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

Interaktiver Editor für Informationsflussmodelle

Bachelorarbeit

im Studiengang Informatik

von

Marco Höppner

Prüfer: Prof. Dr. Kurt Schneider
Zweitprüfer: Prof. Joel Greenyer
Betreuer: M. Sc. Fabian Kortum

Hannover, 12.09.2016

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 12.09.2016

_____

Marco Höppner

Zusammenfassung

Diese Arbeit befasst sich mit der Entwicklung eines interaktiven Explorers zur Erstellung, Bearbeitung, Analyse und Strukturoptimierung von FLOW-Modellen. Bisherige FLOW-Werkzeuge setzten ihren funktionalen Schwerpunkt lediglich auf die Erstellung und Bearbeitung von FLOW-Modellen, jedoch nicht auf die gezielte Analyse und Optimierung dieser. Der in dieser Arbeit entwickelte FLOW-Explorer bietet diese Funktionen, welche den Anwendern die Möglichkeit bietet, komplexe Kommunikationsstrukturen und Informationsflüsse in Softwareprojekten durch systemseitige Verfahren zu vereinfachen, anzupassen oder auch mit weiteren Modellen zu verbinden.Durch die Nutzung des FLOW-Explorers lässt sich von nun an viel Zeit sparen, zudem können Darstellungen vereinfacht und potenzielle Defizite schneller mit Teams sowie Projektverantwortlichen kommuniziert werden.

Abstract

This bachelor's thesis focuses on the development of an interactive explorer for the creation, editing, analysis and structure optimization of FLOW-models. Previous FLOW-tools mainly fulfil the purpose of creation and editing of FLOW-models; however, they do not support the targeted analysis and optimization of such. The FLOW-Explorer developed in this thesis offers those functions, which allow, utilizing system-side techniques, complex communication structures and information flows to be simplified, adjusted and combined with other models. Thereby usage of the FLOW-Explorer will save time, furthermore visual representations can be simplified and potential deficits can be communicated faster with teams and project leaders.

Inhaltsverzeichnis

Kapitel 1

Einleitung

In der Software-Entwicklung werden höchste Anforderungen an Zeit, Kosten und Qualität gestellt. Um den Erfolg eines Projektes sicherzustellen, ist eine aktive Kommunikation zwischen allen Beteiligten, sowie ein Austausch von Erfahrungen und Wissen notwendig. Die an der Leibniz Universität Hannover im Fachgebiet Software Engineering entwickelte FLOW-Modellierung beschäftigt sich mit der Analyse und Optimierung von Informationsflüssen in Softwareprojekten. Diese Arbeit beschäftigt sich mit der Entwicklung eines Softwarewerkzeugs, welches den Schwerpunkt auf die Analyse und Optimierung von FLOW-Modellen legt und dabei auch auf vergangene Arbeiten zurückgreift.

1.1 Problemstellung

Seit einiger Zeit werden and der Leibniz Universität Hannover im Fachgebiet Software Engineering mithilfe des FLOW-Modells die Kommunikationswege und Informationsflüsse in Software-Entwicklungsprozessen durch die FLOW-Notation visualisiert. Dies dient vor allem der Kommunikationsfluss-Gesamtübersicht, welche notwendig ist um Analysen hinsichtlich der Team-Strukturen durchzuführen. Zurzeit werden FLOW-Diagramme hauptsächlich per Hand auf Basis von Daten aus Interviews gezeichnet. Anschließend werden diese strukturell aufbereitet, damit diese übersichtlicher und somit als Modellansicht verwendbar werden, auch für Außenstehende. Besonders aufwendig ist das Zusammenführen der aus den einzelnen Interviews entstandenen FLOW-Modelle.
Die Idee eines Software unterstützten Editors für das FLOW-Modell ist nicht neu. In der Vergangenheit wurde am Institut für Software Engineering unter anderem ein Editor namens ProFLOW entwickelt[1]. Der Fokus dieses

Editors liegt bislang jedoch nicht nur auf dem Erstellen und Bearbeiten von FLOW-Modellen, sowie einer sekundär relevanten Verwendungsmöglichtkeit von weiteren Prozessnotationen, sodass FLOW-Diagramme z. B. mit UML-Diagrammen verbunden werden können. ProFLOW stellt jedoch keine Features zur Analyse und Strukturoptimierung komplexer Modellstrukturen, daher müssen alle Modellierungen und etwaige Anpassungen oder Zusammenführungen manuell von Hand nachgearbeitet werden. Das ist nicht nur zeitaufwendig, sondern stellt zudem einen kritischen Prozess dar, da nicht jede Informationsschnittstelle zwischen zwei Personen mit dem bloßem Auge für den Modellierer ersichtlich ist, wodurch die Darstellungs- und Vereinfachungs-Potenziale nur bedingt ausgeschöpft werden. Der generelle Vorteil von einem solchen Werkzeug ist, dass durch ProFLOW nun ein leicht editierbares digitales Format für FLOW-Modelle existiert, welches ein nachträgliches Editieren und Anpassen von Strukturen durch Teilautomatismen und Erkennungsalgorithmen erleichtert.
Entwickelt wurde ProFLOW zudem als ein Eclipse-Plugin unter Verwendung des Graphical Editing Frameworks(GEF) in der Version 3. ProFLOW ist nur als Plugin nutzbar und lässt sich daher leider nur mit Eclipse benutzen. Das erschwert den universalen Einsatz und erfordert zugleich die Umsetzung eines mobileren Anwendungsverfahrens wie dem des FLOW-Explorers.

1.2 Lösungsansatz

Um FLOW-Modelle digital analysieren, strukturieren und optimieren zu können, wird in dieser Arbeit ein neues Tool namens FLOW-Explorer beschrieben und entwickelt. Der Fokus des FLOW-Explorers liegt dabei rein auf der offiziellen FLOW-Notation. Andere Notationen wie z. B. UML werden nicht weiter unterstützt.
Mit dem FLOW-Explorer soll es nicht nur möglich sein FLOW-Modelle zu erstellen, sondern auch diese zu kombinieren, zu analysieren und in ihrer Struktur hinsichtlich der Komplexität und Überschaubarkeit zu optimieren. Der Informationsfluss soll hinsichtlich quantitativer Facetten sichtbar gemacht werden. Durch automatisierte Analyse von Informationsflüssen und Markierung von FLOW-Patterns sollen potenzielle Informationsüberladungen und andere Probleme, welche vermeidbar sind, frühzeitig identifiziert werden können.

1.3 Struktur der Arbeit

Diese Bachelorarbeit ist wie folgt strukturiert:

Im ersten Kapitel werden für das Verständnis wichtige Grundlagen erklärt. Das FLOW-Modell und das in dieser Arbeit verwendete Framework GEF4 werden vorgestellt.

In Kapitel 3 wird auf das Konzept des FLOW-Explorers eingegangen. Anforderungen an den FLOW-Explorer werden gezeigt sowie die Vorgehensweise genauer erläutert. Besonderheiten und Funktionen des FLOW-Explorers werden verdeutlicht.

Darauf wird in Kapitel 4 auf die Durchführung sowie auf wichtige Details der Implementierung eingegangen. Die einzelnen Klassen werden betrachtet.

In Kapitel 5 wird der FLOW-Explorer evaluiert. Dazu werden Programmtests mit einer Benutzergruppe durchgeführt und ausgewertet.

Zuletzt werden in Kapitel 6 die Ergebnisse zusammengefasst und ein Ausblick auf mögliche Erweiterungen in der Zukunft gegeben. Ein Fazit vollendet diese Arbeit.

Kapitel 2

Grundlagen

Dieses Kapitel beschäftigt sich mit den Grundlagen, welche für das Verständnis dieser Bachelorarbeit notwendig sind. Im Folgenden wird die FLOW-Notation und deren Semantik kurz erläutert. Außerdem wird auf den Datenerfassungsprozess für die FLOW-Modellierung sowie den grundlegenden Entstehungsprozess von FLOW-Modellen eingegangen. Zudem wird das Graphical Editing Framework, auf dem ProFLOW aufbaut, vorgestellt.

2.1 FLOW

2.1.1 FLOW-Notation

Die FLOW-Notation dient dazu, Informationsflüsse, Informationsspeicher und Aktivitäten übersichtlich in einem Diagramm darzustellen. Bei Informationsflüssen wird zwischen festen und flüssigen Informationsflüssen unterschieden.[9] Diese sind wie folgt definiert:

Feste Informationen sind Informationen, die von allen Personen zu jederzeit innerhalb einer Zeitspanne abgerufen und verstanden werden können. Als Informationsspeicher werden sie durch ein Dokumentensymbol modelliert. Informationsflüsse festen Aggregatzustandes werden durch einen durchgezogenen Pfeil modelliert. Bei festen Informationen handelt es sich z.B. um Dokumente, Aufnahmen oder Quellcode.

Flüssige Informationen sind Informationen, die nicht fest sind. Sie werden durch ein Smiley-Symbol modelliert. Informationsflüsse flüssigen Aggregatzustandes werden durch einen gestrichelten Pfeil modelliert. Bei flüssigen Informationen handelt es sich in der Regel um Wissen von

Personen, welches kommuniziert wird, z. B. durch Gespräche, Chats, E-Mails und Notizen. (Definition nach [9][p.7]) In Abbildung 2.1 werden der Zeichenvorrat und die Semantik der FLOW-Notation grafisch verdeutlicht.

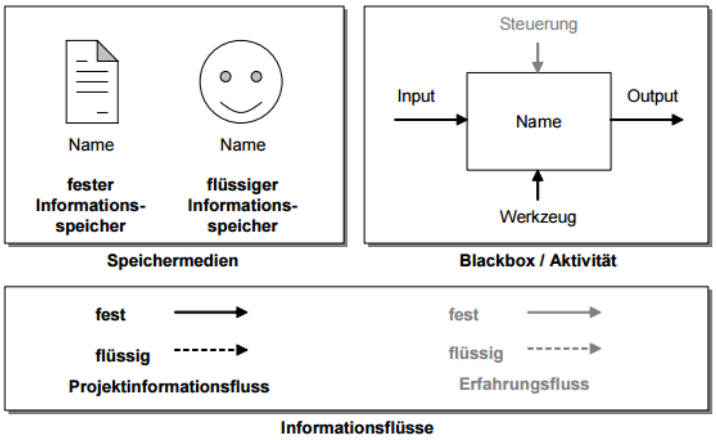


Abbildung 2.1: Zeichenvorrat und Semantik der FLOW Notation aus [8, p.68]

2.1.2 FLOW-Diagramm Erfassung

Vor Beginn der eigentlichen FLOW-Modellierung werden Interviews mit Personen in unterschiedlichen Rollen innerhalb eines Softwareprojektes durchgeführt. Wichtig dabei ist es, möglichst viele verschiedene Sichtweisen zu erfassen. In einem Interview werden Fragen gestellt, um zu ermitteln, inwiefern die befragte Person an dem Projekt beteiligt ist, welche Aufgaben die Person hat, welche Kontakte genutzt werden, wo und welche Informationen fließen. Jede ermittelte Aktivität wird in einem Formular erfasst. Eine Aktivität hat dabei vier Seiten. Danach werden alle von dieser Person ermittelten Aktivitäten in ein FLOW-Modell zusammengefasst. Hinterher werden die so aus verschiedenen Sichtweisen entstandenen FLOW-Modelle mittels vorhandener Schnittpunkte zu einem gesamten FLOW-Modell zusammengefügt, welches im Idealfall den gesamten Prozess eines Projektes in seinen Aktivitäten und Informationsflüssen repräsentiert. Dabei können Redundanzen entstehen, welche manuell beseitigt werden müssen. So können z. B. Personen, Dokumente und Aktivitäten mehrfach an verschiedenen Positionen im Modell vorkommen.

In einem fiktiven Unternehmen sei dadurch beispielsweise folgendes FLOW-Modell eines minimalistischen Softwareentstehungsprozesses entstanden.

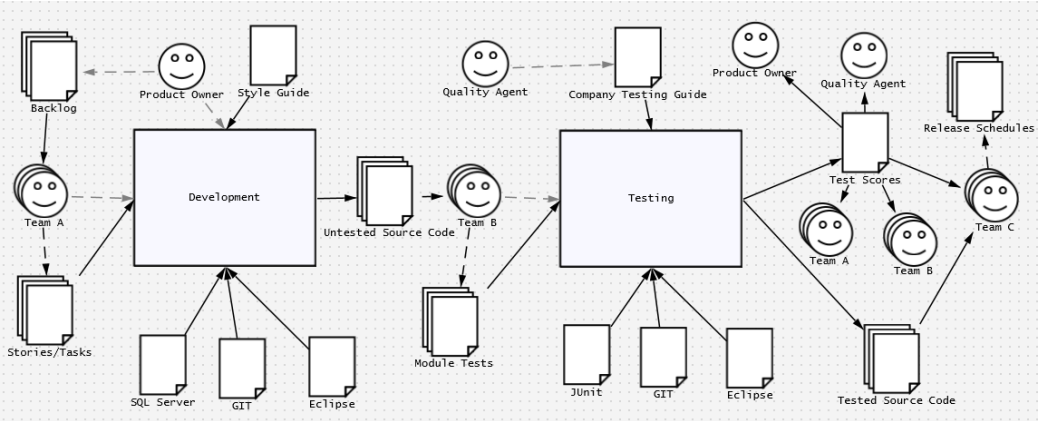


Abbildung 2.2: FLOW-Modell eines fiktiven Softwareentstehungsprozesses

2.2 Graphical Editing Framework

Das Graphical Editing Framework 4 (GEF4) ist ein auf JavaFX basierendes Framework, welches das Entwickeln von grafischen Editoren ermöglicht. ProFLOW benutzt eine ältere GEF3 Version, welche fest in Eclipse integriert ist. GEF4 ermöglicht auch die Entwicklung von standalone Software, welche die Ausführung des FLOW-Explorers ohne vorher aufgesetztes Eclipse-Umgebung ermöglicht. [4]

2.3 MVC Komponente von GEF4

Für diese Arbeit besonders nützlich ist die MVC-Komponente von GEF4, womit sich der neue FLOW-Explorer nach der Model-View-Controller Architektur entwickeln lässt.

Dabei wird ein Viewer mit Visual-Parts befüllt, welche jeweils die angezeigten Grafiken in dem Viewer kontrollieren. Alle Visual-Parts sind in einer Hierarchie angeordnet, wobei eine Wurzel (IRootNode) existiert, von der aus alle Visual-Parts erreichbar sind. Zusätzlich existieren Feedback- und Handle-Parts, welche Benutzer-Interaktion handhaben. Visual-Parts können Ankerbeziehungen miteinander haben, welche nützlich seien können, wenn die Interaktion mit einem Element in einer Änderung eines anderen Elementes resultieren soll (z. B. wenn verankerte Elemente bei einer Verschiebung mit verschoben werden sollen). Außerdem existieren Policies, welche entscheiden wie Visual-Parts, welche an die jeweilige Policy gebunden sind, auf Benutzerinteraktion reagieren (mehr Informationen zur MVC-Komponente von GEF4 finden Sie auf der Eclipse-Wiki-Seite von GEF4 [5]).

Kapitel 3

Konzept

3.1 Verwandte Arbeiten

In der Bachelorarbeit von F. Peschke [6] wurde eine Erweiterung für ProFLOW entwickelt und vorgestellt, welche das Arbeiten mit großen Modellen erleichtern, sowie Usability-Aspekte des Editors verbessern sollte.

In der Bachelorarbeit von E. Marques [3] wurde ein Werkzeug entwickelt und vorgestellt, welches Informationsflüsse mit dem Software-Quanten-Modell modelliert und animiert.

3.2 Aufbau auf bestehender Software

Zuerst wurden existierende FLOW-Editoren untersucht, um zu entscheiden, wie am besten auf Bestehendem aufgebaut werden kann. Es stellte sich heraus, dass ProFLOW [1] nicht alle von der Semantik der FLOW-Notation gesetzten Regeln einhält. So überprüft ProFLOW nicht, dass Informationsflüsse die korrekten Inputs und Outputs einer angebundenen Aktivität verwenden. Im folgenden Beispiel (Abbildung 3.1) sind zwei Fälle gezeigt um dieses zu verdeutlichen. In einem Fall ist Tim Product Owner, wobei der Informationsfluss in den Steuerungseingang der Aktivität fließt. Im anderen Fall ist Tim Entwickler, wobei sein Informationsfluss in den normalen Eingang der Aktivität fließt. In der FLOW-Notation hat dies unterschiedliche Bedeutungen. Das Modell von ProFLOW unterscheidet jedoch nicht zwischen diesen beiden Fällen, wodurch die Analyse eines in ProFLOW erstellten FLOW-Modells mit automatischen Verfahren nicht direkt möglich ist.

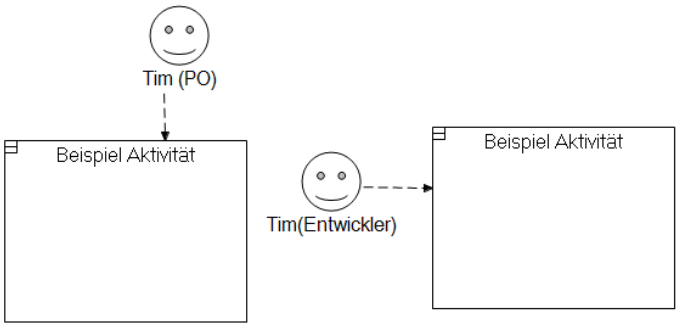


Abbildung 3.1: Problem mit ProFLOW

Im abgespeicherten .ProFLOW Dateiformat, welches auf XML basiert, gibt es transition-Objekte, welche Informationsflüsse realisieren. In solch einem Objekt stehen lediglich die beiden mit diesem Informationsfluss verbundenen Objekte, nicht aber um welchen Typ von Eingang es sich im Fall einer Aktivität handelt. Aus diesem Grund wird das von ProFLOW erstelle Dateiformat im FLOW-Explorer zwar als Einweg-Import unterstützt, aber im FLOW-Explorer bearbeitete Modelle können nicht rückwirkend in ProFLOW geladen werden.

Dazu wird beim Import ermittelt, in welche Eingänge der Aktivität Informationen fließen. Ausgehende Informationsflüsse können nach FLOW-Notation nur den rechten Aktivitäten-Output verwenden, daher sind diese leicht zu ermitteln. Hat ein Informationsfluss als Quellobjekt eine Aktivität, so muss dieser an dem Ausgang der Aktivität verankert werden. Informationsflüsse mit einer Aktivität als Zielobjekt hingegen haben drei verschiedene mögliche Eingänge an der Aktivität, welche diese nutzen könnten. Um welchen Eingang es sich handelt, wird dabei anhand der Position des Quellobjektes des Informationsflusses geschätzt. Dabei können in Grenzfällen allerdings Fehler passieren und der Benutzer sollte überprüfen, ob die Aktivitäten richtig verbunden sind.

Da das ProFLOW Format nur bedingt unterstützt wird, wird für den FLOW-Explorer ein neues XML-Format erzeugt, welches der FLOW-Notation gerecht wird und somit auch für zukünftige Applikationen nützlich sein kann, sofern diese das in dieser Arbeit erzeugte XML-Format unterstützen.

Abbildung 3.2: XML Import / Export

3.3 Wahl des Frameworks

Um die Entwicklung des FLOW-Explorers zu starten, muss entschieden werden, mit welchem Framework der neue FLOW-Explorer entwickelt werden soll. Da ProFLOW bereits GEF3 benutzt, wurde zuerst recherchiert, ob dieses weiterverwendet werden sollte, oder ob der neue FLOW-Explorer etwas anderes benutzen sollte. Dabei stellte sich heraus, dass bereits eine neuere Version des Graphical Editing Frameworks verfügbar ist, welche allerdings nicht direkt auf der altern Version aufbaut. Zum einen wird in der neuen Version JavaFX anstelle von SWT zur Darstellung verwendet, welches für die Zukunft einige Vorteile mit sich bringt, da es sich dadurch gegebenenfalls leicht auf andere Plattformen wie z. B. Tablets portieren lässt. Außerdem ist es mit der neueren Version möglich, einen standalone Editor zu entwickeln, welches mit der früheren Version von GEF leider nicht möglich war. GEF3 hat sich in der Vergangenheit als erfolgreiches Framework etabliert,wird nun jedoch nicht mehr weiterentwickelt, da der Entwicklungsfokus nun auf der Verbesserung von GEF4 liegt.
GEF4 war zu Beginn dieser Arbeit noch in Entwicklung und hat sich daher noch nicht flächendeckend durchgesetzt, bietet aber mehr Funktionen und besitzt mehr Potenzial für die Zukunft, da es noch weiterentwickelt wird. [4] Mittlerweile wurde eine stabile Version erreicht.
GEF4 besitzt unter anderem ein umfangreiches MVC-Modul, mit dem sich Anwendungen nach dem Model-View-Controller Pattern entwickeln lassen.
Daraus resultierend bildete sich die Frage, ob der neue FLOW-Explorer auf GEF3 basieren und ggf. Elemente aus ProFLOW wiederverwenden soll, oder die neuere GEF4 Version verwenden soll.

Einer der für diese Entscheidung wichtigsten Punkte ist, dass GEF4 im standalone Betrieb läuft, während GEF3 nur als Eclipse Plugin nutzbar ist. Dadurch könnte der FLOW-Explorer ohne Weiteres auf jedem Computer ausgeführt werden, auf dem Java installiert ist. Gegen die Nutzung von GEF4 spricht, dass es noch sehr neu ist, und daher gegebenenfalls noch nicht fehlerfrei ist. Sollte der FLOW-Explorer mit GEF4 entwickelt werden, muss extra Zeit für Probleme, welche durch mögliche Fehler in GEF4 auftreten, eingerechnet werden. Außerdem muss anfangs ein größerer

Aufwand erbracht werden, um das neue Framework zu verstehen. Dies sollte allerdings nur bedingt ein größerer Aufwand sein, da alternative Frameworks auch erst verstanden werden müssen. Nachteil an GEF4 ist dabei, dass vorhandene Dokumentation nur bedingt nützlich ist, da sie sich teilweise auf ältere GEF4 Versionen bezieht. Funktionen in GEF4 könnten sich während der Entwicklung des FLOW-Explorers weiter verändern, welches ebenfalls zusätzliche Zeit in Anspruch nehmen könnte. So bringt GEF4 einige Risiken mit sich. Eine Verschiebung des Zeitplans könnte die Folge sein.

Nach Entwicklung eines Prototypen um die Tauglichkeit von GEF4 zu evaluieren, wurde die Entscheidung getroffen, GEF4 für die Entwicklung des FLOW-Explorers zu verwenden.

Als Alternativen wurden Piccolo2D und Graphviz in Erwägung gezogen. Graphviz erlaubt hauptsächlich die Erstellung von Graphen, bietet jedoch nicht so umfangreiche Funktionen zur Erstellung eines interaktiven Editors, wie GEF4 mit der MVC-Komponente. Piccolo2D bietet in dieser Hinsicht mehr Möglichkeiten als Graphviz, wird aber seit längerem nicht mehr weiterentwickelt.

3.4 Anforderungen an den FLOW-Explorer

Bevor mit der Entwicklung des FLOW-Explorers begonnen werden kann, müssen die Anforderungen an den FLOW-Explorer ermittelt werden. Dazu wurden im Rahmen einer Anforderungsanalyse einige Anwendungsfälle betrachtet. Im Folgenden wird genauer auf diese eingegangen.

3.4.1 Grundfunktionen

Der FLOW-Explorer baut nicht direkt auf alter Software wie ProFLOW auf, sondern wird mit dem Graphical Editing Framework 4 neu entwickelt. Damit zur Erstellung und Bearbeitung von FLOW-Modellen nicht auf alte Software zurückgegriffen werden muss, ist die erste Anforderung an den FLOW-Explorer die Erstellung und Bearbeitung von FLOW-Modellen. Dabei ist wichtig, dass die FLOW-Semantik korrekt eingehalten wird, um die Korrektheit der mit dem FLOW-Explorer bearbeiteten FLOW-Modelle zu garantieren. Folgende Use-Cases sind ermittelt worden:

1. ProFLOW Dateien Importieren

Beschreibung	Ein Benutzer importiert eine in ProFLOW erstellte Datei zur Verwendung im FLOW Explorer.
Ablauf	Benutzer klickt FILE -> Import ProFLOW... und wählt im aufgehenden Verzeichnisexplorer eine mit ProFLOW erstelle Datei aus.
Vorbedingung	Benutzer besitzt ein mit ProFLOW erstelltes FLOW-Modell
Effekt	Wenn es sich um eine gültige ProFlow Datei handelt, wird das importierte Modell angezeigt.

2. Speichermedien/Aktivitäten erstellen

Beschreibung	Erstellung von Speichermedien und Aktivitäten.
Ablauf	Benutzer wählt einen der fünf Buttons (Person, Gruppe, Dokument, Dokumente, Aktivität) aus. Danach klickt er auf Positionen in der Zeichenoberfläche. Wahlweise kann er dabei durch gedrückt halten der SHIFT-Taste weitere Objekte des gleichen Typs setzen.
Vorbedingung	—
Effekt	An den Positionen der Mausklicks auf der Zeichenoberfläche werden entsprechende FLOW-Elemente erstellt.

3. Objekte verschieben

Beschreibung	Verschiebung von Flow-Objekten und Wegpunkten.
Ablauf	Der Benutzer wählt das Tool zum Verschieben aus. Danach verschiebt er per drag & drop ein existierendes Objekt oder einen Wegpunkt eines Informationsflusses.
Vorbedingung	Es existiert mindestens ein verschiebbares Objekt.
Effekt	Die per drag & drop eingeleitete Verschiebung wird ausgeführt.

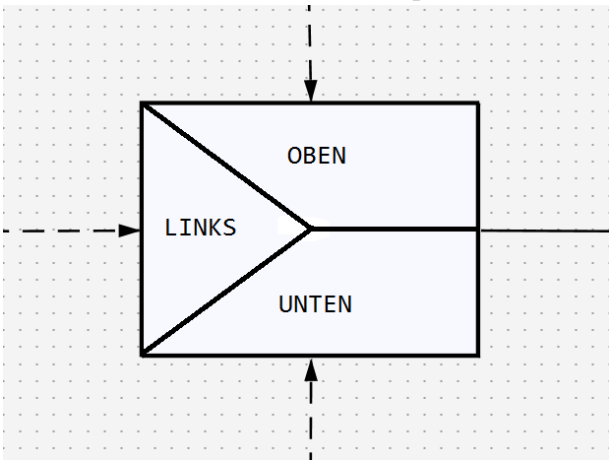
4. Speichern

Beschreibung	Das derzeitig geladene FLOW-Modell wird abgespeichert.
Ablauf	Der Benutzer klickt FILE -> Save und wählt im aufgehenden Verzeichnisdialog einen Dateinamen und ein Verzeichnis aus
Vorbedingung	Es existiert mindestens ein Objekt im geladenen FLOW-Modell.
Effekt	Das derzeitig geladene FLOW-Modell wird in das FLOW-Explorer XML-Format umgewandelt und im vom Benutzer gewählten Verzeichnis unter dem vom Benutzer gewählten Namen abgespeichert.

5. Laden

Beschreibung	Eine im FLOW-Explorer erstelle XML-Datei wird geladen
Ablauf	Der Benutzer klickt FILE -> Load. Gegebenenfalls wird er gefragt, ob das vorher geladene Modell verworfen werden soll. Der Benutzer wählt im aufgehenden Verzeichnisdialog eine XML-Datei aus
Vorbedingung	—
Effekt	Handelt es sich bei der ausgewählten XML Datei um ein gültiges FLOW-Explorer Format, so wird dieses geladen.

6. Objekte verbinden

Beschreibung	Erstellung von neuen Informationsflüssen zwischen Speichermedien/Aktivitäten.
Ablauf	Der Benutzer wählt eines der Tools zum Verbinden aus. Danach wählt er ein bereits existentes Speichermedium oder eine Aktivität aus. Zuletzt wählt er ein weiteres Speichermedium oder eine Aktivität als Ziel aus.
Vorbedingung	Es existieren mindestens zwei Speichermedien und/oder Aktivitäten auf der Zeichenfläche.
Effekt	Ein neuer Informationsfluss wird zwischen den beiden ausgewählten Objekten erstellt. Ist das Quellobjekt ein flüssiger Speicher, so wird die Verbindung auch automatisch flüssig (gestrichelt). Ist es fest, so wird die Verbindung ebenfalls fest. Andernfalls, wenn die Quelle eine Aktivität ist, wird der voreingestellte Aggregatzustand verwendet. Je nach gewähltem Verbindungstool wird die Verbindung grau oder schwarz gefärbt. Zusätzlich muss bei einer Aktivität als Ziel bestimmt werden, an welchen der drei möglichen Eingänge der Informationsfluss verbunden werden soll. Dazu wird die Position des Mausklicks verwendet. Je nachdem, in welchem der in folgender Abbildung erkenntlichen Bereiche der Mausklick stattfindet, wird der Informationsfluss dementsprechend verbunden. 

7. Voreinstellung für Verbindungen ändern

Beschreibung	Die Voreinstellung für den Aggregatzustand (Fest/Flüssig) wird verändert.
Ablauf	Benutzer klickt auf den Solid/Fluid Toggle Button.
Vorbedingung	Programm befindet sich im Verbindungsmodus.
Effekt	Die Voreinstellung des Aggregatzustandes wird umgeschaltet.

3.4.2 Erweiterte Funktionen

Mit den vorherigen Anforderungen haben wir nun einen simplen FLOW-Explorer, der FLOW-Modelle erstellen und bearbeiten kann, sowie in ProFLOW erstellte FLOW-Modelle importieren kann. Im Folgenden werden auf die erweiterten Funktionen des FLOW-Explorers eingegangen.

7. Mehrere Modelle zusammenfügen

Beschreibung	Ein existierendes FLOW-Modell wird mit dem zur Zeit im FLOW-Explorer geladenen Modell zusammengefügt.
Ablauf	Der Benutzer klickt FILE -> Load(add to current Model) und wählt im aufgehenden Verzeichnisexplorer eine mit dem FLOW-Explorer erstellte XML Datei aus.
Vorbedingung	Ein FLOW-Modell ist im FLOW-Explorer geladen (dies kann auch leer sein).
Effekt	Beide FLOW-Modelle wurden zu einem Modell zusammengefügt. Elemente aus beiden vorherigen Modellen überlappen sich nicht.

8. Gruppen von Objekten mit gleichen Namen markieren

Beschreibung	Gruppen von gleichnamigen Objekten werden in unterschiedlichen Farben markiert.
Ablauf	Ein Benutzer klickt 'Find & Merge Redudancies'
Vorbedingung	Es existieren mindestens zwei Objekte mit gleichem Namen und gleicher Klasse.
Effekt	Gruppen von gleichnamigen Objekten werden in verschiedenen Farben markiert.

9. Automatisches Zusammenfügen von Gruppen gleichnamiger Objekten: Vorschau

Beschreibung	Eine Vorschau des Effektes einer automatischen Zusammenführung.
Ablauf	Benutzer klickt auf ein markiertes gleichnamiges Objekt oder lässt sich durch klicken auf 'Suggest Merge' eine Zusammenführung vorschlagen.
Vorbedingung	Gleichnamige Objekte wurden markiert.
Effekt	Das geklickte oder vorgeschlagene Objekt wird grün eingefärbt, alle anderen Objekte mit diesem Namen werden rot markiert. Verbindungen, die die roten Objekte besaßen, werden, falls sie nicht bereits existieren, dem grünen Objekt hinzugefügt. Das Programm befindet sich nun im Vorschau Zustand.

10. Zusammenführung durchführen

Beschreibung	Ein Zusammenführungsprozess wird ausgeführt.
Ablauf	Ein Benutzer bestätigt das Zusammenführen oder bricht es ab.
Vorbedingung	Eine Vorschau ist aktiv.
Effekt	Bestätigt der Benutzer, werden alle rot markierten Verbindungen und Objekte gelöscht. Markierungen grüner Objekte werden aufgehoben. Bricht der Benutzer ab, werden neu hinzugefügte Verbindungen gelöscht und alle Markierungen aufgehoben.

11. Manuelles Zusammenfügen von Objekten gleicher Klassen

Beschreibung	Von Benutzer manuell ausgewählte Objekte werden zur Zusammenführungsvorschau hinzugefügt
Ablauf	Der Benutzer selektiert mehrere Objekte gleicher Klasse. Diese färben sich rot ein. Wird ein Objekt zwei Mal angeklickt, färbt es sich grün. Wird ein zweites Objekt zweimal angeklickt, so färbt dieses sich grün und das vorher grüne Objekt wird wieder rot. Ist der Benutzer mit der Auswahl zufrieden, klickt er 'Preview'.
Vorbedingung	Der Modus 'Manual Merge' ist ausgewählt
Effekt	Wenn kein grünes Objekt existiert, wird das erst ausgewählte rote Objekt grün. Verbindungen, die die roten Objekte besaßen, werden, falls sie nicht bereits existieren, dem grünen Objekt hinzugefügt. Das Programm befindet sich nun im Vorschau-Zustand

12. Layout

Beschreibung	Ein ausgewähltes Layout wird auf das Modell appliziert.
Ablauf	Der Benutzer klickt Layout und wählt eines der angezeigten Layouts aus.
Vorbedingung	Ein FLOW-Modell ist geladen
Effekt	Alle Objekte werden nach dem ausgewählten Layout sortiert.

13. Pattern Detektion

Beschreibung	Negative Flow Patterns im derzeitig geladenen Modell finden
Ablauf	Der Benutzer klickt auf Analyze -> Pattern Detection
Vorbedingung	Ein FLOW-Modell ist geladen
Effekt	FLOW-Patterns werden im geladenen FLOW-Modell gesucht und farbig markiert. Ein Informationsdialog erscheint, welche die Funde erklärt.

3.5 Grafische Benutzeroberfläche des FLOW-Explorers

Ziel des Designs der Benutzeroberfläche ist, ein übersichtliches, leicht zu benutzendes Werkzeug zu erhalten. Es wurde in Anlehnung an ProFLOW,

jedoch mit optimiertem Design und flexiblem UI als anwenderfreundliche standalone Applikation entworfen. Der Hauptteil des Programmfensters wird von der Zeichenoberfläche eingenommen, während darüber die Werkzeuge als große, durch Symbole ansprechende Buttons angeordnet sind. Als Erstes befindet sich auf der linken Seite der Button für den Verschiebemodus. Danach der Button für den Verbindungsmodus, in dem neue Informationsflüsse erstellt werden können. Die nächsten fünf Buttons sind zur Erstellung der einzelnen Informationsspeicher, sowie für Aktivitäten. Als Letztes befindet sich auf der rechten Seite der Button zum Zusammenfügen von Objekten.
Oben in der Menüleiste befinden sich die Menüpunkte File, Edit, Analysis und Layouts. Im File Menü befinden sich die Funktionen zum Laden, Speichern und Importieren von FLOW-Modellen. Im Edit Menü sind Funktionen zum rückgängig machen zu finden. Im Analyse Menü gibt es Funktionen zur Pattern-Detektion sowie zusätzliche Werkzeuge zur Analyse. Als Letztes sind im Layouts Menü Funktionen zur automatischen Anordnung aller FLOW-Objekte zu finden. Ein Mockup dazu ist in Abbildung 3.3 zu sehen.

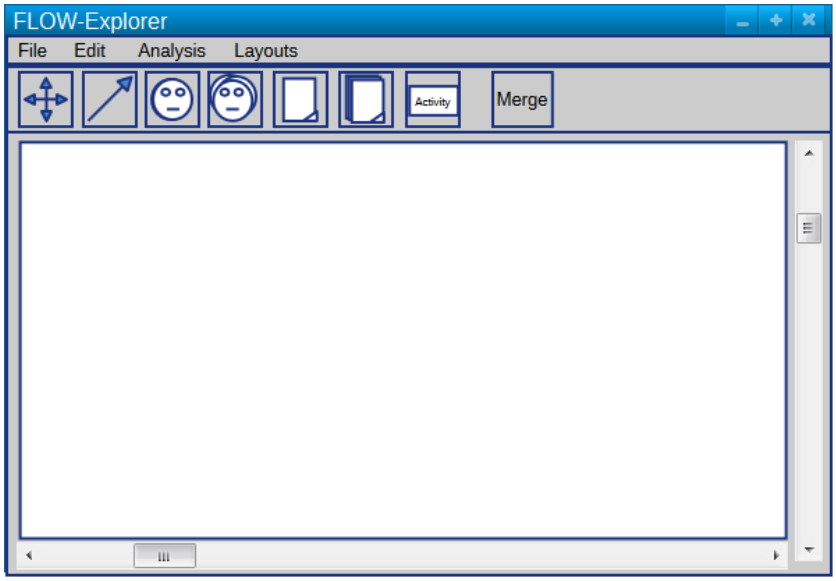


Abbildung 3.3: Mockup des FLOW-Explorers

3.6 Speicherformat des FLOW-Explorers

Um die FLOW-Modelle lesbar abzuspeichern, wird ein XML-Format verwendet.
Eine FLOW-Explorer XML-Datei ist wie folgt aufgebaut: Zuerst wird ein

<process> Element erstellt.
In dem <process> Element können beliebig viele FLOW-Elemente in beliebiger Reihenfolge enthalten sein. Zulässige FLOW-Elemente sind: <Activity>, <Person>, <Document> und <Information>. Jedes FLOW-Element besitzt ein *id* Attribut, welches einen einzigartigen ID String enthält. Im FLOW-Explorer werden diese als UUID generiert. <Person>, <Document> und <Activity> haben die Attribute *text*, x und y, welche den anzuzeigenden Text, sowie die Position des Objektes bestimmen. <Person> und <Document> haben zusätzlich ein *isGroup* Attribut, welches angibt, ob es sich um eine Gruppe handelt. <Information> hat folgende Attribute: *experience*, welches angibt ob es sich um eine Erfahrung handelt, *fluid*, welches den Aggregatzustand angibt, *sourceID* und *targetID*, welche die IDs der Quelle und Ziel dieses Informationsflusses angeben, *targetAnchor*, welches im Falle einer Aktivität als Ziel den zu nutzenden Eingang angibt, und zuletzt *wayPointsList*, welches eine Liste von Koordinaten für Wegpunkte, bzw Bendpoints darstellt.

Ziel des soeben beschriebenen XML-Formats ist es ein taugliches Format für die Zukunft zu erhalten, welches alle essenziellen Informationen eines FLOW-Modells enthält und leicht von anderen Programmen weiterverwendet werden kann und gegebenenfalls leicht erweitert werden kann.

3.7 Informationsflussanalyse

In einem Software-Entwicklungsprozess kann es vorkommen, dass Personen für zu viele Tätigkeiten zuständig sind und überlastet werden oder zu viele Informationen bewerkstelligen müssen. Dies kann dazu führen, dass sich der Zeitplan des Projekts verschiebt. Im schlimmsten Fall kann es sogar passieren, dass eine solche Person ein Burn-out-Syndrom entwickelt. Krankheitsbedingte Ausfälle sind die Folge. Dies stellt ein hohes Risiko dar, denn wenn eine Person ausfällt, müssen andere Personen die Aufgaben der ausgefallenen Person übernehmen, wodurch ggf. neue Informationsüberladungen entstehen. Informationsflussanalysen werden durchgeführt, um einen solchen Fall zu vermeiden, problematische Knotenpunkte aufzulösen und den Prozessfluss effizienter zu gestalten. Dazu werden die Informationsflüsse in einem FLOW-Modell ausgewertet. Im FLOW-Explorer geht dies z. B., indem die relative Menge an Informationsflüssen und Aktivitäten an denen eine Person beteiligt ist, ausgewertet wird. Da Informationsflüsse in FLOW nicht gewichtet sind, lässt sich über eine quantitative Analyse aber nur bedingt eine Aussage

treffen, ob eine Person wirklich überladen ist. Zu viele Informationsflüsse an einer Person können aber dennoch ein Indikator für Überlastung sein.
Der FLOW-Explorer determiniert den relativen Auslastungsgrad jeder Person im Verhältnis zu allen anderen Personen. Somit erhält man eine Gesamtübersicht mit potentiellen Knotenpunkten welche hinsichtlich ihrer kritischen Bewertung farblich hervorgehoben werden. Die am meisten kritisch bewerteten Elemente werden Rot markiert. Je nach Verhältnis zu dem am meisten kritischen Element werden alle anderen Elemente dann in einem Farbverlauf über Gelb bis Grün markiert.

3.8 Pattern-Detektion

In der Masterarbeit von X. Ge [2] werden bereits bekannte FLOW-Patterns in einem FLOW-Patterns-Katalog beschrieben. Im FLOW-Explorer können einige dieser Patterns erkannt werden. Negative, vom FLOW-Explorer erkennbare Patterns sind:

Dead Document: Ein Dead Document ist ein fester Informationsspeicher, der daran erkannt wird, dass er keine ausgehenden Informationsflüsse besitzt. Der feste Informationsspeicher wird in dem Prozess nicht weiterverwendet und seine Erstellung ist daher gegebenenfalls überflüssig.[2]

Stille Post: Bei stiller Post handelt es sich um eine Verkettung von flüssigen Informationsspeichern, welche in eine Richtung Informationen weitergeben, ohne diese zu dokumentieren.[2] Nach Metapher der Software-Quanten[7], kann in solch einer Situation viel Information verloren gehen, und auch neue, nicht erwünschte Information dazu kommen. Der FLOW-Explorer erkennt Sequenzen von drei aufeinanderfolgenden flüssigen Informationsspeichern und kann diese markieren.

Person als Senke: Bei einer Person als Senke handelt es sich um eine Person, welche mindestens zwei eingehende Informationsflüsse besitzt, aber keine ausgehenden. Eine solche Person könnte überfordert sein und leitet daher keine Informationen weiter.[2]

Missing Experience: Um eine 'Missing Experience' handelt es sich, wenn im gesamten FLOW-Modell keine einzelne Erfahrung weitergegeben wird, sprich kein Informationsfluss grau markiert ist.[2]

Da der Fokus momentan auf der Verbesserung von FLOW-Modellen

liegt, können positive Pattern zurzeit noch nicht erkannt werden. Dies wäre eine mögliche Erweiterung für die Zukunft.

3.9 Redundanzen

In dem in Kapitel 2 gezeigten Beispiel wird sichtbar, dass Personen und Dokumente in den Modellen mehrfach vorkommen können. Dies passiert, da Personen oft an mehreren verschiedenen Aktivitäten beteiligt sind und FLOW-Modelle aus mehreren Modellen zusammengesetzt werden. Diese mehrfach vorkommenden Objekte können vom FLOW-Explorer erkannt und zusammengefügt werden. Dabei werden alle Informationsflüsse, die vorher mit dem mehrfach vorkommenden Objekte verbunden waren, nun mit dem neuen, zusammengefügten Objekt verbunden.

Der FLOW-Explorer bietet zwei Modi zum Zusammenfügen von Objekten. Einen manuellen, und einen halb-automatischen.

3.9.1 Manuelles Zusammenfügen

Beim manuellen Zusammenfügen kann der Benutzer beliebige Gruppen von Objekten gleichen Types zusammenfügen. Dazu muss der Modus für das manuelle Zusammenfügen ausgewählt werden. Danach kann der Benutzer ein beliebiges Objekt anklicken. Sobald das erste Objekt angeklickt wurde, wird dieses rot markiert und Objekte anderer Typen werden gesperrt. Der Benutzer kann jetzt zusätzliche Objekte des gleichen Typs selektieren, oder mit einem Rechtsklick deselektieren. Wurden alle Objekte deselektiert, so können wieder Objekte anderen Typs selektiert werden.
Hat der Benutzer sich für eine Gruppe entschieden, welche er zusammenfügen möchte, kann er optional mit einem weiteren Linksklick auf eines der selektierten Objekte dieses grün markieren, das heißt, dass nach Durchführung der Zusammenführung genau dieses Objekt der selektierten Gruppe übrig bleibt. Wählt der Benutzer keins aus, so bleibt das zuerst ausgewählte Objekt übrig. Der Benutzer kann die Auswahl durch den Vorschau-Button(Preview) bestätigen. Wird die Auswahl bestätigt, so wird dem Benutzer vor der Durchführung eine Vorschau der Zusammenfügung gezeigt.

3.9.2 Zusammenfügungsvorschau

Das Zusammenführen von mehreren Objekten gleichen Typs bedeutet, dass alle ein- und ausgehenden Informationsflüsse der einzelnen zusammen-

zufügenden Objekte nach der Zusammenfügung an dem übrigbleibenden Zielobjekt angebunden werden müssen. Dazu werden alle Informationsflüsse, die die zusammenzufügenden Objekte besitzen, falls das Zielobjekt nicht bereits identische Verbindungen besitzt, kopiert und mit dem Zielobjekt verbunden. Um zu signalisieren, welche Verbindungen nach Durchführen des Zusammenfügungsprozesses übrig bleiben, werden diese kopierten Verbindungen grün markiert. Alle Objekte, welche gelöscht werden sollen, werden rot markiert. Der Benutzer hat nun die Wahl diese Zusammenfügung zu bestätigen oder diese abzubrechen. Wird sie bestätigt, so werden alle rot markierten Objekte gelöscht.Die grünen Markierungen werden aufgehoben. Bricht der Benutzer die Zusammenführung ab, so werden alle neuen grün markierten Verbindungen gelöscht.

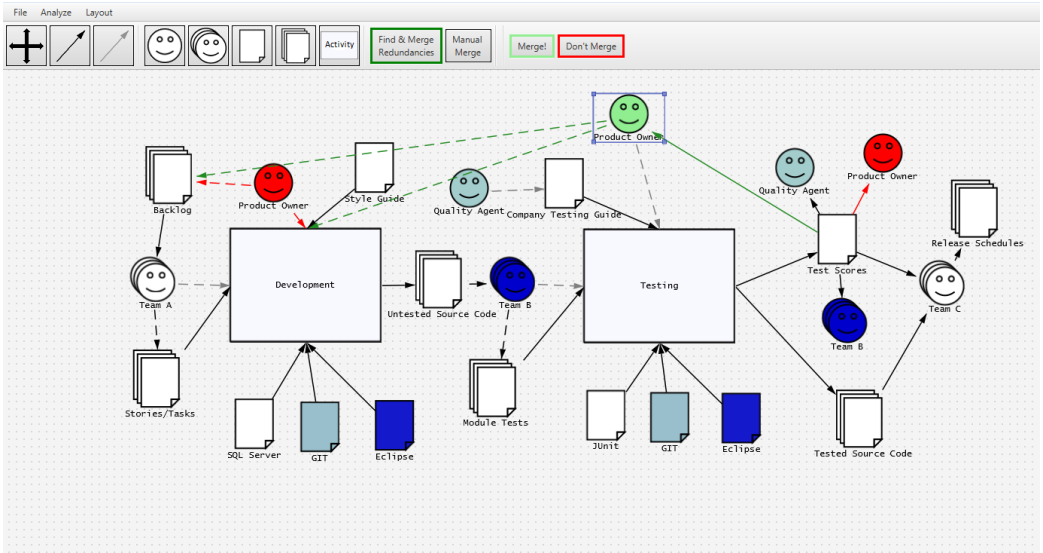


Abbildung 3.4: Zusammenfügen von Redundanzen / Eliminieren von Dopplungen

3.9.3 Automatisches Zusammenfügen

Bei der automatischen Zusammenfügung werden zuerst alle Gruppen von gleichnamigen Objekten in verschiedenen Farben markiert. Der Benutzer kann dann all diese Gruppen Schritt für Schritt mit jeweils einer Vorschau pro Gruppe zusammenfügen. Alternativ kann der Benutzer eine Gruppe anklicken, um genau diese Gruppe zusammenzufügen, dabei wird das angeklickte Objekt als Zielobjekt betrachtet, welches übrig bleiben soll. In Abbildung 3.4 sieht man ein FLOW-Modell, in dem Gruppen von

gleichnamigen Elementen markiert wurden und eine dieser Gruppen sich gerade im Vorschauschritt befindet. Dabei sollen alle drei Instanzen des Product Owners zusammengefügt werden. Hier wurde der grün markierte Product Owner als Ziel entschieden. Die rot markierten Informationsflüsse wurden kopiert und sind nun als grüne Informationsflüsse mit dem grünen Product Owner verbunden. Wird die Auswahl bestätigt, werden alle roten Elemente gelöscht und die Zusammenfügung durchgeführt

3.10 Layouts / Ansichten

Um die FLOW-Objekte übersichtlich zu sortieren, muss das FLOW-Modell erst analysiert werden. Um die finale Objektposition zu bestimmen, werden mehrere Kriterien in Betracht gezogen:

Distanz zu allen verbundenen Aktivitäten:
Die Distanz von einem Speichermedium zu einer Aktivität ist die minimale Anzahl an Sprüngen über durch Informationsflüsse verbundene andere Speichermedien.

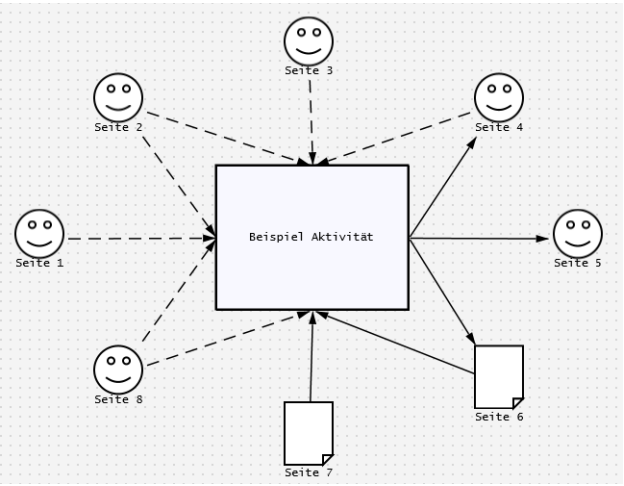


Abbildung 3.5: Seiten einer Aktivität für Positionsberechnungen der automatischen Layout Funktion

Die Seite, an der das Speicherobjekt mit der Aktivität verbunden ist:
Um zu bestimmen, auf welcher Seite der Aktivität ein Objekt angeordnet werden soll, wird die Aktivität in acht Seiten unterteilt.

Die acht Seiten setzten sich aus den vier regulären Seiten sowie aus vier Eckseiten zusammen. Die Eckseiten werden für Objekte genutzt, welche mit zwei nebeneinanderliegenden regulären Seiten verbunden sind.
Im nächsten Schritt wird die Seite der Elemente, welche eine Distanz der Länge 2 zu dieser Aktivität haben bestimmt. Dabei wird überprüft, zu welchen Elementen mit einer Distanz der Länge 1 diese verbunden sind. sind sie mit mehreren Objekten aus der 'Distanz-1-Schicht' verbunden, so wird ein Mittel bestimmt, welches die Seite entscheidet. Dieses wird für alle weiteren Distanzen wiederholt, bis alle Objekte abgearbeitet sind.
Im radialen Layouts Modus werden nun die FLOW-Objekte abhängig von ihrer Distanz und angebundenen Seite zu allen Aktivitäten in Ellipsen um die Aktivität, zu deren sie die kürzeste Distanz haben, herum sortiert.
Dabei wird die Menge an Objekten pro Seite pro Schicht verglichen. Hat eine Seite in einer Schicht mehr Elemente als eine andere Seite, so wird dieser mehr Platz auf der Ellipse zugesprochen. Hat eine Seite deutlich zu viele Elemente, so werden die Objekte mit einem größeren Abstand zur Aktivität platziert.

Die automatische Berechnung von sinnvollen Bendpoints in größeren Modellen hat nicht in einem zufriedenstellenden Maß funktioniert. Daher werden in der derzeitigen Implementierung Bendpoints nicht automatisch berechnet und der Benutzer sollte hinterher noch manuell Anpassungen vornehmen.

Kapitel 4

Durchführung

In diesem Kapitel wird auf Details der Implementierung eingegangen, sowie die Zusammensetzung der Klassen des FLOW-Explorers erläutert.

4.1 Klassen

Der FLOW-Explorer teilt sich in die fünf Pakete *Editor*, *Model*, *Parts*, *Policies* und *Util* auf.

4.1.1 Editor

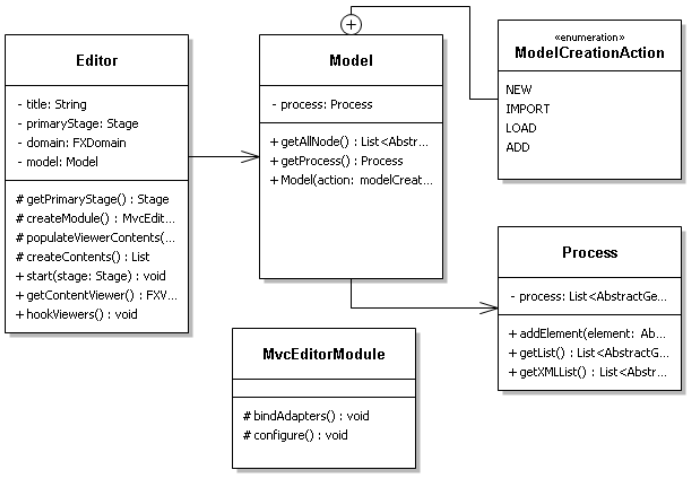


Abbildung 4.1: Klassendiagramm zum Editor-Paket

Das Editor-Paket enthält die Hauptklasse sowie einige Controller-Klassen. Alle Controller-Klassen implementieren das Singleton Design Pattern. Dies hat den Vorteil, dass sie von überall leicht erreichbar sind, und auch nicht mehr als einmal instantiiert werden können. Um Funktionen übersichtlich zu trennen, wurde für die meisten Funktionen ein eigener Controller erstellt.

Editor.java: Die Hauptklasse. Initialisiert das Modell, die Zeichenoberfläche sowie alle anderen Controller.
MvcEditorModule.java: Konfiguriert das MVC-Modul. Bindet Policies an ihre zugehörigen Elemente.
MenuController.java: Singleton, erstellt und kontrolliert die Benutzeroberfläche.
FileController.java: Singleton, zuständig für das Erstellen von neuen Dateien.
MergeController.java: Singleton, der MergeController ist zuständig für das Zusammenfügen von mehrfach vorkommenden Objekten

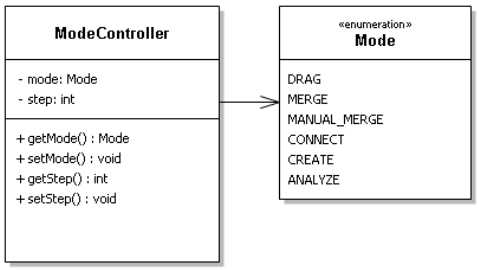


Abbildung 4.2: Klassendiagramm zur ModeController-Klasse

ModeController.java: Singleton, der ModeController speichert den aktuellen State des Programmes. Dieser setzt sich aus dem Modus sowie einem Integer zusammen, welcher verschiedene Schritte innerhalb eines Modus verfolgt. Die Policies aus dem Policy Paket reagieren anders je nach im ModeController eingestellten Modus.
ModelAnalyser.java: Beinhaltet Funktionen zur Analyse des FLOW-Modells
LayoutController.java: Singleton, Zuständig für das automatisierte anordnen der FLOW-Elemente.
DialogController.java: Singleton, Der DialogController ist zuständig für alle an den Benutzer angezeigten Pop-up Dialoge, wie z.B. das Laden und

Speichern von Dateien und Anzeigen von nützlichen Informationen.

4.1.2 Model

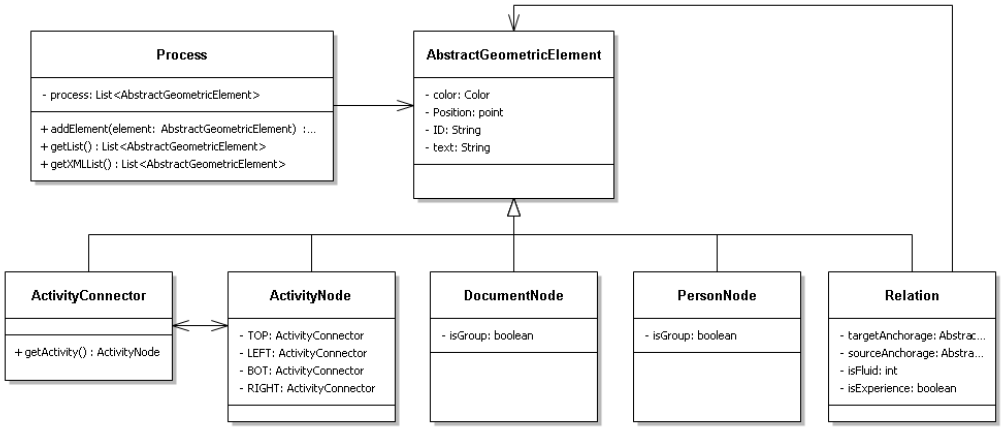


Abbildung 4.3: Klassendiagramm zum Model-Paket

Das Model Paket enthält die Klassen *AbstractGeometricElement*, *ActivityNode*, *ActivityConnector*, *DocumentNode*, *PersonNode*, *Relation* und *Model*. Die Model Klasse enthält einen Prozess, welcher Informationen über alle im derzeitig geladenen FLOW-Modell enthaltenen Objekte enthält, während die Klassen *ActivityNode*, *ActivityConnector*, *DocumentNode*, *PersonNode* und *Relation* Informationen über einzelne FLOW-Objekte enthalten. Der *ActivityConnector* stellt die vier Seiten einer Aktivität dar.

4.1.3 Parts

Analog zum Model-Paket enthält das Part-Paket folgende Klassen: *AbstractPart*, *ActivityConnectorPart*, *ActivityPart*, *DocumentPart*, *PersonPart*, *RelationPart*, *ModelPart* sowie die *ModelPartFactory*. Die *ModelPartFactory* ist zuständig für jedes im Model vorkommende FLOW-Objekt genau ein Part zu instanziieren. Ein Part ist ein Controller, welches jeweils ein visuelles FLOW-Objekt kontrolliert, während die verbundenen Model-Klassen jeweils das Modell des jeweiligen Parts enthält. Die Parts sind in einer Baumstruktur angeordnet. Das *ModelPart* ist die Wurzel, und alle anderen Parts sind dessen Kinder.

4.1.4 Policies

Das Policies-Paket beinhaltet Klassen, welche für die Interaktion mit Parts zuständig sind. Sie nehmen die Maus und Tastatur Inputs an und entscheiden, wie die Parts darauf reagieren. Enthaltene Klassen sind z. B. *NodeOnClickPolicy* und *NodeOnTypePolicy*, welche entscheiden, was passiert, wenn ein Element angeklickt wird, bzw. wenn eine Tastatureingabe gesendet wird, während ein Element selektiert it.

4.1.5 Util

Das Util-Paket enthält zusätzliche Utilities wie den XML-Parser, welcher für den Import von ProFLOW-Modellen sowie für das Speichern und Laden von FLOW-Explorer XML Dateien zuständig ist. Außerdem enthält es eine Klasse ShapeBuilder, welche die Formen der FLOW-Objekte erstellt.

4.2 Probleme und Änderungsmaßnahmen

Die Einarbeitung in das GEF4 Framework hat etwas länger gedauert als erwartet, da vorhandene GEF4 Dokumentationen teils nicht mehr aktuell waren und die Funktionen mittlerweile teils anders implementiert sind. Wie bei der Wahl des Frameworks erwähnt, war dies ein berücksichtigtes Risiko und es wurde daher etwas zusätzliche Zeit für diesen Fall allokiert. Leider gab es zwischendurch auch nach der Einarbeitung immer wieder kleinere Probleme in der Implementierung. Diese sind ebenfalls auf veralterte, sowie unausführliche Dokumentation zurückzuführen. Zusätzlich wurden von GEF4 teilweise inkorrekte Fehlermeldungen ausgegeben, welches die Lösung dieser Probleme komplizierter machte. Daher haben diese Probleme zusätzliche Zeit in Anspruch genommen. Um den dadurch entstandenen Zeitverlust wieder gut zu machen, wurden kleinere Kompromisse eingegangen. So wurde unter Anderem die Implementierung der Undo-Funktion vorerst zurückgestellt. Am Ende wurde eine Undo-Funktion kurzfristig implementiert und funktioniert auch bei einfachen Operationen. Bei komplexeren Operationen sind allerdings noch Fehler aufgetreten, welche sich in verbleibender Zeit nicht mehr haben lösen lassen. Daher wurde dieses Feature für die finale Version deaktiviert und muss gegebenenfalls hinterher noch nachgebessert werden, bevor es wieder aktiviert werden kann. Während der Bearbeitung haben sich auch einige der auftretenden Probleme durch kontinuierliche Updates des GEF4 Frameworks lösen lassen. Trotz der durch Nutzung von GEF4 auftretenden Probleme, war die Entscheidung, GEF4 zu benutzen, keine Fehlentscheidung. Der durch die Probleme entstandene Schaden konnte zwar in der Bearbeitungszeit nicht

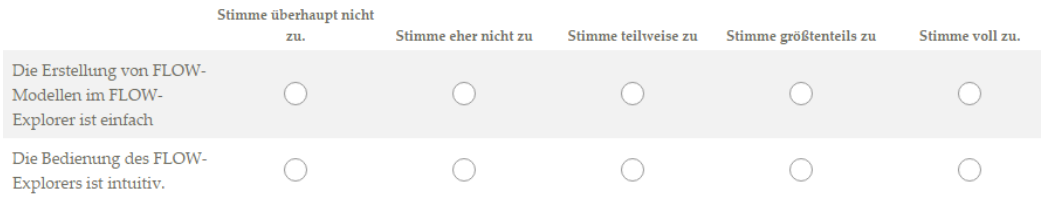
komplett entfernt werden, beeinträchtigt das Arbeiten mit dem FLOW-Explorer aber nur bedingt.

Kapitel 5

Evaluation

Zur Evaluation wurde der FLOW-Explorer im Rahmen einer Umfrage mit einem Szenario getestet. Dieses konnte von den Probanden von zuhause aus am eigenen Computer durchgeführt werden. Dabei sollten die FLOW-Probanden einige Aufgaben mit dem FLOW-Explorer bearbeiten und hinterher Fragen zu diesen Aufgaben beantworten.
An der Umfrage haben 14 Benutzer teilgenommen, wobei vier Benutzer vorher schon mit FLOW-Modellen gearbeitet haben. Die anderen 10 Tester setzten sich aus Informatik-Studenten sowie aus Menschen, welche in der Softwarebranche arbeiten, aber vorher noch nicht mit FLOW gearbeitet haben, zusammen.
Es wurden mehrere verschiedene Aufgaben bearbeitet, welche in den folgenden Sektionen behandelt werden.

5.1 Erstellen und Bearbeiten von FLOW-Modellen



	Stimme überhaupt nicht zu.	Stimme eher nicht zu	Stimme teilweise zu	Stimme größtenteils zu	Stimme voll zu.
Die Erstellung von FLOW-Modellen im FLOW-Explorer ist einfach	○	○	○	○	○
Die Bedienung des FLOW-Explorers ist intuitiv.	○	○	○	○	○

Abbildung 5.1: Auszug aus der Umfrage

In dieser Aufgabe sollte ein simples FLOW-Modell mit Hilfe des FLOW-Explorers erstellt und bearbeitet werden. Ausgewertet wurde, wie

simpel und intuitiv die Erstellung von FLOW-Modellen im FLOW-Explorer empfunden wird. Die Probanden sollten dabei verschiedene Aussagen auf einer Skala von eins bis fünf bewerten, wobei fünf für volle Zustimmung, und eins für überhaupt keine Zustimmung steht. Probanden haben größtenteils angegeben, dass die Erstellung von FLOW-Modellen im FLOW-Explorer einfach gestaltet ist und wurde im Durchschnitt mit 3,29 von 5 Punkten bewertet. Die Bedienung des FLOW-Explorers wurde eher als intuitiv bewertet, wobei die Ergebnisse im einzelnen durchaus unterschiedliche Resonanzen darbot, fiel die durchschnittliche Gesamtbewertung mit 3,07 von 5 Punkte aus.

Abbildung 5.2: Ergebnis: Erstellen und Bearbeiten

Kriterium	1	2	3	4	5	∅
Das Erstellen und Bearbeiten ist einfach	0	3	6	3	2	3,29
Die Bedienung ist intuitiv	1	4	3	5	1	3,07

Durch Feedback in den Kommentar Feldern stellte sich heraus, dass die Funktion zum Umbenennen als nicht intuitiv erachtet wurde, da Objekte nicht sofort nach ihrer Platzierung umbenannt werden konnten. Stattdessen müssen erst alle Objekte platziert und im Nachgang namentlich angepasst werden. Dies wurde in der finalen Version bereits kurzfristig nachgebessert. Nun lassen sich Objekte auch im Platzierungsmodus umbenennen, außerdem wird der Platzierungsmodus nur dann verlassen, wenn bei der Platzierung nicht gleichzeitig die SHIFT-Taste gedrückt wird.

5.2 Mehrere Modelle laden

In der nächsten Aufgabe waren zwei Modelle zu laden und zusammen in eine Datei abzuspeichern. Hierbei sollte festgestellt werden, für wie sinnvoll die automatische Platzierung der zusätzlich geladenen Modelle empfunden wird. Die Probanden bewerteten dieses Feature als nützlich(3,89/5), während die automatische Platzierung der hinzu geladenen Modelle nur als teilweise sinnvoll bewertet wurde.(3,43/5)
Kommentare ergeben, dass die Platzierung zwar nicht schlecht ist, aber dass es teilweise bessere Möglichkeiten zur Platzierung gegeben hätte.

Abbildung 5.3: Ergebnis: Mehrere Modelle laden. Hohe Werte sind besser

Kriterium	1	2	3	4	5	∅
Das Feature ist nützlich	0	1	3	7	3	3,89
Die Bedienung ist intuitiv	0	0	4	8	2	3,86
Die Platzierung ist sinnvoll	0	0	9	4	1	3,43

5.3 Automatisches Zusammenfügen von FLOW-Objekten

In dieser Aufgabe sollte das Feature zum automatischen Zusammenfügen von FLOW-Objekten bewertet werden. Dazu sollten die Tester ein vorher erstelltes FLOW-Modell laden und in diesem alle Elemente, welche mehrfach vorkommen zusammenfügen. Diese Aufgabe sollte zuerst ohne Nutzen des Werkzeugs, welche diesen Prozess automatisiert durchgeführt werden. Hinterher sollte sie mit dem automatischen Werkzeug wiederholt werden. Dazu wurden jeweils die gleichen drei Fragen gestellt. Ein niedriger Wert stellt ein besseres Ergebnis dar.

Abbildung 5.4: Vergleich: Zusammenfügen von Redundanzen. Niedrige Werte sind besser

Kriterium	Modus	∅
Ich habe viele Fehler gemacht	von Hand	2,38
	Automatisch	1,93
Ich habe leicht die Übersicht verloren	von Hand	3,14
	Automatisch	2,21
Die Zusammenfügung war kompliziert	von Hand	3,64
	Automatisch	2,69

Benutzer machten bei manueller Zusammenführung mehr Fehler (2,38/5) als bei Nutzung der automatischen Funktion (1,93/5). Die Übersicht wurde bei manueller Zusammenführung ebenfalls leichter verloren (3,14/5) als bei Nutzen der automatischen Funktion (2,21/5). Zudem wurde das manuelle Zusammenfügen als komplizierter (3,64) als das automatische Zusammenfügen empfunden (2,69/5).

Die Probanden stimmten zu, dass die automatische Funktion nützlich ist (3,64/5) und Zeit spart (3,58/5).

Abbildung 5.5: Ergebnis: Automatisches Zusammenfügen. Höhere Werte sind besser.

Kriterium	1	2	3	4	5	∅
Das Feature ist nützlich	0	1	5	6	2	3,64
Die Bedienung ist intuitiv	0	0	4	8	2	3,21
Das Feature erspart Zeit	0	1	4	6	1	3,58

Im Kommentar-Feld wurde Kritik darüber geäußert, dass bei größeren Modellen das Feature unübersichtlich wird, da man dabei zu weit raus zoomen müsste, und dann den Text nicht mehr richtig lesen könne.

5.4 Layout

Zuletzt sollte die Layout-Funktion bewertet werden. Dazu sollte das Modell, in welchem in der vorherigen Aufgabe Objekte zusammengefügt werden sollten, nun mit der automatischen Layout Funktion sortiert werden.

Abbildung 5.6: Ergebnis: Layout. Höhere Werte sind besser.

Kriterium	1	2	3	4	5	∅
Die Anordnung ist sinnvoll	5	4	3	2	0	2,14
Das Feature ist nützlich	2	8	2	0	2	2.43
Die Bedienung ist intuitiv	3	1	4	4	2	3,07
Das Feature erspart Zeit	1	4	7	1	1	2,79

Die Probanden bewerteten die Platzierung der Elemente als eher nicht sinnvoll (2,14/5). Sie stimmten eher nicht zu, dass durch Nutzen des Layout-Features Zeit gespart werden kann (2,79/5). Die Funktion wurde eher als teilweise intuitiv bewertet (3,07/5).
Der größte Kritikpunkt war, dass es noch teilweise Überlappungen gibt und das keine Bendpoints eingebaut werden.

Abbildung 5.7: Ergebnis: Verbesserungspotenzial

Kriterium	1	2	3	4	5	∅
Ich würde den FLOW-Expl. wieder verwenden	0	4	6	3	0	2,92
Es existiert noch Bedarf zur Verbesserung	0	2	3	5	2	3,58

5.5 Verbesserungspotenzial

Zuletzt wurden die Probanden gefragt, ob sie den FLOW-Explorer wieder verwenden würden und ob noch Bedarf zur Verbesserung besteht. Auf einer Skala von 1 bis 10 wurde der FLOW-Explorer im Durchschnitt mit 6,1 Sternen bewertet. Im letzten Kommentar-Feld wurden zudem einige kleine Verbesserungsvorschläge hinterlassen, welche, sofern möglich, umgesetzt wurden. So sind z. B. die Textfelder nun etwas größer, und der Speicher/Lade Dialog speichert das zuletzt benutzte Verzeichnis für die nächste Anwendung.

5.6 Auswertung & Interpretation

Aussagekräftigkeit der Ergebnisse:
Die Anzahl an Probanden, welche bereits Erfahrung mit FLOW-Modellen hatte, ist mit 4 Personen leider sehr niedrig. FLOW-Experten bewerteten die einzelnen Punkte auch sehr unterschiedlich, sodass hier in den meisten Aufgaben kein repräsentatives Ergebnis für die Gruppe der FLOW-Experten zu ermitteln war. Daher wurden alle Teilnehmer zusammen ausgewertet. Für 14 Teilnehmer ist es leichter einen repräsentativen Mittelwert zur Auswertung zu erhalten. Zudem stellt sich die Frage, wie nützlich die Meinungen von Personen, welche noch nie mit FLOW gearbeitet haben, für die Evaluierung des FLOW-Explorers sind.

Voreingenommenheit:
Es ist nicht auszuschließen, dass Probanden unterbewusst für die Wahl bestimmter Ergebnisse tendieren, da sie mich gegebenenfalls persönlich kennen. Außerdem ist es möglich, dass Probanden Aufgaben beantwortet haben, die sie nicht durchgeführt haben. Da es sich um einen Fern-Test handelt, existiert keine Möglichkeit zur Verifizierung der erhaltenen Ergebnisse.

Trotz der soeben genannten Punkte, waren die Ergebnisse der Umfrage größtenteils wie erwartet. Es zeigte sich, dass die Funktionen zum Zusammenfügen von Modellen und FLOW-Elementen nützlich sind und gegenüber

des manuellen Zusammenfügens Zeit sparen. Die Layout-Funktion hat das größte Potenzial zur Verbesserung und ist in seinem derzeitigen Zustand für die reale Anwendung eher weniger nützlich.

Kapitel 6

Zusammenfassung und Ausblick

6.1 Zusammenfassung

In dieser Arbeit wurde der FLOW-Explorer entwickelt, welcher ein neues Werkzeug zur Handhabung von FLOW-Modellen darstellt. Der FLOW-Explorer ermöglicht es nicht nur, FLOW-Modelle zu erstellen und zu editieren, sondern bietet auch unterstützende Werkzeuge zum Zusammenfügen von Objekten und Modellen sowie zur Analyse und Optimierung der Struktur. Der FLOW-Explorer erweitert dabei nicht einen bereits existierenden Editor, sondern wurde unter Verwendung des Graphical Editing Frameworks 4 neu entwickelt. Um Abwärtskompatibilität sicherzustellen, wurde zusätzlich zum neuen Speicherformat eine Import Funktion für ProFLOW-Modelle implementiert. Um bei der Optimierung von FLOW-Modellen Zeit zu sparen, wurden mehrere Funktionen zur Erleichterung des Zusammenfügens von Elementen implementiert. So können Elemente gleichen Typs mit Hilfe einer automatisieren Funktion, sowie manuell, zusammengefügt werden. Dazu wurde eine praktische Vorschauansicht entwickelt, welche Änderungen vor ihrer Durchführung visuell hervorhebt. Zusätzlich wurde eine Funktion entwickelt, welche alle Elemente nach gewissen Kriterien auf der Zeichenoberfläche sortieren kann. Zuletzt wurde der FLOW-Explorer anhand einer Reihe von Programmtests, welche mit einer Benutzergruppe durchgeführt wurden, evaluiert.

6.2 Ausblick

Der in dieser Arbeit entwickelte FLOW-Explorer hat noch viele Potenziale zur Weiterentwicklung. Da GEF4 auf JavaFX basiert, ist es gegebenenfalls möglich diesen auf Tablets zu portieren, was für das effiziente Erstellen

von FLOW-Modellen einen mobilen Vorteil ermöglicht. Auch lässt sich auf die bestehenden Funktionen noch weiter aufbauen. So könnte die Layout-Funktion erweitert und verbessert werden, indem ein besserer Algorithmus entworfen wird, welcher Wegpunkte/Bendpoints von Informationsflüssen optimiert. Zusätzlich könnten mehrere verschiedene neue Layoutalgorithmen entworfen werden, welche nach anderen Kriterien sortieren. Außerdem ist noch Raum zur Erkennung mehrerer verschiedener FLOW-Patterns.

6.3 Fazit

Die Evaluation des FLOW-Explorers hat gezeigt, dass der FLOW-Explorer in seinen Grundfunktionen gut ist und dass z. B. mit der Funktion zum Zusammenfügen Zeit gespart werden kann. Er hat in manchen Bereichen allerdings noch Bedarf zur Verbesserung. Die Layout-Funktion liefert noch kein zufriedenstellendes Ergebnis und sollte, um den FLOW-Explorer in der Zukunft mit all seinen Funktionen optimal verwenden zu können, nachgebessert werden. Außerdem ist es momentan nicht möglich, Operationen rückgängig zu machen, wodurch bei der Bedienung besondere Vorsicht geboten ist. Aufgrund der Evaluationsergebnisse und dem noch vorhandenen Verbesserungsbedarf, kann ich die Nutzung des FLOW-Explorers in seinem derzeitigen Zustand nur bedingt empfehlen. Sollten die genannten Mängel verbessert werden, so steht der Nutzung allerdings nichts mehr im Weg.

Literaturverzeichnis

[1] L. U. H. Fachgebiet Software Engineering. ProFLOW. `http://www.se.uni-hannover.de/pages/de:projekte_flow_proflow`. [zuletzt besucht am 23.08.2016].

[2] X. Ge. Flow Patterns: Beschreibung und Diskussion von Informationsflussmustern in der Softwareentwicklung. Masterarbeit, Leibniz Universität Hannover, Fachgebiet Software Engineering, 4 2008.

[3] E. Marques. Modellierungs- und Animationswerkzeug für quantitative Informationsflüsse in Softwareprojekten, 07 2005.

[4] A. Nyßen. GEF4. `https://wiki.eclipse.org/GEF/GEF4`. [zuletzt besucht am 11.09.2016].

[5] A. Nyßen. GEF4 MVC Komponente. `https://wiki.eclipse.org/GEF/GEF4/MVC`. [zuletzt besucht am 11.09.2016].

[6] F. Peschke. Durchgängige Werkzeugunterstützung für eine Consulting-Methode, 10 2012.

[7] K. Schneider. *Ausführbare Modelle der Software-Entwicklung. Struktur und Realisierung eines Simulationssystemes.* vdf, 1994.

[8] K. Stapel. Informationsflussoptimierung eines Softwareentwicklungsprozesses aus der Bankenbranche. Masterarbeit, Leibniz Universität Hannover, Fachgebiet Software Engineering, 11 2006.

[9] K. Stapel and K. Schneider. Flow-methode - Methodenbeschreibung zur Anwendung von FLOW. *CoRR*, abs/1202.5919, 2012.