

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

# **Verfeinerung und Visualisierung von Informationsflüssen in Software-Entwickler-Teams**

## **Bachelorarbeit**

im Studiengang Technische Informatik

von

**Lennart Schroth**

**Prüfer: Prof. Dr. Kurt Schneider  
Zweitprüfer: Prof. Dr. Joel Greenyer  
Betreuer: Nils Prenner**

**Hannover, 24. Januar 2019**



## Kurzfassung

Kommunikation und der damit verbundene Informationsfluss ist gerade bei Software-Projekten ein zentraler Aspekt. Unzureichende Kommunikation oder nicht vermittelte Anforderungen können negative Auswirkungen auf das zu erstellende Software-Produkt haben. Die daraus folgende Unzufriedenheit des Kunden verzögert damit das erfolgreiche Abschließen des Produktes oder Projektes und kann zu kostenintensiven Nacharbeiten führen.

Das Fachgebiet Software Engineering (SE) des Instituts für Praktische Informatik der Leibniz Universität Hannover (LUH) beschäftigt sich seit 2008 im Rahmen des Forschungsprojekts FLOW mit Informationsflussanalysen. Die dabei entwickelte FLOW-Methode umfasst die Datenerhebung, die Modellierung bzw. Visualisierung der Informationsflüsse in FLOW-Modellen. Auf Grundlage dieser erhobenen Daten wird eine Analyse durchgeführt mit dem Ziel, die Kommunikation zu optimieren.

Für Analytinnen und Analysten ist in diesem Zusammenhang die Übersichtlichkeit und Verständlichkeit dieser Modelle relevant, um globale Zusammenhänge zu erkennen. Für diesen Zweck werden abstrakte und globale Modelle benötigt. Dem steht allerdings der Bedarf nach gerade detaillierten Informationen gegenüber, die für eine genaue Analyse notwendig sind. Aus diesem Grund müssen verschiedene Informationen in unterschiedlichen Granularitäten für Analysezwecke aufbereitet werden.

Aus dieser Motivation heraus wurden Konzepte entwickelt, um FLOW-Modelle zu verfeinern und damit die Informationsflussanalysen zu optimieren. Die in dieser Arbeit entwickelten Konzepte wurden mit einer Gruppe von Experten des SE diskutiert. Auf dieser Grundlage erfolgte die Implementation des Top-down-Konzeptes für hierarchische Detailgrade in einem Software-Tool. Für die dahinter liegende Datenstruktur fiel die Entscheidung auf das entwickelte Link-Konzept. Die daraufhin implementierten Funktionalitäten zur hierarchischen Verfeinerung von FLOW-Modellen wurden im Anschluss wieder mit den Experten diskutiert.

Zusammenfassend lässt sich festhalten, dass die verschiedenen Granularitäten die FLOW-Diagramme verfeinern und eine Informationsflussanalyse optimieren können, dies aber nur funktioniert, wenn die Anzahl an verwendeten Unternetzwerke und Ebenen bedacht eingesetzt wird.



## Abstract

Communication and the related flows of information are key aspects of software projects. Inadequate communication or not transmitted requirements have a negative impact on the software product being developed. The resulting dissatisfaction of the customer is crucial for the successful completion of the product or the project and can lead to additional costs.

The Department of Software Engineering (SE) of the Institute for Practical Computer Science at the Leibniz Universität Hannover (LUH) has been working on analyzes for flows of information since 2008 as part of the research project FLOW. The developed FLOW-method includes data collection, modeling and visualization of flows of information in FLOW-models. Based on the data collection a analysis is performed to optimize the communication.

For analysts are the clarity and comprehensibility of these models relevant to recognise global interrelationships. For this purpose abstract and global models are needed. In contrary to this detailed information are necessary for an accurate analysis. For this reason, different information on different granularities must be prepared for analysis purposes.

Out of this motivation, concepts were developed to optimize flows of information analyzes. The developed concepts were discussed by a group of experts. On this basis the top-down concept for hierarchical details is implemented in a software tool. For the background data structure, the decision was made on the developed link concept. The implemented functionalities for the hierarchical refinement of FLOW-models were discussed again with the experts.

In summary, the different granularities refine the FLOW-diagrams and allow an analysis for flows of information when the number of subnetworks and levels is used carefully.



# Inhaltsverzeichnis

<b>1</b>	<b>Einleitung</b>	<b>1</b>
1.1	Problemstellung . . . . .	2
1.2	Zielstellung . . . . .	2
1.3	Aufbau der Bachelorarbeit . . . . .	3
<b>2</b>	<b>Stand der Entwicklung</b>	<b>5</b>
2.1	FLOW-Methode . . . . .	5
2.1.1	Grundlegende Konzepte . . . . .	5
2.1.2	Feste und flüssige Informationen . . . . .	6
2.1.3	Grafische FLOW-Notation . . . . .	7
2.2	Merging . . . . .	9
2.3	Verwandte Arbeiten . . . . .	9
<b>3</b>	<b>Konzeptentwicklung</b>	<b>11</b>
3.1	Konzept zur Visualisierung . . . . .	11
3.1.1	Konzept nach der Top-down-Methode . . . . .	13
3.1.2	Konzept nach der Bottom-up-Methode . . . . .	16
3.2	Konzepte zur Strukturverfeinerung . . . . .	19
3.2.1	Strukturkonzept Link . . . . .	19
3.2.2	Strukturkonzept All-in . . . . .	20
3.3	Mögliche kombinatorische Realisierungen . . . . .	21
<b>4</b>	<b>Konzeptentscheidung</b>	<b>23</b>
4.1	Vorbereitungen zur Entscheidungsfindung . . . . .	23

4.2	Durchführung der Konzeptentscheidung . . . . .	24
4.3	Ergebnisse . . . . .	25
4.4	Ergebnis-Interpretation . . . . .	25
<b>5</b>	<b>Konzeptimplementierung</b>	<b>27</b>
5.1	Anforderungen an die Software . . . . .	27
5.2	Softwareseitige Umsetzung . . . . .	29
5.2.1	Link-Strukturverfeinerung . . . . .	30
5.2.2	Top-down-Visualisierung . . . . .	33
5.3	Softwaretest . . . . .	39
5.4	Erweiterbarkeit . . . . .	40
<b>6</b>	<b>Interpretation der Ergebnisse</b>	<b>41</b>
6.1	Planung . . . . .	41
6.2	Durchführung . . . . .	41
6.3	Auswertung und Ergebnisse . . . . .	42
6.4	Reliabilität . . . . .	45
6.5	Empfehlung . . . . .	46
<b>7</b>	<b>Fazit und Ausblick</b>	<b>47</b>
7.1	Fazit . . . . .	47
7.2	Ausblick . . . . .	48
	<b>Abbildungsverzeichnis</b>	<b>50</b>
	<b>Literaturverzeichnis</b>	<b>52</b>
	<b>Anhang</b>	<b>54</b>
	Dokumente . . . . .	54
	FLOW-Diagramme . . . . .	55

# Kapitel 1

## Einleitung

Die Auswirkungen unzureichender Kommunikation in einem Projektteam werden vielfach in der Fachliteratur diskutiert. Unzureichende oder nicht kommunizierte Anforderungen vom Kunden führen oftmals zum Fehlen wichtiger Funktionen im endgültigem Softwareprodukt. Ein Produkt bzw. ein Projekt kann aber nur mit der Zufriedenheit des Kunden auch erfolgreich entwickelt bzw. abgeschlossen werden. Mit dem Ziel der Optimierung von Kommunikation werden Unternehmen vor diesem Hintergrund damit konfrontiert, Informationsflussanalysen zu implementieren. Der Stellenwert der Kommunikation und der damit im Zusammenhang stehende Informationsfluss ist auch bei Software-Projekten weitgehend bekannt [1]. Mit dem Aufkommen der agilen Methodik in der Software-Entwicklung wurde die Thematik der Kommunikation noch höher gewichtet. Dieses wurde in dem ersten der vier Manifeste („Individuen und Interaktionen mehr als Prozesse und Werkzeuge“) folgerichtig festgehalten [2].

Das Fachgebiet Software Engineering (SE) des Instituts für Praktische Informatik der Leibniz Universität Hannover (LUH) beschäftigt sich seit 2008 im Rahmen des Forschungsprojekts FLOW mit dem Informationsfluss in einem Team von Softwareentwicklerinnen und -entwicklern<sup>1</sup>. Dabei geht es um die Ermittlung von Daten zum Informationsfluss, um deren Modellierung bzw. Visualisierung und schließlich um die Analyse der erhobenen Daten mit dem Ziel, die Kommunikation zu optimieren [3]. Die FLOW-Methode wird kontinuierlich verbessert, um eine möglichst präzise Informationsflussanalyse zu gewährleisten [4].

---

<sup>1</sup>Aus Gründen der besseren Lesbarkeit wird in dieser Arbeit nicht konsequent gegendert. Die maskuline Form impliziert aber stets die feminine.

## 1.1 Problemstellung

Zur Optimierung der Informationsflussanalysen wurden in der FLOW-Methode Hierarchie-Stufen vorgesehen, mittels derer detaillierte Informationen in unterschiedlichen Detailgraden in den FLOW-Modellen bereitgestellt werden können. Für Analytistinnen und Analysten ist relevant, dass diese Modelle übersichtlich und verständlich sind, um globale Zusammenhänge zu erkennen. In diesem Zusammenhang ist ein abstrakteres einem zu detaillierten Modell vorzuziehen. Dem steht allerdings der Bedarf nach gerade detaillierten Informationen gegenüber, die für eine genaue Analyse notwendig sind. Aus diesem Grund müssen verschiedene Informationen in unterschiedlichen Granularitäten zur Verfügung gestellt werden. Diese unterschiedlichen Detailgrade werden allerdings noch nicht softwaretechnisch unterstützt. Da viele FLOW-Modelle und FLOW-Analysen noch handschriftlich umgesetzt werden, existiert für diese Weiterentwicklung der FLOW-Diagramme auch noch keine allgemeine Herangehensweise oder Notation.

Eine weitere Besonderheit in der FLOW-Methode ist das Anlegen von sogenannten Activities. Diese Activities abstrahieren verschiedene Informationsspeicher zu einer gröberen Einheit und dienen somit zur Vereinfachung von komplexeren FLOW-Diagrammen. So können mehrere Knoten des Netzwerkgraphen eines FLOW-Diagrammes zu einem abstrakten Knoten zusammengefasst werden. Kiesling et al. haben sich mit Analyse von FLOW-Modellen mittels der Social-Network-Analyses (SNA) beschäftigt. Die festen und flüssigen Informationsspeicher, die bei FLOW grundsätzlich unterschieden werden, können mit diesen SNA's ohne besondere Berücksichtigung analysiert werden. Stand der Dinge ist allerdings, dass Activity-Knoten nur mit einschränkenden Anpassungen analysierbar sind [5]. Daher ist es wichtig, diese Activities aufzulösen.

## 1.2 Zielstellung

Vor diesem Hintergrund ist das Ziel der vorliegenden Arbeit, Modelle für Informationsflüsse zu verfeinern, um Informationsflussanalysen in Software-Entwickler-Teams zu optimieren. Für eine Optimierung von Informationsflussanalysen sollen dazu die FLOW-Modelle mit hierarchischen Stufen erweitert werden können, um sowohl abstraktere, als auch detaillierte FLOW-Diagramme zur Verfügung stellen zu können.

Die hierarchischen Detailgrade können bei der Problematik mit den nicht zu analysierenden Activities ebenfalls genutzt werden. Diese Umsetzung ermöglicht das Auflösen dieser Knoten in rein analysierbare feste und flüssige Knoten. Dies macht Social-Network-Analyses (SNA) einfacher

nutzbar, da keine weiteren Anpassungen notwendig sind.

Für das Erarbeiten dieser hierarchischen Detailgrade werden in dieser Bachelorarbeit verschiedene Konzepte vorgestellt, um softwareseitig die Modelle zu verfeinern und besser zu visualisieren und damit Informationsflussanalysen zu optimieren.

## **1.3 Aufbau der Bachelorarbeit**

Im zweiten Kapitel „Stand der Entwicklung“ wird zunächst die bereits existierende FLOW-Methode bzw. deren Entwicklungsstand vorgestellt. Da die Entwicklung der Konzepte auf den Erkenntnissen vorangehender Arbeiten aufbaut, werden die wichtigsten diesbezüglichen Publikationen kurz vorgestellt.

In Kapitel 3 werden das Konzept zur Visualisierung und das zur Verfeinerung präsentiert. Dabei werden mögliche Ansätze skizziert, die die Grundlage bilden für die Entwicklung von jeweils zwei Konzepten für die Visualisierung der hierarchischen Detailgrade und der dazu erforderlichen Verfeinerung der Datenstruktur.

Kapitel 4 thematisiert die Konzeptentscheidung mit der Vorbereitung, der Durchführung sowie der Entscheidung und schließlich der Diskussion und Interpretation der Ergebnisse.

Gegenstand von Kapitel 5 ist die Implementierung dieser Konzepte in die bereits existierende FLOW-Software. Die dafür aufgestellten Systemanforderungen sowie Softwaretests und mögliche Erweiterungen werden ebenfalls skizziert.

Das Kapitel 6 „Interpretation der Ergebnisse“ beinhaltet die Beschreibung der Planung und Vorgehensweise bei der Einholung des Feedbacks zu den implementierten Konzepten sowie dessen Auswertung und Ergebnisse. Abgeschlossen wird dieses Kapitel durch eine kurze Empfehlung für den Umgang mit Hierarchiestufen in FLOW-Modellen.

Im letzten Kapitel wird ein Fazit dieser Arbeit gezogen, das auch reflektierende Aspekte beinhaltet. Anregungen für mögliche weitere Arbeiten zu Informationsflussanalysen mit der FLOW-Methode werden in dem Ausblick gegeben.



## Kapitel 2

# Stand der Entwicklung

Nachfolgend wird zunächst die FLOW-Methode im Detail vorgestellt und kurz auf den Begriff „Merging“ eingegangen. Anschließend wird auf Arbeiten hingewiesen, an die angeknüpft wird und die Einfluss auf die Konzepterstellung im Rahmen dieser Arbeit haben.

### 2.1 FLOW-Methode

Die FLOW-Methode umfasst ein Verfahren zur Modellierung, Analyse und Optimierung von Informationsflüssen in Software-Entwickler-Teams. Wie einleitend erwähnt, ist diese im Rahmen des Forschungsprojektes FLOW am Fachgebiet Software Engineering (SE) des Instituts für Praktische Informatik der Leibniz Universität Hannover (LUH) entstanden. Eines der entscheidenden Grundlagenwerke zur FLOW-Methode ist „FLOW-Methode - Methodenbeschreibung zur Anwendung von FLOW“[3] von Stapel und Schneider. Die nachfolgenden Ausführungen basieren hauptsächlich auf diesem Werk.

#### 2.1.1 Grundlegende Konzepte

Bei einer nach der FLOW-Methode durchgeführten Analyse wird der Informationsfluss in einem Software-Team oder Unternehmen durch eine Analystin oder einen Analysten dokumentiert und modelliert. Die FLOW-Methode umfasst dabei die nachfolgenden drei Phasen.

Phase 1: Datenerhebung

Daten werden über den internen Informationsfluss in einem Entwicklerteam oder Unternehmen mit Hilfe von Interviews erhoben.

### Phase 2: Datenmodellierung und Datenanalyse

Die gesammelten Informationen werden modelliert und analysiert. Nach den Vorgaben der FLOW-Notation wird ein FLOW-Modell bzw. FLOW-Diagramm generiert. Dieses Diagramm enthält einen Netzwerkgraphen, der die Anforderungen der FLOW-Notation erfüllen muss. Der Begriff des Netzwerkes oder Netzwerkgraphen wird daher auch als Synonym für das FLOW-Diagramm genutzt. Auf die FLOW-Notationen wird weiter unten im Detail eingegangen. Anhand der abgebildeten Strukturen im Diagramm können dann zum Beispiel strukturelle Schwächen erkannt und analysiert werden.

### Phase 3: Erarbeitung von Verbesserungsstrategien

Die Erkenntnisse, die während der Datenerhebung, Modellierung und Analyse gewonnen worden sind, werden dazu genutzt, Verbesserungsstrategien zu entwickeln [3].

## 2.1.2 Feste und flüssige Informationen

Die Besonderheit bei der FLOW-Methode besteht darin, dass zwischen „festen“ und „flüssigen“ Informationen unterschieden wird, wobei damit auf den Informationsfluss in verschiedenen Medien Bezug genommen wird, d.h. während feste Informationen „in einem Betrachtungsbereich von allen jederzeit abgerufen und verstanden werden können“ [3] sind flüssige Informationen „nicht fest“ [3].

Folglich sind Dokumente ein Speicher von festen Informationen. Sie können von verschiedenen Personen eingesehen und kopiert werden, ohne dass sich der Inhalt verändert. Dabei ist es nicht relevant, ob diese in Papierform oder elektronisch abgelegt werden.

Personen repräsentieren bei FLOW einen flüssigen Informationsspeicher. Teilinformationen können vergessen werden oder Meinungen sich ändern. Ebenfalls gelten flüchtige Informationsspeicher wie Notizzettel und White-Board-Notizen als flüssig.

Die Einordnung von E-Mails ist nicht immer eindeutig und stellt daher eine Ausnahme dar. Wird eine E-Mail archiviert, gilt sie als feste Information. Wenn sie aber nur dem schnellen Austausch dient und gelöscht wird oder werden kann, gilt sie als flüssige Information.

### Grundlegende Regeln

Die folgenden Ziele gelten seit der Entwicklung von FLOW als grundlegende Regeln, die für ein effizientes Arbeiten mit FLOW eingehalten werden sollen [3].

1. Informationsflüsse sind Bindeglied zwischen dokumentenlastigen und

kommunikationsintensiven Ansätzen.

2. Modelle dienen in erster Linie als Diskussionsgrundlage und sind daher so einfach wie möglich, d.h. aber auch so detailliert wie nötig.

3. Inhalte werden meist nur grob modelliert.

4. Erfahrungen sind eine besonders wertvolle Art von Informationen.

- Informationen haben einen Aggregatzustand.

Um das Verständnis für die Ausführungen der folgenden Kapitel zu erleichtern, wird noch ein Beispiel eines FLOW-Diagramms gegeben. Dieses entstand im Rahmen einer Einzelfallstudie zu verteilten Softwareprojekten. Hierzu wurden zwei Teams in Spanien und Deutschland während einer Softwareentwicklung begleitet und auf der Grundlage ihrer Kommunikation wurde ein FLOW-Modell erstellt [6]. Dieses FLOW-Diagramm ist in Abbildung 2.1 gezeigt.

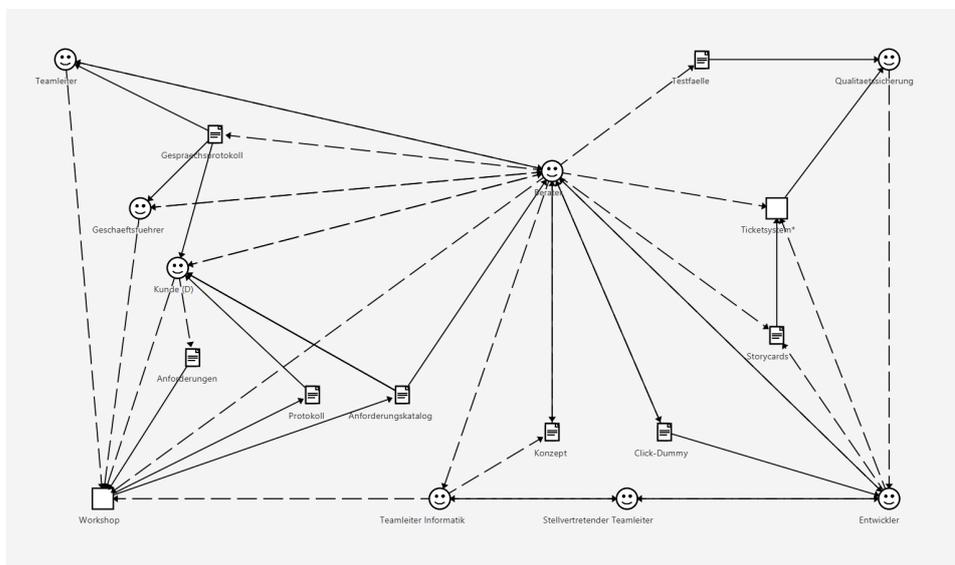


Abbildung 2.1: Beispiel eines FLOW-Diagrammes bei verteilten Softwareprojekten

### 2.1.3 Grafische FLOW-Notation

Die Visualisierung der FLOW-Notation wird auf den Grundlagen der Netzwerktechnik betrieben. Hierbei sind die Knoten die Informationsspeicher und die Kanten die einzelnen Informationsübertragungen. In FLOW wird bei Informationsträgern bzw. Informationsspeichern unterschieden zwischen Personen, Dokumenten und Aktivitäten. Häufig werden dabei auch die englischen Begriffe person, document und activity verwendet. Diese drei bilden die Knoten (engl. vertices) des Netzwerkgraphens. Knoten werden in diesem Zusammenhang auch als Vertex oder Member bezeichnet.

1. Eine Person ist ein Informationsspeicher für flüssige Daten. Dieser wird als runder Kreis mit Smiley-Gesicht visualisiert. Die Kommunikation ist der dazugehörige Austausch von flüssigen Informationen zwischen einzelnen Personen.

2. Ein Dokument ist ein Container von festen Informationen. Es enthält einen festen Informationsgehalt und kann auch größere Werke, zum Beispiel ein Lastenheft, darstellen. Visualisiert wird ein Dokument mit einem rechteckigen Dokumentensymbol. Das Dokumentieren beschreibt entsprechend das Festhalten von Informationen.

3. Eine Aktivität kann verschiedene Informationsspeicher zusammenfassen. Dies dient zur Abstraktion und auch zum Teil als Hierarchiemittel. Personen und Dokumente können in einem Activity-Knoten zusammengefasst werden. Durch das weiße Quadrat können somit unterschiedliche Informationsspeicher in einer abstrakteren Einheit dargestellt werden.

Das Überführen von Informationen wird als Informationsfluss (auch Transformation, Aggregatzustandsübergang) bezeichnet. In der FLOW-Notation bilden die festen und flüssigen Informationsflüsse die Kanten des Netzwerkgraphen im FLOW-Diagramm. Feste Informationsflüsse werden mit einem durchgezogenen Pfeil dargestellt. Flüssige, die zum Beispiel von einer Person ausgehen, werden gestrichelt gezeichnet. Generell werden alle Pfeile schwarz dargestellt. Wenn allerdings bedacht werden soll, dass es sich bei einer Information um eine Erfahrung handelt, kann diese grau gezeichnet werden.

Auf einen Knoten zeigende Pfeile werden standardmäßig als Eingänge interpretiert. Wegführende Pfeilspitzen repräsentieren Ausgänge. Eine Besonderheit bilden hierbei die Activity-Knoten. Pfeile, die von oben oder unten in den Knoten rein- oder rausgehen, werden als Steuerungs- oder Erfahrungsflüsse aufgefasst.

In Abbildung 2.2 wird die FLOW-Notation nochmal graphisch dargestellt.

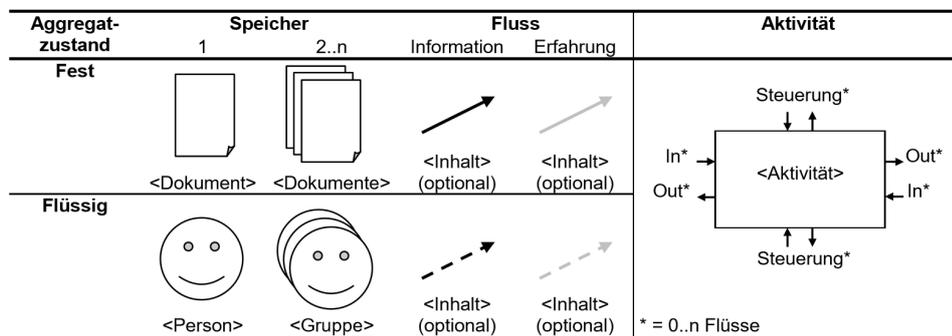


Abbildung 2.2: Syntax der grafischen FLOW-Notation

## 2.2 Merging

Entwickler von Softwareprodukten werden häufig mit dem Begriff „to merge“ oder „merging“ konfrontiert. Dieser kommt aus dem Englischen und bedeutet soviel wie verschmelzen, vereinigen oder zusammenführen. In der Softwareentwicklung wird darunter das Verschmelzen von Dateien verstanden. Gerade durch die Verbreitung des Datenverwaltungstools Git ist dieser Begriff in der Softwareentwicklung fast überall bekannt. Umgangssprachlich wird allerdings auch das „neudeutsche“ Verb mergen verwendet.

## 2.3 Verwandte Arbeiten

### **FLOW-Methode - Methodenbeschreibung zur Anwendung von FLOW**

Stapel und Schneider haben in diesem Werk alle Grundlagen für die FLOW-Methode zusammengefasst. Dieses enthält und beschreibt alle wichtigen Begriffe, Definitionen und Methoden. Alle nicht explizit verwiesenen Grundlagen zur FLOW-Methode sind aus diesem Werk entnommen [3].

### **Informationsflusstheorie der Softwareentwicklung**

In diesem Werk thematisiert Stapel die Informationsflusstheorie und ihrer Verbindung zur FLOW-Methode. Wichtige Grundlagen für die in dieser Arbeit entworfenen Erweiterungen der FLOW-Methode wurden aus dieses Werkes entnommen [7].

### **Applying Social Network Analysis and Centrality Measures to Improve Information Flow Analysis**

Kiesling et al. haben in diesem Paper grundlegende Analyseverfahren für das Arbeiten mit der FLOW-Methode geprüft. Hierzu wurden FLOW-Modelle dahingehend angepasst, dass Social-Network-Analysis (SNA) durchführbar sind. Dazu ist es notwendig, Activity-Knoten aus FLOW-Modellen aufzulösen. Einer der Schwerpunkte dieser Arbeit resultiert aus den Erkenntnissen der SNA mit Activities und den dabei auftretenden Besonderheiten [5].

### **Vergleich von Algorithmen zur Informationsflussanalyse in der Software-Entwicklung**

Obwohl die Bachelorarbeit von Wallat thematisch hauptsächlich die algorithmische Analyse für FLOW thematisiert, ist innerhalb dieser Arbeit ein Software-Analyse-Tool entstanden, auf dem die Implementation der erarbeiteten und diskutierten Konzepte aufbaut. Dafür wurde das von Wallat

erarbeitete XML-Schema und die Datenstruktur sowie die graphische Software aufgegriffen und erweitert [8].

### **Informationsfluss in verteilten Softwareprojekten – eine Einzelfallstudie**

Klünder und Schneider haben sich in dieser Einzelfallstudie mit der FLOW-Methode bei verteilten Softwareprojekten beschäftigt. Hierzu wurden zwei Projektteams in Spanien und Deutschland bei der Softwareentwicklung begleitet und ihre Informationsflüsse modelliert und analysiert [6].

## Kapitel 3

# Konzeptentwicklung

Wie aus der Problemstellung hervorgeht, müssen für die Informationsflussanalyse einerseits verständliche und übersichtliche Modelle zur Verfügung stehen, andererseits werden auch kleinere Details benötigt, um genauere Ergebnisse liefern zu können. Dafür sind in der FLOW-Methode hierarchische Strukturen vorgesehen, die allerdings nur theoretisch vorliegen, aber noch nicht praktisch nutzbar gemacht wurden. Aus dieser Motivation heraus werden Konzepte entwickelt, um eine Verfeinerung der Modelle zu erreichen und damit Informationsflussanalysen zu optimieren.

Nachfolgend werden das Konzept zur Visualisierung und das zur Verfeinerung ausgeführt. Hierbei werden mögliche Ansätze vorgestellt, auf deren Basis die Entwicklung von jeweils zwei Konzepten für die Visualisierung der hierarchischen Detailgrade und der dazu erforderlichen Verfeinerung der Datenstruktur erfolgen kann. Abschließend geht es um die möglichen Kombinationen des Konzepts der Verfeinerung und der Visualisierung.

### 3.1 Konzept zur Visualisierung

Beim Entwerfen von FLOW-Diagrammen wird standardmäßig immer mit groben Netzwerken begonnen, um dann nach und nach durch das Hinzufügen von neuen Kanten und weitere Knoten das Netzwerk zu detaillieren. Intuitiv wird ein grobes Netzwerk immer weiter verfeinert. Ab einem gewissen Grad kann dieses Modell aber schnell unübersichtlich werden. Der Grund dafür können zum einen zu viele Knoten und Kanten in dem Modell sein. Zum anderen kann die Unübersichtlichkeit auch an dem ursprünglichen Positionieren liegen, das nicht optimal ist. Das nachträgliche Umstrukturieren ist zeitaufwendig und dient der Übersichtlichkeit und dem Verständnis nur bedingt. Bei diesen Nacharbeiten werden auch Knoten

zusammengefasst oder gar wieder entfernt.

Die nachfolgend ausgeführten Konzepte dienen dem Ziel, hierarchische Detailgrade in FLOW-Diagrammen für eine Optimierung von Informationsflussanalysen zu implementieren. Für eine effiziente Analyse werden neben detaillierten auch abstrakte Modellen benötigt, um den Überblick beizubehalten und globale Zusammenhänge schneller zu erkennen.

Die zu diesem Zweck entwickelten Konzepte zur Visualisierung umfassen dabei mehr als nur die reine Darstellung von Informationen. Sie beschreiben ebenfalls mögliche Interaktionen und Vorgehensweisen des Benutzers mit den Detailgraden. Die Konzepte wurden auf der Grundlage der bekannten Top-down- und Bottom-up-Methoden entwickelt, da diese einerseits einfache und intuitive Vorgehensweisen hinsichtlich der zu entwickelnden hierarchischen Strukturen vorhalten und andererseits ein ähnliches Vorgehen bereits bei der FLOW-Analyse verfolgt wurde.

### **Bottom-up- und Top-down-Methoden**

Die Bottom-up- und Top-down-Methoden bieten grundlegende Herangehensweisen an ein Problem. Top-down ist eine „Methode, bei der man schrittweise von allgemeinen, umfassenden Strukturen zu immer spezielleren Details übergeht“ [12].

Bei der Bottom-up-Methode handelt es sich um das entgegengesetzte Vorgehen. Dies ist die „Methode, bei der man von speziellen Details ausgeht und schrittweise über immer umfassendere Strukturen die Gesamtstruktur eines Systems errichtet“ [13].

Im Zusammenhang mit der hierarchischen Darstellung von Informationen in FLOW-Modellen wird die abstrakteste Möglichkeit der Darstellung eines Netzwerkes als das Top bzw. Level-0 definiert. Eine solches FLOW-Diagramm dient dem Überblick und der Orientierung, nicht einer detaillierten FLOW-Analyse.

Der Bottom definiert dementsprechend die unterste hierarchische Stufe und entspricht der detailliertesten Darstellung eines FLOW-Diagramms. Alle Informationen, die für eine FLOW-Analyse benötigt werden, sind hier visualisiert. Auf dieser Ebene sind alle abstrakteren Knoten aufgelöst. Abhängig von der Anzahl der Knoten und Kanten und ihrer Anordnung kann dieses Netzwerk unübersichtlich werden.

### **Abstraktion von Knoten eines Netzwerkgraphen**

Sowohl für das Top-down- als auch das Bottom-up-Konzept können verschiedene Knoten eines Netzwerkgraphen zu einem abstrakterem zusammengefasst werden. Ein Beispiel für diese hierarchischen Granularitäten sind in Abbildung 3.1 skizziert. Hierbei werden alle drei Knoten des abstrakterem Netzwerk aufgelöst. Jeder der drei Knoten enthält exemplarisch

zwei detaillierte Knoten.

Der flüssige Informationsspeicher „Team\*“ umfasst in diesem Beispiel „Developer 1“ und „ Developer 2“. Die Activity „Development\*“ abstrahiert die Dokumente „Requirements“ und „Specification“. Das Dokument „Software\*“ beinhaltet die „Documentation“ und den „Code“. Knoten, die mit einen „\*“ im Titel gekennzeichnet sind zeigen, dass diese nur die abstraktere Darstellung von mehreren detaillierten Knoten sind. Diese Kennzeichnung wird im späteren Verlauf aufgegriffen.

Um für Analysezwecke mehr Informationen bereitzustellen, können die abstrakten Knoten durch die detaillierteren ausgetauscht werden. Hierbei wird auch vom Auflösen eines Knotens gesprochen. Mit Hilfe der detaillierten „Unter-Knoten“ wird dann das Netzwerk ausgebaut.

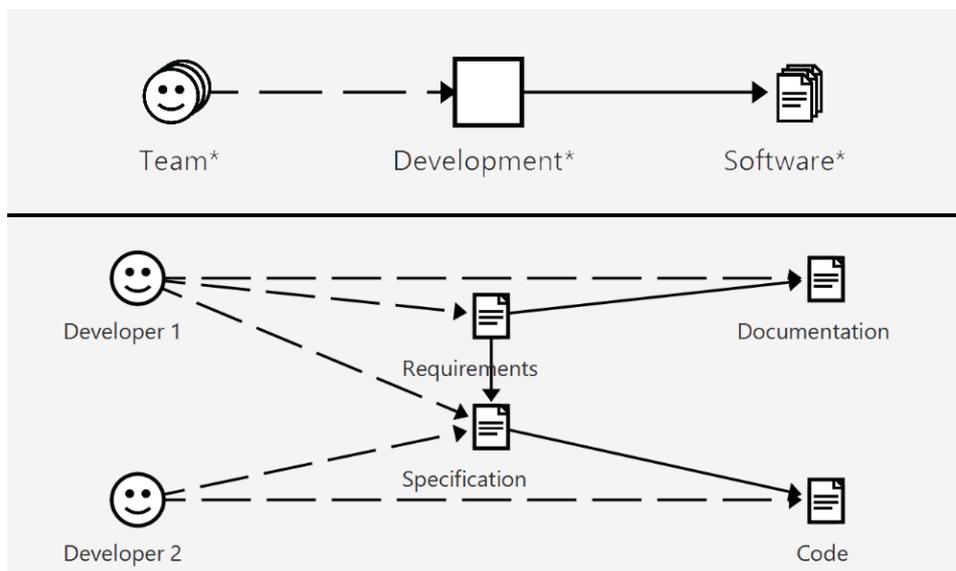


Abbildung 3.1: Beispiel von unterschiedlichen Granularitäten

### 3.1.1 Konzept nach der Top-down-Methode

Dieses Vorgehen wird - in ähnlicher Form - schon bei dem Entwurf von FLOW-Diagrammen genutzt. Das Netzwerk wird mit weiteren Knoten und Kanten immer detaillierter entworfen. Dies geschieht aber noch nicht mit hierarchischen Stufen, sondern nur auf einer Ebene.

Das Prinzip bei diesem Konzept sieht vor, dass bei der obersten abstraktesten Ebene, dem Top, gestartet wird und dann ein schrittweises Vorgehen in tiefere, detailliertere Stufen folgt.

Zu diesem Zweck kann ein neues Unternetzwerk bzw. Subnetzwerk an

einem Knoten aus dem abstrakten Hauptnetzwerk verlinkt werden. In diesem Subnetzwerk werden detailliertere Informationen zum Beispiel über den internen Prozess eines Knotens abgespeichert. Um dieses Beispiel weiter auszuführen, kann ein abstrakter Knoten als „Entwicklung“ einen abstrakten Schritt in einem Hauptnetzwerk repräsentieren. Im dem dazugehörigen Subnetzwerk werden dann die einzelnen Schritte der Entwicklung abgelegt. Um also mehr Details über einen Prozess zu erhalten, ist es im Anschluss möglich, zwischen den Netzwerken zu wechseln. So können mit dem Anlegen und Verknüpfen von Subnetzwerken mehr Details in tieferen Ebenen abgespeichert werden.

Ein beispielhaftes Vorgehen nach dem Top-down-Konzept wird in dem folgendem Mockup noch einmal aufgegriffen und graphisch dargestellt. Diese wurden mit dem Software-Tool von Wallat [8] erstellt.

### Mockup

Dieses Mockup zeigt exemplarisch und kompakt, wie ein Subnetzwerk nach dem Top-down-Konzept angelegt werden kann.

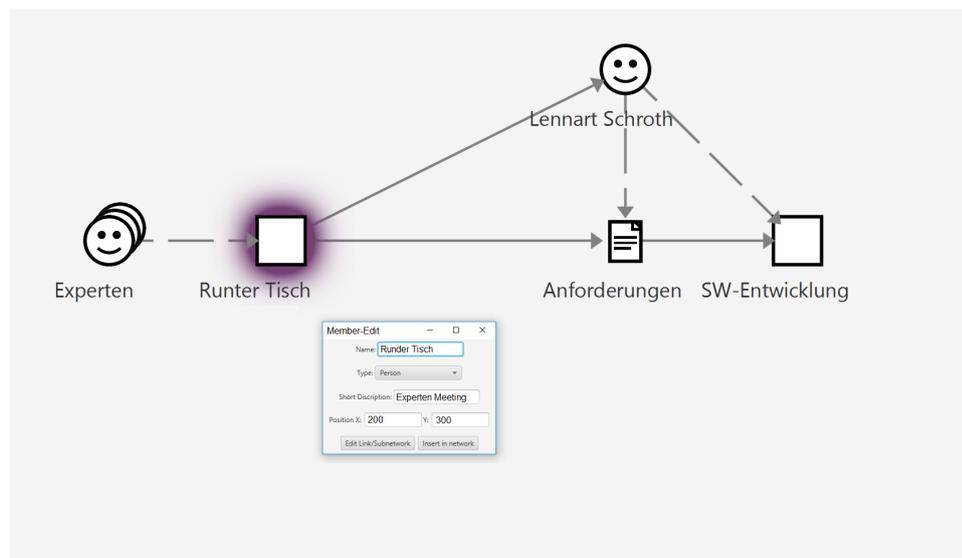


Abbildung 3.2: Top-down-Konzept - Mockup 1

Schritt 1: Durch den Klick auf einen Knoten bzw. ein Member im FLOW-Netzwerk wird das Fenster zum Bearbeiten geöffnet (Abb. 3.2).

Abbildung 3.3: Top-down-Konzept - Mockup 2

Schritt 2: Neben dem eigentlichen Bearbeiten der Attribute ist es möglich, ein neues Subnetzwerk zu verlinken. Mit „Insert“ kann ein verlinktes Subnetzwerk in das übergeordnete Netzwerk gemergt bzw. integriert werden. Der Begriff „Insert“ ist hier als Synonym für „Merging“ zu verstehen (Abb. 3.3).

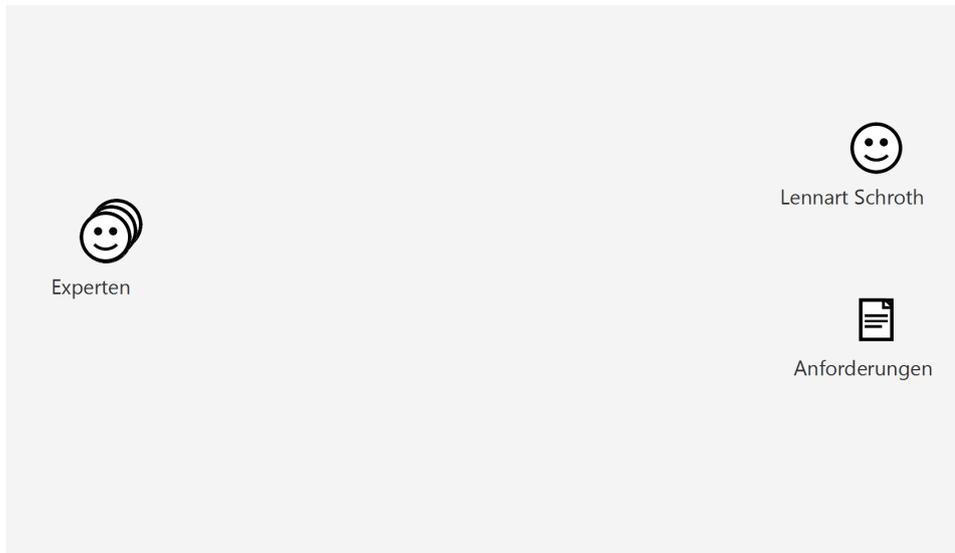


Abbildung 3.4: Top-down-Konzept - Mockup 3

Schritt 3: Das neu angelegte Netzwerk bekommt die über Kanten verbundenen Knoten mit übergeben (Abb. 3.4). Diese Nachbarn werden mit in das Subnetzwerk integriert, um Schnittstellen zum Hauptnetzwerk aufzuzeigen. Nun kann das Subnetzwerk mit weiteren Knoten und Kanten verfeinernd abgespeichert werden. Beim Mergen dieses editieren Subnetzwerkes werden die mit übernommenen Nachbarn erkannt, sodass das Subnetzwerk in

das Hauptnetzwerk besser integriert werden kann.

### **Vor- und Nachteile**

Ein großer Vorteil des Top-down-Konzepts ist die sich bietende Möglichkeit des intuitiven Vorgehens. Zudem bietet es die Möglichkeit, einen beliebigen Detailgrad zu erreichen, da nach „unten“ beliebig viele Netzwerke immer weiter verlinkt werden können. Das Anwenden des Prinzips „vom Groben zum Feinen“ ist in der Software-Entwicklung weit verbreitet.

Als nachteilig wäre die Datenverwaltung zu nennen. Viele einzelne Netzwerke, die alle miteinander verknüpft sind, können ab einem gewissen Maß zu einer hohen Unübersichtlichkeit führen. Abhängig von der Anzahl an hierarchischen Stufen wäre das Erreichen von tieferen Ebenen mit mehr Klicks oder „Scrollen“ verbunden.

### **3.1.2 Konzept nach der Bottom-up-Methode**

Die grundlegende Idee des Bottom-up-Konzept besteht darin, von den untersten detailliertesten Ebenen raus zu zoomen, um in verschiedenen Stufen das Netzwerk zu abstrahieren. Dabei soll das Gefühl entstehen, auf höhere hierarchische Ebenen zu kommen und das Netzwerk immer weiter zu abstrahieren.

Dafür können verschiedene Knoten gewählt werden, die ab einer einstellbaren hierarchischen Stufe (Level) zu einem abstrakteren Knoten zusammengefasst werden. Knoten können somit zu verschiedenen Gruppen abstrahiert werden.

Mehrere Personen als flüssige Informationsspeicher können beispielsweise zu einem Team zusammengelegt werden. Mehrere Teams könnten danach ab einer höheren hierarchischen Stufe zu einer Abteilung zusammengefasst werden. Wiederum können ab einem noch höheren Level dann Teams und Abteilungen zu einer Firma kombiniert werden. Dieses Szenario ist vereinfacht dem folgendem Mockup zu entnehmen.

Ebenfalls könnten feste Informationsspeicher wie Dokumente zu einer Dokumentengruppe zusammengefasst werden.

Gemischte Informationsspeicher können zu Activites kombiniert werden.

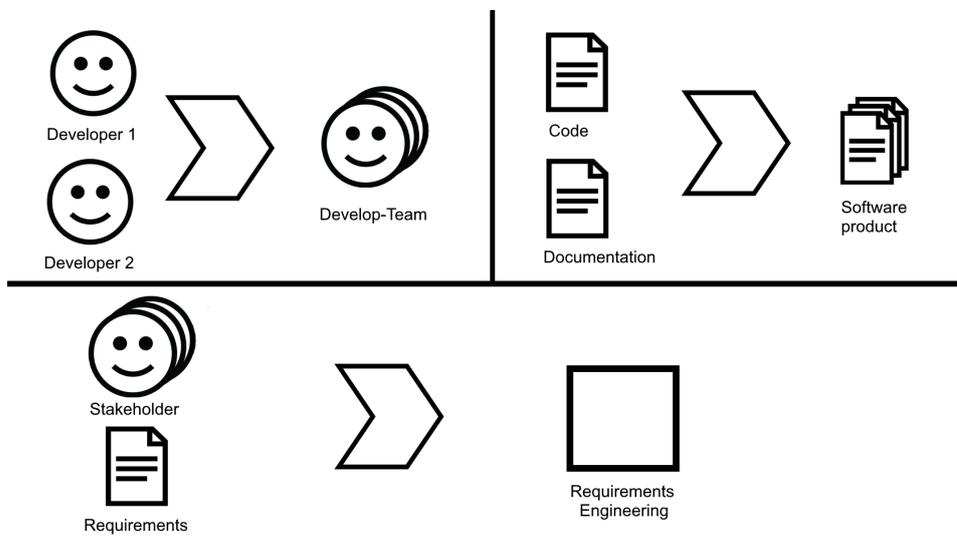


Abbildung 3.5: Abstraktion verschiedener Knoten

**Mockup**

Wie anfangs angedeutet, soll das folgende Mockup eine exemplarische Abstraktion von verschiedenen Personen graphisch aufarbeiten. Hierzu dient das vorgestellte FLOW-Diagramm aus dem Kapitel 2. In diesem Fall sollen verschiedene Mitglieder der Entwicklung zu einer abstrakteren Gruppe zusammengefasst werden.

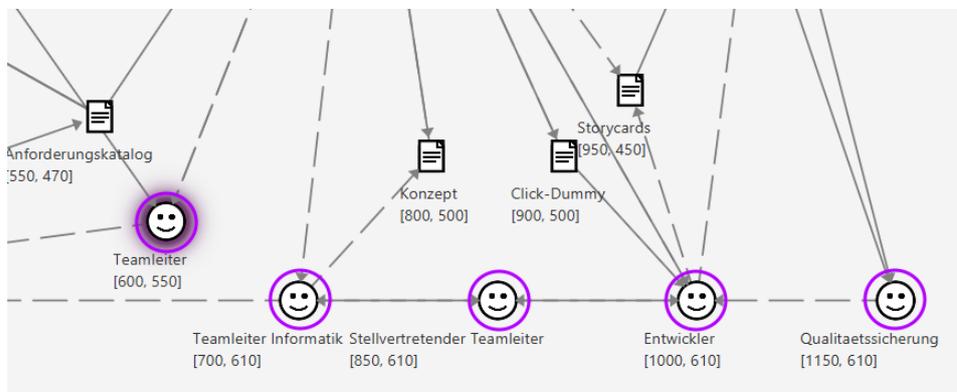


Abbildung 3.6: Bottom-up-Konzept - Mockup 1

Schritt 1: Verschiedene Knoten bzw. Members, die zusammengelegt werden sollen, werden ausgewählt.

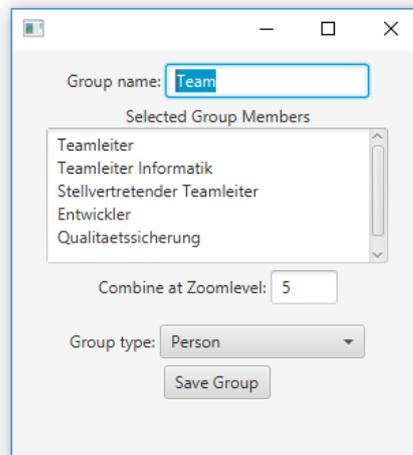


Abbildung 3.7: Bottom-up-Konzept - Mockup 2

Schritt 2: Ab einer einstellbaren Zoomstufe werden die ausgewählten Members zu einer Gruppe zusammengefasst. Der entstehende neue Knoten trägt den Namen der neuen Gruppierung. Abhängig von den Members in der Gruppe kann der abstraktere Knoten wieder vom Typ Person sein. Unterschiedliche Personen können zu einer Personengruppe und einzelne Dokumente zu einer Dokumentengruppe zusammengefasst werden. Bei flüssigen und festen Informationsspeichern können Personen und Dokumente zu einer Activity abstrahiert werden.

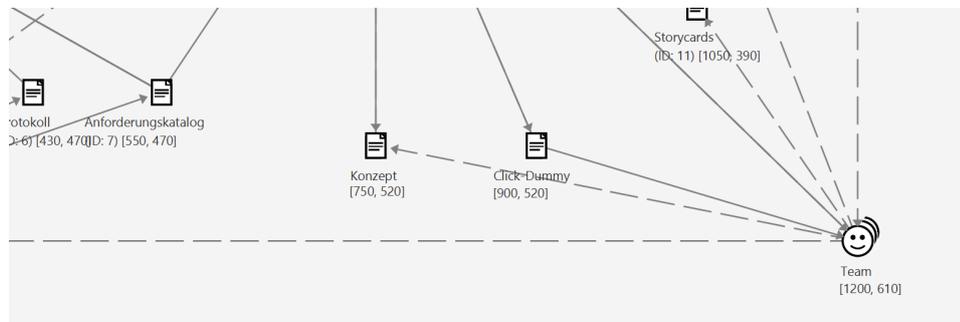


Abbildung 3.8: Bottom-up-Konzept - Mockup 3

Der neu entstandene Knoten enthält alle Kanten der enthaltenen Members.

### Vor- und Nachteile

Da dieses Konzept auf nur einem einzigem Diagramm aufbaut, das schrittweise abstrahiert werden kann, kann das Speichern in einer Datei erfolgen. Für das Verwalten oder Versenden eines FLOW-Diagramms muss nur ein Dokument betrachtet werden. Dieses große und detaillierte Netzwerk wäre zudem als Ausgangslage für eine softwarebasierte FLOW-Analyse

interessant, weil sich alle Informationen schon in dem Netzwerk befinden und keine weiteren Netzwerke hereingeladen werden müssten. Ein weiterer Vorteil ist das schnelle Ein- und Ausblenden von Informationen. Hier könnte das Scrollen mit einer Maus das schnelle Wechseln zwischen verschiedenen Stufen möglich machen.

Demgegenüber ist eher von großem Nachteil, dass zuerst ein solches detailliertes Netzwerk erstellt werden muss, damit es im Nachhinein abstrahiert werden kann. Dieses erste Aufbauen eines detaillierten FLOW-Diagrammes ist mit einem höheren Zeitaufwand verbunden. In einem großen Netzwerk ist es zudem aufwendiger, mehrere Knoten neu zu positionieren.

## 3.2 Konzepte zur Strukturverfeinerung

Die Konzepte der Strukturverfeinerung dienen der Weiterentwicklung der softwarebasierten Arbeit mit FLOW-Modellen, und zwar insbesondere für die zu erweiternde Datenstruktur sowie die neu zu entwickelnde hierarchische Visualisierung.

Grundlegend enthält jedes FLOW-Diagramm einen Netzwerkgraphen, der in einer XML-Datei gespeichert ist. Da mehr Informationen für FLOW-Diagramme bereitgestellt werden, diese aber noch nicht alle abgelegt werden können, muss die Datenstruktur erweitert und verfeinert werden. Bei der Erstellung der beiden im Folgenden vorgestellten Konzepte wurde deswegen darauf geachtet, dass ein gewisses Basismaß an Informationen zur Verfügung steht. Dafür werden unabhängig von dem implementierten Konzept weitere Parameter bzw. Eigenschaften der Datenstruktur hinzugefügt. Knoten (vertices/members) bekommen dafür den zusätzlichen Parameter „short-discription“, der einfache Zeichenketten als String-Datentyp abspeichert. Kanten (edges) bekommen die zusätzlichen Parameter „short-discription“, „weight“ und „channel-types“. Diese ergänzenden Eigenschaften werden in Kapitel 5 näher erläutert.

### 3.2.1 Strukturkonzept Link

Bei dem Link-Konzept wird den Knoten ein zusätzlicher Parameter hinzugefügt, der den Namen von anderen XML-Dateien abspeichern kann, um verschiedene Netzwerkgraphen miteinander zu verlinken. Der Begriff der Verlinkung oder Vernetzung wird hierbei als grundlegende Idee genutzt, um Dokumente miteinander zu verbinden und somit verschiedene Dateien untereinander und miteinander zu verwalten. „Link“ bezeichnet somit die

Verlinkung von XML-Dateien, um verschiedene Netzwerke miteinander zu verknüpfen.

Dieses Konzept ist vor allem in Verbindung mit dem Top-down-Konzept zur Visualisierung interessant. Wie in dem Kapitel „Konzept zur Visualisierung“ näher erläutert wird, handelt es sich bei dem TOP um die oberste abstrakteste Ebene, von der stufenweise heruntergegangen werden kann. Das Netzwerk, das das TOP darstellt, wird auch als Hauptnetzwerk (main-net oder level-0-net) bezeichnet. Ein Hauptnetzwerk kann verschiedene Subnetzwerke unter sich verlinken.

### **Vor- und Nachteile**

Die Verlinkung zeigt dabei verschiedene Vor- und Nachteile auf. Ein großer Vorteil ist die Verknüpfung von verschiedenen Dateiformaten. Dabei besteht nicht nur die Möglichkeit, Netzwerke aus XML-Dateien untereinander zu verknüpfen, sondern auch andere Formate zu verlinken. Dies könnten beispielsweise Dokumente, Kontaktinformationen oder Protokolle sein. Ebenfalls von Vorteil ist die Verwaltung einzelner Dokumente. Es ist dabei nicht notwendig, ein großes Netzwerk zu laden, um einfache Änderungen vorzunehmen. Es besteht die Möglichkeit, direkt in den Subnetzwerken zu arbeiten oder diese gesondert zu betrachten.

Neben diesen Vorteilen gibt es auch Nachteile, die bedacht werden müssen. Die dezentrale Verwaltung vieler Dokumente könnte nicht nur Vorteile bringen, sondern könnte auch schnell zu Unübersichtlichkeit und Verwirrung führen. Ebenfalls ein größerer Nachteil wäre das Teilen oder Versenden von Netzwerken beispielsweise mit anderen FLOW-Analysten. Hierfür müssen alle einzelnen Dokumente verschickt werden, da sonst ein ausgewähltes Subnetzwerk nicht betrachtet werden kann.

### **3.2.2 Strukturkonzept All-in**

Dieses Konzept sieht das Bündeln aller Informationen rund um das FLOW-Modell in einer Datei vor. Dabei werden alle Informationen über das Netzwerk mit allen hierarchischen Eigenschaften in einer XML-Datei abgespeichert. Dieses Konzept wird im Folgenden als „All-in“ bezeichnet.

### **Vor- und Nachteile**

Auch dieses Konzept weist sowohl Vor- als auch Nachteile auf. Ein großer Vorteil besteht in dem Arbeiten mit dem System, denn es gibt nur eine XML-Datei, in der sich alle Informationen befinden. Dieses ermöglicht einen schnellen Austausch von FLOW-Diagrammen mit anderen und vereinfacht so die Datenverwaltung und trägt zur Übersichtlichkeit bei.

Diese zentrale Verwaltung lässt es aber nicht zu, externe Daten zu koppeln. Alle Informationen, die in einer XML-Datei verwaltet werden, müssen dementsprechend auch immer mit dem gleichen Software-Tool (hier dem FLOW-Editor) bearbeitet werden.

### 3.3 Mögliche kombinatorische Realisierungen

Aus den zwei vorgestellten Konzepten zur Visualisierung und den beiden zur Strukturverfeinerung lassen sich nun vier mögliche Kombinationen realisieren. Die Prinzipien und Ideen hinter den möglichen Kombinationen der Konzepte werden mit dem Ziel, zunächst einen Überblick zu geben, in einer kompakten Matrix zusammengefasst (siehe Abb. 3.9). Anschließend werden die vier Kombinationen einzeln erläutert.

Konzept	LINK	ALL-IN
Top-down	Externe abgespeicherte Subnetze werden untereinander verlinken	Alle Subnetze befinden sich in einer XML-Datei
Bottom-up	Ein Netzwerk und verschiedene Gruppierungen in verlinkten Dateien	Ein Netzwerk und alle Gruppierungen in XML-Datei abspeichern

Abbildung 3.9: Konzeptkombinationen

#### Link- und Top-down-Konzept

Wie schon erwähnt, ist die Verbindung des Link-Konzeptes mit dem Top-down-Konzept eine sinnvolle Kombination. Da das Link-Konzept eine Verlinkung repräsentiert und das Top-down-Konzept Subnetzwerke zur Informationsspeicherung vorsieht, wäre die logische Kombination, die Verlinkung von verschiedenen Subnetzwerken. Hierzu kann ein Knoten einen Verweis auf ein bestehendes oder zu erstellendes Subnetzwerk erhalten.

Für dieses Konzept wäre es notwendig, den Kanten den zusätzlichen Parameter „link“ hinzuzufügen. Dabei handelt es sich um eine Zeichenkette vom Datentyp „string“, der den Namen der verlinkten Datei enthält. Falls keine Datei verlinkt werden soll, wird diese Eigenschaft einfach leer bleiben.

#### Link- und Bottom-up-Konzept

Als weitere Möglichkeit ist an dieser Stelle die Kombination zwischen dem Link- und dem Bottom-up-Konzept angeführt. Die Grundidee bei Bottom-up ist das Abstrahieren von mehreren Knoten zu einem abstrakteren Knoten. Diese Kombination ist aber weniger sinnvoll, weil die dabei entstehende

Gruppierung eine so kleine Einheit darstellen würde, dass die externe Verwaltung zu aufwendig wäre.

#### **All-in- und Top-down-Konzept**

Für die Realisierung dieses Konzeptes mit dem Top-down-Konzept muss die Struktur dahingehend erweitert werden, dass das Ablegen von beliebig vielen FLOW-Diagrammen in einer XML-Datei möglich ist. Diese Menge an FLOW-Diagrammen würde alle Haupt- und Unternetzwerke enthalten. Jedes dieser Netzwerke würde eine eindeutige ID erhalten. Hierbei wären dann Subnetzwerke nicht über externe Links zu verknüpfen, sondern über einen internen Verweis mit den einzelnen ID's. Für die interne Datenstruktur könnten darauf aufbauend Bäume oder Listen zum Verwalten genutzt werden.

#### **All-in- und Bottom-up-Konzept**

Da Bottom-up auf der Grundlage eines Netzwerkgraphen besteht und nur die verschiedenen Gruppen zur Abstraktion verwaltet werden müssen, würde sich das Bündeln aller Daten in einer XML-Datei bei dieser Konzeptkombination anbieten.

Diesbezüglich wäre es notwendig, dem Knoten noch ein weiteres Attribut zuzuweisen. Die Identifikationsnummer für eine Gruppierung oder auch „group-id“ würde hierfür notwendig sein. Bei der Erstellung einer Gruppe mit einer neuen ID könnten die dazugehörigen Knoten die ID's als Parameter übergeben bekommen. Nach dem Vereinigen der Knoten in einer Gruppe würden alle ein- und ausgehenden Kanten nur noch von diesem Knoten ausgehen.

## Kapitel 4

# Konzeptentscheidung

Vor dem Hintergrund, Informationsflussanalysen durch eine Verfeinerung von Modellen zu optimieren, wurden die oben vorgestellten Konzepte zur Implementation von hierarchischen Detailgraden diskutiert. Die Entscheidung für ein Konzept, das in das bereits existierende FLOW-Tool implementiert wurde, basierte auf dieser Diskussion.

Dieses Kapitel beinhaltet die Schritte, die zur Konzeptentscheidung geführt haben, sowie die Darstellung des Ergebnisses und dessen Interpretation.

### 4.1 Vorbereitungen zur Entscheidungsfindung

Für die Konzeptentscheidung wurden vier Mitarbeiter vom Fachgebiet Software Engineering des Instituts für Praktische Informatik zu einer „Expertenrunde“ bzw. „Konzeptrunde“ eingeladen. Die Auswahl erfolgte dabei auf der Grundlage der fachlichen Expertise und der Erfahrungen mit der bereits existierenden FLOW-Methode.

Geplant wurde diese Konzeptrunde wie ein Kick-Off-Meeting. Dazu wurden Ziele und Tipps für die Vorbereitungen und die Durchführung aus dem Werk „IT-Projektmanagement“ entnommen [11].

Im ersten Schritt wird den Experten ein Überblick über die entwickelten Konzepte gegeben. Anschließend werden die Vor- und Nachteile aufgezeigt und mit den Experten diskutiert. Abschließend steht die eigentliche Entscheidung aus. Für einen offen und unverbindlichen Austausch von Ideen und Wünsche, wird im Anschluss noch ca. 20 Minuten eingeplant.

Für eine einheitliche Diskussionsgrundlage wurde ein Ausdruck mit allen Konzepten vorbereitet. Dieser stellte die Vor- und Nachteile, kurze Zusammenfassungen und die Mockups graphisch dar. Darüber hinaus

wurden alle wichtigen Aspekten in einer Power-Point-Präsentation (PPP) aufbereitet. Da es zwei verschiedene Konzepte zur Visualisierung und zwei zur Strukturierung gab, wurde eine Entscheidungsmatrix vorbereitet, die alle vier möglichen Kombinationen nochmal zusammenfasste.

## 4.2 Durchführung der Konzeptentscheidung

Bei einem ersten Kick-Off-Meeting wurde der Konzeptrunde zunächst das Thema der Bachelorarbeit vorgestellt und der aktuelle Stand der Bearbeitung kurz skizziert, damit alle Beteiligten auf einem Wissensstand sind. Zudem wurde die Problemstellung und die Motivation zur Implementation der hierarchischen Detailstufen erläutert.

Nach dieser Einleitungsphase folgte die eigentliche Vorstellung aller vier vorbereiteten Konzepte mittels des oben erwähnten Handouts und der PPP.

Für die endgültige Festlegung der Konzepte, die im Rahmen dieser Bachelorarbeit in ein Software-Tool zu implementieren waren, wurde den Experten die im Folgenden dargestellte Entscheidungsmatrix (Abb. 4.1) mit den vier möglichen Konzept-Kombinationen vorgelegt. Ihr sind die Idee hinter der jeweiligen Kombination der strukturellen und visuellen Konzepte sowie die zentralen Vor- und Nachteile zu entnehmen.

Konzept	Link	All-in
Top-down	Externe Subnetze verlinken + Beliebige hierarchische Tiefe - viele einzelne Dokumente	Alle Subnetze in einer XML-Datei + Alle Subnetze in einer XML-Datei - Keine externen Dateien (z.B. PDF)
Bottom-up	Gruppierungen extern abspeichern + wenn die Datei fehlt, kann man neue Gruppierungen anlegen - Ergibt wenig Sinn Gruppierungen extern auszulagern	Gruppierungen in einer XML-Datei abspeichern + Alle Informationen in einer XML-Datei + „Zoomen“ mit Scrollrad zu realisieren - Ein detailliertes Netzwerk anlegen

Abbildung 4.1: Entscheidungsmatrix

## 4.3 Ergebnisse

Während der Diskussion der Konzepte kristallisierte sich früh heraus, dass das Vorgehen nach dem Top-down-Prinzip den FLOW-Analyse-Verfahren sehr nahe kommt. Da häufig zuerst grobe FLOW-Diagramme entworfen werden, die erst im Nachhinein weiter verfeinert werden, bietet sich hier die Visualisierung mit dem Top-down-Konzept an.

Obwohl das Verfahren nach dem Bottom-up-Konzept mit der Gruppierung von Knoten ebenfalls Zuspruch fand, wurde sich für das Top-down-Konzept entschieden. Das entscheidende Argument für dieses Konzept war die sich bietende Möglichkeit, einen detaillierten Netzwerkgraphen zu erstellen, um diesen im Nachhinein mit hierarchischen Detailgraden zu erweitern.

Die Festlegung der dahinter stehenden Struktur wurde aufbauend auf dieser Entscheidung getroffen. Dabei sprachen sich alle Experten für das Link-Konzept aus, das sowohl mit dem Top-down-Prinzip gut realisierbar ist, als auch interessante Erweiterungen in der Zukunft ermöglicht. An dieser Stelle ist beispielsweise das zukünftige Verlinken von anderen Datenformaten zu erwähnen. Da das Link-Konzept auch den Nachteil mit sich bringt, dass mehrere Dateien verwaltet werden müssen, wird an dieser Stelle auf die Erweiterbarkeit verwiesen.

## 4.4 Ergebnis-Interpretation

Auf Grundlage der Diskussion mit den Experten sind die Anforderungen erstellt worden. Diese lassen sich in zwei Kategorien unterteilen, und zwar in grundlegende Anforderungen, die das Konzept betreffen, und in spezielle Anforderungen, die sich auf das System beziehen.

Die Konzeptanforderungen finden sich teilweise auch in denen die Implementation betreffend wieder, werden im Folgenden aber als die relevantesten Anforderungen gebündelt aufgeführt. Die Implementationsanforderungen sind Gegenstand von Kapitel 5.1.

Bei der Umsetzung des Top-down-Konzeptes ist es unabdingbar, dass es möglich ist, Knoten durch die Verlinkung von Subnetzwerken mehr Details zur Verfügung zu stellen. Daraus lassen sich folgende drei notwendigen Funktionen ableiten: Erstens das Wechseln oder auch Hineinzoomen in diese Subnetzwerke und zweitens das Bearbeiten und Speichern dieser. Des Weiteren muss auch eine Funktion realisiert werden, die das Integrieren der Subnetzwerke in das Hauptnetzwerk ermöglicht. Diese Merge-Funktion muss alle Eigenschaften des Subnetzwerkes in das Hauptnetzwerk übernehmen können.

Hintergrund dieser Anforderungen ist das Ersetzen von abstrakten Knoten durch ihre Subnetzwerke, um eine FLOW-Analyse zu optimieren. Das Ersetzen eines abstrakten Knoten durch ein detailliertes Subnetzwerk soll auch dazu genutzt werden können, um das Auflösen von Activities zu realisieren. Eine ergänzende Merge-All-Funktion muss zudem die Erweiterung bringen, alle Subnetzwerke, die sich eine Hierarchiestufe unter dem Hauptnetz befinden, in dieses zu mergen.

# Kapitel 5

## Konzeptimplementierung

In diesem Kapitel wird die Implementation der entwickelten und diskutierten Konzepte behandelt. Dazu werden zunächst die Anforderungen an die zu implementierenden Funktionen der Software erläutert. Anschließend wird die Umsetzung skizziert, wobei der Fokus auf den erstellten und implementierten Konzepten liegt. Das Testen der implementierten Funktionen wird ausgeführt, bevor abschließend auf mögliche Erweiterungen hingewiesen wird.

### 5.1 Anforderungen an die Software

Die erforderlichen Anforderungen an die Implementierung der Funktionen in die bereits existierende FLOW-Software wurden formuliert unter Zuhilfenahme des Werkes „Basiswissen Requirements Engineering“ [10].

Nach der Diskussion über und der Entscheidung für die Konzepte wurde abgewogen, welche Ideen und Anforderungen der Experten in die Systemanforderungen übernommen werden und damit in dieser Arbeit umgesetzt werden. Hierbei wurden allgemeine Anforderungen, die während der Konzeptrunde formuliert wurden, auf die zu entwickelnden Softwarefunktionen zugeschnitten. Zusätzlich zu dieser Spezialisierung kamen weitere Implementationsanforderungen dazu.

Alle weiteren Punkte, die als Ideen von den Experten vorgeschlagen wurden, aber den Umfang dieser Bachelorarbeit überschritten hätten, werden am Ende dieses Kapitels vorgestellt. Aufgrund der zügigen Implementation konnten aber einige dieser Ideen dennoch mit umgesetzt werden. Diese werden unter Top-down-Visualisierung weiter erläutert. Die Funktionen, die als zukünftige Erweiterungen noch in Frage kommen, werden bei Erweiterbarkeit aufgeführt.

Zu einigen Anforderung gibt es ein paar kommentierende Worte und kurze Erläuterungen.

**[R01]** Der Nutzer muss jeder Activity/jedem Member ein Subnetzwerk (subnet) hinzufügen (verlinken) können.

Hierbei geht es um die eigentliche Realisierung des Top-down-Konzeptes. Dafür ist das Erstellen und Verknüpfen von Subnetzwerken eine notwendige Funktionalität.

**[R02]** Subnetze sollen als extra XML gespeichert werden können.

Das Speichern von bearbeiteten Netzwerken ist ebenfalls eine Funktion, die für das Arbeiten mit FLOW-Diagrammen notwendig ist und bisher noch nicht implementiert wurde.

**[R03]** Der Nutzer soll ein Subnetz in das übergeordnete Netzwerk laden/mergen können.

Um detaillierte Netzwerke zu realisieren, müssen Subnetzwerke aus tieferen auch in höhere Ebenen geladen werden können. Dies ist für eine schnelle Erstellung von detaillierten Netzwerken notwendig.

**[R04]** Ein durch die Merge-Funktion zusammengebautes Netz muss der Nutzer extra abspeichern können.

Auch gemergete Netzwerke sollen abgespeichert werden. Dabei ist zu beachten, dass dabei nicht unbedingt das ursprüngliche Hauptnetzwerk überschrieben wird.

**[R05]** Eine Merge-Funktion soll alle Subnetze in ein Hauptnetz integrieren. Die Merge-All-Funktion soll es ermöglichen, alle Subnetzwerke eines Hauptnetzes, also die komplette untere Ebene, in das Hauptnetzwerk zu laden, damit nicht alle Netzwerke einzeln gemerget werden müssen. Über die Notwendigkeit dieser Funktion gibt es einen kurzen Auszug in dem Kapitel Interpretation der Ergebnisse.

**[R06]** Der Nutzer muss Knoten bzw. Member eine Kurzbeschreibung (short description) hinzufügen können.

Diese Anforderung wurde als Möglichkeit aufgegriffen, um kurze Kommentare und Bemerkungen den jeweiligen Kanten hinzufügen zu können. Hierbei geht es um das Hinzufügen von beliebigen Informationen, die weniger die FLOW-Analyse unterstützen, als vielmehr das Verständnis verbessern sollen.

**[R07]** Der Nutzer soll die wichtigen Attribute aller Knoten bzw. Member in der Software editieren können.

Obwohl die Funktion des graphischen Editierens nicht der zentralen Aufgabenstellung dieser Bachelorarbeit entspricht, wurde doch die Notwendigkeit für diese neue Funktion als notwendig erachtet und somit im Laufe dieser Arbeit mit in die Anforderungen aufgenommen.

**[R08]** Der Nutzer muss Kanten einem Kanal-Typ (channel-typ) z.B. Chat oder E-Mail, zuordnen können.

Das Arbeiten mit verschiedenen Kanal-Medien ist erst in den letzten Jahren in den Blickpunkt gerückt und sollte in dieser Bachelorarbeit mit aufgenommen werden [9]. Die Zuordnung zu einem Kanal-Medium sollen bei der FLOW-Analyse weitere Kenntnisse liefern können.

**[R09]** Der Nutzer muss Kanten eine Kurzbeschreibung (short description) hinzufügen können.

Neben den Knoten sollen auch Kanten zusätzliche Details erhalten können, um den Informationsfluss besser überblicken zu können und somit ein tieferes Verständnis der Zusammenhänge zu bekommen.

**[R10]** Der Nutzer muss einer Kante ein Gewicht (weight) zuordnen können. Das Kantengewicht wird derzeit noch beim graphischen Erstellen des FLOW-Diagrammes berechnet. Über den neuen Parameter soll es möglich sein, einen persönlichen Wert zu speichern.

**[R11]** Der Nutzer soll die wichtigen Attribute aller Kanten in der Software editieren können.

Wie bei den Knoten sollen auch relevante Parameter in der Software graphisch verändert werden können, um ein benutzerfreundliches Arbeiten zu ermöglichen.

## 5.2 Softwareseitige Umsetzung

Im Folgenden wird erläutert, wie die Visualisierung der Granularitäten nach dem Top-down-Konzept implementiert wurden. Für die Verfeinerung der Datenstruktur, d.h. zur Vernetzung von weiteren, externen XML-Dateien, diente das vorgestellte Konzept Link. Dazu wurde das existierende Analyse-Tool von Wallat [8] dahingehend erweitert, dass es nicht nur Netzwerke laden und analysieren, sondern auch editieren und speichern kann. Die entscheidenden Funktionen beinhalten aber das Anlegen, Bearbeiten und Mergen von Subnetzwerken, sowie das stufenweise Wechseln zwischen den einzelnen Graphen.

Die Implementierung gliederte sich dabei in mehrere größere Themenbereiche, die nachfolgend skizziert werden. Nach einer kurzen Phase der Einarbeitung in die Entwicklungsumgebung und die vorhandenen Strukturen wurde die Erweiterung der Datenstruktur realisiert. Dafür wurde das benötigte XML-Schema zum Speichern der Informationen erweitert. Im Anschluss wurden grundlegende Funktionen zum Erstellen, Editieren und Speichern von Subnetzwerken implementiert. Dafür wurden einige graphische Neuerungen geschaffen. Danach wurde die Merge-Funktion

erstellt. Sie wurde mehrfach während der Bachelorarbeit verfeinert und verbessert. Aufbauend auf diesen Funktionalitäten wurde die „Merge-All“-Funktion implementiert, die es ermöglicht, alle Subnetzwerke eine Ebene unter dem aktuell betrachteten Netzwerk in dieses zu integrieren. Da für einige Funktionen schon graphische Oberflächen und Interaktionsmöglichkeiten erstellt wurden, wurden anschließend alle weiteren Funktionen für einen FLOW-Editor umgesetzt. Aufgrund der zügigen Implementation der gestellten Anforderungen wurden während der Entwicklungszeit noch weitere Ideen umgesetzt.

Die Umsetzung der Top-down- und Link-Konzepte sowie die aller zusätzlich zu implementierenden Funktionen erstreckte sich nach der Konzeptentscheidung etwa über zwei Monate.

### **Vorbereitungen**

Die Konzepte wurden mit der Programmiersprache Java implementiert. Dazu diente die Entwicklungsumgebung Eclipse Java Photon (Stand 2018). Da das vorhandene Software-Tool mit JavaFx arbeitete, wurde auf dieser Grundlage weiter entwickelt.

Schon während der Einarbeitung in die Entwicklungsumgebung sowie in das Software-Tool wurde die Notwendigkeit eines Editors offensichtlich. Das ursprüngliche Erstellen von FLOW-Diagrammen beschränkte sich auf das manuelle, textbasierte Editieren der XML-Datei mit einfachen Texteditoren. Dies war sehr unpraktisch und zeitaufwendig. Da die Einarbeitung in die Software-Entwicklung etwa parallel zur Erstellung der Konzepte erfolgte, wurden darüber hinausgehende Funktionen zum Editieren der Kanten und Knoten im FLOW-Diagramm entwickelt.

## **5.2.1 Link-Strukturverfeinerung**

Wie in Kapitel 3 schon angerissen wurde, wird der Netzwerkgraph eines FLOW-Diagramms in einer XML-Datei abgespeichert. Um mehr Informationen für das FLOW-Modell und die darauf folgende Analyse bereitzustellen, muss die Datenstruktur erweitert werden. Als grundlegende Erweiterungen sind die folgenden Eigenschaften für Knoten und Kanten im Netzwerk dazugekommen.

### **Kantenerweiterung**

Alle Kanten erhalten für die Strukturverfeinerung die nachfolgenden vier Attribute.

1. Die Kanten (edges) werden ebenfalls um ein Parameter „short-discription“ erweitert. Dies ist dafür gedacht, schnell kurze Informationen hinzuzufügen. Beispielsweise kann auf eine bestimmte Mail oder ein

Meeting hingewiesen werden.

2. Mit „weight“ soll es möglich sein, den Kanten ein explizites Gewicht zuzuweisen. Normalerweise wird dieses durch die Software-basierte Analyse errechnet, soll aber auch benutzerbasiert zugewiesen werden können.

3. Das metaphorische Medium des Kommunikationskanals („channeltyp“) wurde von den Experten als wichtige Zusatzinformation während der Konzepterarbeitung und -entscheidung erachtet. Dabei wird zwischen fünf verschiedenen Kanalarten unterschieden, die sich in der Software-Entwicklung als sinnvoll herauskristallisiert haben [9].

Es kann zwischen den Medien E-Mail, Face-to-Face, Chat, Telefon und Video unterschieden werden. Ursprünglich war vorgesehen, diese Parameter in die Struktur mit aufzunehmen, um sie bei einer späteren Arbeit zu realisieren. Aufgegriffen wurden diese neuen Parameter mit der Visualisierung des Mediums durch Icon's (kleine Bilder) an den einzelnen Kanten.

4. Wie dem Kapitel 2.1 zu entnehmen ist, stellt der Austausch von Erfahrungen (engl. experience) in der FLOW-Methode einen besonderen Stellenwert dar. Mit einem Boolean-Datentyp wird in der erweiterten Datenstruktur abgespeichert, ob eine Kante bzw. ein Informationsfluss einen Erfahrungscharakter aufweist. Diese werden nach der FLOW-Notation, anders als normale schwarze Kanten, in grau dargestellt.

Das folgende Netzwerk visualisiert beispielhaft die erläuterten neuen Parameter für Kanten (siehe Abb. 5.1).

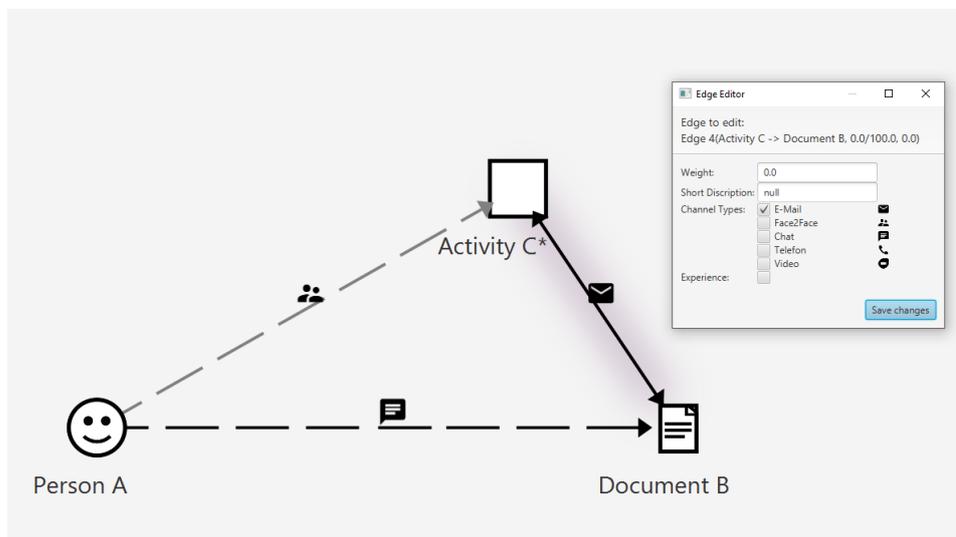


Abbildung 5.1: Beispiel eines FLOW-Diagramms mit erweiterten Kanteigenschaften

## Knotenerweiterung

Die Erweiterung der Struktur bezüglich der Knoten sieht die Ergänzung von zwei Parametern vor.

1. Alle Knoten (vertices/members) erhalten den Parameter „short-discription“ der durch einfache Zeichenketten als String-Datentyp realisiert wird. Ähnlich wie bei den Kanten soll es hiermit möglich sein, kurze Anmerkungen und Gedanken mit abzuspeichern.

2. Die entscheidende Erweiterung des XML-Schemas war das Arbeiten mit Links bei Knoten. Dazu erhielten alle Knoten neben der Kurzbeschreibung auch den neuen Link-Parameter. Der Link ist vom Datentyp ein „String“ und enthält den Namen der XML-Datei, in der das Subnetzwerk abgelegt werden kann. Es ist auch möglich, andere Dokumente dort zu hinterlegen, allerdings werden diese durch den Parser noch nicht beachtet, sodass nur XML-Dateien verwaltet werden können. Das Erweitern des Parsers ist einer der wichtigsten Punkte, der in dem Absatz Erweiterbarkeit aufgegriffen wird.

## XML-Schema und Datenstruktur

Die Abbildung 5.2 zeigt exemplarisch ein Schema in einer XML-Datei, auf das sich im weiteren Verlauf bezogen wird, um das XML-Schema kurz zu skizzieren.

```
<?xml version="1.0" encoding="UTF-8"?>
<BigFLOW>
  <members>
    <member name="Person A" id="1" x="520" y="280" type="person" disc="Beispiel Person A" link=""></member>
    <member name="Document B" id="2" x="810" y="280" type="document" disc="Dokument B" link=""></member>
    <member name="Activity C" id="3" x="730" y="160" type="activity" disc="Aktivitaet C" link="Subnet - Activity C.xml"></member>
  </members>
  <edges>
    <edge from="Person A" from_id="1" to="Document B" to_id="2" id="1" disc="null" weight="0.0" experi="false" ch_Type_0="Chat"></edge>
    <edge from="Person A" from_id="1" to="Activity C" to_id="3" id="2" disc="null" weight="0.0" experi="true" ch_Type_0="Face2Face"></edge>
    <edge from="Document B" from_id="2" to="Activity C" to_id="3" id="3" weight="0.0" experi="false"></edge>
    <edge from="Activity C" from_id="3" to="Document B" to_id="2" id="4" disc="null" weight="0.0" experi="false" ch_Type_0="E-Mail"></edge>
  </edges>
</BigFLOW>
```

Abbildung 5.2: Beispielhaftes Schema in einer XML-Datei

Der von „BigFLOW“ umschlossene Bereich enthält zwei Auflistungen.

Unter dem Absatz „members“ befindet sich eine Liste der Knoten des Netzwerkgraphen. Nach der Erweiterung der Datenstruktur enthalten diese die Parameter short-discription (kurz disc) und link.

Der Absatz „edges“ definiert die Kanten im zu generierenden Netzwerkgraphen. Die Position der anzuzeigenden Icons zur Darstellung des Kanal-Mediums (channel-typ) wird genau wie die Kante selber immer neu berechnet. Die Icons werden dabei mittig zwischen dem Start- und Endknoten (from- and to-vertex) angezeigt. Ebenfalls sind als Parameter die short-

discription (kurz disc), weight und experiance (kurz experi) dem Beispielschema zu entnehmen.

Das aus der XML-Datei (siehe Abb. 5.2) generierte FLOW-Diagramm wird in Abb. 5.3 dargestellt.

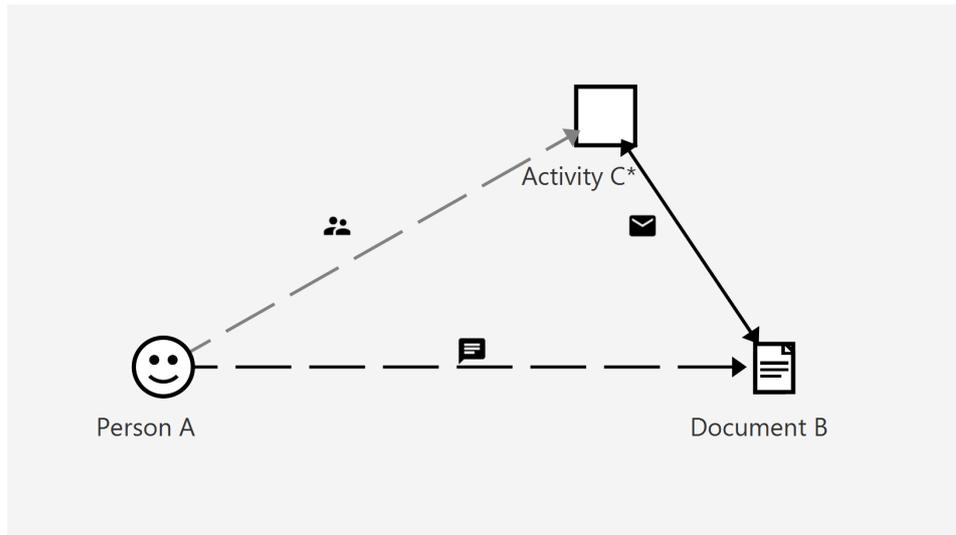


Abbildung 5.3: Generiertes FLOW-Diagramm aus dem XML-Schema

## 5.2.2 Top-down-Visualisierung

Das Hauptaugenmerk bei der Realisierung des Top-down-Konzeptes liegt auf der Verlinkung und auf den Netzwerken. Um einen stufen-basierten, höheren Grad an Details zu realisieren, wurden Funktionen erstellt, die das Hinzufügen und Editieren von verschiedenen Unter- bzw. Subnetzwerken für ein Hauptnetzwerk ermöglichen.

### Verwaltung von Subnetzen

Das Erstellen oder Editieren von Subnetzwerken erfolgt über das graphische Bearbeitungsfenster, das mit einem Rechts-Klick oder einem Doppel-Klick auf den gewünschten Knoten geöffnet wird. Falls bei einem Knoten schon ein weiteres Netzwerk hinterlegt ist, ist die Zeile des Link-Parameters bereits mit dem Namen der Datei gefüllt und an dem Titel des Knotens ist im Diagramm ein „\*“ angehängt. In Abb. 5.4 ist ein Beispielnetzwerk abgebildet, bei dem ein Subnetzwerk im Knoten „Activity C“ hinterlegt werden soll.

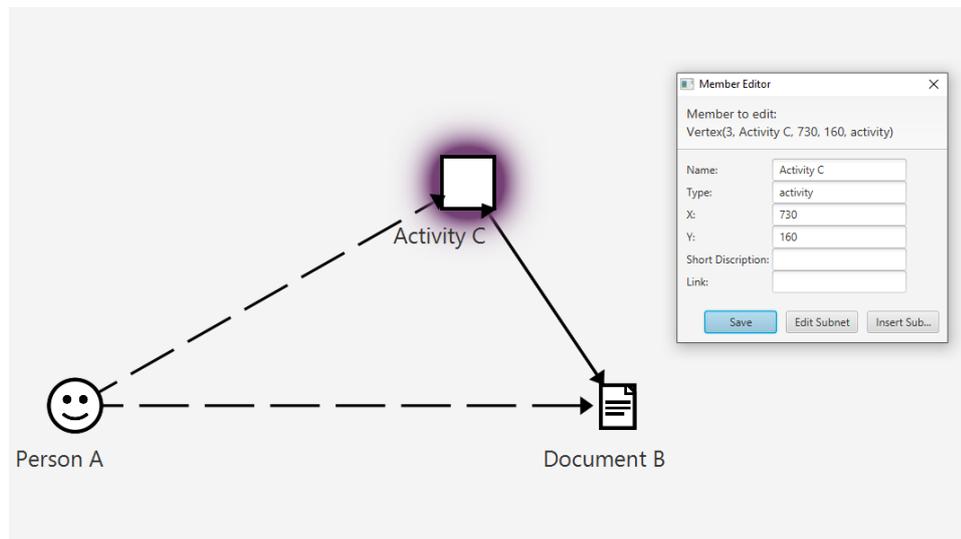


Abbildung 5.4: Bearbeitungsfenster eines ausgewählten Knotens mit dem „edit subnet“-Button

Nach dem Drücken des „edit Subnet“-Button wird das verlinkte Subnetzwerk geöffnet, sodass das Subnetzwerk direkt angesehen oder editiert werden kann. Falls noch kein Subnetzwerk verknüpft ist, wird ein neues erstellt. Dazu werden die ein- und ausgehenden Knoten des Hauptdiagramms mit in das neue Netzwerk übernommen. Alle Eingänge werden auf der linken Seite von oben nach unten aufgelistet und alle Ausgänge auf der rechten Seite. Abbildung 5.5 visualisiert das neu generierte Subnetzwerk.

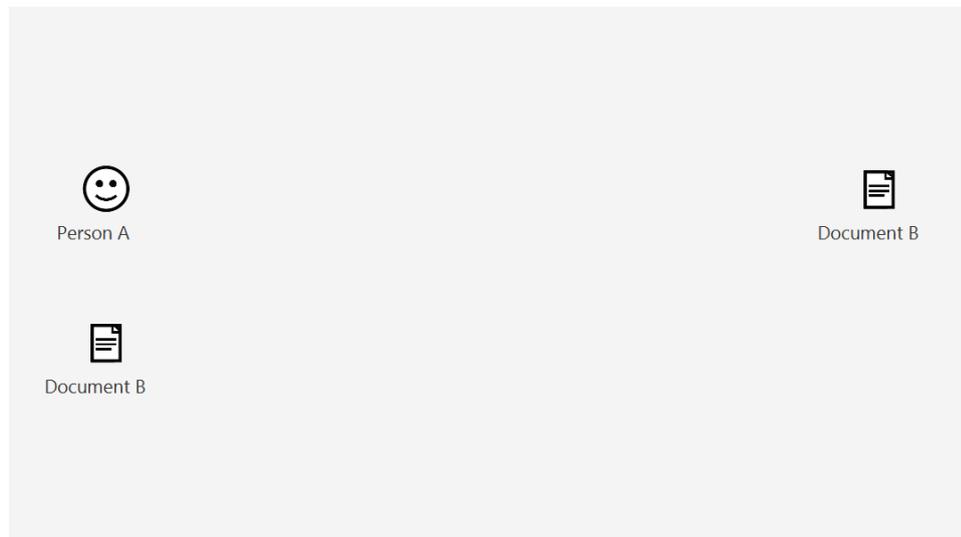


Abbildung 5.5: Erstellung eines neuen Subnetzwerkes mit den ursprünglichen Ein- (links) und Ausgängen (rechts) des übergeordneten Knotens

Durch das Hinzufügen von weiteren Knoten und Kanten kann das Subnetzwerk verfeinert werden (siehe Abb. 5.6).

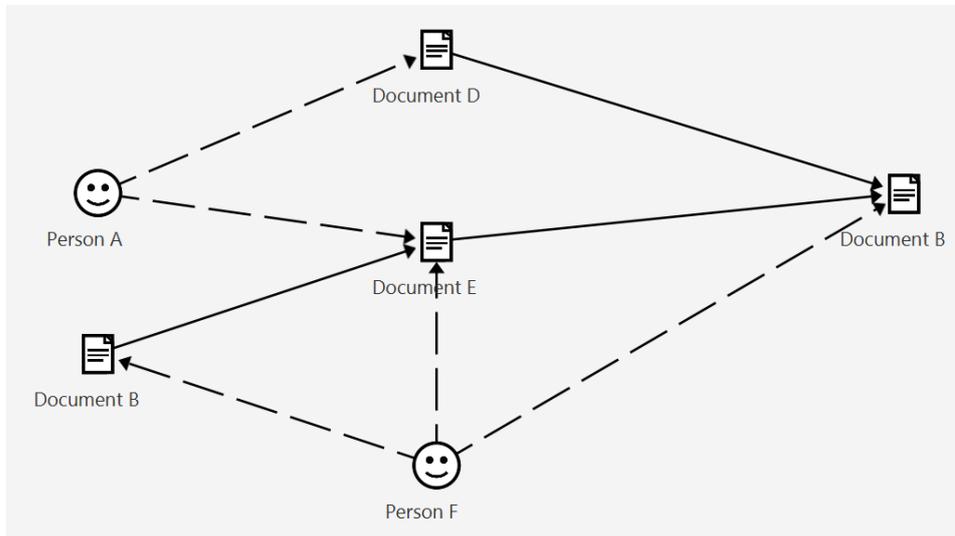


Abbildung 5.6: Anlegen von weiteren Knoten und Details im Subnetzwerk

Nach dem Speichern der Veränderungen im angelegtem Subnetzwerk wird im Hauptnetzwerk die verlinkte Datei angezeigt. Für einen besseren Überblick haben alle Knoten mit einem verlinkten Netzwerk ein „\*“ am Ende des Namens. Der Name des neuen Dokumentes kann im Link-Parameter betrachtet und ggf. bearbeitet werden (siehe Abb. 5.7).

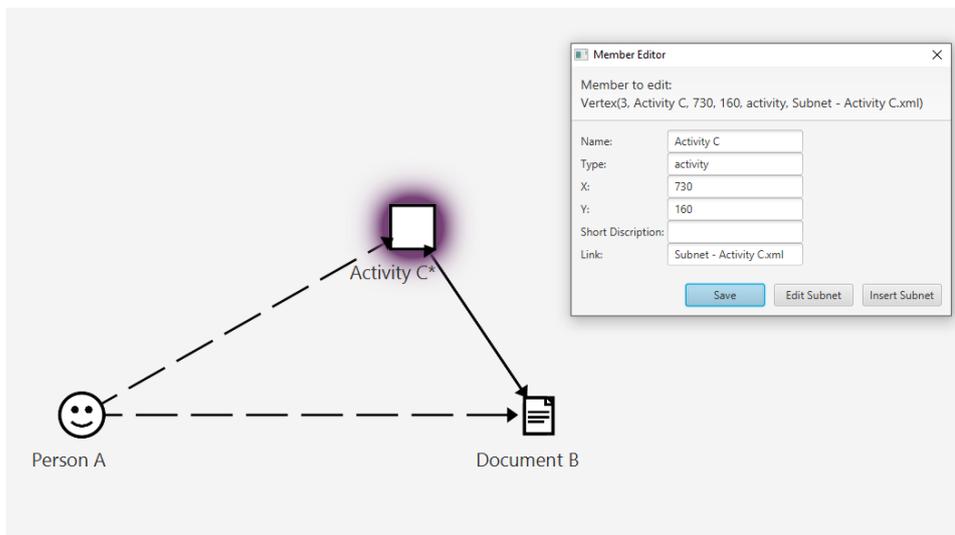


Abbildung 5.7: Hauptnetzwerk mit der verlinkten Datei

Dieses Anlegen und Verwalten von Subnetzwerken kann theoretisch beliebig tief erfolgen. Das Wechseln eines Subnetzwerken kann durch eine

Veränderung des Link-Parameters auch manuell vorgenommen werden. Der Top-down-Ansatz ist folglich nur vom TOP, in diesem Fall dem Hauptnetzwerk, abhängig. Es ist zu bedenken, dass die Übersichtlichkeit eines Netzwerkes ab einem gewissen Grad an Subnetzwerken leidet. Vor diesem Hintergrund werden in Kapitel 6 „Interpretation der Ergebnisse“ Empfehlungen gegeben, wie das Arbeiten mit diesen Hierarchiestufen erfolgen sollte.

### **Merging von Subnetzen**

Die „Merge“- bzw. „Insert“-Funktion ist eine weitere der zentralen Funktionen, die in dieser Arbeit entwickelt wurde. Auf den Begriff des „Merging“ ist bereits in Kapitel 2 „Stand der Entwicklung“ eingegangen worden.

Für das Mergen von einem Haupt- und einem Subnetzwerk werden einerseits beide Netze selbst, als auch der Knoten benötigt, der aufgelöst werden soll. Dafür wird im ersten Schritt berechnet, wie groß das Subnetzwerk in horizontaler x-Richtung und in vertikaler y-Richtung ist. Abhängig davon wird das Hauptnetzwerk um den Knoten herum gestreckt, damit das Subnetzwerk in die entstandene Lücke eingefügt werden kann. Alle Knoten rechts von dem ausgewählten Knoten werden dabei abhängig von der Subnetzgröße nach rechts und alle unter dem Knoten weiter nach unten verschoben. Der Subnetzwerk-Knoten wird aus dem Hauptnetzwerk dementsprechend gelöscht.

Danach wird berechnet, welche Knoten aus dem Subnetzwerk nicht in dem Hauptnetzwerk liegen. Die Knoten, die namentlich nicht in dem Hauptnetzwerk existieren, werden dann abhängig von ihrer Position im Subnetzwerk in das Hauptnetzwerk übernommen. Alle Eigenschaften bis auf die ID und die X-Y-Position werden dabei übernommen.

An der Position des ursprünglichen Knotens ist ein neuer leerer Platz entstanden, wo das Subnetzwerk hineinkopiert wird. Die Kanten werden danach aus dem Subnetzwerk in das Hauptnetzwerk eingefügt. Dabei wird die ursprüngliche Anordnung der Knoten im Subnetzwerk beibehalten.

Abschließend werden alle Kanten, die in dem Subnetzwerk abgelegt wurden, in das Hauptnetzwerk mit übernommen. Da alle Knoten neue ID's zugewiesen bekommen haben, wird dieses auch über den Vergleich der Namen realisiert.

Aufgrund der neuen ID's muss beachtet werden, dass das Mergen der Knoten abhängig von den Namen der Knoten ist. Wenn also Änderungen an dem Namen der standardmäßig eingefügten Knoten vorgenommen wurden, werden diese als neue Knoten kopiert und die gewünschten Kanten zu dem ursprünglichen Hauptnetzwerk können nicht berechnet werden.

Die Abbildung 5.8 zeigt das Hauptnetzwerk mit dem verlinkten Subnetzwerk

„unter“ dem Knoten „Activity C“. Durch einen Klick auf „Insert Subnet“ wird das am Knoten verlinkte Subnetzwerk in das aktuelle (Haupt-)Netzwerk geladen.

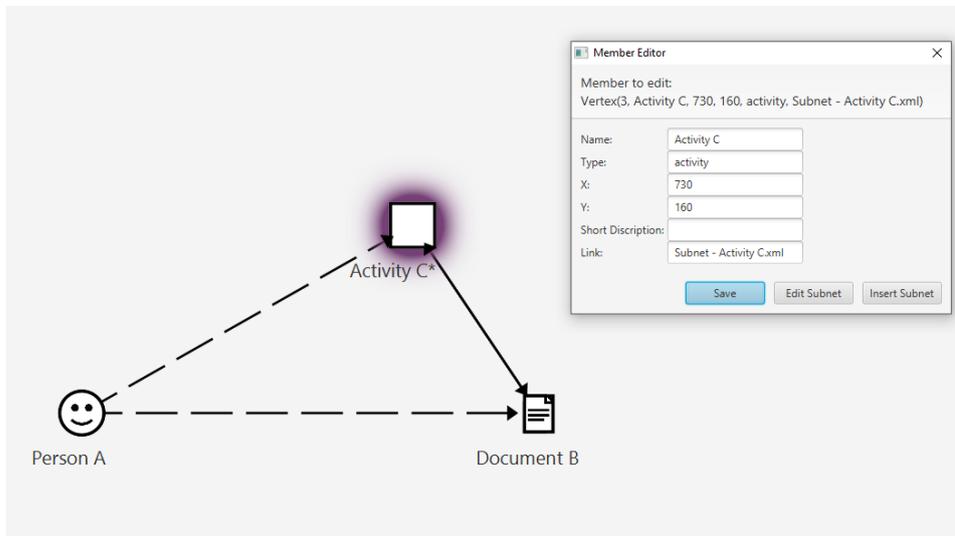


Abbildung 5.8: Beispielnetzwerk mit einem verlinkten Subnetzwerk im Knoten „Activity C“

Nach dem Merging enthält das betrachtete FLOW-Diagramm nun das Subnetzwerk mit den Knoten und Kanten (siehe Abb. 5.9).

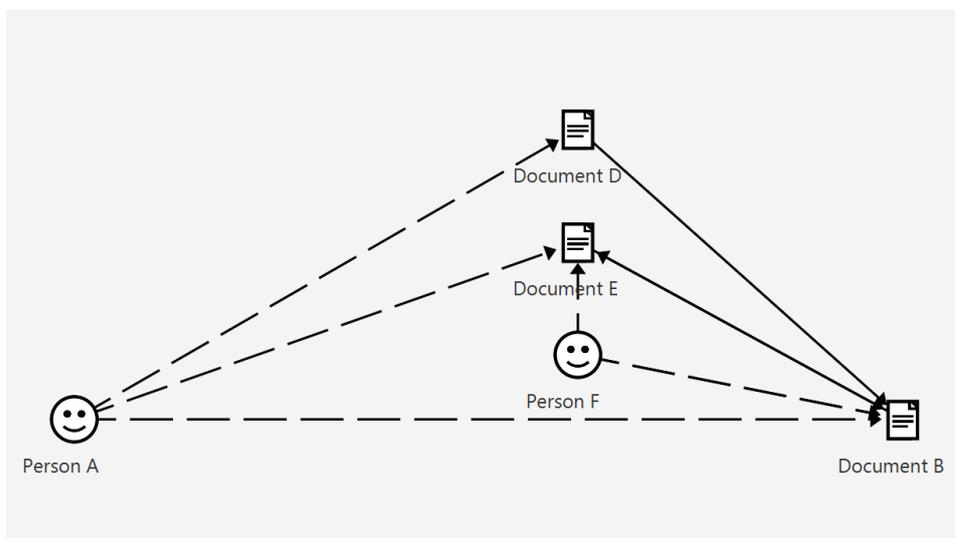


Abbildung 5.9: Beispielnetzwerk mit dem integrierten Subnetzwerk  
**Hierarchiebaum**

Als ergänzende Funktion wurde ein Hierarchiebaum (engl. hierarchie-tree)

implementiert. Über den Reiter „Netzwerk“ wird dieser in einem extra Fenster geöffnet. Die Funktion des Hierarchiebaums durchsucht dafür alle XML-Dateien im aktuellen Dokumentenverzeichnis. Aus allen XML-Dateien wird eine Liste mit FLOW-Diagrammen generiert. Mit Hilfe dieser Liste werden die Abhängigkeiten untereinander analysiert. Alle Netzwerke, die in keinem Knoten eines Netzwerkes verlinkt sind und somit von keinem Netzwerk ein Subnetzwerk sein können, werden als „Level-0“ eingestuft. Sie stellen alle ein eigenes Hauptnetzwerk dar und können Subnetzwerke enthalten. Ausgehend von den Level-0-Netzwerken wird die hierarchische Struktur aller Subnetzwerke berechnet.

### **Graphischer Editor**

Als zusätzliche Implementation wurde das Software-Tool um die Funktionen eines graphischen Editors erweitert. Diese ursprünglich nicht im Rahmen dieser Bachelorarbeit geplanten Funktionalitäten stellte sich aber zu Beginn der Entwicklungsphase als notwendig heraus, sodass sie ebenfalls mit umgesetzt wurden. Parallel zur Einarbeitungszeit und den Vorbereitungen zur Konzeptimplementation entstanden Funktionen, die das einfache Editieren von Knoten und Kanten in dem Netzwerk ermöglichen. Diese wurden während der Entwicklung dann um alle relevanten Attribute erweitert. Zwei einfache Fenster zur direkten Bearbeitung sowie ein Reiter in dem Menü auf der rechten Seite der Software ermöglichten dann die graphische Interaktion.

### **Ergänzende Implementationen**

Die im Folgenden aufgeführten Punkte haben sich im Rahmen der Diskussion während der Konzeptentscheidung ergeben. Diese Anliegen wurden allerdings nicht in die Systemanforderungen aufgenommen, weil die ersten Schätzungen davon ausgingen, dass deren Umsetzung zeitlich zu umfangreich sind. Die folgenden Ideen wurden gegen Ende der Arbeit aber trotzdem mit aufgegriffen und teilweise in abgewandelter Form implementiert.

1. Bei mehreren Übertragungen zwischen zwei Mitgliedern (z.B. bei Nutzung von E-Mail und Chat) sollen mehrere Pfeile diesen parallelen Austausch kenntlich machen. Hierzu wäre es denkbar, dass bei mehreren Pfeilen diese parabelförmig auf einen Knoten zugehen.

Diese Idee wurde zwar nicht direkt realisiert, aber eine andere Alternative hat diese Idee aufgegriffen. Für die Darstellung von verschiedenen Übertragungskanälen wurden nicht verschiedene Pfeile genutzt. An die vorhandenen Kanten wurden verschiedene Icons gesetzt, die die verschiedenen Kanäle kenntlich machen. Da die Art des Informationskanals schon mit implementiert wurde, galt es nur noch, eine geeignete Darstellung zu wählen.

2. Das Öffnen von Subnetzten soll nicht nur per Button, sondern auch beim Hineinzoomen mit dem Scrollrad möglich sein.

Die Funktion des „Zoomens“ zwischen Hierarchiestufen konnte noch mit aufgenommen werden, weil grundlegende Funktionen für das Wechseln zwischen XML-Dateien zuvor implementiert wurden. Hierzu wurde während der Entwicklung ein Verlauf erstellt, der speichert, welches Diagramm wann geladen wurde. Diese einfache Liste an Dateinamen diene zum Vorwärts- und Zurückspringen zwischen einzelnen Dokumenten. Wenn also bis zu einem gewissen Grad an einen ausgewählten Knoten herangezoozt wurde, wurde die XML-Datei im abgelegten Link geöffnet. Falls kein Link vorhanden war, wurde ein neues Subnetzwerk mit den Ein- und Ausgangsknoten des ausgewählten Members angelegt. Somit wurde das Gefühl erzeugt, in einen Knoten hineinzuzoomen. Wie jedes geöffnete Netzwerk wurde auch dieses dem Verlauf hinzugefügt. Wenn nun aus dem neu geöffneten Netzwerk wieder mit dem Scrollrad herausgezoozt wurde, dann wurde einfach das Vorgängernetzwerk geöffnet, das einfach dem Verlauf zu entnehmen ist. Die Funktion dahinter ist die gleiche wie bei dem „Pre Net“-Button in der Menüleiste unter dem Reiter „Netzwerk“. Durch das Öffnen des Vorgängers entsteht das Gefühl, dass in die höhere, abstrahierte Ebene herausgezoozt wurde.

3. Kanten, die als „Erfahrung“- oder „Support“-Idee nach der FLOW-Notation in einen Knoten reingehen, sollen sich farblich unterscheiden

Diese Idee wurde als letzter Wunsch ebenfalls nicht mit in die Anforderungen aufgenommen, aber trotzdem noch mit umgesetzt. Dazu wurde jeder Kante der zusätzliche Parameter „experiance“ als Boolean-Datentyp hinzugefügt. Wie alle wichtigen Eigenschaften kann auch diese in einem aufklappbaren Fenster bearbeitet werden.

## 5.3 Softwaretest

Zum Testen der entwickelten Funktionen wurden sowohl bereits vorhandene Graphen genutzt als auch neue hinzugefügt. Dafür wurde die vorhandene Importfunktion dahingehend ergänzt, dass auch alte Graphen mit dem einfachen XML-Schema importiert werden können. Beim Speichern wurde diese dann dem neuen Schema angepasst.

### Komponenten- und Integrationstest

Nach jeder Implementation einer der Hauptfunktionen, wie dem Merging oder dem Wechseln von Netzwerken, wurden diese Funktionen mehrfach auf ihre korrekten Ergebnisse getestet. Im Anschluss wurden die Schnittstellen zu den weiteren Funktionen überprüft. Dies geschah auch

mit unterschiedlichen Testgeräten, um auszuschließen, dass langsame Computer nicht zu viel Zeit benötigen.

### **System- und Abnahmetest**

Die Abnahme der implementierten Funktionen erfolge auch in Zusammenarbeit mit den Experten der Konzeptrunde und wird in Kapitel „Interpretation der Ergebnisse“ weiter ausgeführt.

## **5.4 Erweiterbarkeit**

Die folgenden Ideen konnten aus Zeitgründen nicht mit umgesetzt werden. Diese könnten in späteren Arbeiten mit aufgenommen werden.

1. Bei zyklischen Vorgängen soll es möglich sein, dass eine Animation dies visualisiert.
2. Durch die Verlinkung von externen Dokumenten wurde zudem die Möglichkeit geschaffen, in naher Zukunft auch andere externe Dokumente dem Projekt anzuhängen. Dazu können beispielsweise bei Dokument-Knoten externe PDF-Dateien verlinkt werden. Bei Personen würde dementsprechend die Möglichkeit bestehen, Kontakt-Profile mit Adressen zu hinterlegen.
3. Um das Verwalten der verschiedenen XML-Dateien der Subnetze und des Hauptnetzwerkes noch weiter zu vereinfachen, sollte die Überlegung vorgenommen werden, ob eine Art „Workspace“ bzw. „Arbeitsmappe“ um ein FLOW-Modell realisierbar ist. Einer der in den Konzepten aufgelisteten Nachteile zu dem Top-down-Konzept war das Teilen oder Versenden von Netzwerken mit Subnetzwerken. Diesbezüglich wäre es sicher praxisgerechter, eine Verwaltung mit Workspaces nachträglich zu implementieren und dann beispielsweise diese in eine ZIP-Datei umzuwandeln, um sie als ein Paket teilen zu können.
4. Bei einer solchen Erweiterung kann ebenfalls überlegt werden, ob das hierarchische Strukturieren mit Ordnern parallel zur virtuellen Struktur der Netzwerke nicht sinnvoll wäre. Das Verwalten von allen Haupt- und Subnetzwerken in einem Arbeitsordner sorgten gegen Ende der Arbeit mit 20 Haupt- und Subnetzen für eine gewisse Unübersichtlichkeit.

# Kapitel 6

## Interpretation der Ergebnisse

Auf der Grundlage von zwei vorgegebenen Szenarien wurde die implementierten Konzepte angewendet und evaluiert. Nachfolgend wird zunächst beschrieben, mit welchem Instrumentarium das Feedback eingeholt wurde. Danach erfolgt die Auswertung der Rückmeldungen, indem diese interpretiert und diskutiert werden. Abschließend werden die FLOW-Analyse betreffend Empfehlungen gegeben.

### 6.1 Planung

Um ein möglichst umfassendes Feedback einzuholen, wurde eine Kombination aus dem exemplarischen Testen mit einem vorgegebenen Anwendungsszenario und einem nachfolgenden Interview gewählt, um sowohl die strukturellen Vorteile eines Tests zu nutzen, als auch das umfangreiche Feedback mittels eines Interviews zu erhalten. Zu diesem Zweck wurden die Experten der Konzeptrunde für die Diskussion zu verschiedenen Terminen eingeladen. Anders als bei der Konzeptrunde absolvierten die Experten das Szenario mit anschließendem Feedbackbogen und einem angeschlossenen Interview alleine.

### 6.2 Durchführung

#### 1. Anwendungsszenario

Die Anwendung der neu entwickelten Funktionen wurde beispielhaft an einer kurzen FLOW-Analyse durch die Experten getestet. Dazu wurden die Teilnehmer aufgeteilt und mit jeweils einem von zwei Anwendungsszenarien konfrontiert. Ausgehend von demselben Hauptnetzwerk mit vier

Subnetzwerken. Die erste Gruppe bekam das Netzwerk mit im Vorhinein gemergtem Subnetzwerken (siehe Anhang 2). Da die zweite Gruppe ohne Merge-Funktion die Analyse durchführen sollten, bekamen deren Probanden nur das FLOW-Diagramm mit den vorbereiteten Subnetzwerken (siehe Anhang 3).

## **2. Interview und Diskussion**

Als Überleitung von dieser testweisen Anwendung zum Interview wurde den Probanden ein Feedbackbogen vorgelegt. Dieser beinhaltete sieben Fragen, die sich in drei Themenbereiche aufteilten (siehe Anhang 1). Die ersten vier Fragen behandelten die Übersichtlichkeit, Verständlichkeit, das Erreichen von Befunden und das Maß an Befunden. Diese bezogen sich auf die absolvierte FLOW-Analyse des Anwendungsszenarios. Beide Gruppen sollten durch die FLOW-Analyse besser abschätzen können, ab welchem Maß Informationen hilfreicher und ab welchem hinderlicher sind.

Der zweite Themenbereich umfasste den Nutzen und die Verwendung von Hierarchiestufen in einem FLOW-Diagramm. Das Szenario sollte auch dabei helfen, abschätzen zu können, wie viele Hierarchiestufen eine FLOW-Analyse ungefähr benötigt.

Im letzten Bereich wurde die Nutzung der implementierten Icons für die Darstellung des Informationskanals abgefragt.

Das abschließende Interview mit den Probanden wurde mit kurzen Fragen zu dem Feedbackbogen und zu der durchgeführten FLOW-Analyse eingeleitet. Diese Fragen sollten zu kleineren Diskussionen bzw. einem offenen und unverbindlichen Austausch anregen und waren deshalb bewusst allgemein formuliert.

Das Ziel dieses gemischten Feedbacks war es nicht, ein „genaues Ergebnis“ zu bekommen. Es sollte stattdessen gewährleisten, dass es zu einem umfangreichen Feedback und einem intensiven Informationsaustausch kommt. Beides wäre durch einen Test oder ein Experiment kaum zu realisieren gewesen.

## **6.3 Auswertung und Ergebnisse**

### **Auswertung des Feedbackbogens**

Wie Anfangs schon erwähnt, wurden die Probanden als Einstieg mit einem von zwei kurzen Szenarien konfrontiert. Die ersten beiden Probanden aus der ersten Gruppe mit aufgelösten Subnetzwerken benötigten für die FLOW-Analyse zehn und zwölf Minuten und lieferten in dieser Zeit acht und 13 Befunde.

Die zweite Gruppe bekam, nach einer kurzen Einweisung in das Arbeiten mit dem Rein- und Rauszoomen zwischen den hierarchischen Ebenen, das gleiche FLOW-Diagramm aber ohne die gemergten Subnetze. Beide Probanden benötigten 15 und 18 Minuten und lieferten sechs und zehn Befunde.

Hieraus lässt sich schon die erste Erkenntnis gewinnen und zwar, dass nach einer kurzen Einarbeitungszeit von 2-3 Minuten für den Umgang mit verschiedenen Ebenen die Experten schon in der Lage waren, ohne Probleme die FLOW-Analyse durchzuführen.

Zusammen mit der Auswertung der ausgefüllten Feedback-Bögen konnten weitere Ergebnisse ermittelt werden

#### 1. Übersichtlichkeit - War das Diagramm übersichtlich?

Die erste Gruppe empfand die Übersichtlichkeit des Netzwerkes als „gut“ bis „sehr gut“. Sie fanden sich schneller zurecht als die zweite Gruppe und begannen schneller mit der Analyse und Erstellung der Befunde. Die beiden anderen Probanden empfanden die Übersichtlichkeit ebenfalls als „gut“, benötigten aber anfangs die erwähnten 2-3 Minuten länger, um mit den vier verlinkten Subnetzwerken und dem Wechseln zwischen den Ebenen zurechtzukommen.

#### 2. Verständlichkeit - War das Diagramm verständlich?

Die Mitglieder der ersten Gruppe vermerkten bei der Verständlichkeit alle ein „sehr gut“. Die zweite Gruppe gab ein einheitliches „gut“ an. Daraus ist zu folgern, dass nach einer kurzen Einarbeitungszeit der Umgang mit den neuen hierarchischen Detailgraden in der FLOW-Analyse ohne Probleme möglich ist.

#### 3. Befunde - Wie einfach war das Erkennen von Befunden?

Obwohl beiden Teams die gleichen Informationen nur jeweils in einer anderen Anordnung zur Verfügung standen, konnte die erste Gruppe aufgrund der bereits gemergten Netzwerke schneller Befunde feststellen. Die Befunde an sich unterschieden sich bei den einzelnen Experten nur leicht. Die entscheidenden Aspekte wurden von allen schnell entdeckt. Fehlanalysen durch die Granularitäten sind in diesem Szenario nicht aufgetreten.

#### 4. Informationen - Wie viele Details benötigt ein FLOW-Diagramm?

Die zweite Gruppe empfand die Anzahl an Informationen in dem abstrakteren Netzwerk als passend. Etwas weniger Details haben sich die Teilnehmer der ersten Gruppe gewünscht, da sie ein paar Knoten eines Subnetzwerkes als unnötig empfanden. Daraus lässt sich schließen, dass es vielleicht nicht immer sinnvoll ist, alle Subnetzwerke in ein Hauptnetzwerk zu mergen.

Die beiden Fragen zu den Hierarchiestufen wurden verständlicherweise

nur von der zweiten Gruppe beantwortet, da die erste die Hierarchiestufen während des Szenarios nur aufgelöst gesehen haben. Ihnen wurden im Anschluss an den Feedbackbogen aber noch die Unterlagen der zweiten Gruppe präsentiert, um auch mit ihnen über den Wechsel und das Mergen von Netzwerken zu diskutieren.

5. Hierarchiestufen 1 - War der Einsatz von Hierarchien hilfreich, um Befunde festzustellen?

Alle Experten waren sich darin einig, dass die neu entwickelten Granularitäten gut zum Feststellen der Befunde sind und die Analyse insgesamt damit zu beschleunigen und zu optimieren ist.

6. Hierarchiestufen 2 - Sollten mehr oder weniger Hierarchiestufen verwendet werden?

Auch hier kamen alle Experten zu einem eindeutigen Ergebnis. Die drei im Szenario verwendeten Ebenen (Level-0, Level-1, Level-2) verschlechterten nicht die Übersichtlichkeit und Verständlichkeit, aber in der späteren Anwendung sollte die Anzahl an Ebenen beschränkt werden. Wie in der Empfehlung weiter ausgeführt wird, sollten nur ein Sublevel bzw. maximal zwei bei FLOW-Analysen verwendet werden.

7. Informationskanäle - War die Beschriftung der Kanten (Icons für E-Mail, Chat etc.) hilfreich?

Auch hier waren sich die Experten einig, dass diese Darstellung hilfreich bis sehr hilfreich für eine FLOW-Analyse ist.

### **Ergebnisse des Interviews**

Für den Start in das offen gehaltene Interview mit kleinen Diskussionsrunden wurden drei Fragen zum Feedbackbogen gestellt. Mit allen Probanden wurde zunächst über die implementierten Hierarchiestufen diskutiert. Hierbei kamen alle zu dem Schluss, dass eine Ebene unter dem Hauptnetzwerk der FLOW-Analyse sehr geholfen hat, aber mehr als zwei Ebenen das Netzwerk zu unübersichtlich gemacht hätten.

Daran anknüpfend wurde mit den Experten über die Subnetzwerke an sich gesprochen. Die implementierte Funktion des Zoomes in Subnetze für die Erstellung und Bearbeitung dieser wurde auch als sehr hilfreich bewertet. Die Funktion „Merge-All“, die alle Subnetzwerke in das Diagramm geladen hat, wurde allerdings von zwei Probanden als eher unwichtig angesehen, da es reichen würde, ausgewählte Subnetze zu mergen.

Auch wenn das Auflösen von Activity-Knoten nicht direkt in dem Szenario hervorgehoben wurde, konnte doch die beispielhafte Anwendung zeigen, dass dieses mit den entwickelten Detailgraden auch lösbar ist. Um Activities aufzulösen, wird ihnen ein Subnetzwerk hinzugefügt, welches nur aus flüssigen und festen Informationen besteht. Durch das Mergen dieser Subnetzwerke werde alle Activity-Knoten ersetzt. Aufbauend auf diesen

gemergten Netzwerken können dann ohne weitere Anpassungen SNA's angewendet werden.

Der nebenbei realisierte Hierarchiebaum, der in einem extra Fenster geöffnet wurde, wurde ebenfalls als hilfreiche Unterstützung aufgenommen.

Als letzter Punkt wurde die Art des Informationskanals diskutiert. Die Darstellung mit kleinen Icons an den Kanten wurde als sehr hilfreiche Zusatzinformation für FLOW-Diagramme von allen Probanden angesehen. Ergänzend wurde von einem Probanden gewünscht, dass durch einen Klick auf ein Icon mehr Informationen zu dem Informationskanal bzw. Medium gezeigt werden.

### **Zusammenfassung**

Zusammenfassend ist festzuhalten, dass die implementierten Konzepte nicht nur den Ansprüchen der Experten und den Anforderungen im vollem Umfang genügen, sondern auch hilfreiche Unterstützung für die FLOW-Analyse bieten. Die unterschiedlich einstellbaren Granularitäten können so einerseits das abstrakte Modell für den Überblick geben, andererseits eine detaillierte Darstellung liefern, um Informationsflussanalysen zukünftig weiter zu optimieren.

Auch für die Problematik mit dem Auflösen der Activities bieten diese Granularitäten eine Lösung. In Activity-Knoten kann nun ein Subnetzwerk hinterlegt werden, welches feste und flüssige Informationsspeicher enthält. Durch das Mergen dieses Subnetzwerkes in das Hauptnetzwerk werden die Activities ersetzt. SNA's sind somit ohne weitere Einschränkungen auf dem Netzwerk möglich.

Obwohl alle Probanden des Experimentes die hierarchischen Ebenen und die zum Einsatz kommenden Subnetzwerke als sehr hilfreiche Funktion bewertet haben, sollten diese nur bis zu einer gewissen Ebene genutzt werden. Mit dem Ziel, über diese Arbeit hinausgehende Erkenntnisse zu gewinnen, wurde schon während der Interviews mit den Probanden zusammen an einer Empfehlung für die Arbeit mit den neuen Hierarchien gefeilt. Diese wird nachfolgend erläutert.

## **6.4 Reliabilität**

Für die Interviews wurden die vier Experten aus der Konzeptrunde zu vier verschiedenen Terminen eingeladen. Obwohl die vier Interviews nicht den Anforderungen an eine Studie genügen, werden aufgrund der Erfahrung der Experten hinsichtlich der aufgeführten Themen die daraus resultierenden Empfehlungen als relativ wertvoll angesehen. Ebenfalls ist

zu beachten, dass diese implementierten Funktionen vor allem für die Experten selber zur Verfügung stehen sollen.

## 6.5 Empfehlung

Dementsprechend ist die folgende Empfehlung weder verpflichtend noch aussagekräftig. Sie soll aber aufzeigen, worauf bei einer FLOW-Analyse geachtet werden sollte.

Obwohl das Arbeiten mit Subnetzen zu einer Verbesserung der FLOW-Analyse führen kann, sollte darauf geachtet werden, dass nicht zu viele Informationen dargestellt werden, sondern, wie auch in den Grundlagen in Kapitel 2.1 aufgegriffen, nur grob modelliert wird. Während der Evaluation haben sich alle Experten darauf geeinigt, dass maximal eine, höchstens zwei Subnetzwerk-Ebenen unter dem Hauptnetzwerk angelegt werden sollten. Ebenfalls sollte das hinzugefügte Subnetzwerk neben den ursprünglichen Ein- und Ausgängen nicht mehr als fünf bis zehn zusätzliche Knoten enthalten.

Wenn beispielsweise zehn Subnetze mit jeweils zehn Knoten angelegt werden und nur bei der Hälfte an Knoten wieder ein Subnetzwerk mit zehn Knoten entsteht, würden beim Mergen aller Subnetzwerke ungefähr 500 weitere Knoten hinzugefügt werden, die die Übersichtlichkeit stark einschränken.

Abschließend lässt sich damit die Aussage treffen, dass die verschiedenen Granularitäten die FLOW-Diagramme verfeinern und eine Informationsflussanalyse optimieren können, dies aber nur funktionieren kann, wenn die Anzahl an verwendeten Subnetzwerken und Ebenen begrenzt wird. Die Empfehlung diesbezüglich lautete ein bis zwei Subnetzwerk-Ebenen mit einer begrenzten Anzahl von fünf bis zehn Knoten in den Subnetzwerken.

# Kapitel 7

## Fazit und Ausblick

In diesem letzten Kapitel wird unter dem Aspekt der Zielsetzung ein Fazit gezogen und ein Ausblick auf Themen gegeben, die perspektivisch im Anschluss an diese Arbeit umzusetzen sind.

### 7.1 Fazit

Entstanden ist in dieser Bachelorarbeit eine hierarchische Struktur für FLOW-Modelle, um Informationsflussanalysen zu optimieren. Dafür wurden im Zuge dieser Arbeit verschiedene Konzepte für die Optimierung von Informationsflussanalysen durch das Implementieren von hierarchischen Detailgraden erstellt. Diese Konzepte wurden unter Berücksichtigung ihrer späteren Verwendung entwickelt und zusammen mit einer Gruppe von Experten diskutiert. Im Anschluss wurde das Top-down-Konzept zur Visualisierung und das Link-Konzept zur Verfeinerung der Datenstruktur in ein bestehendes Software-Tool implementiert.

Die implementierten Granularitäten wurden einzeln von den Experten geprüft. In Verbindung mit einem kleinem Szenario und einem anschließenden Interview wurde ein Feedback eingeholt. Für diesen Zweck wurden zwei verschiedene Szenarien aus einem Netzwerk abgeleitet und die Probanden mit einem davon konfrontiert. Einschränkend war bei der zweiten Gruppe der Computer, der den Probanden zur Verfügung gestellt wurde. Da dieser parallel zu der Nutzung des Software-Tool von der CPU und dem RAM hoch ausgelastet war, dauerte das Einlesen der Dateien entsprechend etwas länger als üblich. Das wirkte sich auf den Wechsel zwischen den einzelnen Netzwerken negativ aus und verschlechterte den Effekt, der beim Zoomen durch die verschiedenen Ebenen entstehen sollte. Nach ein bis zwei Sekunden wurde dementsprechend erst das gewünschte

Ergebnis geliefert. Trotzdem konnte ein aussagekräftiges Feedback über die entwickelten Konzepte und ihrer Implementation eingeholt werden. Die daraus gewonnenen Erkenntnisse bestätigen, dass die hierarchischen Detailstufen die FLOW-Modelle visuell aufbereiten und verfeinern. Dementsprechend stellen die hier arbeiteten Konzepte eine Weiterentwicklung von Informationsflussanalysen dar.

## 7.2 Ausblick

Abschließend lassen sich einige Aspekte aufzählen, die sich während dieser Bachelorarbeit herauskristallisiert haben und an denen zukünftig weiter geforscht werden kann. Wie in Kapitel 5.4 aufgeführt, sind einige softwaretechnische Erweiterungen interessant für die zukünftige Nutzung des Software-Tools.

Nach den Implementierungen ist es möglich, beliebig viele Subnetzwerke hierarchisch zu verlinken. Hierzu könnten Überlegungen angestellt werden, diese softwaretechnisch zu begrenzen oder einen Hinweis ab einem bestimmten Level einzubauen.

Unabhängig von der softwaretechnischen Unterstützung sollte untersucht werden, wie sich die Rohdaten aus der Datenerhebungsphase der FLOW-Methode am besten in die hierarchischen Strukturen implementieren lassen. Beispielsweise könnte eine Automatisierungsfunktion die Prozessschritte eines Prozesses oder eines allgemeinen Vorgehens in ein Subnetzwerk aktiv einbauen.

Des Weiteren gibt es noch kein standardisiertes Vorgehen, um ein FLOW-Diagramm aufzubauen. Wiederkehrende Schemata könnten beispielsweise, wie bei Design-Pattern aus der Softwareentwicklung, erarbeitet und im Nachhinein softwaretechnisch unterstützt werden. In diesem Zusammenhang wäre eine Analyse von Schemata verschiedener Kommunikationssituationen erforderlich. Darauf aufbauend könnte eine Generalisierung von standardisierten „FLOW-Patterns“ erfolgen.

# Abbildungsverzeichnis

2.1	Beispiel eines FLOW-Diagrammes bei verteilten Softwareprojekten . . . . .	7
2.2	Syntax der grafischen FLOW-Notation . . . . .	8
3.1	Beispiel von unterschiedlichen Granularitäten . . . . .	13
3.2	Top-down-Konzept - Mockup 1 . . . . .	14
3.3	Top-down-Konzept - Mockup 2 . . . . .	15
3.4	Top-down-Konzept - Mockup 3 . . . . .	15
3.5	Abstraktion verschiedener Knoten . . . . .	17
3.6	Bottom-up-Konzept - Mockup 1 . . . . .	17
3.7	Bottom-up-Konzept - Mockup 2 . . . . .	18
3.8	Bottom-up-Konzept - Mockup 3 . . . . .	18
3.9	Konzeptkombinationen . . . . .	21
4.1	Entscheidungsmatrix . . . . .	24
5.1	Beispiel eines FLOW-Diagramms mit erweiterten Kanteneigenschaften . . . . .	31
5.2	Beispielhaftes Schema in einer XML-Datei . . . . .	32
5.3	Generiertes FLOW-Diagramm aus dem XML-Schema . . . . .	33
5.4	Bearbeitungsfenster eines ausgewählten Knotens mit dem „edit subnet“-Button . . . . .	34
5.5	Erstellung eines neuen Subnetzwerkes mit den ursprünglichen Ein- (links) und Ausgängen (rechts) des übergeordneten Knotens . . . . .	34
5.6	Anlegen von weiteren Knoten und Details im Subnetzwerk . . . . .	35

5.7	Hauptnetzwerk mit der verlinkten Datei . . . . .	35
5.8	Beispielnetzwerk mit einem verlinkten Subnetzwerk im Knoten „Activity C” . . . . .	37
5.9	Beispielnetzwerk mit dem integrierten Subnetzwerk . . . . .	37
1	Feedbackbogen . . . . .	54
2	Feedback Szenario - Gemerged Netzwerk . . . . .	55
3	Feedback Szenario - Netzwerk . . . . .	56

# Literaturverzeichnis

- [1] Jil Klünder, Kurt Schneider, Fabian Kortum, Julia Straube, Lisa Handke, Simone Kauffeld: *Communication in Teams - An Expression of Social Conflicts*, In HumanCentered and Error-Resilient Systems Development: IFIP WG 13.2/13.5 Joint Working Conference, 6th International Conference on Human-Centered Software Engineering, HCSE 2016, and 8th International Conference on Human Error, Safety, and System Development, HESSD 2016, Stockholm, Sweden, August 29-31, 2016, Proceedings, pages 111-129. Springer International Publishing, 2016.
- [2] *Manifest für Agile Softwareentwicklung*  
<https://agilemanifesto.org/iso/de/manifesto.html>,  
(Abrufdatum: 20.01.2019).
- [3] Kai Stapel, Kurt Schneider: *FLOW-Methode - Methodenbeschreibung zur Anwendung von FLOW*, Technical report, Fachgebiet Software Engineering, Leibniz Universität Hannover, <http://arxiv.org/abs/1202.5919>, 2012.
- [4] *Internetseite des Fachgebietes für Software Engineering der Leibniz Universität Hannover*  
[http://www.se.uni-hannover.de/pages/de:projekte\\_flow](http://www.se.uni-hannover.de/pages/de:projekte_flow),  
(Abrufdatum: 20.01.2019).
- [5] Stephan Kiesling, Jil Klünder, Diana Fischer, Kurt Schneider, Kai Fischbach: *Applying Social Network Analysis and Centrality Measures to Improve Information Flow Analysis*, In Product-Focused Software Process Improvement, Springer International Publishing, 978-3-319-49094-6-25, 2016.

- [6] Jil Klünder, Kurt Scheider: *Informationsfluss in verteilten Softwareprojekten - eine Einzelfallstudie*, Fachgebiet Software Engineering, Leibniz Universität Hannover, 2017.
- [7] Kai Stapel: *Informationsflusstheorie der Softwareentwicklung*, Fachgebiet Software Engineering, Leibniz Universität Hannover, 2012.
- [8] Jonas Wallat: *Vergleich von Algorithmen zur Informationsflussanalyse in der Software-Entwicklung*, Fachgebiet Software Engineering, Leibniz Universität Hannover, 2017.
- [9] Stefan Noll: *Aufwand und Nutzen verschiedener Kommunikationswege in der Software-Entwicklung*, Fachgebiet Software Engineering, Leibniz Universität Hannover, 2018.
- [10] Klaus Polh, Chris Rupp: *Basiswissen Requirements Engineering*, dpunkt.verlag , 4. Auflage, 2015.
- [11] Matthias Geirhos: *IT-Projektmanagement*, Galileo Press, 978-3-8362-1773-6 Bonn, 2011.
- [12] „*Top-down-Methode*“ auf Duden online,  
[https://www.duden.de/rechtschreibung/Top\\_down\\_Methode](https://www.duden.de/rechtschreibung/Top_down_Methode),  
(Abrufdatum: 20.01.2019).
- [13] „*Bottom-up-Methode*“ auf Duden online,  
[https://www.duden.de/rechtschreibung/Bottom\\_up\\_Methode](https://www.duden.de/rechtschreibung/Bottom_up_Methode),  
(Abrufdatum: 20.01.2019).



# Anhang

## Dokumente

Wie viele Informationen braucht eine gute FLOW-Analyse?

### Übersichtlichkeit

War das Diagramm übersichtlich?

Sehr gut	Gut	Befriedigend	Schlecht	Mangelhaft
----------	-----	--------------	----------	------------

### Verständlichkeit

War das Diagramm verständlich?

Sehr gut	Gut	Befriedigend	Schlecht	Mangelhaft
----------	-----	--------------	----------	------------

### Befunde

Wie einfach war das Erkennen von Befunden?

Sehr gut	Gut	Befriedigend	Schlecht	Mangelhaft
----------	-----	--------------	----------	------------

### Informationen

Wie viele Details benötigt ein FLOW-Diagramm?

deutlich weniger	weniger	passend	mehr	viel mehr
------------------	---------	---------	------	-----------

### Hierarchiestufen

War der Einsatz von Hierarchien hilfreich, um Befunde festzustellen?

Sehr gut	Gut	Befriedigend	Schlecht	Mangelhaft
----------	-----	--------------	----------	------------

Sollten mehr oder weniger Hierarchiestufen verwendet werden?

deutlich weniger	weniger	passend	mehr	viel mehr
------------------	---------	---------	------	-----------

### Informationskanäle

War die Beschriftung der Kanten (Icons für E-Mail, Chat, ...) hilfreich?

Helfen sehr	Helfen	Macht keinen unterschied	Eher störend	Nur störend
-------------	--------	--------------------------	--------------	-------------

Weitere Bemerkungen:

# FLOW-Diagramme

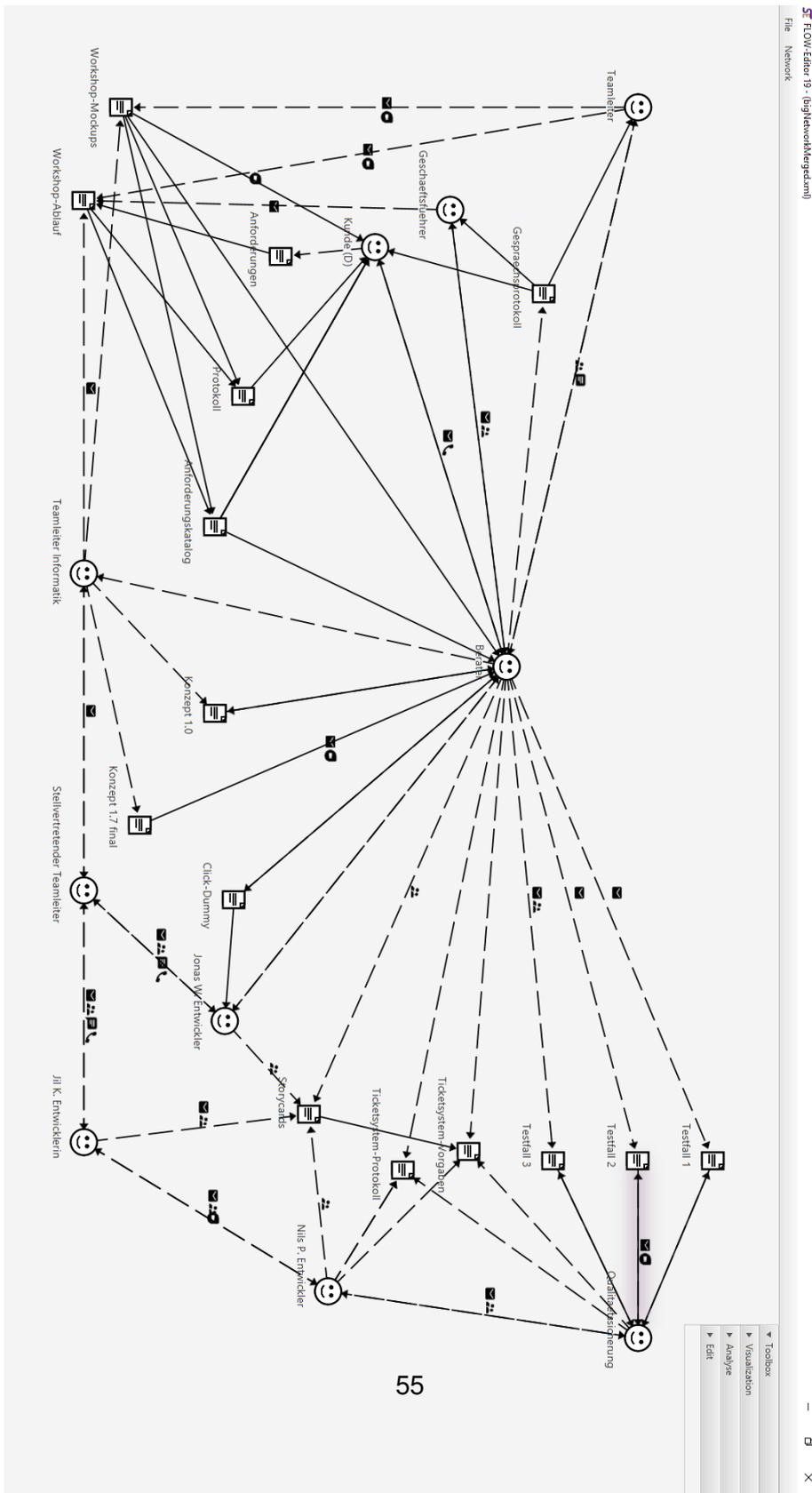


Abbildung 2: Feedback Szenario - Gemerged Netzwerk

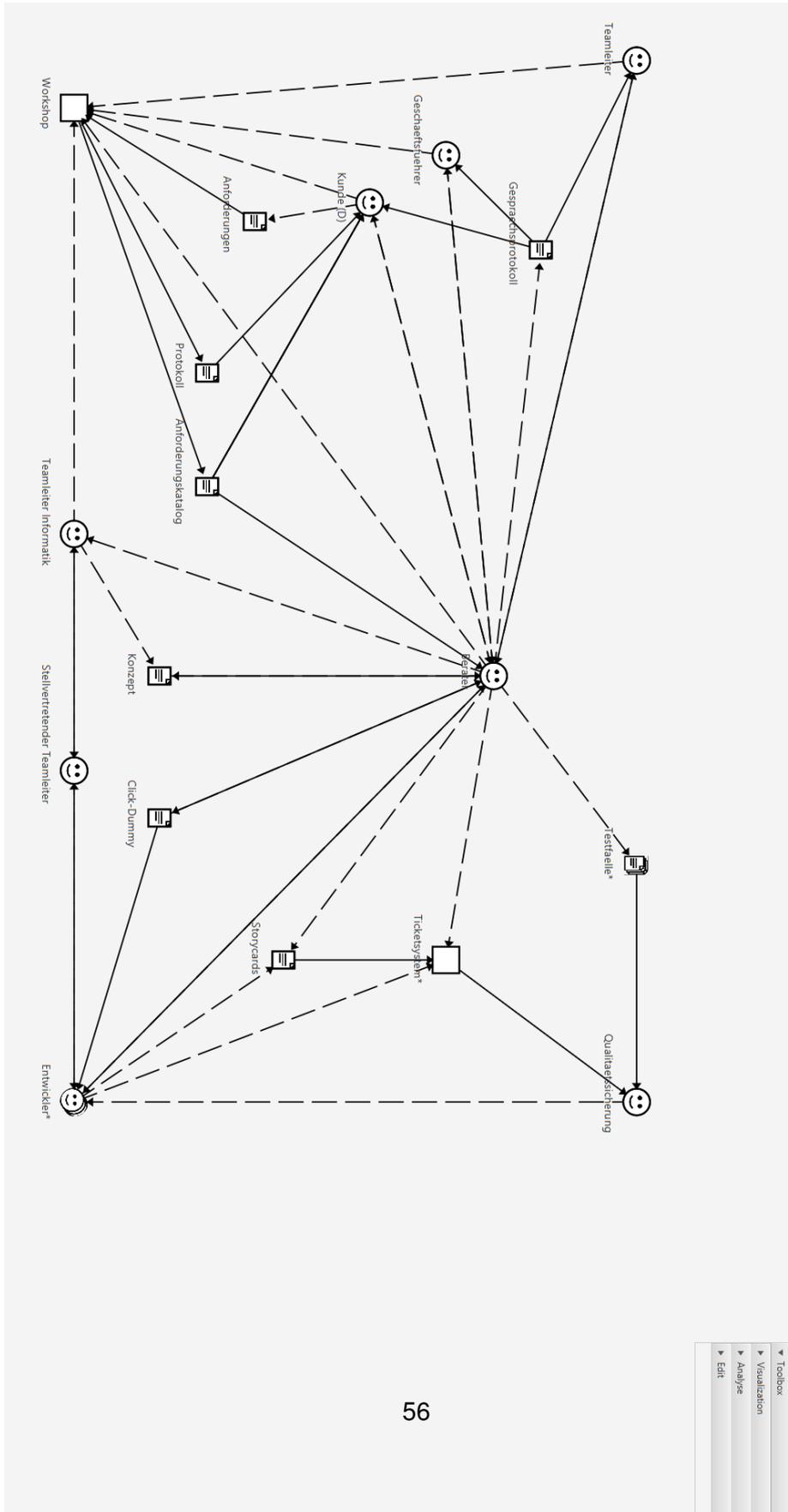


Abbildung 3: Feedback Szenario - Netzwerk

# Erklärung der Selbstständigkeit

Hiermit versichere ich, dass ich die vorliegende Bachelorarbeit selbständig und ohne fremde Hilfe verfasst und keine anderen als die in der Arbeit angegebenen Quellen und Hilfsmittel verwendet habe. Die Arbeit hat in gleicher oder ähnlicher Form noch keinem anderen Prüfungsamt vorgelegen.

Hannover, den 24. Januar 2019

---

Lennart Schroth

