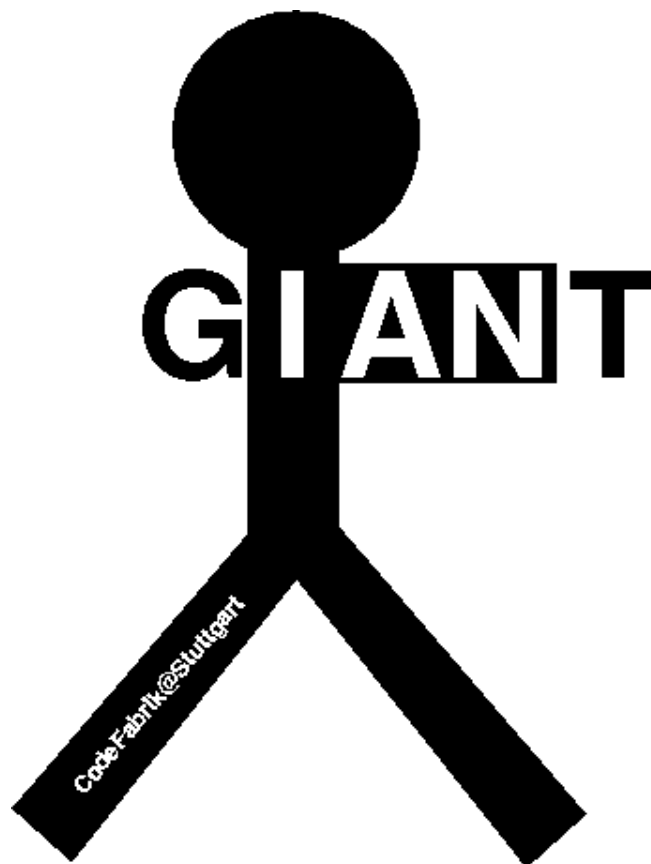


Spezifikation

Version 1.1



Versionsgeschichte

- Version 1.0 (08.04.2003)

Diese Version wurde dem Kunden zur Prüfung im Kundenreview (14.04.2003 - 16.04.2003) vorgelegt.

- Version 1.1 (22.04.2003)

Die Korrekturen gemäß dem Kundenreview vom 14.04.2003 bis zum 16.04.2003 wurden eingearbeitet.

Die Spezifikation der GSL (in Version 1.0 noch GQSL) wurde in ein separates Dokument ausgelagert (siehe Kapitel [10](#)).

Diese Version der Spezifikation ging am 22.04.2003 zur Abnahme an den Kunden.

Inhaltsverzeichnis

1. Einleitung	9
1.1. Über dieses Dokument	9
1.2. Über GIANT – Graphical IML Analysis Navigation Tool	9
2. Produktübersicht	11
2.1. GIANT Projekte	11
2.2. IML-Teilgraphen und Selektionen	11
2.3. Anzeigefenster	12
2.4. Knoten-Annotationen	12
2.5. Anfragen	12
2.6. Drei-Stufen-Konzept	13
3. Allgemeine Funktionale Anforderungen	15
3.1. Fehlerverhalten	15
3.2. Grundlegendes Feedbackverhalten	16
3.3. Zulässige Eingaben	16
3.4. Allgemeine Bedienkonzepte	17
3.5. Verhalten beim Einfügen von IML-Teilgraphen und Selektionen in Anzeigefenster .	19
3.6. Verhalten bei Umwandeln von Selektionen und IML-Teilgraphen	20
3.7. Verhalten beim Entfernen von Fenster-Knoten und Fenster-Kanten	20
3.8. Auseinanderschieben von Fenster-Knoten	21
4. Funktionale Anforderungen	23
4.1. Der Aktor „Benutzer“	23
4.2. Starten von GIANT	23
4.3. Beenden von GIANT	25

4.4.	UC: Neues Projekt	26
4.5.	UC: Projekt öffnen	28
4.6.	UC: Projekt speichern	30
4.7.	UC: Projekt speichern unter	31
4.8.	UC: Leeres Anzeigefenster erzeugen	33
4.9.	UC: Anzeigefenster öffnen	34
4.10.	UC: Anzeigefenster umbenennen	35
4.11.	UC: Anzeigefenster speichern	36
4.12.	UC: Anzeigefenster schließen	37
4.13.	UC: Anzeigefenster löschen	39
4.14.	UC: IML-Teilgraph in Anzeigefenster einfügen	40
4.15.	UC: Selektion in Anzeigefenster einfügen	42
4.16.	UC: Alle Fenster-Knoten und Fenster-Kanten einer Selektion aus einem Anzeigefenster löschen	44
4.17.	UC: Den Visualisierungsstil eines Anzeigefensters ändern	46
4.18.	UC: Anzeigefenster scrollen	47
4.19.	UC: Anzeigefenster zoomen	48
4.20.	UC: Zoomen auf eine gesamte Selektion	49
4.21.	UC: Zoomen auf den gesamten Inhalt eines Anzeigefensters	50
4.22.	UC: Zoomen auf eine Kante	51
4.23.	UC: Verschieben von Fenster-Knoten und Selektionen mittels Cut and Paste	52
4.24.	UC: Verschieben einzelner Fenster-Knoten mittels Drag and Drop	54
4.25.	UC: Platz schaffen	55
4.26.	UC: Pin anlegen	57
4.27.	UC: Pin anspringen	58
4.28.	UC: Pin löschen	59
4.29.	UC: Layout auf Selektion anwenden	60
4.30.	UC: Details zu Knoten in einem neuen Informationsfenster anzeigen	62
4.31.	UC: Details zu Knoten in einem bestehenden Informationsfenster anzeigen	63
4.32.	UC: Anzeige des Quellcodes eines Knotens in externem Editor	64
4.33.	UC: Verschieben des sichtbaren Anzeigehaltes über die Minimap	65

5. Knoten-Annotationen	67
5.1. UC: Fenster-Knoten annotieren	67
5.2. UC: Knoten-Annotation ändern	69
5.3. UC: Knoten-Annotation löschen	70
5.4. UC: Nicht referenzierte Knoten-Annotationen löschen	71
6. Selektionen	73
6.1. UC: Selektion zur aktuellen Selektion machen	73
6.2. UC: Selektion graphisch hervorheben	75
6.3. UC: Graphische Hervorhebung einer Selektion aufheben	76
6.4. UC: Neue Selektion anlegen	77
6.5. UC: Selektion kopieren	78
6.6. UC: Selektion löschen	79
6.7. UC: Selektionen manuell modifizieren	80
6.8. UC: Selektion aus IML-Teilgraph erzeugen	81
6.9. UC: Mengenoperationen auf 2 Selektionen	82
7. Filter	85
7.1. UC: Selektionen ausblenden	85
7.2. UC: Selektionen einblenden	87
7.3. UC: Alles einblenden	88
8. Teilgraphen	89
8.1. UC: IML-Teilgraph graphisch hervorheben	89
8.2. UC: Graphische Hervorhebung von IML-Teilgraphen aufheben	91
8.3. UC: IML-Teilgraph aus einer Selektion erzeugen	92
8.4. UC: IML-Teilgraph kopieren	93
8.5. UC: IML-Teilgraph löschen	94
8.6. UC: Mengenoperationen auf 2 IML-Teilgraphen	95

9. Skripte	97
9.1. UC: Neues Skript ausführen	97
9.2. UC: Skript laden	99
9.3. UC: Skript speichern	101
10. GIANT Scripting Language	103
10.1. GQSL	103
10.2. GSL Beschreibung	103
11. Layoutalgorithmen	105
11.1. Treelayout	105
11.2. Matrixlayout	106
12. GIANT Projektverwaltung	107
12.1. Persistenz der Projekte	107
12.2. Grundlegendes Verhalten von GIANT beim Speichern von Projekten	110
13. Konfiguration von GIANT	113
13.1. Allgemeines	113
13.2. Die globale Konfigurationsdatei	113
13.3. Visualisierungsstile	114
13.4. Skript-Dateien	116
14. Beschreibung der GUI	117
14.1. Über die Benutzeroberfläche	117
14.2. Über dieses Kapitel	117
14.3. Main Window	118
14.4. Anzeigefenster	122
14.5. Knoten-Informationsfenster	126
14.6. Skriptdialog	128
14.7. Allgemeiner Texteingabedialog	129
14.8. Set-Operation-Dialog	129
14.9. Layoutalgorithmen Dialog	131

14.10. Dateneingabe	133
14.11. Ausgabe von Fehlermeldungen	134
14.12. Sicherheitsabfrage	134
14.13. Auswahl von Dateien	135
14.14. Fadenkreuz-Cursor	135
14.15. Fortschrittsanzeige	136
15. Visualisierung des IML-Graphen	139
15.1. Visualisierung von Knoten	139
15.2. Visualisierung von Kanten	140
15.3. Hervorheben von Fenster-Knoten und Fenster-Kanten	140
15.4. Detailstufen beim Zoomen	141
15.5. Minimap	142
16. Nichtfunktionale Anforderungen	143
16.1. Plattformunabhängigkeit	143
16.2. Wartbarkeit	143
16.3. Portabilität	143
16.4. Mengengerüst	144
16.5. Robustheit	144
16.6. Robustheit gegenüber Änderungen der IML-Graph-Spezifikation	144
16.7. Leistungsanforderungen	144
16.8. Antwortverhalten	145
16.9. Sicherheit	145
16.10. Erweiterbarkeit	145
17. Technische Produktumgebung	147
17.1. Software	147
17.2. Hardware	147
17.3. Produkt-Schnittstellen	148

18. Anforderungen an die Umgebung	149
18.1. Compiler und Bibliotheken	149
18.2. Einlesen und Schreiben von XML Dateien	149
18.3. Sprache	149
19. Begriffslexikon	151
20. Index	155

1. Einleitung

1.1. Über dieses Dokument

Diese Spezifikation beschreibt die funktionalen und die nicht-funktionalen Anforderungen an den IML-Browser GIANT sowie die Rahmenbedingungen, unter denen GIANT lauffähig sein muss.

Dieses Dokument ist Grundlage für die weitere Entwicklung von GIANT innerhalb dieses Projekts, das heißt im Besonderen für das Benutzerhandbuch, den Entwurf, die Implementierung und den Test des Softwaresystems. Das Dokument ist an die Mitglieder der Projektgruppe sowie an den Kunden und dessen technische Berater gerichtet.

Dieses Dokument ist Vertragsbestandteil für die weitere Entwicklung von GIANT und somit auch Grundlage für die Abnahme des fertigen Produktes.

1.2. Über GIANT – Graphical IML Analysis Navigation Tool

GIANT ist ein Werkzeug, welches an der Universität Stuttgart von der CodeFabrik@Stuttgart im Rahmen des Studienprojektes A IML-Browser des Studiengangs Softwaretechnik entwickelt wird.

Ziel dieser Entwicklung ist es, die bereits bestehende HTML-basierte Lösung der Bauhaus Reengineering GmbH für IML-Graphen durch ein komfortables graphisches Werkzeug zu ergänzen. GIANT soll es dem Kunden ermöglichen, Teile von großen IML-Graphen zu visualisieren. Durch die Unterscheidung der verschiedenen Kanten- und Knotenklassen sollen darüber hinaus auch im IML-Graphen enthaltene Analyseergebnisse übersichtlich dargestellt werden können.

Das Produkt soll sich durch einen hohen Grad an Wartbarkeit auszeichnen. Dies soll insbesondere dadurch erreicht werden, dass die Anbindung an die Bauhaus-IML-Graph-Bibliothek derart vorgenommen wird, dass Änderungen in der Spezifikation des IML-Graphen, wie z.B. das Einbringen neuer Attribute, möglichst keine Wartungsarbeiten an GIANT selbst zur Folge haben.

2. Produktübersicht

In diesem Kapitel werden einige grundlegende Konzepte von GIANT vorgestellt und kurz beschrieben. Dieses Kapitel stellt lediglich eine kurze Übersicht dar, die hier vorgestellten Konzepte werden weiter hinten in diesem Dokument im Detail spezifiziert – siehe hierzu auch die Verweise innerhalb dieses Kapitels.

2.1. GIANT Projekte

Innerhalb eines Projektes fasst GIANT Informationen wie Anzeigefenster und IML-Teilgraphen für einen vorgegebenen IML-Graphen zusammen und speichert diese persistent. Die Zusammenfassung zu Projekten soll der Übersicht dienen und den Austausch von Teilergebnissen, wie z.B. einzelner Anzeigefenster, erleichtern.

Für weitere Informationen zu Projekten siehe Kapitel [12](#).

2.2. IML-Teilgraphen und Selektionen

Hinsichtlich der logischen Zusammenfassung von IML-Knoten und IML-Kanten unterscheidet GIANT IML-Teilgraphen und Selektionen.

2.2.1. IML-Teilgraphen

Ein IML-Teilgraph ist eine Knoten- und Kantenmenge aus dem IML-Graphen mit der Bedingung, dass die Start- und Zielknoten jeder Kante Teil der Menge sind. Diese sogenannten Graph-Knoten und Graph-Kanten können in Anzeigefenstern hervorgehoben werden. Des weiteren können die Graph-Knoten und Graph-Kanten von IML-Teilgraphen als Fenster-Knoten und Fenster-Kanten in Anzeigefenster eingefügt und layoutet werden. Die IML-Teilgraphen sind global in der Anwendung verfügbar, aber völlig unabhängig von den Anzeigefenstern und enthalten insbesondere keine Layoutinformationen.

2.2.2. Selektionen

Selektionen stellen eine Menge von Fenster-Knoten und Fenster-Kanten dar. Selektionen sind immer einem festen Anzeigefenster zugeordnet und umfassen nur Fenster-Knoten und Fenster-Kanten des Anzeigehaltes dieses Anzeigefensters. Die Fenster-Knoten und Fenster-Kanten einer Selektion

müssen keinen Teilgraphen bilden, sie dürfen also auch Fenster-Kanten ohne die zugehörigen Start- und Zielknoten umfassen.

2.3. Anzeigefenster

Anzeigefenster sind die Fenster von GIANT, in denen eine benutzerdefinierte Auswahl von Fenster-Knoten und Fenster-Kanten visualisiert wird. Es kann beliebig viele Anzeigefenster geben und jedes Anzeigefenster kann beliebig viele Selektionen haben.

2.3.1. Pins

Da bei großen Graphen selten alle zu einem Anzeigeeintrag gehörenden Fenster-Knoten und Fenster-Kanten gemeinsam auf dem Bildschirm sichtbar dargestellt werden können, kann sich der Benutzer zu jedem Anzeigefenster eine Liste von Pins anlegen. In den Pins wird jeweils die Position des sichtbaren Anzeigeeintrages und die Zoomstufe gespeichert, so dass zu beliebigen Zeitpunkten die Position des sichtbaren Anzeigeeintrages rekonstruiert werden kann.

2.3.2. Visualisierungsstile

Mittels sogenannter Visualisierungsstile kann der Benutzer die Darstellung von Fenster-Kanten und Fenster-Knoten auch während der Laufzeit von GIANT beeinflussen. Da über entsprechende XML-Dateien verschiedene Visualisierungsstile definiert werden können, kann GIANT so an spezifische Problemstellungen angepasst werden. Weitere Informationen hierzu sind in Abschnitt [13.3](#) zu finden.

2.4. Knoten-Annotationen

Jeder Knoten kann mit einer textuellen Annotation versehen werden. Diese Annotation kann in einem Fenster außerhalb des Anzeigefensters zur Anzeige gebracht und bearbeitet werden. Für weitere Informationen zu diesem Thema siehe Abschnitt [12.2.4](#).

2.5. Anfragen

Eine vielseitige Anfragesprache – die GIANT Scripting Language GSL – stellt nahezu die gesamte Funktionalität von GIANT zur Verfügung und kann insbesondere auch zum Aufruf via Kommandozeile genutzt werden. Die drei Schritte des anschließend beschriebenen „Drei-Stufen-Konzeptes“ können über diese Anfragesprache auch „auf einen Schlag“ erledigt werden. Die GSL ist unter Kapitel [10](#) im Detail spezifiziert.

2.6. Drei-Stufen-Konzept

Die im Folgenden beschriebenen drei Schritte sollen dem Benutzer die Möglichkeit bieten, von den IML-Knoten und IML-Kanten des IML-Graphen ausgehend geeignete IML-Teilgraphen in Anzeigefenstern zu visualisieren. Die Funktionalität von GIANT ist so konzipiert, dass zur Visualisierung von IML-Graphen in Anzeigefenstern diese drei Schritte sequentiell nacheinander ausgeführt werden. Mittels der Anfragesprache GSL (siehe 10) und mittels der über die GUI zugänglichen UseCases (beschrieben ab Kapitel 4) kann der Benutzer diese Schritte wahlweise einzeln „Step by Step“ oder auch „auf einen Schlag“, d.h. durch Eingabe einer einzigen Anfrage, ausführen.

Der „Visualisierungsvorgang“ von IML-Graphen ist aber rein logisch betrachtet in die folgenden drei Schritte unterteilt:

1. Erzeugen geeigneter IML-Teilgraphen, d.h. Auswahl geeigneter IML-Knoten und IML-Kanten aus dem IML-Graphen mittels der Anfragesprache GSL.
2. Einfügen dieser IML-Teilgraphen in ein Anzeigefenster unter Anwendung eines Layoutalgorithmus.
3. Weitere Bearbeitung der Fenster-Knoten und Fenster-Kanten (wie z.B. Verschieben, Annotieren und Erzeugen von Selektionen).

3. Allgemeine Funktionale Anforderungen

Dieses Kapitel beschreibt die funktionalen Anforderungen an GIANT, die das System als Ganzes und nicht nur einzelne UseCases betreffen. Dies sind z.B. allgemeine Bedienkonzepte zur Selektion von Fenster-Knoten und Fenster-Kanten, sowie das grundlegende Verhalten von GIANT im Fehlerfall.

3.1. Fehlerverhalten

Beim Auftreten von Fehlern während des Betriebs von GIANT werden die folgenden Konventionen eingehalten. GIANT verhält sich bezüglich der Fehlerbehandlung konsistent. Das im Folgenden beschriebene Fehlerverhalten gilt daher für das gesamte System. Abweichungen von diesem Verhalten werden gegebenenfalls an der entsprechenden Stelle explizit spezifiziert.

1. Fehlermeldungen werden grundsätzlich über einen allgemeinen Fehlerdialog ausgegeben (siehe auch [14.11](#)). Ist dies nicht möglich, wird die Fehlermeldung über die Kommandozeile ausgegeben.
2. Sollten mehrere Fehler, wie z.B. Eingabefehler, gleichzeitig auftreten, so werden die entsprechenden Fehlermeldungen falls möglich über einen einzigen allgemeinen Fehlerdialog „auf einen Schlag“ an den Benutzer ausgegeben.

3.1.1. Ungültige oder unvollständige Eingaben bei Dialogen

1. Eingaben innerhalb von Dialogen werden immer dann geprüft, wenn der Benutzer alle Eingaben in dem Dialogfenster durch Klicken auf einen „OK-Button“ bestätigt.
2. Stellt das System dann eine unzulässige oder fehlende Eingabe fest, so erscheint der allgemeine Fehlerdialog mit einem Hinweis, welche Daten falsch eingegeben wurden bzw. fehlen.
3. Nach Schließen des Fehlerdialoges kehrt das System zu der Stelle, die im Ablauf unmittelbar vor der Fehlerüberprüfung lag, zurück. Also typischerweise zu dem Dialog, in dem die fehlerhafte Eingabe getätigt wurde. Die zuvor in den jeweiligen Dialogen getätigten Eingaben des Benutzers bleiben erhalten.

3.1.2. Ungültige oder unvollständige Eingaben bei Kommandozeilenaufruf

Gibt der Benutzer beim Start von GIANT über Kommandozeile ungültige oder unvollständige Eingabedaten (z.B. für die Kommandozeilenparameter) an, so wird der Start von GIANT abgebrochen.

Auf der Kommandozeile wird dann eine entsprechende Fehlermeldung mit Nennung der fehlenden oder ungültigen Daten ausgegeben.

3.1.3. Fehler beim Einlesen von Dateien

1. Können zum Betrieb von GIANT nötige Dateien (wie z.B. die Konfigurationsdateien) nicht an der dafür vorgesehenen Position gefunden werden, so wird das System mit einer entsprechenden Fehlermeldung beendet. In dieser Fehlermeldung werden die nicht gefundenen Dateien und der Pfad, an dem diese eigentlich zu finden sein müssten, angegeben.
2. Das Verhalten bei allen anderen möglichen Fehlern, die im Zusammenhang mit dem Zugriff auf Dateien auftreten können (fehlende Zugriffsrechte, defekte Sektoren etc.), ist undefiniert.

3.1.4. Einlesen unzulässiger Daten aus Dateien

1. GIANT verfügt über keine Mechanismen, die speziell darauf ausgelegt sind, die Korrektheit von Daten, die aus Dateien gelesen werden, zu prüfen.
2. Stellt das System dennoch fest, dass eine einzulesende Datei ungültige Daten enthält, so wird das System mit einer entsprechenden Fehlermeldung beendet.

3.2. Grundlegendes Feedbackverhalten

Während der Berechnung durch Layoutalgorithmen und Anfragen informiert GIANT den Benutzer laufend, dass es gerade mit Berechnungen beschäftigt ist. Dies kann über eine der beiden hier beschriebenen Möglichkeiten geschehen:

1. Anzeige eines Dialoges mit einem sich ständig leicht ändernden Inhalt, so dass erkennbar ist, dass das System noch läuft.
2. Anzeige eines Progressbars, der über den aktuellen Fortschritt einer Berechnung informiert, ohne dabei den Gesamtaufwand zu kennen. Dieser Progressbar protokolliert also nur die bisher geleistete Arbeit (z.B. die Anzahl der Knoten, welche bereits layoutet wurden; siehe hierzu auch [14.15](#)).

3.3. Zulässige Eingaben

Hier wird beschrieben, welche Eingaben für Namen etc. innerhalb von GIANT (z.B. bei den entsprechenden Texteingabedialogen) zulässig sind. Unzulässige Namen werden generell abgelehnt und führen zu einer Fehlermeldung.

3.3.1. Bezeichnertypen

Hier werden Bezeichnertypen, also von GIANT als zulässig akzeptierte Folgen von Ziffern, Buchstaben und Sonderzeichen definiert.

3.3.1.1. Der String „STANDARD NAME“

Hier wird ein String spezifiziert, der nur die im Folgenden beschriebenen Zeichen enthalten darf. Zulässige Zeichen sind:

1. Die lateinischen Groß- und Kleinbuchstaben; nicht zulässig sind Umlaute und das Zeichen "ß". Zulässig sind also nur die ASCII-Zeichen 65 bis 90 und 97 bis 122 (gemäß Spezifikation der ISO-8859-Zeichensätze).
2. Die Ziffern "0" bis "9" (ASCII-Zeichen 48 bis 57)
3. Das Zeichen "_" (ASCII-Zeichen 95).

3.3.2. Zulässige Namen

Für Namen, die der Benutzer z.B. für Projekte vergeben darf, wird hier der jeweils zulässige Bezeichnertyp (siehe 3.3.1) spezifiziert.

1. Name eines Projektes: STANDARD NAME
2. Name eines IML-Teilgraphen: STANDARD NAME
Dieser Name muss für jeden IML-Teilgraphen innerhalb eines Projektes eindeutig sein.
3. Name eines Anzeigefensters: STANDARD NAME
Dieser Name muss für jedes Anzeigefenster innerhalb des Projektes eindeutig sein.
4. Name einer Selektion: STANDARD NAME
Dieser Name muss für jede Selektion innerhalb eines Anzeigefensters eindeutig sein.
5. Name eines Pins: STANDARD NAME
Dieser Name muss für jeden Pin innerhalb eines Anzeigefensters eindeutig sein.

3.4. Allgemeine Bedienkonzepte

Hier werden grundlegende Konzepte der Bedienung von GIANT beschrieben.

3.4.1. Selektieren von Fenster-Knoten und Fenster-Kanten in Anzeigefenstern

Hier werden die Abläufe zur Selektion und Deselektion einzelner Fenster-Knoten und Fenster-Kanten spezifiziert.

1. Alle selektierten Fenster-Knoten und Fenster-Kanten deselektieren.
Wird mit der linken Maustaste auf einen Punkt des sichtbaren Anzeigeinhaltes geklickt, unter dem sich keine Fenster-Knoten bzw. Fenster-Kanten befinden, so werden alle bis dahin selektierten Fenster-Knoten und Fenster-Kanten deselektiert.
2. Selektieren durch Anklicken mit linker Maustaste.
Nur der angeklickte Fenster-Knoten oder die angeklickte Fenster-Kante wird selektiert. Andere bis dahin selektierte Fenster-Knoten und Fenster-Kanten werden deselektiert.
3. Selektieren durch Anklicken mittels Shift + linke Maustaste.
Falls noch nicht selektiert wird der angeklickte Fenster-Knoten oder die angeklickte Fenster-Kante zu der aktuellen Selektion hinzugefügt. Falls bereits selektiert wird der Fenster-Knoten oder die Fenster-Kante aus der aktuellen Selektion entfernt.
4. Selektieren durch Aufziehen eines Rahmens mittels gedrückter linker Maustaste.
Alle Fenster-Knoten und Fenster-Kanten innerhalb des Rahmens werden selektiert, alle Fenster-Knoten und Fenster-Kanten außerhalb des Rahmens werden deselektiert.
5. Selektieren durch Aufziehen eines Rahmens mittels Shift + gedrückte linke Maustaste.
Alle Fenster-Knoten und Kanten innerhalb des Rahmens, die noch nicht selektiert sind, werden der aktuellen Selektion hinzugefügt.
Alle Fenster-Knoten und Kanten innerhalb des Rahmens, die bereits selektiert sind, werden aus der aktuellen Selektion entfernt.
6. Selektieren durch Aufziehen eines Rahmens mittels Control + gedrückte linke Maustaste.
Alle Fenster-Knoten und Fenster-Kanten innerhalb des Rahmens, die noch nicht selektiert sind, werden der aktuellen Selektion hinzugefügt. Fenster-Knoten und Kanten innerhalb und außerhalb des Rahmens, die bereits selektiert sind, bleiben selektiert.

3.4.2. Standard-Selektion

Jedes Anzeigefenster hat genau eine Standard-Selektion. Diese Selektion wird bei der Erzeugung des Fensters angelegt und ist zu Anfang leer. Die Standard-Selektion kann nicht gelöscht werden. Abgesehen davon verhält sich die Standard-Selektion wie vom Benutzer angelegte Selektionen.

3.4.3. Aktuelle Selektion

1. Es gibt zu jedem Anzeigefenster beliebig viele Selektionen (mindestens aber eine – die Standard-Selektion), davon ist immer eine die aktuelle Selektion.
2. Das oben beschriebene Selektieren mittels der linken Maustaste und weiterer Funktionstasten (siehe 3.4.1) betrifft immer nur die aktuelle Selektion.
Alle anderen Selektionen können zwar hervorgehoben werden, aber nicht durch das Selektieren mittels linker Maustaste etc. geändert werden.
3. Der Benutzer kann jederzeit bestimmen, welche der Selektionen eines Anzeigefensters die aktuelle Selektion ist (siehe UseCase 6.1).
4. Die aktuelle Selektion wird immer hervorgehoben (siehe hierzu auch 15.3.1).

3.5. Verhalten beim Einfügen von IML-Teilgraphen und Selektionen in Anzeigefenster

In diesem Abschnitt wird der Einfügevorgang von IML-Teilgraphen und Selektionen in Anzeigefenster spezifiziert.

3.5.1. Vorgabe der Zielposition

Für bestimmte UseCases (siehe z.B. UseCase [4.14](#)) muss der Benutzer manuell Zielpositionen innerhalb eines zu bestimmenden Anzeigefensters auswählen. An dieser Zielposition werden dann z.B. neue Fenster-Knoten eingefügt. Ablauf:

1. Der Benutzer wählt das entsprechende Anzeigefenster aus (dadurch dass er ihm je nach Betriebssystemkonvention den Fokus gibt). Hierbei können nur bereits geöffnete Anzeigefenster des Projektes ausgewählt werden.
2. Befindet sich die Maus über dem Bereich des Anzeigefensters, wo der IML-Graph visualisiert wird, so wird anstatt des Mauszeigers ein Fadenkreuz (siehe auch [14.14](#)) angezeigt.
3. Durch Klicken mit der linken Maustaste wird dann über das Fadenkreuz die Zielposition vorgegeben.

3.5.2. Automatisches Erzeugen neuer Selektionen im Zielfenster

Sämtliche Fenster-Knoten und Fenster-Kanten, die in einem Schritt in ein Anzeigefenster eingefügt werden, werden dort automatisch zur Standard-Selektion, der alte Inhalt der Standard-Selektion geht dabei verloren und die Standard-Selektion bekommt den Status der aktuellen Selektion.

3.5.3. Einfügen von Selektionen in Anzeigefenster

Hier wird das grundlegende Verhalten beim Einfügen von Selektionen in Anzeigefenster (Kopieren einer Selektion aus einem Anzeigefenster in ein anderes) beschrieben.

1. Das Layout der Selektion wird beim Einfügen in ein Anzeigefenster übernommen.
2. Die Position von Fenster-Knoten der einzufügenden Selektion, die bereits in dem Anzeigefenster visualisiert sind, bleibt, falls nicht an entsprechender Stelle anders spezifiziert, unverändert.
3. Fenster-Kanten der Selektion werden nur eingefügt, falls ihr Start- und ihr Zielknoten ebenfalls Bestandteil der Selektion sind oder bereits im Anzeigefenster visualisiert sind.
4. Die neu eingefügte Selektion ist in dem entsprechenden Anzeigefenster (dort wo sie eingefügt wurde) als Standard-Selektion vorhanden, der alte Inhalt der Standard-Selektion wird überschrieben.

3.5.4. Einfügen von IML-Teilgraphen in Anzeigefenster

Hier wird das grundlegende Verhalten des Systems beim Einfügen von IML-Teilgraphen in Anzeigefenster beschrieben.

1. Das Layout für den IML-Teilgraphen wird vor dem Einfügen gemäß eines vom Benutzer vorgegebenen Layoutalgorithmus berechnet.
2. Die Graph-Knoten und Graph-Kanten des IML-Teilgraphen werden dann analog zu [3.5.3](#) als Fenster-Knoten und Fenster-Kanten in das Anzeigefenster eingefügt

3.6. Verhalten bei Umwandeln von Selektionen und IML-Teilgraphen

Dieser Abschnitt beschreibt grundlegende Konventionen für die Umwandlung von IML-Teilgraphen in Selektionen und umgekehrt.

3.6.1. Selektion aus IML-Teilgraphen ableiten

1. Der IML-Teilgraph bleibt unverändert.
2. Im zu bestimmenden Anzeigefenster wird eine neue Selektion erzeugt, die alle Graph-Knoten und Graph-Kanten des IML-Teilgraphen umfasst, welche bereits als Fenster-Knoten und Fenster-Kanten im Anzeigefenster vorhanden sind.
3. Knoten und Kanten des IML-Teilgraphen, die nicht als Fenster-Knoten und Fenster-Kanten im entsprechenden Anzeigefenster vorhanden sind, werden ignoriert.

3.6.2. IML-Teilgraph aus Selektion ableiten

1. Die Selektion bleibt unverändert.
2. Es wird ein neuer IML-Teilgraph erzeugt. Dieser umfasst alle Knoten und Kanten der Selektion, welche einen gültigen Teilgraphen bilden. Kanten der Selektion, deren Start- und Zielknoten nicht ebenfalls Bestandteil der Selektion sind, werden ignoriert.

3.7. Verhalten beim Entfernen von Fenster-Knoten und Fenster-Kanten

Beim Entfernen von Fenster-Knoten und Fenster-Kanten aus einem Anzeigefenster werden nur die graphischen Repräsentationen der IML-Knoten und IML-Kanten des Bauhaus-IML-Graphen in dem betroffenen Anzeigefenster gelöscht. Graph-Knoten und Graph-Kanten von IML-Teilgraphen bleiben davon unberührt.

3.7.1. Entfernen aller Knoten und Kanten einer Selektion

Beim Entfernen aller Fenster-Knoten und Fenster-Kanten einer Selektion gelten die folgenden Konventionen:

1. Alle Fenster-Knoten und Fenster-Kanten der Selektion werden aus dem Anzeigefenster entfernt.
2. Fenster-Kanten im Anzeigefenster, deren Start- oder Zielknoten entfernt wird, werden ebenfalls entfernt (auch wenn sie nicht zur entsprechenden Selektion gehören).
3. Andere Selektionen des Anzeigefensters werden entsprechend aktualisiert, die betroffenen Fenster-Knoten und Fenster-Kanten werden auch aus diesen Selektionen entfernt.

3.8. Auseinanderschieben von Fenster-Knoten

Sollen Fenster-Knoten auf dem Anzeigeinhalt automatisch auseinandergeschoben werden, so geschieht dies nach dem hier beschriebenen Modus. GIANT stellt diese Funktionalität dem Benutzer über einen UseCase zur manuellen Ausführung zur Verfügung. Ein automatisches „Auseinanderschieben von Fenster-Knoten“ durch GIANT selbst ist nicht vorgesehen, der Entwurf wird aber eine einfache Erweiterbarkeit von GIANT diesbezüglich unterstützen.

1. Es gibt einen fixen Punkt, von dem aus alle Fenster-Knoten weggeschoben werden.
2. Die Fenster-Knoten werden immer entlang einer Strecke von diesem Punkt durch den jeweiligen Fenster-Knoten verschoben.
3. Jeder Fenster-Knoten wird um einen konstanten Betrag (kann vom Benutzer eingegeben werden) von dem Punkt weggeschoben.

4. Funktionale Anforderungen

Dieses Kapitel beschreibt die funktionalen Anforderungen an GIANT in Form von UseCases, d.h. die dem Benutzer über die GUI von GIANT zugängliche Funktionalität wird in diesem Kapitel mittels UseCases spezifiziert.

4.1. Der Aktor „Benutzer“

Aktor im Sinne der anschließend spezifizierten UseCases ist immer die Person, die den IML-Browser GIANT gerade bedient, also der „Benutzer“. Neben dem „Benutzer“ sind keine weiteren Aktoren vorgesehen.

Zur Bedienung von GIANT muss der Benutzer über die folgenden Kenntnisse verfügen:

1. Erfahrung im Umgang mit graphischen Benutzeroberflächen und dem entsprechenden Betriebssystem.
2. Kenntnisse über Struktur und Aufbau des IML-Graphen.
3. Kenntnisse in der Skriptsprache GSL von GIANT.

4.2. Starten von GIANT

Der Benutzer startet das Programm GIANT durch Ausführen der entsprechenden Programmdatei. Hierbei hat er bezüglich eventueller Kommandozeilenparameter die folgenden Optionen.

4.2.1. Mögliche Parameter

GIANT kann folgendermaßen via Kommandozeile gestartet werden:

```
giant [(-n file -i file|-l file) [-e file]|-h]
```

Die Klammerung orientiert sich an den gängigen Linux-Konventionen, wie sie z.B. innerhalb von „Manpages“ verwendet werden.

Ausdrücke in eckigen Klammern „[...]“ sind optional, wobei Alternativen durch einen senkrechten

Strich „[A | B]“, getrennt werden. Es darf in diesem Fall also entweder der Ausdruck A oder der Ausdruck B oder gar nichts stehen.

Bei Ausdrücken in runden Klammern werden alternative Ausdrücke ebenfalls durch einen senkrechten Strich getrennt „(A | B)“, es muss aber immer genau einer der alternativen Ausdrücke (hier also entweder A oder B) verwendet werden.

Anstelle der hier angegebenen Kurzform kann natürlich auch die unten beschriebene Langform der Parameter verwendet werden.

Mit „file“ ist immer der qualifizierende Name für eine Datei gemeint, also der Dateiname samt Erweiterung und Pfad (gemäß den Konventionen des verwendeten Betriebssystems).

Anschließend werden nun die Parameter im Einzelnen erläutert:

1. `--newproject=file` oder `-n file`

Angabe des Pfades und des Namens für die neu zu erstellende Projektdatei. Die Datei darf nicht bereits vorhanden sein. Das Verzeichnis der Projektdatei – also das letzte Verzeichnis im Pfad – ist automatisch das Projektverzeichnis und muss bereits existieren. In diesem Verzeichnis darf sich keine weitere Projektdatei befinden.

`--iml=file` oder `-i file`

Angabe einer existierenden IML-Graph Datei als Grundlage für ein neu zu erstellendes Projekt.

2. `--loadproject=file` oder `-l file`

Angabe einer existierenden Projektdatei. Das entsprechende Projekt wird dann beim Start von GIANT geladen.

3. `--execute=file` oder `-e file`

Angabe einer Datei, in der ein GSL Skript gespeichert ist. Dieses GSL Skript wird dann beim Start von GIANT ausgeführt. Einem Skript können beim Kommandozeilenauf Ruf Parameter übergeben werden, wie dies genau geschieht, wird in der GSL Spezifikation erläutert.

4. `--help` oder `-h`

Zeigt eine Hilfe zu den möglichen Kommandozeilenparametern an.

4.2.2. Mögliche Aufrufe

Hier werden beispielhaft mögliche Parameterfolgen zum Start von GIANT via Kommandozeile erläutert.

1. Eingabe von: `giant`

GIANT startet und zeigt das Main Window an. Es ist noch kein Projekt geladen.

2. Eingabe von:

```
giant --newproject=/home/projekt/my_project_file.giant
      --iml=sample.iml
```


Ein neues Projekt wird geöffnet und erhält den Namen der Projektdatei „my_project_file.giant“ (ohne die Erweiterung). Das Verzeichnis „/home/projekt“ wird zum Projektverzeichnis

3. Eingabe von: `giant --loadproject=my_project_file.giant`
Das angegebene Projekt wird geöffnet.
4. Bei 2 und 3 kann zusätzlich der Parameter `--execute=file.script` eingegeben werden. In diesem Falle wird das übergebene GSL Skript ausgeführt.

4.3. Beenden von GIANT

Ablauf:

1. Der Benutzer wählt „Quit“ im Hauptmenü.
2. Ist ein Projekt geöffnet, so fragt GIANT nach, ob das Projekt gespeichert werden soll. Falls der Benutzer das Speichern ablehnt, so gehen alle nicht gespeicherten Informationen verloren. Entscheidet er sich für Speichern, so wird die unter [12.2.1](#) beschriebene Funktionalität ausgeführt.
3. GIANT wird beendet.

Muss GIANT aufgrund eines schwerwiegenden Fehlers beendet werden (man könnte hier auch von einem Programmabsturz reden), so bleibt der Zustand der Projektdateien unspezifiziert, schlimmstenfalls sind diese dann unbrauchbar.

4.4. UC: Neues Projekt

Erstellt ein neues GIANT Projekt. Ein eventuell bereits geöffnetes Projekt wird dabei geschlossen, wobei Änderungen auf Nachfrage vorher gespeichert werden. Weitere wichtige Informationen, die in engem Zusammenhang mit diesem UseCase stehen, sind unter dem Abschnitt [12.1](#) zu finden. Zulässige Namen für Projekte sind unter Abschnitt [3.3.2](#) spezifiziert.

4.4.1. Vorbedingung

1. Das Programm ist gestartet.

4.4.2. Nachbedingung Erfolg

1. Ein neues GIANT Projekt mit dem eingegebenen Namen ist erstellt und geladen.
2. Eine IML-Datei ist geladen.
3. Das angegebene Projektverzeichnis ist gegebenenfalls (falls es noch nicht vorhanden war) erstellt worden (siehe [12.1.2](#)).
4. Die neu erstellte Projektdatei (siehe [12.1.3](#)) liegt im Projektverzeichnis.
5. Ein eventuell zuvor geöffnetes Projekt ist geschlossen. Änderungen an dem eventuell zuvor geöffneten Projekt sind je nach Entscheidung des Benutzers bei der entsprechenden Sicherheitsabfrage gespeichert oder verworfen.

4.4.3. Nachbedingung Fehlschlag

1. Das System bleibt im bisherigen Zustand, falls der Benutzer den UseCase mit Cancel abgebrochen hat.
2. Tritt während des Erstellens des neuen Projektes, nachdem das eventuell zuvor geöffnete Projekt geschlossen wurde, ein Fehler auf, so bleibt das Verhalten des Systems unspezifiziert.

4.4.4. Beschreibung

1. Der Benutzer startet den UseCase über das Menü Project (siehe [14.3.2.1](#)) durch Auswahl von „New Project“.
2. GIANT zeigt den Standard-Filechooser-Dialog und fordert den Benutzer auf eine vorhandene IML-Graph Datei auszuwählen.
3. Der Benutzer wählt im Standard-Filechooser-Dialog eine IML-Datei aus und bestätigt seine Eingabe mit OK (siehe [14.13.1](#)).
4. GIANT zeigt erneut den Standard-Filechooser-Dialog und fordert den Benutzer zur Eingabe des Namens der Projektdatei auf.

5. Der Benutzer gibt im Standard-Filechooser-Dialog den Pfad und den Namen der Projektdatei ein, die Dateiendung wird später von GIANT automatisch gesetzt. Der Name der Projektdatei ist automatisch auch der Name für das Projekt. Das Verzeichnis der Projektdatei wird automatisch zum Projektverzeichnis.
6. Der Benutzer bestätigt er seine Eingabe mit OK.
Existiert die eingegebene Projektdatei bereits, erscheint eine Fehlermeldung gemäß den unter Abschnitt [3.1](#) beschriebenen Konventionen.
Existiert in dem Projektverzeichnis bereits eine andere Projektdatei, so erscheint ebenfalls eine Fehlermeldung.
7. Falls noch ein Projekt geöffnet ist, erscheint eine Sicherheitsabfrage (siehe [14.12](#)) ob dieses gespeichert werden soll. Entscheidet der Benutzer sich für Speichern, so wird die unter [12.2.1](#) beschriebene Funktionalität ausgeführt. Lehnt der Benutzer dies ab, gehen alle nicht gespeicherten Informationen verloren.
8. GIANT schließt das aktuell geöffnete Projekt (falls eines geöffnet war), legt, falls noch nicht vorhanden, ein neues Projektverzeichnis an, erzeugt das neue Projekt und öffnet dieses.

4.4.5. Alternativen

- Bei Punkt [3](#):
Der Benutzer bricht die Verarbeitung mit Cancel ab.
- Bei Punkt [5](#):
Der Benutzer bricht die Verarbeitung mit Cancel ab.

4.5. UC: Projekt öffnen

Öffnet ein GIANT Projekt. Ein eventuell bereits geöffnetes Projekt wird dabei geschlossen, wobei Änderungen auf Nachfrage vorher gespeichert werden.

Sollte der Benutzer die XML-Dateien innerhalb des Projektverzeichnisses (siehe [12.1.2](#)) manuell modifiziert haben, so dass diese von den durch GIANT automatisch erstellten Dateien abweichen, wird keinerlei Garantie für das korrekte Öffnen des Projektes übernommen. Das Verhalten bezüglich eventuell auftretender Fehler ist undefiniert.

4.5.1. Vorbedingung

1. Das Programm ist gestartet.

4.5.2. Nachbedingung Erfolg

1. Das gewünschte GIANT Projekt ist geladen.
2. Die zugehörige IML-Datei ist geladen.
3. Änderungen an einem eventuell zuvor geöffneten Projekt sind je nach Wahl des Benutzers bei der Sicherheitsabfrage gespeichert oder verworfen.

4.5.3. Nachbedingung Fehlschlag

1. Hat der Benutzer den UseCase mit Cancel abgebrochen, bleibt das System im bisherigen Zustand.
2. Muss der UseCase während des Ladens des neuen Projektes aufgrund eines Fehlers abgebrochen werden, so kehrt das System falls möglich in einen Zustand zurück, der dem Start des Systems ohne Öffnen eines Projektes entspricht.

4.5.4. Beschreibung

1. Der Benutzer startet den UseCase über das Menü Project (siehe [14.3.2.1](#) Menü) durch Auswahl von „Load Project“. GIANT zeigt daraufhin den Standard-Filechooser-Dialog und fordert den Benutzer zur Auswahl eines zu öffnenden Projektes auf.
2. Der Benutzer wählt aus dem Standard-Filechooser-Dialog (siehe [14.13.1](#)) eine vorhandene GIANT Projektdatei (siehe [12.1.3](#)) aus und bestätigt mit OK.
3. Falls bereits ein Projekt geöffnet ist, erscheint eine Sicherheitsabfrage (siehe [14.12](#)) ob dieses gespeichert werden soll. Entscheidet der Benutzer sich für Speichern, so wird die unter [12.2.1](#) beschriebene Funktionalität ausgeführt. Lehnt der Benutzer dies ab, gehen alle nicht gespeicherten Informationen verloren.

4. GIANT schließt das alte Projekt (falls eines geöffnet war) und lädt das angegebene neue Projekt.

4.5.5. Alternativen

- Bei Punkt 2:
Der Benutzer bricht die Verarbeitung mit Cancel ab.

4.6. UC: Projekt speichern

Speichert alle Änderungen an einem Projekt. Der Zustand der entsprechenden Verwaltungsdateien im Projektverzeichnis entspricht nach erfolgreicher Ausführung dieses UseCases exakt dem aktuellen Zustand des geöffneten Projektes. Alle Konventionen zur Persistenz von Projekten sind im Abschnitt [12.1](#) exakt spezifiziert.

4.6.1. Vorbedingung

1. Das Programm ist gestartet.
2. Ein Projekt ist geöffnet.

4.6.2. Nachbedingung Erfolg

1. Die Informationen des Projekts (einschließlich aller möglichen Änderungen) sind persistent in die Verwaltungsdateien geschrieben.

4.6.3. Nachbedingung Fehlschlag

1. Das System bleibt im bisherigen Zustand.

4.6.4. Beschreibung

1. Der Benutzer startet den UseCase über das Menü Project (siehe [14.3.2.1](#)) durch Auswahl von „Save Project“.
2. GIANT führt die unter [12.2.1](#) beschriebene Funktionalität aus und speichert alle Informationen zu dem Projekt in der zugehörigen Projektdatei.

4.6.5. Alternativen

- Generell:
Es sind keine alternativen Abläufe vorgesehen.

4.7. UC: Projekt speichern unter

Speichert alle Informationen zu einem Projekt in eine neue Projektdatei (entsprechende Verwaltungsdateien werden ebenfalls dupliziert).

4.7.1. Vorbedingung

1. Das Programm ist gestartet.
2. Ein Projekt ist geöffnet (entweder ein neu erzeugtes oder ein geladenes).

4.7.2. Nachbedingung Erfolg

1. Eine neue Projektdatei ist erzeugt worden.
2. Das angegebene Projektverzeichnis ist, falls es noch nicht vorhanden war, erstellt worden.
3. Die Daten des Projekts sind persistent in die neuen Verwaltungsdateien im Projektverzeichnis der neuen Projektdatei geschrieben (siehe auch [12.1](#)).
4. Das aktuell geöffnete Projekt bleibt in GIANT geöffnet, zukünftiges Speichern (siehe [4.6](#)) betrifft nur die Verwaltungsdateien im neu erzeugten Projekt.
5. Die alte Projektdatei und alle zugehörigen Verwaltungsdateien bleiben unverändert.

4.7.3. Nachbedingung Fehlschlag

1. Das System bleibt im bisherigen Zustand.

4.7.4. Beschreibung

1. Der Benutzer startet den UseCase über das Menü Project (siehe [14.3.2.1](#)) durch Auswahl von „Save Project As...“. Daraufhin zeigt GIANT den Standard-Filechooser-Dialog und fordert den Benutzer auf den Namen und den Ort (Pfad) der neuen Projektdatei einzugeben.
2. Der Benutzer gibt im Standard-Filechooser-Dialog (siehe [14.13.1](#)) das neue Projektverzeichnis (siehe [12.1.2](#)) und den Namen für die neue Projektdatei (siehe [12.1.3](#)) ein, die Dateiendung wird später von GIANT automatisch gesetzt.
Der Name der Projektdatei ist automatisch auch der Name für das Projekt. Zulässige Namen sind unter Abschnitt [3.3.2](#) spezifiziert.
Das Verzeichnis, welches für die Projektdatei angegeben wurde, wird automatisch zum Projektverzeichnis.
3. Der Benutzer bestätigt seine Eingabe mit OK.
Existiert die eingegebene Datei schon, erscheint eine Fehlermeldung.
Existiert in dem angegebenen Projektverzeichnis bereits eine andere Projektdatei, so erscheint

eine entsprechende Fehlermeldung.

4. GIANT speichert das alte Projekt in der neuen Projektdatei, falls noch nicht vorhanden wird auch das neue Projektverzeichnis angelegt. Alle Verwaltungsdateien werden ebenfalls im neuen Projektverzeichnis gespeichert.

4.7.5. Alternativen

- Bei Punkt [2](#):
Der Benutzer bricht die Verarbeitung mit Cancel ab.

4.8. UC: Leeres Anzeigefenster erzeugen

Über diesen UseCase kann der Benutzer neue Anzeigefenster innerhalb eines Projektes anlegen.

4.8.1. Vorbedingung

1. Ein Projekt ist geladen.

4.8.2. Nachbedingung Erfolg

1. Das neue, leere Anzeigefenster ist geöffnet.
2. Das neue Anzeigefenster ist Bestandteil des Projektes.

4.8.3. Nachbedingung Fehlschlag

1. Das System bleibt im bisherigen Zustand.

4.8.4. Beschreibung

1. Der Benutzer startet den UseCase über das Popup-Menü (siehe [14.3.4.2](#)) durch Auswahl von „New Window“.
2. GIANT erzeugt ein neues Anzeigefenster (siehe [14.4](#)) mit einem Standard-Namen und öffnet dieses.

4.8.5. Alternativen

- Generell:
Es sind keine alternativen Abläufe vorgesehen.

4.9. UC: Anzeigefenster öffnen

Dient zum Öffnen eines Anzeigefensters des Projektes.

4.9.1. Vorbedingung

1. Ein Projekt mit mindestens einem Anzeigefenster ist geladen.
2. Es gibt mindestens ein nicht geöffnetes Anzeigefenster.

4.9.2. Nachbedingung Erfolg

1. Das Anzeigefenster ist geöffnet.

4.9.3. Nachbedingung Fehlschlag

1. Es sind keine Fehlerfälle vorgesehen.

4.9.4. Beschreibung

1. Der Benutzer führt einen Doppelklick auf ein nicht geöffnetes Anzeigefenster in der Liste über die Anzeigefenster (siehe [14.3.4.1](#)) durch, oder wählt im zugehörigen Popup-Menü (siehe [14.3.4.2](#)) „Open Window“ aus.
2. GIANT öffnet das entsprechende Anzeigefenster.

4.9.5. Alternativen

- Generell:
Es sind keine alternativen Abläufe vorgesehen.

4.10. UC: Anzeigefenster umbenennen

Dient dem Umbenennen eines Anzeigefensters des Projektes.

4.10.1. Vorbedingung

1. Ein Projekt mit mindestens einem Anzeigefenster ist geladen.

4.10.2. Nachbedingung Erfolg

1. Das Anzeigefenster hat einen neuen Namen.

4.10.3. Nachbedingung Fehlschlag

1. Es sind keine Fehlerfälle vorgesehen.

4.10.4. Beschreibung

1. Der Benutzer führt einen Rechtsklick auf ein Anzeigefenster in der Liste über die Anzeigefenster (siehe [14.3.4.1](#)) durch und wählt im Popup-Menü (siehe [14.3.4.2](#)) „Rename Window“ aus.
2. GIANT öffnet den allgemeinen Texteingabedialog (siehe [14.7](#)).
3. Der Benutzer gibt dort einen zulässigen Namen für das Anzeigefenster ein und bestätigt seine Eingabe mit OK. Zulässige Namen für Anzeigefenster sind unter Abschnitt [3.3.2](#) spezifiziert.
4. GIANT benennt das Anzeigefenster (siehe [14.4](#)) um.

4.10.5. Alternativen

- Generell:
Es sind keine alternativen Abläufe vorgesehen.

4.11. UC: Anzeigefenster speichern

Mit diesem UseCase wird ein Anzeigefenster gespeichert. Näheres zur Persistenz von Anzeigefenstern ist unter Abschnitt [12.1](#) spezifiziert.

4.11.1. Vorbedingung

1. Ein Projekt mit mindestens einem Anzeigefenster ist geladen.

4.11.2. Nachbedingung Erfolg

1. Nach dem letzten Speichern am Anzeigefenster vorgenommene Änderungen sind jetzt auch in der Verwaltungsdatei des Anzeigefensters (siehe [12.1.5](#)) gespeichert. Die in der Verwaltungsdatei gespeicherten Informationen beschreiben jetzt den aktuellen Zustand des Anzeigefensters.
2. Wurde das Anzeigefenster zum ersten mal gespeichert, so existiert jetzt eine Verwaltungsdatei für das Anzeigefenster im Projektverzeichnis.

4.11.3. Nachbedingung Fehlschlag

1. Es sind keine Fehlerfälle vorgesehen.

4.11.4. Beschreibung

1. Der Benutzer führt einen Rechtsklick auf ein Anzeigefenster in der Liste über die Anzeigefenster (siehe [14.3.4.1](#)) durch und wählt im Popup-Menü (siehe [14.3.4.2](#)) „Save Window“ aus.
2. GIANT schreibt alle Änderungen in die Verwaltungsdatei des Anzeigefensters. Anzeigefenster, die zum ersten mal gespeichert werden, haben noch keine Verwaltungsdatei. In diesem Fall erzeugt GIANT jetzt die Verwaltungsdatei für das Anzeigefenster.

4.11.5. Alternativen

- Generell:
Es sind keine alternativen Abläufe vorgesehen.

4.12. UC: Anzeigefenster schließen

Mit diesem UseCase wird ein geöffnetes Anzeigefenster geschlossen. Näheres zur Persistenz von Anzeigefenstern ist unter Abschnitt [12.2.2](#) spezifiziert.

4.12.1. Vorbedingung

1. Ein Projekt mit mindestens einem Anzeigefenster ist geladen.
2. Es gibt mindestens ein geöffnetes Anzeigefenster.

4.12.2. Nachbedingung Erfolg

1. Das Anzeigefenster ist geschlossen.
2. Nach dem letzten Speichern am Anzeigefenster vorgenommene Änderungen (wie z.B. das Verschieben von Fenster-Knoten) sind je nach Wahl des Benutzers bei der Sicherheitsabfrage in der Verwaltungsdatei des Anzeigefensters (siehe [12.1.5](#)) gespeichert oder nicht.
3. Hat der Benutzer bei der Sicherheitsabfrage das Speichern abgelehnt und handelt es sich bei dem geschlossenen Anzeigefenster um ein Anzeigefenster, dass zuvor nicht mindestens einmal gespeichert wurde (ein solches Anzeigefenster hat noch keine Verwaltungsdatei), so gehen alle Informationen zu diesem Anzeigefenster verloren und das Anzeigefenster gehört nicht mehr zum Projekt.

4.12.3. Nachbedingung Fehlschlag

1. Es sind keine Fehlerfälle vorgesehen.

4.12.4. Beschreibung

1. Der Benutzer führt einen Rechtsklick auf ein Anzeigefenster in der Liste über die Anzeigefenster (siehe [14.3.4.1](#)) durch und wählt im Popup-Menü (siehe [14.3.4.2](#)) „Close Window“ aus. Natürlich kann der Benutzer diesen UseCase auch starten, indem er das Anzeigefenster mit den vom jeweiligen Window-Manager bereit gestellten Mitteln (z.B. der typische „X“ Button oben rechts in der Kopfleiste eines Fensters) schließt).
2. GIANT zeigt die allgemeine Sicherheitsabfrage (siehe [14.12](#)) und fragt nach, ob eventuelle Änderungen im Anzeigefenster gespeichert werden sollen oder nicht.
3. Bestätigt der Benutzer mit Yes, werden die Änderungen in die Verwaltungsdatei geschrieben. Anderenfalls gehen sämtliche nicht gespeicherten Änderungen am Anzeigefenster verloren.
4. GIANT schließt das Anzeigefenster.

4.12.5. Alternativen

- Generell:
Es sind keine alternativen Abläufe vorgesehen.

4.13. UC: Anzeigefenster löschen

Mit diesem UseCase werden bestehende Anzeigefenster aus dem Projekt entfernt und gelöscht. Alle Informationen zu dem Anzeigefenster gehen hierbei unwiederbringlich verloren.

4.13.1. Vorbedingung

1. Ein Projekt mit mindestens einem Anzeigefenster ist geladen.

4.13.2. Nachbedingung Erfolg

1. Das gelöschte Anzeigefenster ist nicht mehr Bestandteil des Projektes.
2. Falls das Anzeigefenster vor dem Löschen geöffnet war, so ist das Anzeigefenster jetzt geschlossen.
3. Die Verwaltungsdatei für das Anzeigefenster (siehe [12.1.5](#)) wurde ebenfalls gelöscht.

4.13.3. Nachbedingung Fehlschlag

1. Das System bleibt im bisherigen Zustand.

4.13.4. Beschreibung

1. Der Benutzer führt einen Rechtsklick auf ein Anzeigefenster in der Liste über die Anzeigefenster (siehe [14.3.4.1](#)) durch und wählt im Popup-Menü (siehe [14.3.4.2](#)) „Delete Window“ aus.
2. GIANT zeigt die allgemeine Sicherheitsabfrage (siehe [14.12](#)) an und fragt nach, ob das Anzeigefenster wirklich gelöscht werden soll: „Do you really want to delete the selected window?“
3. Der Benutzer bestätigt mit Yes.
4. GIANT entfernt das Anzeigefenster aus dem Projekt und löscht die zugehörige Verwaltungsdatei (siehe auch [12.1.5](#)).

4.13.5. Alternativen

- Bei Punkt 3:
Der Benutzer bricht die Verarbeitung mit No ab.

4.14. UC: IML-Teilgraph in Anzeigefenster einfügen

Mit diesem UseCase können die Graph-Kanten und Graph-Knoten von IML-Teilgraphen in Anzeigefenster eingefügt werden. Siehe auch [3.5](#) und insbesondere [3.5.4](#).

4.14.1. Vorbedingung

1. Ein Projekt mit mindestens einem geöffneten Anzeigefenster ist geladen.
2. Es gibt mindestens einen IML-Teilgraphen.

4.14.2. Nachbedingung Erfolg

1. Alle Graph-Knoten und Graph-Kanten des IML-Teilgraphen sind in das Anzeigefenster entsprechend dem gewählten Layout an der vorgegebenen Position eingefügt.
2. In dem Anzeigefenster gibt es eine neue aktuelle Selektion, die die neu eingefügten Fenster-Knoten und Fenster-Kanten umfasst.

4.14.3. Nachbedingung Fehlschlag

1. Hat der Benutzer den UseCase an irgendeinem Punkt abgebrochen, kehrt das System zu dem Zustand zurück, in dem es vor dem Start des UseCase war.

4.14.4. Beschreibung

1. Der Benutzer startet den UseCase über das Popup-Menü (siehe [14.3.5.2](#)) im Hauptfenster durch Auswahl des Eintrags „Insert IML Subgraph“. Hierdurch wird der einzufügende IML-Teilgraph bestimmt (immer der IML-Teilgraph, auf dem der Rechtsklick ausgeführt wurde).
2. GIANT zeigt in der Statuszeile im Hauptfenster „Select Position In Display Window For Insertion Of IML Subgraph“ an. Der Benutzer wählt das entsprechende Anzeigefenster aus und gibt über das Fadenkreuz (siehe [14.14](#)) die Position vor, an der die neuen Fenster-Knoten und Fenster-Kanten eingefügt werden sollen.
3. GIANT zeigt den Dialog zur Auswahl von Layoutalgorithmen (siehe [14.9](#)).
4. Der Benutzer wählt einen der vorgegebenen Layoutalgorithmen aus. Bei semantischen Layouts gibt er über den Layoutalgorithmen Dialog (siehe [14.9](#)) auch die Klassenmengen vor, die für das Layout berücksichtigt werden sollen (siehe Kapitel [11](#) für Details zu Layoutalgorithmen). Ggf. ist der Wurzelknoten anhand seiner ID einzugeben.
5. Der Benutzer bestätigt mit OK.
6. GIANT berechnet das entsprechende Layout und zeigt einen Dialog an, der den Benutzer über den Fortschritt der Berechnung informiert (siehe [14.15.2](#)).

Während der Berechnung des Layouts kann das System GIANT nicht bedient werden. Zugänglich ist nur der Button zum Abbruch der Berechnung (siehe [14.15.2](#)).

7. Nach Abschluss der Berechnung fügt GIANT die Fenster-Knoten und Fenster-Kanten in das entsprechende Anzeigefenster ein.

4.14.5. Alternativen

- Bei Punkt [2](#):
Der Benutzer kann den UseCase abbrechen, indem er mit der Maus (dem Fadenkreuzcursor) einen Rechtsklick auf eine beliebige Stelle durchführt.
- Bei Punkt [4](#):
Der Benutzer bricht den UseCase mit Cancel ab.
- Bei Punkt [6](#):
Der Benutzer bricht die Berechnung des Layouts ab.

4.15. UC: Selektion in Anzeigefenster einfügen

Mit diesem UseCase kann eine Selektion aus einem Quell-Anzeigefenster in ein Ziel-Anzeigefenster unter Beibehaltung des Layouts kopiert werden (siehe [3.5](#) und [3.5.3](#)).

4.15.1. Vorbedingung

1. Ein Projekt mit mindestens zwei geöffneten Anzeigefenstern ist geladen.
2. Es gibt mindestens eine Selektion.

4.15.2. Nachbedingung Erfolg

1. Die Position von Fenster-Knoten, die vor dem Einfügen bereits im Ziel-Anzeigefenster vorhanden waren, bleibt je nach Wahl des Benutzers unverändert oder wird ebenfalls geändert.
2. Die kopierte Selektion existiert auch im Ziel-Anzeigefenster als Selektion, sie wird dort zur Standard-Selektion. Der alte Inhalt der Standard-Selektion wird überschrieben.
3. Im Ziel-Anzeigefenster gibt es keine Fenster-Knoten mit der selben ID mehrfach.

4.15.3. Nachbedingung Fehlschlag

1. Hat der Benutzer den UseCase an irgendeinem Punkt abgebrochen, kehrt das System zu dem Zustand zurück, in dem es vor dem Start des UseCase war.

4.15.4. Beschreibung

1. Der Benutzer startet den UseCase über das Popup-Menü der Selektionsauswahlliste (siehe [14.4.4](#)), indem er dort den Menüeintrag „Copy Selection Keeping Existing Layout“ oder den Menüeintrag „Copy Selection Changing Existing Layout“ auswählt. Hierdurch wird automatisch die Quell-Selektion bestimmt (immer die Selektion, auf der der Rechtsklick ausgeführt wurde).
2. Der Benutzer wählt das gewünschte Ziel-Anzeigefenster aus. Das Ziel-Anzeigefenster kann nicht das Quell-Anzeigefenster sein. GIANT zeigt in der Statuszeile im Hauptfenster den Text „Select Position in Display Window for Insertion of copied IML Subgraph“ an. Der Benutzer gibt über das Fadenkreuz die Position vor, an der die neuen Fenster-Knoten und Fenster-Kanten eingefügt werden sollen (siehe auch [14.14](#)). Die mit dem Fadenkreuzcursor vorgegebene Position entspricht hierbei in etwa dem Punkt, an dem der Mittelpunkt eines die eingefügte Selektion umspannenden Rechtecks dann im Ziel-Anzeigefenster liegt.
3. GIANT kopiert die Fenster-Knoten und Fenster-Kanten der Quell-Selektion in das Ziel-Anzeigefenster. Bei neu eingefügten Fenster-Knoten bleibt das Layout, welches diese in der Quell-Selektion relativ zueinander haben, erhalten.
Je nachdem, welchen Eintrag der Benutzer unter Schritt 1 im Popup-Menü ausgewählt hat,

geschieht mit den bereits im Ziel-Anzeigefenster vorhandenen Fenster-Knoten der Selektion folgendes:

- a) Falls „Copy Selection Keeping Existing Layout“ gewählt wurde, wird ihre Position im Ziel-Anzeigefenster nicht verändert.
- b) Falls „Copy Selection Changing Existing Layout“ gewählt wurde, wird ihre Position im Ziel-Anzeigefenster gemäß dem Layout der eingefügten Selektion verändert.

4.15.5. Alternativen

- Bei Punkt 2:
Der Benutzer kann den UseCase abbrechen, indem er mit der Maus einen Rechtsklick auf eine beliebige Stelle durchführt.

4.16. UC: Alle Fenster-Knoten und Fenster-Kanten einer Selektion aus einem Anzeigefenster löschen

Mittels dieses UseCases können alle Fenster-Knoten und Fenster-Kanten einer Selektion aus einem Anzeigefenster gelöscht werden (siehe auch [3.7](#)).

4.16.1. Vorbedingung

1. Ein Projekt mit mindestens einem geöffneten Anzeigefenster ist geladen.
2. Es gibt mindestens eine Selektion (dies kann auch die immer vorhandene Standard-Selektion sein).

4.16.2. Nachbedingung Erfolg

1. Falls die gewählte Selektion nicht die Standard-Selektion war, ist die Selektion jetzt aus dem Anzeigefenster gelöscht und taucht nicht mehr in der Liste über die Selektionen auf.
2. Wurde die Standard-Selektion (siehe [3.4.2](#)) gewählt, so wird diese nicht aus dem Anzeigefenster gelöscht, sondern ist jetzt halt leer.
3. Die Standard-Selektion ist jetzt die aktuelle Selektion (siehe [3.4.3](#)).
4. Alle betroffenen Fenster-Knoten und Fenster-Kanten sind gemäß der unter [3.7](#) beschriebenen Konvention aus dem Anzeigefenster entfernt.
5. Alle anderen Selektionen des Anzeigefensters wurden aktualisiert.

4.16.3. Nachbedingung Fehlschlag

1. Hat der Benutzer den UseCase abgebrochen, kehrt das System zu dem Zustand zurück, in dem es vor dem Start des UseCase war.

4.16.4. Beschreibung

1. Der Benutzer führt einen Rechtsklick auf die entsprechende Selektion in der Selektionsauswahlliste durch (siehe [14.4.4](#)) und wählt im Popup-Menü den Eintrag „Delete Nodes and Edges of Selection“ aus.
2. GIANT zeigt die Sicherheitsabfrage (siehe [14.12](#)) und fragt nach, ob es die Selektion samt ihrer Fenster-Knoten und Fenster-Kanten wirklich löschen soll („Really delete Selection from its window including Nodes and Edges?“).
3. Der Benutzer bestätigt mit Yes.

4. GIANT löscht die Selektion samt allen zugehörigen Fenster-Knoten und Fenster-Kanten aus dem entsprechenden Anzeigefenster. Wurde der UseCase für die Standard-Selektion (siehe [3.4.2](#)) ausgeführt, so werden die Fenster-Knoten und Fenster-Kanten aus dem Anzeigeeinhalt gelöscht, die Standard-Selektion selbst wird geleert aber nicht gelöscht.

4.16.5. Alternativen

- Bei Punkt [2](#):
Der Benutzer bricht den UseCase mit No ab.

4.17. UC: Den Visualisierungsstil eines Anzeigefensters ändern

Mittels dieses UseCase kann der Benutzer die Visualisierung von Fenster-Knoten und Fenster-Kanten innerhalb eines Anzeigefensters zur Laufzeit von GIANT durch Auswahl verschiedener frei definierbarer Visualisierungsstile ändern (siehe auch [13.3](#)).

4.17.1. Vorbedingung

1. Es gibt ein geöffnetes Anzeigefenster.
2. Es gibt mindestens einen benutzerdefinierten Visualisierungsstil.

4.17.2. Nachbedingung Erfolg

1. Die Darstellung der Fenster-Knoten und Fenster-Kanten in dem Anzeigefenster entspricht den Vorgaben des gewählten Visualisierungsstils.
2. Alle anderen Zustände und Eigenschaften des Anzeigefensters, wie z.B. die hervorgehobenen Selektionen, bleiben unverändert.

4.17.3. Nachbedingung Fehlschlag

1. Es sind keine Fehlerfälle vorgesehen.

4.17.4. Beschreibung

1. Der Benutzer ändert den Visualisierungsstil eines Anzeigefensters dadurch, dass er in der Stilauswahl-Combobox des Anzeigefensters (siehe [14.4.5](#)) einen anderen Visualisierungsstil einstellt.
2. GIANT ändert die Darstellung von Fenster-Knoten und Fenster-Kanten entsprechend ab (näheres zur Visualisierung von Fenster-Knoten und Fenster-Kanten ist in Kapitel [15](#) spezifiziert).

4.17.5. Alternativen

- Generell:
Es sind keine alternativen Abläufe vorgesehen.

4.18. UC: Anzeigefenster scrollen

Verändert die Position des sichtbaren Anzeigehaltes.

4.18.1. Vorbedingung

1. Ein Projekt mit mindestens einem geöffneten Anzeigefenster ist geladen.

4.18.2. Nachbedingung Erfolg

1. Die Position des sichtbaren Anzeigehaltes wurde entsprechend abgeändert.

4.18.3. Nachbedingung Fehlschlag

1. Es sind keine Fehlerfälle vorgesehen.

4.18.4. Beschreibung

1.
 - a) Der Benutzer scrollt den sichtbaren Anzeigehalt mittels der horizontalen oder vertikalen Bildlaufleisten des Anzeigefensters (siehe [14.4.7](#)). Dies geschieht mittels der Maus gemäß der Konventionen von GTK/Ada für Bildlaufleisten.
 - b) Es kann auch mittels der Cursortasten gescrollt werden.
Das Drücken der linken Cursortaste führt z.B. dazu, dass der sichtbare Anzeigehalt des aktiven Anzeigefensters nach links verschoben wird; dadurch werden dann die Fenster-Knoten und Fenster-Kanten sichtbar, die sich links vom aktuell sichtbaren Anzeigehalt befinden.
Mittels der Cursortasten werden also die Bildlaufleisten simuliert. Das Drücken der linken Cursortaste entspricht z.B. dann einem Klick auf den linken Pfeil der horizontalen Bildlaufleiste.

4.18.5. Alternativen

- Generell:
Es sind keine alternativen Abläufe vorgesehen.

4.19. UC: Anzeigefenster zoomen

Verändert den Maßstab der Darstellung von Fenster-Knoten und Fenster-Kanten.

4.19.1. Vorbedingung

1. Ein Projekt mit mindestens einem geöffneten Anzeigefenster ist geladen.

4.19.2. Nachbedingung Erfolg

1. Der angezeigte Bereich des sichtbaren Anzeigeinhalts wurde entsprechend vergrößert oder verkleinert. Die Detailstufe (siehe [15.4](#)) wurde ggf. automatisch angepasst.

4.19.3. Nachbedingung Fehlschlag

1. Es sind keine Fehlerfälle vorgesehen.

4.19.4. Beschreibung

1. Der Benutzer gibt in der Combobox zur Zoom-Kontrolle (siehe [14.4.6](#)) des Anzeigefensters einen Wert für die neue Zoomstufe ein, wählt einen der vordefinierten Werte aus oder ändert den Zoomwert in festgelegten Schritten mit den „+“ oder „-“ Buttons.
2. GIANT berechnet den neuen sichtbaren Anzeigeinhalt anhand der neuen Zoomstufe.

4.19.5. Alternativen

- Generell:
Es sind keine alternativen Abläufe vorgesehen.

4.20. UC: Zoomen auf eine gesamte Selektion

Wählt die passende Zoomstufe und scrollt den sichtbaren Anzeiginhalt so, dass eine Selektion im Anzeigefenster vollständig sichtbar ist.

4.20.1. Vorbedingung

1. Ein Projekt mit mindestens einem geöffneten Anzeigefenster ist geladen.

4.20.2. Nachbedingung Erfolg

1. Der sichtbare Anzeiginhalt wurde mittels Zoomen und Scrollen so verändert, dass die ausgewählte Selektion vollständig sichtbar ist. Die Detailstufe (siehe [15.4](#)) wurde ggf. automatisch angepasst.

4.20.3. Nachbedingung Fehlschlag

1. Es sind keine Fehlerfälle vorgesehen.

4.20.4. Beschreibung

1. Der Benutzer klickt auf „Zoom To Make Selection Fill Window“ im Popup-Menü der Selektionsauswahlliste (siehe [14.4.4](#)).
2. GIANT scrollt und zoomt automatisch so, dass die gesamte Selektion im Anzeigefenster sichtbar wird.

4.20.5. Alternativen

- Generell:
Es sind keine alternativen Abläufe vorgesehen.

4.21. UC: Zoomen auf den gesamten Inhalt eines Anzeigefensters

Wählt die passende Zoomstufe und scrollt den sichtbaren Anzeigeeinhalt so, dass der Bereich des Anzeigeeinhaltes, in dem Fenster-Knoten visualisiert sind, vollständig im sichtbaren Anzeigeeinhalt dargestellt wird.

4.21.1. Vorbedingung

1. Ein Projekt mit mindestens einem geöffneten Anzeigefenster ist geladen.

4.21.2. Nachbedingung Erfolg

1. Der sichtbare Anzeigeeinhalt wurde mittels Zoomen und Scrollen so verändert, dass der Bereich des Anzeigeeinhaltes, der Fenster-Knoten enthält, vollständig im sichtbaren Anzeigeeinhalt dargestellt ist. Der sichtbare Anzeigeeinhalt wurde also derart verändert, dass er jetzt das gesamte virtuelle Rechteck, welches alle Fenster-Knoten des Anzeigeeinhaltes umspannt, enthält.
2. Die Detailstufe (siehe [15.4](#)) wurde ggf. automatisch angepasst.

4.21.3. Nachbedingung Fehlschlag

1. Es sind keine Fehlerfälle vorgesehen.

4.21.4. Beschreibung

1. Der Benutzer wählt den Eintrag „Fit in window“ in der Combobox Zoom-Kontrolle aus (siehe [14.4.6](#)).
2. GIANT berechnet die neue Zoomstufe für das Anzeigefenster.

4.21.5. Alternativen

- Generell:
Es sind keine alternativen Abläufe vorgesehen.

4.22. UC: Zoomen auf eine Kante

Wählt die passende Zoomstufe und scrollt den sichtbaren Anzeigeeinhalt so, dass eine Fenster-Kante mit ihren Start- und Ziel-Fenster-Knoten komplett im sichtbaren Anzeigeeinhalt liegt.

4.22.1. Vorbedingung

1. Ein Projekt mit mindestens einem geöffneten Anzeigefenster mit mindestens einer Fenster-Kante ist geladen.

4.22.2. Nachbedingung Erfolg

1. Der sichtbare Anzeigeeinhalt wurde durch geeignetes Zoomen und Scrollen so verändert, dass die ausgewählte Fenster-Kante samt ihrem Start- und ihrem Ziel-Fenster-Knoten vollständig im sichtbaren Anzeigeeinhalt dargestellt ist.
2. Die Detailstufe (siehe [15.4](#)) wurde ggf. automatisch angepasst.

4.22.3. Nachbedingung Fehlschlag

1. Hat der Benutzer den UseCase abgebrochen, kehrt das System zu dem Zustand zurück, in dem es vor dem Start des UseCase war.

4.22.4. Beschreibung

1. Der Benutzer wählt den Button „Pick Edge“ in der Zoomkontrolle (siehe [14.4.6](#)) aus.
2. Daraufhin erscheint in der Statuszeile von GIANT im Hauptfenster der Text „Select edge to be zoomed onto“ und der Fadenkreuz-Cursor (siehe [14.14](#)) wird angezeigt, wenn der Mauscursor über den sichtbaren Anzeigeeinhalt eines Anzeigefensters bewegt wird.
3. Der Benutzer klickt mit der linken Maustaste auf die gewünschte Fenster-Kante im gewünschten Anzeigefenster.
4. GIANT berechnet für das gewünschte Anzeigefenster die Zoomstufe so, dass die gesamte Fenster-Kante, der Start-Fenster-Knoten der Fenster-Kante und der Ziel-Fenster-Knoten der Fenster-Kante im Anzeigeeinhalt sichtbar dargestellt werden.

4.22.5. Alternativen

- Bei Punkt 3:
Der Benutzer kann den UseCase abbrechen, indem er mit der Maus (dem Fadenkreuzcursor) einen Rechtsklick auf eine beliebige Stelle ausführt.

4.23. UC: Verschieben von Fenster-Knoten und Selektionen mittels Cut and Paste

Mit diesem UseCase können Fenster-Knoten und Selektionen auf dem Anzeigehalt verschoben werden. Dieses Verschieben geschieht mittels „Cut and Paste“.

4.23.1. Vorbedingung

1. Ein Projekt mit mindestens einem geöffneten Anzeigefenster und mit mindestens einem Fenster-Knoten oder mindestens einer Selektion ist geladen.

4.23.2. Nachbedingung Erfolg

1. Der einzelne Fenster-Knoten oder alle Fenster-Knoten der Selektion wurden im Anzeigehalt verschoben. Bei den nicht ausgeblendeten Fenster-Knoten einer verschobenen Selektion bleibt das Layout dieser Fenster-Knoten relativ zueinander unverändert.
2. Ausgeblendete Fenster-Knoten, die zu einer verschobenen Selektion gehören, bleiben ausgeblendet.
3. Ausgeblendete Fenster-Knoten bleiben an ihrer alten Position, auch wenn eine Selektion, zu der sie gehören, verschoben wurde.

4.23.3. Nachbedingung Fehlschlag

1. Hat der Benutzer den UseCase abgebrochen, so werden keine Fenster-Knoten verschoben.

4.23.4. Beschreibung

1. Der Benutzer wählt die zu verschiebende Selektion oder den zu verschiebenden Fenster-Knoten aus („Cut“), indem er:
 - a) Einen Rechtsklick auf eine Selektion in der Selektionsauswahlliste durchführt (siehe [14.4.4](#)) und im Popup-Menü „Move Selection“ auswählt,
 - b) oder einen Rechtsklick auf einen Fenster-Knoten durchführt und im Popup Menü (siehe [14.4.2.1](#)) „Move Node“ auswählt.
2. GIANT geht in den „Paste Modus“ über und zeigt dies in der Statusleiste (siehe [14.3.3](#)) des Hauptfensters an. Der Cursor wird, falls er über den sichtbaren Anzeigehalt eines Anzeigefensters bewegt wird, zum Fadenkreuz (siehe [14.14](#)). Die Funktionalität zum Zoomen und Scrollen des Anzeigefensters mittels der beiden UseCases [4.18](#) und [4.19](#) bleibt weiterhin verfügbar, die übrige Funktionalität von GIANT wird gesperrt.
3. Der Benutzer klickt mit der linken Maustaste an eine beliebige Stelle innerhalb des sichtbaren Anzeigehaltes des Anzeigefensters.

4. GIANT verschiebt die ausgewählten Fenster-Knoten an die gewünschte Stelle. Die zuvor mit dem Fadenkreuzcursor vorgegebene Position entspricht jetzt der des Mittelpunktes eines die verschobene Selektion umspannenden Rechtecks.

4.23.5. Alternativen

- Bei Punkt 3:
Der Benutzer kann den UseCase abbrechen, indem er einen Rechtsklick mit der Maus im sichtbaren Anzeigebereich durchführt.

4.24. UC: Verschieben einzelner Fenster-Knoten mittels Drag and Drop

Mit diesem UseCase können einzelne Fenster-Knoten mittels Drag and Drop auf dem sichtbaren Anzeigehalt verschoben werden.

4.24.1. Vorbedingung

1. Ein Projekt mit mindestens einem geöffneten Anzeigefenster und mit mindestens einem Fenster-Knoten ist geladen.

4.24.2. Nachbedingung Erfolg

1. Der einzelne Fenster-Knoten wurde auf dem Anzeigehalt entsprechend verschoben.

4.24.3. Nachbedingung Fehlschlag

1. Es sind keine Fehlerfälle vorgesehen.

4.24.4. Beschreibung

1. Der Benutzer bewegt den Mauscursor über den zu verschiebenden Fenster-Knoten und drückt die linke Maustaste („Drag“), dann bewegt er den Fenster-Knoten an eine beliebige andere Stelle innerhalb des Anzeigehaltes und lässt die linke Maustaste los („Drop“).

4.24.5. Alternativen

- Generell:
Es sind keine alternativen Abläufe vorgesehen.

4.25. UC: Platz schaffen

Dieser UseCase wird benötigt, um Fenster-Knoten auseinander schieben zu können. So kann der Benutzer an einer beliebigen Stelle des Anzeigefensters genügend Platz zum Einfügen neuer Fenster-Knoten und Fenster-Kanten schaffen (siehe auch [3.8](#)).

4.25.1. Vorbedingung

1. Ein Projekt mit mindestens einem geöffneten Anzeigefenster ist geladen.

4.25.2. Nachbedingung Erfolg

1. Alle Fenster-Knoten und Fenster-Kanten des Anzeigefensters sind um den entsprechenden Betrag vom vorgegebenen Punkt innerhalb des Anzeigeeinhaltes weggeschoben worden. An der entsprechenden Stelle im Anzeigeeinhalt ist eine freie Fläche ohne Fenster-Knoten geschaffen worden. Diese Fläche kann aber von Fenster-Kanten gekreuzt werden.
2. Das Layout aller Fenster-Knoten des Anzeigeeinhaltes bleibt ansonsten weitgehend unverändert.

4.25.3. Nachbedingung Fehlschlag

1. Hat der Benutzer den UseCase abgebrochen, so wird der Anzeigeeinhalt nicht verändert.

4.25.4. Beschreibung

1. Der Benutzer führt einen Rechtsklick auf eine beliebige Stelle des sichtbaren Anzeigeeinhaltes durch und wählt im daraufhin erscheinenden Popup-Menü (siehe [14.4.2.1](#)) dem Eintrag „Make Room“ aus.
2. GIANT zeigt in der Statuszeile im Hauptfenster „Select position in display window“. Der Benutzer gibt den Punkt um den herum die Fenster-Knoten (und damit automatisch auch die Fenster-Kanten) auseinander geschoben werden sollen über den Fadenkreuzcursor vor (siehe [14.14](#)).
3. GIANT zeigt einen Dialog an, in dem der Benutzer auswählt, um welchen Betrag die Fenster-Knoten auseinander geschoben werden sollen (siehe [14.10.2](#)).
4. Der Benutzer wählt einen geeigneten Betrag aus und bestätigt mit OK.
5. GIANT schiebt die Knoten entsprechend auseinander (siehe [3.8](#)).

4.25.5. Alternativen

- Bei Punkt 2:
Der Benutzer kann den UseCase abbrechen, indem er mit der Maus einen Rechtsklick auf eine beliebige Stelle durchführt.
- Bei Punkt 3:
Der Benutzer kann den UseCase mit Cancel abbrechen.

4.26. UC: Pin anlegen

Mit diesem UseCase kann ein neuer Pin erzeugt werden (siehe auch [2.3.1](#)).

4.26.1. Vorbedingung

1. Es gibt mindestens ein geöffnetes Anzeigefenster.

4.26.2. Nachbedingung Erfolg

1. In der Pinliste des Anzeigefensters (siehe [14.4.3](#)) befindet sich ein neuer Pin mit dem vom Benutzer definierten Namen.

4.26.3. Nachbedingung Fehlschlag

1. Das System bleibt im bisherigen Zustand.

4.26.4. Beschreibung

1. Der Benutzer führt einen Rechtsklick auf den Anzeigeeinhalt eines Anzeigefensters durch und wählt aus dem Popup-Menü (siehe [14.4.2.1](#)) den Eintrag „New Pin“ aus. Der später erstellte Pin verweist dann auf die Stelle im Anzeigeeinhalt, auf die der Rechtsklick durchgeführt wurde.
2. GIANT öffnet den allgemeinen Texteingabedialog (siehe [14.7](#)).
3. Der Benutzer gibt dort einen zulässigen Namen für den neuen Pin ein und bestätigt mit OK (zulässige Namen für sind in Abschnitt [3.3.2](#) spezifiziert).
4. GIANT speichert die aktuelle Zoomstufe und die Position des sichtbaren Anzeigeeinhaltes in einem neuen Pin.

4.26.5. Alternativen

- Bei Punkt [3](#):
Der Benutzer bricht die Verarbeitung mit Cancel ab.

4.27. UC: Pin anspringen

Stellt die im Pin gespeicherte Position des sichtbaren Anzeigehaltes wieder her (siehe auch [2.3.1](#)).

4.27.1. Vorbedingung

1. Es gibt mindestens ein geöffnetes Anzeigefenster mit mindestens einem Pin.

4.27.2. Nachbedingung Erfolg

1. Der sichtbare Anzeigehalt des Anzeigefensters ist auf die entsprechenden Koordinaten und die entsprechende Zoomstufe, wie sie im ausgewählten Pin hinterlegt waren, gesetzt.

4.27.3. Nachbedingung Fehlschlag

1. Es sind keine Fehlerfälle vorgesehen.

4.27.4. Beschreibung

1. Das Anspringen des Pins kann über die folgenden beiden Möglichkeiten geschehen:
 - a) Der Benutzer führt einen Doppelklick auf den entsprechenden Pin in der Pinliste (siehe [14.4.3](#)) aus.
 - b) Der Benutzer öffnet ein Popup-Menü durch Rechtsklick auf den anzuspringenden Pin in der Pinliste (siehe [14.4.3](#)) und wählt den Eintrag „Focus Pin“ aus.
2. GIANT setzt den sichtbaren Anzeigehalt gemäß den im Pin gespeicherten Informationen.

4.27.5. Alternativen

- Generell:
Es sind keine alternativen Abläufe vorgesehen.

4.28. UC: Pin löschen

Löscht einen Pin (siehe auch [2.3.1](#)).

4.28.1. Vorbedingung

1. Es gibt mindestens ein geöffnetes Anzeigefenster mit mindestens einem Pin.

4.28.2. Nachbedingung Erfolg

1. Der Pin ist gelöscht und nicht mehr in der Pinliste (siehe [14.4.3](#)) des entsprechenden Anzeigefensters auswählbar.

4.28.3. Nachbedingung Fehlschlag

1. Es sind keine Fehlerfälle vorgesehen.

4.28.4. Beschreibung

1. Der Benutzer öffnet das Popup-Menü der Pinliste (siehe [14.4.3](#)) durch Rechtsklick auf den Pin und wählt den Eintrag „Delete Pin“ aus.
2. GIANT löscht den Pin.

4.28.5. Alternativen

- Generell:
Es sind keine alternativen Abläufe vorgesehen.

4.29. UC: Layout auf Selektion anwenden

Mittels dieses UseCases können Selektionen innerhalb eines Anzeigefensters layoutet werden. Die verfügbaren Layoutalgorithmen sind in Kapitel 11 beschrieben.

4.29.1. Vorbedingung

1. Es gibt mindestens ein geöffnetes Anzeigefenster mit mindestens einer Selektion (dies kann auch die immer vorhandene Standard-Selektion sein).

4.29.2. Nachbedingung Erfolg

1. Die Position der Fenster-Knoten der Selektion wurde gemäß den Vorgaben des Layoutalgorithmus geändert.
2. Alle anderen Fenster-Knoten bleiben unverändert.

4.29.3. Nachbedingung Fehlschlag

1. Das System bleibt im bisherigen Zustand.
2. Hat der Benutzer einen Layoutalgorithmus während der Berechnung des Layouts abgebrochen, so bleiben die Positionen aller Fenster-Knoten des Anzeigefensters unverändert.

4.29.4. Beschreibung

1. Der Benutzer wählt aus der Selektionsauswahlliste (siehe 14.4.4) die Selektion aus, deren Fenster-Knoten layoutet werden sollen. Hierzu führt er einen Rechtsklick auf die Selektion durch und wählt im Popup-Menü den Eintrag „Layout Selection“ aus.
2. GIANT zeigt einen Dialog zur Auswahl und Konfiguration des gewünschten Layoutalgorithmus (siehe 14.9).
3. Der Benutzer wählt in diesem Dialog den gewünschten Layoutalgorithmus aus.
4. Falls der gewählte Layoutalgorithmus dies unterstützt, kann der Benutzer in diesem Dialog die Klassenmengen (siehe Begriffslexikon), die beim Layout berücksichtigt werden sollen, vorgeben. Bei baumförmigen Layouts kann er auch die ID des gewünschten Wurzelknotens eingeben.
5. Der Benutzer bestätigt seine Eingaben mit OK.
6. GIANT berechnet das Layout und zeigt einen Progressbar (siehe 14.15.2) an. Während der Berechnung des Layouts ist die GUI mit Ausnahme des Progressbars gesperrt.

4.29.5. Alternativen

- Bei Punkt [3](#):
Der Benutzer bricht die Verarbeitung mit Cancel ab.
- Bei Punkt [4](#):
Der Benutzer bricht die Verarbeitung mit Cancel ab.
- Bei Punkt [6](#):
Der Benutzer kann die Berechnung eines Layouts jederzeit mit Cancel (siehe [14.15.2](#)) abbrechen.

4.30. UC: Details zu Knoten in einem neuen Informationsfenster anzeigen

Dieser UseCase ermöglicht dem Benutzer, sich alle verfügbaren Informationen zu einem Fenster-Knoten (Kanten und Attribute des zugehörigen IML-Knoten) anzeigen zu lassen. Mit diesem UseCase kann der Benutzer beliebig viele Knoten-Informationsfenster (siehe [14.5](#)) öffnen.

4.30.1. Vorbedingung

1. Es gibt mindestens ein geöffnetes Anzeigefenster mit mindestens einem Fenster-Knoten.

4.30.2. Nachbedingung Erfolg

1. Die Daten über den Fenster-Knoten werden in einem neuen Knoten-Informationsfenster angezeigt.

4.30.3. Nachbedingung Fehlschlag

1. Es sind keine Fehlerfälle vorgesehen.

4.30.4. Beschreibung

1. Der Benutzer wählt im entsprechenden Anzeigefenster den gewünschten Fenster-Knoten aus und führt einen Rechtsklick auf diesen Fenster-Knoten durch. Im daraufhin angezeigten Popup-Menü (siehe [14.4.2.1](#)) wählt der Benutzer dann den Eintrag „Show Node Info Window“ aus.
2. GIANT öffnet ein neues Knoten-Informationsfenster (siehe [14.5](#)) und zeigt alle verfügbaren Informationen zu dem IML-Knoten an.

4.30.5. Alternativen

- Generell:
Es sind keine alternativen Abläufe vorgesehen.

4.31. UC: Details zu Knoten in einem bestehenden Informationsfenster anzeigen

Mittels dieses UseCase kann sich der Benutzer alle verfügbaren Informationen zu einem Fenster-Knoten (Kanten und Attribute des zugehörigen IML-Knoten) innerhalb eines bereits geöffneten Knoten-Informationsfensters (siehe [14.5](#)) anzeigen lassen.

4.31.1. Vorbedingung

1. Es gibt mindestens ein geöffnetes Anzeigefenster mit mindestens einem Fenster-Knoten.
2. Es gibt mindestens ein geöffnetes Knoten-Informationsfenster.

4.31.2. Nachbedingung Erfolg

1. Anstatt der Daten, die zuvor in dem Knoten-Informationsfenster angezeigt wurden, werden dort nun die Informationen zu dem gewählten Fenster-Knoten angezeigt.

4.31.3. Nachbedingung Fehlschlag

1. Hat der Benutzer den UseCase abgebrochen, bleibt das System im bisherigen Zustand.

4.31.4. Beschreibung

1. Der Benutzer betätigt in einem geöffneten Knoten-Informationsfenster den Button „Pick“ (siehe [14.5](#)).
2. Daraufhin fordert GIANT den Benutzer auf, einen Fenster-Knoten auszuwählen.
3. Der Benutzer wählt den gewünschten Fenster-Knoten dadurch aus, dass er innerhalb des Anzeigehaltes eines Anzeigefensters mittels eines Fadenkreuzes den Fenster-Knoten als Zielposition vorgibt. Das Verfahren zur Vorgabe der Zielposition ist im Detail unter [3.5.1](#) beschrieben.
4. GIANT stellt die verfügbaren Informationen zu dem ausgewählten Fenster-Knoten im Knoten-Informationsfenster dar.

4.31.5. Alternativen

- Bei Punkt [3](#):
Während der Fadenkreuzcursor aktiv ist, kann der Benutzer den UseCase mit einem Rechtsklick auf eine beliebige Stelle abbrechen.

4.32. UC: Anzeige des Quellcodes eines Knotens in externem Editor

Mittels dieses UseCases kann der zu einem IML-Knoten korrespondierende Quellcode innerhalb eines Editors (unterstützt werden Emacs und vi) zur Anzeige gebracht werden.

4.32.1. Vorbedingung

1. Es gibt mindestens ein geöffnetes Anzeigefenster mit einem oder mehr Fenster-Knoten.

4.32.2. Nachbedingung Erfolg

1. Der zu dem IML-Knoten korrespondierende Quellcode wird in dem über die Konfigurationsdatei vorgegebenen Editor angezeigt (siehe auch [13.2.3](#)). Der Cursor des Editors ist auf die dem IML-Knoten entsprechende Position innerhalb des Quelltextes gesetzt, also z.B. auf einen Bezeichnernamen, den der IML-Knoten repräsentiert.
2. Hat der IML-Knoten keine „Source Code Position“, so wird nur die zugehörige Quellcode-Datei geöffnet.

4.32.3. Nachbedingung Fehlschlag

1. Es sind keine Fehlerfälle vorgesehen.

4.32.4. Beschreibung

1. Der Benutzer wählt in einem Anzeigefenster den gewünschten Fenster-Knoten aus und führt einen Rechtsklick auf diesen Knoten durch. Im darauf angezeigten Popup-Menü (siehe [14.4.2.1](#)) wählt der Benutzer dann den Eintrag „Show Corresponding Source Code“ aus. Der UseCase kann nicht für IML-Knoten durchgeführt werden, für die die Reflektion zur Bauhaus IML-Graph-Bibliothek keine zugehörige Quellcode-Datei liefern kann.
2. GIANT öffnet die Quellcode-Datei für den IML-Knoten in einem Editor.

4.32.5. Alternativen

- Generell:
Es sind keine alternativen Abläufe vorgesehen.

4.33. UC: Verschieben des sichtbaren Anzeigeinhaltes über die Minimap

Der Benutzer kann mittels der Minimap (siehe auch [14.4.1](#)) den sichtbaren Anzeigeinhalt scrollen.

4.33.1. Vorbedingung

1. Es gibt mindestens ein geöffnetes Anzeigefenster.

4.33.2. Nachbedingung Erfolg

1. Der sichtbare Anzeigeinhalt ist auf den Bereich des Anzeigefensters gesetzt, der dem Punkt welcher auf der Minimap angeklickt wurde, entspricht.

4.33.3. Nachbedingung Fehlschlag

1. Es sind keine Fehlerfälle vorgesehen.

4.33.4. Beschreibung

1. Der Benutzer klickt mit der linken Maustaste auf einen Punkt auf der Minimap.
2. GIANT zentriert den sichtbaren Anzeigeinhalt des Anzeigefensters auf den vorgegebenen Punkt.

4.33.5. Alternativen

- Generell:
Es sind keine alternativen Abläufe vorgesehen.

5. Knoten-Annotationen

Die folgenden UseCases beschreiben die Funktionalität zum Erzeugen, Ändern und Löschen von Knoten-Annotationen.

5.1. UC: Fenster-Knoten annotieren

Dieser UseCase dient zum Erzeugen von Knoten-Annotationen (siehe auch [12.2.4](#)).

5.1.1. Vorbedingung

1. Es gibt mindestens ein Anzeigefenster mit mindestens einem Fenster-Knoten.

5.1.2. Nachbedingung Erfolg

1. Die neue Knoten-Annotation ist noch nicht in der Verwaltungsdatei für Knoten-Annotationen (siehe [12.1.7](#)) eingetragen. Dies geschieht erst beim nächsten Speichern des gesamten Projektes.
2. Der annotierte IML-Knoten wird als Fenster-Knoten in allen Anzeigefenstern mittels eines Icons als annotiert gekennzeichnet (siehe [15.1.3](#)).

5.1.3. Nachbedingung Fehlschlag

1. Das System bleibt im bisherigen Zustand.

5.1.4. Beschreibung

1. Der Benutzer führt einen Rechtsklick auf den Knoten, der annotiert werden soll, durch und wählt im Popup Menü für Knoten den Eintrag „Annotation – Add“ aus (siehe [14.4.2.1](#)).
2. GIANT zeigt nun den Knoten Annotations-Dialog (siehe [14.10.1](#)).
3. Der Benutzer gibt dort den gewünschten Text für die Knoten-Annotation ein.
4. Nach Abschluss der Eingabe wählt der Benutzer OK aus.

5. GIANT übernimmt die neue Annotation. Die eingegebene Annotation muss allerdings mindestens ein Zeichen haben, sonst erscheint eine Fehlermeldung.

5.1.5. Alternativen

- Bei Punkt 3:
Der Benutzer kann die Eingabe der neuen Knoten-Annotation jeder Zeit mit Cancel abbrechen.

5.2. UC: Knoten-Annotation ändern

Dient zum Ändern bestehender Knoten-Annotationen (siehe auch [12.2.4](#)). Dieser UseCase ist auch für das Anzeigen von Knoten-Annotationen vorgesehen.

5.2.1. Vorbedingung

1. Es gibt mindestens einen annotierten Fenster-Knoten (siehe auch [15.1.3](#)).

5.2.2. Nachbedingung Erfolg

1. Die Änderung an der Knoten-Annotation ist noch nicht in der Verwaltungsdatei für Knoten-Annotationen eingetragen (siehe [12.1.7](#)). Dies geschieht erst beim nächsten Speichern des gesamten Projektes.
2. Die Änderung der Knoten-Annotation ist dem System bekannt und wird bei der nächsten Ausführung dieses UseCases angezeigt.

5.2.3. Nachbedingung Fehlschlag

1. Das System bleibt im bisherigen Zustand.

5.2.4. Beschreibung

1. Der Benutzer führt einen Rechtsklick auf den Fenster-Knoten, bei dem die Knoten-Annotation des zugehörigen IML-Knotens geändert werden soll, durch und wählt im Popup Menü für Knoten den Eintrag „Annotation – Change“ aus (siehe [14.4.2.1](#)).
2. GIANT zeigt nun den Knoten Annotations-Dialog (siehe [14.10.1](#)). Hier wird der Text für die bisherige Knoten-Annotation dargestellt.
3. Der Benutzer ändert den Text für die Knoten-Annotation entsprechend ab. Der Benutzer wird aber nicht gezwungen die bestehende Knoten-Annotation zu ändern.
4. Nach Abschluss der Eingabe betätigt der Benutzer den OK-Button.
5. GIANT übernimmt die vorgenommenen Änderungen an der Knoten-Annotation. Die neue Knoten-Annotation muss allerdings mindestens ein Zeichen haben, ansonsten erscheint eine Fehlermeldung.

5.2.5. Alternativen

- Generell:
Der Benutzer kann das Ändern der Knoten-Annotation mit Cancel abbrechen.

5.3. UC: Knoten-Annotation löschen

Löscht eine bestehende Knoten-Annotationen (siehe auch [12.2.4](#)).

5.3.1. Vorbedingung

1. Es gibt mindestens einen annotierten Fenster-Knoten (siehe [15.1.3](#)).

5.3.2. Nachbedingung Erfolg

1. Die Knoten-Annotation ist noch nicht aus der Verwaltungsdatei für Knoten-Annotationen entfernt. Dies geschieht erst beim nächsten Speichern des gesamten Projektes.
2. Das Entfernen der Knoten-Annotation ist dem System bekannt und wird entsprechend angezeigt.

5.3.3. Nachbedingung Fehlschlag

1. Das System bleibt im bisherigen Zustand.

5.3.4. Beschreibung

1. Der Benutzer führt einen Rechtsklick auf den Fenster-Knoten, dessen Knoten-Annotation gelöscht werden soll, durch und wählt im Popup Menü für Knoten den Eintrag „Annotation – Delete“ aus (siehe [14.4.2.1](#)).
2. GIANT löscht die entsprechende Knoten-Annotation.

5.3.5. Alternativen

- Generell:
Es sind keine alternativen Abläufe vorgesehen.

5.4. UC: Nicht referenzierte Knoten-Annotationen löschen

Dieser UseCase soll es ermöglichen, nicht mehr benötigte Knoten-Annotationen automatisch entfernen zu lassen. Es werden alle bestehenden Knoten-Annotationen des Projektes, für die es in Anzeigefenstern und IML-Teilgraphen keine Fenster-Knoten und Graph-Knoten gibt, gelöscht.

5.4.1. Vorbedingung

1. Ein Projekt ist geladen.

5.4.2. Nachbedingung Erfolg

1. Die Knoten-Annotationen sind noch nicht aus der Verwaltungsdatei für Knoten-Annotationen (siehe [12.1.7](#)) entfernt. Dies geschieht erst beim nächsten Speichern des Projekts.

5.4.3. Nachbedingung Fehlschlag

1. Das System bleibt im bisherigen Zustand.

5.4.4. Beschreibung

1. Der Benutzer wählt im Untermenü „Tools“ des Hauptmenüs den Eintrag „Delete Unreferenced Annotations“ (siehe [14.3.2.2](#)) aus.
2. GIANT zeigt eine Sicherheitsabfrage und fordert den Benutzer zur Bestätigung auf (siehe [14.12](#)).
3. Der Benutzer bestätigt die Sicherheitsabfrage mit Yes.
4. GIANT löscht alle Knoten-Annotationen für die es keine Fenster-Knoten und Graph-Knoten gibt, d.h. jede Knoten-Annotation deren IML-Knoten weder in einem Anzeigefenster als Fenster-Knoten visualisiert noch als Graph-Knoten Bestandteil eines IML-Teilgraphen ist, wird entfernt.

5.4.5. Alternativen

- Bei Punkt [3](#):
Der Benutzer bricht den UseCase mit No ab.

6. Selektionen

Alle UseCases, die im Zusammenhang mit Selektionen stehen, werden hier beschrieben.

6.1. UC: Selektion zur aktuellen Selektion machen

Dieser UseCase dient dazu, eine Selektion zur aktuellen Selektion zu machen. Dies ist nötig, da nur die aktuelle Selektion mittels der Maus modifiziert werden kann (siehe auch [3.4.3](#)).

6.1.1. Vorbedingung

1. Es gibt mindestens ein Anzeigefenster mit mindestens zwei Selektionen.

6.1.2. Nachbedingung Erfolg

1. Die vorherige aktuelle Selektion ist nicht mehr aktuell.
2. Die gewählte Selektion ist nun die aktuelle Selektion und als solche in der Selektionsauswahlliste (siehe [14.4.4](#)) gekennzeichnet.
3. Die Fenster-Knoten und Fenster-Kanten der jetzt aktuellen Selektion sind entsprechend hervorgehoben (siehe [15.3.1](#)).

6.1.3. Nachbedingung Fehlschlag

1. Es sind keine Fehlerfälle vorgesehen.

6.1.4. Beschreibung

1. Der Benutzer führt mit der linken Maustaste einen Doppelklick auf eine nicht aktuelle Selektion in der Selektionsauswahlliste (siehe [14.4.4](#)) durch. Ausgeblendete Selektionen (siehe UseCase [7.1](#)) können nicht zur aktuellen Selektion gemacht werden.
2. GIANT macht die entsprechende Selektion zur aktuellen Selektion.

6.1.5. Alternativen

- Generell:
Es sind keine alternativen Abläufe vorgesehen.

6.2. UC: Selektion graphisch hervorheben

Dieser UseCase dient zum Hervorheben von Selektionen innerhalb eines Anzeigefensters (siehe auch [15.3](#)).

6.2.1. Vorbedingung

1. Es gibt ein Anzeigefenster mit mindestens einer Selektion.

6.2.2. Nachbedingung Erfolg

1. Die Selektion ist im entsprechenden Anzeigefenster hervorgehoben.
2. War bereits eine andere Selektion mit der vom Benutzer gewählten Farbe hervorgehoben (die Farbe, welche im Popup-Menü der Selektionsauswahlliste unter „Highlight Selection“ gewählt wurde – siehe [14.4.4](#)), so ist diese Selektion nicht mehr hervorgehoben.

6.2.3. Nachbedingung Fehlschlag

1. Es sind keine Fehlerfälle vorgesehen.

6.2.4. Beschreibung

1. Der Benutzer startet den UseCase mit einem Rechtsklick auf die gewünschte Selektion in der Selektionsauswahlliste (siehe [14.4.4](#)) und wählt im zugehörigen Popup-Menü das Untermenü „Highlight Selection“ und dort die gewünschte Farbe aus.
2. GIANT hebt die Selektion mit der ausgewählten Farbe hervor.

6.2.5. Alternativen

- Generell:
Es sind keine alternativen Abläufe vorgesehen.

6.3. UC: Graphische Hervorhebung einer Selektion aufheben

Dieser UseCase dient dazu, die Hervorhebung von Selektionen innerhalb eines Anzeigefensters aufzuheben.

6.3.1. Vorbedingung

1. Es gibt ein Anzeigefenster mit mindestens einer hervorgehobenen Selektion.

6.3.2. Nachbedingung Erfolg

1. Die Selektion ist im entsprechenden Anzeigefenster nicht mehr hervorgehoben.

6.3.3. Nachbedingung Fehlschlag

1. Es sind keine Fehlerfälle vorgesehen.

6.3.4. Beschreibung

1. Der Benutzer startet den UseCase mit einem Rechtsklick auf die gewünschte Selektion in der Selektionsauswahlliste (siehe [14.4.4](#)) und wählt im zugehörigen Popup-Menü den Eintrag „Unhighlight Selection“ aus.
2. GIANT setzt die Hervorhebung der Selektion zurück.

6.3.5. Alternativen

- Generell:
Es sind keine alternativen Abläufe vorgesehen.

6.4. UC: Neue Selektion anlegen

Mit diesem UseCase können neue, leere Selektionen angelegt werden.

6.4.1. Vorbedingung

1. Es gibt mindestens ein geöffnetes Anzeigefenster.

6.4.2. Nachbedingung Erfolg

1. Eine neue Selektion mit dem vorgegebenen Namen ist angelegt und erscheint in der Liste der Selektionen (siehe [14.4.4](#)).
2. Diese neue Selektion hat keinen Inhalt (keine selektierten Fenster-Knoten und Fenster-Kanten).

6.4.3. Nachbedingung Fehlschlag

1. Das System bleibt im bisherigen Zustand.

6.4.4. Beschreibung

1. Der Benutzer startet den UseCase mit einem Rechtsklick auf die Selektionsauswahlliste (siehe [14.4.4](#)) und wählt im Popup-Menü den Eintrag „New Selection“ aus.
2. GIANT öffnet den allgemeinen Texteingabedialog (siehe [14.7](#)).
3. Der Benutzer gibt dort einen zulässigen Namen (siehe [3.3.2](#)) für die neue Selektion ein und bestätigt mit OK.
Hat bereits eine andere Selektion innerhalb des Anzeigefensters den selben Namen, erscheint eine Fehlermeldung.
4. GIANT erzeugt die neue Selektion.

6.4.5. Alternativen

- Bei Punkt 3:
Der Benutzer bricht die Verarbeitung mit Cancel ab.

6.5. UC: Selektion kopieren

Dieser UseCase dient zum Kopieren von Selektionen innerhalb eines Anzeigefensters.

6.5.1. Vorbedingung

1. Es gibt mindestens ein geöffnetes Anzeigefenster mit mindestens einer Selektion.

6.5.2. Nachbedingung Erfolg

1. Eine neue Selektion mit entsprechendem Namen ist angelegt und erscheint in der Liste der Selektionen (siehe [14.4.4](#)).
2. Die neue Selektion umfasst die selben Fenster-Knoten und Fenster-Kanten wie die Selektion, von der kopiert wurde.

6.5.3. Nachbedingung Fehlschlag

1. Das System bleibt im bisherigen Zustand.

6.5.4. Beschreibung

1. Der Benutzer startet den UseCase durch Rechtsklick auf die zu kopierende Quell-Selektion in der Selektionsauswahlliste (siehe [14.4.4](#)) und wählt in dem Popup-Menü den Eintrag „Copy Selection“ aus.
2. GIANT öffnet den allgemeinen Texteingabedialog (siehe [14.7](#)).
3. Der Benutzer gibt dort einen zulässigen Namen für die neue Selektion ein und bestätigt mit OK (siehe [3.3.2](#)).
Hat bereits eine andere Selektion innerhalb des Anzeigefensters den selben Namen erscheint eine Fehlermeldung.
4. GIANT kopiert die Quell-Selektion und legt eine neue Selektion an.

6.5.5. Alternativen

- Bei Punkt 3:
Der Benutzer bricht die Verarbeitung mit Cancel ab.

6.6. UC: Selektion löschen

Dieser UseCase dient zum Löschen von Selektionen innerhalb eines Anzeigefensters. Hierdurch bleiben die Fenster-Knoten und Fenster-Kanten unverändert. Die Standard-Selektion (siehe 3.4.2) kann nicht gelöscht werden.

6.6.1. Vorbedingung

1. Es gibt mindestens ein geöffnetes Anzeigefenster mit mindestens zwei Selektionen.

6.6.2. Nachbedingung Erfolg

1. Die entsprechende Selektion ist gelöscht.
2. Wurde die aktuelle Selektion gelöscht (siehe 3.4.3), so wird die Standard-Selektion zur aktuellen Selektion.
3. Die Fenster-Knoten und Fenster-Kanten, die zur gelöschten Selektion gehören, werden nicht gelöscht.
4. War die Selektion hervorgehoben, so wird die Hervorhebung der Fenster-Knoten und Fenster-Kanten aufgehoben.
5. Waren Fenster-Knoten und Fenster-Kanten der gelöschten Selektion ausgeblendet (siehe Use-Case 7.1), so werden diese Fenster-Knoten und Fenster-Kanten wieder eingeblendet.

6.6.3. Nachbedingung Fehlschlag

1. Das System bleibt im bisherigen Zustand.

6.6.4. Beschreibung

1. Der Benutzer startet den UseCase durch Rechtsklick auf die zu löschende Selektion in der Selektionsauswahlliste (siehe 14.4.4) und wählt aus dem Popup-Menü den Eintrag „Delete Selection“ aus. Falls der Benutzer die Standard-Selektion (siehe 3.4.2) ausgewählt hat, ist der Eintrag „Delete Selection“ deaktiviert.
2. GIANT löscht die entsprechende Selektion.

6.6.5. Alternativen

- Generell:
Es sind keine alternativen Abläufe vorgesehen.

6.7. UC: Selektionen manuell modifizieren

Jeweils die aktuelle Selektion kann mittels der Maus modifiziert werden (siehe auch [3.4.3](#)).

6.7.1. Vorbedingung

1. Es gibt mindestens ein geöffnetes Anzeigefenster mit mindestens einer Selektion.

6.7.2. Nachbedingung Erfolg

1. Die entsprechenden Änderungen an der Selektion werden von GIANT sofort durchgeführt und übernommen.

6.7.3. Nachbedingung Fehlschlag

1. Es sind keine Fehlerfälle vorgesehen.

6.7.4. Beschreibung

1. Falls noch nicht der Fall, macht der Benutzer die zu modifizierende Selektion zur aktuellen Selektion (siehe [6.1](#)).
2. Mittels der unter [3.4.1](#) beschriebenen Möglichkeiten fügt der Benutzer der Selektion neue Fenster-Knoten und Fenster-Kanten hinzu oder entfernt bestehende Fenster-Knoten und Fenster-Kanten aus der Selektion.

6.7.5. Alternativen

- Generell:
Es sind keine alternativen Abläufe vorgesehen.

6.8. UC: Selektion aus IML-Teilgraph erzeugen

Leitet eine Selektion aus einem IML-Teilgraphen ab (siehe [3.6.1](#)).

6.8.1. Vorbedingung

1. Es gibt mindestens ein geöffnetes Anzeigefenster.
2. Es gibt mindestens einen IML-Teilgraphen.

6.8.2. Nachbedingung Erfolg

1. Im Ziel-Anzeigefenster wurde eine neue Selektion mit dem entsprechenden Namen erzeugt.

6.8.3. Nachbedingung Fehlschlag

1. Das System bleibt im bisherigen Zustand.

6.8.4. Beschreibung

1. Der Benutzer führt einen Rechtsklick auf den Quell-IML-Teilgraphen in der entsprechenden Liste im Hauptfenster (siehe [14.3.5](#)) aus und wählt im dazugehörigen Popup-Menü (siehe [14.3.5.2](#)) den Eintrag „Create Window Selection“ aus.
2. Die Statuszeile im Hauptfenster zeigt an „Please select window for Insertion of new window selection“, der Mauszeiger verwandelt sich in ein Fadenkreuz.
3. Der Nutzer klickt auf den sichtbaren Anzeigehalt des Anzeigefensters, in dem er die neue Selektion erstellen will (Ziel-Anzeigefenster).
4. GIANT zeigt den allgemeinen Texteingabedialog (siehe [14.7](#)).
5. Der Benutzer gibt einen Namen (siehe auch [3.3.2](#)) für die neu zu erstellende Selektion ein und bestätigt mit OK.
6. Die Statuszeile im Hauptfenster wechselt wieder zur normalen Anzeige. GIANT erzeugt gemäß der unter Abschnitt [3.6.1](#) beschriebenen Konvention im Ziel-Anzeigefenster eine neue Selektion als Ableitung aus dem Quell-IML-Teilgraphen.

6.8.5. Alternativen

- Bei Punkt [3](#):
Der Benutzer bricht den UseCase durch einen Rechtsklick ab.
- Bei Punkt [5](#):
Der Benutzer bricht den UseCase mit Cancel ab.

6.9. UC: Mengenoperationen auf 2 Selektionen

Zusätzlich zu den Möglichkeiten der Anfragesprache GSL (siehe Kapitel 10) kann der Benutzer die gängigen Mengenoperationen, wie Mengenvereinigung, Schnitt und Differenz, auch direkt über einen entsprechenden Dialog (siehe 14.8) ausführen. Das Vorgehen ist analog zu dem UseCase 8.6.

6.9.1. Vorbedingung

1. Es gibt ein geöffnetes Anzeigefenster mit mindestens zwei Selektionen.

6.9.2. Nachbedingung Erfolg

1. Eine neue Selektion mit entsprechendem Namen (eingegeben unter TARGET) ist angelegt und erscheint in der Liste der Selektionen (siehe 14.4.4).
2. Im Falle einer Mengenvereinigung umfasst, die neue Selektion TARGET alle Fenster-Knoten und Fenster-Kanten aus der LEFT_SOURCE Selektion und der RIGHT_SOURCE Selektion.
3. Im Falle einer Mengendifferenz umfasst, die neue Selektion TARGET alle Fenster-Knoten und Fenster-Kanten aus der LEFT_SOURCE Selektion, die nicht Bestandteil der RIGHT_SOURCE Selektion sind.
4. Im Falle eines Mengenschnitts umfasst, die neue Selektion TARGET alle Fenster-Knoten und Fenster-Kanten, die der LEFT_SOURCE Selektion und der RIGHT_SOURCE Selektion gemeinsam angehören.

6.9.3. Nachbedingung Fehlschlag

1. Das System bleibt im bisherigen Zustand.

6.9.4. Beschreibung

1. Der Benutzer startet den UseCase über durch Auswahl des Menüpunktes „Selection Set Operation“ aus dem Popup-Menü in der Selektionsauswahlliste (siehe 14.4.4).
2. GIANT öffnet den Set-Operation-Dialog (siehe 14.8).
3. Der Benutzer wählt dort die beiden Quell-Selektionen (LEFT_SOURCE und RIGHT_SOURCE) aus, bestimmt die durchzuführende Mengenoperation und gibt unter TARGET den Namen der neu zu erzeugenden Selektion ein (gültige Namen siehe 3.3.2). Dann bestätigt er die Eingabe mit OK.
4. GIANT führt die Mengenoperation aus.

6.9.5. Alternativen

- Generell:
Der Benutzer bricht die Eingabe der Daten mit Cancel ab.

7. Filter

Hier werden die Filter zum Aus- und Einblenden von Fenster-Knoten und Fenster-Kanten beschrieben.

7.1. UC: Selektionen ausblenden

Mit diesem UseCase können Selektionen innerhalb eines Anzeigefensters ausgeblendet werden. Die Standard-Selektion (siehe [3.4.2](#)) und die aktuelle Selektion (siehe [3.4.3](#)) können nicht ausgeblendet werden.

7.1.1. Vorbedingung

1. Es gibt mindestens ein geöffnetes Anzeigefenster mit mindestens zwei Selektionen.

7.1.2. Nachbedingung Erfolg

1. Alle zu der Selektion gehörenden Fenster-Knoten und Fenster-Kanten sind ausgeblendet, d.h. sie sind im Anzeigefenster nicht mehr sichtbar. Dies trifft auch für Fenster-Knoten und Fenster-Kanten, die noch zu weiteren Selektionen gehören, zu.
2. Die ausgeblendeten Selektionen können nicht zur aktuellen Selektion gemacht werden (siehe [6.1](#)) und damit nicht mehr direkt bearbeitet werden, d.h. die Menge der selektierten Fenster-Knoten und Fenster-Kanten kann nicht mehr abgeändert werden (siehe [3.4.1](#)).
3. Die Fenster-Knoten und Fenster-Kanten sind aber immer noch Bestandteil des Anzeigefensters und können über den folgenden UseCase (siehe [7.2](#)) wieder zur Anzeige gebracht werden.

7.1.3. Nachbedingung Fehlschlag

1. Es sind keine Fehlerfälle vorgesehen.

7.1.4. Beschreibung

1. Der Benutzer führt einen Rechtsklick mit der Maus auf die auszublendende Selektion in der Selektionsauswahlliste (siehe [14.4.4](#)) durch und wählt im Popup-Menü den Eintrag „Hide Se-

lection“ aus. Diese Funktionalität kann nicht auf die aktuelle Selektion oder auf die Standard-Selektion angewendet werden.

2. GIANT blendet die Selektion aus.

7.1.5. Alternativen

- Generell:
Es sind keine alternativen Abläufe vorgesehen.

7.2. UC: Selektionen einblenden

Mit diesem UseCase können ausgeblendete Selektionen wieder eingeblendet werden.

7.2.1. Vorbedingung

1. Es gibt mindestens ein geöffnetes Anzeigefenster mit mindestens einer ausgeblendeten Selektion.

7.2.2. Nachbedingung Erfolg

1. Die Selektion ist wieder eingeblendet, alle zu ihr gehörenden Fenster-Knoten und Fenster-Kanten sind im Anzeigehalt sichtbar dargestellt (auch wenn sie noch zu weiteren Selektionen gehören, die ausgeblendet sind).

7.2.3. Nachbedingung Fehlschlag

1. Es sind keine Fehlerfälle vorgesehen.

7.2.4. Beschreibung

1. Der Benutzer führt einen Rechtsklick mit der Maus auf eine ausgeblendete Selektion in der Selektionsauswahlliste (siehe [14.4.4](#)) durch und wählt im Popup-Menü den Eintrag „Show Selection“ aus.
2. GIANT blendet die Selektion ein.

7.2.5. Alternativen

- Generell:
Es sind keine alternativen Abläufe vorgesehen.

7.3. UC: Alles einblenden

Mit diesem UseCase können alle ausgeblendeten Fenster-Knoten und Fenster-Kanten eines Anzeigefensters wieder eingeblendet werden.

7.3.1. Vorbedingung

1. Es gibt mindestens ein geöffnetes Anzeigefenster mit mindestens einer ausgeblendeten Selektion.

7.3.2. Nachbedingung Erfolg

1. Alle Selektionen des Anzeigefensters sind wieder eingeblendet.
2. Alle Fenster-Knoten und Fenster-Kanten sind wieder im Anzeigehalt des Anzeigefensters sichtbar dargestellt.

7.3.3. Nachbedingung Fehlschlag

1. Es sind keine Fehlerfälle vorgesehen.

7.3.4. Beschreibung

1. Der Benutzer führt einen Rechtsklick mit der Maus auf die Selektionsauswahlliste (siehe [14.4.4](#)) durch und wählt im Popup-Menü den Eintrag „Show all“ aus.
2. GIANT blendet alle ausgeblendeten Fenster-Knoten und Fenster-Kanten wieder ein.

7.3.5. Alternativen

- Generell:
Es sind keine alternativen Abläufe vorgesehen.

8. Teilgraphen

Alle UseCases, die sich mit IML-Teilgraphen befassen, sind in diesem Abschnitt zusammen gefasst.

8.1. UC: IML-Teilgraph graphisch hervorheben

Dieser UseCase dient zum Hervorheben von IML-Teilgraphen innerhalb von Anzeigefenstern (siehe auch [15.3.2](#)).

8.1.1. Vorbedingung

1. Es gibt mindestens einen IML-Teilgraphen.

8.1.2. Nachbedingung Erfolg

1. Der IML-Teilgraph ist in jedem geöffneten Anzeigefenster entsprechend hervorgehoben.
2. Der IML-Teilgraph, welcher vorher mit der gleichen Farbe hervorgehoben war, ist nicht mehr hervorgehoben.
3. Fenster-Knoten und Fenster-Kanten, die als Graph-Knoten bzw. Graph-Kanten Bestandteil eines hervorgehobenen IML-Teilgraphen sind, werden, auch wenn sie erst nach der Hervorhebung des IML-Teilgraphen in ein Anzeigefenster eingefügt werden, ebenfalls automatisch hervorgehoben.

8.1.3. Nachbedingung Fehlschlag

1. Es sind keine Fehlerfälle vorgesehen.

8.1.4. Beschreibung

1. Der Benutzer startet den UseCase über das Popup-Menü der Liste über die IML-Teilgraphen (siehe [14.3.5](#)) durch Rechtsklick auf den gewünschten IML-Teilgraphen. In dem Popup Menü (siehe [14.3.5.2](#)) wählt er das Untermenü „Highlight“ und dort die gewünschte Farbe aus.
2. GIANT hebt den IML-Teilgraphen in allen geöffneten Anzeigefenstern mit der ausgewählten Farbe hervor.

8.1.5. Alternativen

- Generell:
Es sind keine alternativen Abläufe vorgesehen.

8.2. UC: Graphische Hervorhebung von IML-Teilgraphen aufheben

Mit diesem UseCase kann die graphische Hervorhebung von IML-Teilgraphen aufgehoben werden.

8.2.1. Vorbedingung

1. Es gibt mindestens einen hervorgehobenen IML-Teilgraphen.

8.2.2. Nachbedingung Erfolg

1. Der IML-Teilgraph ist nicht mehr hervorgehoben.

8.2.3. Nachbedingung Fehlschlag

1. Es sind keine Fehlerfälle vorgesehen.

8.2.4. Beschreibung

1. Der Benutzer startet den UseCase durch Rechtsklick auf den hervorgehobenen IML-Teilgraphen in der Liste über die IML-Teilgraphen (siehe [14.3.5](#)) In dem zugehörigen Popup-Menü (siehe [14.3.5.2](#)) wählt er den Eintrag „Unhighlight In All Windows“ aus.
2. GIANT setzt die Hervorhebung des IML-Teilgraphen zurück.

8.2.5. Alternativen

- Generell:
Es sind keine alternativen Abläufe vorgesehen.

8.3. UC: IML-Teilgraph aus einer Selektion erzeugen

Leitet einen neuen IML-Teilgraphen aus einer Quell-Selektion ab (siehe auch Abschnitt [3.6.2](#)).

8.3.1. Vorbedingung

1. Es gibt mindestens ein geöffnetes Anzeigefenster mit mindestens einer Selektion.

8.3.2. Nachbedingung Erfolg

1. Es wurde ein neuer IML-Teilgraph mit entsprechendem Namen erzeugt.

8.3.3. Nachbedingung Fehlschlag

1. Das System bleibt im bisherigen Zustand.

8.3.4. Beschreibung

1. Der Benutzer führt einen Rechtsklick auf die Quell-Selektion in der Selektionsauswahlliste (siehe [14.4.4](#)) aus und wählt im Popup-Menü den Eintrag „Create New IML Subgraph from This Selection“ aus.
2. GIANT zeigt den allgemeinen Texteingabedialog (siehe [14.7](#)) an.
3. Der Benutzer gibt einen Namen für den neu zu erstellenden IML-Teilgraphen ein und bestätigt mit OK (gültige Namen für IML-Teilgraphen sind unter Abschnitt [3.3.2](#) spezifiziert).
Gibt der Benutzer hier keinen Namen ein, so vergibt GIANT automatisch einen Namen.
4. GIANT erzeugt gemäß der unter Abschnitt [3.6.2](#) beschriebenen Konvention einen neuen IML-Teilgraphen aus der Quell-Selektion.

8.3.5. Alternativen

- Bei Punkt 3:
Der Benutzer bricht den UseCase mit Cancel ab.

8.4. UC: IML-Teilgraph kopieren

Kopiert einen Quell-IML-Teilgraphen in einen neuen IML-Teilgraphen. Bestehende IML-Teilgraphen können nicht überschrieben werden.

8.4.1. Vorbedingung

1. Es gibt mindestens einen IML-Teilgraphen.

8.4.2. Nachbedingung Erfolg

1. Es wurde ein neuer IML-Teilgraph mit entsprechendem Namen erzeugt.
2. Der neue IML-Teilgraph hat alle Graph-Knoten und Graph-Kanten des Quell-IML-Teilgraphen.

8.4.3. Nachbedingung Fehlschlag

1. Das System bleibt im bisherigen Zustand.

8.4.4. Beschreibung

1. Der Benutzer führt einen Rechtsklick auf den zu kopierenden Quell-IML-Teilgraphen in der Liste über die IML-Teilgraphen im Hauptfenster aus (siehe [14.3.5](#)) und wählt im zugehörigen Popup-Menü (siehe [14.3.5.2](#)) den Eintrag „Copy IML Subgraph“ aus.
2. GIANT zeigt den allgemeinen Texteingabedialog (siehe [14.7](#)).
3. Der Benutzer gibt einen Namen für den neu zu erstellenden IML-Teilgraphen ein und bestätigt mit OK (gültige Namen siehe [3.3.2](#)).
Gibt der Benutzer hier keinen Namen ein, so vergibt GIANT automatisch einen Namen.
Gibt der Benutzer den Namen eines bereits vorhandenen IML-Teilgraphen ein, so erscheint eine Fehlermeldung.
4. GIANT kopiert den Quell-IML-Teilgraphen in einen neuen Teilgraphen.

8.4.5. Alternativen

- Bei Punkt 3:
Der Benutzer bricht den UseCase mit Cancel ab.

8.5. UC: IML-Teilgraph löschen

Dieser UseCase löscht einen IML-Teilgraphen. Die zugehörigen Fenster-Knoten und Fenster-Kanten in den Anzeigefenstern bleiben davon unberührt.

8.5.1. Vorbedingung

1. Es gibt mindestens einen IML-Teilgraphen.

8.5.2. Nachbedingung Erfolg

1. Der IML-Teilgraph wurde aus der Liste über die IML-Teilgraphen gelöscht (siehe [14.3.5](#)).
2. War der gelöschte IML-Teilgraph hervorgehoben, so ist die Hervorhebung der zugehörigen Fenster-Knoten und Fenster-Kanten in allen Anzeigefenstern aufgehoben.

8.5.3. Nachbedingung Fehlschlag

1. Es sind keine Fehlerfälle vorgesehen.

8.5.4. Beschreibung

1. Der Benutzer führt einen Rechtsklick auf den zu löschenden IML-Teilgraphen aus und wählt im Popup-Menü der Liste über die IML-Teilgraphen den Eintrag „Delete IML Subgraph“ aus (siehe [14.3.5.2](#)).
2. GIANT löscht den gewählten IML-Teilgraphen.

8.5.5. Alternativen

- Generell:
Es sind keine alternativen Abläufe vorgesehen.

8.6. UC: Mengenoperationen auf 2 IML-Teilgraphen

Ergänzend zu den Möglichkeiten der Anfragesprache (siehe Kapitel 10) kann der Benutzer die gängigen Mengenoperationen, wie Mengenvereinigung, Schnitt und Differenz, auch direkt über einen Dialog ausführen. Für eine genaue Beschreibung des Dialoges siehe auch Abschnitt 14.8. Das Vorgehen innerhalb dieses UseCases ist im Wesentlichen analog zu dem UseCase 6.9.

8.6.1. Vorbedingung

1. Es gibt mindestens zwei IML-Teilgraphen.

8.6.2. Nachbedingung Erfolg

1. Eine neuer IML-Teilgraph mit entsprechendem Namen (im Dialog eingegeben unter TARGET) ist angelegt und erscheint in der Liste über alle IML-Teilgraphen des Projektes (siehe 14.3.5).
2. Im Falle einer Mengenvereinigung umfasst der neue IML-Teilgraph TARGET alle Graph-Knoten und Graph-Kanten aus dem LEFT_SOURCE IML-Teilgraphen und dem RIGHT_SOURCE IML-Teilgraphen.
3. Im Falle einer Mengendifferenz umfasst der neue IML-Teilgraph TARGET alle Graph-Knoten und Graph-Kanten aus dem LEFT_SOURCE IML-Teilgraphen, die nicht Bestandteil des RIGHT_SOURCE IML-Teilgraphen sind.
4. Im Falle eines Mengenschnitts umfasst der neue IML-Teilgraph TARGET alle Graph-Knoten und Graph-Kanten, die sowohl dem LEFT_SOURCE IML-Teilgraphen als auch dem RIGHT_SOURCE IML-Teilgraphen angehören.
5. Hätte der IML-Teilgraph TARGET nach der Mengenoperation Graph-Kanten ohne zugehörigen Start- und Zielknoten, so sind diese Graph-Kanten aus dem IML-Teilgraphen TARGET entfernt worden (dieser Fall kann bei Mengenschnitt und Mengendifferenz eintreten).

8.6.3. Nachbedingung Fehlschlag

1. Das System bleibt im bisherigen Zustand.

8.6.4. Beschreibung

1. Der Benutzer startet den UseCase über das Popup-Menü der Liste über die IML-Teilgraphen (siehe 14.3.5.2) indem er einen Rechtsklick innerhalb der Liste ausführt und im Popup Menü den Eintrag „IML Subgraph Set Operation“ auswählt.
2. GIANT öffnet den Set-Operation-Dialog (siehe 14.8).

3. Der Benutzer wählt dort die beiden Quell-IML-Teilgraphen (LEFT_SOURCE und RIGHT_SOURCE) aus, bestimmt die auszuführende Mengenoperation und gibt unter TARGET den Namen des neu zu erzeugenden IML-Teilgraphen ein (gültige Namen für IML-Teilgraphen siehe [3.3.2](#)). Er bestätigt mit OK.
Existiert bereits ein IML-Teilgraph mit dem unter TARGET eingegebenen Namen, so erscheint eine Fehlermeldung.
4. GIANT führt die Mengenoperation aus und erzeugt den neuen IML-Teilgraphen.

8.6.5. Alternativen

- Bei Punkt [3](#):
Der Benutzer bricht die Eingabe der Daten mit Cancel ab.

9. Skripte

Hier werden UseCases zur Eingabe und Ausführung von GSL Skripten beschrieben.

9.1. UC: Neues Skript ausführen

Mit diesem UseCase kann ein Skript über den Skriptdialog (siehe [14.6](#)) eingegeben werden. Die Möglichkeiten der GSL sind im Detail in Kapitel [10](#) beschrieben.

9.1.1. Vorbedingung

1. Ein Projekt ist geladen.

9.1.2. Nachbedingung Erfolg

1. Das Skript wurde ausgeführt. Alle Ergebnisse liegen vor.

9.1.3. Nachbedingung Fehlschlag

1. Wurde der UseCase vor Beginn der Berechnung des Skriptes abgebrochen, bleibt das System im bisherigen Zustand.
2. Wurde der UseCase während der Ausführung des Skriptes abgebrochen, so werden bereits ausgeführte Aktionen (wie z.B. das Einfügen neuer Fenster-Knoten in Anzeigefenster) nicht wieder rückgängig gemacht.

9.1.4. Beschreibung

1. Der Benutzer startet den UseCase durch Auswahl des Eintrags „Execute GSL Script“ im Menü Tools (siehe [14.3.2.2](#)).
2. GIANT öffnet den Skriptdialog (siehe [14.6](#)).
3. Der Benutzer gibt dort im dafür vorgesehenen Textfeld das GSL Skript (siehe auch Kapitel [10](#)) ein und bestätigt mit „OK“.

4. GIANT prüft das eingegebene GSL Skript. Sollte das Skript nicht den Vorgaben der Grammatik (siehe Kapitel 10) entsprechen, erscheint eine Fehlermeldung (siehe 3.1) und das System kehrt zu Schritt 3 des UseCase zurück.
5. GIANT führt das Skript aus und teilt dem Benutzer den Fortschritt mittels eines Progress-Dialogs (siehe 14.15.2) mit. Während der Ausführung des Skripts ist das System mit Ausnahme des Progress-Dialogs gesperrt.

9.1.5. Alternativen

- Bei Punkt 3:
Der Benutzer bricht mit Cancel ab.
- Bei Punkt 5:
Die laufende Ausführung des Skripts kann vom Benutzer mit Cancel abgebrochen werden.

9.2. UC: Skript laden

Der Benutzer kann zusätzlich zur manuellen Eingabe von GSL Skripten (siehe 9.1) auch gespeicherte Skripte aus einer Datei (siehe 13.4) laden.

9.2.1. Vorbedingung

1. Ein Projekt ist geladen.

9.2.2. Nachbedingung Erfolg

1. Das Skript wurde ausgeführt. Alle Ergebnisse liegen vor.

9.2.3. Nachbedingung Fehlschlag

1. Wurde der UseCase vor Beginn der Berechnung des Skriptes abgebrochen, bleibt das System im bisherigen Zustand.
2. Wurde der UseCase während der Ausführung des Skriptes abgebrochen, so werden bereits ausgeführte Aktionen (wie z.B. das Einfügen neuer Fenster-Knoten in Anzeigefenster) nicht wieder rückgängig gemacht.

9.2.4. Beschreibung

1. Der Benutzer startet den UseCase durch Auswahl des Eintrags „Execute GSL Script“ im Menü Tools (siehe 14.3.2.2).
2. GIANT öffnet den Skriptdialog (siehe 14.6).
3. Der Benutzer betätigt im Dialog den Button „Open...“.
4. Daraufhin zeigt GIANT den Standard-Filechooser-Dialog (siehe 14.13.1).
5. Der Benutzer wählt eine vorhandene Skript-Datei (siehe 13.4) aus.
6. GIANT zeigt das aus der Datei geladene GSL Skript (siehe Kapitel 10) im Textfeld des Skriptdialogs an.
7. Falls gewünscht kann der Benutzer das GSL Skript im Textfeld noch manuell weiter modifizieren.
8. Der Benutzer startet die Berechnung des Skripts durch Betätigung des „Start Query“ im Skriptdialog (siehe 14.6).
9. GIANT prüft das geladene und eventuell modifizierte GSL Skript. Sollte das Skript nicht den Vorgaben der Grammatik (siehe Kapitel 10) entsprechen, erscheint eine Fehlermeldung (siehe 3.1) und das System kehrt zu Schritt 7 des UseCase zurück.

10. GIANT führt das Skript aus und teilt dem Benutzer den Fortschritt mittels eines Progress-Dialogs (siehe [14.15.2](#)) mit. Während der Ausführung des Skripts ist das System mit Ausnahme des Progress-Dialogs gesperrt.

9.2.5. Alternativen

- Bei Punkt [3](#):
Der Benutzer bricht den UseCase mit Cancel ab.
- Bei Punkt [4](#):
Der Benutzer bricht die Auswahl der Datei mit Cancel ab. Das System kehrt dann zu Schritt 2 bei der Abarbeitung des UseCase zurück.
- Bei Punkt [6](#):
Der Benutzer bricht den UseCase mit Cancel ab.
- Bei Punkt [10](#):
Die laufende Ausführung des Skripts kann vom Benutzer mit Cancel abgebrochen werden.

9.3. UC: Skript speichern

Mit diesem UseCase kann der Benutzer GSL Skripte aus dem Skriptdialog (siehe 14.6) in Dateien (siehe 13.4) speichern.

9.3.1. Vorbedingung

1. Der Skriptdialog (siehe 14.6) ist geöffnet und enthält in dem dafür vorgesehenen Textfeld entweder ein manuell eingegebenes oder ein aus einer Datei geladenes und eventuell modifiziertes GSL Skript.

9.3.2. Nachbedingung Erfolg

1. Eine Datei, welche das GSL Skript enthält, wurde angelegt.
2. GIANT zeigt den Skriptdialog, das gespeicherte GSL Skript ist weiterhin in dem Textfeld vorhanden.

9.3.3. Nachbedingung Fehlschlag

1. Das System bleibt im bisherigen Zustand.
2. Es wurde keine Datei erzeugt
3. GIANT zeigt weiterhin den Skriptdialog an (siehe 14.6) und alle dort getätigten Eingaben (insbesondere das GSL Skript im Textfeld des Dialoges) bleiben erhalten.

9.3.4. Beschreibung

1. Der Benutzer betätigt im Skriptdialog (siehe 14.6) den Button „Save As...“.
2. GIANT prüft ob das GSL Skript im Textfeld des Skriptdialogs den Vorgaben der Grammatik (siehe Kapitel 10) entspricht. Falls nicht, erscheint eine Fehlermeldung (siehe 3.1) und das System kehrt zu Schritt 1 des UseCases zurück.
3. GIANT öffnet den Standard-Filechooser-Dialog (siehe 14.13.1).
4. Der Benutzer gibt den Pfad und die Datei, in der das GSL Skript gespeichert werden soll, vor und bestätigt mit OK.
5. GIANT speichert das GSL Skript in der vorgegebenen Datei.

9.3.5. Alternativen

- Bei Punkt 4:
Der Benutzer bricht den UseCase mit Cancel ab.

10. GIANT Scripting Language

In diesem Kapitel werden die Fähigkeiten der GIANT Scripting Language (GSL) beschrieben. Syntax und Semantik der GSL sind einem separaten Dokument, der GIANT Scripting Language Spezifikation zu entnehmen.

10.1. GQSL

In der Version 1.0 dieses Dokuments wurde die Sprache GIANT Query and Skripting Language (GQSL) spezifiziert. Diese Sprache bot die selbe Funktionalität wie die hier beschriebene GSL. In dem Kundenreview am 14. und 15. April 2003 lehnte der Kunde GQSL jedoch ab, weil GQSL für Erweiterungen des Kunden einen zu hohen Anpassungsaufwand erfordere.

Deshalb wird stattdessen die GIANT Scripting Language (GSL) durch die CodeFabrik@Stuttgart nach den präzisierten Wünschen des Kunden entwickelt. GSL wird die selbe Funktionalität bieten wie ihr Vorgänger GQSL, wird jedoch auf einer vom Kunden angeregten, deutlich vereinfachten Syntax basieren.

Das Kapitel über GQSL wurde aus dieser Spezifikation entfernt. GSL wird in einem externen Dokument spezifiziert.

10.2. GSL Beschreibung

GSL Scripts werden von einem GSL Interpreter ausgeführt. Der GSL Interpreter ist Bestandteil von GIANT.

Die GSL dient dazu, IML-Teilgraphen und Selektionen aus einem IML-Graph anzufragen und auf diesen Aktionen auszuführen. Aktionen sind:

1. Erzeugen eines neuen Anzeigefensters
2. Hinzufügen der Graph-Knoten und Graph-Kanten eines IML-Teilgraphen in ein geöffnetes Anzeigefenster mit konfigurierbarem Layout der neuen Graph-Knoten.
3. Entfernen bestimmter Fenster-Knoten und Fenster-Kanten aus einem Anzeigefenster
4. Erzeugen einer neuen Selektion in einem Anzeigefenster
5. Ändern des Inhalts einer Selektion in einem Anzeigefenster
6. Löschen einer Selektion in einem Anzeigefenster

7. Erzeugen eines neuen IML-Teilgraphen
8. Ändern des Inhalts eines IML-Teilgraphen
9. Löschen eines IML-Teilgraphen

Selektionen und IML-Teilgraphen sind in GSL in bestimmten Variablen direkt sichtbar. Durch Inspektionen an diese Variablen kann der Inhalt einer Selektion oder eines IML-Teilgraphen ermittelt werden. Durch eine Zuweisung an eine solche Variable kann der Inhalt verändert werden.

In GSL können Vereinigung, Schnitt und Differenz von IML-Knoten- und IML-Kantenmengen dargestellt werden. Außerdem können durch Anfragen an die Reflektion gezielt bestimmte IML-Knoten oder IML-Kanten ausgewählt und als neue Menge in Variablen zwischengespeichert werden. Hierbei wird eine bestimmte Traversierungsstrategie für den IML-Graph vorgegeben. Ob ein dabei erreichter IML-Knoten oder eine IML-Kante in das Ergebnis aufgenommen wird, kann durch ein benutzerkonfiguriertes Prädikat entschieden werden.

Ein solches Prädikat entscheidet für jeden IML-Knoten und jede IML-Kante ob er/sie gewählt werden soll oder nicht. Dafür stehen dem Benutzer die Möglichkeiten zur Verfügung:

1. Inspektion der Attribute eines IML-Knoten, Vergleich der ermittelten Werte mit GSL-Variablen oder Konstanten oder Test ob der Wert in der Sprache eines regulären Ausdrucks enthalten ist
2. Abfrage der Knotenklasse eines IML-Knoten, Test ob diese Knotenklasse in einer bestimmten Klassenmenge enthalten ist
3. Abfrage der Kantenklasse einer IML-Kante, Test ob diese Kantenklasse in einer bestimmten Klassenmenge enthalten ist

11. Layoutalgorithmen

Hier werden die Layoutalgorithmen von GIANT beschrieben, mittels derer das Layout von Fenster-Knoten innerhalb von Anzeigefenstern automatisch berechnet werden kann.

GIANT bietet zwei verschiedene Layoutalgorithmen an:

1. Treelayout
2. Matrixlayout

11.1. Treelayout

Dieser Layoutalgorithmus ordnet Fenster-Knoten baumförmig auf dem Anzeigeeinhalt an.

11.1.1. Parameter

1. **Selektion**
Die Selektion, die layoutet werden soll.
2. **Wurzelknoten**
Der Benutzer wählt den Wurzelknoten des zu erstellenden Baumes durch Eingabe der ID des Knotens. Sollte die Selektion aus mehreren disjunkten Bäumen bestehen, so werden die weiteren Wurzelknoten automatisch ermittelt.
3. **Klassenmengen**
Wie auch im Begriffslexikon beschrieben werden Knotenklassen und Kantenklassen zu Klassenmengen zusammengefasst. Der Benutzer kann angeben, welche Klassenmengen für das Layout berücksichtigt werden sollen. Der Layoutalgorithmus berücksichtigt dann nur die Kantenklassen, die zu den gewählten Klassenmengen gehören. Werden keine Klassenmengen vorgegeben, so werden alle Kantenklassen für das Layout berücksichtigt.
4. **Zielposition**
Die Position, an die der Mittelpunkt des Wurzelknotens des Baums positioniert werden soll.

11.1.2. Beschreibung

Das Treelayout basiert auf Walkers Algorithmus. Dabei werden folgende Eigenschaften erreicht:

1. Alle Kinder eines Fenster-Knotens sind auf der gleichen Ebene

2. Alle Kinder eines Fenster-Knotens haben den gleichen horizontalen Abstand zueinander

Die Reihenfolge der Kinder von der `IML_Reflection` übernommen. In der Grundfunktionalität werden alle Fenster-Kanten betrachtet.

Der vertikale Abstand von Ebene zu Ebene richtet sich nach der Höhe des höchsten Fenster-Knotens (siehe [15.1.1](#)).

Falls die übergebene Selektion keinen Baum darstellt, werden die Fenster-Knoten in die Ebene eingegliedert, die bei einer Tiefensuche als erstes erreicht wird.

Zerfällt die Selektion in nicht-zusammenhängende Komponenten, so wird das Layout für jede dieser Komponente durchgeführt. Die disjunkten Bäume werden dann horizontal nebeneinander angeordnet.

Als Erweiterung kann der Algorithmus nur bestimmte Fenster-Kanten betrachten. Dabei werden im ersten Lauf nur die Fenster-Knoten betrachtet, die durch die gegebenen Fenster-Kanten erreicht werden können. Im zweiten Lauf werden Fenster-Knoten, die durch die gegebenen Fenster-Kanten nicht erreicht werden konnten, im Matrixlayout (siehe [11.2](#)) rechts von dem letzten Baum angeordnet.

11.2. Matrixlayout

Ein relativ simpler Layoutalgorithmus, der die Fenster-Knoten matrixförmig nahe beieinander auf dem Anzeigebereich anordnet.

11.2.1. Parameter

1. **Selektion**

Die Selektion, die layoutet werden soll

2. **Zielposition**

Die Position, an der die linke obere Ecke des Quadrats liegen soll.

11.2.2. Beschreibung

Im Matrixlayout werden die Fenster-Knoten in einem Rechteck ausgerichtet. Dabei ist die Zahl der Fenster-Knoten pro Zeile und Spalte gleich. Die Breite des Rechtecks wird mittels der abgerundeten Wurzel der Anzahl der zu layoutenden Knoten bestimmt; die Höhe ergibt sich automatisch. Die Fenster-Knoten werden mittels einer Breitensuche besucht und in dieser Besuchsreihenfolge in das Rechteck zeilenweise eingefügt. Dabei hat die obere Begrenzung jedes Fenster-Knotens einer Zeile die selbe Y-Koordinate. Der vertikale Abstand von Zeile zu Zeile richtet sich nach der Höhe des höchsten Fenster-Knotens (siehe [15.1.1](#)).

12. GIANT Projektverwaltung

In diesem Kapitel wird beschrieben, wie persistente Arbeitsergebnisse von GIANT strukturiert und gespeichert werden. Arbeitsergebnisse sind z.B. vom Benutzer erzeugte Anzeigefenster mit als Fenster-Knoten visualisierten IML-Knoten eines IML-Graphen.

12.1. Persistenz der Projekte

GIANT verwaltet persistente Informationen in sogenannten Projekten. Ein Projekt fasst alle Arbeitsergebnisse, die der Benutzer mittels GIANT für einen IML-Graphen erzeugen kann, wie z.B. Anzeigefenster und IML-Teilgraphen, logisch zusammen.

Wichtige Voraussetzungen:

1. Die gesamte in dieser Spezifikation beschriebene Funktionalität, die sich auf das Öffnen bestehender und das Erstellen neuer Projekte während der Laufzeit von GIANT bezieht, kann nur implementiert werden, falls die Reflektion zur Bauhaus IML-Graph Bibliothek dies unterstützt. Der Kunde muss hierzu in der Reflektion die Möglichkeit vorsehen, IML-Graphen zur Laufzeit laden und bereits geladene IML-Graphen zur Laufzeit aus dem Speicher entfernen zu können.
2. Zur sicheren Identifikation der IML-Graph Datei, die zu einem Projekt gehört, muss über die Reflektion eine eindeutige Prüfsumme für jeden IML-Graphen berechnet werden können.

Konventionen der Projektverwaltung:

1. Wird während des Betriebs von GIANT ein neues Projekt angelegt, so erhält es automatisch eine Projektdatei.
2. Ein Projekt besteht aus einem Verweis auf eine IML-Graph-Datei, auf die sich die gespeicherten Informationen beziehen, sowie aus den gespeicherten Informationen für IML-Teilgraphen, Anzeigefenster und Knoten-Annotationen.
3. Der Name eines bereits angelegten Projektes kann mit den Mitteln von GIANT nicht geändert werden (außer dadurch, dass man das Projekt unter neuem Namen neu speichert).
4. Der Benutzer kann beliebig viele Projekte anlegen.
5. In GIANT kann immer nur ein Projekt gleichzeitig geöffnet sein.
6. Während der Arbeit mit GIANT kann ein vorhandenes Projekt geladen oder ein neues Projekt angelegt werden.
7. Ein nicht gerade in GIANT geöffnetes Projekt kann vom Benutzer beliebig modifiziert werden. GIANT unterstützt dies durch den Einsatz von XML (incl. DTDs). Insbesondere kann

der Benutzer die Projektdatei (siehe [12.1.3](#)) modifizieren und so dem Projekt z.B. neue Verwaltungsdateien (siehe [12.1.7](#), [12.1.5](#) und [12.1.6](#)) hinzufügen oder die zugehörige IML-Graph Datei ändern.

Das Fehlerverhalten von GIANT gegenüber Fehlern aufgrund manueller Änderungen des Benutzers an einem Projekt (Editieren der Projektdatei und Ändern der Verwaltungsdateien) ist undefiniert.

8. Um Inkonsistenzen und Konflikte zu vermeiden sollten alle Dateien, die zu einem Projekt gehören, im Projektverzeichnis liegen (GIANT macht dies automatisch immer so). Man sollte in die Projektdatei also manuell keine Referenzen auf Verwaltungsdateien außerhalb der Projektverzeichnisses eintragen; verboten ist dies allerdings nicht. Das Verhalten von GIANT im Fehlerfall bleibt aber diesbezüglich undefiniert.

12.1.1. Überblick über die Struktur eines Projektes

Dieser Abschnitt liefert einen kurzen Überblick über die Struktur eines Projektes.

Alle Dateien, die zu einem Projekt gehören, werden von GIANT im selben Verzeichnis – dem Projektverzeichnis – abgelegt. Jedes Projekt besteht aus den folgenden Dateien:

1. Genau einer Projektdatei (siehe [12.1.3](#)).
2. Verwaltungsdateien für Anzeigefenster.
Jedes zum Projekt gehörende Anzeigefenster hat – sobald es zum ersten mal gespeichert wurde (siehe UseCase [4.11](#)) – seine eigene Verwaltungsdatei. Siehe hierzu auch [12.1.5](#).
3. Verwaltungsdateien für IML-Teilgraphen.
Jeder IML-Teilgraph des Projektes hat seine eigene Verwaltungsdatei (siehe [12.1.6](#)). Ein IML-Teilgraph verfügt nicht sofort nach seiner Erstellung über eine Verwaltungsdatei; diese Datei wird erst erzeugt, wenn das gesamte Projekt nach Erstellung des IML-Teilgraphen gespeichert wird (siehe z.B. UseCase [4.6](#)).
4. Genau einer Verwaltungsdatei für alle Knoten-Annotationen (siehe [12.1.7](#)).

12.1.2. Das Projektverzeichnis

1. GIANT speichert alle Verwaltungsdateien (siehe [12.1.7](#), [12.1.5](#) und [12.1.6](#)) automatisch im Projektverzeichnis.
2. In einem Projektverzeichnis darf nur ein Projekt (insb. nur eine Projektdatei) abgelegt werden.

12.1.3. Die Projektdatei

1. Die Projektdatei liegt als XML-Datei vor.
2. Die Projektdatei befindet sich im Projektverzeichnis und enthält Informationen, die zur Identifikation des zu einem Projekt gehörenden IML-Graphen nötig sind.
3. Der Name der Projektdatei entspricht dem Namen des Projektes.

4. Die Projektdatei enthält Referenzen zu allen Dateien, die Bestandteil des Projektes sind (Verwaltungsdateien für IML-Teilgraphen und Anzeigefenster, sowie die Verwaltungsdatei für Knoten-Annotationen). Verwaltungsdateien, die zwar im Projektverzeichnis liegen, zu denen aber keine Referenz in der Projektdatei existiert, gehören nicht zum Projekt.
5. Der Pfad zu der Datei, die den IML-Graphen enthält, ist in der Projektdatei gespeichert.

12.1.4. Prüfung der IML-Graph Datei

Beim Laden eines Projektes wird überprüft, ob die in der Projektdatei gespeicherte Prüfsumme der Prüfsumme der zu ladenden IML-Graph Datei entspricht. Das Verhalten von GIANT für den Fall, dass eine IML-Graph Datei geladen wird, die zwar die passende Prüfsumme hat, aber nicht den IML-Graphen enthält, der dem Projekt eigentlich zu Grunde liegt, ist undefiniert.

12.1.5. Verwaltungsdateien für Anzeigefenster

1. Diese Datei ist eine Binärdatei.
2. Zu jedem Anzeigefenster gibt es eine Verwaltungsdatei.
3. Diese Verwaltungsdatei enthält alle Informationen zur kompletten Rekonstruktion eines Anzeigefensters. Alle Informationen werden in binärer Form gespeichert.

Innerhalb dieser Verwaltungsdatei werden folgende Informationen gespeichert:

1. Der komplette Anzeigeeinhalt (alle visualisierten Fenster-Knoten und Fenster-Kanten mit Position).
2. Der Zustand (Zoomstufe und Position) des sichtbaren Anzeigeeinhaltes.
3. Alle Pins (gespeicherte Zustände des sichtbaren Anzeigeeinhaltes).
4. Alle Selektionen des Anzeigefensters.
5. Der Zustand aller Selektionen (welche die aktuelle Selektion ist, sowie die Art der Hervorhebung).
6. Der Name des gewählten Visualisierungsstils (nicht aber die genaue Definition des Stils; diese wird in XML-Dateien vorgenommen – siehe [13.3](#)).

Der Name des für das Anzeigefenster gewählten Visualisierungsstils wird zwar gespeichert, da die Visualisierungsstile aber völlig unabhängig von den Projekten über XML-Dateien (siehe [13.3](#)) konfiguriert bzw. definiert werden, kann nicht garantiert werden, dass der gewünschte Stil beim Laden des Projektes auch wieder gefunden wird, in diesem Fall wird dann ein „Standard-Stil“ verwendet.

12.1.6. Verwaltungsdateien für IML-Teilgraphen

1. Diese Datei ist eine Binärdatei.
2. Für jeden IML-Teilgraphen gibt es eine eigene Verwaltungsdatei.

3. Sämtliche erzeugten IML-Teilgraphen werden in Verwaltungsdateien in binärer Form gespeichert.

12.1.7. Die Verwaltungsdatei für Knoten-Annotationen

1. Diese Datei liegt als XML Datei vor.
2. Alle Knoten-Annotationen des Projektes werden in genau einer Verwaltungsdatei gespeichert.

12.2. Grundlegendes Verhalten von GIANT beim Speichern von Projekten

12.2.1. „Alles Speichern“

Diese Funktionalität wird von entsprechenden UseCases genutzt. Hierbei werden alle Anzeigefenster, IML-Teilgraphen und Knoten-Annotationen in die Verwaltungsdateien geschrieben, wobei auch alle Änderungen an noch geöffneten Anzeigefenstern berücksichtigt werden; d.h. der aktuelle Zustand (z.B. die Position der Fenster-Knoten) der offenen Anzeigefenster wird ohne explizite Rückfrage beim Benutzer in die Verwaltungsdateien geschrieben. Der Zustand der persistenten Informationen in den Verwaltungsdateien entspricht nach Ausführung von „Alles Speichern“ exakt dem Zustand des innerhalb von GIANT geöffneten Projektes.

12.2.2. Persistenz von Anzeigefenstern

1. Beim Speichern von Anzeigefenstern in Verwaltungsdateien werden für jedes Anzeigefenster alle Informationen gespeichert, die nötig sind um den Zustand des Anzeigefensters zu rekonstruieren, der möglichst exakt dem Zustand bei der Speicherung des Anzeigefensters entspricht. Insbesondere werden hierbei die folgenden Bestandteile des Anzeigefensters gespeichert:

Alle Selektionen und Pins des Anzeigefensters.

Der exakte Zustand jeder Selektion; d.h. ob und wie die Selektion hervorgehoben ist, ob sie gefiltert ist und ob sie den Status der aktuellen Selektion hat.

Die Position der Fenster-Knoten und Fenster-Kanten des Anzeigefensters.

2. Für den gewählten Visualisierungsstil des Anzeigefensters wird nur der Name gespeichert, da die Visualisierungsstile über die Konfigurationsdateien beliebig editiert und auch gelöscht werden können. Wird beim Laden eines Anzeigefensters aus der Verwaltungsdatei kein Visualisierungsstil gefunden, dessen Name dem gespeicherten entspricht, so wird für das Anzeigefenster statt dessen ein Standard-Visualisierungsstil verwendet.
3. Sämtliche dem Projekt bekannten Anzeigefenster (alle Anzeigefenster zu denen es eine entsprechende Verwaltungsdatei gibt) werden in der Liste über die Anzeigefenster angezeigt.
4. Zu jedem Anzeigefenster eines Projektes gibt es eine Verwaltungsdatei. Bei neu erzeugten Anzeigefenstern wird diese Verwaltungsdatei beim ersten Speichern angelegt.

5. Wird ein geöffnetes Anzeigefenster geschlossen, so fragt GIANT nach, ob es eventuelle Änderungen speichern soll oder nicht. Falls ja, werden eventuelle Änderungen in die für das Anzeigefenster vorhandene Verwaltungsdatei geschrieben, anderenfalls bleibt der Zustand des Anzeigefensters nach der letzten Speicherung vorhanden (die zugehörige Verwaltungsdatei wird nicht verändert).
6. Modifikationen (z.B. das Verschieben von Fenster-Knoten) auf einem Anzeigefenster werden nicht automatisch nach deren Durchführung gespeichert (so kann notfalls ein Rückgängig durchgeführt werden).

12.2.3. Persistenz von IML-Teilgraphen

1. Alle in einem Projekt bereits vorhandenen IML-Teilgraphen werden in einer entsprechenden Liste angezeigt.
2. Neu erzeugte IML-Teilgraphen und Änderungen an bestehenden IML-Teilgraphen können nur über „Alles Speichern“ (siehe oben) gespeichert werden.
3. Wird das Programm beendet, ohne das zuvor „Alles Speichern“ ausgeführt worden ist, so gehen alle nicht gespeicherten Informationen zu den IML-Teilgraphen verloren (alle zwischenzeitlich ausgeführten Modifikationen und alle zwischenzeitlich neu erzeugten IML-Teilgraphen). Der Zustand nach dem letzten Speichern bleibt dann erhalten.
4. Modifikationen an bestehenden IML-Teilgraphen werden nicht automatisch gespeichert. Zu neu erzeugten IML-Teilgraphen wird nicht automatisch eine Verwaltungsdatei erzeugt.
5. IML-Teilgraphen können gelöscht werden. Falls vorhanden, wird dann auch die entsprechende Verwaltungsdatei ebenfalls sofort gelöscht.

12.2.4. Persistenz von Knoten-Annotationen

Hier wird beschrieben, wie Knoten-Annotationen von GIANT persistent verwaltet werden.

1. Änderungen bestehender oder neu erzeugte Knoten-Annotationen werden nur über die Funktionalität „Alles Speichern“ (siehe [12.2.1](#)) in die Verwaltungsdatei für Knoten-Annotationen (siehe [12.1.7](#)) geschrieben.
2. Einmal erzeugte Knoten-Annotationen werden jedem IML-Knoten mit der entsprechenden ID zugeordnet, unabhängig davon in welchem Anzeigefenster diese visualisiert sind. Ein Knoten kann auch annotiert sein, wenn er in keinem Anzeigefenster visualisiert ist. Wird ein annotierter Knoten gelöscht (aus einem Anzeigefenster entfernt), so wird der dazu vorhandene Eintrag für die Annotation in der Verwaltungsdatei nicht automatisch mit gelöscht.

13. Konfiguration von GIANT

Hier wird beschrieben, wie benutzerdefinierbare Einstellungen von GIANT gespeichert und verwaltet werden. In diesem Kapitel wird insbesondere spezifiziert, welche Einstellungen der Benutzer in welchen Konfigurationsdateien vornehmen kann.

13.1. Allgemeines

1. Sämtliche konfigurierbaren Einstellungen werden in XML-Dateien vorgenommen. Für die verschiedenen Dateitypen werden auch Dokumenttyp-Definitionen (DTDs) erstellt.
2. Es gibt genau eine „globale Konfigurationsdatei“. Zudem kann es beliebig viele weitere XML-Dateien zur Konfiguration der weiter unten beschriebenen Visualisierungsstile geben. Diese Dateien liegen in einem von GIANT fest vorgegebenen Verzeichnis.
3. Die unter Punkt 2 beschriebenen Dateien zur Konfiguration können vom Benutzer auch in einem separaten Unterverzeichnis seines Home-Verzeichnisses abgelegt werden. Findet GIANT beim Start ein derartiges Verzeichnis, so werden die Einstellungen aus der dort vorgefundenen Konfigurationsdatei, sowie die dort vorgefundenen Visualisierungsstile geladen.

13.2. Die globale Konfigurationsdatei

Diese Datei ist in XML verfasst. In ihr können die anschließend beschriebenen Einstellungen vorgenommen werden.

13.2.1. Verweise auf GSL-Skripte

Es können GSL-Skript-Makros definiert werden. Jeder Makro wird durch einen eindeutigen Namen und durch einen Verweis auf die Datei, welche das GSL-Skript enthält, spezifiziert. Der Makro kann dann aus einem Popup-Menü heraus aufgerufen werden, wobei dann das in der Datei stehende GSL Skript ausgeführt wird.

13.2.2. Farbe von Hervorhebungen

Einstellung der Farben, mittels derer Selektionen oder IML-Teilgraphen hervorgehoben werden.

1. Eine beliebige Farbe für die aktuelle Selektion.

2. Beliebige Farben für das Hervorheben weiterer Selektionen.
3. Beliebige Farben für das Hervorheben von IML-Teilgraphen.

13.2.3. Editor zur Anzeige des Quellcodes

Der Benutzer legt in der globalen Konfigurationsdatei fest, mit welchem Editor der zu IML-Knoten korrespondierende Quellcode automatisch angezeigt wird (unterstützt werden Emacs und vi).

13.3. Visualisierungsstile

Der Benutzer kann beliebig viele Visualisierungsstile definieren. Mittels dieser Visualisierungsstile kann die Darstellung von Fenster-Knoten und Fenster-Kanten dynamisch zur Laufzeit geändert werden (siehe auch [4.17](#)). Weitere Informationen zur Visualisierung von Fenster-Knoten und Fenster-Kanten sind unter den Abschnitten [15.2](#) und [15.1](#) zu finden.

Für die Visualisierungsstile gelten die folgenden Konventionen:

1. Für jeden Visualisierungsstil muss es eine entsprechende XML-Datei geben.
2. Ein Visualisierungsstil beschreibt, wie die Knoten und Kanten des IML-Graphen innerhalb eines Anzeigefensters graphisch dargestellt werden können.
3. In einem Visualisierungsstil können klassenspezifische Einstellungen vorgenommen werden, die nur für Knoten und Kanten gelten, die zu der entsprechenden Klasse gehören. Bei jeder klassenspezifischen Einstellung für Knoten und Kanten kann daher eine Liste der betroffenen Knoten- und Kantenklassen angegeben werden, für die diese Einstellungen gelten sollen.
4. Wird eine Kanten- oder eine Knotenklasse innerhalb eines Visualisierungsstils mehreren klassenspezifischen Einstellungen zugeordnet, führt dies zu keinem Fehler, es bleibt aber unspezifiziert, welche Einstellung tatsächlich genommen wird.
5. Es gibt jeweils eine Standard-Einstellung, die für alle Knoten- und Kantenklassen genommen wird, für die keine extra klassenspezifischen Einstellungen vorgenommen wurden.
6. Es gibt immer einen von GIANT fest vorgegeben Standard-Visualisierungsstil. Dieser Visualisierungsstil wird für neu erzeugte Anzeigefenster genommen. Kann GIANT den einem Anzeigefenster zugewiesenen Visualisierungsstil nicht finden (falls die zugehörige XML-Datei fehlt) wird ebenfalls der Standard-Visualisierungsstil genommen.

13.3.1. Name des Visualisierungsstils

Jeder Visualisierungsstil erhält einen Namen. Unter diesem Namen ist der Visualisierungsstil für jedes Anzeigefenster einzeln auswählbar.

13.3.2. Einstellungen innerhalb des Visualisierungsstils

1. Die Hintergrundfarbe im Anzeigefenster.

13.3.3. Klassenspezifische Einstellungen für Knoten

Folgende Einstellungen können für Knotenklassen vorgenommen werden. Es muss eine Standard-Einstellung erstellt werden, die für alle Knotenklassen angewendet wird, für die nichts anderes definiert ist.

1. Ein Icon für die Knotenklasse (Verweis auf eine entsprechende Bilddatei). Das Icon muss im Pixmap-Format vorliegen. Die Größe des Icons ist beliebig, bei zu großen Icons wird abgeschnitten. Es wird garantiert, dass Icons bis zu einer Größe von 32*32 Pixel komplett dargestellt werden können.
2. Eine Liste der Attribute der Knotenklasse, welche direkt innerhalb des Anzeigefensters in dem „Rechteck“ für den Knoten dargestellt werden sollen.
3. Die Farbe der Schrift, mit der die Informationen innerhalb des Knoten-Rechtecks dargestellt werden.
4. Die Füllfarbe des „Rechtecks“ in welchem die Attribute zu dem Knoten dargestellt werden.
5. Die Rahmenfarbe des „Rechtecks“.

13.3.4. Klassenspezifische Einstellungen für Kanten

Folgende Einstellungen können für Kantenklassen vorgenommen werden. Es muss eine Standard-Einstellung erstellt werden, die für alle Kantenklassen angewendet wird, für die nichts anderes definiert ist.

1. Die Farbe der Kante.
2. Die Farbe der Kantenbeschriftung.
3. Die Art der Linie der Kante (normal, gestrichelt).
4. Ob die Kante mit ihrer Kantenklasse beschriftet werden soll oder nicht.

Eine Kantenklasse wird durch die Knotenklasse des Start-Knotens und durch den Namen des Attributes, welches die Kante darstellt, eindeutig festgelegt (siehe Begriffslexikon). Will der Benutzer nun Einstellungen für Kantenklassen vornehmen, kann er sich entweder auf genau eine Kantenklasse „Start-Knotenklasse.Attributname“ beziehen oder mittels eines Platzhalters auch auf mehrere:

1. Der Benutzer kann mittels des Platzhalters „*“ eine klassenspezifische Einstellung für alle Kanten, deren Start-Knoten zu einer bestimmten Knotenklasse gehört, vornehmen – also „Start-Knotenklasse.*“.
2. Der Benutzer kann mittels des Platzhalters „*“ eine klassenspezifische Einstellung für alle Kanten, bei denen das sie repräsentierende Attribut den gleichen Namen hat, vornehmen – also „*.Attributname“.

13.4. Skript-Dateien

Eine Skript-Datei ist eine Textdatei, die genau ein GSL Skript enthält. Auf diese Art und Weise können komplexe Skripte gespeichert und wiederverwendet werden.

Geladen werden können solche Dateien beim Kommandozeilenaufruf (siehe [4.2](#)) oder im Dialog zur Eingabe eines Skripts (vgl. UseCase [9.2](#)). Für eine exakte Spezifikation der GSL siehe Kapitel [10](#).

14. Beschreibung der GUI

In diesem Kapitel werden die Dialoge und Menüs der graphischen Benutzerschnittstelle von GIANT beschrieben. Die Funktionalität dieser Dialoge wird im Detail bei den jeweiligen UseCases beschrieben.

14.1. Über die Benutzeroberfläche

1. GIANT besitzt eine graphische Benutzeroberfläche, die per Maus zu bedienen ist.
2. Die Interaktionssprache mit dem Benutzer ist Englisch.

14.2. Über dieses Kapitel

Hier werden alle Elemente der GUI beschrieben. Sofern nicht anders angegeben, sind hintereinander angegebene Elemente (z.B. Listeneinträge) immer von links nach rechts oder von oben nach unten beschrieben. Alle Ausgaben sind immer linksbündig formatiert, sofern nicht anders vermerkt.

14.3. Main Window

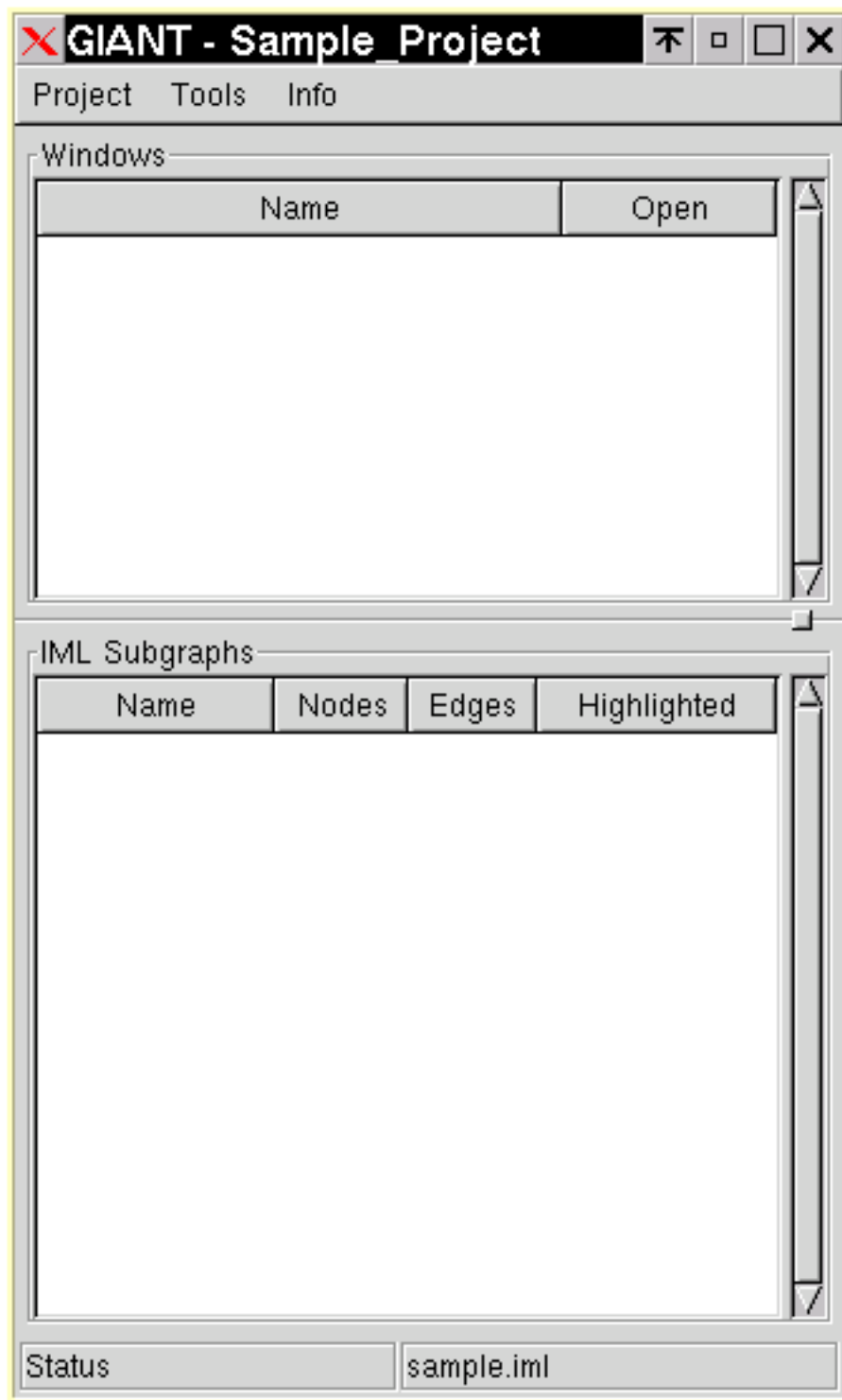


Abbildung 14.1.: Main-Window

Jede Instanz von GIANT hat genau ein Hauptfenster. Im Hauptfenster (siehe Abbildung 14.1) werden die Anzeigefenster und IML-Teilgraphen des aktuell geöffneten Projektes (siehe 12.1) in Listen angezeigt. Über Popup-Menüs können IML-Teilgraphen und Anzeigefenster manipuliert werden.

Im Fenster befinden sich zwei Listen, Window List (2 Spalten), und Subgraph List (4 Spalten).

14.3.1. Kopfzeile

In der Kopfzeile wird „GIANT - <Projektname>“ dargestellt.

14.3.2. Menüleiste (Main Window)

In der Menüleiste befinden sich folgende Einträge:

14.3.2.1. Untermenü Project

1. New Project
2. Load Project
3. Save Project
4. Save Project As...
5. Quit

14.3.2.2. Untermenü Tools

1. Delete Unreferenced Annotations
2. Execute GSL Script

14.3.2.3. Untermenü Info

1. About GIANT

14.3.3. Statuszeile

Die Statuszeile befindet sich ganz unten im Hauptfenster. In der Statuszeile werden Informationen zum aktuellen Zustand von GIANT dargestellt.

14.3.4. Window List

Im Fenster befindet sich zuoberst eine Liste „Window List“, welche alle Anzeigefenster des Projektes (offen und geschlossen anzeigt).

14.3.4.1. Inhalt Window List

Window List hat die Spalten „Name“ und „Open“. In „Name“ wird der Name aller Anzeigefenster dargestellt, in „Open“ befindet sich entweder ein Strich-Symbol (Fenster nicht geöffnet) oder ein Häkchensymbol (Fenster geöffnet).

14.3.4.2. Popup-Menü Window List

Beim Rechtsklick auf Window List öffnet sich ein Popup-Menü mit folgenden Menüpunkten:

1. New Window
2. Open Window
3. Close Window
4. Save Window
5. Delete Window
6. Rename Window

14.3.5. Subgraph List

Unter der Window List befindet sich eine Liste „Subgraph List“ über alle IML-Teilgraphen des Projektes.

14.3.5.1. Inhalt Subgraph List

Subgraph List hat die folgenden Spalten:

1. Name
2. Nodes
3. Edges
4. Highlighted

In der Spalte Name stehen die Namen der existierenden IML-Teilgraphen, in Nodes die Anzahl der Graph-Knoten des jeweiligen IML-Teilgraphen, in Edges die Anzahl der Graph-Kanten und in Highlighted befindet sich ein Kästchen in der Farbe, die für die Hervorhebung des IML-Teilgraphen verwendet wird (siehe hierzu auch [15.3](#)). Bei nicht hervorgehobenen IML-Teilgraphen wird dieses Kästchen nicht angezeigt.

14.3.5.2. Popup-Menü Subgraph List

Bei einem Rechtsklick auf Subgraph List öffnet sich ein Popup-Menü mit folgenden Menüpunkten:

1. Highlight (Menü)

- a) Color 1
- b) Color 2
- c) Color 3
- 2. Unhighlight In All Windows
- 3. Copy IML Subgraph
- 4. Insert IML Subgraph
- 5. Delete IML Subgraph
- 6. Create Window Selection
- 7. IML Subgraph Set Operation

14.4. Anzeigefenster

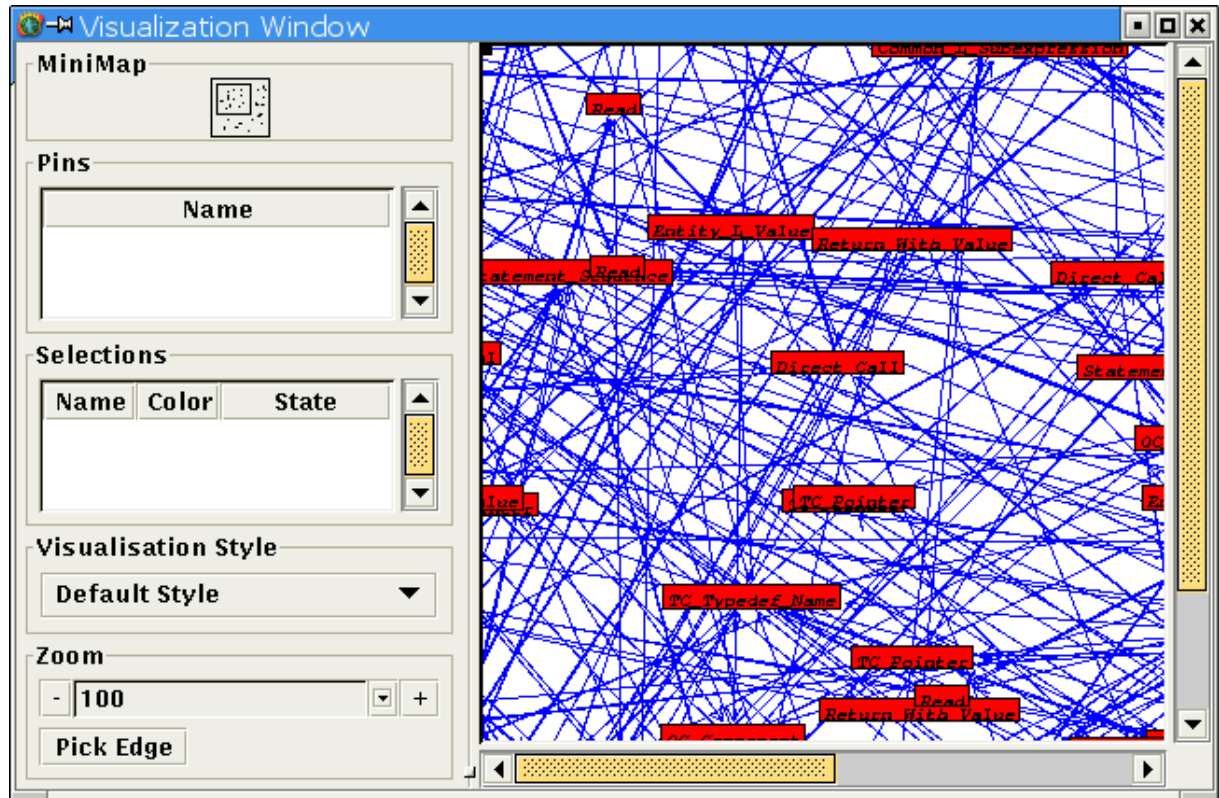


Abbildung 14.2.: Graph-Window

Im Programm kann es mehrere Anzeigefenster (siehe Abbildung 14.2) geben.

In Anzeigefenstern (Visualization Windows) werden Teile des IML-Graphen visualisiert. Sie erscheinen im großen quadratischen rechten Teil („Vis Pane“) des Fensters. Die Vis Pane enthält den Anzeigehalt des Fensters. Im linken Viertel des Fensters, der Toolbar („Vis Toolbar“) befinden sich untereinander die MiniMap, Pinliste, Selektionsliste, die Stilauswahl-Combobox und die Zoom-Kontrolle.

14.4.1. MiniMap

In der MiniMap wird der Anzeigehalt verkleinert dargestellt, der Graph wird jedoch nur durch einen grauen Kasten repräsentiert, Knoten sind nicht sichtbar. Die MiniMap wird von einem Rahmen umschlossen. Der sichtbare Anzeigehalt wird durch einen kleineren Rahmen repräsentiert, der innerhalb der MiniMap durch Mausklicks versetzt werden kann. Je nach Zoomstufe ist dieser Rahmen größer oder kleiner. Wird der Rahmen versetzt, wird der sichtbare Anzeigehalt in der Vis Pane entsprechend angepasst (siehe auch UseCase 4.33).

14.4.2. Visualisierung der Knoten

In der Vis Pane werden die Fenster-Knoten und Fenster-Kanten angezeigt.

Wenn auf einem Fenster-Knoten mit der rechten Maustaste geklickt wird, öffnet sich folgendes Popup-Menü:

14.4.2.1. Node-Popup-Menü

1. Show Node Info Window
2. Show Corresponding Source Code
3. Move Node
4. Annotation (Menü)
 - a) Add
 - b) Change
 - c) Delete

Wenn in der Vis Pane mit der rechten Maustaste auf eine Stelle geklickt wird, an der kein Fenster-Knoten liegt, öffnet sich das folgende Popup-Menü. Die Einträge dieses Menüs sind auch im Node-Popup-Menü (siehe [14.4.2.1](#)) sichtbar.

1. Make Room
2. New Pin

14.4.3. Pins

In der Pinliste („Pin List“) können Pins festgelegt werden.

Im Popup-Menü der Pinliste befinden sich folgende Menüpunkte:

1. Focus Pin
2. Delete Pin

14.4.4. Selektionsauswahlliste

Die Selektionsauswahlliste hat die Spalten „Name“ mit dem Namen der Selektion, „Color“ mit der Farbe der Selektion und „State“ mit dem Status der Selektion (eingebledet oder ausgebledet).

Im Popup-Menü der Selektionsauswahlliste („Selection List“) befinden sich folgende Einträge:

1. New Selection
2. Move Selection
3. Show Selection

4. Show all
5. Hide Selection
6. Layout Selection
7. Create New IML Subgraph From This Selection
8. Highlight Selection (Menü)
 - a) Color 1
 - b) Color 2
 - c) Color 3
9. Unhighlight Selection
10. Zoom To Make Selection Fill Window
11. Copy Selection
12. Copy Selection Keeping Existing Layout
13. Copy Selection Changing Existing Layout
14. Delete Selection
15. Delete Nodes and Edges of Selection
16. Selection Set Operation

14.4.5. Stilauswahl-Combobox

In der Stilauswahl-Combobox („Style Chooser“) kann ein Visualisierungsstil (siehe auch [13.3](#)) ausgewählt werden.

14.4.6. Zoom-Kontrolle

Die Zoom-Kontrolle besteht aus folgenden Elementen:

1. Button „-“
2. Combobox mit vorgefertigten Zoom-Werten und Eingabemöglichkeit
3. Button „+“
4. Button „Pick Edge“

In der ComboBox zur Zoom-Kontrolle befindet sich neben den vorgegebenen Zoomstufen noch der Eintrag „Fit in Window“.

14.4.7. Scrollbars

Mit den beiden Scrollbars am unteren und rechten Rand des Anzeigefensters kann der sichtbare Anzeigehalt gescrollt werden.

14.5. Knoten-Informationsfenster

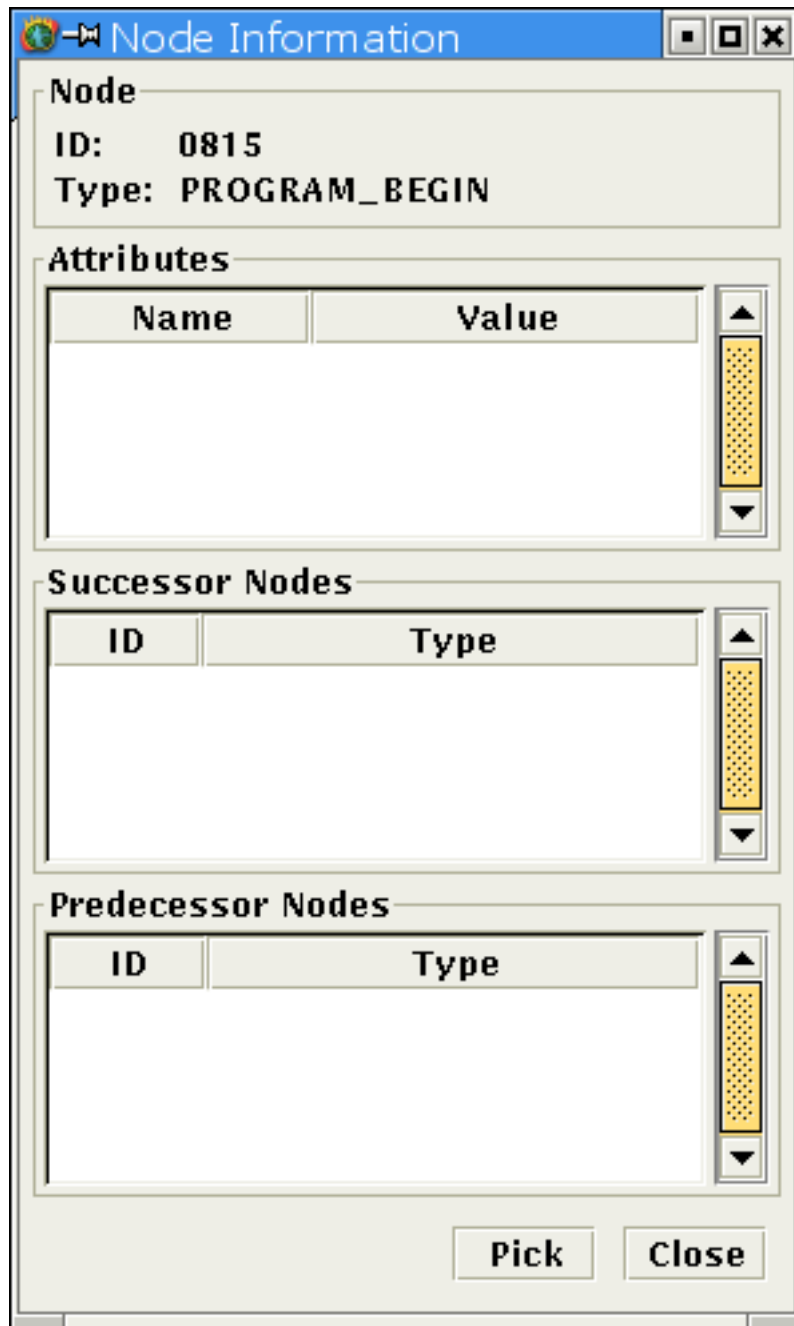


Abbildung 14.3.: Node-Info-Window

In Knoten-Informationsfenstern (siehe Abbildung 14.3) können nähere Informationen zu Fenster-Knoten angezeigt werden.

Im Knoten-Informationsfenster wird die ID und die Knotenklasse des vom Fenster-Knoten repräsentierten IML-Knotens dargestellt, darunter befinden sich untereinander die drei scrollbaren Listen

„Attributes“, „Successor Nodes“ und „Predecessor Nodes“.

Unter den drei Listen befinden sich nebeneinander zwei Buttons „Pick“ und „Close“. Pick dient zur Auswahl eines anderen Fenster-Knotens mittels des Fadenkreuzcursors. Die Informationen zu dem ausgewählten Fenster-Knoten werden dann im Knoten-Informationsfenster dargestellt (siehe hierzu auch UseCase [4.31](#)).

14.5.1. Liste Attributes

Die Liste Attributes enthält die Attribute des Knotens und hat zwei Spalten:

1. Name (der Name des Attributes)
2. Value (der Attributwert)

14.5.2. Liste Successor Nodes

Die Liste Successor Nodes hat eine Zeile für jede vom Knoten abgehende Kante mit zwei Spalten:

1. ID (die ID des Nachfolger-Knotens)
2. Type (die Kantenklasse der abgehenden Kante)

14.5.3. Liste Predecessor Nodes

Die Liste Predecessor Nodes hat eine Zeile für jede beim Knoten eingehende Kante mit zwei Spalten:

1. ID (die ID des Vorgänger-Knotens)
2. Type (die Kantenklasse der eingehenden Kante)

14.6. Skriptdialog

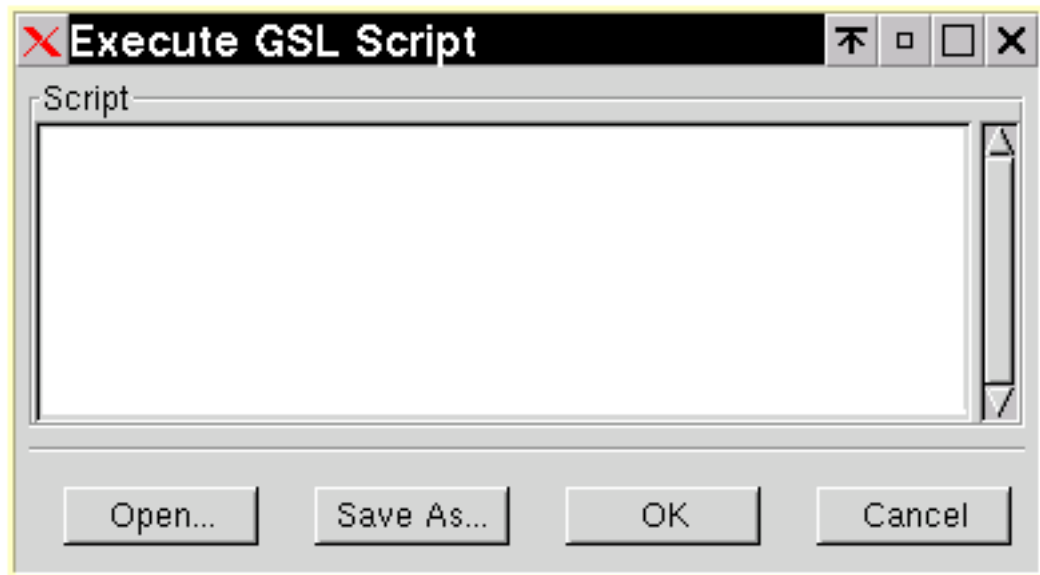


Abbildung 14.4.: Skript-Dialog

Der Skriptdialog „Query Dialog“ (siehe Abbildung [14.4](#)) hat zuoberst ein Textfeld „Query Text“, in das ein GSL Skript eingegeben werden kann.

Unter Query Text befinden sich folgende Buttons:

1. Open...
2. Save As...
3. OK
4. Cancel

14.7. Allgemeiner Texteingabedialog

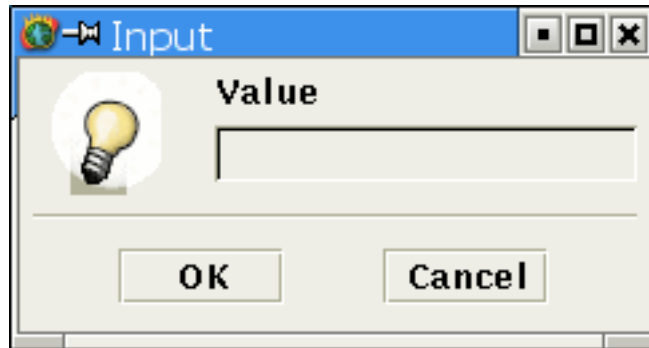


Abbildung 14.5.: Dialog-Window

Ein allgemeiner Texteingabedialog ist ein Fenster Dialog Window (siehe Abbildung 14.5) mit dem Titel „Input“. Im Fenster ist ein einzeliges Textfeld mit einem Prompt-Text, zwei Buttons mit den Beschriftungen „OK“ und „Cancel“.

14.8. Set-Operation-Dialog

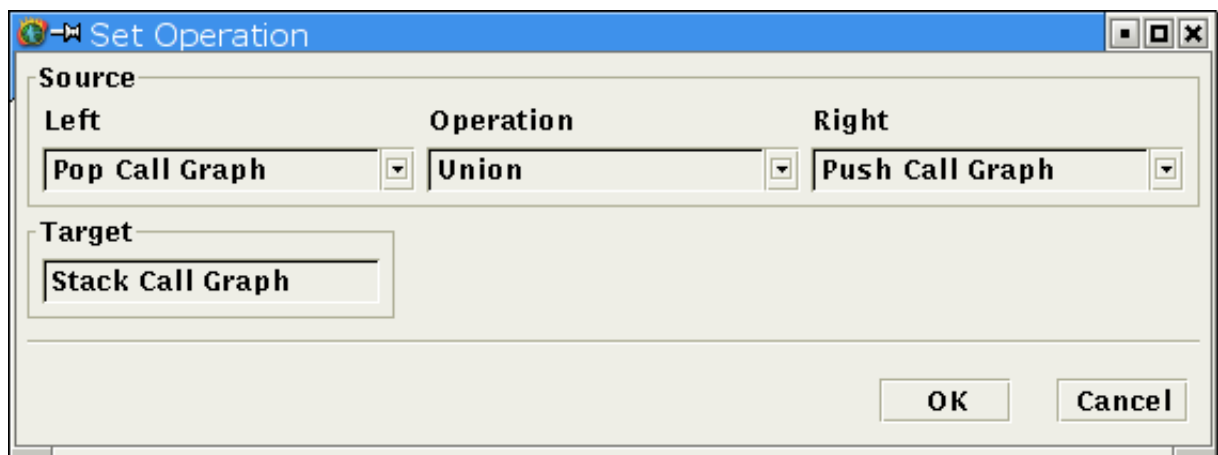


Abbildung 14.6.: Set-Operation-Dialog

Der Set-Operation-Dialog (siehe Abbildung 14.6) dient dazu, Mengenoperationen über Selektionen und IML-Teilgraphen durchzuführen. Die Mengenoperation lässt sich wie folgt beschreiben:

$\text{TARGET} := \text{LEFT_SOURCE} <\text{op}> \text{RIGHT_SOURCE}$

Bestandteile des Dialoges sind:

1. LEFT_SOURCE Combobox

Soll eine Mengenoperation für Selektionen ausgeführt werden, so kann hier eine der Selektionen des entsprechenden Anzeigefensters ausgewählt werden.

Bei einer Mengenoperation über IML-Teilgraphen, werden alle IML-Teilgraphen des Projektes angezeigt, von denen dann einer ausgewählt werden kann.

2. <op> Combobox

Mengenoperation, die ausgeführt werden soll; auswählbar sind: Union, Difference oder Intersection.

3. RIGHT_SOURCE Combobox

Soll eine Mengenoperation für Selektionen ausgeführt werden, so kann hier eine der Selektionen des entsprechenden Anzeigefensters ausgewählt werden.

Bei einer Mengenoperation über IML-Teilgraphen, werden alle IML-Teilgraphen des Projektes angezeigt, von denen dann einer ausgewählt werden kann..

4. TARGET Textfeld

Name der neuen Selektion oder des neuen IML-Teilgraphen als Ergebnis der Mengenoperation.

5. Buttons OK und Cancel

Mit den beiden Buttons OK und Cancel lässt sich die Operation starten bzw. der Dialog ohne Ausführen der Operation verlassen.

14.9. Layoutalgorithmen Dialog

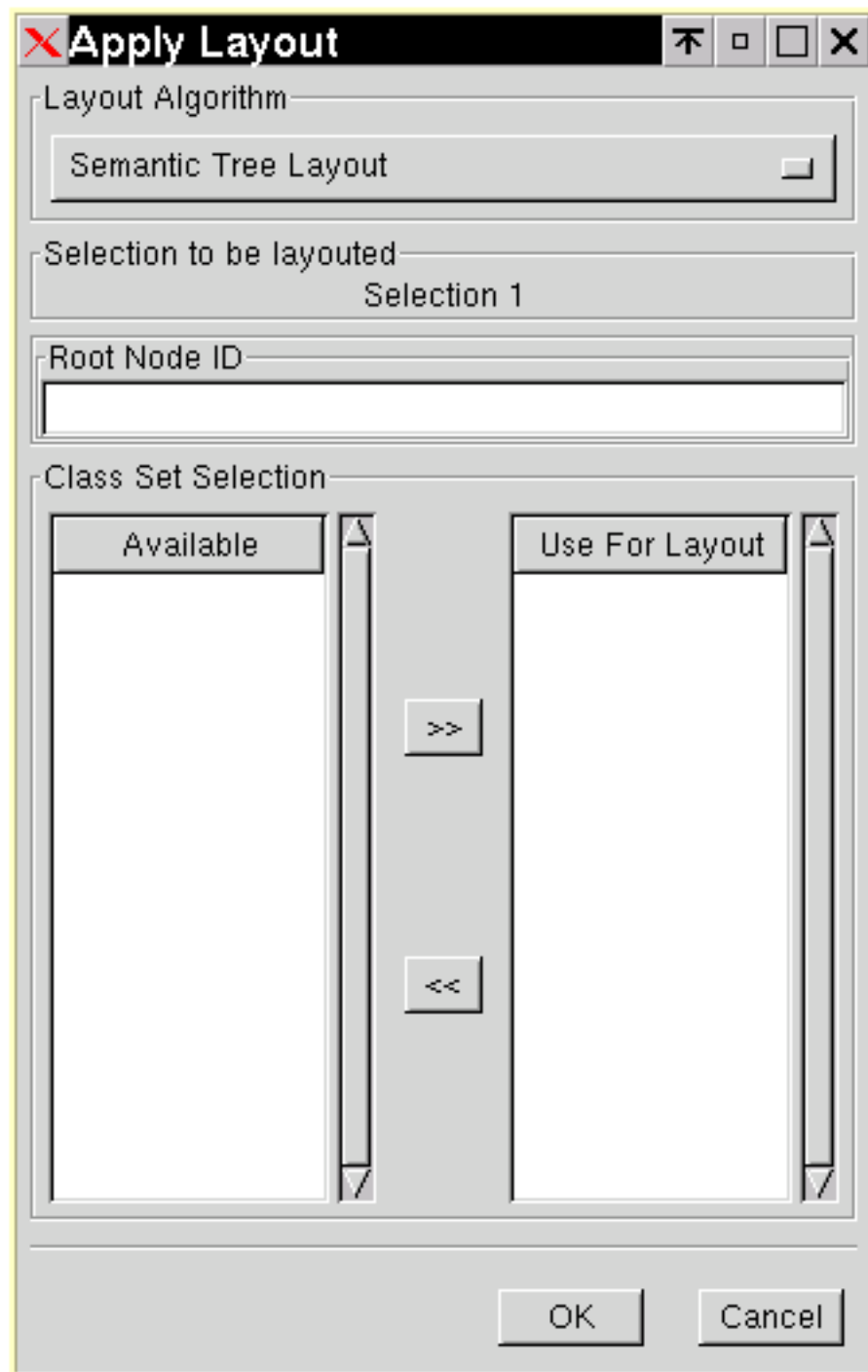


Abbildung 14.7.: Layout-Algorithm-Dialog

Der Layoutalgorithmen Dialog (siehe Abbildung 14.7) bietet folgende Möglichkeiten:

1. Auswahl des Layoutalgorithmus

2. Ggf. Anzeige der zu layoutenden Selektion
3. Ggf. Eingabefeld für die ID des Wurzelknotens bei Treelayouts
4. Eingabe der zu berücksichtigenden Klassenmengen bei semantischen Layouts, Hierbei können mittels der Buttons „»“ und „«“ die verfügbaren definierten Klassenmengen zwischen den beiden Spalten „Available“ und „Use For Layout“ hin und her geschoben werden. Nur Klassenmengen, die sich in der „Use For Layout“ Spalte befinden, werden für das Layout berücksichtigt.
5. OK Button
6. Cancel Button

14.10. Dateneingabe

Beschreibung von verschiedenen GUI Elementen zur Eingabe von Daten.

14.10.1. Knoten-Annotations-Dialog

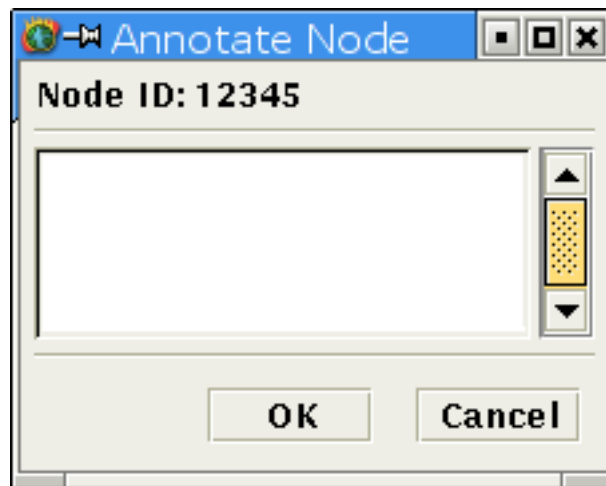


Abbildung 14.8.: Node-Annotation-Dialog

Der Knoten-Annotations-Dialog (siehe Abbildung 14.8) besteht aus einem Fenster mit einem Label, welches die ID des Knotens anzeigt, und einem Textfeld für den Annotationstext. Mittels der beiden Buttons OK und Cancel können die Änderungen im Textfeld übernommen oder verworfen werden.

14.10.2. Platz Schaffen-Dialog



Abbildung 14.9.: Make-Room-Dialog

Im Dialogfenster Make Room (siehe Abbildung 14.9) wird in einem Textfeld als Zahl eingegeben, um wie viel Pixel die Fenster-Knoten an einer vorher markierten Stelle auseinandergeschoben werden sollen. Mit dem Button OK kann bestätigt werden, mit dem Button Cancel abgebrochen werden.

14.11. Ausgabe von Fehlermeldungen

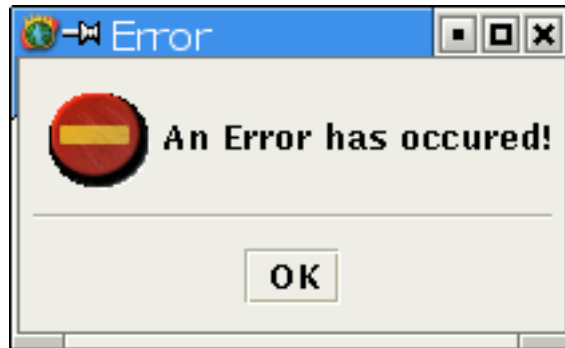


Abbildung 14.10.: Error-Window

Ein allgemeiner Fehlerdialog (siehe Abbildung 14.10) ist ein Fenster „Error Window“ mit dem Titel „Error“. Im Fenster befindet sich ein Label zur Ausgabe der Fehlermeldung und ein Button mit der standardmäßigen Beschriftung „OK“. Für weitere Informationen zum allgemeinen Fehlerverhalten von GIANT siehe auch Abschnitt 3.1.

14.12. Sicherheitsabfrage

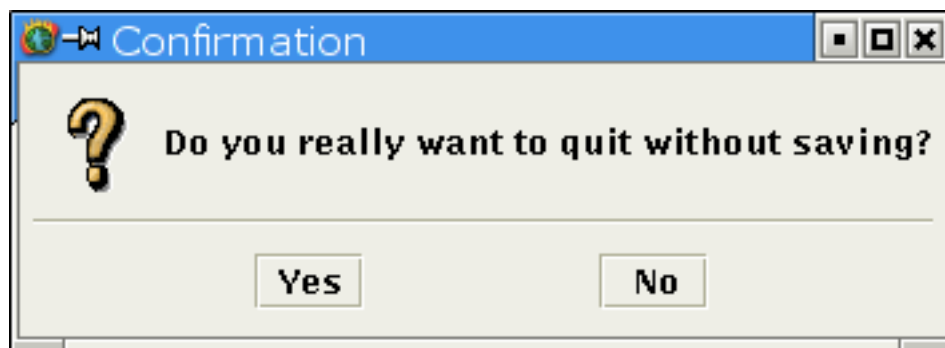


Abbildung 14.11.: Confirmation-Window

Eine allgemeine Sicherheitsabfrage (siehe Abbildung 14.11) ist ein Fenster („Confirmation Window“) mit dem Titel „Confirmation“. Im Fenster ist ein Fragetext und ein Button mit der standardmäßigen Beschriftung „Yes“, sowie ein Button mit der standardmäßigen Beschriftung „No“. In dem Fragetext steht jeweils kontextabhängig die Information, die der Benutzer zur Beantwortung der Sicherheitsabfrage benötigt.

14.13. Auswahl von Dateien

14.13.1. Der „Standard-Filechooser-Dialog“

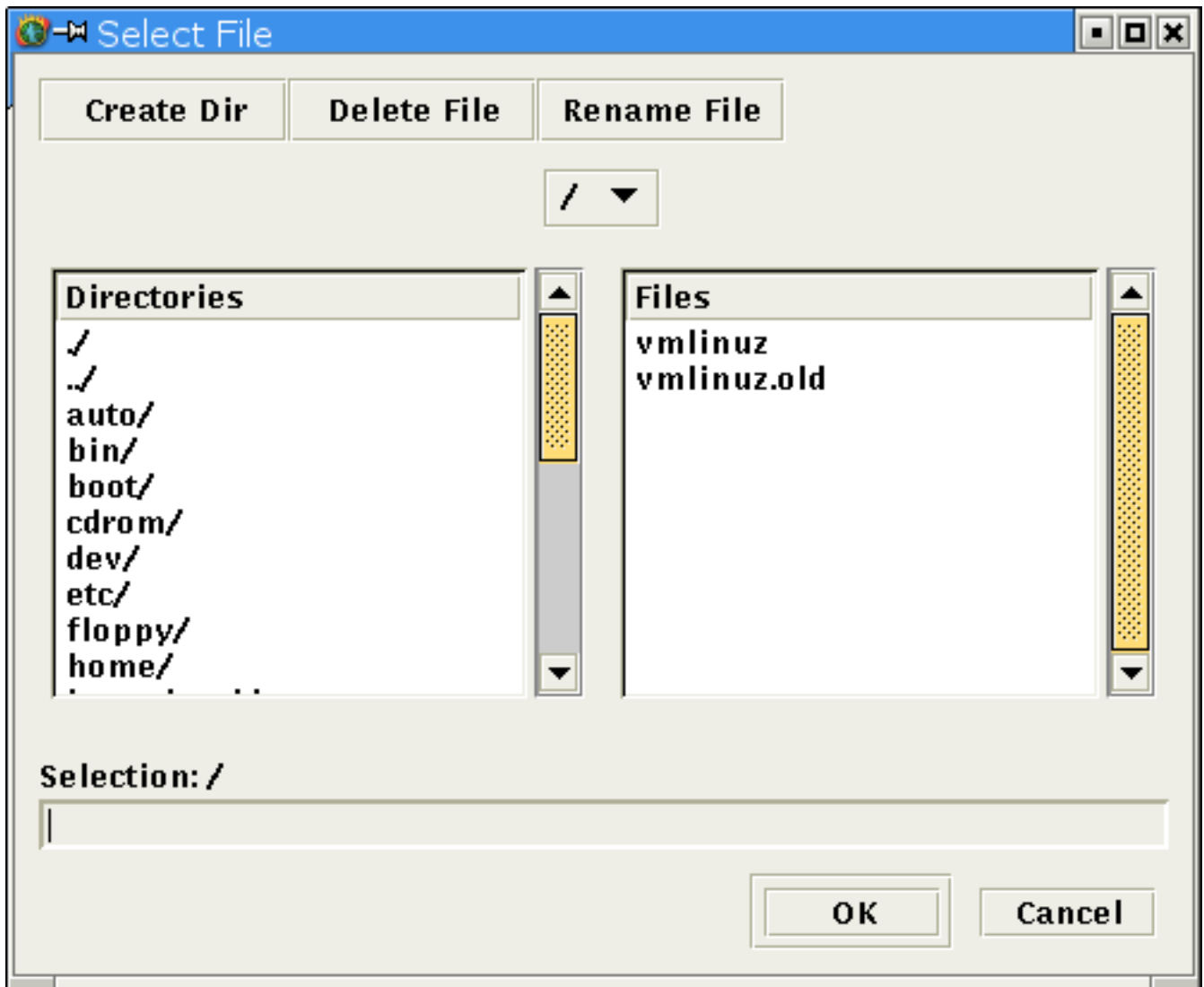


Abbildung 14.12.: Filechooser-Window

Die Auswahl von Dateien (Laden/Speichern) passiert bei GIANT mittels des Standard-Filechooser-Dialogs (siehe Abbildung 14.12). Dieser Dialog ist Bestandteil der Widget-Bibliothek von GTK.

14.14. Fadenkreuz-Cursor

Der Cursor verwandelt sich nach Auswahl bestimmter Funktionen (siehe z.B. UseCase 4.25) in ein Fadenkreuz. Durch Linksklick auf einen Fenster-Knoten oder eine Fenster-Kante in einem Anzeige-

fenster wird diese(r) für die jeweilige Funktion ausgewählt. Der Fadenkreuz-Cursor wird auch zur Vorgabe von Positionen, an denen dann z.B. neue Fenster-Knoten eingefügt werden, verwendet. Nach dem Linksklick verwandelt sich der Fadenkreuz-Cursor wieder in den Standard-Cursor. Durch einen Rechtsklick bei aktivem Fadenkreuz-Cursor lässt sich die aktuelle Funktion abbrechen.

14.15. Fortschrittsanzeige

Es gibt in GIANT zwei Möglichkeiten zur Fortschrittsanzeige während laufender Berechnungen. Jede Progressbar ist mit „The system is busy. Please be patient.“ beschriftet.

14.15.1. Progressbar-Runs

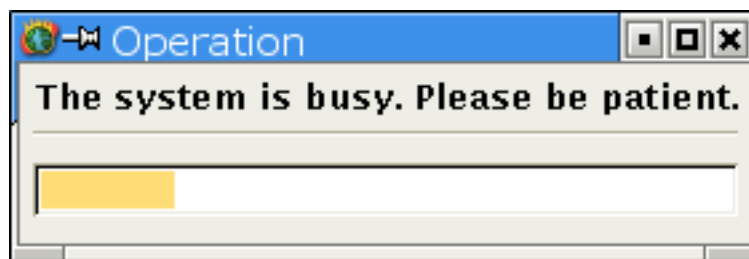


Abbildung 14.13.: Progressbar-Runs

Hierbei handelt es sich um ein Fenster mit einem sich bewegenden bzw. sich ständig änderndem Inhalt (z.B. ein Progress-Balken), so dass für den Benutzer erkennbar ist, dass das System noch arbeitet (beispielhafte Grafik siehe Abbildung 14.13).

14.15.2. Progressbar-Modal

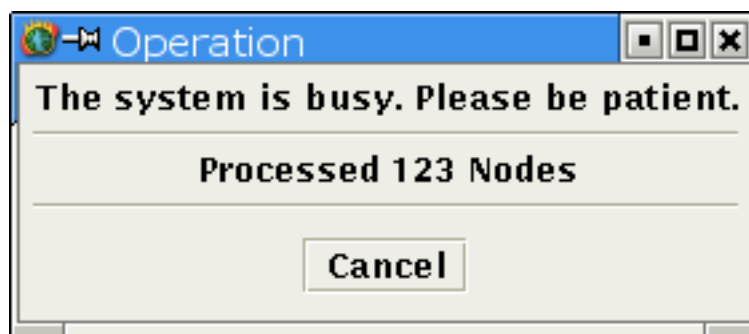


Abbildung 14.14.: Progressbar-Modal

Der „Progressbar-Modal“ (siehe Abbildung 14.14) ist ein Fenster, das über den aktuellen Fortschritt einer Berechnung informiert, wobei der Gesamtaufwand unbekannt sein kann. Hierzu wird kontinuierlich die Anzahl der schon bearbeiteten Datensätze textuell ausgegeben, und, sofern bekannt, die

Prozentzahl der bereits bearbeiteten Datensätze im Verhältnis zur gesamten Datenmenge. Während dieser Dialog sichtbar ist, ist der Rest der GUI von GIANT gesperrt. Im Fenster existiert ein Button „Cancel“, mit dem eine laufende Berechnung abgebrochen werden kann.

15. Visualisierung des IML-Graphen

Dieses Kapitel beschreibt, wie IML-Knoten und IML-Kanten des IML-Graphen innerhalb von Anzeigefenstern als Fenster-Knoten und Fenster-Kanten dargestellt werden. Spezifiziert wird hier das Ergebnis der Visualisierung, also das Aussehen der Fenster-Knoten und Fenster-Kanten, nicht aber die Art und Weise, wie GIANT diese Visualisierung erzeugt bzw. berechnet.

15.1. Visualisierung von Knoten

Hier wird die Visualisierung eines Fenster-Knoten innerhalb eines Anzeigefensters beschrieben. Die tatsächliche Darstellung hängt stark von der aktuellen Zoomstufe und dem gewählten Visualisierungsstil ab.

15.1.1. Knoten-Rechteck

1. Grundlage der Darstellung eines Fenster-Knotens ist immer ein Rechteck mit fester, automatisch berechneter Breite.
2. Die Höhe des Rechtecks hängt von der Anzahl der Attribute des IML-Knotens, die direkt im Anzeigefenster dargestellt werden, ab.
3. Jedes Knoten-Rechteck hat eine konfigurierbare Füllfarbe und einen Rahmen, dessen Farbe ebenfalls konfiguriert werden kann.

15.1.2. Knoten-Icon

Jeder Fenster-Knoten kann über ein vordefiniertes Icon verfügen, welches im linken oberen Eck des Knoten-Rechtecks dargestellt wird. Alle Fenster-Knoten für die kein Icon definiert wurde, haben ein Standard-Icon.

15.1.3. Icon für Knoten-Annotationen

Der Benutzer kann Fenster-Knoten annotieren. Jeder Fenster-Knoten, für den so eine Annotation verfasst wurde, zeigt dies über ein entsprechendes Icon im Knoten-Rechteck an. Für weitere Informationen zu Knoten-Annotationen siehe auch Abschnitt [12.2.4](#).

15.1.4. Klassenname und ID

Die ID eines IML-Knotens wird innerhalb des Knoten-Rechtecks rechts oben als Hexadezimalzahl angezeigt.

Der Name der Knotenklasse wird bei den Attributen angezeigt.

15.1.5. Attribute des Knoten

Innerhalb des Knoten-Rechtecks können auch Attribute dargestellt werden (Attributname und Wert), da das Knoten-Rechteck eine fixe Breite hat, wird der zugehörige Text abgeschnitten, falls er zu lang ist. Dieses Abschneiden wird dem Benutzer durch das Anfügen von „...“ an das Ende des sichtbaren Teils des Textes angezeigt.

15.2. Visualisierung von Kanten

1. Fenster-Kanten sind immer gerade Linien von einem Start- zu einem Zielknoten.
2. Fenster-Kanten können sich hinsichtlich ihrer Linienfarbe und der Art der Linie (z.B. gestrichelt oder durchgezogen) unterscheiden.
3. Selbstkanten werden als kreisförmige Schleifen dargestellt.

15.3. Hervorheben von Fenster-Knoten und Fenster-Kanten

Hier ist spezifiziert, wie Fenster-Knoten und Fenster-Kanten hervorgehoben werden. Die genaue Art der Hervorhebung kann sich aber u.U. ändern, wenn die Implementierung zeigen sollte, dass ein anderes Vorgehen sinnvoller ist.

Innerhalb eines Anzeigefensters können

1. die „aktuelle Selektion“,
2. bis zu drei weitere Selektionen,
3. und bis zu drei IML-Teilgraphen

unterscheidbar hervorgehoben werden.

15.3.1. Hervorheben von Selektionen und der aktuellen Selektion

Beim Hervorheben der Fenster-Knoten und Fenster-Kanten von Selektionen werden diese mittels einer konfigurierbaren Farbe eingefärbt.

1. Jeder Fenster-Knoten einer Selektion wird durch entsprechendes Einfärben des Knoten-Rechtecks hervorgehoben.

2. Gehört ein Fenster-Knoten zu mehreren Selektionen, die hervorgehoben sind, so wird das Knoten-Rechteck entsprechend mehrfarbig eingefärbt.
3. Jede Fenster-Kante wird dadurch hervorgehoben, dass ihre Linie entsprechend dicker gezeichnet wird, wobei der Teil, um den die Fenster-Kante verbreitert wird jeweils entsprechend eingefärbt wird.
4. Gehört eine Fenster-Kante zu mehreren Selektionen, die hervorgehoben werden, so wird sie entsprechend weiter verbreitert und mehrfarbig eingefärbt.

15.3.2. Hervorheben von IML-Teilgraphen

1. Jeder Fenster-Knoten eines IML-Teilgraphen wird dadurch hervorgehoben, dass er mit einem weiteren rechteckigen Rahmen versehen wird, der entsprechend eingefärbt ist.
2. Gehört ein Fenster-Knoten zu mehreren hervorgehobenen IML-Teilgraphen kommen entsprechend weitere Rahmen dazu.
3. Jede Fenster-Kante wird dadurch hervorgehoben, dass ihre Linie entsprechend dicker gezeichnet wird, wobei die der Teil, um den die Fenster-Kante verbreitert wird, entsprechend eingefärbt wird.
4. Gehört eine Fenster-Kante zu mehreren IML-Teilgraphen, die hervorgehoben werden, so wird sie entsprechend weiter verbreitert und mehrfarbig eingefärbt.

15.4. Detailstufen beim Zoomen

1. Beim Herauszoomen müssen Details abstrahiert werden. Dies geschieht bei GIANT in mehreren Stufen.
2. Eine exakte Spezifikation der Detailstufen, insbesondere ab welcher Zoomstufe welche Detailstufe gewählt wird, ist an dieser Stelle nicht sinnvoll. Optimale Parameter hierfür werden im Laufe der Implementierung ermittelt.
3. Beim Herauszoomen wird GIANT zuerst versuchen, die Größe der Knoten-Rechtecke durch Verringerung der Schriftgröße zu verringern. Erst wenn die Schrift klein genug ist (aber noch gut lesbar) wird GIANT Elemente der Darstellung der Fenster-Knoten entfernen.

Es wird in etwa die folgenden Detailstufen geben.

15.4.1. Maximale Detailstufe

Bei maximaler bis hoher Zoomstufe (nahe ans Bild herangezoomt).

Alle oben beschriebenen Bestandteile von Fenster-Knoten und Fenster-Kanten werden angezeigt.

Alle Beschriftungen in maximaler Schriftgröße.

15.4.2. Sehr hohe Detailstufe

Nur die Schriftgröße für Knoten- und Kantenbeschriftungen wird verringert.

15.4.3. Hohe Detailstufe

Alle oben beschriebenen Bestandteile von Fenster-Kanten werden angezeigt.

Bei Fenster-Knoten werden nur noch das Icon, der Klassenname und die ID im Knoten-Rechteck angezeigt, die anderen Attribute aber nicht mehr.

15.4.4. Durchschnittliche Detailstufe

Die Beschriftungen von Fenster-Kanten werden nicht mehr angezeigt.

Bei Fenster-Knoten enthält das Knoten-Rechteck nur noch das Icon.

15.4.5. Geringe Detailstufe

Bei sehr geringer Zoomstufe.

Bei Fenster-Knoten wird nur noch das reine Knoten-Rechteck mit Rahmen und Füllfarbe angezeigt (kein Icon mehr).

15.4.6. Minimale Detailstufe

Bei sehr geringer bis hin zur minimalen Zoomstufe.

Die Knoten-Rechtecke werden immer kleiner gezeichnet, bis sie ihre minimale Größe erreicht haben.

15.5. Minimap

Zu jedem Anzeigefenster gibt es genau eine Minimap, welche einen Bereich des Anzeigeeinhaltes als Rechteck abstrahiert darstellt. Dargestellt wird der Bereich des Anzeigeeinhaltes, der einem alle Fenster-Knoten umspannendem Rechteck entspricht.

Die Minimap bietet die folgende Funktionalität:

1. Der aktuell sichtbare Anzeigeeinhalt wird auf der Minimap mit seiner relativen Position und Größe im Verhältnis zum abstrahierten Anzeigeeinhalt dargestellt.
2. Über die Minimap kann auch gescrollt werden. Klickt der Benutzer auf einen Punkt innerhalb der Minimap, so wird der sichtbare Anzeigeeinhalt automatisch auf den diesem Punkt entsprechenden Bereich des Anzeigeeinhaltes gesetzt.

16. Nichtfunktionale Anforderungen

Hier werden die wesentlichen nichtfunktionalen Anforderungen an GIANT spezifiziert. Insbesondere sind dies Anforderungen aus dem Bereich des Software-Engineering, wie z.B. Wartbarkeit und Erweiterbarkeit.

16.1. Plattformunabhängigkeit

Das Produkt soll ein hohes Maß an Plattformunabhängigkeit, wie sie sich aus der eingesetzten Entwicklungsumgebung ergibt, verfügen (also alle Systeme auf denen GTK/ADA 1.2.12 lauffähig ist). Explizit während der Entwicklung getestet und damit garantiert werden kann dies aber nur für Sun Solaris und Linux.

16.2. Wartbarkeit

Das System soll sich durch ein hohes Maß an Wartbarkeit und Erweiterbarkeit auszeichnen. Erreicht werden soll dies durch die folgenden Grundsätze.

1. Wartbarkeit und Erweiterbarkeit gehen über Performanz.
2. Strukturierung des Entwurfes nach Kriterien des Software-Engineerings.
3. Strikte Trennung von GUI und Funktionalität.
4. Einsatz von XML für alle editierbaren Konfigurationsdateien.
5. Konfigurierbarkeit der Menüeinträge über eine XML Datei. Vorbehaltlich der Machbarkeit regelt der Entwurf hierzu näheres.

16.3. Portabilität

1. Generell soll GIANT Betriebssystem unabhängig entworfen und Implementiert werden.
2. Weitere Maßnahmen zur expliziten Unterstützung der Portabilität sind nicht vorgesehen.

16.4. Mengengerüst

1. Ein explizites Mengengerüst, z.B. hinsichtlich der minimalen und maximalen Anzahl von darstellbaren Fenster-Knoten, gibt es nicht.
2. Größenbeschränkungen (Größe der zu ladenden IML-Graph-Datei, Anzahl Fenster-Knoten in einem Anzeigefenster ...) sollen sich alleine aus dem verfügbaren Arbeitsspeicher ergeben.
3. Falls der verfügbare Arbeitsspeicher erschöpft ist, soll das System nicht abstürzen, sondern vielmehr die entsprechende Aktion kontrolliert abbrechen und dem Benutzer ein Weiterarbeiten ermöglichen.

16.5. Robustheit

1. GIANT soll möglichst auf alle Fehler mit einer qualifizierenden Fehlermeldung reagieren und nicht unkontrolliert abstürzen.
2. Wo immer möglich soll dem Benutzer nach Auftreten eines Fehlers ein sinnvolles Weiterarbeiten ermöglicht werden.
3. Insbesondere bei Fehlern, die mit dem Überschreiten der verfügbaren Hauptspeicherkapazität in Zusammenhang stehen, kann solch ein robustes Verhalten nicht garantiert werden.
4. Speicherlecks können im Falle Abbruchs einer Aktion im Fehlerfalle nicht ausgeschlossen werden.

16.6. Robustheit gegenüber Änderungen der IML-Graph-Spezifikation

Das System soll so an Bauhaus angebunden werden, dass Änderungen in der Spezifikation des IML-Graphen möglichst keine Wartungsarbeiten erfordern. Besonders soll dies für die verschiedenen Attribute, Knoten- und Kantenklassen des Bauhaus-IML-Graphen gelten, da diese sich oft ändern können. Die Klassen und Attribute der Bauhaus-IML-Graph Bibliothek werden durch den IML-Browser nur so unterschieden, dass neu hinzukommende Attribute automatisch mit erfasst und damit angezeigt werden können.

16.7. Leistungsanforderungen

1. Bei GIANT handelt es sich um ein striktes Einbenutzersystem.
2. Das Systemverhalten für den Fall, dass zwei Benutzer mittels verschiedener laufender Instanzen von GIANT (als Prozess im Betriebssystem) gleichzeitig auf denselben Daten (Projekte und entsprechende Verwaltungsdateien) arbeiten, ist undefiniert.

16.8. Antwortverhalten

1. Bei Anfragen und Layoutalgorithmen kann keine maximale Rechenzeit garantiert werden.
2. Der Benutzer hat aber zu jeder Zeit die Möglichkeit, entsprechende Aktionen abubrechen.

16.9. Sicherheit

1. Ein automatisches Schreiben von Änderungen etc. in die Projektdateien ist nicht vorgesehen, der Benutzer muss dies explizit veranlassen.
Bevor Informationen verloren gehen (z.B. beim Beenden von GIANT) erscheint aber eine Sicherheitsabfrage.
2. Muss GIANT auf Grund eines Fehlers beendet werden, so gehen alle bis dahin nicht persistent für das Projekt gespeicherten Informationen unwiederbringlich verloren.

16.10. Erweiterbarkeit

Eine spätere Erweiterbarkeit von GIANT ist besonders hinsichtlich der im Folgenden beschriebenen Funktionalität vorgesehen. Dies wird im Rahmen des Entwurfes berücksichtigt.

1. Erweiterung um neue Layoutalgorithmen.
2. Erweiterung zur graphisch unterstützten Eingabe von GSL-Skripten.
3. Erweiterung um Kantenknickpunkte.

17. Technische Produktumgebung

Hier wird die zum Betrieb von GIANT nötige Hardware und die nötige Software spezifiziert.

17.1. Software

Das Programm stellt an die installierte Software folgende Anforderungen:

- Sun Solaris oder Linux Betriebssystem
- Bauhaus Tools
- Emacs oder vi Texteditor

17.2. Hardware

Das Programm läuft auf SPARC Workstations und x86 kompatiblen PCs. Im Folgenden sind die minimalen Hardwareanforderungen zur Arbeit mit kleinen und mittleren Projekten beschrieben. Bei großen Projekten ist ein Speicherausbau von 2 GB und mehr empfehlenswert.

17.2.1. Hardwareanforderungen SPARC

- UltraSPARC-II 300 MHz
- 512 MB Hauptspeicher
- 8 Bit Grafik mit einer min. Auflösung von 1024*786
- Maus mit mindestens zwei Tasten

17.2.2. Hardwareanforderungen x86

- Pentium III 600 MHz
- 512 MB Hauptspeicher
- 8 Bit Grafik mit einer min. Auflösung von 1024*786
- Maus mit mindestens zwei Tasten

17.3. Produkt-Schnittstellen

Das Programm soll in die Bauhaus Suite integriert werden können. Als Schnittstelle dient dabei die Bauhaus IML Bibliothek. Weiterhin kann die Verwendung von Kommandozeilenparametern und der GSL zur Integration in das vorhandene System genutzt werden.

18. Anforderungen an die Umgebung

Hier werden die wesentlichen Anforderungen an die Entwicklungsumgebung von GIANT spezifiziert. Dies sind Werkzeuge, Programmiersprachen und Bibliotheken, die bei der Entwicklung eingesetzt werden müssen, und Standards, die beachtet werden sollen. Spezifiziert wird hier auch die Sprache aller zum Lieferumfang von GIANT gehörender Dokumente.

18.1. Compiler und Bibliotheken

Das System soll in der Sprache Ada95 mit GNAT 3.14p entwickelt werden, wobei GTK/ADA 1.2.12 als graphische GUI-Bibliothek eingesetzt werden soll. Das System baut auf der vom Kunden bereit gestellten IML-Graph-Bibliothek auf und soll des weiteren zur Unterstützung der Wartbarkeit möglichst auch die vom Kunden zur Verfügung gestellten Datenstrukturen (wie z.B. Hashtables aus Bauhaus/reuse/src) nutzen.

18.1.1. Lizenzrechtliches zu den Paketen des Kunden

Folgendes gilt nicht für die IML-Graph-Bibliothek, sondern nur für die Datenstrukturen aus der „Bauhaus Reuse Bibliothek“.

Die vom Kunden zur Verfügung gestellten Datenstrukturen werden den Entwicklern von GIANT ohne lizenzrechtliche Bedingungen überlassen. Die Nutzungsrechte der Entwickler am Produkt GIANT werden durch Einsatz dieser Datenstrukturen in keinsten Weise berührt.

18.2. Einlesen und Schreiben von XML Dateien

Auf XML-Dateien soll mittels des DOM (Document Object Model) Parsers aus XML/Ada 0.7.1 zugegriffen werden. XML/Ada 0.7.1 unterliegt lizenzrechtlich der „GNAT Modified GNU Public License“ (GMGPL).

Als Alternative ist der XML-Parser aus GTK/Ada vorgesehen – Paket Glib.XML.

18.3. Sprache

Hier wird beschrieben, in welcher Sprache die einzelnen Dokumente des Systems GIANT verfasst werden.

18.3.1. Spezifikation

Die Spezifikation – dieses Dokument – ist in deutscher Sprache verfasst.

18.3.2. Benutzerhandbuch

Das Benutzerhandbuch wird in deutscher Sprache verfasst.

18.3.3. Entwurf

Der Entwurf wird in Englisch verfasst.

18.3.4. Interne Dokumentation

Die interne Dokumentation von GIANT – Kommentare im Quellcode – erfolgt in englischer Sprache.

18.3.5. Interaktion mit dem Benutzer

Die GUI von GIANT interagiert mit dem Benutzer ausschließlich in englischer Sprache.

18.3.6. Konfigurationsdateien

Die Knoten und Attribute der XML-Konfigurationsdateien und der zugehörigen DTDs (Dokumenttyp-Definitionen) werden mit englischen Bezeichnern benannt.

19. Begriffslexikon

Das Begriffslexikon nennt Begriffe aus der „realen Welt“ und definiert ihre besondere Bedeutung innerhalb dieser Spezifikation und darauf aufbauender Dokumente. Des weiteren wird hier die englische Übersetzung dieser Begriffe für alle englischsprachigen Dokumente von GIANT vorgegeben.

- **Anfrage** (Query)
Eine Anfrage beschreibt einen Vorgang, bei dem über geeignete Kriterien IML-Knoten und IML-Kanten aus dem IML-Graphen oder aus IML-Teilgraphen ausgewählt werden.
- **Anzeigefenster** (Visualization Window)
Ein Fenster in dem ein Teilgraph des IML-Graphen nach bestimmten Kriterien visualisiert ist. Jedem Anzeigefenster ist ein Anzeigeinhalt zugeordnet.
- **Anzeigeinhalt** (Window Content)
Eine „virtuelle“ Oberfläche auf der die Objekte des visualisierten Teilgraphen (also Fenster-Knoten und Fenster-Kanten) angeordnet sind, d.h. räumliche Layoutinformation zu allen Objekten des entsprechenden Anzeigefensters. Abhängig von der Zoomstufe ist jeweils nur ein bestimmter Teil des Anzeigeinhaltes sichtbar - der sichtbare Anzeigeinhalt. Die Größe des Anzeigeinhaltes ist theoretisch unbegrenzt.
- **Fenster-Kante** (Window Edge)
Die graphische Repräsentation einer IML-Kante innerhalb eines Anzeigefensters.
- **Fenster-Knoten** (Window Node)
Die graphische Repräsentation eines IML-Knoten innerhalb eines Anzeigefensters.
- **Graph-Kante** (Graph Edge)
Eine Kante des IML-Graphen welche Bestandteil eines IML-Teilgraphen ist.
- **Graph-Knoten** (Graph Node)
Ein Knoten des IML-Graphen welcher Bestandteil eines IML-Teilgraphen ist.
- **IML-Graph** (IML Graph)
Der IML-Graph, wie er von der Bauhaus Reengineering GmbH gestellt wird. Auf diesen Graphen wird über das sogenannte Reflection Model zugegriffen.
- **IML-Kante** (IML Edge)
Eine Kante des IML-Graphen.
- **IML-Knoten** (IML Node)
Ein Knoten des IML-Graphen.
- **IML-Teilgraph** (IML Subgraph)
Eine Menge über Knoten und Kanten des IML-Graphen, die so gestaltet ist, dass sie einen

Teilgraphen des IML-Graphen darstellt.

- **Kantenklasse** (Edge Class)
Die Einteilung der IML-Kanten des IML-Graphen in verschiedene Klassen, wie sie sich aus der IML-Graph-Bibliothek von Bauhaus ergibt.
Innerhalb dieser Spezifikation ist der Begriff Kantenklasse so zu verstehen, dass jede IML-Kante eindeutig zu genau einer Kantenklasse gehört, eine Vererbungshierarchie existiert nicht. Die Zuordnung einer IML-Kante zu einer Kantenklasse wird durch die Knotenklasse des Start-Knotens und den Namen des Attributes (aus dem Bauhaus-IML-Graphen), welches die IML-Kante beschreibt, festgelegt. Jede vorkommende Kombination aus der Knoten-Klasse eines Start-Knotens und dem Namen eines Attributes, welches eine Kante beschreibt, ist somit eine eigene Kantenklasse
- **Klassenmenge** (Class Set)
Eine durch die IML-Graph Bibliothek vorgegebene Zusammenfassung von Kantenklassen und Knotenklassen. Es kann mehrere Klassenmengen geben. Die selben Knotenklassen und Kantenklassen können gleichzeitig zu mehreren Klassenmengen gehören.
- **Knoten-Annotationen** (Node Annotation)
Eine textuelle Beschreibung zu einem bestimmten Knoten des IML-Graphen.
- **Knotenklasse** (Node Class)
Die Einteilung der IML-Knoten des IML-Graphen in verschiedene Klassen, wie sie sich aus der IML-Graph-Bibliothek von Bauhaus ergibt.
Im Sinne der Verwendung dieses Begriffes innerhalb dieser Spezifikation liegt den Knotenklassen keine Vererbungshierarchie zu Grunde, jeder IML-Knoten gehört also eindeutig zu genau einer Knotenklasse
- **Layout** (Layout)
Die zweidimensionale räumliche Anordnung von Fenster-Knoten und Fenster-Kanten innerhalb eines Anzeigefensters auf dem sogenannten Anzeigeeinhalt.
- **Reflektion** (Reflection Model)
Die Schnittstelle zum Zugriff auf den IML-Graphen.
- **Schleife** (Loop)
Eine Kante mit identischem Start- und Zielknoten. Wird oft auch als Selbstkante bezeichnet.
- **Selektion** (Selection)
Eine Auswahl von Fenster-Knoten und Fenster-Kanten eines visualisierten Teilgraphen des IML-Graphen innerhalb eines Anzeigefensters.
- **Sichtbarer Anzeigeeinhalt** (Visible Window Content)
Der Bereich des Anzeigeeinhaltes eines Anzeigefensters, der zur Zeit sichtbar dargestellt wird.
- **Zoomstufe** (Zoom Level)
Dieser Faktor beschreibt die Größe des sichtbaren Anzeigeeinhaltes. Bei einer sehr niedrigen Zoomstufe (auch: weit weg gezoomt) ist ein größerer Teil des im Anzeigefenster visualisierten IML-Graphen sichtbar als bei einer hohen Zoomstufe (auch: sehr nach heran gezoomt).
- **hervorheben** (to highlight)
Hervorheben bedeutet, dass in einem Anzeigefenster visualisierte Fenster-Knoten oder Fenster-

Kanten z.B. durch eine farbige Umrahmung von anderen Fenster-Knoten oder Fenster-Kanten unterscheidbar gemacht werden.

- **selektieren** (to select)

Selektieren beschreibt einen Vorgang über den der Benutzer, z.B. durch Anklicken von Fenster-Knoten oder Fenster-Kanten mit der Maus, eine Selektion aufbaut.

20. Index

- Aktor, [23](#)
- Aktuelle Selektion, [18](#)
- Anforderungen
 - an die Hardware, [147](#)
 - an die Software, [147](#)
- Anfragen
 - aus Datei laden, [99](#)
 - in eine Datei speichern, [101](#)
- Anfragesprache, [12](#)
- Antwortverhalten, [145](#)
- Anzeigefenster, [12](#), [122](#)
 - öffnen, [34](#)
 - erzeugen, [33](#)
 - löschen, [39](#)
 - Pins, [12](#)
 - schließen, [37](#)
 - Scrollen, [47](#)
 - speichern, [36](#)
 - umbenennen, [35](#)
 - zoomen, [48](#)
 - zulässige Namen, [17](#)
- auseinanderschieben, [21](#)
- Bauhaus-Reengineering GmbH, [9](#)
- Bedienkonzepte, [17](#)
- Beenden von GIANT, [25](#)
- Benutzer, [23](#)
- Benutzeroberfläche, [117](#)
- Bibliotheken, [149](#)
- Compiler, [149](#)
- Datensicherheit, [145](#)
- Detailstufen, [141](#)
- Drei-Stufen-Konzept, [13](#)
- Editoren
 - Anzeige des Quellcodes, [114](#)
- Eingaben
 - zulässige, [16](#)
- Einlesen von Daten, [16](#)
- Erweiterbarkeit des Produktes, [145](#)
- Feedbackverhalten, [16](#)
- Fehlermeldungen, [15](#)
- Fehlerverhalten, [15](#)
 - Eingabefehler in Dialogen, [15](#)
 - Einlesen von Dateien, [16](#)
 - Kommandozeilenaufruf, [15](#)
- Fenster-Kanten
 - entfernen, [20](#)
 - hervorheben, [140](#)
 - löschen, [44](#)
 - Visualisierung, [140](#)
- Fenster-Knoten
 - auseinanderschieben, [21](#), [55](#)
 - entfernen, [20](#)
 - hervorheben, [140](#)
 - löschen, [44](#)
 - Visualisierung, [139](#)
- GIANT, [9](#)
- GNAT, [149](#)
- Graph-Kanten, [11](#)
- Graph-Knoten, [11](#)
- GSL, [12](#)
 - Beschreibung, [103](#)
 - Makros, [113](#)
 - Skriptdateien, [116](#)
- GTK/ADA, [149](#)
- Hardwareanforderungen, [147](#)
- Hauptfenster, [119](#)
- hervorheben, [11](#)
 - Farben konfigurieren, [113](#)
- hervorheben von Knoten und Kanten, [140](#)
- IML-Browser, [9](#)
- IML-Graph Bibliothek, [148](#)
- IML-Graph Datei
 - Identifikation, [109](#)
- IML-Graph-Bibliothek

- robuste Anbindung, 144
- IML-Teilgraph
 - aus Selektion erzeugen, 92
 - hervorheben, 89
 - Hervorhebung aufheben, 91
 - kopieren, 93
 - löschen, 94
 - Mengenoperationen, 95
- IML-Teilgraphen, 11
 - Ableiten aus Selektionen, 20
 - Einfügen in Anzeigefenster, 19, 20
 - hervorheben, 141
 - in Anzeigefenster einfügen, 40
 - zulässige Namen, 17
- Interaktionssprache, 150
- Kantenknickpunkte, 145
- Knoten-Annotationen, 12, 110, 111, 133
 - Icon, 139
- Kommandozeile, 23
- Kommandozeilenparameter, 23
- Konfiguration, 113
 - Visualisierungsstile, 114
- Konfigurationsdatei, 113
 - benutzerdefinierte, 113
 - globale, 113
- Layoutalgorithmen, 13, 105
 - Matrixlayout, 106
 - Treelayout, 105
- Leistungsanforderungen, 144
- Makros, 113
- Mengengerüst, 144
- Minimap, 122, 142
- Pakete der Bauhaus-Reengineering GmbH, 149
- Persistenz, 11, 107
 - Anzeigefenster, 110
 - IML-Teilgraphen, 111
 - Knoten-Annotationen, 111
 - Visualisierungsstile, 110
- Pins, 12, 123
 - anlegen, 57
 - anspringen, 58
 - löschen, 59
 - zulässige Namen, 17
- Plattformunabhängigkeit, 143
- Portabilität, 143
- Progressbar, 16, 136
- Projektdatei
 - Referenzen, 109
- Projekte, 11, 107
 - öffnen, 28
 - manuelle Modifikation, 107
 - neues Projekt erstellen, 26
 - Projektdatei, 108
 - Projektverzeichnis, 108
 - speichern, 30
 - Speichern von Projekten, 110
 - unter neuem Namen speichern, 31
 - Verwaltungsdateien für Anzeigefenster, 109
 - Verwaltungsdateien für IML-Teilgraphen, 109
 - Verwaltungsdateien für Knoten-Annotationen, 110
 - zulässige Namen, 17
- Rahmen, 18
- Robustheit, 144
- Schleifen, 140
- Schnittstellen, 148
- Scrollen, 47
 - via MiniMap, 142
- Selbstkanten, 140
- Selektieren, 17
- Selektionen, 11
 - Ableiten aus IML-Teilgraphen, 20
 - Einfügen in Anzeigefenster, 19
 - Entfernen von Knoten und Kanten, 21
 - hervorheben, 140
 - in Anzeigefenster einfügen, 42
 - layouten, 60
 - zulässige Namen, 17
- Sicherheitsabfrage, 134
- Skript
 - ausführen, 97
- Skriptdateien, 116
- Softwareanforderungen, 147
- Speicherverhalten, 144
- Sprache der Dokumente, 149
- Standard-Selektion, 18
- Starten von GIANT, 23
- Statuszeile, 119

Sun SPARC, [147](#)

Umwandeln von Selektionen und IML-Teilgraphen,
[20](#)

verschieben

 einzelne Knoten, [52](#), [54](#)

 ganze Selektionen, [52](#)

Verwaltungsdateien, [109](#), [110](#)

Visualisierung

 Attribute, [140](#)

 Kanten, [140](#)

 Klassenamen, [140](#)

 Knoten, [139](#)

 Knoten-Icon, [139](#)

 Knoten-Rechteck, [139](#)

 Knoten ID's, [140](#)

Visualisierungsstile, [12](#)

 innerhalb eines Anzeigefensters, [46](#)

 klassenspezifische Einstellungen, [115](#)

 Konfiguration, [114](#)

 Standard-Visualisierungsstil, [114](#)

 Stilauswahl-Combobox, [122](#)

Wartbarkeit des Produktes, [143](#)

XML Dateien, [149](#)

Zielposition, [19](#)

Zoom-Kontrolle, [122](#)

Zoomen

 Detailstufen, [141](#)

zoomen, [48](#)

zoomen auf eine Kante, [51](#)

zoomen auf gesamten Anzeigeeinhalt, [50](#)

zoomen auf Selektion, [49](#)

Zoomstufe, [12](#), [48](#), [49](#), [124](#)