ab.

Auf Basis der Modelle und der SUC-Beschreibungen wird der Prozess anschließend entwickelt. Dafür werden das logische Modell (SPF) und die wohldefinierten SUCs manuell in ein ausführbares Modell (BPEL) transformiert.

Einbeziehung der Endbenutzer

In den Gesamtprozess sind sowohl Business Users (End-User) als auch Business Developer (Local Developer) eingebunden. Primär werden die Endbenutzer in der Analysephase durch den Business Developer vertreten. Dieser soll das nötige fachliche Wissen für die Spezifikation der Anforderungen besitzen. Die Business User aus den Fachabteilungen werden bei der Analyse der abstrakten Anforderungen sowie zur Diskussion von Lösungsideen und -alternativen („*Meinst du das wirklich?*“ „*Ist dies nicht ein besserer Weg etwas zu tun?*“) miteinbezogen. Dabei werden, wie geschildert, zwar bereits einzelne Schritte der Prozesse betrachtet, diese aber ausschließlich informell erfasst. In den Gesprächen mit den Nutzern wird daher in der Regel lediglich mit textuellen Beschreibungen gearbeitet. Zu einem gewissen Grad werden mit den Anwendern jedoch bereits vereinfachte Modelle skizziert, die „[...] ein wenig ähnlich zu SPFs, nur mit deutlich weniger Details [sind].“ Diese stellen aber keine offiziellen Prozessdokumente dar und müssen manuell in SPF-Modelle transformiert werden. Nach der Erfahrung des Interviewten bedingt sich der Verzicht auf eine Modellierung mit den Endbenutzern durch eine zu hohe Komplexität der SPF-Notation für Endbenutzer und dadurch, dass es „[...] einiges an Erfahrung und Wissen [erfordert], um zu wissen was es bedeutet, dass ausschließlich Prozesse im SPF definiert und betrachtet werden und nicht die Benutzerschnittstellen oder andere Aspekte.“ Die genaue Gesprächskonstellation bei der Einbeziehung der Business User ist abhängig von der Art der Lösung. Meist sitzen jedoch Business Developer und verschiedene Business User, die über das nötige fachliche Wissen verfügen und bei der organisatorischen Umsetzung helfen können, zusammen. Die

Spezifikation der Modelle erfolgt aktuell zwar hauptsächlich durch Solution Architects, langfristig besteht jedoch die Bestrebung den Business Developer anzuleiten Prozesse in einer formalisierten Sprache (zumindest in Teilen) selbstständig formal zu spezifizieren.

Praxisfeedback zur Idee eines papierbasierten Modellierungsinterfaces

Um ein erstes Praxisfeedback zu erhalten, wurde ein erstes Grobkonzept einer interaktiven, papierbasierten Modellierung als Ergänzung zur aktuellen Arbeitspraxis kurz erläutert.

Im Vergleich zur aktuellen Modellierung auf einem Bildschirm wurde vor allem die Möglichkeit, einfach eine große Zeichenfläche nutzen zu können, als positiv empfunden. Auch wurde die Hoffnung geäußert, dass durch Prozessskizzen auf Papier deutlich einfacher kreativ modelliert werden kann, was als wichtiger Punkt für Prozessmodellierung herausgestellt wurde. Als praktisch und zeitsparend wurden auch die Möglichkeiten zur „automatischen“ Erfassung von Prozessen und die direkte Zuordnung von Anforderungen gesehen (zeitsparend, „[...] *da man die Sachen erledigt, während man sie eh macht.*“).

Skeptisch gesehen wurde vor allem die Einbettung in die aktuelle Arbeitspraxis, insbesondere die Einbindung in die existierenden Werkzeuge und das bestehende Vorgehensmodell. So macht „[...] *das Zeichnen von Prozessen [...] nur einen kleinen Teil der ersten Phase aus.*“ Deutlich wichtiger als das reine Zeichnen von Prozessmodellen sind das Sammeln von Informationen und das Erfassen von Anforderungen. Durch die reine Unterstützung des Zeichnens von Prozessmodellen auf dem Papier würde also lediglich ein kleiner Teil Prozessfassung abgedeckt werden. Zudem wurde kritisch angemerkt, dass die existierende Visualisierung von Prozessen auf Basis von SPFs zwar gut geeignet ist um Prozesse zu dokumentieren, jedoch sehr schlecht verständlich für Endbenutzer ist. Auch wurde befürchtet, dass bei einer skizzenbasierten Prozessmodellierung mit Endbenutzern ein starker Fokus auf den Details der einzelnen Aktivitäten („[...] *und dann machen wir dies, und wenn wir dahin gehen machen wir das, und so weiter.*“¹⁶) und nicht auf dem eher abstrakt orientierten Prozessablauf liegen würde.

3.1.4 Bewertung und Fazit

Die Fallstudie gab wertvolle Einblicke in die BPM-Praxis großer Unternehmen. Bei der Bewertung der Ergebnisse muss jedoch berücksichtigt werden, dass das untersuchte Unternehmen eine gewisse Vorreiterrolle in diesem Feld einnimmt. Auch existieren durch die Größe des Unternehmens Rollenaufteilungen, die sich in kleineren Unternehmen vermutlich nicht eins zu eins wieder finden. So würde in mittelständischen Unternehmen ein Business Developer vielleicht nicht in der IT-Abteilung, sondern in der Fachabteilung sitzen. Es wird

¹⁶ Der Interviewte befürchtet, dass sich die Nutzer in Details verlieren würden.

jedoch deutlich, dass die Prozesserstellung in SOA ein komplexer, langfristiger und mehrstufiger Prozess ist. Eine vollständige Service-Orchestrierung durch Endbenutzer erscheint somit schwierig. Besonders in den ersten Entwicklungsschritten können Endbenutzer jedoch mittels interaktiven Papiers unterstützt werden, aktiv an einer Prozesserstellung teilzunehmen. Ihre Anforderungen und Prozessideen können sie auf diese Weise semi-formal einbringen und so ihre Anforderungen bereits sehr konkret weitergeben. Im Folgenden sind die wichtigsten Ergebnisse der Fallstudie zusammengefasst:

1. Das Zeichnen von Prozessmodellen deckt nur einen (kleinen) Teil des Gesamtprozesses ab. Ebenso wichtig ist die (informelle) Erfassung der Prozessanforderungen in den SUCs, die viel Zeit in Anspruch nimmt.
2. Die Prozesserstellung erfolgt in der Regel als iterativer Prozess in einer oder mehreren kollaborativen Sitzungen, an denen verschiedene Benutzertypen (End-User, Local Developer, Professional Programmer) mit unterschiedlicher Modellierungserfahrung teilnehmen.
3. Die eigentliche Modellierung findet aktuell primär durch Modellierungsexperten statt. Erfahrene Endbenutzer (Local Developer) werden schrittweise beteiligt und modellieren selbst formale Prozessbeschreibungen.
4. End-User sind nur in die informelle Prozesserfassung einbezogen, da die verwendete SPF-Notation für sie zu komplex ist. Zum Teil werden mit End-Usern jedoch bereits vereinfachte, aber SPF ähnliche, Prozessmodelle skizziert.
5. Interaktives Papier bietet einige Möglichkeiten, die als positiv für den, durch Informalität und Kreativität geprägten, Prozess der Modellierung angesehen werden: große Modellierungsfläche, Unterstützung der Kreativität, einfaches Erfassen und Zuordnen von informellen Aspekten und automatische Transformation in digitale Prozessmodelle.
6. Als kritisch für eine papierbasierte Modellierung werden die Gestaltung der Prozessnotation und die Gefahr eines zu starken Fokus auf Aktivitätsdetails statt auf den Prozessablauf gesehen.

3.2 Design-Workshop: Kollaboratives BPM durch Endbenutzer

Die zweite Säule der Vorstudie bildet ein Prozessdesign-Workshop, der mit Endbenutzern aus der Praxis durchgeführt wurde. Ziel des Workshops waren Erkenntnisse über die Akzeptanz und Herangehensweise von Endbenutzern an eine Geschäftsprozessmodellierung mit Stift und Papier zu erlangen. Dazu wurden die Endbenutzer mit der Aufgabe konfrontiert, ihnen bereits bekannte Geschäftsprozesse kollaborativ mit Stiften und Papier Mock-Ups zur erstellen. Durch die Auswertung des Workshops konnten wertvolle Designerkenntnisse für die Gestaltung eines papierbasierten Modellierungsansatzes gewonnen werden. Die Beschreibung

des Workshops setzt auf einer ersten, verkürzten Darstellung aus einer vorangegangenen Arbeit (siehe [20]), auf.

3.2.1 Methodik

Mit Fokus auf die Gestaltung einer endbenutzerorientierten Designumgebung wurde, in Anlehnung an das Konzeptes des Participatory Designs (siehe [18, 117] und Seite 21), eine Designstudie mit Einbeziehung von Nutzern durchgeführt. Der Workshop fand in einer “in-between”-Region, wie es Muller [116] bezeichnet, statt. Dies bedeutet, dass der Workshop sowohl Elemente aus der täglichen Arbeitsumgebung der Endbenutzer, als auch Elemente aus der Arbeitsumgebung der Entwickler umfasste. Zwar war das Feld der Modellierung von Geschäftsprozessen für die Teilnehmer bisher eher unbekannt, jedoch besaßen sie bereits Erfahrungen in der Anpassung des unternehmenseigenen SAP-Systems. Durch die Wahl eines Szenarios aus der Arbeitspraxis der Teilnehmer, konnte an diese Erfahrungen angeknüpft werden. Indem das Szenario für die Modellierung eines Geschäftsprozesses mit derselben Funktionalität genutzt wurde, konnten Erkenntnisse gesammelt werden, wie Endbenutzer Modellierungselemente intuitiv verwenden und welche Semantik ihnen natürlich erscheint. Für die Modellierung standen den Teilnehmern die in den meisten EUD-Modellierungsansätzen verwendeten Elemente der Boxes-and-Wires Metapher (siehe Kapitel 2.2.3) zur Verfügung, die über eine unterspezifizierten Semantik definiert waren. Die Kästchen (kleine vorstrukturierten Papierkärtchen) repräsentierten eine bestimmte Funktionalität und die Verbindungen der Boxen (Striche auf dem Papier) den Datenfluss. Der gesamte Workshop wurde auf Video aufgezeichnet, ergänzt durch handschriftliche Notizen der Beobachter und zusätzliche Fotos.

3.2.2 Konzeption

Am Workshop nahmen insgesamt neun Personen teil. Drei davon waren Beschäftigte eines mittelständischen Unternehmens (mit circa 150 Beschäftigten) der Produktionsbranche (im Folgenden bezeichnet als Nutzer). Die restlichen sechs Teilnehmer waren Forscher der Universität Siegen. Alle drei Nutzer besaßen durch die häufige Nutzung von Microsoft Excel und dem unternehmenseigenen SAP-System bereits technische Erfahrung und hatten im Unternehmen die formale Rolle des „Key Users“ für das SAP-System inne. Einer der Nutzer war der Leiter der IT-Abteilung, die anderen zwei (weiblichen) Nutzer arbeiteten als Assistentin der Geschäftsleitung und als Einkaufsmanagerin. Die Forscher setzten sich zu jeweils zwei Personen aus den Gruppen Entwickler (die am Design des Systems arbeiteten), Moderator und Beobachter zusammen und stammten aus dem Bereich der Wirtschaftsinformatik oder der Sozialwissenschaft. Der fünfstündige Workshop war unterteilt in eine Einführungsphase am Vormittag und eine Entwurfsphase am Nachmittag und wurde in einem speziellen Besprechungsraum der Universität Siegen durchgeführt. Als

Arbeitsmaterialien wurden Papierbögen in mehreren Formaten, Haftnotizen, verschiedene farbige Stifte und eine Tafel bereitgestellt. Die zentrale Modellierungsfläche bildete ein circa DIN B0 (1m x 1,40m) großes beschichtetes und daher wiederbeschreibbares Papier.

3.2.3 Ablauf

Zu Beginn des Workshops wurden verschiedene analytische Szenarien aus dem Arbeitskontext der Teilnehmer spezifiziert und anschließend eines ausgewählt. Das gewählte Szenario bildet einen wichtigen Teilaspekt der Planung von Produktbestimmungen ab. Um die entsprechende Bestellmenge zu planen wird eine Übersicht benötigt, die für alle Produktgruppen eine, nach bestimmten Attributen sortierte und zeitlich nach Monaten untergliederte Liste des Eingangsbestandes, der Bestandsveränderung, der geplanten Verkäufe, der aktuellen Verkäufe und des geplanten Endbestandes zeigt. Die Komplexität des Szenarios ergibt sich daraus, dass die benötigten Daten aus verschiedenen Modulen des ERP-Systemes ausgewählt und anschließend in Microsoft Excel konsolidiert werden müssen. Zudem müssen der in Excel erstellten Tabelle verschiedene berechnete Spalte (z.B. der voraussichtliche Endbestand) hinzugefügt werden.

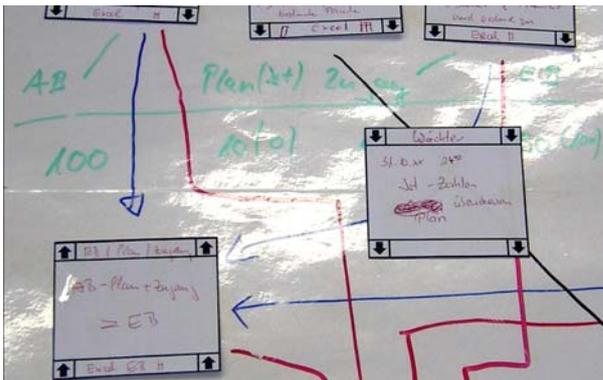


Abbildung 11: Kästchen und Verbindungen



Abbildung 12: Designfläche des Workshops

Nach der Auswahl des Szenarios wurde den Nutzern das gewählte Designparadigma der Boxes-and-Wires vorgestellt. Durch das Beschriften der vorstrukturierten Kästchen wurde deren Inhalt oder Funktion beschrieben. Die beschrifteten Kästchen konnten dann auf dem Designpapier angeordnet und durch entsprechende Linien verbunden werden. Die Kästchen wurden verbunden, indem der Ausgabeport eines Kästchens mit dem Eingabeport eines anderen Kästchens verknüpft wurde (Abbildung 11). Dabei mussten die Nutzer annotieren, welche Daten ein Kästchen liefert. Zusätzliche Annotationen konnten verwendet werden, um spezielle Funktionen oder die Semantik der Verbindungslinien zu spezifizieren. Der zentrale Modellierungsbereich war für alle Nutzer gut zugänglich (Abbildung 12). Verschiedenfarbige Stifte erlaubten eine Unterscheidung verschiedener semantischer Bedeutungen von Verbindungen und Annotationen. Bei Problemen während der Modellierung der Lösung

konnten sich die Nutzer an die Softwareentwickler wenden. Die Moderatoren stellten das gegenseitige Verständnis der Gruppen und den Fokus auf die Problemstellung sicher.

3.2.4 Ergebnisse

Zu Beginn der Modellierung wurde von den Nutzern in einer gemeinsamen Diskussion überlegt, welches Ziel erreicht werden soll. Das gewünschte Ergebnis wurde dann in Form einer Tabelle auf einem Papier skizziert. Die einzelnen Spalten wurden zusätzlich mit der Bezeichnung der jeweiligen SAP-Module, die die entsprechenden Daten für die Spalte liefern sollten, annotiert. Das zur Erstellung einer solchen Lösung entwickelte und skizzierte Modell ist in einer vereinfachten Form in Abbildung 13 dargestellt.

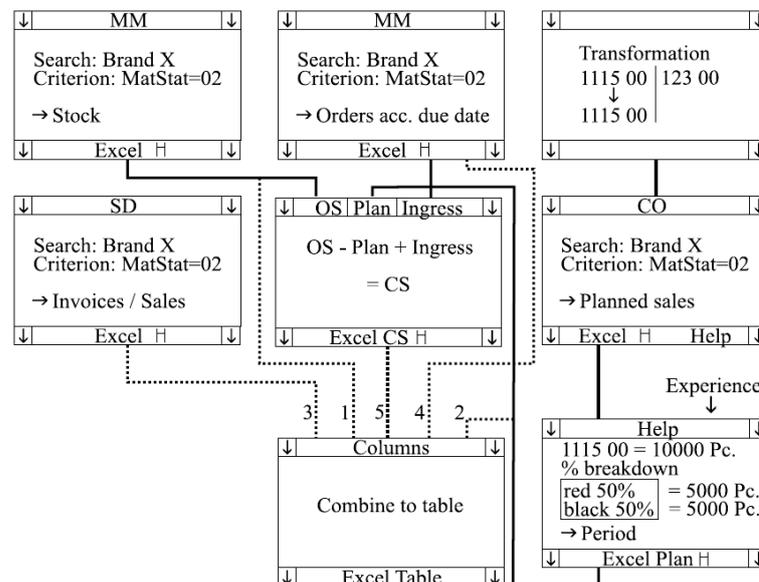


Abbildung 13: Vereinfachte Darstellung der skizzierten Lösung [20]

Den eigentlichen Modellierungsprozess starteten die Endbenutzer mit einer Diskussion, wie die einzelnen Kästchen, die die benötigten Daten repräsentieren, zu spezifizieren seien. Sie entschieden sich dafür jedem SAP-Modul, aus dem Daten benötigt werden, ein eigenes Kästchen zuzuordnen. Soweit nötig, wurde der mittlere Bereich der Kästchen zusätzlich mit Kriterien für die Auswahl einer Teilmenge von Daten annotiert. Der obere Bereich (Eingabepart) der Kästchen wurde mit einer kurzen Beschreibung der gewünschten Daten gefüllt und der untere Bereich (Ausgabepart) mit dem gewünschten Ausgabeformat („Excel“) und der Anzahl der zu erstellenden Spalten bezeichnet. Der Eingabebereich eines Kästchens wurde zusätzlich mit einem besonderen farbigen Kästchen mit der Aufschrift „Erfahrung“ verbunden. Die Nutzer wollten damit ausdrücken, dass für diese Planungsfunktion Erfahrung als „Eingabe“ benötigt wird. Der genaue Ablauf der Planung wurde in der Mitte dieses Kästchens beschrieben. Der Ausgabebereich dieses Kästchens und zwei weiterer Kästchen wurden mit dem Eingabebereich eines speziellen „Berechnungskästchens“ verbunden. Der

Eingabebereich dieses Kästchens wurde mit den abgekürzten Bezeichnungen der Daten, die die anderen Kästchen an diese weitergeben sollen, beschrieben. Der mittlere Bereich enthält eine Formel, die auf die übergebenen Eingabedaten angewendet werden soll. Der Ausgabebereich aller Kästchen, die eine Spalte zu der Endtabelle beitragen sollen, wurde mit dem Kästchen, das das Endergebnis repräsentiert, verbunden. Dessen Eingabebereich wurde mit „Spalten“ bezeichnet. Die dort eingehenden Verbindungslinien besaßen eine spezielle Farbe (rot) und waren mit Zahlen annotiert. Die Semantik der Zahlen gibt die Reihenfolge der Spalten in der Ergebnistabelle wieder. Die Mitte des Kästchens war mit der Funktion „Kombiniere in Tabelle“ beschrieben.

3.2.5 Bewertung und Fazit

Der Workshop hat gezeigt, dass Endbenutzer in der Lage sind das Konzept der Prozessmodellierung in SOA zu verstehen und anzuwenden, besonders wenn sie dabei durch einfache Werkzeuge unterstützt werden und sich eine eigene Vorstellung von der Prozessmodellierung machen können. Auch wurde bestätigt, dass ein papierbasiertes Interface von Endbenutzern akzeptiert wird und dass es eine kreative, informelle und kollaborative Modellierung gut unterstützen kann. Zudem wurden Beobachtungen aus dem Bereich der skizzenbasierten Modellierung bestätigt (siehe Damm et al. [36] und Kapitel 2.3.3). Im Gegensatz zu einer typischen Prozessmodellierung, bei der in der Regel der zeitlich-logische Ablauf modelliert wird, fällt auf, dass im Workshop der Datenfluss abgebildet wurde. Dies ist bedingt durch die Wahl des Orchestrierungsszenarios aus der tabellen- und datenorientierten Denkwelt der Nutzer. Die wichtigsten Ergebnisse des Workshops sind im Folgenden kurz zusammengefasst:

1. Das Visualisierungskonzept der Boxes-and-Wires und die verwendete Begrifflichkeit wurde mit wenig Unterstützung und einer kurzen Einarbeitungszeit durch die Endbenutzer verstanden.
2. Dennoch waren einige Inkonsistenzen bei der Nutzung der Kästchen zu beobachten: Der mittlere Kästchenbereich wurde sowohl für die Beschreibung der Funktion des Services, als auch für die Beschreibung der gewünschten Ausgabe verwendet. Auch wurden zu verarbeitenden Daten als Funktionen (z.B. „Suche: Marke X“) statt als Eingabeparameter beschrieben.
3. Neben den formalen Elementen haben die Endbenutzer auch Annotationen (z.B. die Charakterisierung der Spaltenreihenfolge) und informelle Erweiterungen (z.B. Erfahrung als Eingabeobjekt sowie die gewünschte Ergebnistabelle) verwendet und diese fließend miteinander kombiniert.
4. Zum Beispiel über verschiedenfarbige Verbindungslinien verwendeten die Nutzer verschiedene *Sekundäre Notationen* (siehe Seite 22).

4 Konzeption einer papierbasierten Servicemodellierung

Aufbauend auf den Erkenntnissen aus der Literatur und der durchgeführten Vorstudie wird in diesem Kapitel gemäß der in Kapitel 2.4 geschilderten Idee ein Konzept zur *papierbasierten serviceorientierten Modellierung* (abgekürzt *PaSeMod*) entwickelt. Zu Beginn werden der Gestaltungsrahmen (Kapitel 4.1) und die Anforderungen (Kapitel 4.2) eines solchen Konzeptes beschrieben. Kapitel 4.3 bis 4.5 stellen im Anschluss, untergliedert in Grund-, Interaktions- und Feedbackkonzept, die konkrete Gestaltung des PaSeMod-Konzeptes vor. Da die Modellierung auf Papier nur einen ersten Schritt der Gesamtprozesserstellung darstellt, werden in Kapitel 4.6 abschließend Aspekte der digitalen Weiterverarbeitung der papierbasierten Prozessmodelle beschrieben.

4.1 Gestaltungsrahmen

Als Grundlage für die Anforderungen und die Gestaltung des Konzeptes, wird im Folgenden der Gestaltungsrahmen für PaSeMod beschrieben. Zu Beginn wird auf die Positionierung des Konzeptes in den Forschungsfeldern und die Abgrenzung zu bestehenden Systemen eingegangen. Im Anschluss werden ein beispielhafter, papierbasierter Modellierungsprozess sowie Rahmenbedingungen und Einflussfaktoren für die Gestaltung von PaSeMod beschrieben. Abschließend werden kognitive Gestaltungsaspekte für das Konzept diskutiert.

4.1.1 Positionierung und Abgrenzung des Konzeptes

Die Literatur (Kapitel 2.1.2) und die Fallstudie (Kapitel 3.1.3) haben gezeigt, dass die Prozesserstellung in serviceorientierten Architekturen ein mehrstufiger Prozess mit einer ansteigenden Formalisierung und Komplexität ist. Ausgehend von informell geäußerten Bedürfnissen und einfachen Prozessskizzen werden Anforderungen strukturiert erfasst, ein formales Prozessmodell erstellt, dieses mit konkreten Services verknüpft und schlussendlich die Services in einem ausführbaren Modell orchestriert. Gemäß der Forderung des „flow of learning“ ([144], siehe Seite 19), sollten Endbenutzer befähigt werden, in einem graduellen Lernprozess schrittweise und mit ansteigender Mächtigkeit und Komplexität die benötigten Fähigkeiten zur Bewältigung einer serviceorientierten Modellierung bzw. Orchestrierung zu erwerben. Eine Integration aller Modellierungsschritte in einem System ist aufgrund der stark unterschiedlichen Ausrichtung der Schritte jedoch nicht möglich. Vielmehr müssen für die einzelnen Schritte jeweils geeignete endbenutzerorientierte Konzepte und Interfaces (vgl. Wulf et al. [174]) angeboten werden.

Die in Kapitel 2.2.3 vorgestellten endbenutzerorientierten Modellierungsansätze (z.B. *SiSeOr*, siehe Seite 26) ermöglichen, wenn auch nur für vergleichsweise kleine Prozesse, eine endbenutzergeeignete Orchestrierung bestehender Services und sind somit in eine

fortgeschrittene und stark formal geprägte Phase der Prozesserstellung einzuordnen. Professionelle Modellierungswerkzeuge (z.B. *WebSphere Business Modeler*, siehe Kapitel 2.1.3) unterstützen meist eine formale Modellierung und eine Orchestrierung. Darüber hinaus ermöglichen elektronische skizzenbasierte Modellierungswerkzeuge (z.B. *KNIGHT*, siehe Kapitel 2.3.5) auch eine informelle Modellierung. Beide sind aufgrund ihrer Komplexität für Endbenutzer jedoch ungeeignet (vgl. [63, 104]). Eine endbenutzergerechte Unterstützung informeller oder teilformeller Modellierungsaspekte existiert bisher nicht.

An dieser „ersten Stufe“ der Prozesserstellung positioniert sich das PaSeMod Konzept. In dieser durch Kreativität und Informalität geprägten ersten Phase kommen die Vorteile eines interaktiven Papierinterfaces besonders zu tragen. In durch Formalisierung und Konkretisierung geprägten Aspekten späterer Phasen sind dagegen GUI-basierte Systeme besser geeignet. Demgemäß wird der „Gentle Slope of Complexity“ einer Prozesserstellung auf mehrere Systeme aufgeteilt: Erste, teilformeller Prozessskizzen und Prozessanforderungen werden vor allem unter Einbeziehung von End-Usern papierbasiert mit PaSeMod erstellt. Anschließend wird das Prozessmodell in ein geeignetes (endbenutzerorientiertes) GUI-basiertes Modellierungssystem überführt und dort konkretisiert.

4.1.2 Idealtypischer Modellierungsablauf

Basierend auf den Ergebnissen der Vorstudie und der geschilderten Positionierung des Konzeptes wird im Folgenden die Idee eines idealtypischen, papierbasierten Modellierungsablaufes, unterteilt in fünf Schritte, beispielhaft dargestellt. In den ersten drei Schritten erfolgt die Prozesserstellung primär durch End-User, die zum Teil durch Local Developer unterstützt und angeleitet werden. Während dieser drei papierbasierten Modellierungsschritte werden die Benutzer durch kontextbezogene Interaktionsmethoden über einen mit dem Stift verbundenen Computer unterstützt. Interaktionsmethoden sind zum Beispiel die Erklärung syntaktischer und semantische Aspekte, die Bestätigung einer formal erkannten Skizze oder eine weitergehende Visualisierung des Prozessmodells.

1. Einer oder mehrere End-User sowie gegebenenfalls Local Developer erstellen auf einem interaktiven Papier eine erste grobe Prozessskizze. Die Anforderungen einzelner Aktivitäten werden separat beschrieben und mit der Skizze verknüpft.
2. In einem nächsten Schritt wird das papierbasierte Prozessmodell von (anderen) Endbenutzern auf dem Papier verfeinert und ergänzt. Zusätzliche Anmerkungen und Aspekte unbekannter Prozesskonzepte werden über informelle Annotationen beschrieben.
3. Anschließend wird der Prozess gemeinsam mit modellierungserfahrenen Local Developern und gegebenenfalls professionellen Modellierern (Professional Programmers) auf Basis der bestehenden papierbasierten Skizzen, soweit papierbasiert

sinnvoll, vervollständigt. Durch die Beobachtung von Benutzern mit mehr Erfahrung können die Endbenutzer dabei ihr Modellierungswissen anreichern.

4. Als nächster Schritt wird das Papiermodell für eine Weiterbearbeitung (insb. eine Konkretisierung) mit anderen BPM-Werkzeugen vom System automatisch in ein digitales, visuelles Prozessmodell transformiert. Formelle Skizzen werden in die entsprechenden formalen Prozesskonzepte überführt. Informelle Skizzen werden im digitalen Modell bewahrt, um eine spätere Formalisierung zu ermöglichen und den Prozessfortschritt und die Bedeutung einzelner Skizzen zu dokumentieren.
5. Das digitale Prozessmodell wird dann durch geeignete Werkzeuge am Computer vollständig formalisiert, konkretisiert und abschließend in ein ausführbares Modell überführt. Die digitale Weiterverarbeitung erfolgt gemeinsam von Local Developern und Professional Programmern. Die Erstellung eines ausführbaren Prozessmodells erfolgt, aufgrund notwendiger spezifischer Detailkenntnisse, ausschließlich durch Professional Programmer.

4.1.3 Rahmenbedingungen und Einflussfaktoren

Durch die beschriebene Einbindung in einen ganzheitlichen Modellierungsprozess, muss sich die Gestaltung des Modellierungskonzeptes, insbesondere die der Prozessnotation, an bestehenden serviceorientierten Konzepten orientieren.

Weitere Rahmenbedingungen ergeben sich durch die Nutzung interaktiver Papierinterfaces. Neben den beschriebenen Vorteilen der Informalität und Flexibilität bringen diese eine Reihe von Einschränkungen für die Gestaltung des Konzeptes mit sich (siehe Kapitel 2.3.2). Insbesondere die fehlende Darstellbarkeit dynamischer Aspekte grenzt die Nutzung verschiedener Unterstützungsmechanismen und EUD-Gestaltungsaspekte, wie zum Beispiel das Verhindern syntaktischer Fehler (siehe Repenning und Ioannidou [144]) oder ein dynamisches Testen, stark ein. Auch schließt sich eine kombinierte Modellierung auf dem Papier und über einen verbundenen Computer aus, da die Änderungen am Computer nicht automatisch auf das Papier übertragen werden können. Auf Seite 29 wurden verschiedene interaktive Papieransätze vorgestellt. Von diesen erscheint die Anoto-Technologie, aufgrund der *Location-Awareness* Funktionalität und ihrer Flexibilität, am besten geeignet. Jedoch schränkt sie die Erkennung auf reine Papierskizzen und verschiedene Papiertypen ein. Auch ein Entfernen einmal erstellter Skizzen wird nicht unterstützt. Zwar können Striche unter bestimmten Bedingungen physisch wieder entfernt werden,¹⁷ jedoch gibt es keine

¹⁷ Zum Beispiel durch die Verwendung von Bleibstiftminen oder spezieller, radierbarer Kugelschreiberminen in Verbindung mit Radiergummi.

Möglichkeit dies auch digital zu erkennen. Zudem ist die Verarbeitung der Eingabe bei Anoto nur auf einem Computer und nicht direkt im Stift möglich.¹⁸

Wie bereits geschildert, sind dynamische EUD-Ansätze, aufgrund eines fehlenden Feedbacks von Papier, nicht anwendbar. Auch das Konzept des Programming by Example ist wegen der Informalität der Modellierung schlecht einsetzbar. Aufgrund der Orientierung an bestehenden serviceorientierten Modellierungsansätzen stehen vielmehr Aspekte der visuellen Programmierung (VP) im Vordergrund. Diese kann, wie auf Seite 23 beschrieben, zudem helfen die Kommunikation zu unterstützen. Da sie syntaktische Strukturen abschwächt, eignet sie sich auch gut für die Abbildung semi-formale Aspekte. In Zusammenhang mit der VP kommt besonders den Cognitive Dimensions (Seite 22) als Gestaltungskriterien für gebrauchstaugliche, visuelle Entwicklungssysteme eine große Bedeutung zu. Hinsichtlich der Gestaltung eines natürlichen, für die Nutzer einfach verständlichen Interaktionskonzeptes sind ebenfalls Aspekte der natürlichen Programmierung (Seite 24) einzubeziehen.

4.1.4 Diskussion kognitiver Gestaltungsdimensionen

Bei der Gestaltung benutzbarer Systeme sind von Beginn an Usability-Aspekte zu berücksichtigen (siehe Seite 22). Für das konkrete Konzept von PaSeMod ist besonders zu beachten, dass papierbasierte Systeme grundlegend andere Herausforderungen an die Interaktionsgestaltung stellen als GUI-basierte Systeme und dass bisher keine vergleichbaren Ansätze einer papierbasierten Modellierung existieren. Um eine fundierte Gestaltungsbasis zu erlangen, werden daher im Folgenden Gestaltungsdimensionen für ein papierbasiertes, endbenutzerorientiertes und visuelles Modellierungssystem diskutiert. Als Diskussionsbasis wird das, von Green und Petre [53, 54] als „Diskussionswerkzeug“ für die Usability von Entwicklungs- und Interaktionssystemen vorgeschlagene, Cognitive Dimensions-Framework verwendet. Tabelle 4 zeigt eine Übersicht der beinhalteten Gestaltungsdimensionen. Im Anschluss werden, untergliedert in fünf Gruppen, die wichtigsten Gestaltungsaspekte diskutiert. Im Mittelpunkt der Diskussion steht das Gesamtsystems, bestehend aus der Notation, dem Systemumfeld und dem verwendeten Interaktionsmedium Papier. Da zwischen einzelnen Dimensionen Zielkonflikte bestehen und somit ein „perfektes“ System nicht gestaltet werden kann, sind die Ergebnisse der Diskussion nicht als harte Anforderungen, sondern als Gestaltungsideen zu sehen.

Abstraction gradient	Verfügbarkeit und Typen von Abstraktionsmechanismen.
Closeness of mapping	Nähe des Systems zur Anwendungsdomäne.
Consistency	Erschließbarkeit neuer Aspekte aus bereits Bekanntem.
Diffuseness / terseness	Verbosität der Sprache (Benötigter Platz für eine Visualisierung).

¹⁸ Mit Ausnahme bestimmter vordefinierte und nicht beeinflussbarer Anoto-Punktmuster.

Error-proneness	Inwieweit lädt die Notation ein Fehler zu machen?
Hard mental operations	Gibt es hohe Anforderungen an die kognitive Verarbeitung?
Hidden dependencies	Fehlende Sichtbarkeit von wichtigen Verbindungen zwischen Entitäten.
Juxtaposability	Möglichkeiten zwei Komponenten nebeneinander zu positionieren.
Premature commitment	Beschränkungen der Reihenfolge einzelne Schritte zu erledigen.
Progressive Evaluation	Grad der Möglichkeit zur Evaluierung unfertiger Lösungen.
Role-expressiveness	Möglichkeit die Bedeutung einzelner Komponenten zu erschließen.
Secondary notation and escape from formalism	Aufzeichnung zusätzlicher Informationen mit anderen Mittel als der formalen Syntax.
Viscosity	Widerstand des Systems gegen Änderungen am Modell.
Visibility	Möglichkeit Komponenten ohne Aufwand zu betrachten.

Tabelle 4: Übersicht der Cognitive Dimensions von Green et al. [53, 54]

Visibility und Juxtaposability: Da auf Papier alle Informationen ohne weiteren Aufwand direkt sichtbar sind, besitzt ein Papierinterface, vor allem bei der Verwendung großformatiger Papiere, eine gute *Sichtbarkeit* (= *Visibility*). Lediglich überfrachtungskritische Aspekte, wie eine unstrukturierte Darstellung gepaart mit einer hohen Informationsdichte (z.B. ausführliche Beschreibungen der Prozessschritte), sollten durch strukturierte Darstellungsmöglichkeiten unterstützt werden. Nicht implizit unterstützt wird dagegen die Sichtbarkeit von Interaktionsmöglichkeiten (z.B. verfügbare Befehle und Notationselemente). Zur Verfügung stehende Interaktions- und Modellierungsmöglichkeiten sollten den Nutzern daher jederzeit klar kommuniziert werden. Das *Nebeneinanderstellen* (= *Juxtaposability*) von Informationen wird durch die Flexibilität von Papier und die Möglichkeit zur Verwendung mehrerer Papiere sehr gut unterstützt.

Secondary notation, Closeness of mapping und Role-expressiveness: Die Möglichkeit *sekundäre Notationen* zu nutzen ist vor allem für Endbenutzer wichtig, da ihnen dies die Möglichkeit gibt unbekannte Aspekte informell zu artikulieren (z.B. „Erfahrung“ als Input im Workshop). Generell werden *sekundäre Notationen* durch Papier sehr gut unterstützt, da Benutzer problemlos informelle Annotationen erstellen können. Wichtig ist jedoch, dass die erstellten Annotationen auch für eine digitale Weiterverarbeitung geeignet und möglichst natürlich sichtbar gemacht werden sollten. Die Vertrautheit von Papier und die einfache Integration informeller Aspekte und *sekundärer Notationen* verringert die Gefahr, dass das System zu stark an der Entwicklungsdomäne (= fehlendes *Closeness of mapping*) ausgerichtet ist. Auch die Verwendung einer nichtdomänenspezifischen Visualisierung ist aufgrund der Erfahrungen aus den Vorstudien (z.B. Workshop E1¹⁹) wenig kritisch zu sehen. Durch

¹⁹ Mit „Workshop Ex“ sei auf das Ergebnis x. der Zusammenfassung der Workshop-Auswertung in Kapitel 3.2.5 verwiesen. Analoges gilt für „Fallstudie Ex“ und deren Ergebniszusammenfassung in Kapitel 3.1.4.

informelle Beschreibungen ermöglicht ein papierbasiertes Modellierungssystem zudem eine *aussagekräftige Beschreibung von Rollen* (= *Role-expressiveness*).

Viscosity, Premature commitment und Abstraction gradient: Die fehlende Möglichkeit Änderungen an erstellten Papierstrichen vorzunehmen, führt zu einer hohen *Viskosität* (= *Viscosity*) von Papier. Diese erfordert, vor allem beim Skizzieren umfangreicher Prozesse, eine hohe *Vorausplanung* (= *Premature Commitment*), da vor dem Zeichnen die Größe und Anordnung der Skizzen bedacht werden müssen. Grundsätzlich lässt sich die *Viskosität* nicht vermeiden, jedoch sollte ihr durch die Bildung verschiedener *Abstraktionsstufen* (z.B. eine Prozesssicht und Detailsicht) und die Aufteilung von Informationen auf verschiedene Papiere, entgegengewirkt werden. Zudem sollte über das Interaktionskonzept und die Notation zumindest digital eine Änderung der Bedeutung von Skizzen (z.B. Konkretisieren) und ein möglichst freies Einfügen zusätzlich Prozesselemente (z.B. durch „Überkleben“) möglich sein. Da verschiedene *Abstraktionsstufen* das Prozessverständnis der Endbenutzer erschweren, sollten sie allgemein vermieden werden. Nicht vermeidbar ist jedoch die, auch in der Fallstudie beobachtete, BPM-typische Abstraktion in Prozess- und Detailebene. Da eine klare Trennung dieser Ebenen aber gerade für Endbenutzer schwierig erscheint (Fallstudie E6), sollte sie im System durch eine klare Abgrenzung betont werden. Den geschilderten *Viskositäts-* und *Vorausplanungs-Problemen* kann begegnet werden, indem die Notation die inkrementelle Ausbildung von Abstraktionen (z.B. durch Subprozesse) ermöglicht.

Progressive evaluation, Consistency und Diffuseness: Diese Dimensionen sind als weniger kritisch zu sehen oder unterscheiden sich in ihren Implikationen nicht zu typischen Systemen. Eine *progressive Evaluation* ist aufgrund der hohen Informalität skizzierter Prozessmodelle nicht möglich, aber in dieser Phase auch nicht sinnvoll. Jedoch sollte für den Benutzer die Möglichkeit gegeben sein über Feedbackmechanismen den aktuellen Modellierungsstatus (was wurde formal erkannt) zu evaluieren. Zwar besitzt die visuelle Programmierung implizit eine hohe *Verbosität* (= *Diffuseness*), jedoch ist diese bei der Verwendung großformatiger Papiere eher zu vernachlässigen. Vielmehr sollte ausschließlich für die Interaktion benötigte *Verbosität* (z.B. das Skizzieren von Interaktionsgesten) möglichst gering gehalten werden.

Hidden dependencies, Hard mental operations und Error-proneness: *Versteckte Abhängigkeiten* (= *Hidden dependencies*) treten insbesondere dann auf, wenn Aspekte durch das System erkannt werden, die auf dem Papier nicht explizit visualisiert sind (z.B. Verbindungen zwischen zwei Papieren). Verringert werden können sie durch formale Notationselemente, die eine Verbindung anzeigen, und durch die Möglichkeit Abhängigkeiten abzufragen. Der Grad der *mental*en *Herausforderung* (= *Hard mental operations*) bedingt sich vor allem durch die Mächtigkeit der Notation. Mächtige Notationen ermöglichen die Modellierung komplexerer Strukturen und sind so auch schwieriger zu verstehen. Zur Verminderung *mentaler Herausforderungen* bei der Interaktion sollte jedoch die bereits

angesprochene *Sichtbarkeit* zur Verfügung stehender Interaktionsmechanismen und des aktuellen Systemstatus sichergestellt werden. Insbesondere gilt dies, da beide Aspekte einen großen Einfluss auf die *Fehleranfälligkeit* (= *Error-proneness*) des Systems besitzen. Um Fehler zu vermeiden, ist zudem eine klare und eindeutige Gestaltung der Notation (wenige klar unterscheidbare Notationselemente) und eine gute Visualisierung der Syntax wichtig. Letztere muss aufgrund der fehlenden Interaktivität von Papier eindeutig, aber möglichst implizit über zusätzliches Feedback indiziert werden.

4.2 Anforderungen an die Gestaltung von PaSeMod

Die Grundlage für die Anforderungen bilden in erster Linie die durchgeführten Vorstudien, der beschriebene Gestaltungsrahmen und ergänzende Erkenntnisse aus der Literatur. Bezüglich letzteren sind vor allem die Basisanforderungen von EUD (siehe Seite 19), vor allem zur Bereitstellung endbenutzergerechter Konzepte und Interfaces und zur Gestaltung von EUD als Lernerfahrung, sowie papierbasierte Herausforderungen (siehe Seite 30), insbesondere das Design einer durchgängigen Papiernutzung, zu berücksichtigen. Auch wird auf einer ersten Analyse papierbasierter Orchestrierungsherausforderungen, die in einer vorausgegangenen Arbeit [21] beschrieben wurde, aufgebaut. Die Anforderungen sind anhand ihres Gestaltungsfokus aufgegliedert in die vier Bereiche:

- allgemeine Anforderungen (4 Anforderungen),
- Anforderungen an das Modellierungskonzept (5 Anforderungen),
- Anforderungen an die Prozessnotation (4 Anforderungen) und
- Anforderungen an die Feedbackgestaltung (3 Anforderungen).

4.2.1 Allgemeine Anforderungen

Zu Beginn werden allgemeine Gestaltungsanforderungen an das Gesamtkonzept beschrieben.

A1 Papierbasierte Modellierung als Vorstufe computerbasierter Modellierung

Wie in der Positionierung des Systems (Kapitel 4.1.1) geschildert, positioniert sich das System als „erster Schritt“ einer Prozesserstellung. Ein papierbasiert erstelltes Prozessmodell muss daher jederzeit in ein austauschbare, digitales Modell, das mit bestehenden BPM-Werkzeugen weiterverarbeitet werden kann, transformiert werden können und sich so in einen bestehenden Entwicklungsprozesse einbetten. Dies deckt sich mit der von Damm et al. geforderten „Tool Integration“ von Sketching-Systemen [36] und der von Dittrich et al. [38] beschriebenen nötigen Einbettung von EUD-Werkzeugen in existierende Prozesse. Neben den Aspekten des Nutzungskontextes muss daher auch eine Orientierung an bestehenden Modellierungs- und Orchestrierungskonzepten erfolgen.

A2 Flexible Nutzbarkeit des Systems

Die Fallstudie (Kapitel 3.1) hat gezeigt, dass die Prozessmodellierung kein einmaliger und an feste Räumlichkeiten gebundener Prozess ist, sondern iterativ, flexibel und zum Teil ad hoc erfolgt (Fallstudie E2²⁰). Ein System, das diesen Prozess unterstützen soll, muss daher möglichst flexibel, das heißt unabhängig von technischen und räumlichen Restriktionen und ohne lange oder umständliche Rüstzeiten genutzt werden können. Aus technischer Sicht bedeutet dies, dass im Extremfall auch eine Modellierung ohne Verbindung der Digitalstifte mit einem Computer unterstützt werden muss. Auch muss das Modell über einen längeren Zeitraum an verschiedenen Orten weiterentwickelt werden können (siehe Anforderung 8).

A3 Verlässliche Erkennung der Skizzen und Fehlertoleranz

Untersuchungen in der Literatur (z.B. [36, 108]) haben gezeigt, dass eine verlässliche Erkennung von Elementen und Texten eine Grundvoraussetzung für skizzenbasierte Systeme ist. Da an dem Prozess der Modellierung verschiedene und wechselnde Nutzergruppen beteiligt sind (Fallstudie E2), darf diese Erkennung, um flexibel nutzbar zu sein, nicht von einem vorangegangenen Training individueller Skizzierstile abhängig sein. Da eine vollständig korrekte Erkennung, sei es aufgrund mangelnder Erkennungsgenauigkeit oder aufgrund einer ungenauen Eingabe durch den Nutzer, in der Praxis nicht zu gewährleisten ist, muss der Nutzer zudem die Möglichkeit haben falsche Eingaben zu erkennen und sie mit geringem Aufwand zu korrigieren (siehe auch Anforderung 8).

A4 Meta-Domänenorientierung

Ein in der EUD-Literatur häufig geforderter Ansatz ist die Gestaltung aufgaben- und domänenspezifischer Systeme (siehe Seite 20). Die Nähe zur Anwendungsdomäne erleichtert das Verständnis durch Endbenutzer, erfordert jedoch für jede Nutzungsdomäne die spezifische Anpassung von Metaphorik und Visualisierung und schränkt so eine generelle und flexible Nutzbarkeit ein. Auch würde die Einbettung des Systems in den Gesamtprozess der serviceorientierten Modellierung (Anforderung 1) erschwert, da auch nachfolgende Werkzeuge und Entwickler mit verschiedenen Domänenkonzepten umgehen können müssten oder jeweils eine spezifische Überführung der Modelle nötig wäre. Da eine nichtdomänenspezifische Visualisierung unkritisch erscheint (siehe CD-Analyse²¹, Seite 57), muss das System vielmehr meta-domänenspezifische Konzepte (siehe [144]) anbieten, die sich zwischen Entwicklungs- und Anwendungsdomäne positionieren.

²⁰ Mit „Fallstudie Ex“ sei auf das Ergebnis x. der Zusammenfassung der Fallstudienauswertung in Kapitel 3.1.4 verwiesen. Analoges gilt für „Workshop Ex“ und dessen Ergebniszusammenfassung in Kapitel 3.2.5.

²¹ „CD-Analyse“ bezieht sich auf die Diskussion der kognitiven Gestaltungsdimensionen in Kapitel 4.1.4.

4.2.2 Anforderungen an das Modellierungskonzept

Das Modellierungskonzept ist entscheidend für das Verständnis durch Endbenutzer, muss aber gleichzeitig aktuelle, serviceorientierte Modellierungspraxis berücksichtigen.

A5 Kollaborative Modellierung durch mehrere Benutzer

Die Erfassung von Prozessen erfolgt typischerweise in kollaborativen Sitzungen, in die Nutzer mit verschiedenen Kenntnisständen involviert sind (Fallstudie E2). Vor allem Brahe und Schmidt [23] heben die Notwendigkeit zur Einbeziehung von Nutzern in die Gestaltung von serviceorientierten Prozessen hervor. Auch legen verschiedene Untersuchungen nahe, dass Endbenutzer Entwicklungsprozesse oft gemeinsam durchführen (siehe Kapitel 2.2.2). Beim Betrachtungsfeld Geschäftsprozesse kommt zusätzlich hinzu, dass diese oft aus Arbeitsabläufen bestehen, in die verschiedene Benutzer und Benutzergruppen involviert sind.

Das System muss daher die kollaborative Modellierung durch mehrerer Benutzer ermöglichen und unterstützen. Damit mehrere Teilnehmer gleichzeitig oder zeitnah aktiv modellieren können, muss jeder Benutzer direkten Zugriff auf ein Eingabemedium (Digitalstift) haben. Weitere Gestaltungsaspekte für die Unterstützung der Kollaboration ergeben sich aus den von Hundhausen vorgeschlagenen Communicative Dimensions (Seite 22). Jedoch sind von den vorgeschlagenen Dimensionen für auf PaSeMod-Konzept lediglich die Dimensionen *Story Content* (= Darstellung eines Prozesses als „Geschichte“) und *Referencability* (= Referenzierbarkeit einzelner Prozessaspekten) anwendbar.²²

A6 Erfassung des Prozessablaufes sowie von Aktivitätsbeschreibungen

Die serviceorientierte Modellierung beschreibt, im Sinne der Programmierung im Großen, die Zusammenstellung mehrerer eigenständiger, bestehender Services (siehe Seite 9). Kein eigentlicher Bestandteil einer serviceorientierten Modellierung, jedoch ein wichtiger und zeitintensiver Kernaspekt der Gesamtprozesserstellung, ist die Erfassung der Anforderungsdetails der einzelnen Teilschritte (siehe Fallstudie E1). In der Vorstudie wurden die Anforderungen an die Teilschritte in Aktivitätsbeschreibungen (Use-Cases) teilformell beschrieben und mit den Prozessschritten im Modell verknüpft (Fallstudie E1). In Anknüpfung an diese Beobachtung und unter Berücksichtigung der Abstraktionsaspekte (siehe CD-Analyse, Seite 57), soll das System eine klar getrennte Erfassung der Ebenen „Details“ und „Ablauf“, mit der Möglichkeit Verknüpfungen zwischen den Ebenen herzustellen, unterstützen. Eine klare Trennung der Ebenen verringert auch die in der Fallstudie geäußerte Gefahr, dass bei der Modellierung ein zu starker Fokus auf die

²² Die Dimensionen *Modifiability* und *Controllability* können nicht angewendet werden, da die skizzierten Prozesse nicht ausführbar sind. Die Dimension *Programming Saliency* deckt sich mit einer möglichen Domänenorientierung. Eine *Provisionality* ist durch das Skizzieren bereits implizit gegeben.

Aktivitätsdetails gelegt wird (Fallstudie E6). Um Inkonsistenzen bei der Beschreibung der Aktivitätsdetails, wie im Workshop zu beobachten (Workshop E2), zu vermeiden, sollten die Endbenutzer zudem durch eine strukturierte Erfassung der Aktivitätsbeschreibungen unterstützt werden. Dies knüpft an den von Repenning und Ioannidou beschriebenen Designaspekt des „Scaffold typical design“ an [144]. Eine strukturierte Erfassung der Aktivitätsbeschreibungen kann zudem als Grundlage für eine strukturierte Weiterverwendbarkeit (z.B. Identifizierung und Zuordnung konkreter Services) dienen.

A7 Einfache Integration formeller und informeller Modellierungsaspekte

Ein Kernkonzept skizzenbasierter Systeme ist, in Abgrenzung zur Beschränkung „typischer“ Modellierungssysteme, die Integration formeller und informeller Modellierungsaspekte (siehe Kapitel 2.3.3). Wie der Workshop (Workshop E3) und andere Untersuchungen (z.B. Damm et al. [36]) zeigen, verwenden Nutzer bei der Erstellung von Skizzen implizit informelle Aspekte zur Darstellung unstrukturierter, unbekannter oder zu komplexer Modellierungsaspekte. Im Workshop wurden zum Beispiel zusätzliche Annotationen und informelle Erweiterungen fließend mit formellen Notationselementen kombiniert. Durch informelle Skizzen haben auch Benutzer ohne jegliche Modellierungskenntnisse die Möglichkeit ihre Vorstellungen aktiv einzubringen. Auch lassen sich durch informelle Skizzen domänenspezifische Aspekte und sekundäre Notationen, wie sie im Workshop verwendet wurden (Workshop E4 und CD-Analyse, Seite 57), erstellen.

Die Endbenutzer müssen daher sowohl formale Prozesselemente als auch informelle Skizzen erstellen und diese miteinander kombinieren können. Um eine durchgängige Modellierung zu ermöglichen, dürfen Benutzer dabei nicht explizit zwischen formeller und informeller Modellierung unterscheiden müssen. Vielmehr muss das System auf Basis der Eingabe selbstständig differenzieren. Dies erfordert auch eine verlässliche Erkennung durch das System (Anforderung 3) und ein geeignetes Feedback (siehe Anforderung 14-16).

A8 Erweiterbarkeit und Veränderbarkeit skizzierter Prozessmodelle

Die Fallstudie zeigte, dass die Prozesserstellung ein iterativer Prozess ist, in dessen Verlauf ein Prozessmodell verschiedene Veränderungen und Weiterentwicklungen durchläuft (Fallstudie E2). Auch im Workshop war zu sehen, dass im Laufe der Modellierung schrittweise Änderungen am Prozessmodell vorgenommen werden (siehe Kapitel 0). So sollte eine Erweiterung des Modells, eine Formalisierung informeller Prozessaspekte und das Entfernen von Prozesselementen unterstützt werden. Da bei der Anoto-Technologie einmal erstellte Skizzen physisch nicht verändert werden können (siehe Kapitel 4.1.3), muss das System spezielle Konzepte bieten, um die digitale Repräsentation zu verändern (Löschen oder Formalisieren) und die Veränderungen gleichzeitig auf dem Papier sichtbar zu machen (z.B.

Durchstreichen, um ein Element zu löschen). Da ein Papiermodell persistent ist und über mehrere Systemausführungen hinweg bestehen kann, muss das System zudem bereits skizzierte Prozessaspekte speichern und wieder laden können.

A9 Anleitung und aktive Unterstützung der Endbenutzer bei der Modellierung

Eine Grundvoraussetzung zur aktiven Teilnahme an der Modellierung ist das Verständnis der Syntax und Semantik der verwendeten Notation. Der Workshop hat gezeigt, dass Endbenutzer nach einer kurzen Einführung prinzipiell in der Lage waren formale Elemente zu verstehen und einzusetzen (Workshop E1), jedoch nicht immer eine korrekte oder durchgängige Syntax verwendeten (Workshop E2). Wie bereits in der CD-Analyse angesprochen, muss das System daher die Endbenutzer bei der Aneignung und dem Verständnis der Syntax und der Semantik unterstützen. Neben einer Übersicht der verfügbaren Elemente sollten die Benutzer die Möglichkeit haben, über Feedbackmechanismen (kontextbezogene) Informationen zur Verwendung der Elemente zu erhalten. Jedoch lassen sich durch die geforderte technische Mediation nicht alle Aneignungsaspekte unterstützen (z.B. die Abgrenzung und Granularität des Aktivitätsbegriffs). Dies muss, wie im Workshop erfolgreich geschehen, durch den Austausch mit erfahrenen (End-)Benutzer unterstützt werden.

4.2.3 Anforderungen an die Prozessnotation

Damit sich das PaSeMod-Konzept geeignet in die Modellierungspraxis einbetten kann (siehe Anforderung 1), muss sich die Notation an bestehenden serviceorientierten Notationen (siehe Seite 14) orientieren. Als Betrachtungsbasis erscheint BPMN geeignet, da diese sowohl eine konzeptionelle Nähe zu SOA als auch eine geringe Grundkomplexität besitzt.

A10 Differenzierte Mächtigkeit der Prozessnotation

Der Fokus von PaSeMod liegt auf der Unterstützung der (kollaborativen) Erstellung erster strukturierter Prozessskizzen, die geprägt sind durch informelle und kreative Aspekte (Fallstudie E5). Komplexe und stark formalisierte Aspekte (z.B. die Fehlerbehandlung) müssen somit durch die Notation nicht unterstützt werden. Zudem richtet sich das System an eine heterogene Nutzergruppe mit unterschiedlichem Modellierungswissen (siehe Anforderung 5). So müssen auf der einen Seite Endbenutzer, um aktiv teilnehmen zu können, sich die Notation erschließen können. Die in der Fallstudie vorgestellte SPF-Notation ist dafür aufgrund ihrer Komplexität und Mächtigkeit nicht geeignet (Fallstudie E4). Eine Notation mit wenigen Elementen wurde dagegen von Endbenutzern gut verstanden (siehe Workshop E1 und Loser [106]). Auf der anderen Seite findet die Modellierung primär durch Local Developer statt, die bereits Modellierungserfahrung besitzen. Diese sollten ihr Wissen nutzen können, um erweiterte Prozessaspekte auch direkt formal darzustellen. Das System muss

daher eine begrenzte Anzahl von Elementen, welche die Grundkonzepte einer servicebasierten Modellierung abdecken, anbieten.

A11 Skizzierbare und differenzierbare Gestaltung der Notationselemente

Da das Visualisierungsparadigma der Boxes-and-Wires von Endbenutzern akzeptiert wurde (Workshop E1), auf Papier gut umsetzbar ist und sich mit bestehenden serviceorientierten Modellierungsansätzen (siehe Seite 14) deckt, sollte dies die Ausgangsbasis für die Notation des Konzeptes bilden. Zudem sollte die Prozessnotation gut skizzierbar und differenzierbar sein. Eine gute Skizzierbarkeit begründet sich darin, dass die Elemente per Hand gezeichnet werden müssen. Hinsichtlich der Cognitive Dimensions bedeutet dies, dass die Elemente möglichst minimalistisch gestaltet sein sollten (geringe *Verbosität*) und so durch eine einfache Struktur keine Zeichenfehler provozieren (geringe *Fehleranfälligkeit*). Die Differenzierbarkeit der Notationselemente ist wichtig für die Lesbarkeit und die eindeutige Erkennung der Modelle. So sollten die einzelnen Elemente untereinander gut unterscheidbar sein und sich deutlich von möglichen informellen Skizzen abheben. Auch sorgt eine klare, differenzierte Gestaltung der Elemente für einen hohen Wiedererkennungswert und unterstützt die Endbenutzer so beim Erlernen der Notation.

A12 Meta-domänenspezifische Benennung der Notationselemente

Neben der Auswahl und Visualisierung der verfügbaren Prozesselemente spielen die verwendeten Begrifflichkeiten eine wichtige Rolle für das Prozessverständnis der Benutzer (vgl. Loser [106]). Eine schlechte Benennung der Elemente verhindert, wie Loser beschreibt, ihre Übernahmen in den Sprachgebrauch der Benutzer. Gerade in kollaborativen Modellierungsszenarien ist aber ein gemeinsames Verständnis der Modellierungsbegriffe wichtig. Typische Begriffe einer serviceorientierten Modellierung, wie „Service“, „Gateway“ oder „Sequence Flow“ (alle BPMN), entstammen nicht der Begriffswelt der Benutzer und sind für sie daher schwieriger zu „erlernen“. Aufgrund des domänenunspezifischen Ansatzes (siehe Anforderung 4), muss die Notation daher meta-domänenspezifische Begriffe verwenden, die für Endbenutzer verständlich sind und gleichzeitig die Bedeutung des jeweiligen Elementes widerspiegeln.

A13 Methoden zur Visualisierung versteckter Abhängigkeiten

Versteckte Abhängigkeiten entstehen bei papierbasierten Anwendungen vor allem dann, wenn durch das System formale Aspekte erkannt werden, die auf dem Papier nicht explizit oder nur schwach visualisiert sind (z.B. eine Verbindung zwischen zwei Papieren; siehe CD-Analyse, Seite 57). Durch das erforderliche Merken der Abhängigkeiten führen diese zu einem hohen kognitiven Aufwand (= *Hard Mental Operations*) und beeinträchtigen, insbesondere für Endbenutzer, die Nutzbarkeit des Systems. Wie in der CD-Analyse (Seite 57) beschrieben,

muss die Notation daher Elemente für das Visualisieren und Aufdecken der *versteckten Abhängigkeiten* auf dem Papier anbieten.

4.2.4 Anforderungen an die Feedbackgestaltung

Aufgrund der statischen Natur und dem fehlenden dynamischen Feedback von Papier (siehe Kapitel 2.3.2) sind für die Unterstützung der Nutzer externe Feedbackmechanismen nötig. Nachfolgend sind die wichtigsten Anforderungen an deren Gestaltung dargestellt.

A14 Direktes eindeutiges Feedback auf Stiftinteraktion

Wichtige Gestaltungsaspekte von EUD-Systemen sind Funktionen, die den Endbenutzern helfen syntaktische Fehler zu vermeiden und die Semantik des Systems zu verstehen (vgl. Repenning und Ioannidou [144]). Eine große Rolle spielt dabei dynamisches Feedback (z.B. Syntax-Highlightning oder Auto-Vervollständigung) und die Beschränkung auf die Zusammenstellung vorgegebener Elemente. Bei einer papierbasierten Modellierung sind jedoch weder dynamische Veränderungen des Prozessmodells möglich, noch können (und sollen) die Modellierungsmöglichkeiten eingeschränkt werden. Die Benutzer müssen daher über zusätzliche Feedbackmechanismen ein direktes, eindeutiges Feedback (positiv und negativ) auf ihre Eingabe erhalten, das ihnen anzeigt, ob die durchgeführte Eingabe zulässig ist und wie sie erkannt wurde. Wichtig ist zudem, dass das Feedback den Prozess der durchgängigen Papiernutzung möglichst wenig beeinträchtigt.

A15 Kombination aus stiftbasiertem und allgemeinem Feedback

Bei der Verwendung mehrerer Eingabemedien (siehe Anforderung 5) ist zwischen Feedback, das nur für den skizzierenden Benutzer relevant ist und dem erzeugenden Stift eindeutig zuordenbar sein muss, und Feedback, das für alle Prozessteilnehmer wahrnehmbar sein muss, zu differenzieren (siehe auch Seite 38). Zur ersten Kategorie gehört zum Beispiel die Bestätigung formal erkannter Inhalte oder der Wechsel eines Stiftmodus. Um den Modellierungsfluss möglichst wenig zu stören und andere Benutzer nicht zu verwirren, muss dieses Feedback unaufdringlich gestaltet werden. Allgemeines, aufgabenbezogenes Feedback ist in erster Linie für die Visualisierung des aktuellen Modellierungsstatus (siehe CD-Analyse, Seite 57) und für weitergehende Prozessinformationen nötig. Das System muss beide Feedbackarten unterstützen und geeignet miteinander kombinieren.

A16 Flexible Feedbackgestaltung

Eine flexible Systemnutzung (Anforderung 2) verlangt auch eine weitgehend unabhängige Nutzung von räumlich gebundenen Feedbackmechanismen. Jedoch erfordern verschiedene Feedbackarten ein Mindestmaß an Reichtum des Feedbackmediums (siehe auch Anforderung 14 und 15). So lässt sich dynamische Hilfe in der Regel nur visuell sinnvoll darstellen. Im

Extremfall muss ein papierbasiertes System jedoch auch nur mit Stift und Papier (offline) und entsprechend ohne dynamisches Feedback nutzbar sein. Das System muss die Modellierung daher, wenn entsprechende Mechanismen zur Verfügung stehen, um dynamisches Feedback „anreichern“, so dass die Benutzer Feedback entsprechend der zur Verfügung stehenden Medien erhalten. Dabei ist zu beachten, dass ein stiftbezogenes Feedback zur Bestätigung der Eingabe (siehe Anforderung 14) möglichst immer verfügbar sein sollte.

4.2.5 Übersicht der Anforderungen

Tabelle 5 zeigt zusammenfassend eine Übersicht der beschriebenen Anforderung.

Allgemeine Anforderungen	
A1	Papierbasierte Modellierung als Vorstufe computerbasierter Modellierung
A2	Flexible Nutzbarkeit des Systems
A3	Verlässliche Erkennung der Skizzen und Fehlertoleranz
A4	Meta-Domänenorientierung
Anforderungen an das Modellierungskonzept	
A5	Kollaborative Modellierung durch mehrere Benutzer
A6	Erfassung des Prozessablaufes sowie von Aktivitätsbeschreibungen
A7	Einfache Integration formeller und informelle Modellierungselemente
A8	Erweiterbarkeit und Veränderbarkeit skizzierter Prozessmodelle
A9	Anleitung und aktive Unterstützung der Endbenutzer bei der Modellierung
Anforderungen an die Prozessnotation	
A10	Differenzierte Mächtigkeit der Prozessnotation
A11	Skizzierbare und differenzierbare Gestaltung der Notationselemente
A12	Meta-domänenspezifische Benennung der Notationselemente
A13	Methoden zur Visualisierung versteckter Abhängigkeiten
Anforderungen an die Feedbackgestaltung	
A14	Direktes eindeutiges Feedback auf Stiftinteraktion
A15	Kombination aus stiftbasiertem und allgemeinem Feedback
A16	Flexible Feedbackgestaltung

Tabelle 5: Übersicht der Anforderung an PaSeMod

4.3 Aufbau der Grundkonzeptes

Das Gesamtkonzept von PaSeMod untergliedert sich in fünf konzeptionelle Aspekte (siehe Abbildung 14), die im Folgenden kurz vorgestellt werden.

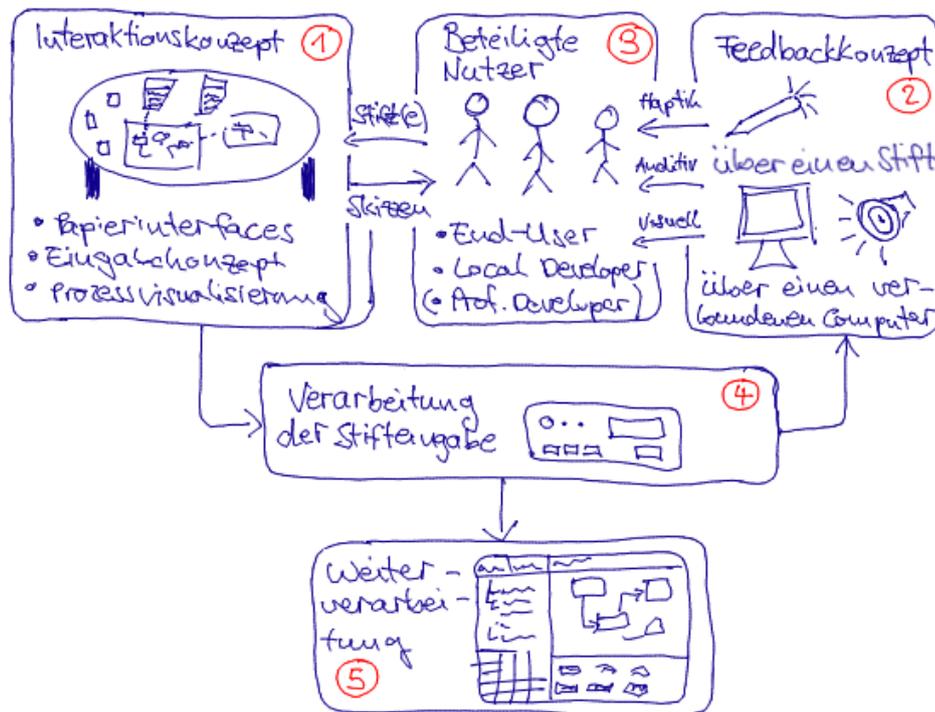


Abbildung 14: Gesamtkonzept von PaSeMod

Der **1. (und umfassendste) Aspekt** ist die rein papierbasierte Interaktion (Kapitel 4.4), deren ausschließliche Eingabemöglichkeit die Digitalstifte sind und die als einziges Feedback die Visualisierung der Stiftskizzen auf dem Papier besitzt. Inhalte des *papierbasierten Interaktionskonzeptes* sind die Gestaltung:

- der Eingabeschnittstellen (Bestimmung und Gestaltung benötigter Papierinterfaces),
- des Eingabekonzeptes über verschiedene Stiftfunktionen und -modi, sowie
- der Visualisierung von Prozessinformationen (Prozessablauf und Prozessdetails) und zusätzlicher Modellierungsinformationen.

Der **2. Konzeptaspekt** ist das *Feedbackkonzept* (Kapitel 4.5). Dieses bietet den Nutzern über das statische Feedback der rein papierbasierten Interaktion hinausgehende, dynamische Feedbackmechanismen, wie Audio und eine visuelle Anzeige. Das dynamische Feedback wird durch die Verarbeitung der Stifteingabe erzeugt und ist daher, im Gegensatz zum papierbasierten Interaktionskonzept, nur verfügbar, wenn der Stift mit einem in der Nähe stehenden Computer verbunden ist.

Den **3. Aspekt** bilden die an der Modellierung *beteiligten Nutzer*. Wie auf Seite 55 ausgeführt, erfolgt die papierbasierte Modellierung kollaborativ durch End-User und Local Developer, wobei letztere gegenüber den End-Usern als Moderator fungieren. Jeder Benutzer kann über einen Stift mit dem System interagieren und erhält kontextabhängig ein allgemeines und/oder stiftbezogenes Feedback.

Der **4. Aspekt** ist die *Verarbeitung der Stifteingabe*. Er umfasst die Verarbeitungslogik des Systems, in der die Stifteingabe verarbeitet, ein digitales Prozessmodell erstellt und dynamisches Feedback erzeugt wird.

Die sich an die papierbasierte Modellierung anschließende, digitale *Weiterverarbeitung* (Kapitel 4.6) des skizzierten Prozessmodells stellt den **5. Aspekt** dar. Die Weiterverarbeitung ermöglicht eine Formalisierung und Konkretisierung des Modells, also eine weitergehende Modellierung nicht auf Papier abbildbarer Aspekte (z.B. Validierung des Modells), sowie die Überführung in ausführbare Prozessmodelle.

4.4 Papierbasiertes Interaktionskonzept

Im folgenden Abschnitt wird die Gestaltung des papierbasierten Interaktionskonzeptes, das heißt ohne die Berücksichtigung dynamischer Feedbackaspekte, beschrieben. Dies umfasst die Gestaltung der benötigten Papierinterfaces, der Visualisierung von Prozessinformationen und des konkreten Eingabekonzeptes.

4.4.1 Papierbasiertes Grundkonzept

Eine erste Übersicht der verschiedenen Interaktionskomponenten zeigt Abbildung 15. Grundsätzlich setzt sich das Interaktionskonzept aus mehreren Stiften und verschiedenen Papiertypen zusammen. Die Papiertypen (siehe Abbildung 15) unterscheiden sich in große *Prozesspapiere* (1), auf denen der Prozessablauf modelliert wird, in vorstrukturierte *Aktivitätsbeschreibungen* (2), die eine detaillierte Beschreibung einzelner Prozessschritte ermöglichen, in zweiseitige *Aktionskarten* (3), die auf der einen Seite eine Übersicht der Prozessnotation zeigen und auf der anderen Seite eine Auswahl verschiedener Modellierungsmodi ermöglichen, und in verschiedenfarbige *Post-Its* (4), die eine Alternative zu skizzierten Prozesselementen darstellen. Nachfolgend wird das Grundkonzept der papierbasierten Interaktion beschrieben.



Abbildung 15: Gestaltung der Papierinterfaces

Verschiedene Prozessebenen

Eine klare Differenzierung der beiden Modellierungsebenen „Prozessablauf“ und „Aktivitätsdetails“ (Anforderung 6) lässt sich durch die Gestaltung von zwei verschiedenen Papiertypen erreichen: *Prozesspapier* und *Aktivitätsbeschreibung*. Auf dem Prozesspapier wird ein Prozess als Abfolge von Aktivitäten, die lediglich benannt und ggf. rudimentär beschrieben werden, skizziert. Auf den Aktivitätsbeschreibungen werden die Aktivitätsanforderungen im Detail beschrieben. Über eine eindeutige Referenzierung (eindeutige Nummer der Aktivitätsbeschreibungen) und eine physische *Verbindungsgeste*²³ lassen sich skizzierte Aktivitäten mit der entsprechenden Aktivitätsbeschreibung verknüpfen. Die beschriebene Auftrennung der Prozessdarstellung auf verschiedene Papiertypen ermöglicht einen besseren Fokus auf den Prozessablauf (Anforderung 6), verhindert die Überfrachtung des Prozessmodells und bietet aus Sicht der Cognitive Dimensions eine bessere *Sichtbarkeit*, *Nebeneinanderstellbarkeit* und *Viskosität* (siehe CD-Analyse, Seite 57) und ist somit besser änderbar und erweiterbar (Anforderung 8). Indem jeder Nutzer auf ein Blatt zugreifen und es bearbeiten kann, unterstützen mehrere Papiere zudem die Kollaboration bei der Prozesserstellung (Anforderung 5).

Individueller Stift pro Benutzer

Wie bereits in den Rahmenbedingung (Kapitel 4.1.3) beschrieben, erfolgt aus Gründen der Konsistenz und einer durchgängigen Papiernutzung die Prozesserstellung ausschließlich stiftbasiert. Damit jeder Benutzer aktiv an der Modellierung partizipieren kann (Anforderung

²³ Gesten, die mit dem Stift erstellt werden, sind bestimmte Formen und Figuren (hier ein einfacher Strich), die auf digitalem Papier aufgezeichnet werden und permanent sichtbar sind. Mit jeder Geste in eine individuelle Bedeutung verbunden.

5), erhält möglichst jeder Teilnehmer einen eigenen Stift. Zusätzlich existiert ein Spezialstift, dem eine besondere Funktion (das „Löschen“ formaler Elemente) zugeordnet ist und der eine andersfarbige Mine besitzt (siehe auch Kapitel 4.4.2). Durch eine zusätzliche Verwendung von (nicht digitalen) Textmarkern können für die Kommunikation wichtige Aspekte hervorgehoben werden. Dies erhöht die *Referenzierbarkeit* (siehe Communicative Dimensions, Seite 62) des Systems und unterstützt so die Kollaboration. Bei Textmarkern besteht, im Gegensatz zu nicht-digitalen Kugelschreibern, zudem nicht die Gefahr, dass die erstellten Annotationen „aus Versehen“ für die weitere Modellierung verwendet werden und so eine Inkonsistenz zwischen dem physischen und dem digitalen Modell entsteht.

Verknüpfung verschiedener (Prozess-)Papiere

Die digitale Verknüpfung zweier Papiere (z.B. Aktivitätsbeschreibung mit Aktivitäten) erfolgt über eine physische Verbindungsgeste, indem mit dem Stift ein durchgehender Strich von einem auf das andere Papier gezogen wird. Um *versteckte Abhängigkeiten*, wie sie in Anforderung 13 und der CD-Analyse (Seite 57) beschrieben wurden, zu vermeiden, muss die digitale Verknüpfung auch auf dem Papier deutlich sichtbar gemacht werden. Dies erfolgt über das Skizzieren zusätzlicher *Notationsgesten* (siehe Kapitel 4.4.3).

Aktionskarten

Aktionskarten dienen der Aneignungsunterstützung, indem sie den Benutzern die verfügbaren Modellierungs- und Interaktionsmöglichkeiten aufzeigen (Anforderung 9) und so eine syntaktische (und z.T. semantische) Hilfe bieten. Sie ersetzen in gewisser Weise die bei Papierinterfaces fehlende Funktionalität von Menü- oder Werkzeugleisten. Die Möglichkeit Modellierungs- und Interaktionsaspekte einfach „nachschaun“ zu können, unterstützt ein schrittweises Erlernen der Modellierung und verringert durch eine bessere *Sichtbarkeit* der dargestellten Aspekte den benötigten *kognitiven Aufwand* (siehe CD-Analyse).

Größe des Interfaces

Die Größenangaben sind keine absoluten Vorgabewerte, sondern dienen als Orientierung für das Verständnis des Konzeptes. Aktivitätsbeschreibungen bzw. Aktionskarten besitzen aufgrund eines festen Inhalts eine feste Größen (circa DIN A6 bzw. A5). Die Größe der Prozesspapiere ist variabel und abhängig vom Umfang des geplanten Prozesses. Für eine kollaborative Modellierung erscheint jedoch die Verwendung großformatiger Papiere (siehe Seite 37) sinnvoll. Damit alle Nutzer aktiv teilnehmen können (Anforderung 5), sollte das Prozesspapier daher, abhängig von der Teilnehmerzahl, mindestens DIN A2 groß sein.

4.4.2 Differenzierung der Stiftmodi

Um eine fließende Modellierung und Integration formeller und informeller Modellierungsaspekte zu gewährleisten (Anforderung 7), erfolgt die Differenzierung formeller und informeller Inhalte implizit durch das System. Der Nutzer muss nicht explizit zwischen verschiedenen Eingabemodi wechseln. Er erhält bei der Erkennung formeller Aspekte (bei einer online Nutzung) lediglich ein kurzes Feedback zur Bestätigung (siehe Seite 81). Neben dieser „normalen Modellierung“ sind folgende weitere Eingabemodi bzw. Stiftmodi nötig:

1. **Informelles Modellieren**, bei dem das System nicht versucht formale Aspekte zu erkennen. Dies ist nötig für das bewusste Skizzieren konzeptioneller Modellaspekten.
2. **Löschen** von formalen Prozesselementen und Verbindungen (siehe Anforderung 8).
3. **Abfragen des Status** von Prozesselementen. Dies erhöht die *Sichtbarkeit* der Syntax (wie wurde ein Element formal erkannt) und deckt *versteckte Abhängigkeiten* auf (wurde eine Aktivitätsbeschreibung zugeordnet) (siehe CD-Analyse, Seite 57).
4. **Aufrufen kontextbezogener Hilfe**. Diese liefert dem Benutzer zusätzliche stiftkontextbezogene, dynamische Hilfe und Anleitung bei der Modellierung (Anforderung 9).
5. Zusätzliche **auditive Annotationen** ermöglichen die Aufzeichnung schlecht skizzierbarer *sekundärer Informationen* und unterstützen so die Kollaboration und Delegation. Ähnlich dem *LiveScribe*-Konzept (Seite 33) können daher parallel zu Stiftskizzen Audioinformationen aufgezeichnet und kombiniert werden.

Eine Möglichkeit die Eingabemodi einer stiftbasierten Eingabe zu wechseln sind Stiftgesten (siehe Seite 37). Dies erscheint für PaSeMod nicht angemessen, da Gesten zum einen auf Papier permanent sichtbar sind und zum anderen einen kognitiven Aufwand für Endbenutzer darstellen, da sie diese erlernen und sich merken müssen. Eine Alternative sind verschiedene Stifttypen und ein Wechsel der Stiftmodi durch Antippen vordefinierter Felder. Da das Löschen von Elementen eine kritische Aktion darstellt und um eine erhöhte *Sichtbarkeit* der Aktion auf dem Papier zu erzeugen, soll das Löschen (respektive das Durchstreichen) bei PaSeMod primär über einen andersfarbigen Stift erfolgen. Dies verhindert auch, dass über eine entsprechende Geste mit dem normalen Stift „aus Versehen“ ein Element entfernt wird. Da diese Aktion seltener benötigt wird, ist vermutlich ein gemeinsamer Löschstift für alle Benutzer ausreichend. Andere Stiftmodi lassen sich durch das Antippen vordefinierter, selbstbeschreibender Felder auf den Aktionskarten auswählen. Diese Möglichkeit sollte aus Gründen der Flexibilität (Anforderung 2; z.B. falls kein Spezialstift verfügbar ist) auch für das Löschen angeboten werden.

4.4.3 Visualisierung des Prozessablaufes

Die Visualisierung des Prozessablaufes beschreibt den Aufbau und die Gestaltung des Prozesspapiers und der Prozessnotation.

Kombinierte Prozessmodellierung mit Post-Its und Skizzen

Das visuelle Prozessmodell wird typischerweise erstellt, indem die Benutzer verschiedene formelle und informelle Elemente direkt auf einem oder mehreren kombinierten Prozesspapieren skizzieren und das System diese zu erkennen versucht. Diese Modellierung über Skizzen hat, wie in den Kapiteln 2.3.3 und 2.3.5 gezeigt, verschiedene Vorteile. Das Skizzieren von Elementen geht schnell und direkt, ist auch für unerfahrene Benutzer intuitiv und die unfertige („low fidelity“) Erscheinung führt zu einem Fokus auf den Inhalt. Jedoch schränken die Skizzen auf Papier die Veränderbarkeit des Prozessmodells (Anforderung 8) ein und führen zu einer hohen *Vorausplanungsnotwendigkeit* (vgl. CD-Analyse, Seite 57), da die Anordnung der Elemente vor dem Zeichnen bestimmt werden muss.

Eine Alternative ist die Verwendung kleiner, beweglicher Eingabeobjekte (siehe Seite 37) in Form von Post-Its an Stelle von gezeichneten Elementen. Aktivitäten und Kommentare werden jeweils durch einen eigenen Post-It-Typ repräsentiert und können durch Aufkleben und Verbinden anhand des Stiftes mit dem Prozesspapier verknüpft werden. Für eine bessere Zuordbarkeit der Post-Its (höhere *Role-expressiveness*) werden Aktivitäten und Kommentare über die Post-It-Farbe differenziert (Aktivitäten sind rot und



Abbildung 16: Verschiedenfarbige Prozess-Post-Its

Kommentare blau, siehe Abbildung 16). Der Prozessfluss (Verbindung der Aktivitäten) sowie weitere Prozesselemente werden weiterhin direkt auf dem Prozesspapier skizziert. Da Post-Its wieder entfernt, umpositioniert oder durch neue ersetzt werden können (= geringere *Viskosität*), führt die Verwendung von Post-Its zu einer besseren Änderbarkeit des Modells (Anforderung 8). Da Aktivitäts-Post-Its gemäß eines „Handlungsablaufes“ (Communicative Dimension des *Story Content*, siehe Seite 62) auf dem Prozesspapier dynamisch an- und umgeordnet werden können, verbessern sie zudem die Kommunikation und Kollaboration (Anforderung 5). Jedoch führt die Modellierung über Post-Its zu einer Einschränkung der beschriebenen unfertigen Erscheinung und besonders der ununterbrochenen und intuitiven Papiernutzung. Auch erschwert sich der Abgleich des physischen und des digitalen Modells.

Aufgrund der Tatsache, dass beide Varianten situationsabhängig Vor- und Nachteile für die Modellierung aufweisen, erscheint die Möglichkeit, beide Alternativen frei kombinieren zu können, sinnvoll.

Mächtigkeit der Notation und verfügbare Modellierungselemente

Aufgrund der Einbettung in die bestehende Praxis, müssen die Modellierungselemente an existierende Prozessnotationen anknüpfen. Bei der Auswahl der formalen Notationselemente ist abzuwägen zwischen einer ausreichenden Mächtigkeit und einer guten Erlernbarkeit und Skizzierbarkeit (siehe Anforderung 10). Um gut erlernbar und skizzierbar (differenziertere Gestaltung bei vielen Elementen nötig) zu sein, soll die Notation nur wenige, syntaktisch und semantisch klar unterscheidbare Basiselemente umfassen.

Kernelemente für eine rudimentäre Modellierung (Anforderung 10) sind *Aktivitäten* und ihre *Verbindung* über Prozessflüsse. Ergänzend ermöglichen formale *Kommentare* eine bewusste Beschreibung von Prozessaspekten und sind über eine Verbindung formalen Elementen zuordenbar. Eine weitergehende Spezifizierung des Modells ist über die typischen Prozesselemente Verzweigungen bzw. Vereinigungen (i.F. *Gateways*) und *Bedingungen* möglich. Über Bedingungen lassen sich zum Beispiel die im Workshop beschriebenen Datenauswahlkriterien (siehe Kapitel 0) modellieren. Zwar stellen diese Elemente eine zusätzliche Herausforderung für Endbenutzer dar, sie ermöglichen aber eine genauere Beschreibung des Prozessablaufes und stellen so eine ausreichende Mächtigkeit (Anforderung 10) des Systems sicher. Innerhalb der Basiselemente können verschiedene Informationen, die für einen Prozessüberblick wichtig sind, skizziert werden. Neben einer informellen Beschreibung ist auch eine weitere formale Differenzierung der Elemente möglich. Beispielsweise kann der Typ einer Verzweigung durch eine Notationsgeste formal spezifiziert werden oder durch Freitext oder ein individuelles Symbol informell beschrieben werden.

Der Prozessfluss wird über die Verbindung der Elemente modelliert. Eine Modellierung des Datenflusses, wie zum Beispiel im Workshop, ist nicht möglich. Zum einen werden Datenaspekte bei der Prozesserstellung typischerweise erst in einer späteren Phase betrachtet (siehe Kapitel 3.1.3) und zum anderen besteht bei einer Kombination von Prozess- und Datenfluss die Gefahr einer Überfrachtung des Modells (insb. aufgrund der Modellierung auf Papier), sowie einer zu hohen Komplexität für Endbenutzer. Der Datenfluss kann jedoch implizit über die Beschreibung der Ein- und Ausgabedaten der Aktivitäten (auf den Aktivitätsbeschreibungen) abgebildet werden (siehe Seite 76).

Gestaltung der Modellierungselemente

Alle Prozesselemente müssen eine gut skizzierbare Form besitzen und untereinander klar differenzierbar sein (Anforderung 11). Eine diesen Anforderungen entsprechende meta-

domänenorientierte (Anforderung 4) Visualisierung der Basiselemente zeigt Abbildung 17. Sie orientiert sich an der Darstellung der Basiselemente typischer prozessorientierter Modellierungssprachen. Die Verwendung unterschiedlicher Farben dient der besseren Unterscheidbarkeit der Elementtypen auf den Aktionskarten und der Differenzierbarkeit der Elementtypen bei einer etwaigen Visualisierung im Zusammenhang mit einem dynamischen Feedback (siehe Kapitel 4.5). Für das Skizzieren der Elemente ist die Farbe jedoch nicht von Bedeutung. Eine weitere Differenzierung der Visualisierungen, zum Beispiel verschiedener Gateway-Arten (wie z.B. bei der SPF-Notation) oder verschiedener Aktivitätstypen (z.B. manuell oder automatisch), ist aus Gründen der Komplexitätsvermeidung und der *Konsistenz* (CD-Analyse, Seite 57) an dieser Stelle nicht sinnvoll. Vielmehr lässt sich der Typ einer Aktivität auf einer zugeordneten Aktivitätsbeschreibung spezifizieren. Bei den Gateway-Elementen ist, ähnlich wie bei anderen Prozessnotationen, eine formale Differenzierung der Verzweigungs- bzw. Verbindungsart über eine textliche Beschreibung (*UND*, *ODER*) oder über Notationsgesten möglich.



Abbildung 17: Gestaltung der Modellierungselemente

Notationsgesten

Notationsgesten sind vordefinierte, elementunspezifische Stiftgesten, mit denen jedem Element zusätzliche Bedeutungen zugewiesen werden können. Dazu werden die Notationsgesten entweder innerhalb eines formalen Elementes skizziert oder mit einem solchen verbunden. Die Notationsgesten können folgende Bedeutungen besitzen:

1. Um die durch die Papiermodellierung (z.B. Verknüpfung mehrerer Papiere) entstehenden *versteckten Abhängigkeiten* sichtbar zu machen (siehe Anforderung 13), müssen Verweise auf die versteckten Informationen, wie eine Verbindungen von Aktivitätsbeschreibungen zu modellierten Aktivitäten, über bestimmte Symbole in der Aktivität auf dem Prozesspapier visualisiert werden.
2. Sie ermöglichen typische und häufig auftretende informelle Aspekte, wie die Spezifikation von vagen (siehe Kunau et al. [89]) oder unbekanntem Aspekten, die sonst über eine *sekundäre Notation* (z.B. ein Fragezeichen) dargestellt werden, formal zu kennzeichnen und so die Kollaboration und Kommunikation anzureichern.
3. Mit ihnen können weitergehende Prozessdetails, wie z.B. der konkrete Typ eines Gateways (z.B. *ODER/UND*), spezifiziert werden.

Gleichsam wie für die Notationselemente gilt, dass die Notationsgesten eine einfache und klare Struktur besitzen müssen (Anforderung 11) und nicht „aus Versehen“ durch informelle sekundäre Annotationen erstellt werden dürfen. Sie setzen sich daher aus einem „Container“-Element und der eigentlichen Geste, welche die Bedeutung trägt, zusammen. Eine Darstellung möglicher Notationsgesten findet sich in Abbildung 18. Die Zahlen in den Klammern ordnen die Gesten den oben beschriebenen Bedeutungen zu.

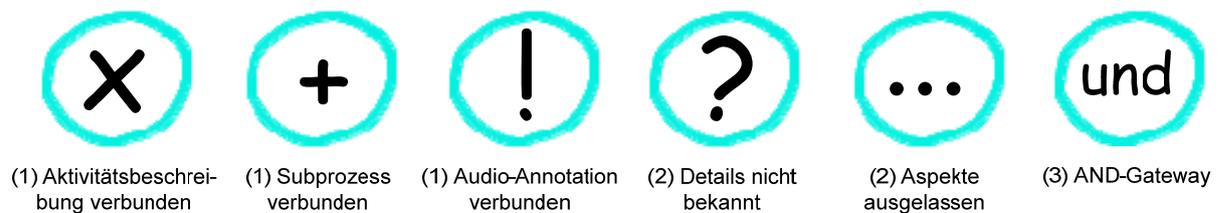


Abbildung 18: Gestaltung der Notationsgesten

4.4.4 Visualisierung der Prozessdetails

Die *Aktivitätsbeschreibungen* bilden eine Detailsicht auf einzelne Prozessaktivitäten und ermöglichen eine detaillierte Erfassung der Prozessanforderungen auf Basis der Aktivitätsdetails (Anforderung 6). Im Gegensatz zu dem Prozesspapier, auf dem nur eine knappe Übersicht der Aktivitäten existiert, können diese hier detailliert beschrieben werden. Dem, durch die Auftrennung entstehenden Problem *versteckter Abhängigkeiten* (ist eine Aktivitätsbeschreibung mit einer Aktivität verbunden?), muss durch Modellierungskonventionen auf Basis von Notationsgesten und durch erweiterte Feedbackmechanismen (Kapitel 4.5) begegnet werden.

Die Aktivitätsbeschreibungen sind aufgebaut als vorstrukturierte Fragebögen mit verschiedenen offenen Fragen, in denen die Anforderungen natürlichsprachig beschrieben werden können, sowie mit symbolbasierten Auswahlfragen zum Ankreuzen. Diese können bei der Modellierung schrittweise ausgefüllt und der Gesamtprozess so iterativ spezifiziert werden. Die Vorstrukturierung folgt dem EUD-Gestaltungsaspekt des *Scaffold typical design* von Repenning und Ioannidou [144] und ermöglicht es auch unerfahrenen Endbenutzern ohne Kenntnis von Service-Strukturen und syntaktischer Aspekte, Aktivitäten teilstrukturiert zu beschreiben. In Anlehnung an die in der Fallstudie beobachtete Anforderungserfassung auf Basis der „System Use-Cases“ (siehe Seite 45), lassen sich für den Aufbau der Aktivitätsbeschreibungen folgende relevante Inhalte einer Aktivität identifizieren: Aktivitätsname, Beschreibung und/oder eine kurze Skizze, Funktionalität, benötigte Daten (Inputvariablen), Ergebnisdaten (Outputvariablen), Typ der Aktivität (automatisch, manuell oder user experienced) und ob die Aktivität aus Teilaktivitäten besteht. Zudem besitzt jede

Aktivitätsbeschreibung eine eindeutige Nummer zur Referenzierung. Abbildung 19 zeigt in verkürzter Form eine daraus resultierende Aktivitätsbeschreibungskarte.

Aktivitätsbeschreibung (#001)		Besteht aus Teilaktivitäten? <input type="checkbox"/>	
Typ der Aktivität?	 (Manuell) <input type="checkbox"/>	 (Maske) <input type="checkbox"/>	 (Automatisch) <input type="checkbox"/>
Name der Aktivität?			
Wie lässt sich die Aktivität beschreiben?			
Welche Funktionen deckt die Aktivität ab?			
Welche Daten werden für die Durchführung der Aktivität benötigt?			
Welche Ergebnisdaten liefert die Aktivität?			

**Abbildung 19: Verkürzte Darstellung einer Aktivitätsbeschreibungskarte
(auf den tatsächlichen Karten sind die freien Antwortfelder mehrzeilig)**

Durch die vorstrukturierte Erfassung können die erstellten Aktivitätsbeschreibungen auf Basis einer Texterkennung in nachfolgenden elektronischen Modellierungsschritten weiter verarbeitet werden. Sie können in wohldefinierte SUCs umgewandelt werden und dann als Grundlage für die Zuordnung konkreter Services dienen.

4.4.5 Erweitertes Interface

Aktionskarten sind circa DIN A6 große (laminierte) Karten, die unterteilt in Vorder- und Rückseite zwei grundlegende Bedeutungen haben. Auf der einen Seite sind die Aktionskarten eine Art Fernbedienungsinterface über das Benutzer mit dem System interagieren können. Primär ermöglicht dies die Auswahl verschiedener Stiftmodi durch das Antippen entsprechender Felder. Weitere Interaktionsmöglichkeiten sind das Speichern und der Export von Prozessmodellen. Vergleichbar mit kleinen Übersichtskärtchen mancher Gesellschaftsspiele, die eine kurze Übersicht der Regeln und verfügbaren Aktionen bieten, zeigen die Aktionskarten auf der anderen Seite eine Übersicht der Notationselemente und Notationsgesten und dienen so als Aneignungsunterstützung für die Endbenutzer. Wie Loser [106] beschreibt, ermöglicht ein „Sichtbarmachen“ von Modellierungsmöglichkeiten (Anforderung 9) es auch unerfahrenen Endbenutzern Prozessmodelle schnell zu verstehen und zu erlernen und dadurch aktiv am Modellierungsprozess zu partizipieren. Zusätzlich lässt sich über die Notationsübersicht die digitale Bedeutung bereits gezeichneter Elemente ändern (Anforderung 8). Zur Differenzierung der Modellierungsfähigkeiten der Nutzer können für verschiedene Nutzertypen zudem verschiedene Aktionskarten angeboten werden. Ähnlich des Konzeptes von pRemote [67, 90], bei dem auf Basis von interaktivem Papier

Fernbedienungsinterfaces selbst gestaltet werden können, ist es denkbar, dass Benutzer am Computer individuelle Aktionskarten selbst erstellen können.

Eine mögliche Gestaltung einer Aktionskarte zeigt Abbildung 20. Dort wurde ein hoher Wert auf eine natürliche und möglichst selbsterklärende Gestaltung der Symbole und Notationselemente gelegt. Zusammengefasst besitzt eine Aktionskarte folgende Inhalte:

- Auswahl von Stiftmodi und Kommandos zur Interaktion mit dem System. (Linke Seite der Abbildung 20)
- Kurzübersicht über die Notation mit der Möglichkeit über diese die Bedeutung gezeichneter Elemente zu ändern. (Rechte Seite der Abbildung 20)
- Kurzübersicht über die Notationsgesten. (Nicht abgebildet)

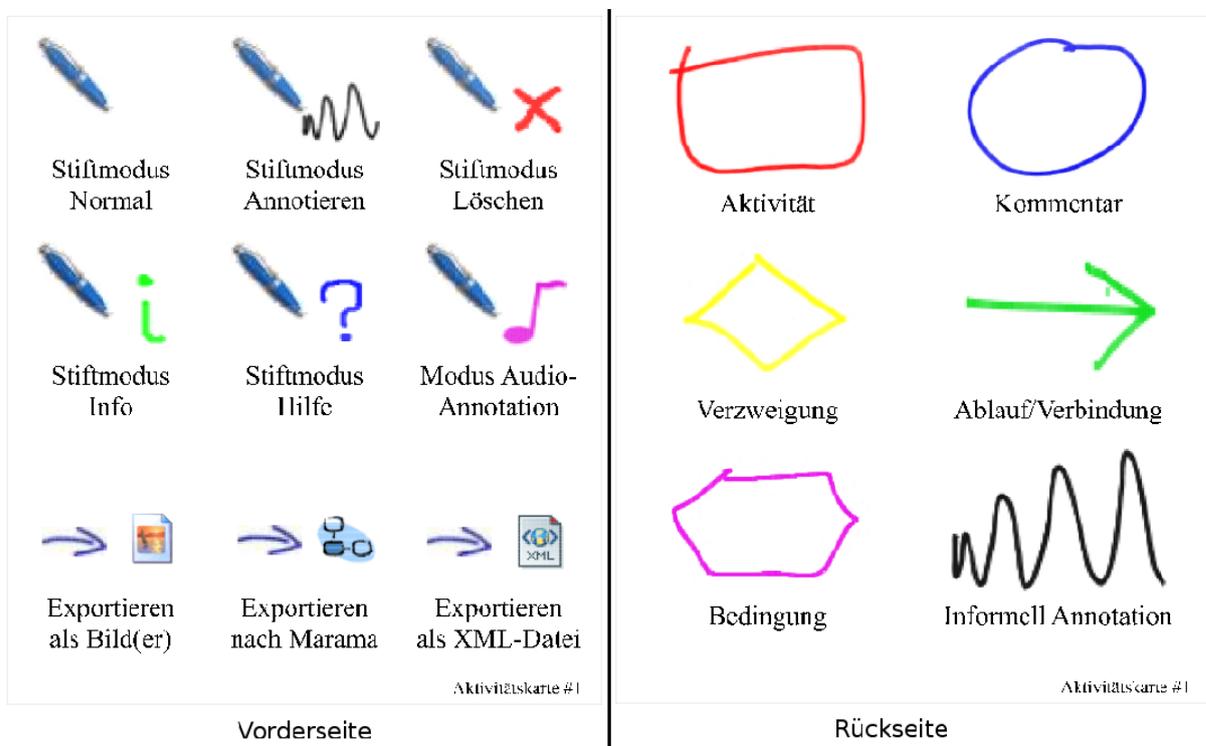


Abbildung 20: Gestaltung der Aktionskarten – Interaktionselemente auf der Vorderseite (links) und Modellierungselemente auf der Rückseite (rechts)

4.4.6 Beispielhafter Ablauf der Modellierung

Um einen Einblick in die Funktionsweise des vorgestellten Interaktionskonzeptes zu geben, wird anhand verschiedener Aktionen eine beispielhafte Modellierung mit PaSeMod beschrieben.

Erstellung und Konkretisierung des Modells

Die Prozesserstellung beginnt, indem in einer gemeinsamen Diskussion die benötigten Aktivitäten bestimmt und für diese jeweils ein Post-It erstellt wird. Durch das Aufkleben der

erstellten Aktivitäten wird der Prozessablauf dann als eine Art „Storyboard“ auf dem Prozesspapier visualisiert. Wenn einzelne Prozessaspekte feststehen, werden die Post-Its durch einen kurzen Strich von einer Ecke des Post-Its auf das Prozesspapier in das Prozessmodell eingebunden. Um versteckte Abhängigkeiten zu vermeiden, wird die Verbindung auf dem Prozesspapier mit einer Notationsgeste und der Post-It-Nummer annotiert. Alternativ können Aktivitäten auch direkt auf dem Prozesspapier skizziert werden, was jedoch die Veränderbarkeit einschränkt. Im nächsten Schritt wird der Prozessfluss spezifiziert, indem die Aktivitäten miteinander verbunden werden. Die Verwendung von Gateway-Elementen zwischen den Aktivitäten ist abhängig vom gewünschten Prozessfluss und dem vorhandenen Modellierungswissen. Zwar führt ein Gateway-Element zu flexibler veränderbaren Prozessflüssen, beeinträchtigt jedoch unter Umständen die Lesbarkeit des Gesamtprozesses. Zur weiteren Spezifikation des Prozessablaufes können Bedingungen mit dem Prozess verknüpft werden, indem sie gezeichnet und mit einem Prozessfluss oder einer Aktivität (anstelle aller eingehenden Prozessflüsse) verbunden werden. Das System versucht (im online Modus) während des Zeichnens auf dem Prozesspapier formale Elemente durch die gezeichneten Formen zu erkennen. Wenn ein Element erkannt wurde, gibt das System dem Nutzer ein Feedback (siehe Kapitel 4.4). Nicht erkannte Skizzen sind informelle Annotationen, bis sie gegebenenfalls konkretisiert werden.

Erfassung von Prozessanforderungen

Die Spezifikation von Prozessanforderungen anhand der Aktivitätsbeschreibungen erfolgt unabhängig von der Modellierung des Prozessablaufes und kann sowohl vor, als auch nach der Erstellung der Aktivitäten im Prozess stattfinden. Die Zuordnung einer Aktivitätsbeschreibung zu einer konkreten Aktivität im Prozess erfolgt, indem die Aktivitätsbeschreibung neben die Aktivität gelegt und beide mit einem kurzen Strich verbunden werden. Zur Aufdeckung der *versteckten Abhängigkeit* wird über eine Notationsgeste in der Aktivität zusätzlich die Nummer der Aktivitätsbeschreibung vermerkt.

Veränderung des Modells

Jede Änderung am physischen Modell, also an erstellten Skizzen und an angehefteten Post-Its, muss mit der digitalen Repräsentation abgeglichen werden. Da eine Erkennung durch das System nicht immer automatisch möglich ist (z.B. beim Verschieben oder Entfernen von Post-Its), müssen Benutzer diese Änderungen dem System manuell signalisieren. Das Löschen eines Elements erfolgt durch das Durchstreichen des entsprechenden Elements mit dem Löschstift oder mit einem normalen Stift im Löschmodus. Veränderungen einer Verknüpfung von Papieren erfolgt durch das Löschen und Neuerstellen der Verknüpfung. Verbindungen werden analog zu ihrer Erstellung gelöscht, nur dass der Stift sich im Löschmodus befinden muss. Falsch erkannte Elemente können über die Aktionskarten

geändert werden. Um den digitalen Elementtyp zu ändern wird das physische Element mit dem Stift angetippt und dann auf der Aktionskarte der neue Elementtyp ausgewählt.

Statusinformationen, kontextbasierte Hilfe und Auditive Annotation

Statusinformationen zu Elementen und die interaktive Hilfe lassen sich aufrufen, indem der Stift über eine Aktionskarte in den entsprechenden Modus versetzt wird und dann mit diesem das entsprechende Element, zu dem Informationen bzw. Hilfe angezeigt werden soll, angetippt wird. Auf die Ausgestaltung des Feedbacks der Informationen bzw. Hilfe wird in Kapitel 4.5.4 eingegangen. Über die Hilfe können Benutzer weiterführende Erklärungen der Notationselemente und der Aktivitätsbeschreibungen abrufen. Audio-Annotationen werden über das Antippen eines Elementes und gedrückt Halten des Stiftes erstellt und mit einer Notationsgeste visualisiert. Abgespielt werden sie durch ein einfaches Antippen der erstellten Notationsgeste.

Verknüpfen von Teilprozessen

Das Verknüpfen verschiedener Teilprozesse, die sich auf verschiedenen Prozesspapieren befinden, ist nötig, wenn einzufügende Prozessteile aus Platzgründen auf einem anderem Prozesspapier modelliert werden oder wenn bewusst voneinander getrennte modellierte Teilprozesse miteinander verknüpft werden sollen. Eine Verknüpfung zweier Teilprozesse wird erstellt, indem auf einem der Prozesspapiere eine Aktivität, die den anderen Teilprozess repräsentiert, skizziert wird und diese Aktivität dann über eine Verbindungsgeste mit dem anderen Prozesspapier verbunden wird. Um die Verknüpfung auf dem Papier zu visualisieren, wird die Aktivität mit einer entsprechenden Notationsgeste ergänzt.

4.5 Feedbackkonzept

Als Ergänzung zur papierbasierten Interaktion, die nur ein statisches Feedback (den auf dem Papier skizzierten Prozess) liefert, ermöglicht das Feedbackkonzept ein dynamisches Feedback (z.B. eine visuelle Anzeige) über zusätzliche Feedbackmechanismen.

4.5.1 Feedbackarten

Anforderung 16 beschreibt, dass die Feedbackgestaltung flexibel und möglichst unabhängig von der Verfügbarkeit technischer Hilfsmittel benutzbar sein sollte. Im Extremfall soll daher auch eine Modellierung unterstützt werden, ohne dass die Stifteingabe durch einen Computer verarbeitet wird (offline Nutzung). Die Eingabe wird in diesem Fall dann verarbeitet, wenn der Stift mit einem Computer verbunden wird. Diese Nutzungsform beschränkt jedoch die kollaborative Modellierung, da aus Gründen der Synchronisierung nur mit einem Stift modelliert werden kann und ermöglicht auch kein direktes Feedback auf die Stifteingabe (z.B. ob die Eingabe formal erkannt wurde). Wenn möglich, sollte die Modellierung daher in

Verbindung mit einem Computer, der die Stifteingabe verarbeitet und ein differenziertes und direktes Feedback ermöglicht, einhergehen (online Nutzung, siehe Anforderung 16).

Die Feedbackmechanismen lassen sich grundlegend in stiftbezogenes und in allgemeines Feedback (siehe Kapitel 2.3.6) unterteilen. Das stiftbezogene Feedback ermöglicht ein individuelles Feedback für jeden Digitalstift und eignet sich daher zur Bestätigung individueller Aktionen. Konzeptionell wird in Anlehnung an Liao et al. [99] und *Fly Pentop* [95] angenommen, dass als stiftbezogenes Feedback mehrfarbige LEDs, die Vibration des Stiftes und eine Audio-Ausgabe über den Stift zur Verfügung stehen. Da letztere eine nicht nur stiftbezogene Wirksamkeit besitzt, ist sie jedoch sparsam einzusetzen. Zusätzlich ist über einen verbundenen Computer ein allgemeines auditives (Ton und Sprache) und ein visuelles (per Bildschirm oder Beamer) Feedback möglich. Beim visuellen Feedback ist zu beachten, dass es die durchgehende Stiftnutzung unterbricht (der Benutzer muss vom Papier aufsehen) und nicht in jeder Nutzungssituation verfügbar ist (z.B. zugeklappter Laptop oder Bildschirm auf einem anderen Tisch). Eine Übersicht der Feedbackmechanismen zeigt Tabelle 6.

Stiftbezogenes Feedback	Allgemeines Feedback
<ul style="list-style-type: none"> • Auditiv: Ton und Sprache • Visuell: verschiedenfarbige LEDs • Haptisch: Vibration 	<ul style="list-style-type: none"> • Auditiv: Ton und Sprache (zu vernachlässigen, da auch über den Stift) • Visuell: Bildschirm oder Beamer

Tabelle 6: Mögliche Feedbackmechanismen für das Konzept.

4.5.2 Identifikation von benötigtem und sinnvollem Feedback

Dynamisches Feedback lässt sich gemäß der Anforderungen 9 und 14 in unmittelbares Feedback auf die Stifteingabe und in Feedback zur Unterstützung der Modellierung unterteilen. Für diese beiden Kategorien wird im Folgenden beschrieben, welches Feedback aufgrund des vorgestellten Interaktionskonzeptes, der Analyse der Cognitive Dimensions und der identifizierten Anforderungen für PaSeMod nötig oder sinnvoll erscheint.

Direktes eindeutiges Feedback auf Stifteingabe

Im Gegensatz zu GUI-basierten Systemen, in denen Benutzer durch die GUI ein direktes Feedback erhalten, ist hierfür in papierbasierten Systemen ein zusätzliches Feedbackdesign notwendig. Ein eindeutiges Feedback ist wichtig, um den aktuellen Modellierungsstatus zu erkennen und den Benutzer beim Erlernen der Systemnutzung, im speziellen einer formalen Modellierung, zu unterstützen (Anforderung 14). So können die Nutzer durch ausbleibendes Feedback erkennen, dass ihre Eingabe nicht oder nicht formal erkannt werden konnte. Auch die Sichtbarkeit des aktuellen Systemzustands spielt für die Benutzbarkeit des Systems eine große Rolle (CD-Analyse, Seite 57). In dieser Kategorie lassen sich folgende Feedbackfunktionen identifizieren:

- Bestätigung, dass formale Prozesselemente und Notationsgesten erkannt wurden. Dies unterstützt das Erlernen der formalen Prozessmodellierung.
- Bestätigung, dass eine Verknüpfung zwischen verschiedenen (Prozess-)Papieren erfolgreich erstellt wurde.
- Anzeige des aktuellen Stiftmodus, sowie Bestätigung eines Wechsels des Stiftmodus.
- Bestätigung von expliziten Aktionen, die über andere Stiftmodi ausgeführt werden. Dazu zählt zum Beispiel das Löschen von Elementen oder Verbindungen.
- Hinweis, dass eine Aktion nicht eindeutig zugeordnet werden konnte oder ungültig ist.

Feedback zur Unterstützung der Modellierung

Dies umfasst Funktionen, die für die Modellierung nicht zwingend erforderlich sind, aber die Modellierung, besonders für Endbenutzer, erleichtern und sie zusätzlich anleiten (Anforderung 9). Hierunter fällt besonders Feedback zur Anzeige von Prozessinformationen und von kontextbezogener Hilfe.

- Abfrage des formalen Status von Prozessaspekten. Dies ermöglicht den Nutzern zu evaluieren, ob und wie ein Element formal erkannt wurde. (*progressive Evaluation*)
- Zusätzliche Erklärung von Elementen und Interaktionsmöglichkeiten, sowie Hilfe bei der Erstellung von Elementen.
- Abfrage, ob für ein Element eine Verbindung zu anderen Prozesspapieren besteht. So können auf dem Papier *versteckte Abhängigkeiten* zum Beispiel durch das visuelle Einblenden verknüpfter Aktivitätsbeschreibungen aufgedeckt werden.
- Abspielen von erstellten auditiven Annotationen.
- Detaillierte Ausgabe oder Anzeige von Modellierungsfehlern. Der Benutzer kann dynamisch darauf hingewiesen werden, wenn er beispielsweise eine Verknüpfung erstellt, die nicht zulässig ist (z.B. Verknüpfung von Bedingung und Gateway).

4.5.3 Gestaltung des Feedbacks

Die konkrete Gestaltung des Feedbacks basiert auf einer intuitiven und konsistenten Kombination von stiftbezogenem und allgemeinem Feedback, um zum einen stiftbasierte Aktionen zu bestätigen und zum anderen allgemein die Modellierung zu unterstützen.

Bestätigung von Aktionen

Feedback zur Bestätigung von Aktionen muss einfach zu identifizieren und gleichzeitig möglichst unaufdringlich und implizit gestaltet sein (Anforderungen 14 und 15). Damit die Benutzer progressiv ein formales Prozessmodell erstellen können, muss das Feedback zudem jederzeit (im online Modus) und direkt verfügbar sein (Anforderung 16). Da es sich um Feedback auf individuelle Stiftaktionen handelt, wird es durch ein stiftbezogenes Feedback bestätigt. Tabelle 7 zeigt eine mögliche, konkrete Gestaltung des Bestätigungsfeedbacks.

Feedbackfunktionen	Gestaltung des Feedbacks
<ul style="list-style-type: none"> • Bestätigung einer formalen Erkennung. • Bestätigung eines Wechsels des Stiftmodus. 	Vibration des Stiftes kombiniert mit einem Blinken der LED in der der Funktion zugeordneten Farbe (Abbildung 20).
<ul style="list-style-type: none"> • Bestätigung einer erfolgten Verknüpfung. • Bestätigung erfolgreicher expliziter Aktion. 	Vibration des Stiftes kombiniert mit einem weißen Blinken (= erfolgreiche Aktion) der LED.
Anzeige des aktuellen Stiftmodus.	LED durchgängig, entsprechend der Farbe des Modus auf der Aktionskarte (Abbildung 20).
Aktion nicht eindeutig erkannt oder Fehler.	Piepton + LED rot blinken.

Tabelle 7: Feedbackgestaltung für die Bestätigung von Aktionen

Unterstützung der Modellierung

Aufgabenbezogenes Feedback wird explizit über den entsprechenden Funktionsaufruf des Informations- oder des Hilfemodus ausgelöst. Um Verständnisprobleme und einen *hohen kognitiven Aufwand* (siehe CD-Analyse, Seite 57) bei der Interpretation des Feedbacks zu vermeiden, sollten sich die Funktionen durch ein reiches Feedback auszeichnen. Daher liegt es nahe für dieses Feedback eine visuelle Anzeige zu verwenden. Bei der kollaborativen Modellierung hat diese Form von Feedback zudem den Vorteil, dass sie im Gegensatz zu einer auditiven Ausgabe unaufdringlich für andere Benutzer ist und parallel mehrere Ereignisse anzeigen kann. Da die Unterstützung jedoch flexibel und nicht zwingend abhängig von der Verfügbarkeit einer visuellen Anzeige sein soll (Anforderung 16), ist wichtiges unterstützendes Feedback auch über stiftbezogene Feedbackmechanismen (LED und Audio) verfügbar. Eine mögliche, konkrete Ausgestaltung des Feedbacks beschreibt Tabelle 8.

Feedbackfunktionen	Gestaltung des Feedbacks
Feedback zu dem formalen Status skizzierter Prozesselemente.	LED blinkt entsprechend der Elementtypfarbe und Spracheausgabe des Elementtyps über TTS.
Abspielen auditiver Annotationen.	Auditiv Ausgabe.
<ul style="list-style-type: none"> • Feedback zu Modellierungsfehlern. • Anzeige von Verbindung zu anderen Prozesspapieren. 	Primär über einen Bildschirm (dort kann direkt Lösungshilfe gegeben werden), ansonsten ein kurzer Hinweis per Sprachausgabe.
<ul style="list-style-type: none"> • Visualisierung des digitalen Modells. • Zusätzliche Erklärungen und Assistenz bei der Erstellung von Elemente. 	Feedback über einen Bildschirm oder Beamer.

Tabelle 8: Feedbackgestaltung für die Unterstützung der Modellierung

4.5.4 Gestaltung des visuellen Feedbacks

Um die konzeptionelle Distanz zwischen der Papiermodellierung und der digitalen Visualisierung gering zu halten, muss das visuelle Feedback möglichst natürlich gestaltet werden und sich an der Papiervisualisierung orientieren. Orientiert an dem natürlichen Entwicklungssystem *HANDS* von Landay und Myers [93] sollte das grafische Interface am physischen Arbeitsumfeld (z.B. ein runder Tisch auf dem verschiedene Blättern angeordnet sind) ausgerichtet werden. Allgemeines Feedback könnte anhand verschiedenfarbiger Sprechblasen, die jeweils dem auslösenden Stift zugeordnet sind, angezeigt werden. Um an das physische Modell anzuknüpfen, sollte für die Visualisierung des digitalen Prozessmodells zudem eine low-fidelity Darstellung (siehe Seite 35) verwendet werden.

4.6 Weiterverarbeitung

Im Anschluss an die papierbasierte Modellierung können das skizzierte Prozessmodell und die Aktivitätsbeschreibungen in ein Austauschformat exportiert werden, um sie dann mit einem GUI-basierten BPM-Werkzeug weiter zu konkretisieren. Um sich in das zweistufige Modell aus papier- und GUI-basierter Modellierung (siehe Kapitel 4.1.1) einzugliedern, muss ein solches BPM-System verschiedene Funktionen erfüllen. So sollte ein idealtypisches Weiterbearbeitungssystem unter anderem folgende Anforderungen berücksichtigen:

- Typische Funktionalität einer serviceorientierten Prozessmodellierung (z.B. Validierung des Modells und Transformation in ein ausführbares Prozessmodell).
- Möglichkeit zur Formalisierung („Beautification“) des Prozessmodells.
- Integration formeller und informeller Prozessaspekte und verknüpfter Informationen.
- Natürliche Repräsentation sekundärer Notationselemente und informeller Annotationen (besonders Kommentare und die Prozessanordnung).
- Verarbeitung der Aktivitätsbeschreibungen.
- Endbenutzergerechte Konzepte für eine weitergehende Modellierung durch Local Developer. Ansätze dazu finden sich in den in Kapitel 2.2.3 vorgestellten Systemen.

4.7 Zusammenfassung des PaSeMod-Konzeptes

Aufgrund der Komplexität der Prozesserstellung wurde das Konzept in einem zweistufigen Vorgehen positioniert. Dieses setzt sich zusammen aus einer papierbasierten Modellierung informeller und teilformeller Aspekte und einer anschließenden, rechnergestützten Formalisierung und Weiterverarbeitung der Modelle. Daran anknüpfend wurde ein beispielhaftes Nutzungsszenario für PaSeMod beschrieben.

Da vergleichbare Konzepte bisher fehlten, war eine wichtige Voraussetzung für die Gestaltung des Konzeptes die Identifikation von Anforderungen. Auf Basis der Literatur und der Vorstudie konnten, unterteilt in allgemeine Anforderungen, Anforderungen an das

Modellierungskonzept, Anforderungen an die Prozessnotation und Anforderungen an die Feedbackgestaltung, insgesamt 16 Anforderungen identifiziert werden. Zur Gestaltung eines gebrauchstauglichen Systems wurden zudem kognitive Gestaltungsdimensionen für PaSeMod diskutiert.

Aufbauend auf den identifizierten Anforderungen wurde ein Konzept zur papierbasierten, serviceorientierten Modellierung für Endbenutzer entwickelt und vorgestellt. Gemäß den formulierten Forschungszielen (siehe Kapitel 2.4) wurde der Schwerpunkt auf die Gestaltung eines geeigneten Interaktionskonzeptes (unterteilt in ein papierbasiertes Interaktionskonzept und ein Feedbackkonzept) und eines geeigneten (implizit behandelten) Modellierungskonzeptes gelegt. Eine Kernherausforderung war dabei insbesondere die Integration endbenutzerorientierter, papierbasierter und serviceorientierter Gestaltungsaspekte in einem geeigneten Konzept.

Das **Modellierungskonzept** unterteilt den Gesamtprozess in einen „Prozessablauf“ und in die „Aktivitätsdetails“, die beide separat erfasst werden aber miteinander verbunden werden können. Dies soll eine nicht gewollte Vermischung der beiden Ebenen verhindern. Um die Notation für Endbenutzer verständlich zu gestalten, wurde ihre Mächtigkeit auf die fünf Basiselemente Aktivität, Kommentar, Verbindung, Gateway und Bedingung beschränkt. Die Visualisierung der Elemente baut auf dem Grundparadigma der Boxes-and-Wires auf und orientiert sich an bestehenden Modellierungssprachen, vordergründig an BPMN. Die Erfassung der Aktivitätsdetails erfolgt im Gegensatz zu dem Prozessablauf, der auf einem Prozesspapier visuell skizziert wird, durch eine natürlichsprachige, textuelle Beschreibung auf vorstrukturierten Aktivitätsbeschreibungen. Dies soll auch unerfahrenen Endbenutzern ohne Kenntnis von Service-Strukturen ermöglichen, Services strukturiert zu beschreiben.

Das **papierbasierte Interaktionskonzept** sieht vor, dass sich jeder Nutzer über einen eigenen Stift aktiv in eine kollaborative Modellierung einbringen kann. Typischerweise versucht das System dabei formelle Notationselemente zu erkennen. Allerdings lassen sich über andere Stiftmodi differenzierte Aktionen (z.B. Löschen oder Hilfe abrufen) durchführen. Der Prozessablauf wird auf einem großen Prozesspapier skizziert, Aktivitätsdetails werden durch Aktivitätsbeschreibungen festgehalten. Aktivitätskarten bieten durch eine Übersicht der Notationselemente eine Aneignungsunterstützung und ermöglichen eine Auswahl verschiedener Stiftmodi und verschiedene Exportmöglichkeiten. Um dem Problem der fehlenden Entfernbarkeit erstellter Skizzen zu begegnen, sieht das Konzept als Alternative zu dem Skizzieren von Aktivitäten und Kommentaren die Verwendung von wieder entfernbaren Post-Its vor. Notationsgesten sollen vor allem die Verständlichkeit des Modells erhöhen, indem sie durch das System digital erkannte Aspekte auf dem Papier visualisieren.

Das **Feedbackkonzept** begegnet über zusätzliche Feedbackmechanismen dem Problem eines fehlenden dynamischen Feedbacks von Papier. Dazu kombiniert es Feedback über den Stift (über LEDs, Vibration und Audio) und ein allgemeines Feedback über einen verbundenen Computer (visuelle Anzeige über einen Bildschirm und Audio). Weiterhin beschreibt es die Feedbackgestaltung differenziert nach Feedback zur Bestätigung von Aktionen und nach Feedback zur Unterstützung der Modellierung.

Zur Verdeutlichung des entwickelten Interaktionskonzeptes wurde ein beispielhafter Ablauf eines Modellierungsprozesses mit PaSeMod vorgestellt. Abschließend wurden kurz Anforderungen an die rechnerbasierte Weiterverarbeitung der erstellten Modelle beschrieben.

Zusammenfassend bietet das vorgestellte PaSeMod-Konzept eine ganzheitliche Antwort auf die in Kapitel 2.4 formulierte Forschungsfrage der Gestaltung einer papierbasierten und endbenutzerorientierten (kollaborativen) Modellierung von Geschäftsprozessen in serviceorientierten Architekturen.

5 Realisierung von PaSeMod

Um die in Kapitel 4 entwickelten Konzepte und Gestaltungsideen zu evaluieren, wurde PaSeMod prototypisch umgesetzt. In diesem Kapitel wird zuerst die grundlegende Architektur des Systems und die verwendeten Technologien vorgestellt und anschließend die prototypische Entwicklung dargestellt.

5.1 Architektur des Systems

Als Grundlage für die Umsetzung, werden im Folgenden das Architekturkonzept und darauf aufbauend die technischen Anforderungen des Systems beschrieben.

5.1.1 Architekturkonzept

Basierend auf dem Designkonzept wurde für die Realisierung ein aus sieben Komponenten bestehendes Architekturkonzept entwickelt (siehe Abbildung 21). Bevor die einzelnen Komponenten vorgestellt werden, wird anhand der Verarbeitung eines skizzierten Elementes beispielhaft illustriert, wie die einzelnen Komponenten im Gesamtsystem einzuordnen sind und wie sie miteinander kommunizieren.

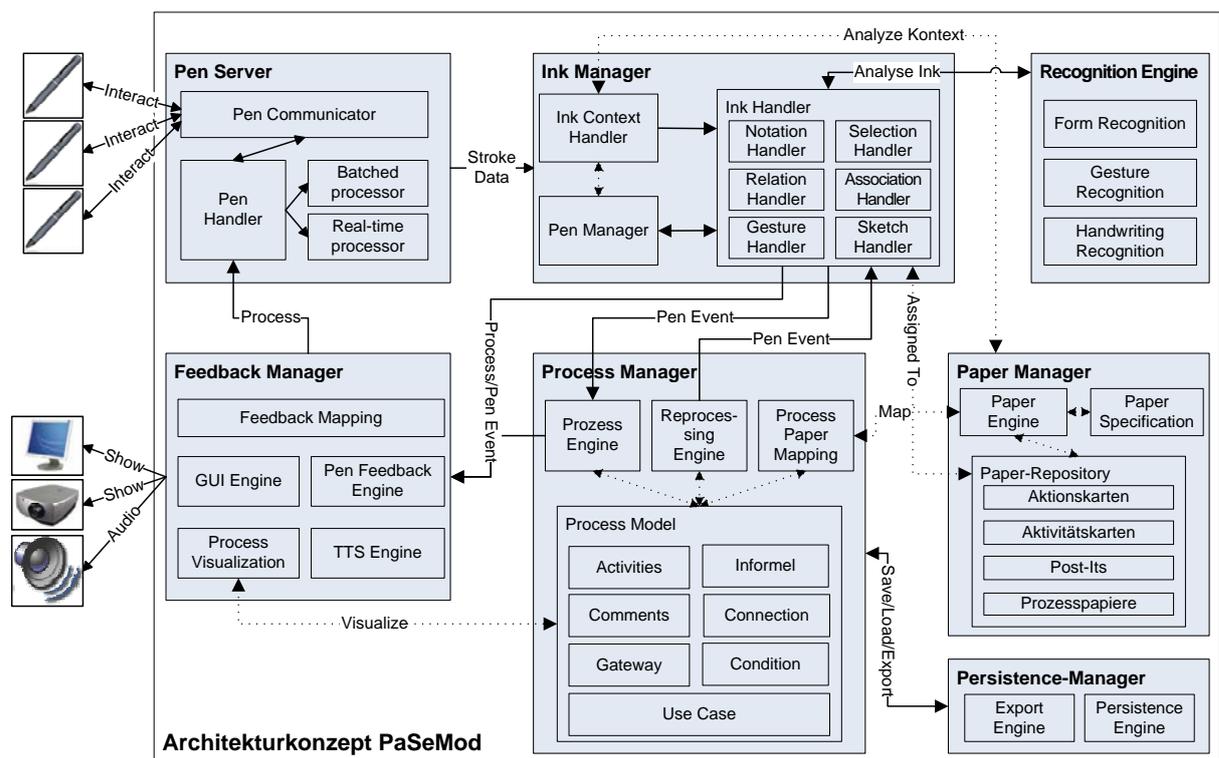


Abbildung 21: Architekturkonzept von PaSeMod

Wenn eine Skizze erstellt wird, werden die Punktinformationen vom Stift an die *Pen Server* Komponente übermittelt, von dieser in Positionsinformationen umgewandelt und

anschließend an den *Ink Manager* weitergeleitet. Dieser erkennt über einen Aufruf des *Paper Managers*, dass es sich um eine Skizze auf einem Prozesspapier handelt und erfragt daher bei der *Recognition Engine*, ob es sich bei dem Strich um ein formales Element handelt. Das Ergebnis übergibt er dem *Process Manager*, der das Element in das digitale Prozessmodell einfügt und über den *Paper Manager* den Papierbereich, auf dem das Element skizziert wurde, mit dem digitalen Element verknüpft. Abschließend teilt er dem *Feedback Manager* das Ereignis mit. Dieser erzeugt daraufhin das entsprechende Feedback und gibt es entweder als allgemeines Feedback direkt aus oder übergibt es bei stiftbasiertem Feedback dem *Pen Server*, der es an den Stift weiterleitet. Der *Persistence Manager* wird nur für das Laden, Speichern oder Exportieren aufgerufen.

Komponente: Pen Server (low-level Stiftinteraktion)

Die *Pen Server* Komponente ist das Bindeglied zwischen den Digitalstiften und dem System und ist für die low-level Stiftinteraktionen zuständig. Über den *Pen Communicator* erfolgt die Kommunikation mit einem Digitalstift. Er nimmt den per Bluetooth oder USB übertragenen Datenstrom entgegen, verarbeitet ihn und sendet auf Anforderung Feedback an den Stift zurück. Der *Pen Handler* erkennt mehrere Digitalstifte, verwaltet diese und führt die Eingabedaten mehrerer Stifte zusammen. Der *Batched* und der *Real-time processor* verarbeiten die Eingabe logisch und setzen sie in einheitliche Anoto-Punktkoordinaten um.

Komponente: Ink Manager (high-level Verarbeitung der Stifteingabe)

Der *Ink Manager* sorgt für die high-level Verarbeitung der Stifteingabe. Dazu nimmt der *Ink Context Handler* eingehende *Stroke*-Daten²⁴ entgegen und analysiert ihren Kontext. Dieser wird bestimmt durch den Papiertyp, den Stift und den aktiven Stiftmodus. Anschließend wird ein *Pen*-Ereignis erzeugt und damit die entsprechenden *Ink Handler* aufgerufen. Die *Ink Handler* sind jeweils mit bestimmten Kontexten (z.B. Regionen auf einem Papier) verknüpft und verarbeiten entsprechende *Pen*-Ereignisse: Der *Notation Handler* erkennt und verarbeitet formale Notationselemente. Notationsgesten werden durch den *Gesture Handler* behandelt. Für die physische Verbindung verschiedener Papiere ist der *Relation Handler* zuständig, für die logische Verbindung formal erkannter Prozesselemente der *Association Handler*. Der *Selection Handler* behandelt „Antipp“-Ereignisse bei einem Aufruf von Aktionen (z.B. Auswahl von Stiftmodi über eine Aktionskarte) und der *Sketch Handler* ist für alle anderweitigen Eingaben (z.B. das Ausfüllen von Aktivitätsbeschreibungen) zuständig. Ausgehend von der Ereignisbehandlung werden dann Komponenten wie das Prozess- oder Feedbackmanagement aufgerufen. Der *Pen Manager* verwaltet die beteiligten Digitalstifte

²⁴ Ein *Stroke* ist ein abgeschlossener Strich, an dessen Ende der Stift abgesetzt wurde.

logisch und stellt zum Beispiel dem *Ink Context Handler* Stiftinformationen (Stifttyp und aktuellen Stiftmodus) zur Verfügung.

Komponente: Recognition Engine

Die *Recognition Engine* stellt verschiedene Erkennungsfunktionen für die Verarbeitung der Stifteingabe bereit. Sie wird von den *Ink Handlern* aufgerufen und übergibt die Stroke-Daten abhängig vom Kontext den entsprechenden Erkennungsroutinen. Die *Handwriting Recognition* verarbeitet handgeschriebenen Text, vor allem beim Ausfüllen der Aktivitätsbeschreibungen und beim Export. Die Erkennung der Notationselemente erfolgt über die *Form Recognition*. Da diese auf einzelne Striche beschränkt ist, erweitert die *Gesture Recognition* die Erkennung auf zusammengesetzte Gesten (z.B. Notationgesten).

Komponente: Feedback Manager

Der *Feedbackmanager* wird über *Pen-* und *Process-*Ereignisse aufgerufen, bestimmt über die *Feedback Specification* das genaue Feedback für das jeweilige Ereignis und löst es aus. Das konkrete Feedback erfolgt durch die *GUI Engine* oder die *Pen Feedback Engine*. Letztere erzeugt logisch das Feedback für Digitalstifte und sendet es zur technischen Übertragung an den *Pen Server*. Unterstützt wird die Feedbackerzeugung durch die Visualisierung des Prozessmodells (*Process Visualization*) und die Spracherzeugung über die *TTS Engine*.

Komponente: Process Manager

Die Komponente des *Process Managers* verknüpft das digitale und das physische Prozessmodell. So wird bei Änderungen am physischen Modell über *Pen-*Ereignisse die *Process Engine* aufgerufen. Diese dient als Proxy, der die Ereignisse überprüft und filtert, Veränderungen an das *Process Model* weitergibt und die Zuordnung von physischen Papierregionen und logischen Prozesselementen über das *Process Paper Mapping* regelt. Das *Process Paper Mapping* stellt über eine Zuordnungstabelle die Verbindung von Papierregionen zu digitalen Prozesselementen und umgekehrt her. Dies ist nötig, um den Kontext eingehender Striche zu bestimmen (z.B. ob ein Löschruch innerhalb der Region eines formalen Prozesselementes erstellt wurde). In manchen Fällen ist es nicht nur nötig eingehende Ereignisse zu verarbeiten, sondern auch früher erstellte Skizzen neu zu analysieren (z.B. wenn eine Aktivität um schon bestehende Skizzen herum gezeichnet wird). Dazu ermittelt die *Reprocessing Engine* diese Skizzen, erzeugt ein entsprechendes *Pen-*Ereignis und übergibt dieses zur Neuauswertung an den *Ink Handler*. Die zentrale Verwaltungskomponente des digitalen Prozesses bildet das *Process Model*. Dort werden, für jedes Prozesspapier getrennt, alle formalen Elemente, informellen Skizzen, erstellte Aktivitätsbeschreibungen (*Use-Cases*), sowie deren Verbindungen verwaltet.

Komponente: Paper Manager

Die *Paper Manager* Komponente kümmert sich um die Verwaltung der interaktiven Blätter. Das *Paper Repository* trägt den verschiedenen Papiertypen Rechnung und verwaltet die verwendeten Papiere, Papierlayouts und Regionen. Über die *Paper Engine* kann dann anhand der Positionsinformationen der Anoto-Technologie festgestellt werden, auf welchem Papier und in welcher Region ein Strich erstellt wurde. Dies stellt die Grundlage für die Kontextbestimmung des *Ink Managers* und die Zuordnung des *Process Managers* dar. Wenn zu einer Strichinformation noch kein Papier vorhanden ist, wird anhand der *Paper Specification* der Papiertyp bestimmt und das neue Papier dem Gesamtprozess hinzugefügt.

Komponente: Persistence Manager

Der *Persistence Manager* sorgt für den Export des Modells und für eine zentrale Speicherung des Systemzustands. Über die *Export Engine* wird das skizzierte Prozessmodell inklusive der Aktivitätsbeschreibungen für die Weiterverarbeitung in verschiedene Austauschformate, wie zum Beispiel XML oder anwendungsspezifische Formate, exportiert. Die *Persistence Engine* sichert für spätere Weiterbearbeitungen (Anforderung 8) den gesamten Systemzustand (Prozessmodell einschließlich der Papier- und Mapping-Daten) und liest ihn wieder ein.

5.1.2 Technische Anforderungen

Auf Basis der fachlichen Anforderungen, des Designkonzeptes und des Architekturkonzeptes ergeben sich folgende technische Anforderungen:

- Die Möglichkeit, die Skizzen sowohl direkt an das System zu „streamen“, als auch sie erst im Stift zu speichern und nachträglich zu verarbeiten.
- Die Unterstützung der parallelen Erfassung mehrerer Stifte.
- Die Differenzierung zwischen Text und Notation, sowie eine zuverlässige Handschrift- und Gestenerkennung.
- Die Möglichkeit, das Papierinterface flexibel zu gestalten und es drucken zu können.
- Die Verwendung eines Stiftes mit möglichst reichen Feedbackmöglichkeiten.
- Die Unterstützung einer Sprachausgabe.

5.2 Technologieauswahl und verwendete Technologien

Im Folgenden werden die Basistechnologien für die Umsetzung von PaSeMod vorgestellt. Primär sind dies die interaktive Papiertechnologie von Anoto und ein entsprechendes Entwicklungsframework. Diese werden ergänzt um Technologien zur Handschrift- und Gestenerkennung, sowie zur Sprachausgabe.

5.2.1 Interaktive Papiertechnologie

In Kapitel 2.3.1 wurden verschiedene interaktive Papiertechnologien beschrieben. Wie in den Rahmenbedingungen in Kapitel 4.1.3 ausgeführt, erscheint von diesen die Anoto-Technologie am besten geeignet. Sie besitzt neben den dort bereits aufgeführten Vorteilen der Location-Awareness und einer hohen Flexibilität auch eine hohe technische Zuverlässigkeit und eine gute Unterstützung durch verschiedene Entwicklungsframeworks.

Anoto-Technologie

Die Anoto-Technologie setzt sich zusammen aus einem Papier mit einem speziellen aufgedruckten Punktmuster und einem um digitale Funktionen angereicherten Stift (siehe Abbildung 8, Seite 29), der über eine integrierte Kamera das Punktmuster erkennt.

Bei dem Anoto-Papier handelt es sich um normales Papier, das mit einem sehr feinen Punktmuster bedruckt ist. Dieses besteht aus vielen kleinen Punkten, die 0,3 mm auseinander liegen und so das Papier leicht grau erscheinen lassen. Indem die einzelnen Punkte jeweils leicht von einem gleichmäßigen Raster abweichen (siehe Abbildung 22) und der Stift jeweils einen Bereich von 6 mal 6 Punkten erfasst und analysiert, lässt sich insgesamt ein Bereich von 60 Millionen Quadratmetern eindeutig identifizieren. Die Aufzeichnung erfolgt durch eine im Stift integrierte Infrarotkamera, die 50 Aufnahmen pro Sekunde erstellt. Durch einen integrierten Prozessor werden die Informationen dann in absolute Punktkoordinaten mit x- und y-Werten umgewandelt. Durch die Zuordnung von Punktbereichen zu bestimmten Seiten und Bereichen, lassen einzelne Striche eindeutig einzelnen Papierseiten zuordnen. Der Beginn und das Ende eines Striches werden über einen Drucksensor im Stift erfasst. Diese Daten werden dann auf dem Stift gespeichert und anschließend per Bluetooth oder über eine USB-Dockingstation auf einen Rechner übertragen (Batched-Modus). Bei einigen Stiften und mit speziellem Papier können sie alternativ auch direkt per Bluetooth an einen Computer gesendet werden (Streaming-Modus).

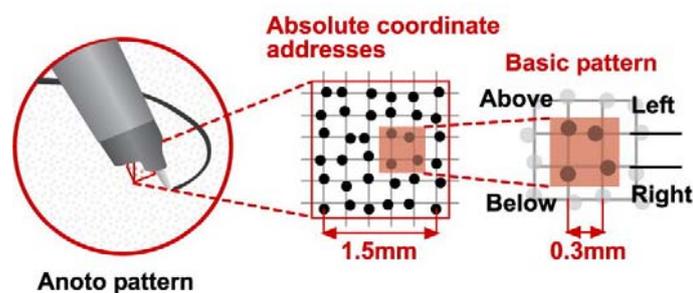


Abbildung 22: Aufbau des Anoto Punktmusters

Das Anoto-Papier kann erworben oder mit einer Lizenz selbst gedruckt werden. Aufgrund der sehr feinen Anordnung der Punkte ergeben sich aber komplexe Anforderungen an die

Drucktechnologie (siehe [7-9]). So lassen sich mit Tintenstrahldrucken nur bedingt und unter sehr spezifischen Bedingungen Anoto-Papiere erstellen. Da die Punkterkennung durch die Infrarotkamera auf Basis bestimmter Tintenbestandteile erfolgt, kann jedoch Drucktinte ohne diese Bestandteile (z.B. farbige Tinte) problemlos mit dem Punktmuster kombiniert werden.

Anoto-Digitalstifte

Da Anoto keine eigenen Digitalstifte produziert, sondern diese nur lizenziert, stehen verschiedene Stiftmodelle mit unterschiedlichen Ausstattungen zur Auswahl. Tabelle 9 zeigt eine Übersicht der Hersteller und verfügbaren Stiftmodelle. Von diesen unterstützen lediglich der Logitech io2 Pen Commercial, in Verbindung mit einem speziellen Punktmuster und der io2 Software und der Nokia SU-1B, mit einer speziellen Firmware-Version, aber auf jedem Anoto-Papier, das Streaming von Daten. Wie bereits in Kapitel 2.3.6 angesprochen, unterstützt keiner der Stifte bewusst ansteuerbare dynamische Feedbackaspekte, wie zum Beispiel eine Vibration des Stiftes oder farbige LEDs.

Hersteller	Stiftmodelle
Livescribe	Pulse Smartpen
Logitech (jetzt Destiny)	io Pen (*), io2 Pen (*), io2 Pen Commercial (*)
Maxell	DP-101 (*), DP-201
Nokia	Digital Pen SU-1B (*), Digital Pen SW-27
Sony Ericsson	Chatpen CHA-30 (*)

Tabelle 9: Digitalstifthersteller (Anoto-Technologie) und Stiftmodelle²⁵

5.2.2 Interaktives Papier- und Sketching-Framework

Die Entwicklung von papier- bzw. skizzenbasierten Systemen wird durch verschiedene Entwicklungsframeworks vereinfacht. Tabelle 10 zeigt die wichtigsten Vertreter beider Kategorien. Der Fokus der Papierframeworks liegt auf der Erfassung der Stifteingabe und der Zuordnung der Stifteingabe zu interaktiven Papieren. Sketching-Frameworks bieten Funktionen für die Erkennung und Verarbeitung von Skizzen. Auf eine detaillierte Beschreibung der einzelnen Frameworks wird aus Platzgründen verzichtet. Details finden sich auf den angegebenen Webseiten und in den referenzierten Arbeiten. Eine detaillierte Beschreibung und Gegenüberstellung des Anoto Development Kits und des Paper Toolkits (dort unter dem Namen R3) findet sich zudem bei Küstermann [90].

²⁵ Die mit einem * gekennzeichneten Modelle werden nicht mehr vertrieben.

Interaktive Papierframeworks	
<i>Anoto Development Kit</i> ²⁶	Basisframework von Anoto, das für eine Batched-Verarbeitung unumgänglich ist. Jedoch bietet es lediglich low-level Verarbeitungsfunktionen und unterstützt kein Streaming.
<i>iPaper</i> ²⁷ [126, 155]	High-level Framework, das über eine Bibliothek aktiver Komponenten verschiedene Verarbeitungsfunktionen mitliefert. Das Framework ist Open Source, unterstützt jedoch kein Streaming.
<i>Logitech io2 Software SDK</i> ²⁸	Framework primär für die Entwicklung von Plug-Ins für die Logitech io2-Software und für den Export von unverarbeiteten Ink-Daten (low-level Framework). Integriert ist jedoch bereits eine Schrifterkennung. Das Logitech SDK ist Closed Source, unterstützt aber eine Streaming-Eingabe.
<i>Paper Toolkit</i> ²⁹ [178]	Open Source Framework für die integrierte Verarbeitung von Batched- und Streaming-Eingabe, das auch mehrere Stifte gleichzeitig unterstützt. Als High-level Framework beinhaltet das Paper Toolkit bereits verschiedene rudimentäre Verarbeitungs-, Ereignisbehandlungs- und Interaktionsfunktionen.
Sketching-Frameworks (alle Open Source)	
<i>InkKit</i> ³⁰ [141]	InkKits Stärke liegt in einer fortgeschrittenen und flexibel anpassbaren Erkennungsroutine. Durch die fortgeschrittene Erkennung müssen Benutzer nicht zwischen verschiedenen Eingabemodi wechseln. Über Add-Ins kann InkKit zudem um Export- und Interaktionsmöglichkeiten erweitert werden.
<i>Marama Sketch</i> ³¹ [56]	Erweiterung für das Meta-Diagramm-Werkzeug Marama [57], das eine Transformation von Skizzen in frei definierbare (Marama-)Diagramme ermöglicht. Stärken des Systems sind eine flexible Erkennung und Transformation sowie die Integration informeller und formeller Inhalte.
<i>SATIN</i> ³² [72]	Der Fokus von SATIN liegt auf der Bereitstellung eines flexiblen Sketching-Interfaces, durch verschiedene Sichten, Funktionen zum Drehen und Zoomen von Skizzen sowie die Integration mit anderen Eingabemechanismen.

Tabelle 10: Übersicht interaktive Papier- und Sketching-Frameworks

Von den Papierframeworks kommen aufgrund der Streaming-Unterstützung lediglich das Logitech SDK und das Paper Toolkit in Frage. Aufgrund der Unterstützung mehrerer Stifte,

²⁶ <http://partner.anoto.com/cldoc/aop15.htm>

²⁷ <http://www.globis.ethz.ch/research/paper>

²⁸ Die Logitech Digitalstifttechnologie wurde an Destiny (<http://www.destinyplc.co.uk>) verkauft und wird nicht weiter vertrieben.

²⁹ <http://hci.stanford.edu/paper/>

³⁰ <http://www.cs.auckland.ac.nz/~beryl/inkkit.html>

³¹ <https://wiki.auckland.ac.nz/display/csidst/Marama+Sketch>

³² <http://dub.washington.edu/projects/satin/>

vorhandenen high-level Funktionen und einer sehr guten Anpassbarkeit (da Open Source) wurde für die konkrete Realisierung das Paper Toolkit ausgewählt. Die Entscheidung für das Paper Toolkit impliziert auch die Verwendung des Nokia SU-1B Digitalstiftes. Von den Sketching-Frameworks kommt lediglich das InkKit in Frage. Wegen seiner geschlossenen Systemarchitektur und einer schlechten Integrierbarkeit (das Paper Toolkit ist in Java realisiert, das InkKit in .NET) erscheint eine Verknüpfung mit der Paper Toolkit jedoch zu komplex. Marama Sketch erscheint interessant für die Weiterverarbeitung der Skizzen.

5.2.3 Sonstige Technologien

Weitere wichtige Technologien für die Realisierung des Systems betreffen die Erkennung formaler Prozessaspekte und von Handschrift, sowie eine Sprachausgabe.

Handschrift- und Gestenerkennung

Die Handschrift- und Gestenerkennung stellt eine entscheidende Komponente für die zuverlässige Erkennung (Anforderung 3) dar. Kernpunkte sind die zuverlässige und schnelle Erkennung und Differenzierung von Text und Gesten ohne nutzerbezogenes Training des Erkennungsmechanismus. Zudem müssen Gesten (Notationselemente und -gesten) für ein entsprechendes Feedback sofort erkannt werden („eager recognition“). Eine Schrifterkennung ist während der Modellierung dagegen nicht nötig. Textskizzen werden daher unverarbeitet gespeichert und erst beim Export analysiert („lazy recognition“). Zusammengesetzte Gesten spielen eine untergeordnete Rolle, da sie nur für Pfeile (Schaft plus Spitze) und Notationsgesten zwingend erforderlich sind. Um die Komplexität des Prototyps zu reduzieren, wird daher auf die Erkennung zusammengesetzter Gesten verzichtet und ein Single-Stroke Verfahren verwendet. Dieses ermöglicht ein sofortiges Feedback, weil nicht der nächste Strich oder eine gewisse Zeit abgewartet werden muss und reduziert die Differenzierungskomplexität. Die Differenzierung der Eingabearten folgt dem ganzheitlichen Klassifizierungsansatz von Patel et al. [137], der gegenüber anderen Ansätzen eine hohe Zuverlässigkeit zeigt.

Für die Gestenerkennung wird der *\$I Recognizer* von Wobbrock et al. [170] verwendet. Dieser ermöglicht bei einer geringen Sprachmächtigkeit eine sehr schnelle und, gegenüber dem weit verbreiteten *Rubines*-Algorithmus (siehe Kapitel 2.3.6), auch ohne nutzerspezifisches Training eine zuverlässige Erkennung. Die Handschrifterkennung erfolgt über das für Windows-Systeme frei verfügbare *Tablet PC Recognizer Pack*³³ des *Microsoft Tablet PC Platform SDK*³⁴. Dieses bietet auch ohne Training eine schnelle und differenzierte Erkennung von Texten.

³³ <http://www.microsoft.com/austria/windowsxp/tabletpc/muiprodguide.msp>

³⁴ <http://msdn.microsoft.com/en-us/library/ms840465.aspx>

Sprachausgabe

Um die Sprachausgabe dynamisch zu gestalten wird ein Sprachsynthese-System verwendet. Wie die umfangreiche Übersicht kommerzieller und nichtkommerzieller deutscher Sprachsynthese-Systeme von Burkhardt [27] zeigt, existieren in diesem Bereich eine Vielzahl von Lösungen. Die Schwierigkeit ergibt sich in der Integration in PaSeMod. Java bietet als Lösung die *Java Speech API*³⁵, für die jedoch keine deutschsprachige Implementierung verfügbar ist. Eine Alternative bietet die in Microsoft Windows bereits integrierte *Microsoft Speech API*³⁶. Tests haben gezeigt, dass diese für kleine Sprachbausteine, wie sie in PaSeMod benötigt werden, gute Ergebnisse liefert.

5.3 Implementierung von PaSeMod

Im Folgenden wird die konkrete prototypische Realisierung des PaSeMod-Konzeptes vorgestellt. Zu Beginn wird auf Einschränkungen gegenüber dem vorgestellten Konzept eingegangen und anschließend das realisierte System beschrieben.

5.3.1 Einschränkung der zu realisierenden Systemaspekte

Um eine Evaluation des Gesamtkonzeptes zu ermöglichen, wurde bei der Gestaltung der Komplexitätsreduktion Wert darauf gelegt, möglichst alle Kernkonzepte der papierbasierten Modellierung umzusetzen und lediglich die Funktionstiefe und ergänzende Funktionen einzuschränken. Eine Ausnahme stellt, aufgrund technischer Restriktionen, die Feedbackgestaltung dar. Da die in Frage kommenden Digitalstifte keine ansteuerbaren Feedbackmechanismen besitzen (siehe Seite 92), muss sämtliches Feedback über einen Computer erfolgen und das Feedbackkonzept dementsprechend angepasst werden.

Da es sich bei den Einschränkungen oft um Detailspekte handelt, sind im Folgenden lediglich die wichtigsten Einschränkungen beschrieben:

- *Keine Batched-Verarbeitung.* Der Batched-Modus ermöglicht einen flexiblen Einsatz des Systems, entspricht aber nicht der intendierten, „typischen“ Nutzung (siehe Kapitel 4.1.2) und hat zudem keinen Einfluss auf das Kernkonzept.
- *Kein persistentes Sichern des Systemzustandes.* Ebenso wie der Batched-Modus dient das Speichern der Flexibilität des Systems und ist für eine grundlegende Evaluation nicht erforderlich. Dies betrifft jedoch nicht die Exportfunktionalität.
- *Beschränkung auf die Stiftmodi „Normal“, „Informell“ und „Löschen“.* Die Stiftmodi „Info“ und „Hilfe“ können bei der Evaluation durch entsprechende Moderation ersetzt

³⁵ <http://java.sun.com/products/java-media/speech/>

³⁶ <http://www.microsoft.com/speech/speech2007/speechdevarticle.msp>

werden und würden einen unverhältnismäßig hohen Implementierungsaufwand erfordern, der im Rahmen dieser Arbeit nicht erbracht werden kann.

- *Reduktion der Notationsmächtigkeit um die fortgeschrittenen Modellierungselemente Gateway und Bedingung.* Diese würden lediglich die Verarbeitungskomplexität stark erhöhen, sind aber für eine Basismodellierung nicht zwingend erforderlich.
- *Keine formale Erkennung der Notationsgesten.* Analog zu Gateways und Bedingungen werden Notationsgesten aus Komplexitätsgründen nicht formal erkannt.
- *Einschränkung der Erkennungsmächtigkeit.* Dies folgt dem vorangegangenen Punkt und schlägt sich in verschiedenen Detailentscheidungen wieder. Vorrangig genannt sei der bereits beschriebene Verzicht auf die Erkennung zusammengesetzter Gesten.

5.3.2 Realisiertes PaSeMod-System

Abbildung 23 zeigt einen Ausschnitt des entwickelten PaSeMod-Systems während einer Modellierungssituation. Im Folgenden wird kurz auf die Realisierung der einzelnen Teile des Systems eingegangen.



Abbildung 23: Nutzung von PaSeMod

Realisierung der Papierinterfaces

Die Papierinterfaces wurden weitgehend so umgesetzt, wie im Konzept in Kapitel 4.4 (siehe Abbildung 15, Seite 70) beschrieben. Probleme bereitete jedoch die Realisierung von großformatigen Prozesspapieren und von Post-Its. Beim Druck großformatiger Papiere stellte sich das Problem, dass Tintenstrahl-Plotter nur unter sehr bestimmten, nicht eindeutig rekonstruierbaren Voraussetzungen ein korrektes Punktmuster erzeugen, Laser-Plotter nicht verfügbar waren und ein Offset-Druck zu teuer war. Ursprünglich war geplant, Prozesspapiere in der Größe DIN A1 zu erstellen. Trotz intensiver Recherchen und einer Vielzahl an Tests

mit unterschiedlichen Tintenstrahl-Plottern konnte kein zufriedenstellendes Ergebnis erzielt werden. Auch die Möglichkeit, einen großen, durchgehenden Patternraum zu "sezieren", Teilflächen zu drucken und ihn wieder zusammzusetzen, war unter Kostengesichtspunkten (Preise ab 2.000 Euro) für das prototypische System nicht tragbar. Das größtmögliche verfügbare Format war daher mit Laserdruck produziertes DIN A3 Papier (siehe Abbildung 23). Ein ähnliches Problem ergab sich beim Druck der Post-Its, die nur über zertifizierte Anoto Druckereien erhältlich waren. Die dort, aufgrund des verwendeten Offset-Drucks und zusätzlich nötiger Anoto-Lizenzen, anfallenden Kosten waren ebenfalls nicht tragbar³⁷. Für den Prototyp wurde daher eine provisorische Lösung gewählt, indem Post-Its mit verschiedenfarbigen Aufklebern, die mit Anoto Punktmuster bedruckt waren, beklebt wurden. Die Aktivitätsbeschreibungen wurden, um den haptischen Charakter hervorzuheben und zu vermeiden, dass sie „labbrig“ wirken, auf dickeres Papier gedruckt.

Realisierung des Feedbacks

Als Ersatz für das nicht realisierbare Feedback, direkt über die Digitalstifte (siehe Seite 92), wird eine Kombination aus auditivem und visuellem computerbasierten Feedback eingesetzt. Das aufmerksamkeitsregende Element der Vibration wird über einen Benachrichtigungston abgebildet. Kritische Hinweise (z.B. Modellierungsfehler) erfolgen über eine Sprachausgabe. Beide haben jedoch den Nachteil, dass sie nicht stiftspezifisch sind. Detaillierte Rückmeldungen und Statusinformationen erfolgen daher über stiftspezifische Feedbackboxen am rechten Rand des grafischen Interfaces (siehe Abbildung 24). In diesen werden über selbstbeschreibende Symbole und zusätzliche textuelle Beschreibungen, der aktuelle Stiftstatus (Symbole analog zu den Aktionskarten) sowie die letzte Modellierungsaktion (siehe Abbildung 25) angezeigt. Die Zuordnung der Feedbackboxen zu den Stiften erfolgt über einen eindeutigen Farbcode mit dem Stifte und Boxen versehen sind.

³⁷ Die Produktion von je 100 Blöcken in zwei verschiedenen Farben würde circa 5.975 Euro kosten.

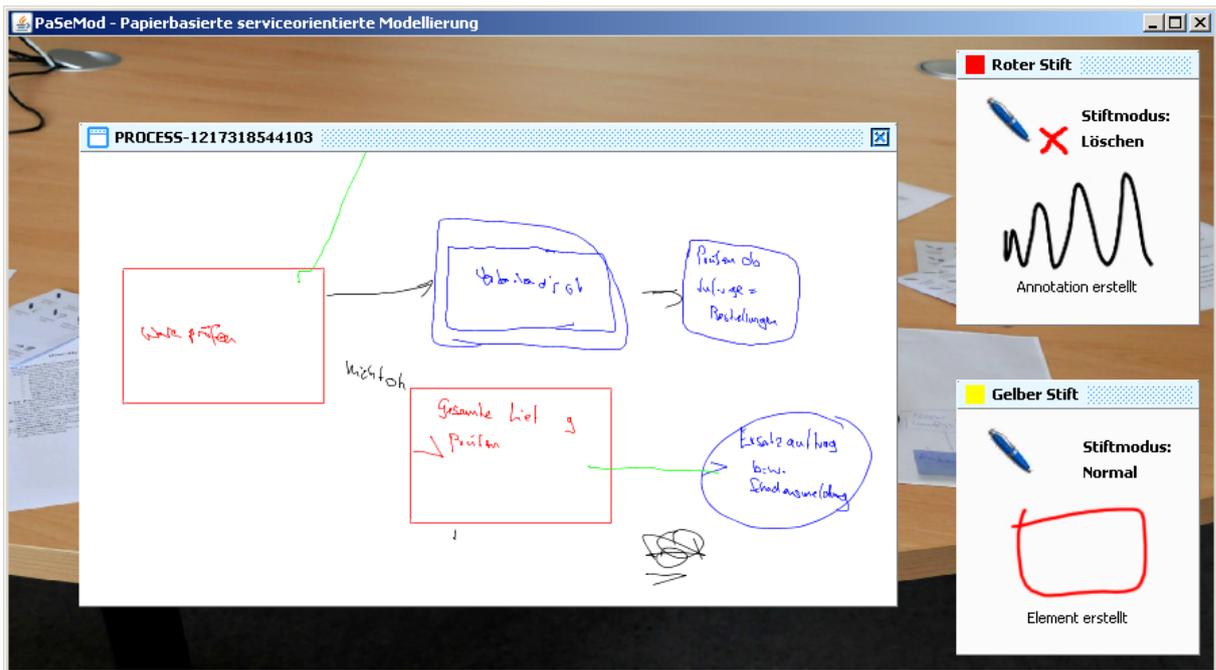


Abbildung 24: PaSeMod GUI mit Prozessvisualisierung (links) und stiftbasierten Feedbackboxen (rechts)

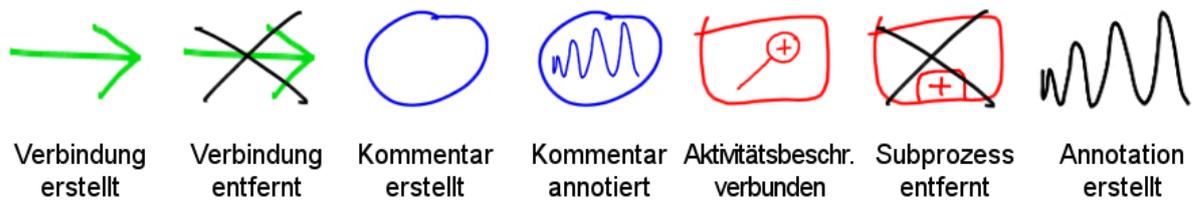


Abbildung 25: Auswahl an visuellen Rückmeldungen der letzten Modellierungsaktion

Die ganzheitliche Gestaltung des visuellen Feedbacks (Abbildung 24) orientiert sich an dem in Kapitel 4.5.4 geschilderten Konzept. Auf der linken Seite der GUI wird das digitale Prozessmodell visualisiert. Die Visualisierung ist möglichst nah am physischen Modell gehalten (low-fidelity Visualisierung), ergänzt dieses aber um eine farbliche Visualisierung der Syntax (entsprechend dem Farbcode der Notationselemente auf den Aktionskarten). Die farbliche Visualisierung ermöglicht es, formal erkannte Elemente eindeutig zu identifizieren. Eine Gegenüberstellung der physischen und der digitalen Prozessvisualisierungen zeigt Abbildung 26. Aktivitätskarten werden im Gegensatz zu dem Prozesspapier und zu den Post-Its in der GUI nicht dargestellt. Dies ist nicht nötig, da keine direkte Verarbeitung der beschrifteten Aktivitätsbeschreibungen erfolgt.

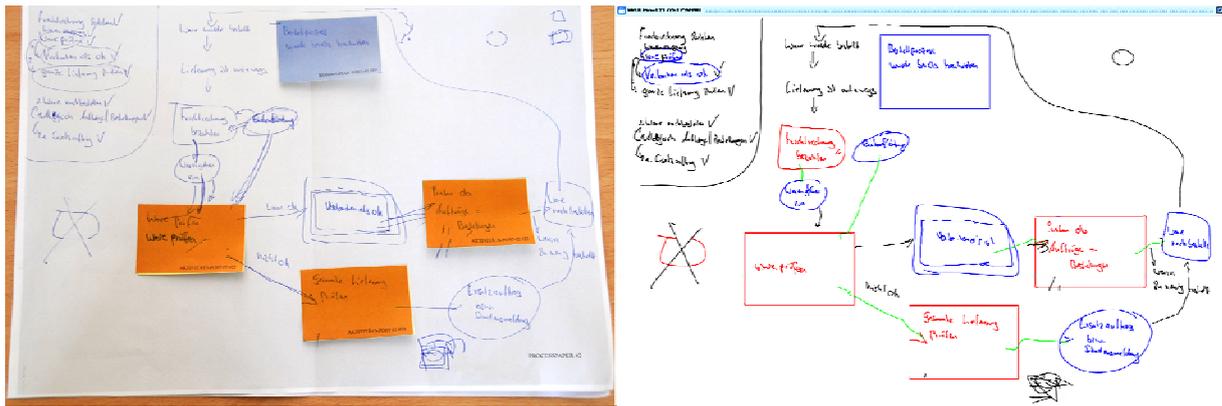
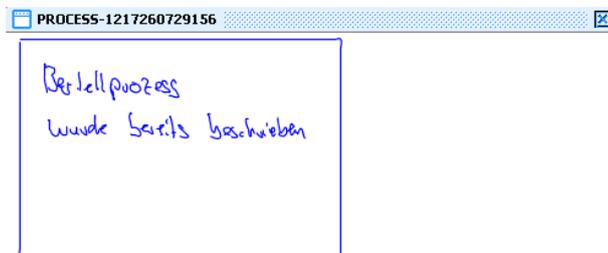


Abbildung 26: Vergleich papierbasiertes Modell (links) und computerbasierte Visualisierung (rechts)

Da die Verwendung der Post-Its von ihrer natürlichen Nutzungsweise leicht abweicht (Post-Its müssen zusätzlich durch einen Stich mit dem Prozesspapier verbunden werden), erhalten Nutzer bei der Verwendung von Post-Its eine zusätzliche Hilfestellung (Abbildung 27).



Bitte das Post-It durch einen kurzen Strich mit dem Prozesspapier verbinden!

Hinweis: Wenn mehrere unverbundene Post-Its existieren überlagern diese sich.

Abbildung 27: Hinweis beim Erstellen von Post-Its

Anbindung mehrerer Stifte

Ein Kernaspekt von PaSeMod ist die kollaborative Modellierung mit mehreren Stiften. Prinzipiell bieten sowohl die Bluetooth-Technologie als auch das Paper Toolkit die Möglichkeit mit mehreren Stiften parallel zu arbeiten. Aufgrund architektonischer Beschränkungen kann jedoch mit jeder Betriebssysteminstanz immer nur ein Digitalstift verbunden werden. Als Lösung wird für jeden Digitalstift eine eigene Instanz einer virtuellen Maschine (VM) auf Windowsbasis erzeugt, dieser jeweils ein Bluetooth-Empfänger zugeordnet und der jeweilige Stift mit ihr verbunden. Die einzelnen Stifte wurden zur besseren Differenzierbarkeit mit einer eindeutigen Farbkennzeichnung versehen und können so den Feedbackboxen der GUI direkt zugeordnet werden (siehe Abbildung 28).

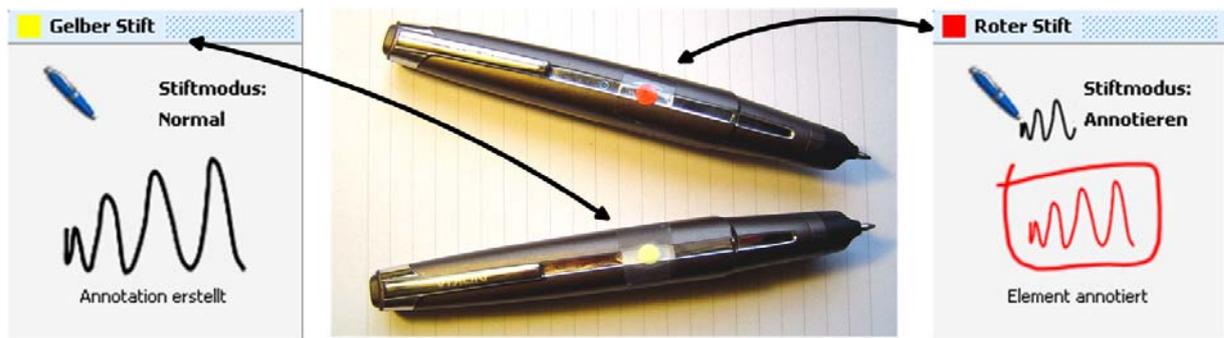


Abbildung 28: Digitalstifte mit eindeutiger Zuordnung zu den Feedbackboxen

Weiterverarbeitung

Für eine flexible Weiterverarbeitung der skizzierten Prozesse wurden drei verschiedene Exportmöglichkeiten entwickelt. Die rudimentärste Form stellt die Speicherung des digitalen Prozessmodells als Bilddatei (Abbildung 29) dar. Um beispielhaft zu zeigen, wie eine Weiterverarbeitung des Prozessmodells in einem BPM-Werkzeug aussehen kann, unterstützt der Prototyp zudem einen Export nach Marama ([57], siehe Seite 36). Wie Abbildung 30 zeigt, ermöglicht das erstellte Marama-Diagramm eine integrierte Darstellung formeller und informeller Inhalte. Mit dem Plug-In Marama Sketch (siehe Kapitel 5.2.2) können zudem Skizzen in formelle Diagrammelemente transformiert werden. Um eine allgemeine Weiterverwendung der skizzierten Prozesse zu ermöglichen, kann das Prozessmodell auch als XML-Datei exportiert werden. Die XML-Struktur wird durch eine entsprechende DTD (Document Type Definition) beschrieben (siehe Anhang C).

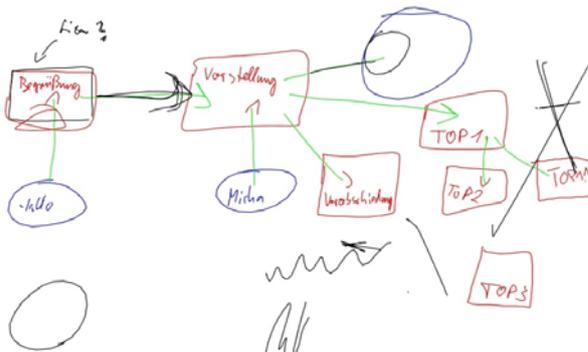


Abbildung 29: Prozessexport als Bild

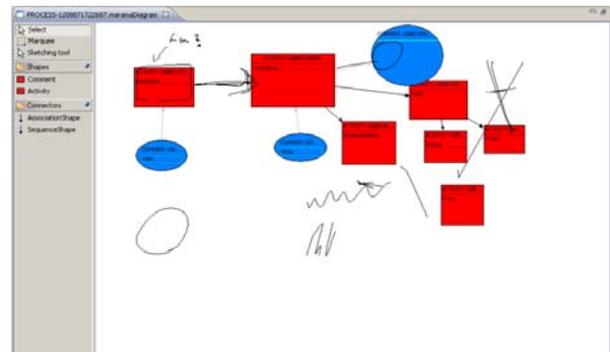


Abbildung 30: Prozessexport nach Marama

5.4 Zusammenfassung der PaSeMod-Realisierung

Als Grundlage für die Realisierung des entwickelten Konzeptes wurde zu Beginn des Kapitels ein aus den sieben Komponenten *Pen Server*, *Ink Manager*, *Recognition Engine*, *Feedback Manager*, *Process Manager*, *Paper Manager* und *Persistence Manager* bestehendes Architekturkonzept entwickelt. Diese klare Aufteilung in die einzelnen Komponenten schafft eine für Anpassungen und Erweiterungen offene Architektur. Darauf aufbauend wurden

wichtige technische Anforderungen an das System beschrieben, von denen insbesondere die Streaming-Unterstützung und die parallele Erfassung mehrerer Stifte genannt seien.

Anschließend wurden die für die Realisierung verwendeten Technologien vorgestellt. Die Grundlage bildet die interaktive Papiertechnologie von *Anoto* in Verbindung mit dem Streaming-fähigen *Digitalstift SU-1B* von Nokia. Als Entwicklungsframework wurde das *Paper Toolkit*, ein Open Source Framework mit der Unterstützung einer Streaming-Eingabe durch mehrere Stifte gleichzeitig, ausgewählt. Als weitere Komponenten wurden gewählt:

- *\$I Recognizer* (Gestenerkennung)
- *Tablet PC Recognizer Pack* (Handschrifterkennung)
- *Microsoft Speech API* (Sprachsynthese)

Im Abschluss wurde die konkrete prototypische Umsetzung von PaSeMod beschrieben. Dabei wurden zu Beginn notwendige Maßnahmen zur Komplexitätsreduktion der zu entwickelnden Systemaspekte beschrieben. Für eine Auflistung dieser sei auf die entsprechenden Ausführungen auf Seite 95 verwiesen. Bei der Beschreibung der konkreten Realisierung wurden insbesondere Abweichungen vom Konzept und Realisierungsprobleme hervorgehoben. So konnte aufgrund einer fehlenden Verfügbarkeit geeigneter Digitalstifte das konzipierte stiftbasierte Feedback nicht umgesetzt werden und musste primär durch eine GUI (und z.T. auditiv) ersetzt werden. Auch ließen sich aufgrund von Druckproblemen die Prozesspapiere und Post-Its nur eingeschränkt realisieren. Dies resultiert darin, dass maximal Prozesspapiere der Größe DIN A3 verfügbar sind. Als Vorbereitung für eine digitale Weiterverarbeitung der skizzierten Modelle bietet das System verschiedene Exportmöglichkeiten der erstellten Modelle.

Insgesamt konnte ein solides System entwickelt werden, welches alle Kernkonzepte des entwickelten PaSeMod-Konzeptes beinhaltet (mit Ausnahme des stiftbasierten Feedbacks) und so eine geeignete Basis für die Evaluation bereitstellt.

6 Evaluation von PaSeMod

Im vorangegangenen Kapitel wurde die auf der interaktiven Papiertechnologie von *Anoto* aufbauende, prototypische Realisierung von PaSeMod vorgestellt und beschrieben, wie sich diese von dem in Kapitel 4 vorgestellten Konzept unterscheidet. Um ein erstes Feedback zu der Umsetzung von PaSeMod und dadurch implizit zu der grundsätzlichen Idee und dem entwickelten Konzept (soweit im Prototyp realisiert) zu erhalten, wurde auf Basis des entwickelten Systems im Rahmen der Arbeit eine erste kleine Evaluation von PaSeMod durchgeführt. Die Evaluation erfolgte, in Anknüpfung an die Vorstudie, als Modellierungs-Workshop mit zwei Anwendern aus der Praxis.

6.1 Ziel

Im Fokus der Untersuchungen stand die Evaluation der *Benutzbarkeit* und der *Brauchbarkeit* des Systems. Hinsichtlich der *Benutzbarkeit* wurde insbesondere durch Beobachtungen analysiert, inwieweit die Benutzer das umgesetzte Modellierungs- und Interaktionskonzept verstehen und mit ihm kollaborative Modellierungsszenarien effektiv und effizient lösen können. Dabei wurden kritische Nutzungsprobleme, wie das grundsätzliche Verständnis der papierbasierten Modellierung, Unstimmigkeiten im Modellierungs- und Interaktionskonzept sowie fehlende Funktionen oder Rückmeldungen erfasst und analysiert. Die Analyse der *Brauchbarkeit* richtete sich vor allem auf die Bewertung der praktischen Einsetzbarkeit des PaSeMod-Konzeptes (sowohl des Gesamtkonzeptes als auch von Teilaspekten) allgemein und im Vergleich zur Modellierung am Rechner (z.B. mit Microsoft Visio) bzw. dem Vorzeichnen von Modellen auf Papier. Dazu wurde anhand der persönlichen Meinung der Teilnehmer untersucht, welche Bereitschaft existiert, ein solches neuartiges System zu verwenden, welcher persönliche Nutzen für den eigenen Arbeitskontext gesehen wird, wie die Einsetzbarkeit im praktischen Arbeitsalltag gesehen wird und ob die zu Grunde gelegten Anforderungen und die daraus entwickelten Konzepte korrekt sind.

6.2 Methodik

Der Untersuchung liegt eine formative und qualitative Evaluationsmethodik (siehe Myers und Avison [119]) zugrunde. Ein formatives Vorgehen beschreibt eine Evaluation, die auf Basis von Prototypen bereits während des Entwicklungsprozesses stattfindet. Bei PaSeMod kann mit diesem Vorgehen frühzeitig ein aussagekräftiges Feedback zu dem grundsätzlichen Verständnis und der Akzeptanz der neu entwickelten Konzepte erfragt werden.

Für die Evaluation wurde ein Usability-Test mit einem zweistufigen Vorgehen (vgl. Nielsen [124]), bestehend aus einer expertenzentrierten und einer nutzerzentrierten Phase, gewählt. Im ersten Schritt wurden in einem so genannten Experten Walkthrough szenariobasierte

Handlungsabläufe des Systems betrachtet und kritische Nutzungsprobleme („Critical Incidents“, CIs) des Prototyps identifiziert. Bei der Identifikation möglicher Probleme wurde im Sinne einer heuristischen Evaluation zudem ein zuvor ausgearbeiteter Kriterienkatalog hinzugezogen. Die CIs wurden anschließend hinsichtlich ihrer Relevanz bezüglich der nutzerbezogenen Untersuchung bewertet und kritische Probleme vor der Durchführung der Benutzungstests behoben. Die Liste der identifizierten CIs findet sich in Anhang D.

Die nutzerzentrierte Evaluationsphase setzte sich zusammen aus einer Nutzerbeobachtung (Benutzungstest) und nachfolgenden Interviews. Der Benutzungstest diente vor allem der Analyse der Nutzbarkeit durch die Identifikation konkreter Nutzungsprobleme. In der Art eines szenariobasierten Walkthroughs wurden die Teilnehmer mit einem Modellierungsszenario aus ihrem Arbeitskontext konfrontiert, in dessen Verlauf verschiedene Modellierungsaufgaben bewältigt werden mussten. Um die Informalität und Kreativität möglichst wenig einzuschränken und eine möglichst natürliche Nutzungssituation zu erzeugen wurde jedoch kein strikt sequentieller Ablauf, sondern nur ein grober, an den Funktionen des Systems orientierter Vorgehensleitfaden gewählt. Das konkrete Szenario war an einem Prozess aus der tatsächlichen Arbeitspraxis der Nutzer angelehnt. So wurden die Nutzer in eine für sie möglichst normale Nutzungssituation versetzt und konnten den Prozess (weitgehend) „aus dem Kopf“ modellieren. Um die persönlichen Gedankengänge der Probanden zu erfassen, wurden die Teilnehmer während des Benutzungstests animiert ihre Gedanken verbal auszudrücken („Think Aloud“-Methode). Direkt im Anschluss an den Benutzungstest fanden mit den Teilnehmern individuelle Gespräche statt. In diesen wurde die persönliche Meinung der Probanden, vor allem in Bezug auf die Nutzerzufriedenheit und Einschätzung der Brauchbarkeit, analysiert. Die Gespräche waren als semi-strukturierte Interviews mit wenigen, möglichst offenen, nicht-direktiven Fragen gestaltet. Der Interviewleitfaden (Anhang E) orientierte sich an den auf Seite 42 beschriebenen Gestaltungsvorschlägen von Merton und Kendall [113].

Für eine spätere Analyse wurden der Arbeitsplatz der Nutzer und die GUI von PaSeMod während des Benutzungstests per Video aufgezeichnet. Diese Dokumentation wurde ergänzt durch die Notizen von Beobachtern. Die Interviews wurden per Audio-Rekorder aufgezeichnet und für die Auswertung weitestgehend wörtlich transkribiert.

6.3 Konzeption des Workshops

Im Folgenden wird die Konzeption des Workshops anhand der Teilnehmer, des Untersuchungsrahmens und des Workshop-Ablaufes beschrieben.

6.3.1 Teilnehmer

In den Benutzungstest und die nachfolgenden Interviews waren zwei Probanden (i.F. Nutzer) und drei Forscher involviert. Beide Nutzer waren bereits am Workshop der Vorstudie (siehe Kapitel 3.2) beteiligt und konnten so an die dort gesammelten Erfahrungen anknüpfen. Auch ermöglichte dies gewissermaßen die Validierung der Umsetzung der aus dem Workshop generierten Anforderungen. Beide Probanden stammten aus einem mittelständischen Unternehmen (circa 150 Beschäftigten) der Produktionsbranche und hatten dort die formale Rolle des „Key Users“ für das SAP-System inne. Einer der Nutzer war der Leiter der IT-Abteilung und hatte bereits Erfahrungen mit der Modellierung von Prozessen in Microsoft Visio. Die andere, weibliche Nutzerin war Einkaufsmanagerin und besaß noch keine Modellierungserfahrung am Computer. Stattdessen zeichnete sie sich in der Praxis häufig Modelle mit Stift und Papier auf. Beide Probanden lassen sich nach Nardi ([122], siehe Seite 18) als Local Developer klassifizieren. Die Forscher setzten sich zusammen aus einem Moderator für den Benutzungstest sowie aus zwei Beobachtern. Einer der Beobachter war Experte im Bereich Usability-Studien und führte gemeinsam mit dem Moderator die Interviews durch. Der andere Beobachter war der Entwickler des Prototyps und stand während des Benutzungstestes bei technischen Fragen und Problemen zur Verfügung.

6.3.2 Untersuchungsrahmen

Der Benutzungstest fand im gleichen Besprechungsraum der Universität Siegen statt, wie bereits der Workshop der Vorstudie (siehe Seite 50). Für die Evaluation wurden die Aktionskarten von PaSeMod so angepasst, dass die nicht implementierte Funktionalität nicht abgebildet war und so die Nutzer nicht verwirrt wurden (siehe Abbildung 31). Die Nutzer saßen während des Testes gemeinsam mit dem Moderator nebeneinander an einem großen, runden Tisch und führten die Modellierung gemeinsam durch (siehe Abbildung 32). Das grafische Interface von PaSeMod wurde über einen Beamer an die ihnen gegenüberliegende Wand projiziert. Zwischen den Nutzern lagen auf dem Tisch mehrere Prozesspapiere der Größe DIN A3 und A4, sowie mehrere Aktivitätsbeschreibungen. Zudem hatte jeder Nutzer einen eigenen, farblich markierten Stift, eine Aktionskarte und mehrere Aktivitäts- und Kommentar-Post-Its vor sich liegen. Die Stifte waren mit einem in der Nähe stehenden Laptop verbunden, über den zugleich das Audio-Feedback abgespielt wurde.

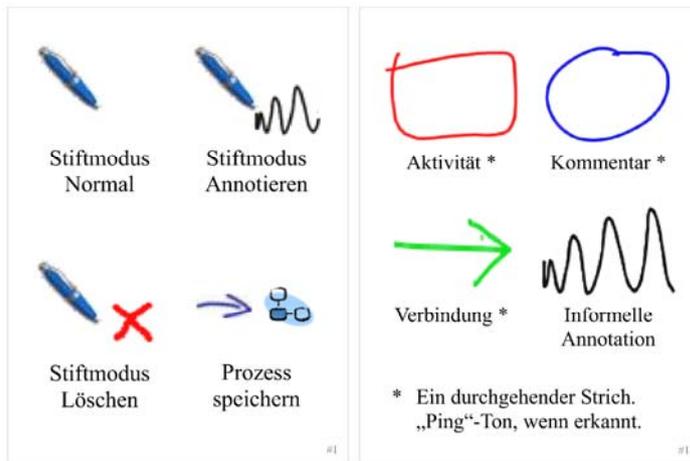


Abbildung 31: Angepasste Aktionskarten



Abbildung 32: Workshop-Setting

6.3.3 Ablauf des Workshops

Insgesamt lief die Evaluation über zwei Stunden und war unterteilt in eine kurze Einführung in das System, in die Durchführung des Benutzungstestes und in eine abschließende Interviewphase. Um auch die Unterstützung der Kollaborationsaspekte des Systems zu untersuchen, fand die Evaluation, bis auf die Interviews, mit beiden Nutzern gleichzeitig statt.

In der Einführungsphase wurde den Nutzer die interaktive Papiertechnologie und das Grundkonzept von PaSeMod vorgestellt. Während die Nutzer sich „spielerisch“ mit dem System vertraut machten, wurde die für die Erstellung rudimentärer Prozesse benötigte Grundfunktionalität (die Prozesselemente und die Stiftmodi) eingeführt. Zu Beginn des Benutzungstestes wurden die Nutzer in das Szenario eingeführt. Den Kern des Szenarios bildete die Modellierung des Prozesses „Aufnahme eines neuen Artikels in das Sortiment“. Dieser Prozess stammte direkt aus dem Arbeitsalltag der Nutzer, so dass sie sich schnell mit dem Szenario identifizieren konnten. Er umfasste die Analyse der Absatzmöglichkeiten für einen neuen Artikel, das Einstellen des neuen Artikels in das SAP-System, das Anlegen entsprechender Sichten im SAP-System, die Planung der Absatzmengen, das Anstoßen eines Bestellprozesses, die Bezahlung und Prüfung der Ware (bei erfolgter Lieferung), sowie, wenn die Ware nicht in Ordnung ist, eine Ersatzbeschaffung, ansonsten eine Prüfung der Auftragslage und gegebenenfalls das Anstoßen einer nachfolgenden Bestellung. Anhang F zeigt ein aus 25 Schritten bestehendes Modell des Prozesses, visualisiert als eEPK-Prozesskette. Dieses diente vor allem zur Orientierung für den Moderator. Im ersten Schritt wurde ein Teilausschnitt des Prozesses gewählt, den die Benutzer mit den ihnen erklärten Funktionen von PaSeMod aus dem Kopf heraus modellieren sollten. Während der Modellierung wurden ihnen schrittweise, sobald nötig, weitere Funktionalitäten (z.B. das Ändern der Bedeutung skizzierter Elemente über die Aktionskarten oder das Speichern des

Prozesses) erklärt. Nachdem die Nutzer einigermaßen sicher im Umgang mit dem System waren, wurde das Konzept der Post-Its eingeführt. Im zweiten Schritt wurde dann ein anderer Prozessausschnitt gewählt und dieser auf einem neuen Prozesspapier vorrangig mit den Post-Its modelliert. Währenddessen wurden die Nutzer gebeten, für ein paar Aktivitäten Aktivitätsbeschreibungen zu erstellen und diese mit dem Prozess zu verknüpfen. Nachdem die Nutzer sich intensiv mit dem System auseinandergesetzt und die ihnen gestellten Aufgaben bewältigt hatten, wurde sie in den Interviews hinsichtlich ihrer Beurteilung des Systems befragt.

6.4 Ergebnisse

Die Auswertung der Evaluation stützt sich auf die Videoaufzeichnungen, die Interviewtranskripte und die Notizen der Beobachter. Für die Analyse der Benutzbarkeit wurde auf Basis der Beobachtungen und einer Analyse der Videos eine Liste kritischer Nutzungsprobleme identifiziert. Die einzelnen Probleme wurden den jeweiligen Systemaspekten zugeordnet und mit Aussagen aus den Interviews ergänzt. Die Analyse der Brauchbarkeit basiert in erster Linie auf einer strukturierten Zusammenfassung der Kernaussagen beider Interviews zur Einschätzung der Brauchbarkeit.

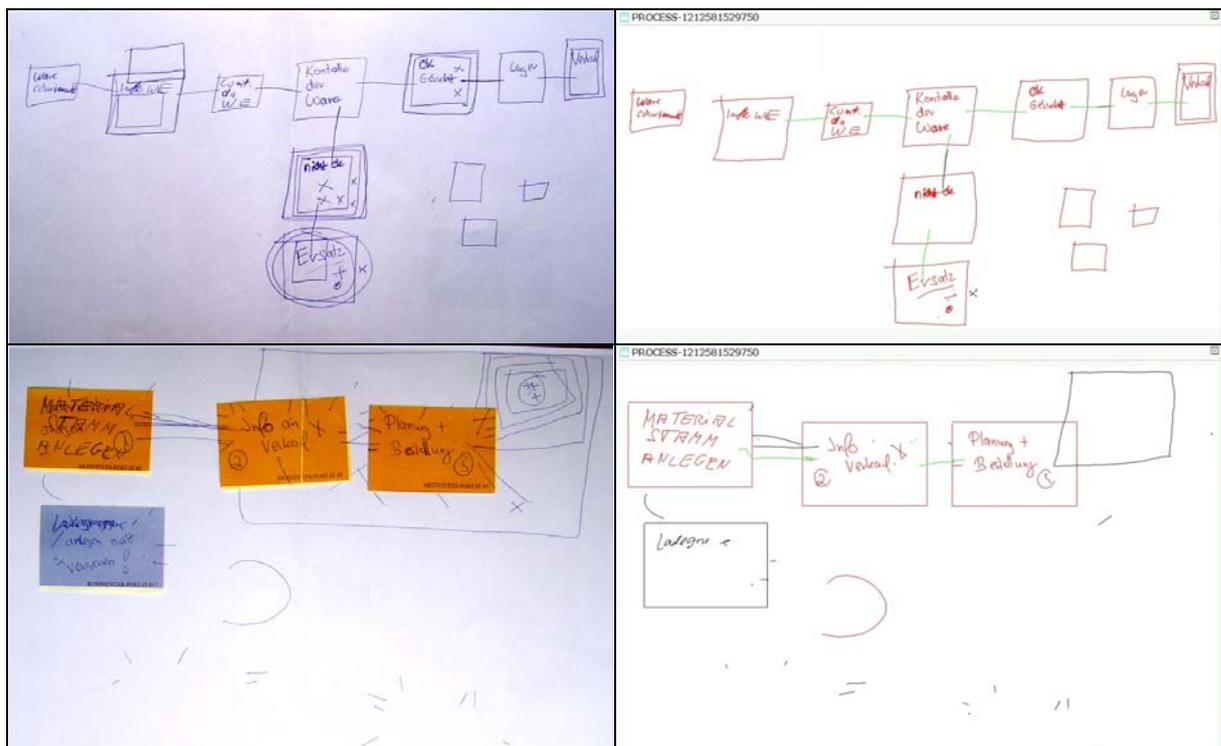


Abbildung 33: Übersicht der skizzierten Prozessmodelle (oben der Prozessteil ohne Post-Its, unten der mit Post-Its; links abgebildet ist jeweils das Papiermodell, rechts das entsprechende digitale Prozessmodell)

Eine grobe Übersicht der zwei während des Benutzungstestes erstellten Prozessmodelle ist als Gegenüberstellung von papierbasiertem und digitalem Modell in Abbildung 33 dargestellt. Im

Folgendes wird zuerst kurz auf technische Hindernisse, die während der Evaluation auftraten, eingegangen und im Anschluss werden die Ergebnisse zur Benutzbarkeit und zur Brauchbarkeit dargestellt.

6.4.1 Technische Hindernisse

Während des Benutzungstestes zeigten sich einige kleinere technische Hindernisse des Systems. Besonders problematisch war der Ausfall eines Digitalstiftes während der ersten Modellierungsphase. Da die Bluetooth-Verbindung des Stiftes abgebrochen war, wurden keine Daten mehr an das System gesendet. Zudem waren die Nutzer kurzzeitig irritierte, da sich der Verbindungsabbruch nur durch drei unauffällig blinkende LEDs am Stift bemerkbar machte. Die Ursache des Problems lag in einer instabilen Bluetooth-Verbindung, die sich aus der Kombination einer speziellen Firmware des Stiftes und der Verwendung einer VM für die Bluetooth-Verbindung ergab. Als Resultat musste die erste Modellierungsphase mit nur einem Stift fortgeführt werden und erst nach einem Neustart des Systems standen in der zweiten Phase wieder beide Stifte zur Verfügung.

Ein weiteres Problem war die schlechte Erfassung handschriftlicher Texte auf dem Prozesspapier. Dies führte dazu, dass die Nutzer Aktivitätsnamen zum Teil mit dem Stift nachfahren mussten. Dieses Problem ist insofern sehr kritisch, da, wie ein Benutzer folgerichtig feststellte, ohne eine Bildschirmvisualisierung die schlechte Erfassung nicht erkennbar gewesen wäre. Zurückzuführen ist das Problem auf eine schlechte Druckqualität bei der Erzeugung der DIN A3 Prozesspapiere oder auf ein zu schwaches Aufdrücken des Stiftes auf das Papier, so dass der Stift die Eingabe nicht korrekt erkennen konnte.

6.4.2 Benutzbarkeit

Für die Bewertung der Benutzbarkeit wurden kritische Nutzungsprobleme identifiziert und den verschiedenen Konzeptaspekten zugeordnet. In Anknüpfung an die in Kapitel 2.4 beschriebenen Forschungsaspekte erfolgt die Beschreibung der Ergebnisse differenziert nach Modellierungs- und Interaktionskonzept. Die komplette Liste der identifizierten CIs findet sich in Anhang G.

Allgemein war zu beobachten, dass die Nutzer die Prozessausschnitte weitgehend problemlos abbilden konnten. Dementsprechend war ihr Feedback zur Benutzbarkeit des Gesamtsystems positiv und es wurde lediglich ein gewisser Gewöhnungsaufwand der Handhabung angemerkt:

„[...] nur eine Handlings- bzw. Übungssache und wenn man dann weiß wie das geht, dann geht das in Fleisch und Blut über.“ (N1)³⁸

„Man muss sich an den Stift gewöhnen und braucht am Anfang ein bisschen, dann geht es aber ganz gut.“ (N2)

Modellierungskonzept

Das Modellierungskonzept umfasst insbesondere die Abstraktion in Prozess- und Detailebene, die Gestaltung der Prozessnotation und die Kombination informeller und formeller Aspekte. Kritischer Faktor für die Benutzbarkeit des Modellierungskonzeptes ist das Verständnis durch Endbenutzer. Da an der Evaluation lediglich Local Developer beteiligt waren, lassen sich zur Nutzbarkeit des Modellierungskonzeptes nur begrenzt Aussagen treffen.

Die Differenzierung zwischen Prozessablauf und Aktivitätsdetails war für die Nutzer klar verständlich, wurde aber durch das Szenario nicht weiter untersucht. Die Syntax und die Semantik der Prozessnotation wurden von den Nutzern schnell verstanden und die Elemente intuitiv verwendet. Lediglich für die Zuordnung der Farben zu den Elementen musste einer der Nutzer auf seiner Aktionskarte nachschauen. Bezüglich der Mächtigkeit der Notation war zu beobachten, dass anfangs einer der Nutzer ein zusätzliches formales Element („Zustand“), das den aktuellen Prozessstatus beschreiben sollte, vermisste (CI 1)³⁹. Der Nutzer äußerte sich daher dahingehend, dass es *„[...] auch noch ganz interessant [wäre], wenn man so was [ein eigenes Element] selbst entwerfen kann.“* Ein Zustands-Element existiert in PaSeMod nicht, da ein Zustand das Ergebnis einer Aktivität ist und daher direkt in der Aktivitätsbeschreibung oder als Kommentar zu modellieren wäre.

Zu der Integration informeller und formeller Prozessaspekte lassen sich kaum Aussagen treffen, da die Nutzer von dieser Möglichkeit kaum Gebrauch machten (siehe Abbildung 33). Vielmehr haben sie versucht das digitale Prozessmodell möglichst „sauber“ von ungewollten informellen Elementen zu halten (CI 2). Dabei produzierten sie auf dem Papier jedoch mehr Striche als nötig, was zu einer gewissen Unübersichtlichkeit und Divergenz zwischen Papiermodell und digitalem Modell führte (siehe Abbildung 33). Dies führte wiederum dazu, dass die Nutzer am Schluss schwerpunktmäßig mit der Beameransicht arbeiteten (CI 3). Gründe für die Vermeidung informeller Inhalte können in einem fehlenden Verständnis der Positionierung des Systems (mit einer späteren Weiterverarbeitung) oder in der Anknüpfung an das dem einen Nutzer aus Visio bekannte Modellierungsvorgehen liegen.

³⁸ Zur besseren Nachvollziehbarkeit sind die Aussagen teilweise den Nutzern zugeordnet: N1 = Nutzer 1, N2 = Nutzer 2.

³⁹ „CI x“ verweist auf das x. Critical Incident. Die vollständige Auflistung aller kritischen Nutzungsprobleme findet sich in Anhang G.

Interaktionskonzept

Konzept der Stiftmodi: Das Konzept der verschiedenen Stiftmodi wurde von den Benutzern akzeptiert und weitgehend verstanden. Neben einem Missverständnis der Bedeutung des Annotationsmodus (ein Nutzer wechselte zeitweise in den Annotationsmodus, wenn er ein formales Element „annotieren“ wollte, siehe CI 4) waren kleinere Schwierigkeiten bezüglich des Bewusstseins des aktiven Stiftmodus auszumachen. So zeigen zwei CIs (CI 5 und 6), dass einer der Nutzer sich eines falschen Stiftmodus nicht bewusst war. Folgende Nutzeraussage spiegelt das Problem wieder: *„Mir war es klar [dass es unterschiedliche Modi gibt]. Nur, dass ich es oben rechts im Fenster sehen kann, darauf habe ich anfangs nicht geachtet.“* bzw. *„Am Anfang habe ich relativ wenig geschaut in welchem Modus ich bin, daher hatte ich am Anfang auch einige Probleme.“* Der Nutzer wusste zwar theoretisch, dass er am Bildschirm den aktiven Modus sehen kann, hat ihn aber nicht explizit genug wahrgenommen. Dieses Problem ist zurückzuführen auf die Beschränkung des Prototyps, der lediglich allgemeine Feedbackmechanismen bietet. Konzeptionell wäre der aktive Stiftmodus direkt über eine Stift-LED identifizierbar (siehe Kapitel 4.5.3) und besäße so eine deutlich höhere Sichtbarkeit.

Konzept der Aktionskarten: Der Ansatz, Funktionen und Stiftmodi über die Aktionskarten zu wählen, war intuitiv und wurde mit *„[...] finde ich gut.“* und *„Gute Lösung.“* beschrieben. Die Möglichkeit, die Bedeutung von Elementen anhand der Aktivitätskarten zu ändern wurde von den Nutzern als sehr hilfreich empfunden (*„Ah, [...]. Das ist ja einfacher“*), vor allem da das System mehrfach eine Aktivität nicht erkannte oder fälschlicherweise als Kommentar erkannte⁴⁰.

Konzept der Post-Its: Die Grundidee der Post-Its und das Verbinden mit dem Prozesspapier wurde ebenfalls problemlos verstanden. Als besonders hilfreich wurde das Feedback für das Verbinden der Post-Its empfunden (*„Das hat er ja direkt als Hinweis, als eine Art ‚Online-Hilfe‘ gegeben.“*). Größere Probleme ergaben sich jedoch beim Umpositionieren (*„Verschieben“*) der Post-Its (siehe CI 7). Dass man für das Verschieben der Post-Its erst die Verbindung von Post-Its und Prozesspapier durch einen kurzen Strich im Löschmodus entfernen muss und dann im normalen Modus das Post-Its wieder verbinden kann, war für die Nutzer nicht intuitiv. Stattdessen hatten sie eine Art *„Verschiebemodus“* gesucht. Zusätzlich wurden die Nutzer durch das Feedback, das beim Löschen der Verbindung kam, irritiert: *„Wenn man eine Verbindung gelöscht hat, hatte man direkt wieder den Hinweis, um es zu verbinden. Das war am Anfang kurz verwirrend. [...] da hätte er einen spezifischen Hinweis*

⁴⁰ Im Benutzungstest wurden insgesamt 21 Aktivitäten skizziert, von denen zwei nicht formal und drei falsch (als Kommentare) erkannt wurden. Zudem wurden zwei Kommentare und zwölf Verbindungen skizziert, die alle korrekt erkannt wurden.

bringen können.“ Eine Unstimmigkeit fand sich in dem Systemverhalten, dass beim Löschen eines Post-Its auch dessen Inhalt verloren geht (CI 8). Zwar wurde dies von den Nutzern auch so erwartet, es führt jedoch zu Inkonsistenzen zwischen dem papier- und dem digitalen Modell, wenn ein gelöschtes Post-It wieder neu hinzugefügt wird.

Konzept der Aktivitätsbeschreibungen: Da die Aktivitätsbeschreibung nur kurz angeschnitten wurden, lässt sich keine Aussage über ihre Benutzbarkeit treffen. Jedoch verstanden die Nutzer intuitiv, dass sie die Aktivitätsbeschreibung mit der Aktivität über einen kurzen Strich verbinden mussten. Sie merkten aber an, dass nicht zu erkennen ist, ob mit einer Aktivität eine Aktivitätsbeschreibung verbunden ist (CI 16). Dies bedingte sich durch die fehlende Umsetzung der Notationsgesten und des Hilfemodus.

Gestaltung des Feedbacks: Die Aussagen zur Nutzbarkeit des Feedbacks variierten zwischen *„Ich fand’s gut.“* und *„Könnte vielleicht an ein paar Stellen noch besser sein.“*. Kritisch angemerkt wurde vor allem die weiter oben bereits beschriebene schlechte Sichtbarkeit des aktiven Stiftmodus. Ein Nutzer äußerte sich dahingehend, dass sich dieses Problem nach einer kurzen Einarbeitung vermutlich nicht mehr stellt: *„Sobald ich das Handling verinnerlicht habe und weiß, wie er reagiert, hat man das ‚Ratz Fatz‘ drinnen.“* Weitere, auf das Feedback zurückführbare CIs sind:

- Das System lieferte kein Feedback, warum ein Element nicht erkannt wurde (CI 9). Der Nutzer wusste daher nicht sofort, ob es sich um ein Erkennungsproblem handelt, er im falschen Stiftmodus ist oder das System nichts erkennt, weil sich Elemente überlappen (CI 10) oder ein Element um ein anderes herum gezeichnet wurde.
- Im Gegensatz zu dem Prozesspapier und zu den Post-Its werden die Aktivitätskarten nicht visualisiert. Dies ist auch nicht zwingen nötig, da keine direkte Verarbeitung der Eingabe erfolgt, verwirrte jedoch die Nutzer durch das vollständige Ausbleiben eines Feedbacks (CI 11).
- Die Reaktion, dass der Prozess gespeichert wurde, kam zum Teil stark verzögert zu dem Funktionsaufruf (CI 12). Der Benutzer dachte daher, dass das System seinen Befehl nicht erkannt hat.

Als zusätzliche Hilfe wurde das Feedback über die Sprachausgabe (z.B. beim Ändern des Stiftmodus) empfunden, da man nicht extra auf den Bildschirm schauen muss. Jedoch wurde angemerkt, dass es in größeren kollaborativen Sitzungen störend wirken würde: *„In der Sitzung wäre es also nicht so vorteilhaft, alleine hätte man den Blick hoch gespart.“* Der Vorschlag des Nutzers war daher die Sprachausgabe optional an- und abschalten zu können. *„So als ‚Vorinfo‘, dass man diskutieren kann, bevor man es verbindet“*, würde der Benutzer zudem erwarten (CI 13), dass man mit dem Stift und dem Papier bestimmte Elemente auf dem Bildschirm *„zum Beispiel über einen Knopf mit einem Laserpointer oder einem Mauszeiger“* hervorheben kann. Dies wäre natürlich nur in Sitzungen mit mehreren Nutzern relevant.

Weitere Beobachtungen: Die im Folgenden lose aufgezählten Beobachtungen sind großteils auf den prototypischen Charakter des Systems zurückzuführen.

- Die Benutzer erwarteten, dass sie auch informelle Elemente löschen können (CI 14).
- Die Benutzer erwarteten, dass sie auch gerichtete Verbindungen erstellen können (CI 15). Dies ging nicht, da der Prototyp zusammengesetzte Gesten nicht erkennt.
- Die Nutzer waren irritiert, dass das System den Prozess speichert, ohne dass ein Name für den Prozess vergeben wurde (CI 17).

6.4.3 Brauchbarkeit

Ziel von PaSeMod ist es, die teilformelle und kollaborative Erfassung erster Prozessskizzen und Prozessanforderungen zu ermöglichen, sowie Endbenutzer aktiv in diese Modellierung einzubeziehen. Die Brauchbarkeit zeigt somit allgemein die Eignung für die Abbildung der Prozesse, sowie speziell den Nutzen von PaSeMod für die Benutzer in ihrem Arbeitskontext.

Die ersten Äußerungen beider Nutzer zeigen eine positive Einstellung gegenüber einem papierbasierten Modellierungssystem:

„Finde es toll [das System].“ „Es erspart einem die Arbeit.“ (N1)

„Ganz lustig.“ „Es hat den Spieltrieb geweckt.“ (N2)

Besonders die Äußerung zu dem Spieltrieb ist positiv hervorzuheben, da dieser Aspekt Endbenutzer, die sich aufgrund ihrer fehlenden Erfahrung zurückhalten würden, motivieren kann, sich aktiv an der Modellierung zu beteiligen.

Bei der Analyse des persönlichen Nutzens, den die Teilnehmer für sich und ihren Arbeitskontext sehen, ergibt sich jedoch ein divergentes Ergebnis.

„Es erspart einem die Arbeit.“ (N1) und

„Könnte ich mir sehr gut vorstellen [den praktischen Einsatz]“ (N1) gegenüber

„Würde [...] auf jeden Fall an einen Rechner gehen und dort [...] entwickeln.“ (N2)

Nutzer 1 ist *„[...] es von jetzt gewohnt, die Dinge auf normalem Papier aufzuzeichnen und muss sie dann noch mal [in den Computer] umsetzen.“* Dementsprechend passt sich PaSeMod direkt in seine bestehende Arbeitspraxis ein und erspart ihm Arbeit, indem es die stattfindende Papiermodellierung um interaktive Funktionen anreichert und das Modell automatisch in den Computer überträgt. Nutzer 2 sieht dagegen keinen grundsätzlichen Bedarf für ein solches System und würde direkt an einem Computer modellieren.

Die divergente persönliche Einschätzung erklärt sich zum Teil durch ein unterschiedliches Verständnis des PaSeMod-Systems und eine unterschiedliche Herangehensweise an die Prozesserstellung. Nutzer 1 versteht für sich das im Konzept implizierte zweistufige

Modellierungsvorgehen aus papierbasierter und nachfolgender GUI-basierter Modellierung bzw. Orchestrierung. Er geht folgerichtig „[...] davon aus, dass hier [bei PaSeMod] die Umsetzung größtenteils schon im PC erfasst ist und nur noch [...] nachbearbeitet werden muss.“ Nutzer 2 versteht das System dagegen „als ob [...] [er] auf ein normales Stück Papier geschrieben hätte und es danach einscane.“ Für ihn „[...] war halt keine Funktion dahinter.“ Dies ist auch darauf zurückzuführen, dass in der Evaluation nicht explizit darauf eingegangen wurde, wie die Weiterverarbeitung stattfindet. Den Nutzern wurde lediglich erklärt, dass das Modell zum Beispiel als Bild exportierbar ist. Auch versuchte Nutzer 2 während des Benutzungstestes ein möglichst konkretes und formales Prozessmodell zu erstellen, was jedoch kein Ziel des PaSeMod-Konzeptes ist.

Ebenfalls sehr unterschiedlich bewerteten die Nutzer die Brauchbarkeit von PaSeMod im Bezug zu anderen Modellierungssystemen. Nutzer 1 würde situationsabhängig unterscheiden: „Wenn ich schon einen festen Prozess im Kopf hätte, würde ich mich eher direkt an den PC setzen und würde es da machen. [...] Hier das [PaSeMod] sehe ich eher für die kreative Phase [...] [und] für das Brainstorming.“ Diese Einordnung entspricht exakt der in Kapitel 4.1.1 beschriebenen Positionierung des Systems. Nutzer 2 dagegen „[...] würde es lieber elektronisch abbilden, da es in keinem Umfeld benutzt werden würde, in dem man keinen Zugriff auf den Rechner hat.“ Für sich sieht er „[...] keine Vorteile, dass es auf Papier ist, außer in Umgebungen, in denen der Rechner nicht verfügbar ist.“ In den Vorteilen, die Papier gegenüber GUI-basierten Systemen besitzt (z.B. seine Flexibilität, Natürlichkeit und die Unterstützung kreativer Aspekte), sieht er für sich keinen persönlichen Nutzen.

Ein Kernaspekt von PaSeMod ist die Unterstützung einer kollaborativen Modellierung, die auch beide Nutzer in dem System sehen. So beschreibt Nutzer 1 Gruppensitzungen, in denen Skizzen auf Flipcharts erstellt und diese dann als PDF abfotografiert an das Sitzungsprotokoll angehängt werden. Unter anderem für ein solches Szenario kann er sich „[...] sehr gut vorstellen, wenn man das dann so macht [mit PaSeMod] und es dann gleich im PC hat.“ Auch für gemeinsame Diskussionen und eine Optimierung bestehender Prozesse sieht er Vorteile, dies mit PaSeMod zu erledigen. Nutzer 2 sieht den Vorteil eher in einer möglichen Zeitersparnis („[...] man [kann] sich dadurch natürlich Zeit sparen.“). Für die Unterstützung der Diskussion von Prozessmodellen befürchtet er, dass die Papiermodelle nicht flexibel genug angepasst werden können: „Wenn man gemeinsam entwickelt muss man auch die Möglichkeit haben Sachen zu verschieben.“

Zusammenfassend lässt sich bezüglich der Brauchbarkeit festhalten, dass Nutzer 1 sich den Einsatz sowohl für sich selbst als auch in kollaborativen Szenarien sehr gut vorstellen kann und sich davon eine Arbeitsersparnis verspricht. Bis auf die Bearbeitung bereits bestehender ausgedruckter Prozesse würde er, soweit nach dem kurzen Benutzungstest feststellbar, auch keine grundlegenden Aspekte vermissen: „Müsste man mal eine Woche testen, dass man mal

mit echten Problemen damit arbeitet. So zum Anfang ist da alles dabei.“ Nutzer 2 dagegen sieht keine Vorteile für seine Arbeitspraxis und begründet dies vor allem mit der fehlenden Änderbarkeit und der Endlichkeit von Papier (im Benutzungstest konnte nur DIN A3 Prozesspapier verwendet werden), sowie mit fehlenden Weiterverarbeitungsmöglichkeiten. Zudem besaß Nutzer 2 schon ein gewisses Prozessverständnis und versuchte den Prozess bereits auf einer sehr konkreten Ebene zu gestalten, was mit PaSeMod nur begrenzt möglich ist und auch nicht das Ziel des Konzeptes ist. Dazu passt auch die im Benutzungstest gemachte Beobachtung, dass Nutzer 2 versuchte, das digitale Modell möglichst frei von „unnötigen“ informellen Elementen zu halten, was jedoch auf Kosten der Übersichtlichkeit des Papiermodells ging (siehe auch Abbildung 33). Es entstand der Eindruck, dass Nutzer 2 das System als Ersatz für eine computerbasierte Modellierung sah.

6.5 Zusammenfassung und Bewertung

Auch wenn die Evaluation aufgrund ihrer Konzeption nur als erster Anhaltspunkt dienen kann, lieferte sie wertvolle Ergebnisse für das PaSeMod-Grundkonzept und für den entwickelten Prototyp.

In den Einsatzszenarien einer konzeptionellen und kollaborativen Modellierung durch Endbenutzer (für die PaSeMod konzipiert wurde) wurde das System als gut geeignet („brauchbar“) für die Arbeitspraxis eingeschätzt und so die grundsätzliche Motivation für ein solches System bestätigt. Für die formale Modellierung durch einzelne oder wenige bereits erfahrene Benutzer ist PaSeMod dagegen nicht geeignet. Hier kommen endbenutzergerechte Modellierungssysteme zum tragen. Dies entspricht der in Kapitel 4.1.1 geschilderten konzeptionellen Positionierung des Systems.

Weiter bestätigte die Evaluation der Nutzbarkeit die grundsätzliche Eignung sowohl des Modellierungs-, als auch des Interaktionskonzeptes für eine endbenutzergerechte, serviceorientierte Prozessmodellierung. Nutzungsprobleme sind insbesondere auf prototypische Einschränkungen und auf die Nichtumsetzbarkeit des konzipierten stiftbasierten Feedbacks zurückzuführen. Neben diesen beiden Punkten findet sich Optimierungspotential vor allem bei den Feinheiten der konkreten Feedbackgestaltung (z.T. noch direkteres und reicheres Feedback nötig) und bei einigen Interaktionsabläufen (z.B. dem Verschieben von Post-Its). Im Anhang G findet sich eine Bewertung der Nutzungsprobleme zusammen mit potentiellen Handlungsmöglichkeiten.

Die Ergebnisse motivieren somit eine weitergehende, umfangreiche Evaluation des PaSeMod-Konzeptes unter Berücksichtigung der identifizierten kritischen Nutzungsprobleme und unter Einbeziehung sowohl von Local Developern als auch von End-Usern. Dabei sollte auch die Integration des Systems in die Gesamtprozesserstellung untersucht werden.

7 Zusammenfassung und Ausblick

Die vorliegende Arbeit befasst sich mit der endbenutzergerechten Erstellung von Prozessen, die durch den Einsatz papierbasierter Interfaces realisiert werden soll. Die Möglichkeit, die eigenen Prozesse und Systeme flexibel an dynamische Entwicklungen und sich verändernde Umweltbedingungen anpassen zu können, ist ein wichtiges Ziel für Unternehmen (vgl. [172]) und wird häufig über eine Verbindung der Konzepte BPM und SOA realisiert. Um eine flexible Prozesserstellung zu ermöglichen, wurden im Kontext von BPM und SOA verschiedene professionelle Modellierungs- und Orchestrierungssysteme entwickelt. Diese Systeme sind für professionelle Entwickler und Modellierer konzipiert und daher für Endbenutzer zu komplex (vgl. [63, 104, 161]). Gerade zu Beginn der aus mehreren Phasen bestehenden Prozesserstellung besteht jedoch die Notwendigkeit zur Einbeziehung von Endbenutzern, um zum Beispiel Kommunikationsprobleme zu vermeiden (vgl. [23]). Speziell für Endbenutzer konzipierte Orchestrierungswerkzeuge (vgl. [64, 104, 130, 135, 160]) sind in einer solchen frühen Phase nicht geeignet. Sie unterstützen z.B. keine Darstellung informeller Prozessaspekte und ermöglichen nur stark eingeschränkt eine kollaborative Modellierung. Diese Einschränkungen bedingen sich zum Teil durch die Verwendung von GUI-Interfaces und der WIMP-Interaktionsmetapher. Daher erscheint die Verwendung des alternativen Interaktionskonzeptes einer skizzenbasierten Modellierung auf Papier vielversprechend. Papier und Skizzen besitzen gemeinsam eine hohe Natürlichkeit, eine gute kollaborative Nutzbarkeit, sowie die Möglichkeit informelle Aspekte zu skizzieren.

Motiviert durch diese Vorteile von Papier und von Skizzen war die Zielsetzung der Arbeit die Entwicklung, Umsetzung und Evaluation eines Konzeptes, das durch die Verwendung papierbasierter Interfaces Endbenutzer bei einer kollaborativen, serviceorientierten Prozessmodellierung unterstützt. Der Schwerpunkt der Arbeit lag auf der Gestaltung eines geeigneten Interaktionssystems sowie der zugehörigen Gestaltung eines geeigneten Modellierungskonzeptes.

In einer anfänglichen Literaturanalyse zeigte sich, dass bisher kaum Untersuchungen zum Ablauf einer (kollaborativen) serviceorientierten Modellierung in der Praxis und zur Einbeziehung von Endbenutzern in diese, existieren. Als Ausgangsbasis für die Konzepterstellung wurde daher in einer Fallstudie die serviceorientierte Prozesserstellung in einem großen Unternehmen analysiert. Zudem wurden in einem Prozessdesign-Workshop Gestaltungsaspekte einer endbenutzerorientierten, papierbasierten Modellierung untersucht. Auch wenn die Vorstudie aufgrund des geringen Umfangs (ein bzw. drei Teilnehmer) nur eine eingeschränkte Aussagekraft besitzt, so konnten doch einzelne Erkenntnisse aus der Literatur bestätigt (insb. [21, 36]) und wertvolle Gestaltungsaspekte für das Konzept gewonnen werden. In der Fallstudie zeigte sich vor allem, dass die Prozesserstellung ein

komplexer, mehrstufiger Prozess ist, in dem Endbenutzer in den ersten Schritten mittels interaktiven Papiers unterstützt werden könnten, ihre Prozessideen semi-formal einzubringen.

Aufgrund der unterschiedlichen Ausrichtung der verschiedenen Modellierungsschritte kann ein System nicht alle Schritte geeignet unterstützen. Daher wurde ein zweistufiges Konzept entwickelt, das für die einzelnen Schritte jeweils geeignete Systeme anbietet. Dies setzt sich zusammen aus 1.) einer *vorangehenden, papierbasierten Modellierung* teilformeller Aspekte und 2.) einer *nachfolgenden, rechnergestützten Konkretisierung* der Modelle. In der Arbeit wurde nur der erste Schritt betrachtet. Da in der Literatur vergleichbare Konzepte bisher fehlten, war ein wichtiger Aspekt der Arbeit die Identifikation von Anforderungen an ein papierbasiertes Modellierungssystem für Endbenutzer. So konnten auf Basis der Literatur und der Vorstudie 16 konkrete Anforderungen an das Konzept identifiziert werden.

Unter Berücksichtigung der identifizierten Anforderungen wurde ein integriertes Konzept einer *papierbasierten serviceorientierten Modellierung* für Endbenutzer (genannt *PaSeMod*) entwickelt. Im Gegensatz zu den bisherigen Modellierungsansätzen unterstützt das PaSeMod-Konzept eine kollaborative Modellierung (durch die Verwendung großformatiger Papiere und mehrerer Stifte), sowie eine Integration formeller und informeller Prozessaspekte. Dadurch können Endbenutzer aktiv in eine frühe Modellierungsphase einbezogen werden. Gemäß den formulierten Forschungszielen wurde der Schwerpunkt auf die Gestaltung des Interaktionskonzeptes (unterteilt in das papierbasierte Interaktionskonzept und das Feedbackkonzept) und des (implizit behandelten) Modellierungskonzeptes gelegt.

Durch eine Unterscheidung und separate Erfassung der Modellierungsebenen „Prozessablauf“ und „Aktivitätsdetails“, begegnet das *Modellierungskonzept* der Komplexität einer serviceorientierten Prozesserstellung und macht beide Ebenen für Endbenutzer explizit sichtbar und differenzierbar. Über eine fließende Integration formeller und informeller Skizzen wird zudem der Forderung nach einem „unbeschränkten Modellierungssystem“ (vgl. [78]) entsprochen. Eine geringe Mächtigkeit der Prozessnotation und die Möglichkeit Aktivitätsdetails natürlichsprachig auf vorstrukturierten Formularen zu beschreiben, ermöglichen es auch Endbenutzern ohne Modellierungskennntnisse die Notation schnell zu verstehen, sich aktiv zu beteiligen und Prozesse teilstrukturiert zu beschreiben.

Ein Grundproblem von Papierinterfaces ist das fehlende dynamische Feedback. Diesem Grundproblem begegnet das *Interaktionskonzept* durch eine Integration der papierbasierten Interaktion mit dynamischen Feedbackmechanismen. Durch eine Kombination von Feedback über die Digitalstifte (z.B. über LEDs) und von Feedback über einen verbundenen Computer (z.B. externe GUI), ermöglicht das System eine feine Differenzierung des Feedbacks nach der Relevanz des auslösenden Ereignisses und der nötigen Sichtbarkeit für die Nutzer. Eine kollaborative Modellierung und eine aktive Einbeziehung von Endbenutzern werden

unterstützt, indem jeder Nutzer einen eigenen Stift für die Modellierung erhält. Das Konzept der Aktivitätskarten unterstützt Endbenutzer bei der Aneignung der Notation. Durch die Verwendung von Post-Its, die als Alternative zum Skizzieren von Aktivitäten und Kommentaren verwendet werden können, bietet das Konzept eine Lösung für die fehlende Veränderbarkeit einmal erstellter Skizzen und erhöht somit die Veränderbarkeit der Modelle.

Nur kurz angeschnitten wurde die rechnergestützte Weiterverarbeitung der papierbasiert erstellten Prozessmodelle. Auch konnten einige, im Rahmen der Arbeit aufgekommene, konzeptionelle Ideen nicht eingebunden werden. Sie erscheinen jedoch für eine weitere Betrachtung interessant. Eine Idee ist die Gestaltung eigener (z.B. domänenspezifischer) Visualisierungen der Notation (z.B. durch „Blanko-Aktionskarten“). Weitere Ideen betreffen eine Erweiterung der Feedbackfunktionalität. So könnten zum Beispiel eine Validierung gezeichneter Prozessmodelle und eine erweiterte Modellierungshilfe Endbenutzern beim Erlernen der Modellierung helfen.

Für eine Evaluation wurde das entwickelte PaSeMod-Konzept auf Basis der Anoto-Technologie [6] prototypisch realisiert. Aufgrund einer fehlenden Verfügbarkeit geeigneter Digitalstifte konnte das stiftbasierte Feedback jedoch nicht umgesetzt werden und musste primär durch eine GUI ersetzt werden. Weitere Einschränkungen betreffen die verfügbare Modellierungsmächtigkeit und die Größe der Prozesspapiere. Insgesamt wurde jedoch ein solides System entwickelt, das alle Kernkonzepte für die Evaluation bereitstellte.

In dieser kurzen Evaluation zeigte sich, dass PaSeMod eine konzeptionelle und kollaborative Modellierung durch Endbenutzer gut unterstützt und dass das entwickelte Interaktions- und Modellierungskonzept für Endbenutzer geeignet ist. Somit kann PaSeMod die formulierten Forschungsziele grundlegend umsetzen. Die „Handhabungsschwierigkeiten“, die sich dennoch zeigten, sind größtenteils auf den prototypischen Charakter und auf nötige Feinabstimmungen des Systems zurückzuführen. Es zeigte sich aber auch, dass die Benutzbarkeit durch das Fehlen eines stiftbasierten Feedbacks und durch technische Probleme mit dem Stift beeinträchtigt wurde. Die Ergebnisse sind jedoch als ein erster Anhaltspunkt zu sehen, da aufgrund des Zeitrahmens der Arbeit nur eine kurze Untersuchung durchgeführt werden konnte.

Auf Basis der Evaluation lassen sich für eine Weiterentwicklung von PaSeMod folgende zentrale Problemstellungen identifizieren: Zum einen sollte, um eine vollständige Evaluation zu ermöglichen, die prototypische Umsetzung unter Berücksichtigung der Evaluation vervollständigt werden. Dabei sollte auch eine Realisierung des stiftbasierten Feedbacks angestrebt werden. Des Weiteren sollte die Robustheit und Verlässlichkeit des Systems, zum Beispiel durch eine Integration bestehender Erkennungsframeworks, verbessert werden. Auch sollte eine Anpassung bestehender Prozesse und die Verbindung zu einem

Weiterverarbeitungssystem erforscht werden. Schließlich sollte eine weitergehende Evaluation stattfinden, die vor allem die kollaborative Modellierung durch verschiedene Endbenutzertypen und die Integration des Systems in die Gesamtprozesserstellung untersucht.

Die vorgestellte Arbeit leistet einen wichtigen Beitrag zur Schaffung eines Gesamtkonzeptes, das Endbenutzern auf Basis einer serviceorientierten Prozesserstellung eine flexible Anpassung ihrer Systeme ermöglicht. Sie liefert wertvolle Erkenntnis für das Verständnis einer serviceorientierten Prozesserstellung und für die Gestaltung von Systemen, die Endbenutzer bei einer solchen unterstützen können. Mit PaSeMod nimmt sich die Arbeit der Forschungslücke eines fehlenden Systems für eine endbenutzergerechte Prozessmodellierung an, indem sie ein Konzept aufzeigt, das es Endbenutzern durch eine semi-formale Modellierung auf einem Papierinterface ermöglicht, sich aktiv in die erste Phase der Prozesserstellung einzubringen. Der Beitrag der Arbeit liegt insbesondere in der Integration endbenutzerorientierter, papierbasierter und serviceorientierter Gestaltungsaspekte in einem geeigneten Interaktions- und Modellierungskonzept. Zudem liefert die Arbeit wichtige Hinweise auf das Potential des natürlichen Interaktionsmediums Papier für den Bereich des End-User Developments und zeigt Chancen und Herausforderungen eines neuen Nutzungsfeld (der Modellierung) für papierbasierte Anwendungen auf.

Das entwickelte Konzept ist allerdings nicht isoliert zu sehen, sondern bettet sich in die Vision, Endbenutzer durch die Komposition von Services zu einer flexiblen Anpassung ihrer Systeme zu befähigen, ein und liefert einen wichtigen Lösungsbaustein. Die Vision ist mit PaSeMod jedoch noch nicht vollständig umgesetzt. So besteht weiterer Forschungsbedarf, um das Konzept in eine ganzheitliche Prozesserstellung einzubinden (v.a. Aspekte der Weiterverarbeitung und der Anpassung bestehender Prozesse) und die organisatorischen (v.a. Bereitschaft Endbenutzer einzubeziehen) und technischen (v.a. Erweiterung des SOA-Konzeptes um EUD-Aspekte) Strukturen für einen praktischen Einsatz zu schaffen. Im Bereich papierbasierter Anwendungen sollte zudem eine Realisierung stiftbasierter Feedbackmechanismen angestrebt werden.

Danksagung

Ganz besonders herzlich bedanke mich bei Dipl.-Wirt.-Inform. Christian Dörner für die vielfältigen Ideen und Anregungen, sowie die vorbildhafte Betreuung. Darüber hinaus bedanke ich mich besonders bei Prof. Dr. Volkmar Pipek, bei Prof. Dr. Volker Wulf und beim Team des Lehrstuhls Wirtschaftsinformatik und Neue Medien, die diese Arbeit ermöglicht und durch wertvolle Diskussionen unterstützt haben.

Weiterer Dank gebührt den an der Vorstudie und der Evaluation beteiligten Nutzern und Unternehmen, die einen wesentlichen Beitrag zur Schaffung einer fundierten Gestaltungsbasis und zur Validierung von PaSeMod lieferten. Namentlich waren dies Herr Jensen von der Danske Bank, sowie Herr Beres, Frau Friedly und Frau Zimbrich von der Firma Sternjakob. Auch danke ich Torben Wiedenhöfer und wieder Christian Dörner für die Unterstützung bei der Durchführung der Untersuchungen.

Zudem möchte ich mich bei Guy Küstermann und bei der Firma Anoto für die Unterstützung bei der Realisierung des papierbasierten Systems bedanken.

Mein besonderer Dank geht an meine Freundin, an meine Geschwister und an meine Eltern. Ohne sie wäre ich nie soweit gekommen. Sie standen jederzeit hinter mir und haben mich immer wieder motiviert, unterstützt und gefördert.

Zuletzt danke ich Steffen Rumpf für seine wertvollen Korrekturen sowie Benedikt Ley und Markus Hoffman für die erholsamen Ablenkungen.

Literaturverzeichnis

1. Active Endpoints, Adobe, BEA, IBM, Oracle and SAP. WS-BPEL Extension for People, 2008.
2. Agrawal, A., Amend, M., Das, M., Ford, M., Keller, C., Kloppmann, M., König, D., Leymann, F., Müller, R., Pfau, G., Plösser, K., Rangaswamy, R., Rickayzen, A., Rowley, M., Schmidt, P., Trickovic, I., Yiu, A. and Zeller, M. Web Services Human Task (WS-HumanTask), Version 1.0, 2007.
3. Agrawal, A., Amend, M., Das, M., Ford, M., Keller, C., Kloppmann, M., König, D., Leymann, F., Müller, R., Pfau, G., Plösser, K., Rangaswamy, R., Rickayzen, A., Rowley, M., Schmidt, P., Trickovic, I., Yiu, A. and Zeller, M. WS-BPEL Extension for People (BPEL4People), Version 1.0, 2007.
4. Alda, S., Kuck, J. and Cremers, A.B., Tailorability of personalized BPEL-based Workflow Compositions. in *Services, 2007 IEEE Congress on*, (2007), 245-252.
5. Amoroso, A. and Quirini, L., A paper-digital agenda [hand-written to digital agenda synchronization system]. in *Consumer Communications and Networking Conference, 2004. CCNC 2004. First IEEE*, (2004), 697-699.
6. Anoto Group. Anoto Technology, <http://www.anoto.com>, 2007.
7. Anoto Group. Digital Printing Requirements *Anoto Reference documentation*, 2007.
8. Anoto Group. Laser Printing Guidelines *Anoto Reference documentation*, 2007.
9. Anoto Group. Offset Printing Requirements *Anoto Reference documentation*, 2007.
10. Anoto Group AB. Anoto Technology, 2007. <http://www.anoto.com>.
11. Arkin, A., Askary, S., Bloch, B., Curbera, F., Golland, Y., Kartha, N., Liu, C.K., Thatte, S., Yendluri, P. and Yiu, A. Web Services Business Process Execution Language Version 2.0. OASIS Standard. WS-BPEL TC OASIS, -. ed. <http://docs.oasis-open.org/wsbpel/2.0/OS/wsbpel-v2.0-OS.pdf>, 2007.
12. Baroth, E. and Hartsough, C. Visual programming in the real world. in Burnett, M.M., Goldberg, A. and Lewis, T.G. eds. *Visual object-oriented programming: concepts and environments*, Manning Publications Co., Prentice Hall, 1995, 21 - 42.
13. Beiß, W. Pilotstudie zum Digitalen Wahlstift, Freie und Hansestadt Hamburg, Landeswahlleiter, Hamburg, 2005.
14. Benyon, D., Turner, P. and Turner, S. *Designing Interactive Systems: People, Activities, Contexts, Technologies*. Addison-Wesley, 2005.
15. Bernstein, M., Robinson-Mosher, A., Yeh, R.B. and Klemmer, S.R., Diamond's Edge: From Notebook to Table and Back Again. in *Conference supplement to Ubicomp 2006: Posters*, (Irvine, CA, 2006).

16. Black, A. Visible planning on paper and on screen: The impact of working medium on decision-making by novice graphic designers. *Behaviour and information technology*, 9 (4). 283 - 296.
17. Blackwell, A.F. Ten years of cognitive dimensions in visual languages and computing. *Journal of Visual Languages and Computing & Computing*, 17. 285 - 287.
18. Bødker, K., Kensing, F. and Simonson, J. *Participatory IT Design: Designing for Business and Workplace Realities*. MIT Press, Cambridge, Massachusetts; London, England, 2004.
19. Borggräfe, B. Abbildung von BPMN-Modellen auf BPEL *Unveröffentlicht*, Universität Siegen, Siegen, 2007.
20. Borggräfe, B., Dörner, C., Heß, J. and Pipek, V. Drawing services: towards a paper-based interface for end-user service orchestration *Proceedings of the 4th international workshop on End-user software engineering*, ACM, Leipzig, Germany, 2008.
21. Brahe, S. BPM on Top of SOA: Experiences from the Financial Industry. in *Business Process Management*, 2007, 96-111.
22. Brahe, S. Early Experiences on Adopting BPM and SOA - An Empirical Study, IT University of Copenhagen, Copenhagen, 2007.
23. Brahe, S. and Schmidt, K. The story of a working workflow management system *Proceedings of the 2007 international ACM conference on Conference on supporting group work*, ACM, Sanibel Island, Florida, USA, 2007.
24. Brancheau, J.C. and Brown, C.V. The management of end-user computing: status and directions. *ACM Comput. Surv.*, 25 (4). 437-482.
25. Brooks, F.P. No Silver Bullet - Essence and Accident in Software Engineering. *IEEE Computer*, 20 (4). 10-19.
26. Brücher, H. and Endl, R. Erweiterung von UML zur geschäftsregelorientierten Prozessmodellierung. in Becker, J. and Knackstedt, R. eds. *Wissensmanagement mit Referenzmodellen. Konzepte für die Anwendungssystem- und Organisationsgestaltung*, Springer Verlag, Berlin, 2002, 145 - 162.
27. Burkhardt, F. Deutsche Sprachsynthese, syntheticspeech.de, 2008.
28. Burnett, M., Cook, C. and Rothermel, G. End-user software engineering. *Communications of the ACM*, 47 (9). 53-58.
29. Burnett, M., Wiedenbeck, S., Grigoreanu, V., Subrahmaniyan, N., Beckwith, L. and Kissinger, C. Gender in end-user software engineering *Proceedings of the 4th international workshop on End-user software engineering*, ACM, Leipzig, Germany, 2008.
30. Chappell, D.A. *Enterprise Service Bus. Theory in Practice*. O'Reilly Media, Beijing, 2004.

31. Chen, Q., Grundy, J. and Hosking, J., An e-whiteboard application to support early design-stage sketching of UML diagrams. in *Human Centric Computing Languages and Environments, 2003. Proceedings. 2003 IEEE Symposium on*, (2003), 219-226.
32. Conway, J. and Watts, S. *Software Engineering with LabVIEW*. Pearson Education, 2003.
33. Costabile, M.F., Fogli, D., Pero, M. and Piccinno, A. End-User Development: The Software Shaping Workshop Approach. in Lieberman, H., Paternò, F. and Wulf, V. eds. *End-User Development*, Springer, Dordrecht, 2006, 183 - 206.
34. Cypher, A. *Watch What I Do: Programming by Demonstration*. The MIT Press, Cambridge, Massachusetts & London, England, 1993.
35. Dai, G., Pen-based user interface. in *Computer Supported Cooperative Work in Design, 2004. Proceedings. The 8th International Conference on*, (2004), 32-36.
36. Damm, C.H., Hansen, K.M. and Thomsen, M. Tool support for cooperative object-oriented design: gesture based modelling on an electronic whiteboard *Proceedings of the SIGCHI conference on Human factors in computing systems*, ACM, The Hague, The Netherlands, 2000.
37. Dijkstra, E.W. On the Cruelty of Really Teaching Computing Science. *Communications of the ACM*, 32 (12). 1398 - 1404.
38. Dittrich, Y., Lindeberg, O. and Lundberg, L. End-User Development as Adaptive Maintenance. in Lieberman, H., Paternò, F. and Wulf, V. eds. *End-User Development*, Springer, Dordrecht, 2006, 295 - 314.
39. Donaldson, A. and Williamson, A., Pen-based Input of UML Activity Diagrams for Business Process Modelling. in *Proc. HCI 2005 Workshop on Improving and Assessing Pen-based Input Techniques*, (Edinburgh, 2005).
40. Dörner, C., Heß, J. and Pipek, V. Improving Information Systems by End User Development: A Case Study *15th Europ. Conference on Information Systems (ECIS 2007)*, University of St. Gallen, St. Gallen, Switzerland, 2007.
41. Dörner, C., Pipek, V., Weber, M. and Wulf, V. End-user development: new challenges for service oriented architectures *Proceedings of the 4th international workshop on End-user software engineering*, ACM, Leipzig, Germany, 2008.
42. Elliott, A. and Hearst, M.A. How large should a digital desk be?: qualitative results of a comparative study *CHI '00 extended abstracts on Human factors in computing systems*, ACM, The Hague, The Netherlands, 2000.
43. Fischer, G. Domain-oriented design environments *Proceedings of the 18th international conference on Software engineering*, IEEE Computer Society, Berlin, Germany, 1996.

44. Fischer, G. and Giaccardi, E. Meta-design: A Framework for the Future of End-User Development. in Lieberman, H., Paternò, F. and Wulf, V. eds. *End-User Development*, Springer, Dordrecht, 2006, 427 - 458.
45. Fischer, G., Giaccardi, E., Ye, Y., Sutcliffe, A.G. and Mehandjiev, N. Meta-design: a manifesto for end-user development, ACM, 2004, 33-37.
46. Fischer, G. and Scharff, E. Meta-design: design for designers *Proceedings of the 3rd conference on Designing interactive systems: processes, practices, methods, and techniques*, ACM, New York City, New York, United States, 2000.
47. Frankish, C., Morgan, P. and Noyes, J., Pen computing: some human factors issues. in *IEE Colloquium on Handwriting and Pen-Based Input*, (1994), 5/1-5/3.
48. Gadatsch, A. *Grundkurs Geschäftsprozess-Management*. Vieweg & Sohn Verlag, Wiesbaden, 2005.
49. Gaitanides, M., Scholz, R. and Vrohling, A. Prozeßmanagement - Grundlagen und Zielsetzungen. in Gaitanides, M., Scholz, R., Vrohling, A. and Raster, M. eds. *Prozessmanagement. Konzepte, Umsetzungen und Erfahrungen des Reengineering*, München, 1994, 1 - 19.
50. Gavran, I., Milanovic, A. and Srbljic, S. End-user programming language for service-oriented integration. *Proceedings of WDAS*, CA, USA, 2006.
51. Goranson, H.T. *The agile virtual enterprise: cases, metrics, tools*. Quorum Books, Westport, 1999.
52. Grasso, A., Karsenty, A. and Susani, M. Augmenting paper to enhance community information sharing *Proceedings of DARE 2000 on Designing augmented reality environments*, ACM, Elsinore, Denmark, 2000.
53. Green, T. and Blackwell, A. Cognitive Dimensions of Information Artefacts: a tutorial, 1998.
54. Green, T.R.G. and Petre, M. Usability Analysis of Visual Programming Environments: A 'Cognitive Dimensions' Framework. *J. Vis. Lang. Comput.*, 7 (2). 131-174.
55. Grille, M. Die nächsten drei Buchstaben - BPM, BPM-Netzwerk, 2006.
56. Grundy, J. and Hosking, J. Supporting Generic Sketching-Based Input of Diagrams in a Domain-Specific Visual Language Meta-Tool *Proceedings of the 29th International Conference on Software Engineering*, IEEE Computer Society, 2007.
57. Grundy, J., Hosking, J., Li, L. and Liu, N., Performance Engineering of Service Compositions. in *Proc. International Workshop of Service-oriented Software Engineering*, (2006), 26 - 32.
58. Guimbretiere, F., Paper Augmented Digital Documents. in *Proceedings of UIST'03*, (2003), pp. 51 - 60.

59. Gwizdka, J., Louie, J. and Fox, M.S. EEN: a pen-based electronic notebook for unintrusive acquisition of engineering design knowledge *Proceedings of the 5th International Workshops on Enabling Technologies: Infrastructure for Collaborative Enterprises (WET ICE'96)*, IEEE Computer Society, 1996.
60. Hale, P. History of End User Programming., EzineArticles.com, 2006.
61. Haller, M., Brandl, P., Leithinger, D., Leitner, J., Seifried, T. and Billinghamurst, M. Shared Design Space: Sketching Ideas Using Digital Pens and a Large Augmented Tabletop Setup. in *Advances in Artificial Reality and Tele-Existence*, Springer Berlin / Heidelberg, 2006, 185-196.
62. Hammond, T. and Davis, R., Tahuti: A Geometrical Sketch Recognition System for UML Class Diagrams. in *AAAI Spring Symposium on Sketch Understanding*, (2002).
63. Han, J., Han, Y., Jin, Y., Wang, J. and Yu, J. Personalized Active Service Spaces for End-User Service Composition *Proceedings of the IEEE International Conference on Services Computing*, IEEE Computer Society, 2006.
64. Han, Y., Geng, H., Li, H., Xiong, J., Li, G., Holtkamp, B., Gartmann, R., Wagner, R. and Weissenberg, N. VINCA – A Visual and Personalized Business-Level Composition Language for Chaining Web-Based Services. in *Service-Oriented Computing - ICSOC 2003*, Springer Berlin / Heidelberg, 2003, 165-177.
65. Harada, S., Saponas, T.S. and Landay, J.A. Voicepen: augmenting pen input with simultaneous non-linguistic vocalization *Proceedings of the 9th international conference on Multimodal interfaces*, ACM, Nagoya, Aichi, Japan, 2007.
66. Henderson, A. and Kyng, M. There's no place like home: continuing design in use. in *Design at work: cooperative design of computer systems*, Lawrence Erlbaum Associates, Inc., 1992, 219-240.
67. Heß, J., Küstermann, G. and Pipek, V. pRemote - a user customizable remote control *ACM Conference on Human Factors in Computing (CHI'08)*, ACM Press, Florence, Italy, 2008, 3279-3284.
68. Hilliger von Thile, A., Melzer, I. and Steiert, H.-P. Managers Don't Code: Making Web Services Middleware Applicable for End-Users. in *Web Services*, Springer, Berlin / Heidelberg, 2004, 139-151.
69. Hofmann, M. Softwareanpassungen mittels der Box-and-Wire-Metapher. Eine Fallstudie der Gestaltung eines Dispositionssystems eines Logistikbetreibers *Unveröffentlicht*, Universität Siegen, Siegen, 2008.
70. Hofmann, M., Ley, B. and Dörner, C. Endbenutzergerechte Anpassung von serviceorientierten Softwaresystemen *Multikonferenz Wirtschaftsinformatik*, München, 2008.

71. Holman, D., Vertegaal, R., Altosaar, M., Troje, N. and Johns, D. Paper windows: interaction techniques for digital paper *Proceedings of the SIGCHI conference on Human factors in computing systems*, ACM, Portland, Oregon, USA, 2005.
72. Hong, J.I. and Landay, J.A. SATIN: a toolkit for informal ink-based applications *Proceedings of the 13th annual ACM symposium on User interface software and technology*, ACM, San Diego, California, United States, 2000.
73. Hu, C., Liu, Q., Liu, X., Liao, C. and McEvoy, P., POEMS: A Paper Based Meeting Service Management Tool. in *Multimedia and Expo, 2007 IEEE International Conference on*, (2007), 1167-1170.
74. Hundhausen, C.D. Using end-user visualization environments to mediate conversations: a 'Communicative Dimensions' framework. *Journal of Visual Languages & Computing*, 16 (3). 153-185.
75. Hundhausen, C.D. and Douglas, S.A. Low-Fidelity Algorithm Visualization. *Journal of Visual Languages & Computing*, 13 (5). 449-470.
76. Iacucci, G., Pipek, V., Jacucci, G. and Kuutti, K. Continuing Design in Use of Tangible Computing Environments. Haddon, L., Mante-Meijer, E., Sapio, B., Kommenon, K.-H., Fortunati, L. and Kant, A. eds. *The Good, the Bad and the Irrelevant - The User and the Future of Information and Communication Technology (COST 269 conference)*, University of Art and Design, Helsinki, Finland, 2003, 209-216.
77. Ideogramic. Pervasive UML <http://www.ideogramic.com/>, 2008.
78. Jarzabek, S. and Huang, R. The case for user-centered CASE tools. *Communications of the ACM*, 41 (8). 93-99.
79. Jiang, H., Yeh, R.B., Winograd, T. and Shi, Y., DigiPost: Writing on Post-its with Digital Pen to Support Collaborative Editing Tasks on Tabletop Displays. in *Adjunct Proceedings of the 20th annual ACM Symposium on User Interface Software and Technology (UIST '07)*, (Newport, RI, USA, 2007).
80. Kaindl, H. and Carroll, J.M. Symbolic Modeling in Practice. *Communications of the ACM*, Vol. 42 (No.1). 28 ff.
81. Kant, M.v.d., Wilson, S., Bekker, M., Johnson, H. and Johnson, P. PatchWork: A software tool for early design *Human Factors in Computing Systems: CHI 98 Summary*, ACM Press, New York, 1998, pp. 221 - 222.
82. Klemmer, S.R., Newman, M.W., Farrell, R., Bilezikjian, M. and Landay, J.A. The designers' outpost: a tangible interface for collaborative web site *Proceedings of the 14th annual ACM symposium on User interface software and technology*, ACM, Orlando, Florida, 2001.

83. Kloppmann, M., König, D., Pfau, G. and Scheible, M. Von singulären Web Services zu integrierten SOA-Plattformen: Die Evolution serviceorientierter Architekturen und Anwendungen. in *IT*, Springer Berlin, Heidelberg, 2007, 7-23.
84. Kloppmann, M.K., D.; Leymann, F.; Pfau, G.; Rickayzen, A.; Von Riegen, C.; Schmidt, P.; Trickovic, I. WS-BPEL Extension for People – BPEL4People, 2005.
85. Koehler, J. and Vanhatalo, J. Process anti-patterns: How to avoid the common traps of business process modeling - Modeling control flow. *IBM WebSphere Developer Technical Journal*.
86. Kolberg, M. and Magill, E.H., Programming a PVR with pen and paper. in *Consumer Communications and Networking Conference, 2006. CCNC 2006. 3rd IEEE*, (2006), 1332-1333.
87. Kolberg, M., Magill, E.H., Wilson, M., Birstwistle, P. and Ohlstenius, O., Controlling appliances with pen and paper. in *Consumer Communications and Networking Conference, 2005. CCNC. 2005 Second IEEE*, (2005), 156-160.
88. Krafzig, D., Banke, K. and Slama, D. *Enterprise SOA: Service-Oriented Architecture Best Practices (The Coad Series)*. Prentice Hall PTR, 2004.
89. Kunau, G., Loser, K.-U. and Herrmann, T. Im Spannungsfeld zwischen formalen und informellen Aspekten: Modellierung von Dienstleistungsprozessen mit SeeMe. in Herrmann, T., Kleinbeck, U. and Kremer, H. eds. *Konzepte für das Service Engineering - Modularisierung, Prozessgestaltung und Produktivitätsmanagement*, Physica-Verlag HD, Heidelberg, 2005, 149 - 166.
90. Küstermann, G. pRemote – ein stift- und papierbasiertes Fernbedienungskonzept, Universität Siegen, Siegen, 2008.
91. Kvale, S. *Interviews: An introduction to qualitative research interviewing*. Sage Publications, Thousand Oaks, 1996.
92. Landay, J.A. and Myers, B.A. Interactive sketching for the early stages of user interface design *Proceedings of the SIGCHI conference on Human factors in computing systems*, ACM Press/Addison-Wesley Publishing Co., Denver, Colorado, United States, 1995.
93. Landay, J.A. and Myers, B.A. Sketching Interfaces: Toward More Human Interface Design. *Computer*, 34 (3). 56-64.
94. Le, K., Mingxu, W. and Zhongcheng, W., An overview of pen computing. in *Information Acquisition, 2005 IEEE International Conference on*, (2005), 8 pp.
95. LeapFrog. Fly Pen, <http://www.flypentop.com>, 2007.
96. Lehmann, F. *Integrierte Prozessmodellierung mit ARIS*. dpunkt.verlag, Heidelberg, 2008.
97. Li, Y., Hinckley, K., Guan, Z. and Landay, J.A. Experimental analysis of mode switching techniques in pen-based user interfaces *Proceedings of the SIGCHI*

- conference on Human factors in computing systems*, ACM, Portland, Oregon, USA, 2005.
98. Liao, C., Guimbretière, F. and Hinckley, K. PapierCraft: a command system for interactive paper *Proceedings of the 18th annual ACM symposium on User interface software and technology*, ACM, Seattle, WA, USA, 2005.
 99. Liao, C., Guimbretière, F. and Loeckenhoff, C., E. Pen-top feedback for paper-based interfaces *Proceedings of the 19th annual ACM symposium on User interface software and technology*, ACM, Montreux, Switzerland, 2006.
 100. Lieberman, H. *Your Wish is My Command: Programming by Example*. Morgan Kaufmann, San Francisco, CA, USA, 2001.
 101. Lieberman, H. and Liu, H. Feasibility Studies for Programming in Natural Language. in Lieberman, H., Paternò, F. and Wulf, V. eds. *End-User Development*, Springer, Dordrecht, 2006, 459 - 474.
 102. Lieberman, H., Paternò, F. and Wulf, V. End-User Development: An Emerging Paradigm. in Lieberman, H., Paternò, F. and Wulf, V. eds. *End-User Development*, Springer, Dordrecht, 2006, 1 - 8.
 103. Lieberman, H., Paternò, F. and Wulf, V. *End User Development*. Springer, Dordrecht, 2006.
 104. Liu, X., Huang, G. and Mei, H. Towards End User Service Composition *Proceedings of the 31st Annual International Computer Software and Applications Conference - Vol. 1- (COMPSAC 2007) - Volume 01*, IEEE Computer Society, 2007.
 105. Livescribe. Pulse Smartpen, <http://www.livescribe.com/>, 2007.
 106. Loser, K.-U. Unterstützung der Adoption kommerzieller Standardsoftware durch Diagramme, Universität Dortmund, Dortmund, 2005.
 107. Mackay, W.E. Patterns of sharing customizable software *Proceedings of the 1990 ACM conference on Computer-supported cooperative work*, ACM, Los Angeles, California, United States, 1990.
 108. MacKenzie, I.S. and Zhang, S.X. The immediate usability of graffiti *Proceedings of the conference on Graphics interface '97*, Canadian Information Processing Society, Kelowna, British Columbia, Canada, 1997.
 109. MacLean, A., Carter, K., Lennart Lövstrand and Moran, T. User-tailorable systems: pressing the issues with buttons *Proceedings of the SIGCHI conference on Human factors in computing systems: Empowering people*, ACM, Seattle, Washington, United States, 1990.
 110. McGee, D.R., Cohen, P.R. and Wu, L. Something from nothing: augmenting a paper-based work practice via multimodal interaction *Proceedings of DARE 2000 on Designing augmented reality environments*, ACM, Elsinore, Denmark, 2000.

111. Mehandjiev, N. and Bottaci, L. User-Enhanceability for Organizational Information Systems through Visual Programming *Proceedings of the 8th International Conference on Advances Information System Engineering*, Springer-Verlag, 1996.
112. Melzer, I. *Service-orientierte Architekturen mit Web Services: Konzepte - Standards - Praxis*. Spektrum Akademischer Verlag, München, 2007.
113. Merton, R.K. and Kendall, P.L. The focused interview. *American Journal of Sociology*, 51. 541 - 557.
114. Mørch, A.I. and Mehandjiev, N.D. Tailoring as Collaboration: The Mediating Role of Multiple Representations and Application Units. *Computer Supported Cooperative Work (CSCW)*, 9 (1). 75-100.
115. Mørch, A.I., Stevens, G., Won, M., Klann, M., Dittrich, Y. and Wulf, V. Component-based technologies for end-user development. 47 (9). 59-62.
116. Muller, M.J. Participatory design: the third space in HCI. in *The human-computer interaction handbook: fundamentals, evolving technologies and emerging applications*, Lawrence Erlbaum Associates, Inc., 2003, 1051-1068.
117. Muller, M.J. and Kuhn, S. Participatory design. *Commun. ACM*, 36 (6). 24-28.
118. Myers, B.A., Pane, J.F. and Ko, A. Natural programming languages and environments. *Commun. ACM*, 47 (9). 47-52.
119. Myers, M.D. and Avison, D.E. *Qualitative Research in Information Systems: A Reader*. Sage Publications, London, 2002.
120. Nakagawa, M., Kato, N., Machii, K. and Souya, T., Principles of pen interface design for creative work. in *Document Analysis and Recognition, 1993., Proceedings of the Second International Conference on*, (1993), 718-721.
121. Nardi, B.A. *A Small Matter of Programming: Perspectives on End User Computing*. MIT Press, 1993.
122. Nardi, B.A. and Miller, J.R. Twinkling lights and nested loops: distributed problem solving and spreadsheet development. *Int. J. Man-Mach. Stud.*, 34 (2). 161-184.
123. Newman, M.W., Lin, J., Hong, J.I. and Landay, J.A. DENIM: An Informal Web Site Design Tool Inspired by Observations of Practice. *Human-Computer Interaction*, 18 (3). 259 - 324.
124. Nielsen, J. *Usability Engineering*. Morgan Kaufmann Publishers Inc., 1993.
125. Norrie, M. and Signer, B., Overlaying Paper Maps with Digital Information Services for Tourists. in *Proceedings of the 12th International Conference on Information Technology and Travel and Tourism (ENTER 2005)*, (Innsbruck, Austria, 2005).
126. Norrie, M.C., Signer, B. and Weibel, N. General Framework for the Rapid Development of Interactive Paper Applications *CoPADD 2006, Workshop on Collaborating over Paper and Digital Documents*, Banff, Canada, 2006.

127. Norrie, M.C., Signer, B. and Weibel, N. Print-n-link: weaving the paper web *Proceedings of the 2006 ACM symposium on Document engineering*, ACM, Amsterdam, The Netherlands, 2006.
128. Objekt Management Group (OMG). Business Process Modeling Notation (BPMN) Version 1.0. OMG Final Adopted Specification, OMG, 2006.
129. Objekt Management Group (OMG). Unified Modeling Language (UML) Version 2.1.2, OMG, 2007.
130. Oinn, T., Addis, M., Ferris, J., Marvin, D., Greenwood, M., Goble, C., Wipat, A., Li, P. and Carver, T. Delivering web service coordination capability to users *Proceedings of the 13th international World Wide Web conference on Alternate track papers & posters*, ACM, New York, NY, USA, 2004.
131. Ouyang, C., Dumas, M., Hofstede, A.H.M.t. and Aalst, W.M.P.v.d. From BPMN Process Models to BPEL Web Services *Proceedings of the IEEE International Conference on Web Services*, IEEE Computer Society, 2006.
132. Ouyang, C., Dumas, M., van der Aalst, W.M.P. and ter Hofstede, A.H.M. From Business Process Models to Process-oriented Software Systems: The BPMN to BPEL Way. *BPMcenter.org, BPM-06-27*.
133. Oviatt, S. Human-centered design meets cognitive load theory: designing interfaces that help people think *Proceedings of the 14th annual ACM international conference on Multimedia*, ACM, Santa Barbara, CA, USA, 2006.
134. Oviatt, S., Arthur, A. and Cohen, J. Quiet interfaces that help students think *Proceedings of the 19th annual ACM symposium on User interface software and technology*, ACM, Montreux, Switzerland, 2006.
135. Paczynski, F. Simple Service ORchestration *Unveröffentlicht*, Universität Siegen, Siegen, 2008.
136. Pane, J.F. and Myers, B.A. More Natural Programming Languages and Environments. in Lieberman, H., Paternò, F. and Wulf, V. eds. *End-User Development*, Springer, Dordrecht, 2006, 31 - 50.
137. Patel, R., Plimmer, B., Grundy, J. and Ihaka, R. Ink Features for Diagram Recognition *Sketch Based Interfaces and Modeling IEEE*, 2007.
138. Pautasso, C. and Alonso, G. The JOpera visual composition language. *Journal of Visual Languages and Computing & Computing, Volume 16* (Issue 1-2). 119–152.
139. Pipek, V. and Kahler, H. Supporting Collaborative Tailoring. in Lieberman, H., Paternò, F. and Wulf, V. eds. *End-User Development*, Springer, Dordrecht, 2006, 315 - 346.
140. Pires, J.N., Godinho, T., Nilsson, K., Haage, M. and Meyer, C. Programming industrial robots using advanced input-output devices: test-case example using a CAD

- package and a digital pen based on the Anoto technology. *International Journal of Online Engineering (iJOE)*, Vol 3 (No 3).
141. Plimmer, B. and Freeman, I., A Toolkit Approach to Sketched Diagram Recognition. in *Proceedings of the HCI07 Conference on People and Computers XXI 2007*, (2007), p. 21.
 142. Plimmer, B.E. and Apperley, M. Freeform: A Tool for Sketching Form Designs *BHCI*, Bath, 2003.
 143. Reichert, M. and Stoll, D. Komposition, Choreographie und Orchestrierung von Web Services – Ein Überblick. *EMISA Forum, Band 24* (Heft 2). 21 - 32.
 144. Repenning, A. and Ioannidou, A. What Makes End-User Development Tick? 13 Design Guidelines. in Lieberman, H., Paternò, F. and Wulf, V. eds. *End-User Development*, Springer, Dordrecht, 2006, 51 - 86.
 145. Repenning, A., Ioannidou, A. and Zola, J. AgentSheets: End-User Programmable Simulations. *The Journal of Artificial Societies and Social Simulation (JASSS)*, Vol. 3 (No. 3).
 146. Rubine, D. Specifying Gestures by Example. *Computer Graphics*, Vol. 25 (No. 3). pp. 329 - 337.
 147. Scaffidi, C., Shaw, M. and Myers, B. An approach for categorizing end user programmers to guide software engineering research *Proceedings of the first workshop on End-user software engineering*, ACM, St. Louis, Missouri, 2005.
 148. Scheer, A.-W. *Architektur integrierter Informationssysteme. Grundlagen der Unternehmensmodellierung*. Springer, Berlin et al., 1991.
 149. Scheer, A.-W. and Werth, D. Geschäftsprozessmanagement für das Unternehmen von morgen. in Karagiannis, D. and Rieger, B. eds. *Herausforderungen in der Wirtschaftsinformatik*, Springer, Berlin Heidelberg New York, 2006, 49-64.
 150. Schiffer, S. Visuelle Programmierung - Potential und Grenzen. in *Beherrschung von Informationssystemen*, Heinrich C. Mayr, Oldenburg, 1996, 267-286.
 151. Schmelzer, H.J. and Sesselmann, W. *Geschäftsprozessmanagement in der Praxis: Produktivität steigern, Wert erhöhen, Kunden zufrieden stellen*. Carl Hanser Verlag, München Wien, 2007.
 152. Sellen, A., J., and Harper, R., H. R. *The Myth of the Paperless Office*. MIT Press, 2003.
 153. Serena. From Business Need to Business Mashup in Three Simple Steps, 2007.
 154. Shinomi, H. and Adams, S.S. Say goodbye to complexity when developing Web services - ADIEU: The End User Computing Tool for Web applications and Web services. IBM ed., 2005.
 155. Signer, B. Fundamental Concepts for Interactive Paper and Cross-Media Information Spaces, SWISS FEDERAL INSTITUTE OF TECHNOLOGY, ZURICH, 2005.

156. Signer, B. and Norrie, M.C. PaperPoint: a paper-based presentation and interactive paper prototyping tool *Proceedings of the 1st international conference on Tangible and embedded interaction*, ACM, Baton Rouge, Louisiana, 2007.
157. Smith, D.C. *A Computer Program to Model and Stimulate Creative Thought*. Birkhauser, Basel, 1977.
158. Snyder, C. *Paper Prototyping: The Fast and Easy Way to Design and Refine User Interfaces*. Elsevier LTD, Oxford, 2003.
159. Song, H., Guimbretière, F., Hu, C. and Lipson, H., ModelCraft: capturing freehand annotations and edits on physical 3D models. in *Proceedings of the 19th annual ACM symposium on User interface software and technology*, (Montreux, Switzerland, 2006), ACM.
160. Sousa, J.P., Schmerl, B., Poladian, V. and Brodsky, A. uDesign: End-User Design Applied to Monitoring and Control Applications for Smart Spaces *Seventh Working IEEE/IFIP Conference on Software Architecture (WICSA 2008)*, Vancouver, BC, 2008, 71-80.
161. Spahn, M., Dörner, C. and Wulf, V., End User Development: Approaches Towards a Flexible Software Design. in *Proceedings of 16th European Conference on Information Systems*, (Galway, Ireland, 2008).
162. Subrahmonia, J. and Zimmerman, T., Pen computing: challenges and applications. in *Pattern Recognition, 2000. Proceedings. 15th International Conference on*, (2000), 60-66 vol.62.
163. Suchman, L.A. *Plans and Situated Actions: The Problem of Human-Machine Communication*. Cambridge University Press, New York, 1987.
164. Thomas, O., Leyking, K. and Dreifus, F., Using Process Models for the Design of Service-Oriented Architectures: Methodology and E-Commerce Case Study. in *Hawaii International Conference on System Sciences, Proceedings of the 41st Annual*, (2008), 109-109.
165. Thurow, S. and Hummel, T. Wo BPM-Tools an Grenzen stoßen. *Computerwoche.de*.
166. van der Aalst, W., ter Hofstede, A. and Weske, M. Business Process Management: A Survey. in Reichert, M., Rinderle, S. and Dadam, P. eds. *Business Process Management*, Springer, Eindhoven, 2003, 1019-1019.
167. Weiser, M. The Computer for the 21st Century. *Scientific American*, 265 (3).
168. Wellner, P. Interacting with paper on the DigitalDesk, ACM, 1993, 87-96.
169. Winograd, T. From Programming Environments to Environments for Designing. *Comm. of the ACM, Vol. 38 (No. 6)*. 65 - 74.
170. Wobbrock, J.O., Wilson, A.D. and Li, Y. Gestures without libraries, toolkits or training: a \$1 recognizer for user interface prototypes *Proceedings of the 20th annual*

- ACM symposium on User interface software and technology*, ACM, Newport, Rhode Island, USA, 2007.
171. Wong, J. and Hong, J.I. Making mashups with marmite: towards end-user programming for the web *Proceedings of the SIGCHI conference on Human factors in computing systems*, ACM, San Jose, California, USA, 2007.
 172. Wulf, V. Anpaßbarkeit im Prozeß evolutionärer Systementwicklung. *GMD-Spiegel*, 24. Jg., 3/1994. S. 41-46.
 173. Wulf, V. and Jarke, M. The economics of end-user development. *Commun. ACM*, 47 (9). 41-42.
 174. Wulf, V., Pipek, V. and Won, M. Component-based tailorability: Enabling highly flexible software applications. *Int. J. Hum.-Comput. Stud.*, 66 (1). 1-22.
 175. Xiang, X. and Madey, G. End User Oriented Ad Hoc Web Services Composition in a Scientific Application Portal Architecture, University of Notre Dame, Notre Dame, 2004.
 176. Yeh, R.B., Brandt, J., Boli, J. and Klemmer, S.R., Interactive Gigapixel Prints: Large, Paper-Based Interfaces for Visual Context and Collaboration. in *Ubicomp*, (2006).
 177. Yeh, R.B., Brandt, J., Klemmer, S.R., Boli, J., Su, E. and Paepcke, A. Interactive Gigapixel Prints: Large Paper Interfaces for Visual Context, Mobility and Collaboration, Stanford University Computer Science Department Technical Report, 2006.
 178. Yeh, R.B., Klemmer, S.R. and Paepcke, A. Design and Evaluation of an Event Architecture for Paper UIs: Developers Create by Copying and Combining *Technical Report*, Stanford University Computer Science Department, 2007.
 179. Yeung, L., Plimmer, B., Lobb, B. and Elliffe, D. Levels of formality in diagram presentation *Proceedings of the 2007 conference of the computer-human interaction special interest group (CHISIG) of Australia on Computer-human interaction: design: activities, artifacts and environments*, ACM, Adelaide, Australia, 2007.

Abbildungsverzeichnis

Abbildung 1: Mentales Modell der Arbeit	4
Abbildung 2: BPM und WFM Lebenszyklus [166]	8
Abbildung 3: Serviceorientierte Architekturen am Beispiel von Web Services („SOA-Dreieck“) [143]	9
Abbildung 4: Geschäftsprozesse als Komposition von Services und menschlichen Aktivitäten [23]	12
Abbildung 5: Beispiele für die Modellierungsnotationen (a) BPMN, (b) eEPK und (c) UML (Aktivitätsdiagramm).....	15
Abbildung 6: Kontinuum von Endbenutzern nach Fischer [46]	19
Abbildung 7: Gestaltung eines EUD-Systems mit einem idealisierten Gentle Slope of Complexity [161].....	21
Abbildung 9: SPF-Notation [85]	44
Abbildung 10: Identifizierter Ablauf der Erstellung von BPM-Lösungen.....	46
Abbildung 11: Kästchen und Verbindungen	51
Abbildung 12: Designfläche des Workshops	51
Abbildung 13: Vereinfachte Darstellung der skizzierten Lösung [20].....	52
Abbildung 14: Gesamtkonzept von PaSeMod	68
Abbildung 15: Gestaltung der Papierinterfaces.....	70
Abbildung 17: Gestaltung der Modellierungselemente.....	75
Abbildung 18: Gestaltung der Notationsgesten.....	76
Abbildung 19: Verkürzte Darstellung einer Aktivitätsbeschreibungskarte (auf den tatsächlichen Karten sind die freien Antwortfelder mehrzeilig)	77
Abbildung 20: Gestaltung der Aktionskarten – Interaktionselemente auf der Vorderseite (links) und Modellierungselemente auf der Rückseite (rechts).....	78
Abbildung 21: Architekturkonzept von PaSeMod	87
Abbildung 22: Aufbau des Anoto Punktmusters.....	91
Abbildung 23: Nutzung von PaSeMod.....	96
Abbildung 24: PaSeMod GUI mit Prozessvisualisierung (links) und stiftbasierten Feedbackboxen (rechts).....	98

Abbildung 25: Auswahl an visuellen Rückmeldungen der letzten Modellierungsaktion	98
Abbildung 26: Vergleich papierbasiertes Modell (links) und computerbasierte Visualisierung (rechts).....	99
Abbildung 27: Hinweis beim Erstellen von Post-Its	99
Abbildung 28: Digitalstifte mit eindeutiger Zuordnung zu den Feedbackboxen	100
Abbildung 29: Prozessexport als Bild	100
Abbildung 30: Prozessexport nach Marama.....	100
Abbildung 31: Angepasste Aktionskarten.....	105
Abbildung 32: Workshop-Setting.....	105
Abbildung 33: Übersicht der skizzierten Prozessmodelle (oben der Prozessteil ohne Post-Its, unten der mit Post-Its; links abgebildet ist jeweils das Papiermodell, rechts das entsprechende digitale Prozessmodell)	106

Abkürzungsverzeichnis

ARIS	Architektur integrierter Informationssysteme
BPEL	Business Process Execution Language
BPM	Business Process Management
BPMN	Business Process Modeling Notation
BPR	Business Process Reengineering
CD	Cognitive Dimension
CORBA	Common Object Request Broker Architecture
eEPK	Erweiterte Ereignisgesteuerte Prozessketten
EUD	End-User Development
EUSE	End-User Software Engineering
HCI	Human Computer Interaction
IT	Informationstechnologie
NLP	Natural Language Programming
NP	Natural Programming
PADD	Paper Augmented Digital Documents
PbE	Programming by Example
PBUI	Papierbasiertes Benutzerinterface
PSD	Process Solution Description
RPC	Remote Procedure Calls
SOA	Serviceorientierte Architekturen
SPF	Solution Process Flow
SUC	System Use-Case
UML	Unified Modeling Language
VP	Visuelle Programmierung
WFM	Workflow-Mangement
WSDL	Web Service Description Language
XML	Extensible Markup Language

Tabellenverzeichnis

Tabelle 1: Übersicht endbenutzerorientierter Modellierungs- und Orchestrierungssysteme ...	28
Tabelle 2: Übersicht der Chancen und Herausforderungen papierbasierter Anwendungen	32
Tabelle 3: Gestaltung von stiftbasiertem Feedback [99].....	38
Tabelle 4: Übersicht der Cognitive Dimensions von Green et al. [53, 54]	58
Tabelle 5: Übersicht der Anforderung an PaSeMod	67
Tabelle 6: Mögliche Feedbackmechanismen für das Konzept.....	81
Tabelle 7: Feedbackgestaltung für die Bestätigung von Aktionen.....	83
Tabelle 8: Feedbackgestaltung für die Unterstützung der Modellierung	83
Tabelle 9: Digitalstifthersteller (Anoto-Technologie) und Stiftmodelle	92
Tabelle 10: Übersicht interaktive Papier- und Sketching-Frameworks.....	93

Anhang

A. Leitfaden der Fallstudie

I. Introduction

- **Organization Issues**
- **Background and aim of the interview**

II. Interview

1. Short information about personal data and organizational knowledge.

- What is your Job in the organization? / Where are you located within the organization?
- In which department do you work and how many people work there?
- What tasks do you typically have to process?
- How long do you work for the company?

2. Understanding und type of processes

- What is a process in your eyes? How would you define it?
 - Do user interactions play a role in these processes?
 - Is this process understanding shared in the whole organization?
- What kind of processes do you model within the organization? (Automatic/user-centered)?
 - How complex are those processes?
 - How critical are they?
 - Do you have any private processes?
- Which problems are solved by the processes?
- How many processes do you have in your organization?

3. Process Development

- Process Evolution
 - Can you try to estimate how often processes and single services have to be changed?
- How often you create new processes?
- How long does it take to develop a typical process?
- How often are single services used in different processes?
- Process modeling
 - Is there some kind of model, which should be followed during the process?
 - Which persons are involved in the modeling process and which competencies do these persons have?
 - What contribute the departments?
 - What contributes the IT?
 - How much influence/power should the departments have in the development of processes?
 - How important is collaboration between the departments in the IT in the modeling process?
 - How do you capture requirements to model the process?
 - Which media and visual representations do you use when you talk with the users from the departments about the processes?
 - Do you also use whiteboards, paper, etc.?
 - In which spatial context does the modeling take place?
 - Is there a special formalism (e.g. modeling language xyz)?
 - Do you capture formal and/or informal aspects? How?

- What is the result of the analysis? A process model?
- Which tools and techniques are used to model?
 - What are the reasons for those tools?
 - (Do you use BPEL or BPMN?)
- How detailed (how many aspects are included – exception handling, message flows, etc.) are the processes that you model?
- What are typical problems during the modeling process?
- To what extent do you think have users the ability to model processes?
- Process development
 - How does the transformation from requirements to process models work?
 - Does the process modeling have iterations?
 - What do you do, if the process is specified imprecise?
 - Is it possible to reuse parts of the processes in other processes?
 - Are processes/services are shared within the company?
 - Which processes could be adapted by end users (from a business point of view)?
 - Is there a kind of beta-testing?
- Feedback for the design idea
 - How would you imagine the modeling of processes on digital paper? (First give a short impression of our solution idea of paper-based modeling)

4. Technical details

- How good are services/interfaces documented?
 - What is missing?
 - Is there any user documentation?
- How do you know, which services exist? How does the "search process" work?
 - Is there a central repository?
- Do you use internal and/or external services?
 - Services are sliced according to which criteria?
- Think about a function that could recommend services? Would it be useful and why?

5. Reconciliation

- What are main differences between processes in your previous environment and your post-SOA environment?

B. Beispiel eines wohldefinierten System Use-Cases

WSUC for Automatic: <Create cards>

Change log

Version	Author	Date	Description of change

Description

Briefly describe why this operation is called

This service is called once for every card to be created automatically according to the customer package and the results of the previous automatic activity <Find internal account number>.

Operation

Type: Operation in KR<Sub flow | Micro flow | Mediation flow | Operation in KR>

Name: KCSI_OPRETKORT<Name of flow or name of operation in KR>

Version: 02<Version of flow or version of operation in KR>

Operation attributes

Max. of retry: 3<Number of retry for runtime errors before escalation>

Timeout: <Number of seconds before timeout, default is the KR definition+10>

Escalation Type: Manual activity<Peregrine | Manual activity | both>

If Manual activity is involved a WSUC for the manual activity should be made

If the automatic creation of a card fails, the card should be created with the manual activity <Create Card Manually>

Pre-conditions

Activity pre-condition

This service should be called when the card can be created automatically. I.e., when the card data from the customer package does not contain the tag <FORCEMANU> and the internal account number is known or was successfully found by the automatic WSUC <Find Internal Account Number>.

Service Input

Specification of constraints on input fields for this activity, e.g. how should they be assigned. All input field should be mention.

KCSI_FASTI_V01

Input field name	Value	Reference	Description
<field name>	<constraints on the value, format or fixed value or how to assign it from other field>		<Description of the field>
DYNCALL_PGMNR	""		

VER_I	1		
VER_O	1		
(23 more to come)

KCSI_OPRKORT_IDL1

Input field name	Value	Reference	Description
<field name>	<constraints on the value, format or fixed value or how to assign it from other field>		<Description of the field>
KKDYNPAKKEID	“0”		
OPRETP	“”		
KALDSYS	“7”		
(62 more to come)

Results (Post conditions)

Output:

Specification of return fields that should be post validated. This is only technical or runtime errors that should be mention here. Business error should be model in the SPF. The action taken must be related to the number of retry and the escalation type.

Output fields	Validation	Escalation
<Fields that are contained in a validation>	<Description of the validation>	<Set activity stopped Peregrine Manual activity Peregrine and Manual activity>
KCSI_OPRETKORT/ KCSI_FASTO_V01/RTKD	0 = OK	! = 0: Card creation must be handled manually by WSUC <Create Card Manually>

C. Dokumententypdefinition (DTD) der XML-Exportstruktur

```

<!ELEMENT SketchedProcess (ProcessModelList, UseCaseList?)>
<!ELEMENT ProcessModelList (ProcessModel+)>
<!ELEMENT UseCaseList (UseCase*)>
<!ELEMENT ProcessModel (Activity*, Comment*, Connection*,
InformalAnnotations*)>
<!ATTLIST ProcessModel
    id ID #REQUIRED
    subprocessOf IDREF #IMPLIED>
<!ELEMENT Activity (ElementStroke, ContainedStrokes?)>
<!ATTLIST Activity
    id ID #REQUIRED
    posX CDATA #REQUIRED
    posY CDATA #REQUIRED
    attachedUseCase IDREF #IMPLIED
    hasSubprocess IDREF #IMPLIED>
<!ELEMENT Comment (ElementStroke, ContainedStrokes?)>
<!ATTLIST Comment
    id ID #REQUIRED
    posX CDATA #REQUIRED
    posY CDATA #REQUIRED>
<!ELEMENT Connection (ElementStroke)>
<!ATTLIST Connection
    id ID #REQUIRED
    posX CDATA #REQUIRED
    posY CDATA #REQUIRED
    type (SEQUENCE|ASSOCIATION) #REQUIRED
    source IDREF #REQUIRED
    target IDREF #REQUIRED>
<!ELEMENT InformalAnnotations (InkStroke*)>
<!ELEMENT ElementStroke (InkStroke)>
<!ELEMENT ContainedStrokes (InkStroke*)>
<!ELEMENT UseCase (UseCase_Name, UseCase_Annotation, UseCase_Description,
    UseCase_Function, UseCase_Input, UseCase_Output)>
<!ATTLIST UseCase
    id ID #REQUIRED
    hasSubprocess (true|false) "false"
    activityType (MANUAL|USER_EXPERIENCED|AUTOMATIC|UNDEFINED) #REQUIRED>
<!ELEMENT UseCase_Name (InkStroke*)>
<!ATTLIST UseCase_Name
    recognisedText CDATA #IMPLIED>
<!ELEMENT UseCase_Annotation (InkStroke*)>
<!ATTLIST UseCase_Annotation
    recognisedText CDATA #IMPLIED>
<!ELEMENT UseCase_Description (InkStroke*)>
<!ATTLIST UseCase_Description
    recognisedText CDATA #IMPLIED>
<!ELEMENT UseCase_Function (InkStroke*)>
<!ATTLIST UseCase_Description
    recognisedText CDATA #IMPLIED>
<!ELEMENT UseCase_Input (InkStroke*)>
<!ATTLIST UseCase_Input
    recognisedText CDATA #IMPLIED>
<!ELEMENT UseCase_Output (InkStroke*)>
<!ATTLIST UseCase_Output
    recognisedText CDATA #IMPLIED>
<!ELEMENT InkStroke (InkPoint+)>
<!ATTLIST InkStroke
    <!--id ID #REQUIRED-->

```

```
    startTimestamp NMTOKEN #REQUIRED
    endTimestamp NMTOKEN #REQUIRED>
<!ELEMENT InkPoint EMPTY>
<!ATTLIST InkPoint
  x CDATA #REQUIRED
  y CDATA #REQUIRED
  timestamp CDATA #REQUIRED>
```

D. Critical Incidents des Experten Walkthroughs

CI	Beschreibung	Designempfehlung	Überarbeitung	Ge- löst
Aktionskarten				
1	Der Benutzer wundert sich über die Bedeutung nicht verfügbare Stiftmodi (2. Zeile der Stiftmodi auf den Aktionskarten). Der Benutzer wundert sich ebenso über die nicht verfügbaren Prozesselemente von der Rückseite der Aktionskarte.	Elemente für den Prototyp weglassen	Aktionskarte dementsprechend umgestalten (nicht verfügbare Elemente weglassen) und gleichzeitig die Aktionskarte kleiner machen.	X
2	Der Benutzer versteht die Bedeutung der Exportieren-Felder nicht (entspringen nicht der Sprache der Endbenutzer).	Nur Feld „Geschäftsprozess Speichern“ (als Bild)	Nur ein Feld zum exportieren: „Geschäftsprozess speichern“.	x
3	Der Benutzer weiß nichts mit der Beschriftung „Aktivitätskarte #x,, anzufangen.	Weglassen, da für den Benutzer irrelevant	Weglassen.	x
20	Der Benutzer wundert sich, dass sich, wenn er zweimal den Löschmodus aktiviert, der Stift wieder im normalen Modus befindet	Inkonsistenz beseiten.	Toggle des Modus deaktivieren.	x
Aktivitätsbeschreibungen				
4	Benutzer wunder sich, was für eine Bedeutung das Feld Annotation der Aktivitätsbeschreibungen hat.	Feld weglassen, da Prozessgesten nicht verfügbar sind.	Weglassen. (Oder ändern in „Wer führt die Aktivität durch?“)	x
5	(Aktivitätsbeschreibungen) Der Benutzer wundert sich, warum einmal nach Informationen und einmal Ergebnisdaten gefragt wird.	Beide Ergebnisdaten	Beide mal den Begriff „Ergebnisdaten“ verwenden.	x
6	(Aktivitätsbeschreibungen) Der Benutzer wundert sich, warum nur im ersten Feld Skizzen möglich sind. (Irritierende Beschriften „dürfen auch Skizzen ...“)	Hinweis weglassen, da diese überall möglich sind.	Zusatz weglassen.	x
7	Der Benutzer wundert sich warum bei der Aktivitätsbeschreibung der Typ der Aktivität durchgestrichen werden muss anstatt es zum Beispiel einzukreisen.		Felder vergrößern und Feedback geben, wenn ein Typ ausgewählt wurde.	
8	Der Benutzer findet die Aktionsbeschreibungen „labbrig“ und dünn.		Wenn möglich auf dickem Papier (160g?) ausdrucken.	
Prozesspapier				
9	Der Benutzer wundert sich über die englische Bezeichnung der Prozesspapiere („PROCESSPAPER	Nach deutsch übersetzen	„Prozesspapier #x“.	x

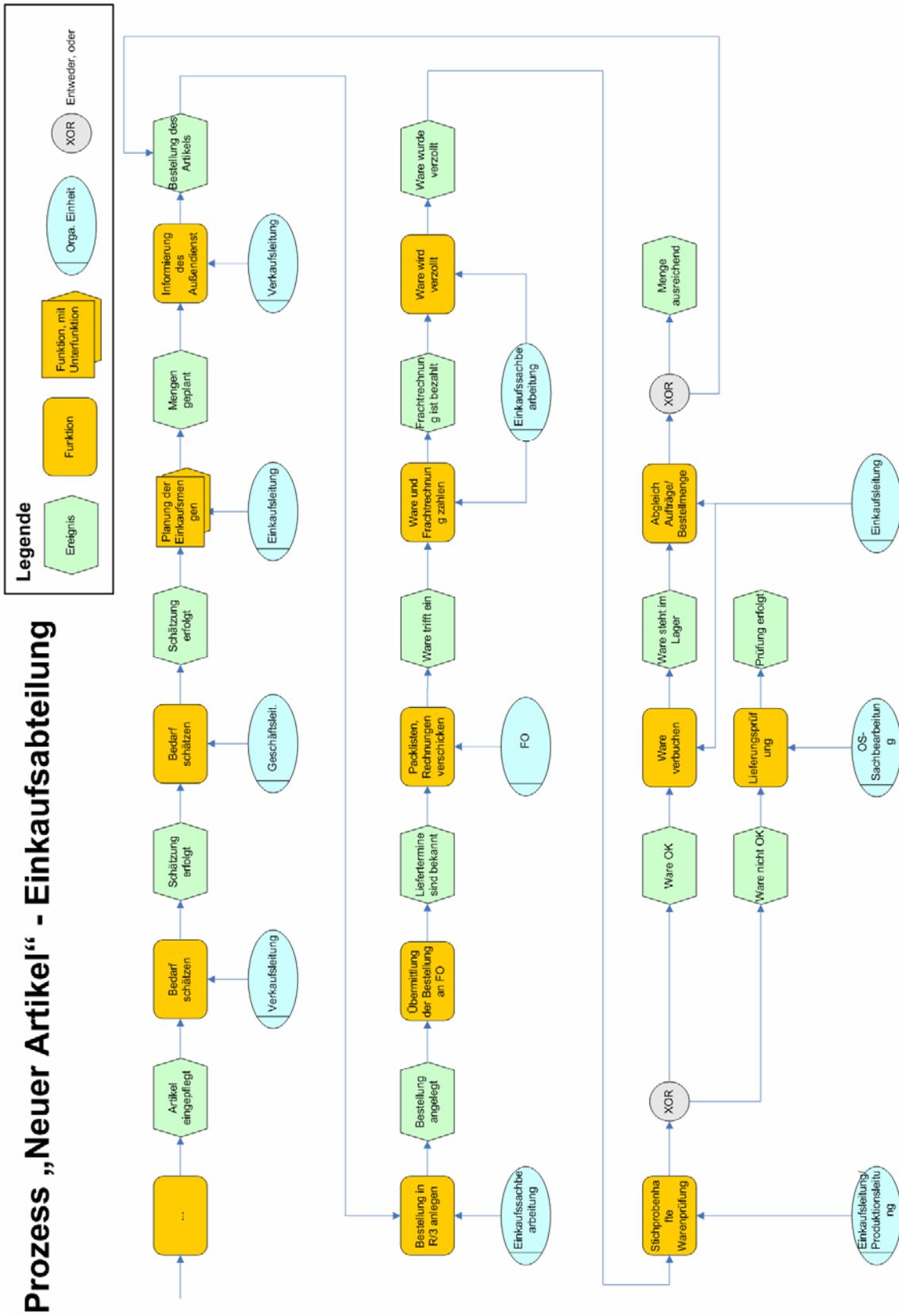
	#x“).			
Feedback				
10	Der Benutzer wundert sich über das „Ping“, wenn er ein Element gezeichnet hat.	In der Einführung/Moderation erklären, was das „Ping“ bedeutet	Keine Änderung des Systems/Interfaces. In der Einführung berücksichtigen/erklären.	x
11	Der Nutzer fragt sich, warum der Stift manchmal vibriert.	Einführung: Erklären, dass dies im Prototyp ignoriert werden muss.	Keine Änderung möglich. In der Einführung berücksichtigen/erklären.	x
12	Der Benutzer wundert sich was er machen muss, dass das „aufgepoppte“ (wenn ein neues Post-It aktiviert wurde) Fenster wieder verschwindet.	Hinweistext, dass er das Post-It verbinden sollte	Wenn mit akzeptablem Aufwand möglich dem Benutzer einen Hinweistext anzeigen, dass er das Post-It mit einem Prozesspapier verbinden muss um es zuzuordnen.	
13	Der Benutzer kann den Text auf dem Bildschirm/Beamer schlecht erkennen (zu klein geschrieben).	Zoom der Tinten-Anzeige und der Textanzeige korrigieren	Standard-Zoom ändern und größere Schrift verwenden. (Systemfontgröße auf extra-groß stellen!)	x
14	Der Benutzer wundert sich, dass manche auf dem Papier skizzierten Dinge nicht auf dem PC erscheinen.	Darauf Hinweisen – ist implementierungstechnisch schwierig umsetzbar	Keine Änderung des Systems/Interfaces.	x
15	Der Benutzer wundert sich, warum es verschiedene Hinweistöne gibt, und welche Bedeutung diese haben.	Differenzierung deutlich machen und erklären oder entfernen.	Keine Änderung des Systems/Interfaces. In der Einführung berücksichtigen/erklären.	x
16	Der Benutzer wundert sich über fehlendes Feedback, wenn er ein Post-It mit dem Prozesspapier verbunden hat. (Kein Ton beim verbinden von Post-Its...)	Ton, wie beim Verbinden der Aktivitätsbeschreibung mit einer Aktivität.	Korrigieren.	x
17	Der Benutzer kann GUI-Elemente (Beamer) schlecht erkennen, da schwarze Farbe auf grauem Grund und Symbole auf grauem Grund.	Verwendung kontrastreicherer Farbe in der GUI.	Farbschema der GUI ändern.	x
21	Der Benutzer fragt sich, welches Element er als letztes erstellt hat (und dementsprechend ändern kann).	Zuletzt gezeichnetes Element besonders hervorheben.	Erstmal nicht implementieren.	x
22	Der Benutzer findet es schwierig in der GUI zwischen dem stiftbezogenen Feedback und der Prozessvisualisierung hin und her zu wechseln			
Allg. Modellierung				
18	Der Benutzer wundert sich, dass er nachträglich immer nur der letzte Strich digital korrigiert werden (Änderung der formalen Bedeutung).	Jedes (formale) Element nachträglich veränderbar machen.	Keine Änderung des Systems/Interfaces. In der Einführung berücksichtigen/erklären.	x

19	Der Benutzer wundert sich, dass er ein neues Papier immer erst „aktivieren“ muss und dass kein Hinweis kommt, wenn dies nicht geschieht.	Darauf explizit hinweisen, oder im System ändern	Keine Änderung des Systems/Interfaces. In der Einführung berücksichtigen/erklären.	x
23	Der Benutzer ärgert sich, dass nicht alle Elemente so erkannt werden, wie er sie gemeint/skizziert hat.	Bessere Erkennung einbauen.	Erstmal nicht implementieren.	x

E. Interviewleiten der Evaluationsinterviews

- 1) Was bietet das System? Was kann man mit dem System machen?
- 2) Wie ist generell Ihr "erster" Eindruck von diesem System? (Hat das System die Erwartungen erfüllt, die Sie im Vorfeld an ein solches System stellen?)
- 3) Wie war der Umgang mit dem System?
 - 3.1) War der Umgang mit dem System mit (vielen) Problemen behaftet?
 - 3.2) Konnten Sie die Aufgaben, die Sie mit dem System gelöst haben, ohne Umschweife lösen?
 - 3.3) Präsentiert das System die Informationen klar und verständlich?
 - 3.4) Haben Sie das Gefühl, dass Sie Unterstützung benötigen, wenn Sie das System verwenden?
 - 3.5) Sind die Hinweise und Antworten durch das System hilfreich? Wie gut haben Ihnen insbesondere die Rückmeldungen des Systems genützt?
 - 3.6) Waren für Sie die Informationen und Interaktionselemente (wie Darstellung und Bedeutung der Prozesselemente, verschiedene Papiere, Stiftstatus) verständlich? Waren die Bedeutung der verschiedenen Papiere verständlich?
 - 3.7) Konnten Sie mit akzeptablem Aufwand Fehler korrigieren oder rückgängig machen?
 - 3.8) Konnte Sie das System bei der gemeinsamen Prozesserstellung unterstützen? Wie könnte das gemeinsame Arbeiten besser unterstützt werden?
 - 3.9) Inwiefern wirke sich die Verwendung des Stiftes auf die Prozessmodellierung aus?
- 4) Sehen Sie einen grundsätzlichen Bedarf nach einem solchen System (bei Ihnen selbst, bei Kollegen, in Ihrer Firma)?
- 5) Wie müsste ein solches System gestaltet sein/verbessert werden, damit Sie das System in Ihrem Arbeitsalltag nutzen (bzw. was wären die Voraussetzungen/Bedingungen)?
 - a. Wie wichtig ist es, dass ein solches System (im Arbeitsalltag) übersichtlich gestaltet und klar gegliedert ist?
- 6) Glauben Sie, dass Ihre Bedürfnisse als Benutzer in diesem System berücksichtigt worden sind?
- 7) Gibt es etwas, dass Sie sich wünschen würden, was dieses System bis jetzt noch nicht bietet?

F. Szenarioprozess der Evaluation



G. Critical Incidents der Evaluation (Benutzungstest)

CI	Beschreibung / Kommentar
1	<p><i>Der Benutzer wundert sich, dass es kein Zustands-Element gibt. Er würde aufgrund seiner Erfahrung ein eigenes Element („Zustand“) für die Beschreibung eines Prozessstatus verwenden.</i></p> <p>Von der Idee her, würden „Zustände“ als informelle Elemente skizziert, da sie keine Relevanz für die Umsetzung der serviceorientierte Prozessmodellierung haben.</p> <p>Jedoch wäre zu überlegen, ob nicht Nutzern die Möglichkeit geboten werden sollte, eigene Elemente zu spezifizieren oder vorgegebenen Elementen eine spezifische Bedeutung zuzuweisen.</p>
2	<p><i>Der Benutzer versucht die digitale Repräsentation des Prozesses auf dem Bildschirm möglichst „sauber“, das heißt frei von ungewollten informellen Elementen, zu halten und löscht alle nicht beabsichtigten Striche aus dem digitalen Modell. Dabei produziert er auf dem Papier jedoch mehr Striche als nötig, was zu einer gewissen Unübersichtlichkeit und Divergenz zwischen Papiermodell und digitalem Modell führt.</i></p> <p>Den Nutzer muss bewusst gemacht werden, dass das Ziel nicht ein möglichst finales und formales Modell ist, sondern die Erfassung semi-formaler und kreativer Aspekte. Dazu muss den Nutzern auch die Weiterverarbeitung deutlich gemacht werden.</p>
3	<p><i>Der Benutzer ärgert sich, dass durch viele Änderungen das Blatt unübersichtlich ist. Daher arbeitet er schließlich schwerpunktmäßig mit der Beameransicht.</i></p> <p>Dies bedingt sich durch das in CI 2 geschilderte Verhalten.</p>
4	<p><i>Der Nutzer wechselt öfter bewusst aus dem normalen Modus in den Annotationsmodus, um eine formales Element zu beschreiben (zu „annotieren“).</i></p> <p>Hier irritiert die ungeschickte Wortwahl von „Annotationsmodus“ und „Element annotieren“ den Benutzer und suggeriert ihm so eine Verbindung zwischen beiden. Dem könnte durch eine bessere Benennung des Modus entgegengewirkt werden.</p>
5	<p><i>Der Nutzer versucht im falschen Modus (Annotationsmodus) formale Elemente zu erstellen. Er erkennt es nicht auf Anhieb, dass sich der Stift im falschen Modus befindet und wundert sich, dass das Element nicht erkannt wird.</i></p> <p>Dies bedingt sich durch das Fehlen einer direkt sichtbaren Visualisierung des aktiven Modus direkt über den Stift. Jedoch wäre in der derzeitigen Form ein implizites Feedback möglich, wenn der Stift sich nicht im normalen Modus befindet.</p>
6	<p><i>Der Benutzer löscht aus Versehen ein Element, weil er sich im falschen Stiftmodus (Löschmodus) befindet. Er ärgert sich danach, dass er das Element nicht wiederherstellen kann.</i></p> <p>Hier gilt dasselbe wie für CI 5. Zudem erscheint eine Undo-Funktion oder eine besondere Bestätigung der Löschfunktion überlegenswert.</p>
7	<p><i>Der Benutzer erwartet, dass er Post-Its Umplatzieren („Verschieben“) kann, weiß aber nicht wie. Zu Beginn sucht er einen extra „Verschiebmodus“ für den Stift. Die korrekte Lösung wird nur mit viel Hilfe erreicht.</i></p> <p>Für den Nutzer ist nicht intuitiv, dass er erst die Verbindung von Post-Its und Prozesspapier durch einen kurzen Strich im Löschmodus entfernen muss und dann im normalen Modus das Post-Its wieder verbinden kann. Das vorher notwendige Entfernen ist bedingt durch den Prototyp.</p>
8	<p><i>Der Benutzer löscht ein Kommentar-Post-It und ärgert sich, dass beim neuen Verbinden des Post-Its der Inhalt des Kommentars verschwunden bleibt (obwohl er dieses Verhalten bereits befürchtet hatte).</i></p> <p>Um Inkonsistenzen zwischen dem papier- und dem digitalen Modell zu vermeiden könnte der Inhalt, trotz dass das Element gelöscht wurde, gespeichert und dann beim Neuerstellen wieder eingefügt werden. Dies würde auch ein Verschieben von Post-Its (CI 7) und ein versehentliches Löschen von Elementen (CI 6) vereinfachen bzw. vermindern.</p>
9	<p><i>Der Nutzer versucht ein Element um ein anderes herum zu zeichnen. Dies wird von System geblockt. Er wundert sich, warum ein formales Notationselement nicht als solche erkannt wird.</i></p> <p>Das System liefert keinerlei Feedback warum ein Element nicht erkannt wurde. Wenn möglich, sollte der Nutzer ein Feedback über die Ursache erhalten. (Handelt es sich um ein Erkennungsproblem handelt, ist</p>

	ein falscher Stiftmodus aktiv oder erkennt das System bewusst nicht, weil sich Elemente überlappen oder er ein Element um ein anderes herum gezeichnet wurde.)
10	<i>Der Benutzer versucht sich überlagernde Elemente zu zeichnen und wundert sich, warum das System diese nicht erkennt.</i>
	Hier gilt dasselbe wie für CI 9.
11	<i>Der Benutzer wundert sich, dass die Aktivitätskarte auf dem Computer nicht visualisiert wird.</i>
	Das vollständige Ausbleiben des Feedbacks irritiert den Nutzer. Hier sollte das System in irgendeiner Form Feedback geben.
12	<i>Der Benutzer wundert sich, dass das System nicht speichert, obwohl er mehrfach die Speicherfunktion mit dem Stift angetippt hat.</i>
	Hier kam das Feedback des Systems zu spät, da intern erst der Server zur Handschriftverarbeitung geladen werden musste. Das Feedback kam also erst nachdem die Speicheraktion abgeschlossen war. Hier sollte das Feedback direkt kommen.
13	<i>Der Benutzer erwartet, dass er bestimmte Elemente auf dem Bildschirm hervorheben kann, damit andere erkennen welches Element er meint.</i>
	Ein solcher Modus bzw. eine solche Funktion entspricht der Communicative Dimension der Referencability und würde somit die Kollaboration anreichern.
14	<i>Der Nutzer erwartet intuitiv, dass er jede erstellte Skizze (also formelle sowie informelle Elemente) löschen kann und wundert sich, dass dies bei informellen Elementen nicht möglich ist.</i>
	Dies knüpft an die Ausführungen von CI 2 an. Ein geeignetes Löschen informeller Inhalte ist jedoch weder gut realisierbar noch erscheint er sinnvoll, da dies zu Inkonsistenzen zwischen dem digitalen und dem Papiermodell führt und so die bei CI 2 geschilderten Verhaltensweisen auslöst.
15	<i>Der Benutzer erwartet, dass man auch Richtungspfeile an die Verbindungen anbringen kann.</i>
	Dies ging nicht, da der Prototyp keine zusammengesetzte Gesten, sondern nur Einstrichgesten erkennt. Mit einem komplexen Algorithmus ist dies jedoch realisierbar.
16	<i>Die Benutzer wunderten sich, dass nicht zu erkennen war, ob mit einer Aktivität eine Aktivitätsbeschreibung verbunden ist.</i>
	Dies ist auf Fehlen der Notationsgesten und des Informationsmodus (über den der Status abgefragt werden könnte) im Prototyp zurückzuführen.
17	<i>Der Benutzer ist kurz irritiert, dass er dem Prozess beim Speichern keinen Namen zuweisen kann. Er geht dann davon aus, dass nach dem Speichern das System eine interne und für ihn verständliche Kennung vergibt.</i>
	Dies ist bedingt durch die prototypische Natur des Systems. Die Speicherung findet daher nur implizit statt.