

Gottfried Wilhelm Leibniz Universität Hannover
Fakultät für Elektrotechnik und Informatik
Institut für Praktische Informatik
Fachgebiet Software Engineering

**Entwurf und Implementierung
eines Werkzeugs zur Unterstützung
von Interviews in der Prozessmodellierung**

Bachelorarbeit

im Studiengang Informatik

von

Lars de Vries

Prüfer: Prof. Dr. Kurt Schneider
Zweitprüfer: Prof. Dr.-Ing. Helena Szczerbicka

Hannover, 18. August 2006

Erklärung

Hiermit versichere ich, dass ich die vorliegende Bachelorarbeit selbstständig und ohne fremde Hilfe verfasst habe und keine anderen als die in der Arbeit angegebenen Quellen und Hilfsmittel verwendet habe.

Hannover, 18.08.2006 Lars de Vries

Inhaltsverzeichnis

1 Einleitung	4
1.1 Motivation	4
1.2 Aufgabenstellung und Randbedingungen	4
1.2.1 Anforderungserhebung für eine entsprechende Software	5
1.2.2 Entwurf und Umsetzung grundlegender Eigenschaften.....	5
1.2.3 Prozessmodelle	5
1.3 Zur Erläuterung der zu erstellenden Software: Use Cases	6
1.3.1 Use Case 1	6
1.3.2 Use Case 2	7
1.4 Gliederung der Arbeit	8
2 Zu verwendende Notationen	9
2.1 IDEFØ	9
2.2 FLOW	11
3 Recherchen und Erkenntnisse	12
3.1 Zusammenfassung der Recherche und Einzelergebnisse	12
3.1.1 Untersuchung von Interviews im Allgemeinen	12
3.1.2 Untersuchung von Interviews in der Anforderungserhebung.....	13
3.1.3 Gesamtauswertung der Untersuchungen	17
3.1.4 Extraktion von Anforderungen für die Modellierungssoftware	19
3.2 Erweiterung der Aufgabenstellung im Zwischenvortrag	20
3.2.1 Problematik Informationsfluss/Dokumentenfluss	20
3.2.2 Lösungsansätze.....	20
3.2.3 Entscheidung und Lösung	23
4 Ergebnisse	25
4.1 Anforderungserhebung.....	25
4.1.1 Zusammenfassung der gesammelten Anforderungen.....	25
4.1.2 Anforderungsdokument	27
4.2 Prozessmodell	31
4.3 Entwurf	31
4.3.1 Usability	31
4.3.2 Datenhaltung und -verwaltung.....	34
4.4 Software	35
4.4.1 Wiederverwendbare Steuerelemente	36
4.4.2 Funktionen des Programms.....	38
4.4.3 Schnittstellen	38
4.4.4 Nebenaspekte und Abstriche.....	39
5 Ausblick	41
6 Quellen- und Abbildungsverzeichnis	42
7 Anhang	A
7.1 Software-Installation	A
7.1.1 Benötigtes Framework.....	A
7.1.2 Beiliegender Quellcode.....	A

1 Einleitung

1.1 Motivation

In Softwareprojekten ist es die Aufgabe von *Entwicklern*, für einen *Auftraggeber* eine Software zu erstellen. Meistens sind die genauen Anforderungen an das zu entwickelnde Produkt nicht genau formuliert, Details und Umfang der zu implementierenden Funktionalitäten müssen während einer ersten Phase des Projekts, der *Anforderungserhebung*, von den Entwicklern selbst geklärt werden.

Hierbei gehört es zum üblichen Vorgehen, *Interviews* mit Beteiligten durchzuführen, die entweder die Software später nutzen sollen oder den Ist-Zustand der Umgebung kennen, in dem die Software zum Einsatz kommen soll.

Die zu entwickelnde Software soll möglicherweise die bestehenden *Geschäftsprozesse* aufgreifen und verwenden, sie gegebenenfalls sogar verbessern, auf jeden Fall aber mit Ihnen zusammenarbeiten. Um den Software-Entwicklern diese Prozesse zu vermitteln müssen *Prozessmodelle* zum Einsatz kommen, mit denen die vorliegenden Prozesse beschrieben werden.

In besonderen Fällen kann es sein, dass noch nie eine derartige explizite Formulierung der vorliegenden Prozesse vorgenommen wurde. Die Beteiligten auf der Seite des Auftraggebers werden im Rahmen von Interviews mit den beauftragten Entwicklern vielleicht erstmalig die notwendigen Prozesse nachvollziehen und zusammenhängend erläutern müssen.

Aus den Ergebnissen dieser Interviews (Notizen, Mitschnitte, Skizzen etc.) erarbeiten dann die Entwickler anschließend entsprechende Prozessmodelle. Diese werden in den meisten Fällen zunächst unvollständig und/oder fehlerhaft sein und erst nach mehrfacher Rücksprache und einigen Korrekturen zufrieden stellend die Geschäftsprozesse des Auftraggebers wiedergeben.

Vor diesem Hintergrund stellt sich die Frage, ob es mittels eines auf dem Laptop mitgebrachten Software-Werkzeuges den Entwicklern möglich wäre, sofort nach dem Ende eines Interviews – oder sogar zwischendurch – die bis dahin erhobenen Prozesse darzustellen und mögliche offene Punkte und Inkonsistenzen zu markieren, so dass das Gespräch gezielt auf diese Fragen gelenkt werden könnte.

In wie weit existierende Prozessmodellierungstools oder moderne Technik wie die Verwendung von Tablet-PCs Vorteile im Interview bringen, sind weiterreichende Themen, die in diesem Zusammenhang von Interesse sind.

Die Idee, sich im Rahmen dieser Aufgabenstellung mit dieser Thematik zu befassen, stammt aus der Doktorarbeit von Markus B. Strohmaier [1].

1.2 Aufgabenstellung und Randbedingungen

Die Aufgabenstellung dieser Arbeit gliedert sich grundsätzlich in die Bereiche:

- Anforderungserhebung für eine entsprechende Software (s. oben),
- Entwurf einer diesen Anforderungen entsprechenden Software und
- Umsetzung des Entwurfs im Rahmen dieser Arbeit,

und wurde im Verlauf ergänzt um die

- Untersuchung des zu verwendenden Prozessmodells.

Im Folgenden werden die Aspekte der Aufgabenstellung genauer erläutert.

1.2.1 Anforderungserhebung für eine entsprechende Software

Wenn der Zweck einer Software, wie sie unter 1.1: *Motivation* beschrieben wird, auch feststeht, sind jedoch noch die Details der Anforderungen zu klären. Eine entsprechende Anforderungserhebung umfasst die Klärung der

- Mission des Projekts,
- Rahmenbedingungen,
- funktionalen Anforderungen und
- Qualitätsanforderungen, sowie der
- Probleme und Risiken.

Diese Punkte sollen untersucht, begründet und beschrieben werden.

1.2.2 Entwurf und Umsetzung grundlegender Eigenschaften

Die Aufgabenstellung umfasst weiterhin die Umsetzung der erhobenen Anforderungen in eine funktionsfähige Software, so weit es im (Zeit-)Rahmen dieser Arbeit möglich ist. Der dazugehörige Entwurf soll verwendete Konzepte vorstellen, unter Beachtung der Prinzipien des Software Engineering erstellt und aufgrund begründeter Entscheidungen realisiert werden.

1.2.3 Prozessmodelle

Für die Prozessmodellierung ist die Modellierungssprache IDEFØ¹ zu verwenden. Zudem sollen die im Rahmen von FLOW² definierten Aggregatzustände für Informationen in der Darstellung zum Einsatz kommen.

Als im Verlauf der Arbeit deutlich wurde, dass eine Vereinigung der unterschiedlichen Ansätze dieser Modelle nicht ohne weiteres geschehen kann (siehe Abschnitt 3.2.1: *Problematik Informationsfluss/Dokumentenfluss*), wurde die Aufgabenstellung um den Auftrag erweitert, Verbindungsmöglichkeiten dieser Darstellungen zu untersuchen und ein Konzept für die gemeinsame Darstellung zu entwickeln.

¹ siehe 2.1: *IDEFØ*

² siehe 2.2: *FLOW*

1.3 Zur Erläuterung der zu erstellenden Software: Use Cases

Teil der Aufgabenstellung ist es, eine Anforderungserhebung für die zu erstellende Software durchzuführen. Der „Auftraggeber“ der Software ist dabei Herr Prof. Kurt Schneider als Steller dieser Bachelorarbeit. Die in Gesprächen mit ihm herausgearbeitete gewünschte Funktionsweise der geforderten Software wird hier wiedergegeben. Ihre Notwendigkeit und den Zweck schildert die „Hintergrundgeschichte“ in 1.3.1. Diese Darstellung ist für Außenstehende und Laien besonders gut verständlich. Auf Details der Anwendung wird in der Form eines klassischen Use Case in Abschnitt 1.3.2 eingegangen. (Obwohl diese Use Cases Ergebnis der Untersuchungen dieser Arbeit sind, werden sie nicht unter *Recherchen und Erkenntnisse* aufgeführt, sondern dienen an dieser Stelle zur Erläuterung.)

1.3.1 Use Case 1

Use Case Nr. 1	Background Novel: Wozu diese Software?
<p>Wenn Bob und Nancy während der Anforderungserhebung einen Kunden im Interview zu der in Auftrag gegebenen Software befragt haben, kehren Sie jedes Mal mit einem Stapel Notizen in ihr Büro zurück. Als Profis der Softwareentwicklung haben sie zudem längst ein Bild im Kopf: Dieses zeigt – in der firmenüblichen Modellierungssprache notiert – die Struktur der Prozesse, die der Kunde ihnen grade beschrieben hat. In ihren Notizen finden sich außerdem bereits einige Skizzen dieser Prozesse, die sie gemeinsam mit dem Kunden während des Interviews angefertigt haben. Bei jedem Ihrer Interviews lassen sie zudem ein Tonband mitlaufen – keine Aussage des Kunden kann ihnen verloren gehen.</p> <p>Nachdem sie einige Tage später all ihr Material ausgewertet haben, kehren Nancy und Bob zum Kunden zurück. Im Gepäck haben sie all das, was der Kunde ihnen geschildert hatte, inzwischen ordentlich aufbereitet, mit Diagrammen und Prozessbeschreibungen veranschaulicht. Doch als sie ihrem Interviewpartner ihre Ausarbeitungen vorlegen, schüttelt der nur mit dem Kopf. Offenbar hatte es beim Gespräch einige Missverständnisse gegeben: mehrere Prozesse sind falsch dargestellt, wichtige Abläufe sind ungenügend modelliert, einige Zusammenhänge fehlen sogar ganz.</p> <p>Hätte der Kunde das nicht früher sagen können? Wozu hatten sie ihm beim Interview Skizzen gezeichnet? Nun müssen Bob und Nancy wieder zurück ans Zeichenbrett. Sehr ärgerlich – was das an Zeit kostet! Hoffentlich haben sie beim zweiten Anlauf wenigstens alles richtig verstanden. Und wenn der Kunde endlich mit Ihrer Prozessdarstellung einverstanden ist, müssen sie die Informationen auch noch in die firmeneigene Datenbank übertragen, um die Prozesse an Ihre Kollegen in den Bereichen Entwurf und Test weiterzugeben...</p> <p style="text-align: center;">---</p> <p>Nancy und Bob haben eine neue Software: ein Werkzeug, das ihnen hilft, Missverständnisse zu vermeiden und so einigen Aufwand beim Kundentermin und während der Nachbearbeitung im Büro einzusparen:</p> <p>Direkt im Interview mit dem Kunden kann man damit einfach und schnell die gewonnenen Informationen in die EDV übernehmen. Das Programm macht es leicht, die Hinweise des Kunden zu Prozessschritten zusammenzuführen – und auch gleich mehrere Schritte hintereinander in einem Diagramm darzustellen. Dieses Diagramm ist so einfach zu verstehen, dass auch der Kunde seinen Prozess sofort wieder erkennt – und Fehler in der Modellierung selbst schnell aufspüren kann.</p> <p>Dadurch, dass die verschiedenen Ein- und Ausgabeparameter eines Prozessschritts gespeichert werden, sind nicht nur alle Daten sofort bereit zur innerbetrieblichen Weitergabe, sondern könnten von der Software außerdem nach bisher nicht entdeckten Abhängigkeiten oder logischen Fehlern durchsucht werden.</p> <p>Wenn Nancy und Bob mit mehreren Leuten in einem Betrieb sprechen, können Sie die in den Interviews erstellten Prozesse von der Software vergleichen lassen und finden so Unterschiede und Übereinstimmungen in den Aussagen ihrer Gesprächspartner.</p> <p>Am meisten hilft Ihnen, dass sie, unterstützt von Ihrem Werkzeug, <i>gemeinsam mit den Kunden</i> und <i>noch während des Interviews</i> deren Prozesse verstehen können. Viele Missverständnisse und etlicher damit verbundene Aufwand gehören damit der Vergangenheit an.</p>	

1.3.2 Use Case 2

Use Case Nr. 2	Prozess während Interview modellieren
Umfeld	fremdes Büro, Laptop mit Beamer oder für alle sichtbarer Bildschirm
Systemgrenzen	Beamer zeigt allen Anwesenden den Bildschirm-Inhalt, kein Internet-Zugang oder Netzwerk
Ebene	Hauptanwendung des Systems
Hauptakteur	Prozess-Protokollant
Stakeholder und deren Interessen	Prozess-Protokollant: will möglichst viele im Interview aufkommenden Prozess-Aspekte im Rechner festhalten, möchte dabei fließend arbeiten können, damit er dem Gespräch folgen kann
	Interview-Moderator / weitere Interview-Protokollanten / Interview-Partner: möchten sehen, was der Protokollant notiert und modelliert, um ergänzen und korrigieren zu können, vielleicht kurzzeitig Steuerung übernehmen
	Softwareentwicklungsfirma: Aufwandsreduktion und Kostenersparnis durch kürzere (Feedback-) Zyklen in der Erfassung, Abgleich mit anderen Interviews, gegenseitige Ergänzung der Aussagen
	Auftraggeber/Kunde: Kostenreduktion durch effizientere Arbeitsweise
Voraussetzungen	alle Anwesenden sind über die Funktionsweise der verwendeten Prozessmodellierung (IDEFØ, SADT) informiert, die Vorgehensweise, das Interview mit Unterstützung der Prozessmodellierungs-Software zu unterstützen, ist mit allen abgesprochen, Notation ist einfach genug dargestellt, so dass alle Beteiligten sich problemlos beteiligen können
Garantien	Die gemeinsame Erstellung des Prozessmodells schafft Verständigung und Verständnis über die Prozesse. Wichtige Aspekte sind in der EDV erfasst.
Erfolgsfall	Entwicklung einer vollständigen Prozessmodellierung mit grafischer Darstellung, die keiner weiteren Ergänzung bedarf, vom interviewten Gesprächspartner verstanden und bestätigt.
Auslöser	Interview-Team trifft sich mit Interview-Partner zum Interview
Beschreibung	<ol style="list-style-type: none"> 1. Mitarbeiter des SE-Interview-Teams bauen Laptop und Beamer auf, so dass alle Anwesenden die Beamer-Projektion sehen können 2. Prozess-Protokollant modelliert während des Gesprächs die geschilderten Prozesse mit der Software 3. alle Anwesenden verfolgen die Modellierung während des Interviews 4. alle Anwesenden (insbesondere der Interview-Partner) stimmen mit dem modellierten Prozess überein 5. das Interview wird beendet
Erweiterungen	<ol style="list-style-type: none"> 2a_1. ein Anwesender bemerkt eine Unstimmigkeit und bringt diese zur Diskussion 2a_2. alle Anwesenden besprechen die Anmerkung und weisen den Prozess-Protokollanten an, entsprechende Änderungen an der Modellierung vorzunehmen 2b_1. ein Anwesender bemerkt eine Unstimmigkeit und bittet den Prozess-Protokollanten, ihm die Steuerung des Modellierungsprogramms zu überlassen, um seinen Einwand zu erläutern, indem er mit Hilfe der Software Änderungen an der Darstellung vornimmt 3a. für die Verfeinerung eines Prozessschrittes oder für einen verwandten Prozess wird auf andere Gesprächspartner verwiesen, die in einem Folgeinterview dazu befragt werden können
Technologie	Ein transportabler Drucker ermöglicht das Ausdrucken einer grafischen Darstellung des modellierten Prozesses, wenn dieser "druckreif" dargestellt werden konnte. Die Prozessschritte sind jedenfalls druckbar. Ein Export auf Data-Sticks oder andere Medien ermöglichen den Transfer, übliche Formate (pdf, Bilder) erlauben sofortige Auswertung.

1.4 Gliederung der Arbeit

Diese Arbeit ist wie folgt aufgebaut:

Kapitel 1	Nachdem einleitend die Motivation und Aufgabenstellung erläutert wurden,
Kapitel 2	werden die zu verwendenden Notationen eingeführt.
Kapitel 3.1	Im folgenden Kapitel werden die Recherchen geschildert, die zur Anforderungserhebung unternommen worden sind. Hierbei werden die gewonnen Erkenntnisse zunächst ohne weitere Be- oder Auswertung zusammengestellt. Anschließend werden die Erkenntnisse ausgewertet und Schlussfolgerungen für die Anforderungserhebung gezogen.
Kapitel 3.2	Im zweiten Teil der Recherchen wird die Problematik der gemeinsamen Darstellung von Informations- und Dokumentenflüssen untersucht. Verschiedene Lösungsansätze werden diskutiert und ein Ansatz begründet als Lösung gewählt.
Kapitel 4.1	Im Folgenden werden die Ergebnisse der Anforderungserhebung wiedergeben.
Kapitel 4.1.1	Die gesammelten Anforderungen werden zusammenfassend beschrieben
Kapitel 4.1.2	und zu einem Anforderungsdokument zusammengefügt, das eine entsprechende Software, wie sie in der Aufgabenstellung gefordert ist, detaillierter beschreibt.
Kapitel 4.2	Als ein weiteres Ergebnis wird auf die Untersuchung des Prozessmodells verwiesen.
Kapitel 4.3	Im anschließenden Entwurf folgt eine Zusammenfassung der Entscheidungen, die getroffen und der Konzepte, die entwickelt wurden, um eine Software gemäß der aufgestellten Anforderungen erstellen zu können.
Kapitel 4.4	Schließlich wird auf die erstellte Software eingegangen.
Kapitel 5	Die Arbeit schließt mit einem Ausblick, was an der erstellten Software noch zu tun bleibt.

Zu Beginn der längeren Kapitel und Abschnitte findet sich eine kurze Beschreibung des Aufbaus.

2 Zu verwendende Notationen

2.1 IDEFØ

Die im Rahmen dieser Arbeit zu verwendende Modellierungssprache zur Darstellung von Prozessen wird als „Integration Definition for Function Modeling“, kurz IDEFØ, bezeichnet [2]. Die Grundlagen dieser Notation sind einfach zu verstehen und macht sie daher ideal für die Kommunikation zwischen Fachleuten und Laien.

Prozesse müssen hier nicht zwingend als zusammenhängender Ablauf von Ereignissen dargestellt werden. Die Funktionen, oder *Prozessschritte*, können zunächst einzeln betrachtet werden, wobei jeder Schritt über vier Seiten verfügt, an denen *Daten oder Objekte* entweder „ankommen“ oder den Schritt „verlassen“. Einfach ausgedrückt gibt es *Inputs*, die mittels Steuerungsobjekten (*Controls*) und *Mechanismen* zu *Outputs* verarbeitet werden.

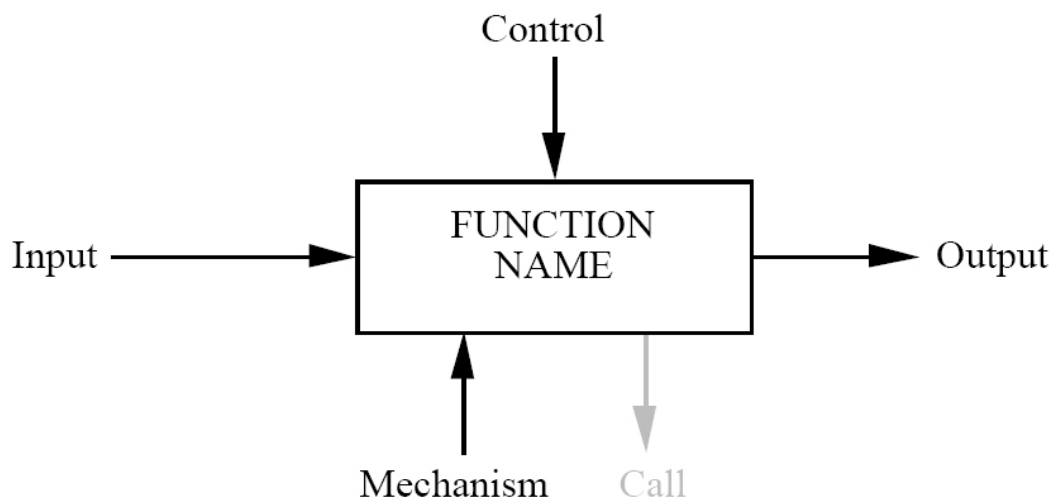


Abbildung 1: Ein Prozessschritt in der IDEFØ-Darstellung

Im weiteren Verlauf wird auf die Form dieser „Daten und Objekte“, von denen in der offiziellen Definition die Rede ist, noch detailliert eingegangen, es soll an dieser Stelle verallgemeinernd angenommen werden, dass es sich um Dokumente handelt. Ein Beispiel für einen solchen Prozessschritt wäre dann etwa:

Input	Die Umsetzung eines <u>Anforderungsdokuments</u>
Mechanismus	durch eine <u>Arbeitsgruppe von Softwareentwicklern</u> ,
Steuerungsdokumente	unter Berücksichtigung von <u>internen Dokumentations-Richtlinien</u> und <u>LID-Auswertungen</u> sowie anhand von bereits erstellten <u>Use Cases</u> ,
Output	die als Ergebnis den <u>Quellcode einer Software</u> hat.

Da sich auch komplexe Prozesse zwar auf diese Weise in einem einzigen Schritt zusammenfassen lassen, dieser dann aber wenig über die genauen Abläufe aussagt, können Prozessschritte in IDEFØ wiederum in mehrere Schritte aufgeteilt werden. Zudem können über Dokumente, die in verschiedenen Schritten verwendet werden, Verbindungen zwischen den Prozessschritten entstehen. Zum Beispiel könnte eine als Output eines Schrittes produzierte Checkliste in einem anderen Prozessschritt als Steuerungsdokument eingesetzt werden.

Um solche Zusammenhänge oder Verfeinerungen abzubilden sieht die IDEFØ-Notation eine hierarchische Darstellung vor, wie sie unten gezeigt ist.

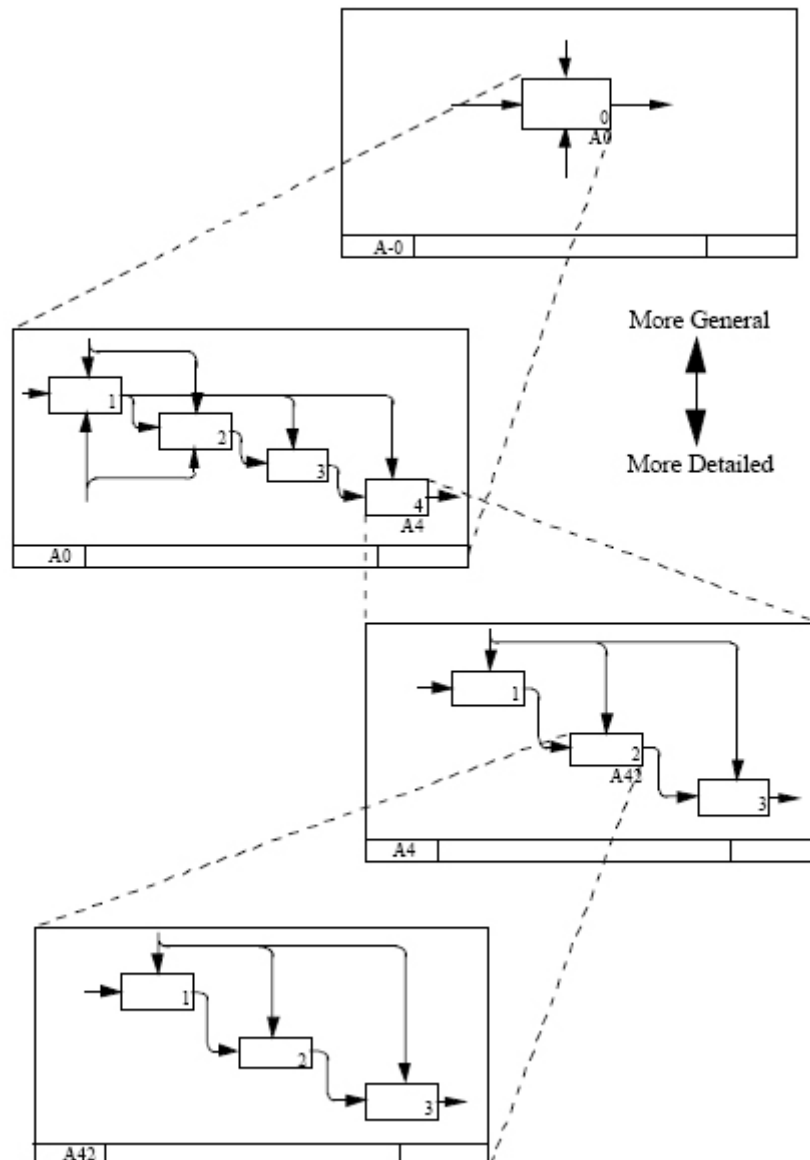


Abbildung 2: Ein Prozess mit mehreren Detaillierungsstufen

Die Regeln der korrekten IDEFØ-Darstellung beschreiben weiterhin Details wie die Form der Pfeile, die Nummerierung der Ebenen etc. und sind sehr komplex und umfassend. Hier sollen jedoch die Grundprinzipien dieser Modellierungssprache ausreichen.

2.2 FLOW

FLOW ist ein Forschungsprojekt des Fachgebiets Software Engineering an der Leibniz Universität Hannover [3, 4], in dem die Bedeutung von *Informationsflüssen* in Softwareprojekten herausgearbeitet wird. Die „Aggregatzustände“ von Informationen werden dabei als fest, flüssig und gasförmig definiert, die mit untenstehenden Symbolen dargestellt werden. Die dazugehörigen Informationsflüsse werden durch die unterschiedlichen Linienarten repräsentiert.



Abbildung 3: Die Aggregatzustände von Informationen in FLOW

Diese Gleichnisse aus der Physik stehen für

fest	wiederholt abrufbare, kopierbare und nicht personengebundene Informationen (meist textlich festgehalten),
flüssig	Informationen, die im Kopf oder privaten Notizen von Personen stecken oder in Gesprächen gewonnen werden und
gasförmig	Gerüchte und Trends sowie allgemeine Überzeugungen, die im Hintergrund der Prozesse bestehen und sich auf deren Ablauf auswirken.

Hinweis:

IDEFØ bezieht sich auf Daten und Objekte, oder, wie oben verallgemeinernd angenommen, zumeist auf Dokumente, die zwischen den Prozessschritten „weitergereicht“ werden. Diesen so genannten *Dokumentenflüssen* stehen die Informationsflüsse gegenüber, auf die sich die Betrachtung der Aggregatzustände in FLOW konzentriert. Wie diese unterschiedlichen Ansätze in einem gemeinsamen Konzept zusammengefasst werden können, wird im Abschnitt 3.2.1: *Problematik Informationsfluss/Dokumentenfluss* behandelt.

3 Recherchen und Erkenntnisse

Um die Anforderungen an eine Prozessmodellierungssoftware benennen zu können, die im Rahmen von Interviews eingesetzt werden soll, habe ich mit verschiedenen Personen gesprochen, die mit Interviews oder der Modellierung von Prozessen zu tun haben. Manche meiner Gesprächspartner haben auch Erfahrungen mit Interviews eben zum Zwecke der Anforderungserhebung, und konnten mir daher genaue Hinweise geben, welche Aspekte sie für wichtig halten.

3.1 Zusammenfassung der Recherche und Einzelergebnisse

Es wird zunächst die Untersuchung von Interviews im Allgemeinen durchgeführt und kurz zusammenfassend ausgewertet (Abschnitt 3.1.1).

Es folgt die Wiedergabe der speziell zu Interviews in der Anforderungserhebung durchgeführten Untersuchungen (Abschnitt 3.1.2).

Abschließend wird eine Gesamtauswertung vorgenommen und eine Extraktion der Anforderungen durchgeführt, die sich hieraus für eine Prozessmodellierungssoftware ergeben (Abschnitte 3.1.3 und 3.1.4).

3.1.1 Untersuchung von Interviews im Allgemeinen

Zunächst soll die allgemeine Situation „Interview“ beleuchtet werden. Welche Bedingungen liegen für die Gesprächspartner vor, gibt es einen üblichen Verlauf oder häufig vorkommende Situationen? Es zeigt sich, dass beide Gesprächsseiten – wenn auch im Rahmen des festgelegten Themas – flexibel den Gesprächsverlauf formen. Eher das Fehlen von „Standardsituationen“ ist charakteristisch für die Dynamik von Interviews, nicht eine bestimmte immer wiederkehrende Routine.

Gespräch mit Ulrich Milde

Ich sprach mit Herrn Ulrich Milde, der als Redaktionsleiter bei der Leipziger Volkszeitung Interviews mit Spitzenpolitikern wie Angela Merkel und Helmut Kohl geführt hat und über einen langjährigen Erfahrungsschatz bezüglich dieser Art der Gesprächsleitung verfügt.

Mein Interesse galt dabei Herrn Mildes Vorbereitung eines solchen Gesprächs, dem üblichen Verlauf und seinem Umgang mit unvorhergesehenen Wendungen, die sich ergeben können.

Es stellte sich zunächst heraus, dass nach Erfahrung des Journalisten gute Kenntnis des behandelten Stoffes unbedingt nötig ist, um die Möglichkeit zu haben, flexibel zu reagieren. Es sei sehr viel Improvisation gefragt, und die Vorbereitung für den Interviewenden sei auch eher in Form eines stichwortartigen „roten Fadens“ sinnvoll, anstatt mittels ausformulierter Fragen. Dies liege daran, dass im Gespräch ohnehin nur grob die Grenzen festgelegt werden könnten, in denen das Gegenüber jedoch die Möglichkeit hat, frei zu reden. Eine gewisse Abweichung vom Thema und Abschweifungen des Gesprächspartners seien tatsächlich manchmal sogar erwünscht, um interessante Hintergründe und vermeintliche Nebensächlichkeiten aufzudecken, die wichtige Informationen enthalten können. Es sei zu erwarten, dass der Gesprächspartner in seiner Rede gedankliche Sprünge macht; ihn durch vorbereitete Fragen „gängeln“ zu wollen könnte ihn verärgern und das Verhältnis zu ihm stören. Zudem sei es zur Einschätzung des Interviewten wichtig, ihn sich frei entfalten zu lassen, um ein möglichst authentisches Bild zu erhalten.

Erste Auswertung

Obwohl ein Interview, das als „Befragung ... mit dem Ziel, ... Sachverhalte zu ermitteln“¹ definiert wird, einen starren Ablauf entlang einer Liste von Fragen vermuten lässt, stellt sich hier ein anderes Bild dar. Es wird ein weitgehend freies Gespräch geschildert, dessen Verlauf nicht vorhergesagt werden kann. Der Versuch einer *zu* detaillierten Vorbereitung (z.B. in Form eines strikten Fragenkatalogs) wird scheitern und den Interviewenden wie auch den Interviewten möglicherweise eher behindern als den effizienten Ablauf fördern.

Interessant ist der Hinweis auf die Nebensächlichkeiten, die durch die freie Gesprächsführung „ans Licht“ kommen. Bezogen auf Interviews zur Anforderungserhebung kann man hier übertragen, dass Vorurteile und falsche Vorstellungen des Interviewenden die Aussage des Gegenübers beeinflussen, da die Fragen unter dem Eindruck eigener Annahmen zu eng formuliert werden. Eine Prozessbeschreibung könnte so durch zu restriktive Fragestellung verfälscht werden, während eine „freie“ Befragung den Interviewten dazu bringen kann, Aspekte anzusprechen, die dem Fragesteller nicht bewusst waren und somit wertvolle (nämlich neue) Informationen darstellen.

3.1.2 Untersuchung von Interviews in der Anforderungserhebung

Weitere Gespräche wurden durchgeführt, um festzustellen, inwiefern sich speziell zum Zweck der Anforderungserhebung durchgeführte Interviews von „normalen“ Interviews unterscheiden, welche Besonderheiten hier zu beachten sind und ob es Gründe gegen oder für die Benutzung einer Prozessmodellierungssoftware während solcher Gespräche gibt. Zusammenfassend ist bei allen Befragten die Bedienbarkeit unter den hektischen Bedingungen während des Interviews ein zentraler Punkt.

Ich habe zu diesem Thema mit Personen gesprochen, die Erfahrungen mit Interviews haben, die im Rahmen von Anforderungserhebungen durchgeführt wurden. Ich sprach mit

- Herrn Andreas Röpke, Projektleiter und Software-Qualitätsexperte bei IBM Deutschland,
- Frau Thao Nguyen vom Institut für Praktische Informatik der Leibniz Hannover, sowie
- den Teilnehmern der Projektarbeit Requirements Engineering im Sommersemester 2006.

Im Folgenden schildere ich die einzelnen Gespräche und Aussagen. Anschließend folgt eine Zusammenfassung aller gewonnenen Informationen.

Gespräch mit Andreas Röpke

Herrn Röpke habe ich speziell in Bezug auf die Bedienbarkeit einer Software befragt und ihm Screenshots meines GUI-Mockups vorgelegt. Ich wollte wissen, wie sich nach seinen Erfahrungen ein solches Werkzeug in reale Gesprächssituationen einbinden lässt.

Meine im Mockup dargestellten Versuche, eine leicht verständliche Oberfläche zu präsentieren, unterstützte Herr Röpke als „Schritt in die richtige Richtung“. Er wies in diesem Zusammenhang darauf hin, dass nicht nur ein protokollierender Benutzer auf Seiten der Softwareentwickler eine solche Modellierungssoftware verstehen müsse,

¹ <http://de.wikipedia.org/wiki/Interview>

sondern dass der Bildschirminhalt während des Interviews mittels eines Beamers projiziert werden sollte, um alle Anwesenden in die Prozessmodellierung mit Hilfe dieses Werkzeuges einzubeziehen. Eine „stille Post“, bei der Informationen verloren gehen oder verfälscht werden können, sollte vermieden werden, statt dessen soll die gemeinsame Erstellung zu Austausch und Diskussion anregen. Wichtig sei hierfür natürlich auch, dass die Gesprächspartner vorher über den Zweck der Software und über die zu verwendende Notation aufgeklärt werden, und dass diese gemeinsame Arbeitsweise auch akzeptiert werde. Herr Röpke erinnerte weiterhin an die Mindmap-Methode, um in einem nicht strukturiert verlaufenden Gespräch Stichworte zu sammeln und anschließend zu ordnen und zu hierarchisieren. Des weiteren äußerte er die Idee, dass SOLL und IST-Zustände durch das Hin- und Herschalten zwischen zwei Prozessmodellierungen verglichen werden könnten.

Gespräch mit Thao Nguyen

Als Grundlage meiner Fragen an Frau Nguyen habe ich ebenfalls meinen Mockup zum Oberflächenentwurf vorgestellt. Da die Mitarbeiterin des Institutes für Praktische Informatik im Rahmen ihrer Tätigkeit Prozessmodellierungen anhand von in Interviews gewonnenen Informationen durchführt, konnte ich mich zudem direkt nach ihren Erfahrungen bei dieser Tätigkeit erkundigen. Ich interessierte mich für ihre Einschätzung, welche Auswirkungen die Verwendung einer Modellierungssoftware auf ihre Vorgehensweise bei Interviews haben würde.

Die von mir probeweise zusammengestellte Benutzeroberfläche wirkte auf Frau Nguyen sehr einschränkend. Sie erzählte mir, dass sie in Ihren Interviews am liebsten einen Stift und ein weißes Blatt Papier als Arbeitsmittel verwendet, da hiermit die freiesten Gestaltungsmöglichkeiten für Notizen und Skizzen zur Verfügung stehen. Um die im Gespräch ungeordnet auf Sie einströmenden Informationen zunächst festzuhalten und in eine grobe Ordnung zu bringen, zieht sie diese Freiheit eines leeren Blattes einem herkömmlichen Modellierungstool vor.

Derartige Software, wie beispielsweise Microsoft Visio oder das Open Source-Tool Dia, unterstützt meist eher die Umsetzung einer fertigen Vorlage, als die Erschaffung eines neuen Modells. Das heißt, es ist bei der Benutzung von Vorteil, eine gewisse Ordnung und Struktur von vorne herein zu kennen, damit man die richtigen Elemente aus der Werkzeugleiste auswählen kann, um die Darstellung daraus zusammenzufügen. Eine Umwandlung eines einmal gewählten Elementtyps ist meist nicht möglich. Wenn während der Erstellung auffällt, dass die Modellierung nicht mit dem Darzustellenden übereinstimmt, ist eine Überarbeitung so meist sehr aufwändig. Solche Tools wären in der Interview-Situation, in der schnelle und häufige Änderungen des entstehenden Prozessmodells sehr wahrscheinlich notwendig sind, nicht zu gebrauchen.

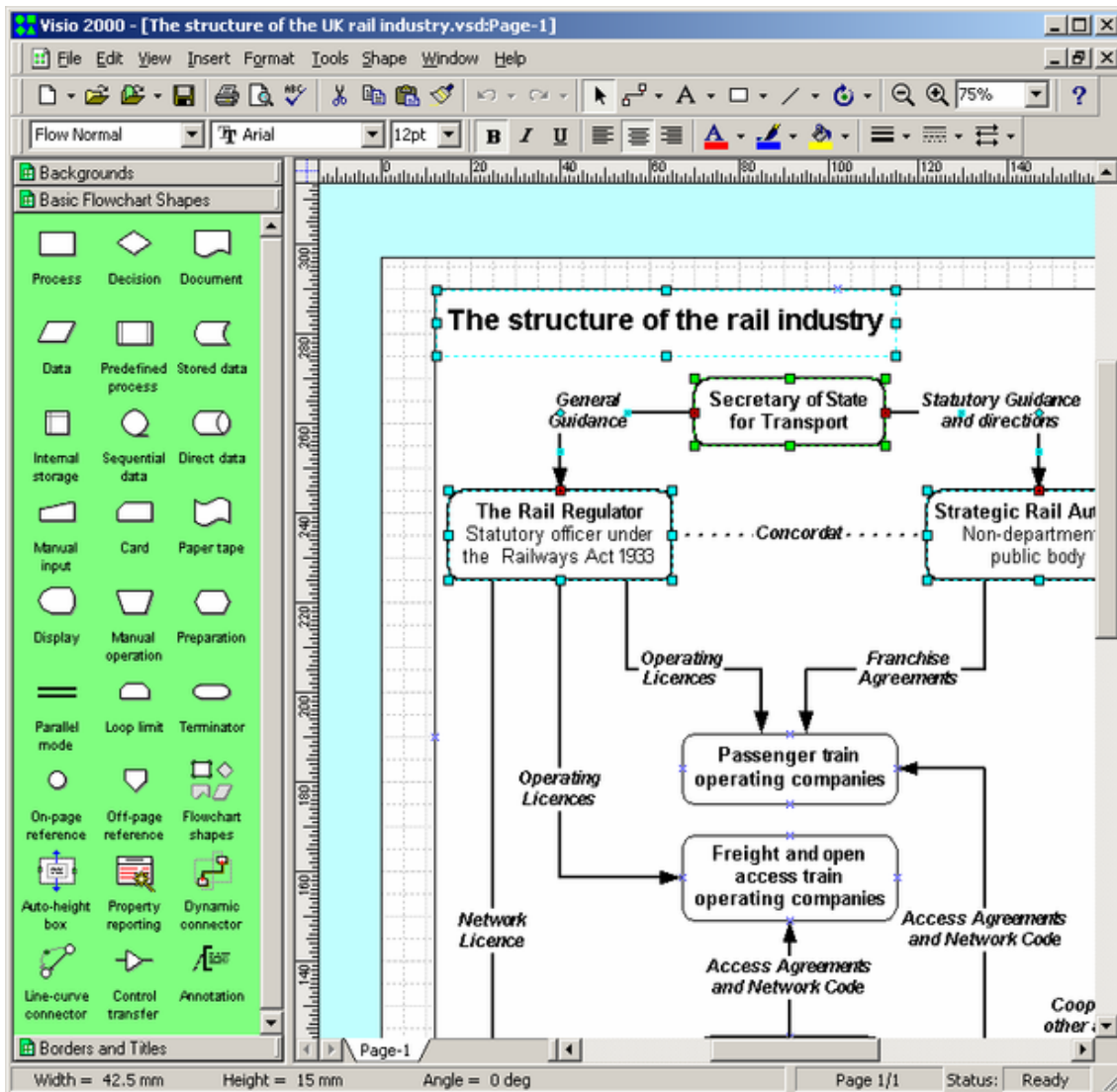


Abbildung 4: komplexe Bedienoberfläche der Visualisierungssoftware Microsoft Visio 2000

Zudem brachte Frau Nguyen zur Sprache, dass ihre übliche Vorgehensweise ein „Top-Down“-Ansatz ist: sie versucht während des Gesprächs zunächst die grobe Struktur der Prozesse und deren Zusammenhänge zu erfassen und geht erst hinterher auf die Details wie verwendete Dokumente oder Informationsflüsse ein.

Umfrage bei den Teilnehmern der Projektarbeit Requirements Engineering

An die Studenten in der Projektarbeit Requirements Engineering habe ich mich zunächst vor deren Projekt mittels Fragebögen gewandt, und noch einmal anschließend in persönlichen Gesprächen. Die Teilnehmer sollten in diesem Projekt Interviews mit Vertretern verschiedener Einrichtungen der Universität führen, um Anforderungen für dort benötigte Software zu erheben.

Ich interessierte mich zunächst unter anderem für ihre Einschätzung, ob eine Prozessmodellierungssoftware sie bei ihrer Aufgabe während der Interviews unterstützen könnte und welche Gründe dafür bzw. dagegen sprechen würden. Nach Abschluss der Projektarbeit habe ich mich nach den gemachten Erfahrungen erkundigt und darauf aufbauend direkt nach Eigenschaften gefragt, die die Teilnehmer sich für eine unterstützende Software wünschen würden, wenn sie die Interviews des Projekts erneut durchführen sollten.

Ebenso wie Frau Nguyen empfanden sämtliche Teilnehmer zunächst die gängige Visualisierungssoftware als zu unhandlich und konnten sich nicht vorstellen, diese während eines Interviews gewinnbringend einzusetzen. Als ein weiteres Gegenargument wurde die benötigte Technik als „ein fremdes Medium“ genannt, das „zusätzlichen Abstand zwischen die Parteien bringt“. Bleistift und Papier zur Darstellung von Skizzen würden vermutlich leichter akzeptiert als zusätzliche Technologie wie Tablet-PCs und eine Software, die „einschüchternd professionelle“ Diagramme erstellt.

Allgemein wurden einer Software jedoch Chancen eingeräumt, die Modellierungsarbeit während Interviews sinnvoll unterstützen zu können, sofern sie sowohl Spielraum lässt, aber auch mitdenkt und situationsbedingt „clevere Shortcuts“ anbietet.

Nach der Projektarbeit ergaben die Gruppen-Befragungen folgende Ergebnisse:

- Keine Gruppe hatte während der Interviews eine Modellierungs-Software benutzt.
- Prozessmodellierungen wurden im Allgemeinen während der Interviews nicht benötigt, da die Beschreibungen der Gesprächspartner teilweise noch sehr abstrakt und weit von einer modellierbaren Struktur entfernt waren.
- Überwiegend waren die Teilnehmer der Ansicht, dass vor allem beim ersten Treffen und Interview mit Ihren „Kunden“ jegliche Software im Gespräch störend gewesen wäre, da zunächst ein Verhältnis zum Gegenüber aufgebaut und eine gemeinsame Richtung gefunden werden müsse.
- Eine akustische Aufzeichnung der Gespräche wurde von allen Gruppen empfohlen, da hiermit hastige Notizen beim Anhören der Aufnahmen wieder in den Kontext gebracht werden können. Da aber schon normale Notizen in einem schnellen, nicht zusammenhängenden Gespräch Schwierigkeiten machten, waren viele Teilnehmer des Projektes davon überzeugt, dass eine Software unter den selben Bedingungen noch schlechter geeignet wäre, Daten zu sammeln, geschweige denn diese noch „vor Ort“ strukturiert zusammenzufassen.

3.1.3 Gesamtauswertung der Untersuchungen

Im Folgenden werden die zu Interviews im Allgemeinen und zu Interviews speziell in der Anforderungserhebung gesammelten Informationen gemeinsam ausgewertet. Es werden abschließend im Abschnitt *Extraktion von Anforderungen für die Modellierungssoftware* die wichtigsten Erkenntnisse über Anforderungen für eine Prozessmodellierungssoftware, die in Interviews unterstützend eingesetzt werden soll, zusammengefasst.

Alle während der Recherche befragten Personen sind sich darin einig, dass ein Interview kein starrer, sondern ein sehr lebendiger Vorgang ist. Die Gesprächspartner müssen aktiv aufeinander eingehen und sich auf aufkommende Themen und Fragen einlassen, um ein konstruktives Ergebnis zu ermöglichen.

In der speziellen Situation eines Interviews zur Anforderungserhebung bieten sich dem Interviewten somit viele Möglichkeiten, vom Thema abzuweichen oder seine Darstellung, wenn auch vielleicht unbeabsichtigt, zu verkomplizieren: So muss die Reihenfolge seiner Schilderungen nicht mit dem Ablauf der Prozesse übereinstimmen, die er grade erläutert, oder eine Ausführung nicht zwangsläufig sinnvoll in den Zusammenhang des Gesprächs passen.

Wie von Herrn Ulrich Milde berichtet¹, nutzt der Reporter grade diesen lebendigen Vorgang, um an interessante Informationen heranzukommen, und weicht auch gerne von seinen „starrten“ vorbereiteten Fragen ab, um dem Gesprächsverlauf dynamisch folgen zu können. Auch Frau Thao Nguyen, die über Interviews zu Zwecken der Anforderungserhebung berichtete², schätzt die Möglichkeit, auf den individuellen Verlauf eines Interviews eingehen zu können. Die dort erwähnten Freiheiten eines „weißen Blattes Papiers“, auf dem alle sich ergebenden Wendungen des Gesprächs „in Echtzeit“ notiert, skizziert, verändert und verworfen werden können, machen eben solch ein schlichtes „Schmierblatt“ zu einem wertvollen Werkzeug.

Viele der während der Recherche befragten Personen haben daher ihre Sorge zum Ausdruck gebracht, von einem Tool während eines Interviews in Ihrem Reaktionsfreiraum eingeschränkt zu werden.

Allerdings ist noch ein anderer Aspekt in diesem Zusammenhang wichtig: Anders als bei einem zwanglosen Gespräch muss der Leiter eines Interviews nämlich in der Lage sein, die Richtung, die die Unterhaltung einschlägt, unter Kontrolle zu behalten. Dies ist u. a. notwendig, um die gegebenenfalls komplexen Sachverhalte, die im Interview verstanden werden sollen, richtig aufzunehmen. Ein weißes Blatt Papier bietet keine Anleitung, keine Richtlinien, an die man sich halten kann, um während des Gesprächs nicht abzuschweifen und das Ziel (hier: Prozessmodellierung) nicht aus den Augen zu verlieren. Auch der Reporter greift, bei aller Improvisationsfreude, im Zweifelsfalle auf seine Stichworte zurück, damit er seinen Gesprächspartner an den „roten Faden“ seines Interviews binden kann.

¹ siehe Abschnitt 3.1.1: *Gespräch mit Ulrich Milde*

² siehe Abschnitt 3.1.2: *Gespräch mit Thao Nguyen*

Man könnte also zusammenfassen:

Gesucht ist ein Werkzeug, so einfach zu bedienen und so frei in den Gestaltungsmöglichkeiten wie ein weißes Blatt Papier, welches Anleitung und Hilfe gibt, das gesteckte Ziel im Verlauf eines dynamisch verlaufenden Gesprächs nicht aus den Augen zu verlieren.

Die hierin enthaltenen Anforderungen „leichte Bedienbarkeit“, „freie Entfaltungsmöglichkeiten in der Darstellung“ und „Benutzerführung“ fallen alle unter den Qualitätsaspekt der *Usability* und scheinen teilweise komplementär und unvereinbar. Abstriche in der einen oder anderen Hinsicht sind unvermeidbar und werden sogar teilweise schon durch die Aufgabenstellung dieser Arbeit vorgegeben:

1. Zur Darstellung von Prozessabläufen sind die Modellierungen des IDEFØ-Standards zu verwenden, was die Freiheit der Darstellung einschränkt.
2. Eine bessere Paint-Anwendung (die computerisierte Umsetzung eines Zeichenblattes) zur Verfügung zu stellen, stellt für den Wunsch nach „grenzenlosen“ Möglichkeiten der Datenerfassung keine Lösung dar. Selbst wenn die Erfassung der Daten durch formloses Zeichnen von Prozessen auf einem Grafiktablett oder Tablet-PC geschehen würde, müsste erst eine Auswertung der so gezeichneten Prozess-Abläufe durchgeführt werden. Hierfür wären Bilderkennungsalgorithmen für die automatisierte Überführung in eine speicherbare Datenstruktur in der EDV nötig. Dies kann im Rahmen dieser Arbeit nicht geleistet werden. Zudem ist anzunehmen, dass für diese Technik präzise einzuhaltende Formalismen (um der Software die Erkennung der gezeichneten Elemente zu ermöglichen) die Arbeit eher erschweren würden.
3. Das Ziel, einen strukturierten Prozessgraphen zu erstellen, steht – selbst wenn die zur Darstellung zu verwendende Modellierungssprache nicht vorgegeben wäre – im Widerspruch zu der Erkenntnis, dass ein Interview dynamisch abläuft und häufig inhaltliche Sprünge enthalten kann.

Zum letzten Punkt kann nun aber gerade der Aspekt der offenbar „einschränkenden“ Benutzerführung einer Software Boden gut machen: Denn der gegebenenfalls unstrukturierte Ablauf eines Interviews kann durch entsprechende Führungsmechanismen in einer solchen Software „geordnet“ und überschaubar gemacht werden. Einen Ansatz bietet hier beispielsweise das von Herrn Röpke angesprochene „Mindmap“-Verfahren¹, hier mit der Bezeichnung „Sandbox“: Der Benutzer soll dabei die Möglichkeit haben, im Gespräch gewonnene Erkenntnisse, die sich noch nicht in einen Zusammenhang einordnen lassen, in einem Sammelcontainer abzulegen und dort aufzubewahren, bis er sie in einen Gesamtkontext bringen kann. So sollen „Gesprächsfetzen“, Stichworte und abgebrochene Verzweigungen des Interviews zwischengespeichert und später wieder aufgenommen werden können, um nicht im Durcheinander eines möglicherweise unstrukturierten Gesprächs verloren zu gehen. Eine nahe liegende Umsetzung findet sich in der Anwendung Drag&Drop-Verfahren moderner Betriebssysteme; wie reale Objekte könnten die Zwischenergebnisse „bei Seite“ gelegt und später wieder hervorgeholt werden (dieser Gedanke wird im Kapitel 4.3.1: *Konzept: „Sandbox“* detailliert aufgegriffen).

¹ siehe Abschnitt 3.1.2: *Gespräch mit Andreas Röpke*

3.1.4 Extraktion von Anforderungen für die Modellierungssoftware

Es ergeben sich bereits in Abschnitt 3.1.1 nach der Untersuchung von Interviews im Allgemeinen Anforderungen an eine Software, die während Interviews zur Prozessmodellierung verwendet werden soll:

- Ein solches Werkzeug darf durch seine Funktionen nicht eine bestimmte Sicht auf einen Prozess vorgeben, die die tatsächliche Sachlage verschleiert.
- Es muss die Möglichkeit bieten, flexibel auf Änderungen der Darstellung zu reagieren,
- nicht in den Zusammenhang passende Aussagen müssen unabhängig protokollierbar sein, gespeichert und später sortiert werden können,
- und zwar ohne den Gesprächspartner in seiner Rede zu unterbrechen.

Es wird auch in den in Abschnitt 3.1.2 folgenden Untersuchungen deutlich: Usability ist die im Vordergrund stehende Qualitätsanforderung.

Das bedeutet übersichtliche Oberflächen, kurze Wege, intuitive Benutzersteuerelemente und „clevere Shortcuts“, wie es ein Teilnehmer des RE-Projekts formulierte. Ein entsprechender Entwurf kann es ermöglichen, dass eine Software unter den hektischen Bedingungen eines Interviews hilft, möglichst viele Informationen zu sammeln und den Überblick zu behalten, indem sie Verbindungen findet, logische Ergänzungen anbietet und zur Strukturierung anleitet. Gleichzeitig muss darauf geachtet werden, dass der Benutzer auf rasche Gesprächswendungen reagieren kann und flexibel eine Struktur, die falsch erklärt oder verstanden wurde, neu aufbauen kann.

Auch die vorgegebene Prozessmodellierungssprache und ihre Anwendung spielt eine wichtige Rolle. Die Einfachheit von IDEFØ muss richtig eingesetzt werden, um den Auftraggeber, der wohl in den seltensten Fällen Experte für Prozessmodellierung ist, dazu zu bringen, seine Erklärungen der Modellierung anzupassen und gemeinsam mit den Entwicklern in der Software seine Prozesse darzustellen. So kann die geschickte Einbindung der Software in die Durchführung von Interviews ein großes Problem bei solchen Begegnungen, den „Gulf of Understanding“, überwinden helfen. Die Anforderungen gehen also über die Software „an sich“ hinaus und beziehen sich auch auf die richtige Benutzung – diese durch die Gestaltung des Programms aber zu forcieren ist wieder eine Aufgabe des Entwicklers, der die Software gestaltet.

(Auf diese Anforderungen wird in 4.1.1.2: *Weitere Anforderungen durch die Integration der Software in den Ablauf von Interviews* weiter eingegangen. Ideen, wie diese Anforderungen in einer Software umgesetzt werden können, erläutere ich im Kapitel 4.3: *Entwurf*.)

3.2 Erweiterung der Aufgabenstellung im Zwischenvortrag

Im Zwischenvortrag zu dieser Bachelorarbeit wurden die bisherigen Ergebnisse diskutiert. Hier kam zur Sprache, dass die von der Aufgabenstellung vorgegebene Modellierungssprache IDEFØ sich auf Dokumentenflüsse bezieht – im verwendeten Definitionspapier [2] ist von „Daten und Objekten“ die Rede. Dementsprechend galt zu diesem Zeitpunkt die Darstellung im Konzept und in der bereits erstellten Software der Weiterreichung von „Datenobjekten“, also beispielsweise Dokumenten. Es wurde festgestellt, dass eine Verbindung dieser Objekte mit den Aggregatzuständen aus FLOW, die sich eigentlich auf Informationsflüsse beziehen, ohne weiteres nicht sinnvoll ist. Entsprechend wurde die Aufgabenstellung dieser Bachelorarbeit erweitert, dieses Problem zu untersuchen und mehrere Lösungsansätze zu erarbeiten. Im Folgenden wird hierauf eingegangen.

3.2.1 Problematik Informationsfluss/Dokumentenfluss

Die mit FLOW als fest, flüssig oder gasförmig vorliegend bezeichneten Informationen können ihren „Träger“ wechseln, zum Beispiel kann eine Fachkraft, die über bestimmte Informationen verfügt, diese in einem Dokument zusammenstellen. Somit wäre ein Übergang von „flüssig“ nach „fest“ vollzogen worden. In einem Informationsfluss kann das Medium also wechseln. Dokumentenflüsse, d.h. die Weitergabe von Dokumenten, sind diesbezüglich starr, da ein Dokument grade das Medium darstellt. Die in Dokumenten enthaltenen Informationen können hier nur durch die Bezeichnungen der Dokumente unterschieden werden.

Ein Beispiel: ein Formular, das in einem Prozessschritt verwendet wird, wird dabei sicherlich bearbeitet, indem Daten eingetragen werden. Das anfänglich leere Formblatt, welches als „Input“ angegeben ist, wird mit Informationen gefüllt und stellt in dieser Form den „Output“ dieses Schrittes dar.

Es gilt also, Informationsflüsse mit einer Notation deutlich zu machen, die für Dokumentenflüsse ausgelegt ist.

3.2.2 Lösungsansätze

Für das oben geschilderte Problem lege ich verschiedene Lösungsansätze vor und diskutiere deren Vor- und Nachteile. Abschließend begründe ich meine Entscheidung für den vierten Ansatz, der in einer Software am sinnvollsten zum Einsatz kommen kann.

3.2.2.1 Textliche Unterscheidung

Zum oben genannten Beispiel wäre es möglich, das Dokument, das in einem Prozessschritt verwendet wird, anfänglich als „leeres Formular B3“ zu bezeichnen und als Input anzugeben. Dieses wird durch die Funktion des Prozessschrittes in ein „Formular B3 mit ausgefüllter Kopfzeile“ umgewandelt, das den Output dieses Schrittes darstellt.

Bei dieser Lösung wäre der Informationsgewinn dem Dokument deutlich anzusehen. Eine Unterscheidung ist auch mit einer Notation möglich, die vollständig auf die Darstellung des Dokumentenflusses ausgelegt ist.

Problematisch ist hierbei allerdings, dass nun anscheinend zwei Dokumente existieren. Letztendlich handelt es sich aber um das selbe Formular „B3“, auf dem lediglich einige Einträge vorgenommen wurden. Das „leere Formular B3“ existiert nicht weiter, sondern wurde in das „Formular B3 mit ausgefüllter Kopfzeile“ umgewandelt. Es er-

geben sich also Situationen in der Darstellung, die nicht mit der abzubildenden Wirklichkeit übereinstimmen. Zudem müsste nach dieser Vorgehensweise für jede in einem Prozessschritt vorgenommene neue Eintragung ein neues Dokument benannt werden (beispielsweise „Formular B3 mit ausgefüllter Kopfzeile und Prüfsignatur“ usw.) – was nicht nur die Anzahl der scheinbar existierenden Dokumente erhöht, sondern auch durch die zunehmende Länge der Dokumentnamen unübersichtlich und schwer zu warten ist (ändert sich die Bezeichnung des Formulars in „B4“ müssen alle davon abgeleiteten Bezeichnungen aktualisiert werden).

3.2.2.2 Nummerierung/Versionisierung

Um die Wartbarkeit und die Übersichtlichkeit zu verbessern, könnte anstelle langer Textbezeichnungen eine numerische Hierarchisierung oder Versionisierung der Dokumente vorgenommen werden. Das Vorgehen ist hierbei ähnlich zu dem in 3.2.2.1 beschriebenen, nur dass die unterschiedlichen enthaltenen Informationen, also Dokument-Versionen, durch Buchstaben-/Zahlenkombinationen dargestellt werden. Das leere „Formular B3“ würde dann zu „Formular B3 (v1)“, nachdem erste Änderungen vorgenommen wurden, zu „Formular B3 (v1.1)“, wenn diese Änderungen modifiziert wurden, oder zu „Formular B3 (v2)“, wenn wiederum andere Informationen hinzugefügt wurden. Eine automatisch mitgeführte Versionisierungsliste könnte die Bedeutungen der Nummerierung erläutern und Aktualisierungsvorgänge verwalten.

Dieser Ansatz löst nicht das Problem, dass zur Darstellung zusätzliche, nicht existente Dokumente geschaffen werden. Wiederum erschwert das Nichtübereinstimmen von Darstellung und Wirklichkeit die Übersicht und Verständlichkeit. Zudem sind die als Vereinfachung gedachten abkürzenden Versionsnummern nur schwer „human readable“. Wie man bereits an den obigen Beispielen sieht wird durch ihre Verwendung die Darstellung nicht wirklich vereinfacht.

3.2.2.3 Container 1: Informationen mit Verweis auf Dokumente

Eine weitere Darstellungsmöglichkeit ist ein Containermodell, bei dem Informationen, die ja im Prozess „fließen“ sollen, die Dokumente oder Objekte, in denen sie sich befinden, als Attribute mit sich führen. So wäre es möglich, die Idee der Aggregatzustände aus FLOW aufzugreifen. Beispielsweise könnte das „Wissen um Ablauf XY“ zunächst als „flüssige“ Information im „Kopf von Fachkraft Z“ existieren, der dann in einem Prozessschritt „Leitfaden erstellen“ durch die Erstellung eines Papiers „Leitfaden zu Ablauf XY“ eine Transformation des Aggregatzustandes vornimmt. Das „Wissen um den Ablauf XY“ hätte vor dem Prozessschritt das Attribut „Kopf von Fachkraft Z“, und wäre anschließend mit dem „Leitfaden zu Ablauf XY“ verknüpft. Hier würde die Festlegung auf Dokumente wegfallen.

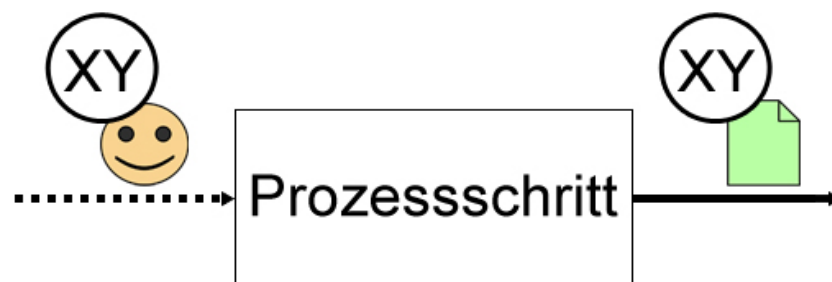


Abbildung 5: Transformation des Aggregatzustandes einer Information in Containermodell 1

Dieser Ansatz sorgt schon für eine wesentlich deutlichere Sichtweise auf die Darstellung, die leichter nachzuvollziehen ist. Allerdings ist die Vorstellung, dass Informationen (als nicht greifbare Abstraktionen) weitergegeben werden, die die Darstellungsform oder den Träger wechseln, nicht alltäglich, und ist in einigen Fällen auch nur schwierig darstellbar. Zwar existiert nun tatsächlich nur ein Formular, wie auch in der abzubildenden Wirklichkeit, auf das verwiesen wird. Bei dem Beispiel des weitergereichten Formulars wäre es allerdings mühsam, alle darin vorhandenen Informationen immer mit dem Verweis auf das Formular weiterzureichen: Wir wollen annehmen, dass beispielsweise 5 Schritte durchlaufen werden, in denen immer neue Informationen durch bearbeitende Personen in das Formular eingetragen werden. Die Transformation der „flüssigen“ Informationen in den Köpfen der Bearbeiter in die „feste“ Form des Formulareintrags ist, wie oben demonstriert, ausgezeichnet darstellbar. Allerdings sammeln sich die Informationen immer weiter an, so dass von Schritt 4 zu 5 bereits vier Informationsflüsse gezeichnet werden müssten, da die Informationen der ersten Schritte ja ebenfalls weitergereicht werden, diese mit dem Vorgang in Schritt 4 aber eigentlich gar nichts zu tun haben. Dies soll in der folgenden Skizze verdeutlicht werden:

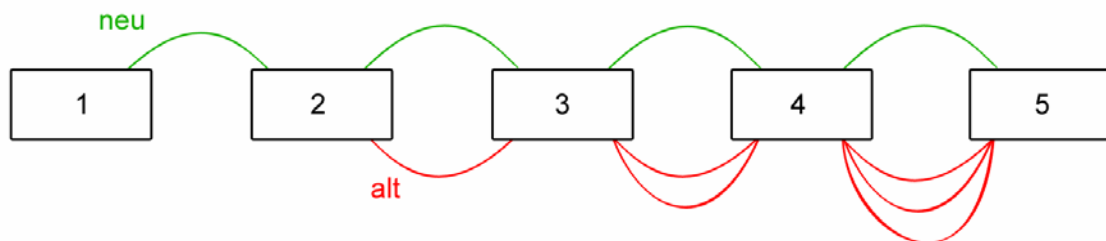


Abbildung 6: Ansammlung einer großen Menge an Informations-Objekten

3.2.2.4 Container 2: Dokumente als Container für Informationen

Die Idee eines Containers beibehaltend ist es auch möglich, die Informationen an die Dokumente anzuhängen, die weitergegeben werden. Dieser Ansatz deckt sich mit den meisten Vorgängen in der realen Welt, wie zum Beispiel in dem Fall des weitergegebenen Formulars, das immer um weitere Informationen ergänzt wird. Ein beispielhafter Vorgang lässt sich wie folgt grafisch darstellen:

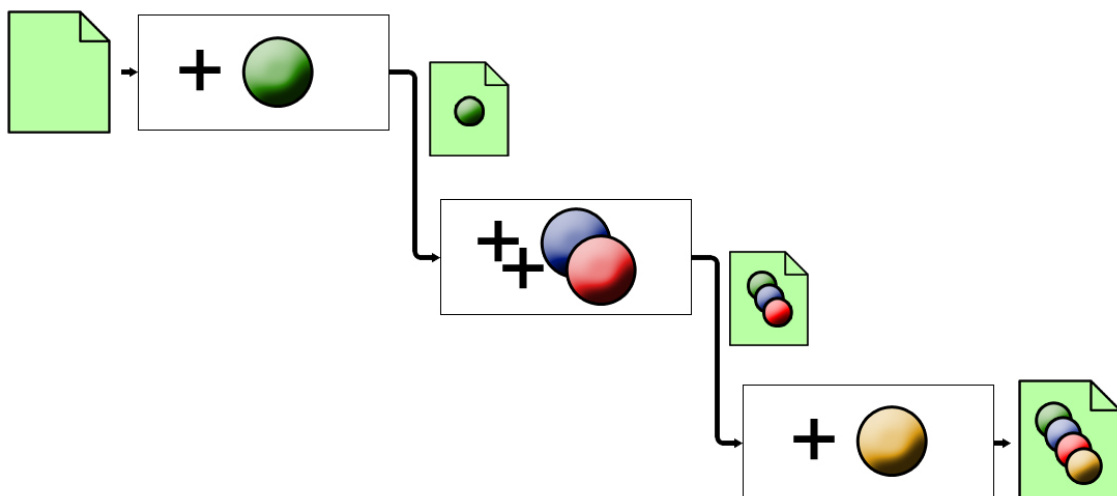


Abbildung 7: Weitergabe eines sich im Prozess mit Informationen füllenden Dokuments

Wie hier zu sehen bleibt das Dokument als Container immer erhalten, es wird von Schritt zu Schritt weitergereicht. Dabei können Informationen, hier durch die farbigen

Kugeln dargestellt, in den Prozessschritten diesem Dokument angehängt werden. Lässt man die Informationen nicht einfach in den Prozessschritten „entstehen“ (was durchaus der Realität entsprechen kann, wenn ein Schritt beispielsweise darin besteht, einen Sachverhalt zu erforschen), und zeichnet man außenstehende Quellen für Informationen ein, wird klar, dass auch eine Transformation von einem Aggregatzustand zum andern auf diese Weise darstellbar ist:

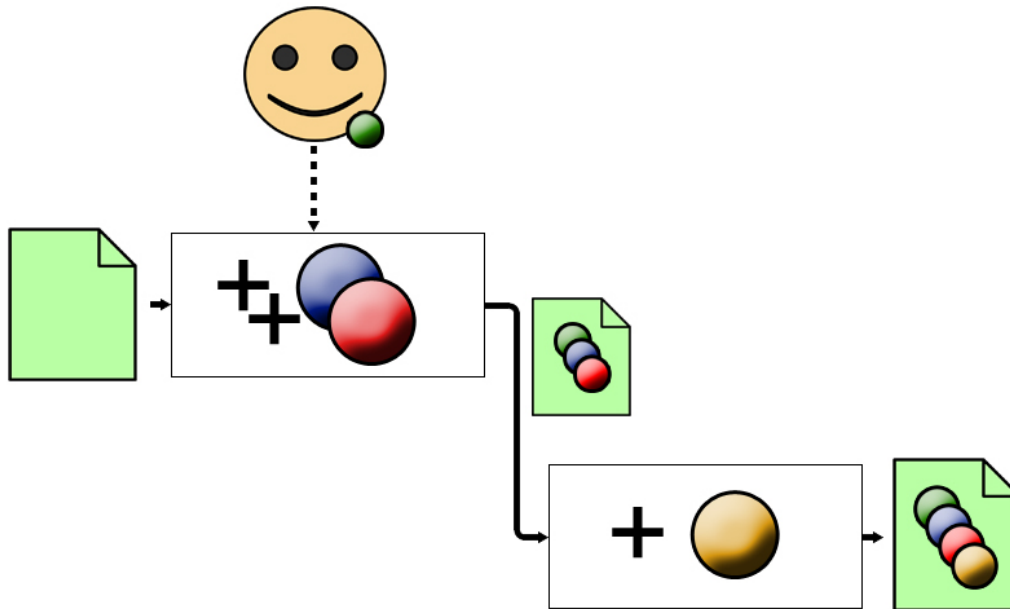


Abbildung 8: Transformation des Aggregatzustandes einer Information in Containermodell 2

Hier wird ein leeres Dokument in den Prozess hinein gegeben, und im ersten Schritt werden mit dem Wissen, das eine bearbeitende Person in „flüssiger“ Form einbringt (grüne Kugel), neue Informationen erzeugt (blau und rot). Alle drei Informationen werden in dem Dokument „fest“ untergebracht und dieses weitergereicht.

Ein Beispiel für einen solchen Prozess wäre ein leeres Formular, das von einer Person um einen ihr bekannten Wert (grün) ergänzt wird. Zwei weitere Felder (blau und rot) können nun berechnet werden, so dass das Dokument nun mit den nötigen Informationen versehen ist, um im nächsten Schritt weiterverarbeitet zu werden.

Die in FLOW vorgesehenen Symbole für die Aggregatzustände können die Datenträger darstellen, die als Container für die Informationen dienen. So können Wechsel von einem zum anderen Aggregatzustand einfach verdeutlicht werden.

3.2.3 Entscheidung und Lösung

Am besten löst der vierte Ansatz durch die Kapselung von Informationen in „Datenträgern“¹ die Schwierigkeit, unter Verwendung der Modellierungssprache IDEFØ, die zur Darstellung von Dokumentenflüssen am ehesten geeignet ist, auch Informationsflüsse zu modellieren.

¹ Im Folgenden werden Dokumente oder Personen, die Informationen beinhalten, als Datenträger bezeichnet. Auch Gerüchte oder „allgemeine Überzeugungen“, wie sie im Rahmen von FLOW für gasförmige Informationen vorsieht, sind in dieser Hinsicht Datenträger, die symbolisch dargestellt werden können.

3.2.3.1 Begründung

- Datenträger zum Transport von Informationen zu verwenden ist ein alltägliches Konzept, das hier zur Anwendung in der Prozessmodellierung kommt. Es ist einfach zu verstehen und erfordert kein Umdenken, wie es bei dem in 3.2.2.3 vorgestellten Containermodell der Fall ist.
- IDEFØ soll zur Darstellung verwendet werden, da in diesem Konzept Datenträger weitergegeben werden und diese Prozessmodellierungssprache die Weitergabe von „Daten oder Objekten“ vorsieht, ist eine Anpassung leicht möglich.
- Auf zusätzliche Dokumente etc., die nur zu Darstellungszwecken erstellt werden müssen, obwohl sie in der Realität nicht vorhanden sind, wie sie in den ersten beiden Ansätzen benötigt werden, kann hier verzichtet werden. Jeder Datenträger kommt einmalig im ganzen Modell vor, wie auch in der Wirklichkeit, und nur die in ihm gehaltenen Daten ändern sich und damit sein Zustand.
- Die Übersicht ist einfacher zu gewährleisten, da weniger und einfachere Namen und Bezeichnungen für die Datenträger vorkommen. Im nächsten Abschnitt wird eine Darstellung vorgeschlagen, die gerade innerhalb einer Software gut visualisierbar umzusetzen ist.

3.2.3.2 Erläuterungen zur Umsetzung

Es sollen Dokumentenfluss wie auch Informationsfluss in der Prozessmodellierung dargestellt werden. Mit diesem Ansatz ist dies nun möglich. Eine Umsetzung kann relativ einfach geschehen, indem eine Software zunächst entsprechend IDEFØ den modellierten Prozess anzeigt. Die Verbindungen zwischen den Schritten, die die Weitergabe von Datenträgern darstellen, lassen sich nun um Tooltips ergänzen, die genaue Informationen über den entsprechenden Datenträger bieten. An dieser Stelle kann leicht eine Liste von enthaltenen Informationen eingeblendet werden, die zu diesem Zeitpunkt in diesem Datenträger abgelegt sind. Mit Hilfe des vordergründig dargestellten Dokumentenflusses können somit auch Informationsflüsse nachvollzogen und ausgewertet werden. Es muss so eine Datenbank von Informationen existieren, die ebenso den Datenträgern zugewiesen werden können wie die Datenträger den Prozessschritten (siehe Anmerkung in Abschnitt 4.3.2: *Datenhaltung und -verwaltung*). Unterstützend kann eine Software den Benutzer darauf hinweisen, dass nach dem Hinzufügen einiger Datenträger als „Eingangsobjekte“ zu einem Schritt, die über bestimmte Informationen verfügen, diese Informationen höchstwahrscheinlich auch in den „Ausgangsobjekten“ des selben Schrittes wieder vorkommen müssen (siehe 4.3.1: *Konzept: „Informationserhaltungssatz“*).

4 Ergebnisse

Im Folgenden werden die Ergebnisse der Arbeit zusammengestellt. Diese gliedern sich entsprechend der Aufgabenstellung in vier Teile: die durchgeführte *Anforderungserhebung* (4.1), die Erkenntnisse zum *Prozessmodell* (4.2) sowie den *Entwurf* (4.3) und die erstellte *Software* (4.4).

4.1 Anforderungserhebung

Zur Erfüllung der Aufgabenstellung wurde auf Grundlage der Untersuchungen ein vollständiges Anforderungsdokument für eine Software erzeugt, wie sie oben beschrieben wird, das in Abschnitt 4.1.2 zu finden ist. Zunächst werden jedoch die Ergebnisse der Untersuchungen zusammengefasst und ausgewertet.

4.1.1 Zusammenfassung der gesammelten Anforderungen

In der Recherche wurden Hinweise gesammelt, welche Aspekte der zu entwickelnden Software besonders entscheidend dafür sind, ob sie mit Erfolg eingesetzt werden kann.

4.1.1.1 Anforderungen an die Benutzeroberfläche durch den Charakter von Interviews

Welche Anforderungen durch die Interview-Situation an die Benutzeroberfläche einer Software gestellt werden, wurde im Anschluss an die Recherche zu Interviews im Abschnitt 3.1.4 bereits zusammengestellt.

4.1.1.2 Weitere Anforderungen durch die Integration der Software in den Ablauf von Interviews

Wie in Abschnitt 3.1.4 bereits angedeutet, ist eine sorgfältige Gestaltung der Benutzeroberfläche nur eine erste Möglichkeit, wie man den widerstreitenden Anforderungen an diesen wichtigen Aspekt *Usability* begegnen kann.

Eine andere Möglichkeit findet sich an grundlegenderer Stelle, die nicht nur vom Entwickler einer entsprechenden Software, sondern auch von den Benutzern Einsatz fordert:

Während meiner Ausbildungszeit in einem Kreditinstitut wurde ich nach Kundengesprächen gelegentlich gebeten, den Kunden anschließend nicht die hübschen Informationsbroschüren zum Thema zu überlassen (zum Beispiel Informationsmaterial mit Beispielrechnungen zu Bausparverträgen), sondern meine während des Gesprächs angefertigten Berechnungen und Skizzen zur Erläuterung von Wohnbauprämie und vermögenswirksamen Leistungen. Diese, wenn auch zweifellos weniger ansprechend gestaltet, mit Korrekturen und Streichungen versehen, hatten die Kunden *gemeinsam* mit mir erstellt, sie waren am Erstellungsprozess dieser Unterlage *beteiligt* gewesen und hatten die dort niedergeschriebenen Erläuterungen „so verstanden“.

Statt den Kunden, in dem hier behandelten Fall den Interviewpartner, also von der Umwandlung seiner Erläuterungen in ein IDEFØ-konformes Prozessmodell auszuscheiden, halte ich es ebenso wie damals in der Bank für eine gute Idee, ihn in den Entwicklungsprozess mit einzubeziehen. Ein Beamer, der während des Gesprächs den Vorgang der Prozesserfassung an die Wand projiziert, macht es dem Interviewpartner möglich, gemeinsam mit den Interviewenden an dem Modell zu arbeiten, Einwände zu äußern oder Fragen zu stellen, statt einige Tage später eine fremd-

erstellte, undurchsichtige Prozessdarstellung mit möglicherweise unzureichenden Erläuterungen präsentiert zu bekommen.

Diese Praktik gibt allerdings verschiedene Punkte zu berücksichtigen:

- Zum einen muss – sehr profan, aber nicht zu unterschätzen – auf Eigenschaften wie die Schriftgröße im Programm geachtet werden: Alle Bedienelemente müssen auch noch in einer Wandprojektion kontrastreich erkennbar und lesbar beschriftet sein.
- Zudem darf die Oberfläche einen „neuen Benutzer“, den der Interviewpartner hier darstellt, nicht überfordern, die Aktionen des bedienenden Mitarbeiters sollten nachvollziehbar und nötigenfalls leicht zu erklären sein.
- Viel wichtiger ist aber noch die Herangehensweise an den Interview-Vorgang an sich. Der Gesprächspartner, von dem im Folgenden eine gemeinsame Umsetzung seiner Prozess-Vorstellungen in IDEFØ erwartet wird, muss auf diese Rolle vorbereitet werden^{1,2}.

¹ Ich gehe im Folgenden davon aus, dass der Interviewpartner ebenso wie die mit der Anforderungserhebung betrauten Mitarbeiter dazu bereit sind, auf diese Weise gemeinsam an der Prozesserstellung zu arbeiten – wenn einer der Beteiligten diese Arbeitsweise ablehnt, kann es schwerlich Aufgabe einer Software sein, ihn dazu zu überreden.

² Zu Vorzügen von IDEFØ und FLOW bezüglich der Verständlichkeit und der Möglichkeit, die jeweiligen Modellierungskonzepte zu erlernen, siehe Kapitel 2: *Zu verwendende Notationen*.

4.1.2 Anforderungsdokument

Für eine zusammenfassende Darstellung der im vorigen Abschnitt genannten Anforderungen wurde die Vorlage für Anforderungsdokumente des Fachgebiet Software Engineering verwendet, die zur Bearbeitung der dort jährlich durchgeführten Softwareprojekte zur Verfügung gestellt wird.

Durch diese Zusammenfassung soll es vereinfacht werden, die als Erkenntnisse dieser Arbeit ermittelten Anforderungen bei einer Entwicklung einer entsprechenden Software zu verwenden.

(Hinweis: Dieses Anforderungsdokument beschreibt *nicht* die Anforderungen, an die im Rahmen dieser Arbeit zu erstellende Software. Es ist für entsprechende Projekte im Allgemeinen abstrakt gehalten. Die für diese Arbeit geltenden Nebenbedingungen wurden bewusst ausgeblendet.)

4.1.2.1 Mission des Projekts

Erläuterung des zu lösenden Problems

In der Anforderungserhebung sollen Entwickler bei der Modellierung von Prozessen noch während der Interviews durch eine Software unterstützt werden. Geschilderte Prozesse sollen vor Ort modelliert werden. Die Software soll dabei den Anwender unterstützen, Vergleiche zwischen den in verschiedenen Gesprächen erstellten Prozessmodellen anzustellen. Innerhalb eines Prozesses sollen Verbindungen zwischen den Prozessschritten automatisch eingezeichnet werden. Das Verständnis für die Prozesse soll auf beiden Seiten gefördert werden. Durch gemeinsame Erstellung möglichst korrekter Prozessmodelle sollen zudem Missverständnisse bei der Erläuterung der vorliegenden Prozesse ausgeräumt werden.

Wünsche und Prioritäten des Kunden

Das Erfassen und Sichern von Informationen zu Prozessen und das schnelle Ordnen dieser Daten unter hektischen Bedingungen, sowie der Zugriff auf wieder verwendbare Daten (zum Beispiel von einem anderen Gesprächspartner genannte Dokumente, der zum selben Prozess befragt wurde) steht im Vordergrund.

Domänenbeschreibung

Der Anwendungsbereich der zu entwickelnden Software liegt hauptsächlich im Einsatz auf einem mobilen PC, während eines Interviews in einem fremden Büro.

Dies bedeutet einerseits, dass eigene Hardware eingesetzt wird und das Programm „in Ruhe“ auf dem System installiert und die Nutzung vorbereitet werden kann. Eine Portierbarkeit auf Systeme vor Ort oder ähnliches ist nicht erforderlich.

Zur Abgrenzung gegenüber Produkten wie Microsoft Visio und ähnlichen Produkten ist andererseits die Hektik der Interview-Situation zu berücksichtigen: die Anwendung soll wesentlich mehr als gängige Modellierungssoftware die schnelle und einfache Erfassung und Darstellung von Daten ermöglichen.

Maßnahmen zur Anforderungsanalyse

Es wurden Gespräche mit Fachleuten auf dem Gebiet von Interviews im Allgemeinen und im Rahmen der Anforderungserhebung geführt (3.1: *Zusammenfassung der Recherche und Einzelergebnisse*). Zudem wurden die zu verwendenden Notationen hinsichtlich einer gemeinsamen Darstellung von Dokumenten- und Informationsflüssen untersucht (3.2.1: *Problematik Informationsfluss/Dokumentenfluss*).

4.1.2.2 Rahmenbedingungen und Umfeld

Einschränkungen und Vorgaben

Es ist die Prozessmodellierungssprache IDEFØ in Verbindung mit den Aggregatzuständen für Informationsflüsse aus FLOW zu verwenden. Um Dokumentenflüsse wie Informationsflüsse gemeinsam darstellen zu können, soll der Lösungsansatz „Container 2: Dokumente als Container für Informationen“ implementiert werden.

Eine Bildschirmauflösung von 1024x768 Pixeln ist nicht zu überschreiten.

Die Darstellung muss für die Projektion mit einem Beamer geeignet sein.

Anwender

Das Produkt soll von geschulten Fachkräften bedient werden. Allerdings soll es in Beisein ungeschulter Personen zum Einsatz kommen, die in der Lage sein sollen, die Aktionen des Benutzers auf dem Bildschirm nachzuvollziehen.

Es sollen Informationen gesammelt und zu einem Prozessmodell zusammengestellt werden. Die Atmosphäre während der Verwendung dürfte hektisch sein, schnelle Reaktionen bei der Datenerfassung, Änderungsmöglichkeiten und Unterstützung durch Anbieten kontextabhängiger, sinnvoller Optionen sind erforderlich, um unnötiges Suchen zu ersparen.

Erfasste Daten sollen sofort gespeichert werden, um für den Fall eines Systemfehlers, Stromausfalls etc. den aktuellen Stand gesichert zu haben und daran anknüpfen zu können.

Ordnen, Verwalten und Nachbereiten von Daten soll möglich sein, um während des Einsatzes im Interview keine Zeit mit Organisation zu verlieren.

Schnittstellen und angrenzende Systeme

Ein Export der Daten per XML muss möglich sein, um die Darstellung und Nachbereitung der erfassten Prozesse in anderen Werkzeugen zu ermöglichen.

Eine grafische Darstellung der Prozesse im Programm selbst muss möglich sein, um die Besprechung der Modellierung zu ermöglichen.

4.1.2.3 Funktionale Anforderungen

Use Case-Diagramm

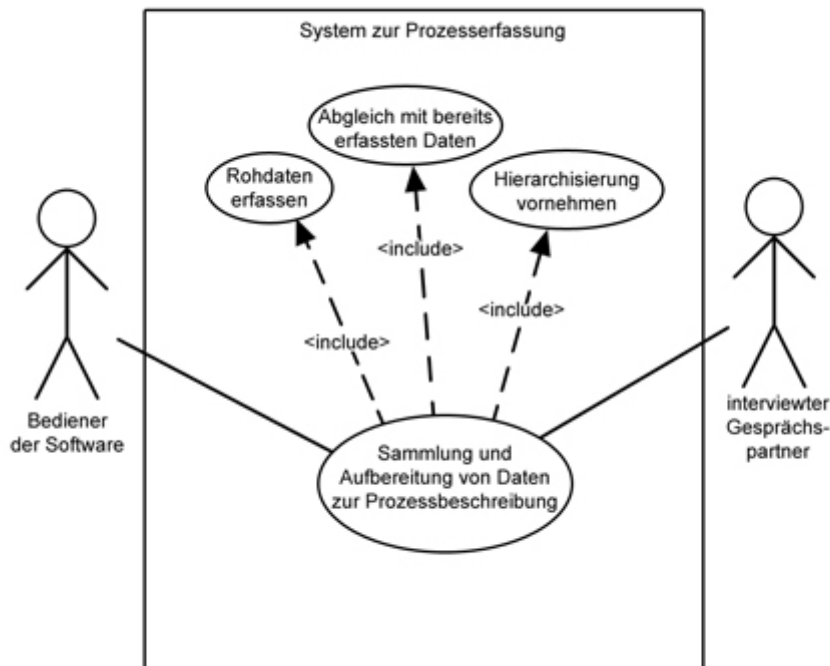


Abbildung 9: Use Case Diagramm zur Software

Use Case-Beschreibungen

Siehe 1.3: *Zur Erläuterung der zu erstellenden Software: Use Cases*

Technische Anforderungen

Die Darstellung per Beamer erfordert einen deutlichen Aufbau der Benutzeroberflächen und gut lesbare Texte.

Die Datenbank soll jederzeit konsistent sein und den aktuellen Stand der erfassten Daten sichern.

4.1.2.4 Qualitätsanforderungen

Qualitätsziele des Projekts

Die schnelle Bedienbarkeit in einem hektischen Umfeld, das rasche Einpflegen von Änderungen und die gute Übersicht und Verständlichkeit der Oberfläche (auch für ungeschulte Beobachter der Bedienung) stehen im Vordergrund.

Es sollen effizient, schnell und korrekt (siehe unten) die beschriebenen Prozesse modelliert werden können.

Qualitäts-Prioritäten des Kunden

Die effiziente, schnelle und korrekte (siehe unten) Erfassung von Daten und Modellierung von Prozessen stellen die Priorität des Kunden dar.

Die Darstellung von Dokumentenflüssen und Informationsflüssen soll unterstützt werden, die Notationen von IDEF0 und FLOW sollen hierbei verwendet werden.

Es ist dabei nicht entscheidend, dass die Darstellung innerhalb der Software allen formalen Anforderungen der IDEF0-Definition genügt, solange sie die Prozesse ver-

ständig wiedergibt und das Navigieren durch die verschiedenen Verfeinerungsebenen ermöglicht.

Exportmöglichkeiten erlauben die Nachbearbeitung der gesammelten Daten und gehören ebenfalls zu dessen Qualitäts-Prioritäten.

Wie Qualitätsziele erreicht werden sollen

In dieser Arbeit werden einige Möglichkeiten vorgestellt, wie es erreicht werden kann, den Anforderungen entsprechend „effizient, schnell und korrekt“ Prozesse zu modellieren. Zusammenfassend sollen diese Aspekte unter anderem erreicht werden durch Wiederverwendung, Automatisierung und Vergleichsmöglichkeiten für den Benutzer:

Schnelligkeit	Bereitstellen einer Möglichkeit, Prozesse zu vergleichen,
Effizienz	Anbieten von bereits verwendeten Datenobjekten (Glossar) und
korrekte Erfassung	Anzeigen von logischen Verbindungen bzw. Fehlern in erstellten Prozessen, um gezieltes Nachfragen zu ermöglichen.

4.1.2.5 Probleme und Risiken

Die Benennung von Risiken ist projektabhängig und kann an dieser Stelle nicht vorgenommen werden.

(Allgemein ist die Programmierung einer Benutzeroberfläche eine sehr zeitaufwändige Aufgabe, die oft unterschätzt wird. Damit die Bedienung des fertigen Produkts leicht von der Hand geht und intuitiv geschehen kann, ist während der Entwicklung sehr viel Fingerspitzengefühl und häufig einfach auch „Ausprobieren“ verschiedener Varianten gefragt, bis sich die optimale Gestaltung eines Menüs gefunden hat oder ein möglichst leicht verständliches Bedienkonzept umgesetzt werden konnte. Während der Entwicklung der zu dieser Arbeit gehörenden Software wurde diese Erfahrung – wieder einmal – gemacht, und daher soll hier als Hinweis angebracht werden, den „Qualitätsaspekt Usability“ hinsichtlich des darin verborgenen Arbeitsaufwandes nicht zu unterschätzen.)

4.2 Prozessmodell

Um die gemeinsame Darstellung von Informations- und Dokumentenflüssen zu ermöglichen, wurde in Kapitel 3.2.1: *Problematik Informationsfluss/Dokumentenfluss* die Untersuchung verschiedener möglicher Lösungsansätze durchgeführt.

Eine begründete Entscheidung für den im Abschnitt 3.2.2.4 unter dem Titel „Container 2: Dokumente als Container für Informationen“ vorgestellten Ansatz wird dort getroffen.

4.3 Entwurf

Oben wurden verschiedenste Anforderungen genannt, die eine Prozessmodellierungssoftware zur Unterstützung von Interviews erfüllen muss, um den erwünschten Nutzen zu bringen. Im Folgenden werden Ideen aufgeführt, diese Anforderungen in einem Entwurf umzusetzen.

4.3.1 Usability

Es wurde in Abschnitt 3.1.4: *Extraktion von Anforderungen für die Modellierungssoftware* auf die Wichtigkeit benutzerfreundlicher Bedienung hingewiesen. Dort wurden „clevere Shortcuts“ gefordert und die Unterstützung des Benutzers durch eine schnörkellose Software, die unkompliziert logische Ergänzungen anbietet und zur Strukturierung anleitet, ohne dass der Benutzer in seiner Freiheit eingeschränkt ist, auf Gesprächswendungen zu reagieren.

Es gibt verschiedene Möglichkeiten, die geforderte „künstliche Intelligenz“ in Form situationsabhängiger, sinnvoller Optionen in der Software anzubringen.

Konzept: „Scheuklappen statt Vogelperspektive“

Die Vorgabe, die durch die zu verwendende Modellierungssprache gemacht wird, legt ein „Bottom-Up“-Verfahren¹ nahe: statt zunächst eine Übersicht und die Zusammenhänge von Prozessschritten darzustellen und später die Details dieser Zusammenhänge zu klären, sollte hier von den einzelnen Schritten aus gearbeitet werden. Um die Übersicht zu behalten, konzentriert sich das Interview immer nur auf einen Prozessschritt auf einmal, der mit allen ein- und ausgehenden Daten und Objekten definiert wird. Dabei werden die verwendeten Datenobjekte in ein Glossar aufgenommen. Nach Erfassung einiger Schritte werden Objekte auftauchen, die bereits verwendet wurden, und so ergeben sich im Hintergrund Zusammenhänge zwischen den einzelnen Schritten (ohne dass sich der Erfasser und der interviewte Gesprächspartner mit diesen Zusammenhängen auseinandersetzen müssen): Dokument A, das als Ausgabe von Schritt I angegeben wurde, erscheint bei Schritt III als Steuerungsdokument. Diese Verbindung kann von der Software leicht ermittelt und dargestellt werden, was zwei Reaktionen auslösen kann:

- die Erkenntnis, dass man sich über diesen Zusammenhang nicht klar war, oder
- die Feststellung, dass eine Eingabe fehlerhaft ist, da die eingezeichnete Verbindung nicht der Wirklichkeit entspricht.

In jedem Fall hat die Software hier dazu beigetragen, den Modellierungsprozess zu unterstützen.

¹ Im Gegensatz zu der von Frau Nguyen favorisierten Top-Down-Methode, die mit der Erfassung der Schritt-Zusammenhänge beginnt (siehe 3.1.2: *Gespräch mit Thao Nguyen*).

Konzept: „übergeordnete Ebenen auswerten“

Wenn die Verfeinerung eines Prozessschrittes vorgenommen wird, indem die in ihm stattfindenden Vorgänge durch weitere Schritte detaillierter dargestellt werden (vergleiche Abbildung 10: *Verfeinerung eines Prozessschrittes*), kann die Software die Datenobjekte, die an den „Vater“-Schritt gebunden sind, bei der Bearbeitung der „Kinder“-Schritte anbieten, da sie hier irgendwo vorkommen *müssen*.

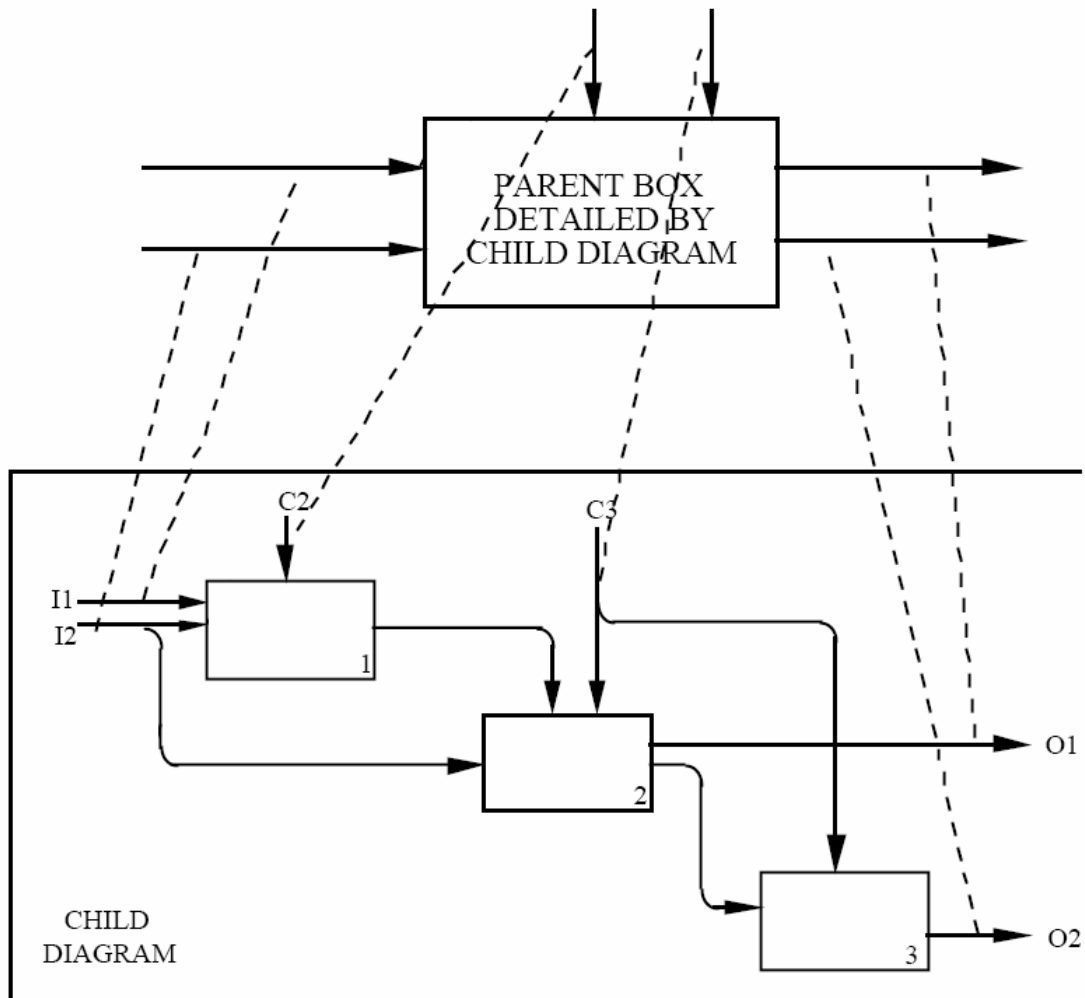


Abbildung 10: Verfeinerung eines Prozessschrittes

Speziell für die Verwendung als „Kontrollen“ können die Datenobjekte C2 und C3 angeboten werden, I1 und I2 als „Eingänge“ und O1 und O2 als „Ausgänge“. An welchen der verfeinernden Schritte diese Objekte nun „angeschlossen“ werden, bleibt dem Benutzer überlassen, er wird sie aber ohnehin für diese Verfeinerung benötigen und muss sie so nun nicht erst aus dem Glossar heraussuchen und kann kein Objekt vergessen. Sobald ein Datenobjekt, seiner Anwendung im „Vater“-Schritt entsprechend, in einem der „Kinder-Schritte“ verwendet worden ist, könnte es aus der Liste der angebotenen Objekte entfernt werden (allerdings ist es auch möglich, dass ein Datenobjekt in der Verfeinerung mehrfach verwendet werden soll, wie zum Beispiel C3 in obiger Zeichnung; hier wäre vielleicht eine Kopier-Funktion hilfreich, mit der bereits verwendete Datenobjekte einfach erneut an einen anderen Schritt gebunden werden können).

Konzept: „Informationserhaltungssatz“

Eine ähnliche Situation findet sich bei der Umsetzung des unter *Container 2: Dokumente als Container für Informationen* beschriebenen Konzepts, bei dem Informationen an die Datenobjekte angehängt werden, um den Informationsfluss darzustellen (zur Umsetzung siehe auch Anmerkung in Abschnitt 4.3.2). Hierzu ist im Folgenden die oben beschriebene Situation der Aggregat-Transformation noch einmal dargestellt:

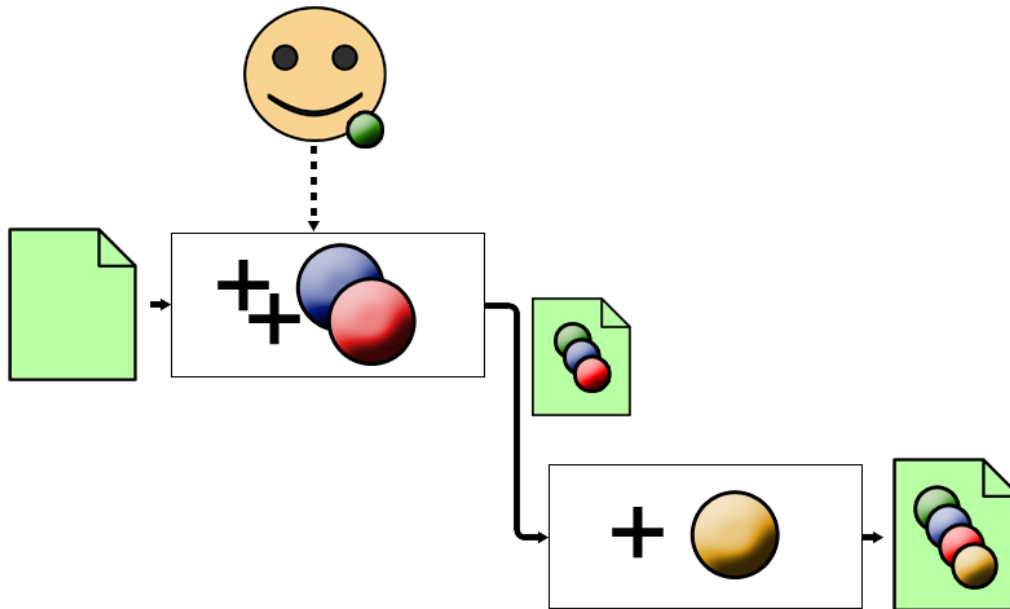


Abbildung 11: Transformation des Aggregatzustandes einer Information in Containermodell 2

Wenn eine Information in einen Schritt hineingegeben wird (wie hier durch die grüne Kugel symbolisiert), gibt es die Möglichkeit, dass sie den Träger wechselt und in der Ausgabe wieder auftaucht, wie es oben dargestellt ist. Dies muss zwar nicht der Fall sein, da die Information ja auch einfach zur Bearbeitung des Schrittes benötigt werden kann, ohne in den Ergebnissen aufzutauchen. Dennoch bietet sich hier wieder die Möglichkeit, dem Benutzer eine sinnvolle Aktion anzubieten um seinen Aufwand zu mindern: Die Übernahme der Informationen, die in den in einen Schritt hinein fließenden Dokumenten vorhanden sind, in diejenigen Dokumente, die hinaus fließen.

Konzept: „Sandbox“

Um im Interview die Möglichkeit zu haben, jederzeit den Gedankensprüngen des Gesprächspartners zu folgen und in jeder Situation Daten festhalten zu können, müssen die entsprechenden Eingabemasken jederzeit erreichbar sein und dürfen nicht von anderen Fenstern verdeckt werden. Da spontan erfasste Daten aber vielleicht nicht sofort in eine Struktur wie einen Hierarchie-Baum, der das ganze Projekt darstellt, eingeordnet werden können, werden sie zunächst in einem Zwischenspeicher, der so genannten „Sandbox“ abgelegt. Diese ist in Kategorien unterteilt, so dass die Objekte schnell wiedergefunden werden können. Es ist möglich, Datenobjekte, Prozessschritte und sogar ganze Prozesse hier abzulegen und sie auch nachzubearbeiten. Sobald der Einsatz der so „geparkten“ Elemente im Projekt sinnvoll erscheint, können sie per Drag&Drop an die entsprechende Position verschoben werden.

Dieses Konzept setzt die Notwendigkeit um, frei und spontan mit der Erfassung folgen zu können, egal wie unvorhergesehen das Interview sich entwickelt. So gehen keine Daten verloren, weil aufgenommene Informationen erst in einen Kontext gebracht werden müssen: sie werden ohne Zusammenhang erstmal „gesichert“.

4.3.2 Datenhaltung und -verwaltung

Für die Prozessdaten bietet sich eine relationale Datenbank an, sowie eine Klassenstruktur, die diese während des Programmablaufs durch Objekte abbildet.

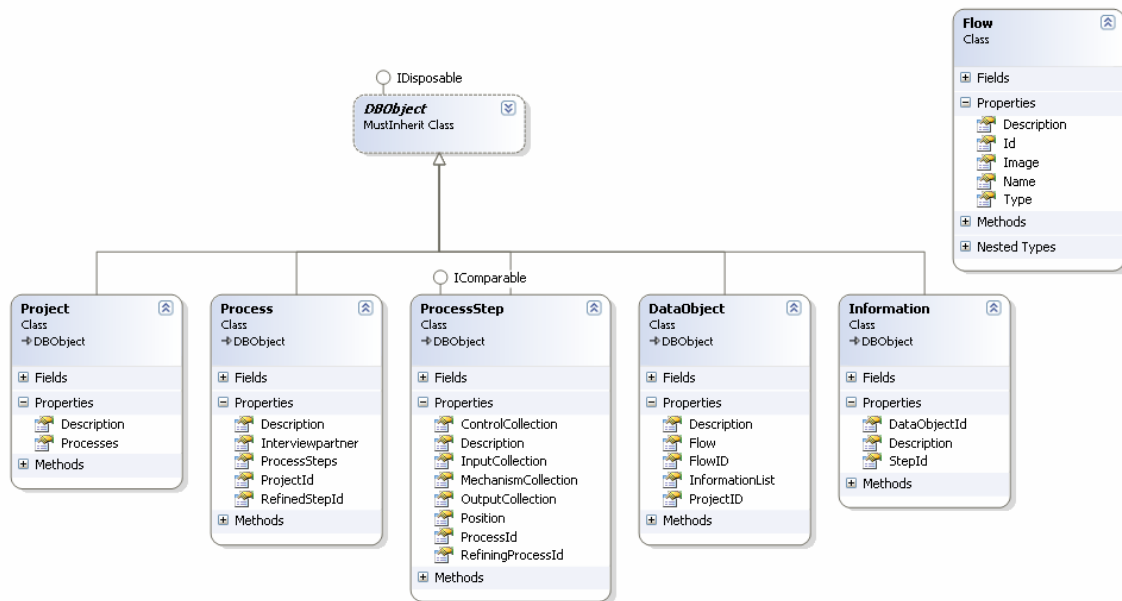


Abbildung 12: Klassenstruktur zur Abbildung einer Datenbank für Prozessdaten

In der dargestellten Struktur stehen Projekte als Organisations-Einheiten auf oberster Ebene und beinhalten die Prozesse, die ihnen zugeordnet wurden. Prozesse wiederum bestehen aus Prozessschritten. Diesen sind Listen mit Datenobjekten zugewiesen, die als Ein-, Ausgänge, Kontrollen und Mechanismen für den jeweiligen Schritt auftreten. Datenobjekte können in jeder Funktion verwendet werden und sind daher nicht an eine bestimmte Verwendung gebunden. Ihnen wird einer der Aggregatzustände zugewiesen, der durch ein Objekt der Klasse „Flow“ als fest, flüssig, gasförmig oder unbekannt beschrieben wird.

Anmerkung: Ebenfalls in obigem Diagramm zu sehen ist ein Ansatz, das im Abschnitt 3.2.2.4 beschriebene Konzept *Container 2: Dokumente als Container für Informationen* umzusetzen. Zur Lösung des Problems, Informationen in die Dokumentenfluss-Darstellung mit einzubeziehen, sollen hier Objekte der „Information“-Klasse an eine Liste des Datenobjekts angehängt werden, wodurch die im Datenobjekt vorhandenen Informationen dargestellt werden. Jede Information bezieht sich auf einen Prozessschritt, in dem sie „generiert“ wird, also der Liste des Datenobjekts hinzugefügt wird. Dieses Konzept konnte aufgrund des späten Zeitpunkts, zu dem die entsprechende Aufgabenstellung ergänzt wurde (siehe 3.2: *Erweiterung der Aufgabenstellung im Zwischenvortrag*), nicht mehr vollständig getestet werden und ist daher nur in Form dieses Fragments angedacht.

4.4 Software

Als erste wichtige zu treffende Entscheidung für die Umsetzung des Entwurfs im Rahmen dieser Arbeit wurde als Programmiersprache Visual Basic .NET 2005 von Microsoft gewählt. Dies geschah angesichts des hohen Stellenwerts, den der Entwurf der Benutzeroberfläche in dieser Software einnimmt. Mit VB .NET 2005 stehen komfortable Steuerelemente zur Verfügung, Konzepte der Bedienung wie Drag&Drop lassen sich einfach umsetzen und es besteht die Möglichkeit, komplexe eigene Steuerelemente mit geringem Aufwand zu erstellen, was in diesem Fall zweckmäßig erschien. Zudem ist VB .NET 2005 eine komplett objektorientierte Programmiersprache und erlaubt ggf. den Einsatz von Design-Patterns ebenso wie es beispielsweise unter Java möglich wäre.

Leider war es im Rahmen dieser Arbeit nicht möglich, neben der Erhebung der Anforderungen, der Untersuchung einer Möglichkeit zur gemeinsamen Darstellung von Informations- und Dokumentenfluss und dem Entwurf von Lösungsideen für die gesammelten Anforderungen, alle Ideen des Entwurfs auch tatsächlich umzusetzen. Es wurde vom „Auftraggeber“ der Software, Herrn Professor Schneider, eine Priorisierung der verschiedenen Aspekte durchgeführt, um die wichtigsten Konzepte in der Software vorstellen zu können.

So war gefordert, dass die erfassten Prozesse in einer IDEFØ-nahen Notation dargestellt werden. Die Konsistenz der Datenbank, in der die Prozessdaten hinterlegt werden, war neben einer den hohen Usability-Anforderungen entsprechenden Benutzeroberfläche ebenfalls eine Haupt-Priorität. Der Export in XML und die Umsetzung der Verbindung von Informations- und Dokumentenfluss wurden dagegen hinten gestellt (auch deshalb, weil die Informations-/Dokumentenfluss-Problematik erst etwa nach der Hälfte der Bearbeitungszeit ergänzend zur Aufgabenstellung dazukam und dann zunächst untersucht werden musste).

Im Laufe der Umsetzung habe ich dann selbst eine weitere Bewertung und Abstriche vornehmen müssen, da nicht alle Ideen des Entwurfs termingerecht umsetzbar waren. Diese Punkte zusammen mit bekannten Problemen, die nicht rechtzeitig gelöst werden konnten, sind unter 4.4.4: *Nebenaspekte und Abstriche* zusammengefasst.

Zunächst werden in Abschnitt 4.4.1 wiederverwendbare Steuerelemente vorgestellt, die im Rahmen der Arbeit erstellt wurden. Was das Werkzeug bereits leistet, wird in Abschnitt 4.4.2 beschrieben. In Abschnitt 4.4.3 wird auf Schnittstellen der Software eingegangen.

4.4.1 Wiederverwendbare Steuerelemente

Entsprechend der vorgegebenen Prioritäten ist die IDEF0-Darstellung sehr weit entwickelt. Es wurde ein Steuerelement erstellt, das einen übergebenen Prozess mit der Möglichkeit, durch die Verfeinerungsebenen der Prozessschritte zu navigieren und die Schritte per Drag&Drop in einer sinnvollen Reihenfolge anzuordnen. Zudem kann durch Anklicken eines Schrittes dessen Detailansicht geöffnet und seine Verknüpfungen mit Datenelementen bearbeitet werden. Die Verbindungen zwischen den Schritten werden automatisch ermittelt und eingezeichnet. Dabei werden die Aggregatzustände der Daten durch die entsprechenden Linienstile der FLOW-Notation dargestellt (auch wenn es sich hier nicht um Informationsflüsse, sondern um weitergegebene oder aktiv werdende „Datenträger“ handelt, wie Dokumente oder Personen – die Form, in der sie die nicht näher bezeichneten Informationen vorhalten, wird durch die Aggregate fest, flüssig und gasförmig aus FLOW bezeichnet).

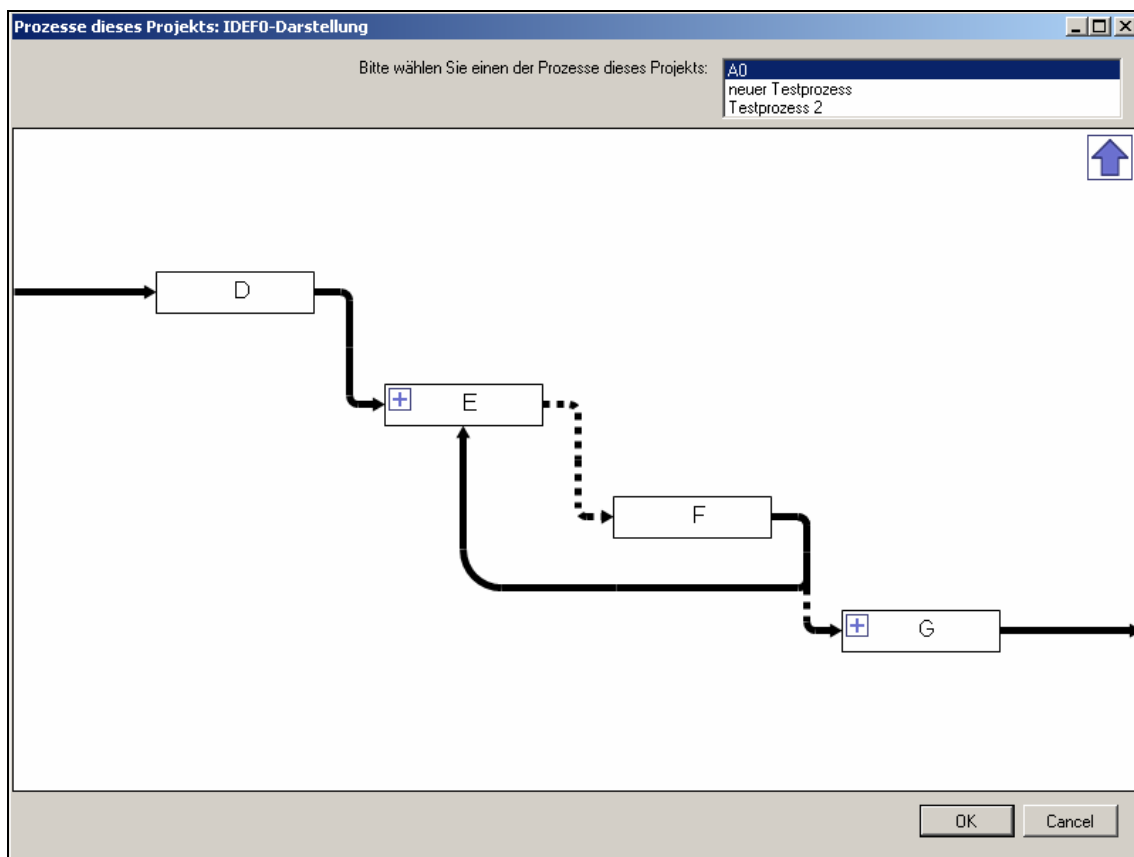


Abbildung 13: Das Steuerelement "IDEF0ProcessGraph"

Dieses Steuerelement „IDEF0ProcessGraph“ (die weiße Zeichenfläche im obigen Screenshot) kann problemlos von Entwicklern unter Visual Basic .net 2005 weiterverwendet werden, die die Klassen des zugehörigen IDEF0-Pakets verwenden.

Ebenfalls um ein wieder verwendbares Steuerelement handelt es sich bei der Klasse „IDEF0StepDetail“, die eine Eingabeoberfläche (oben erwähnte „Detailansicht“) für die Daten eines Prozessschrittes darstellt:

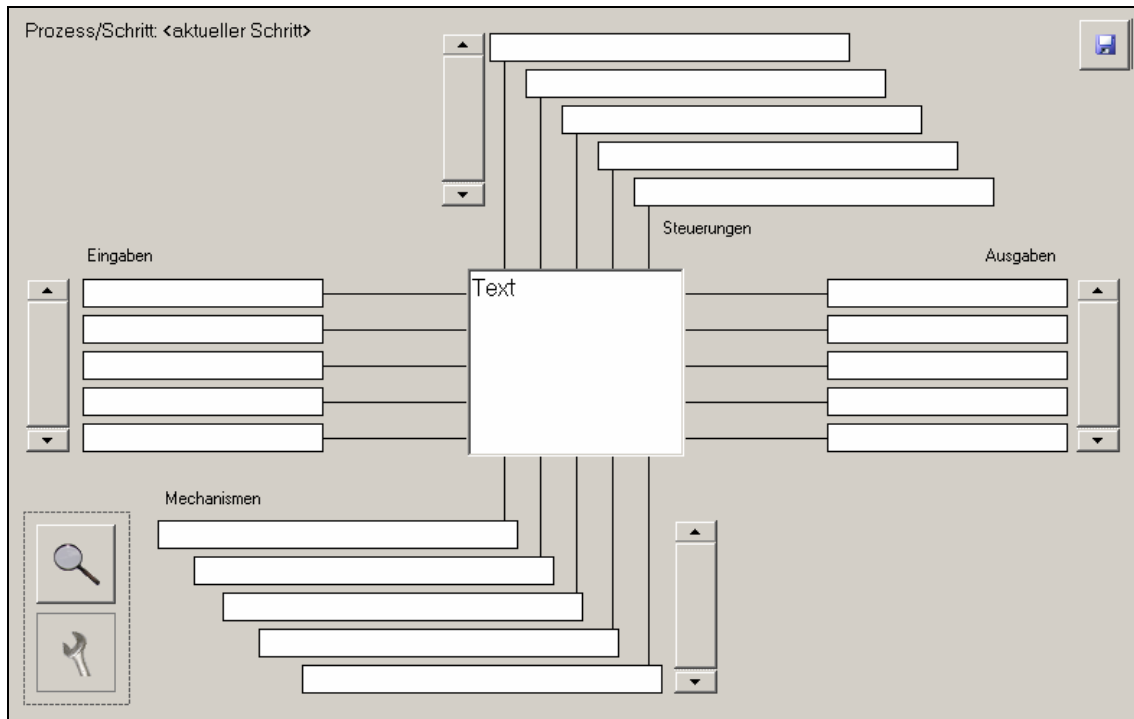


Abbildung 14: Das Steuerelement "IDEF0StepDetail"

Dieses Steuerelement kann zwischen einem Bearbeitungs- und einem Ansichtsmodus umgeschaltet werden, um die zur Dateneingabe nötigen Elemente auszublenden und eine übersichtlichere Darstellung zu ermöglichen. Von den Objekten, die als Eingaben, Ausgaben, Steuerungen und Mechanismen mit dem Schritt verknüpft werden können, sind maximal fünf sichtbar, alle weiteren sind über Scroll-Leisten zu erreichen. Ein Problem bei der Verwendung dieses Steuerelements ist der große Platzbedarf, den die Übersicht erfordert. Die Lesbarkeit der Schrift kann nur durch eine weitere Vergrößerung erreicht werden, was dazu führt, dass außer diesem Element nichts anderes auf die Benutzeroberfläche passt.

Die obige Ansicht verwendet weitere Steuerelemente, wie zum Beispiel die für die Darstellung der FLOW-Aggregatzustände erstellte Klasse „FlowArrow“, die einen der Notation entsprechenden Pfeil anzeigt, bei Bedarf auch mit dem zugehörigen Aggregat-Symbol. Über ein Kontext-Menü lässt sich der Aggregatzustand ändern.

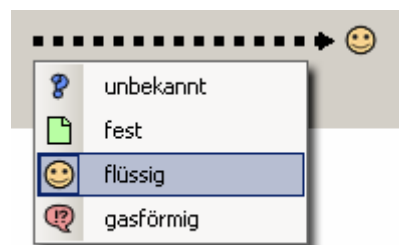


Abbildung 15: Das Steuerelement "FlowArrow " mit ausgeklapptem Kontextmenü

4.4.2 Funktionen des Programms

Die Software ermöglicht das Anlegen von Projekten, in denen Prozesse erzeugt und Prozessschritte eingefügt werden können. Prozessschritte lassen sich verfeinern, das heißt durch andere Prozesse darstellen. Die Anzeige der Hierarchie in einer herkömmlichen Baumstruktur ist möglich, ebenso die IDEF0-Darstellung durch das oben beschriebene Steuerelement IDEF0GraphView.

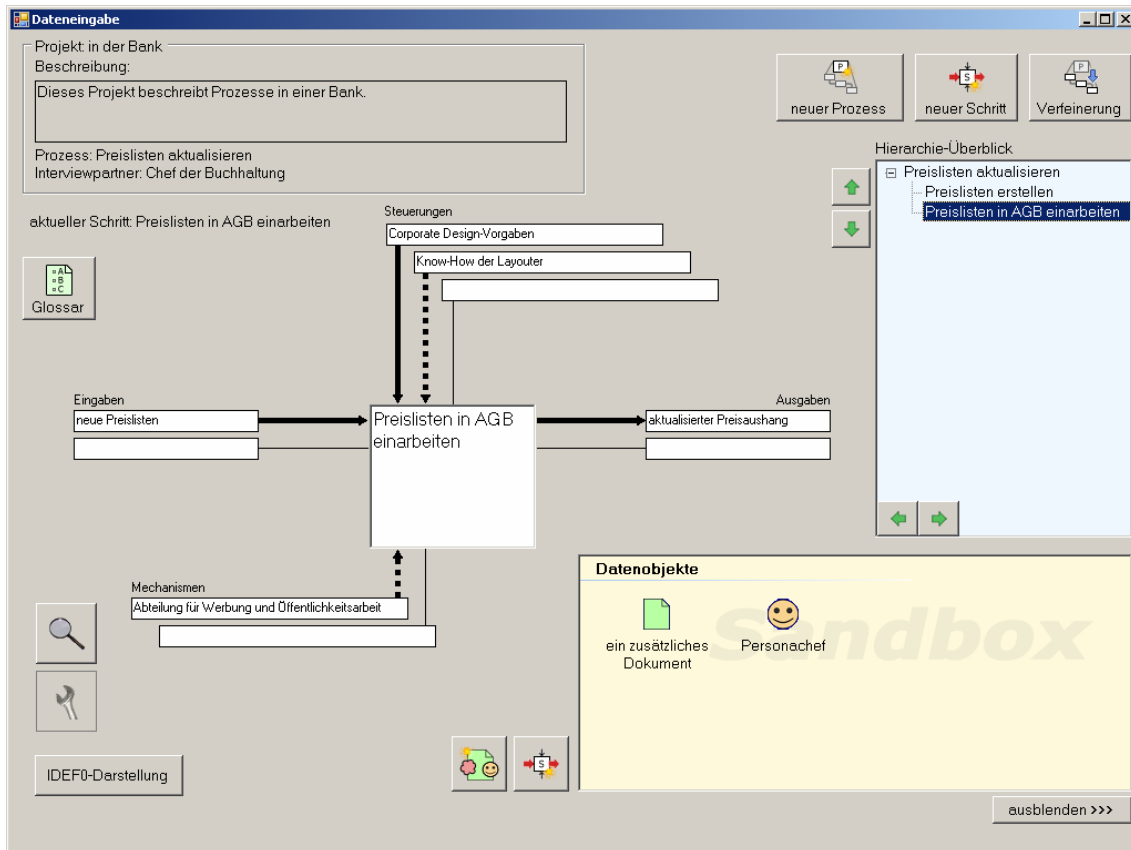


Abbildung 16: Das als "DataGrabber" bezeichnete Hauptmenü der Software

Des Weiteren wurde das Konzept der „Sandbox“ teilweise umgesetzt, es ist möglich, Datenobjekte eines beliebigen Aggregatzustands zu erstellen und dort abzulegen.

Alle Elemente (Projekte, Prozesse, Schritte und Datenobjekte) lassen sich umbenennen und mit einer Beschreibung versehen.

In der Datenbank korrekt hinterlegte Projekte können von den verschiedenen Ansichten dargestellt werden.

4.4.3 Schnittstellen

Vor dem Hintergrund einer Zusammenarbeit der erstellten Software mit anderen Werkzeugen, in denen die gesammelten Daten noch präziser dargestellt werden (Meißner-Projekt) und/oder nachbearbeitet werden können, sind Schnittstellen nach außen notwendig. Auch wäre es denkbar, dass in anderen Programmen erstellte Prozessdarstellungen in das Programm übernommen werden sollen.

4.4.3.1 XML-Export

Die Übertragung der Daten mittels eines XML (Extensible Markup Language)-Formats bietet sich hier an. Um im Bereich des Fachgebietes Software Engineering die gemeinsame Nutzung von Prozessdaten zu vereinfachen wurde versucht, mit

den Bearbeitern anderer Projekte, die mit solchen Daten umgehen, einen gemeinsamen Standard, wie GML (Graph Modeling Language) oder GXL (Graph eXchange Language), zu vereinbaren. Aufgrund der unterschiedlichen Reifegrade der Projekte konnte jedoch keine Einigung getroffen werden.

Der Export der dargestellten Prozesse ist in der vorliegenden Software nur versuchs- und teilweise implementiert. Es stellte sich heraus, dass beispielsweise das zur Verdeutlichung der Prozess-Hierarchie verwendete TreeView-Steuerelement eine Umwandlung der dargestellten Daten in eine XML-Notation sehr leicht ermöglicht. Nach dieser Feststellung wurden die diesbezüglichen Versuche eingestellt; die Priorität in der Entwicklungsphase lag eindeutig auf der Sicherstellung, dass die gesammelten Daten zunächst konsistent in der verwendeten MS Access-Datenbank vorliegen. Diese kann auch außerhalb der Software in ein XML-Format überführt werden.

4.4.4 Nebenaspekte und Abstriche

Die Entwicklung einer Benutzeroberfläche ist sehr aufwändig, und grade in diesem speziellen Fall, wo Bedienkomfort eine besondere Rolle spielt, fällt es schwer ins Gewicht, dass einige Funktionen nicht bis zum Abschluss dieser Arbeit implementiert werden konnten. Dennoch kann auf die Implementierung einer bestimmten, besonders einfachen Bedienung (wie Drag&Drop zum Übernehmen von Dokumenten aus dem Glossar in einen Prozessschritt), ggf. verzichtet werden, wenn es eine andere Möglichkeit gibt, auch wenn diese vielleicht weniger komfortabel ist.

Zum Zeitpunkt des Abschlusses dieser Arbeit war die wichtige Aufgabe, Datenobjekte mit bestimmten Rollen in einem Prozessschritt zu verknüpfen, nicht funktionsfähig umgesetzt. Aus Zeitgründen mussten die Arbeiten leider eingestellt werden, es liegen jedoch keine unlösbaren strukturellen Probleme vor. Meiner Einschätzung nach ist die Fertigstellung und Umsetzung der in dieser Arbeit vorgestellten Konzepte in einem überschaubaren Zeitrahmen möglich.

Weitere Informationen, was noch nicht umgesetzt werden konnte und noch zu tun bleibt, wurden in Kapitel 5: *Ausblick* zusammengestellt.

4.4.4.1 Sicherheit

Auf Sicherheitsaspekte wurde im Rahmen dieser Arbeit keinen Wert gelegt. Die verwendete Datenbank ist unverschlüsselt. Sie beinhaltet mehrere Projekte, so dass während eines Interviews bei einem Kunden theoretisch auch das Projekt eines anderen Kunden geöffnet werden könnte. Der Datenschutz ist also nicht berücksichtigt, was durch Verschlüsselung der Daten, Passwortvergabe und Anlegen getrennter Datenbanken für verschiedene Projekte verbessert werden könnte.

4.4.4.2 Performance

Aus Performance-Gründen wurde beim Entwurf der Prozessklassen eine dem Fewton-Pattern ähnliche Idee umgesetzt: Da jeder Datensatz nur einmal in der Datenbank existiert, wird nur, wenn das erste Mal auf ihn zugegriffen wird, ein entsprechendes Objekt erstellt. Dieses wird in einer Hash-Liste registriert, und sollte derselbe Datensatz später erneut benötigt werden, wird nun zunächst das Element aus dieser Liste zurückgegeben. Wenn beispielsweise ein Dokument von 100 verschiedenen Schritten verwendet wird, würden beim Gleichzeitigen Laden dieser Schritte nicht 100 Instanzen der DataObject-Klasse mit den Daten dieses Dokuments

erzeugt, sondern nur eine einzige. Auf diese Weise kann der Speicherbedarf des Programms deutlich gesenkt werden.

Bei der Berechnung der Verbindungen, die aufgrund der anhängenden Datenobjekte zwischen den Schritten eines Prozesses bestehen, sind ineinander verschachtelte Schleifenkonstruktionen notwendig, um alle Konstellationen zu prüfen. Da es sich aber meist nur um wenige Schritte auf einer Prozessebene handelt (mehr als sechs Schritte sind laut der offiziellen Definition von IDEF nicht zugelassen), stellen diese Schleifen kein Performance-Problem dar.

4.4.4.3 Portierbarkeit

Dieser Aspekt war von vorne herein von untergeordneter Bedeutung. Die Software soll im typischen Anwendungsfall in der Interviewsituation auf einem mitgebrachten Laptop ausgeführt werden, was eine Kompatibilität mit möglichst vielen anderen Systemen unnötig macht. (Dennoch benötigt die Umsetzung in Microsoft Visual Basic .net grundsätzlich nur das .net-Framework, um die Software auszuführen. Es ist zu erwarten, dass für viele Plattformen in Zukunft ein Framework zur Verfügung stehen wird. Dies ist möglich weil Microsoft die Spezifikationen über das ECMA Komitee freigegeben und standardisiert hat.)

4.4.4.4 Dokumentation und Kommentare

Der Quellcode ist an den wichtigsten Stellen mit Kommentaren versehen. Eine spezielle Formatierung der Kommentare erlaubt den XML-Export und die Erzeugung einer Dokumentation.

5 Ausblick

Die grundlegende Struktur, die benötigt wird, um die angestrebte Software fertig zu stellen, wurde vorbereitet und kann genutzt werden. Es wurden keine schwerwiegenden Probleme festgestellt, die es grundsätzlich unmöglich machen würden, die in dieser Arbeit vorgestellten Konzepte umzusetzen.

So ist die fehlende Funktionalität des Steuerelements „IDEF0StepDetail“, die es verhindert, dass Datenobjekte an einen Schritt gebunden werden, auf den Zeitmangel zurückzuführen, der eine Fertigstellung verhinderte.

Bekannte Probleme

- Im als „DataGrabber“ bezeichneten Hauptmenü findet bei Änderungen an der Hierarchie oder bei den Bezeichnungen der aufgeführten Prozesse, Schritte, etc. manchmal keine Aktualisierung der Anzeige statt. (Um die Benutzung der Oberfläche einigermaßen zu ermöglichen wurde eine mit „refresh“ bezeichnete Schaltfläche in der Nähe der Baumansicht platziert, durch deren Betätigung eine Aktualisierung aller Objekte erzwungen werden kann.)
- Das Anfügen von Datenobjekten in der Schritt-Detailansicht aus dem Glossar oder aus der Sandbox per Drag&Drop ist nicht implementiert.
- Das Anfügen von Datenobjekten in der Schritt-Detailansicht durch manuelle Eingaben ist nicht implementiert.
- Die automatische Ergänzung in der Schritt-Detailansicht durch Glossareinträge ist nicht implementiert.
- Es können Laufzeitfehler auftreten, die zum Absturz der Software führen, wenn ungültige Datenbanksituationen auftreten (nachdem z.B. die Datenbank manuell verändert wurde oder ein Systemabsturz unvollständige Datensätze erzeugt hat).
- Es können Darstellungsfehler im Steuerelement „IDEF0ProcessView“ auftreten, wenn zu viele Schritte in einem Prozess dargestellt werden sollen.
- Es können Darstellungsfehler im Steuerelement „IDEF0ProcessView“ auftreten, wenn die Anordnung der Schritte eine nicht unterstützte Linienform oder unmögliche Winkel provoziert.

6 Quellen- und Abbildungsverzeichnis

6.1.1.1 Quellen:

1. „B-KIDE: A Framework and a Tool for Business Process Oriented Knowledge Infrastructure Development“; Markus B. Strohmaier; Shaker Verlag, Januar 2005 (ISBN: 3-8322-3620-1)
2. „Announcing the Standard for Integration Definition for Function Modeling (IDEF0)“; National Institute of Standards and Technology; Draft Federal Information - Processing Standards Publication 183, 21.12.1993
3. „Software Engineering nach Maß mit FLOW“; Kurt Schneider; SQMcongress 2006, Düsseldorf;
4. „Software Process Improvement from a FLOW Perspective“; Kurt Schneider; LSO 2006

6.1.1.2 Abbildungen:

(Abbildungen ohne Quellenangabe sind selbst erstellt.)

Abbildung 1: Ein Prozessschritt in der IDEF0-Darstellung (Seite 9)	aus: „Announcing the Standard for Integration Definition for Function Modeling (IDEF0)“; Draft Federal Information - Processing Standards Publication 183; National Institute of Standards and Technology, 21.12.1993
Abbildung 2: Ein Prozess mit mehreren Detaillierungsstufen (Seite 10)	
Abbildung 10: Verfeinerung eines Prozessschrittes (Seite 32)	
Abbildung 3: Die Aggregatzustände von Informationen in FLOW (Seite 11)	nach Abbildungen aus: „Aggregatzustände von Anforderungen erkennen und Nutzen“, Vortrag von Prof. Dr. Kurt Schneider auf dem GI-Fachgruppentreffen RE 2005 in Hannover, 24. und 25.11.2005
Abbildung 4: komplexe Bedienoberfläche der Visualisierungssoftware Microsoft Visio 2000 (Seite 15)	Quelle: http://en.wikipedia.org Datei: .../wiki/Image:Screengrab_-_Microsoft_Visio_2000_-_example_diagram.png
Abbildung 5: Transformation des Aggregatzustandes einer Information in Containermodell 1 (Seite 21)	
Abbildung 8, Abbildung 11: Transformation des Aggregatzustandes einer Information in Containermodell 2 (Seiten 23, 33)	
Abbildung 6: Ansammlung einer großen Menge an Informations-Objekten (Seite 22)	
Abbildung 7: Weitergabe eines sich im Prozess mit Informationen füllenden Dokuments (Seite 22)	
Abbildung 9: Use Case Diagramm zur Software (Seite 29)	
Abbildung 12: Klassenstruktur zur Abbildung einer Datenbank für Prozessdaten (Seite 34)	
Abbildung 13: Das Steuerelement "IDEF0ProcessGraph" (Seite 36)	
Abbildung 14: Das Steuerelement "IDEF0StepDetail" (Seite 37)	
Abbildung 15: Das Steuerelement "FlowArrow " mit ausgeklapptem Kontextmenü (Seite 37)	
Abbildung 16: Das als "DataGrabber" bezeichnete Hauptmenü der Software	

7 Anhang

7.1 Software-Installation

Eine Installation der Software an sich ist nicht notwendig. Die auf der CD vorhandene ausführbare Datei muss gemeinsam mit der Datenbank, die die Prozessinformationen enthält, im selben Verzeichnis liegen.

7.1.1 Benötigtes Framework

Für den Betrieb der Software muss das Microsoft .NET-Framework 2.0 installiert sein, für das sich eine Installationsanwendung auf der CD befindet. Für die Ausführung dieser Installation ist Microsoft Windows xp (SP2) erforderlich.

7.1.2 Beiliegender Quellcode

Der Quellcode der erstellten Anwendung ist ebenfalls auf der CD zu finden.

Zum Betrachten des Codes eignet sich die kostenlose Entwicklungsumgebung Microsoft Visual Basic 2005 Express Edition, die unter folgender Adresse heruntergeladen werden kann:

<http://www.microsoft.com/germany/msdn/vstudio/products/express/vb>