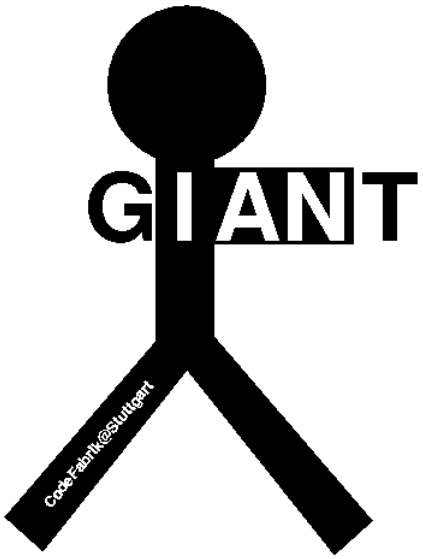
Universität Stuttgart
Studienprojekt A – IML Browser
CodeFabrik@Stuttgart

Handbuch

Version 1.3



Versionsgeschichte

- Version 1.0

  Diese Version wurde mit dem Kunden durchgesprochen, nötige Änderungen wurden festgehalten.

- Version 1.1

  Änderungen laut Kundenwunsch und Anpassungen an geänderte/wegefallene Funktionalitäten durchgeführt. Änderungen und Korrekturen laut internem Review durchgeführt.

- Version 1.2

  Änderungen an hinzugekommener Funktionalität durchgeführt. Änderungen und Korrekturen laut internem Review durchgeführt. Tutorial mit endgültigem Produkt hinzugefügt.

- Version 1.3 (24.09.03)

  Kleinere Korrekturen durchgeführt, Index aktualisiert.

Inhaltsverzeichnis

1. Einleitung

1.1. Über dieses Dokument

Dieses Handbuch beschreibt die Installation und Benutzung des IML-Browsers GIANT (Graphical IML Analysis Navigation Tool).

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

1.3. Aufbau dieses Dokuments

In den ersten Kapiteln dieses Handbuches wird beschrieben, wie Anwender mit GIANT anhand einiger Beispiele Analysen durchführen können. Dies wird anhand einer Art Tutorials beschrieben. Dies ermöglicht erfahrenen Anwendern einen schnellen Start („Top-Down“) mit GIANT.

In den späteren Kapiteln wird GIANT Fenster für Fenster und Menüpunkt für Menüpunkt genau erklärt („Bottom-Up“). Dies ermöglicht auch das Nachschlagen von Details zu einzelnen Funktionen.

2. Produktübersicht

In diesem Kapitel werden einige grundlegende Konzepte und Möglichkeiten von GIANT vorgestellt und kurz beschrieben.

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 8.

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 Anzeigeinhaltes 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 Anzeigeinhalt 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 Anzeigeinhaltes und die Zoomstufe gespeichert, so dass zu beliebigen Zeitpunkten die Position des sichtbaren Anzeigeinhalts 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 9.3 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 8.2.4.

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 GSL ist unter Kapitel 7 im Detail spezifiziert.

3. Technische Produktumgebung

Hier wird die zum Betrieb von GIANT nötige Hardware und die nötige Software sowie die Installation von GIANT auf Linux-Systemen beschrieben.

3.1. Software

Zum Betrieb von GIANT wird folgende Hardware und Software benötigt:

- Sun Solaris, Linux oder Windows Betriebssystem
- Emacs oder vi Texteditor für die Anzeige vom Sourcecode
- GTK 2.0 oder höher
- GTKAda 2.2

3.2. 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.

3.2.1. Hardwareanforderungen SPARC

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

3.2.2. Hardwareanforderungen x86

- Pentium III 600 MHz
- 512 MB Hauptspeicher

- 8 Bit Grafik mit einer min. Auflösung von 1024*786
- Maus mit mindestens zwei Tasten

3.3. Installation

Die Giant-Installation unter Linux geht folgendermaßen vonstatten:

Zunächst muß das Installationsarchiv (welches sich im Userverzeichnis, z.B. /home/sysop befindet), entpackt werden. Bei der GIANT-Version 1.1 ist das entpackte Distributions-Archiv ca. 450 MB groß.

Entpacken des gzip-Archives:

```
cd /home/sysop

gzip -d giant-1.1.0.tar.gz
```

Entpacken des darin befindlichen Tar-Archives:

```
tar xf giant-1.1.0.tar
```

Im nun existierenden Verzeichnis giant-1.1.0 kann GIANT mit ./giant gestartet werden.

3.4. Compilieren

GIANT kann durch Ausführen von `make` im Verzeichnis src compiliert werden.

4. Konzepte und Begriffe

Dieses Kapitel führt die für das Verständnis dieses Handbuches und die Benutzung von GIANT wichtigen Begriffe ein. Eine tabellarische Beschreibung der einzelnen Begriffe findet sich in der Spezifikation von GIANT, Kapitel Begriffslexikon.

4.1. Begriffe

Eine GSL-Anfrage (Query) beschreibt einen Vorgang, bei dem über geeignete Kriterien IML-Knoten und IML-Kanten aus dem IML-Graphen oder aus IML-Teilgraphen ausgewählt werden. GIANT greift über eine Schnittstelle, das Reflection Model von Bauhaus Reengineering, auf den IML-Graphen zu.

Die Visualisierung des Graphen erfolgt in Anzeigefenstern. Mit GIANT können Teilmengen des IML-Graphen gebildet werden, diese heißen IML-Teilgraphen.

In dieser Anleitung weden für Knoten und Kanten des Graphen die Begriffe Fenster-Knoten und Fenster-Kante verwendet, wenn diese in einem Anzeigefenster sichtbar sind, oder allgemein IML-Knoten bzw. IML-Kante. Die Namen Graph-Knoten und Graph-Kante werden verwendet, wenn Knoten bzw. Kante Bestandteil eines IML-Teilgraphen sind.

Der Begriff Kantenklasse (Edge Class) bezeichnet die Einteilung der IML-Kanten des IML-Graphen in verschiedene Klassen, wie sie sich aus der IML-Graph-Bibliothek von Bauhaus ergibt.

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.

Eine Knotenklasse (Node Class) ergibt sich aus der Einteilung der IML-Knoten des IML-Graphen in verschiedene Klassen, wie sie sich aus der IML-Graph-Bibliothek von Bauhaus ergibt.

Eine Klassenmenge (Class Set) ist eine durch IML-Dateien festgelegte Zusammenfassung von Kantenklassen und Knotenklassen. Es kann mehrere Klassenmengen geben. Die selben Knotenklassen und Kantenklassen können gleichzeitig zu mehreren Klassenmengen gehören.
Der Aufbau dieser Klasenmengendateien ist in Kapitel 9.3.6 beschrieben.

Eine Knoten-Annotationen (Node Annotation ist eine eine textuelle Beschreibung zu einem bestimmten Knoten des IML-Graphen, die der Nutzer hinzufügen kann.

Die zweidimensionale räumliche Anordnung von Fenster-Knoten und Fenster-Kanten innerhalb eines Anzeigefensters auf dem sogenannten Anzeigeinhalt wird Layout genannt.

Eine Auswahl von Fenster-Knoten und Fenster-Kanten eines visualisierten Teilgraphen des IML-Graphen innerhalb eines Anzeigefensters. heißt Selektion (Selection).

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

Der Nutzer kann in GIANT-Anzeigefenster zoomen, d.h. die Zoomstufe (Zoom Level) jedes Fensters verändern. Dieser Faktor beschreibt die Größe des sichtbaren Anzeigeinhaltes. 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 von Knoten 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.

4.2. Bezeichnertypen

Folgende Zeichen bzw. Zeichenfolgen werden von GIANT als gültige Bezeichner, z.B. als Namen für Fenster, akzeptiert:

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).

4.3. Allgemeine Bedienkonzepte

Hier werden grundlegende Konzepte der Bedienung von GIANT beschrieben.

4.3.1. 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.

4.3.2. Aktuelle Selektion

Es gibt zu jedem Anzeigefenster beliebig viele Selektionen (mindestens aber eine – die Standard-Selektion), davon ist immer eine die aktuelle Selektion.

Selektieren mittels der linken Maustaste und weiterer Funktionstasten (siehe 4.3.3) betrifft immer nur die aktuelle Selektion.
Alle anderen Selektionen können zwar hervorgehoben werden, aber nicht durch das Selektieren mittels linker Maustaste etc. geändert werden.

Der Benutzer kann jederzeit mittels Kontextmenü bestimmen, welche der Selektionen eines Anzeigefensters die aktuelle Selektion ist.

Die aktuelle Selektion wird immer hervorgehoben.

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

Hier wird beschrieben, wie man mittels der Maus und Tastatur einzelne Fenster-Knoten und Fenster-Kanten selektiert und deselektiert:

Die so erzielte Auswahl bezieht sich immer auf die aktuelle Selektion (siehe Selektionsauswahlliste im Fenster, Abschnitt 6.3.4).
Änderungen der Selektionen sind auch in der Selektionsauswahlliste im Fenster sichtbar.

Allgemein gilt: Selektiert werden kann mit der Maus durch Klick mit der linken Maustaste oder Aufziehen eines Rahmens mittels der gedrückten linken Maustaste.

Falls während des Selektierens mit der Maus (beide Methoden) die Shift-Taste auf der Tastatur gedrückt gehalten wird, so werden die markierten Knoten und Kanten der bisherigen Selektion hinzugefügt, bzw. entfernt, wenn diese schon hinzugefügt waren. Wird während des Aufziehens eines Rahmens mit der linken Maustaste die Control-Taste auf der Tastatur niedergedrückt gehalten, so werden alle Knoten und Kanten innerhalb des Rahmens hinzugefügt, egal, ob diese schon markiert waren.

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.

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

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

4.4.1. Vorgabe der Zielposition

Für bestimmte Funktionen muss der Benutzer manuell Zielpositionen innerhalb eines zu bestimmenden Anzeigefensters auswählen. An dieser Zielposition werden dann z.B. neue Fenster-Knoten eingefügt. Ablauf:

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.

Durch Klicken mit der linken Maustaste wird dann die Zielposition vorgegeben.

4.4.2. Einfügen von Selektionen in Anzeigefenster

Das grundlegende Verhalten beim Einfügen von Selektionen in Anzeigefenster (Kopieren einer Selektion aus einem Anzeigefenster in ein anderes) ist wie folgt:

Die Position von Fenster-Knoten der einzufügenden Selektion, die bereits in dem Anzeigefenster visualisiert sind, bleibt, falls nicht an entsprechender Stelle anders spezifiziert, unverändert.

Fenster-Kanten der Selektion werden nur eingefügt, falls ihr Start- und ihr Zielknoten ebenfalls Bestandteil der Selektion sind oder bereits im Anzeigefenster visualisiert sind.

4.4.3. Einfügen von IML-Teilgraphen in Anzeigefenster

Beim Einfügen von IML-Teilgraphen in Anzeigefenster beschrieben verhält sich GIANT wie folgt:

Das Layout für den IML-Teilgraphen wird vor dem Einfügen gemäß eines vom Benutzer vorgegebenen Layoutalgorithmus berechnet.

Die Graph-Knoten und Graph-Kanten des IML-Teilgraphen werden dann analog zu 4.4.2 als Fenster-Knoten und Fenster-Kanten in das Anzeigefenster eingefügt.

4.5. Verhalten bei Umwandeln von Selektionen und IML-Teilgraphen

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

4.5.1. Selektion aus IML-Teilgraphen ableiten

Beim Ableiten von Selektionen aus IML-Teilgraphen gilt folgendes:

Der IML-Teilgraph bleibt unverändert. 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.

Knoten und Kanten des IML-Teilgraphen, die nicht als Fenster-Knoten und Fenster-Kanten im entsprechenden Anzeigefenster vorhanden sind, werden ignoriert.

4.5.2. IML-Teilgraph aus Selektion ableiten

Beim Ableiten von IML-Teilgraphen aus Selektionen bleibt die Selektion unverändert.

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.

4.6. 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.

4.6.1. Entfernen aller Knoten und Kanten einer Selektion

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

Alle Fenster-Knoten und Fenster-Kanten der Selektion werden aus dem Anzeigefenster entfernt.

Fenster-Kanten im Anzeigefenster, deren Start- oder Zielknoten entfernt wird, werden ebenfalls entfernt (auch wenn sie nicht zur entsprechenden Selektion gehören).

Andere Selektionen des Anzeigefensters werden entsprechend aktualisiert, die betroffenen Fenster-Knoten und Fenster-Kanten werden auch aus diesen Selektionen entfernt.

4.7. Auseinanderschieben von Fenster-Knoten

Sollen Fenster-Knoten auf dem Anzeigeinhalt auseinandergeschoben werden, so geschieht dies wie hier beschrieben und dürfte schneller von der Hand gehen als das manuelle Auseinanderschieben durch Verschieben jedes einzelnen Knotens. GIANT stellt diese Funktionalität dem Benutzer über eine Funktion zur manuellen Ausführung zur Verfügung, siehe Abschnitt 6.3.2.2, „Make Room“. Ein automatisches „Auseinanderschieben von Fenster-Knoten“ durch GIANT selbst ist nicht vorgesehen.

Die Fenster-Knoten werden immer entlang einer Strecke von diesem Punkt durch den jeweiligen Fenster-Knoten verschoben. Jeder Fenster-Knoten wird um einen konstanten Betrag (kann vom Benutzer eingegeben werden) von dem Punkt weggeschoben.

5. Tutorial

Dieses Kapitel beschreibt einige häufig vorkommende Analysen mit GIANT und wie sie vom Benutzer ausgeführt werden können, um dem Benutzer einen schnellen Start der Arbeit mit GIANT zu ermöglichen.
Eine genauere Beschreibung der Funktionen befindet sich in Kapitel 6.

In diesem Tutorial gehen wir davon aus, daß sich das Programm GIANT im Verzeichnis /projects/tmp/haeusepp/giant-1.1.0 befindet.

5.1. Starten des Programms

Das Programm wird mit:

```
cd /projects/tmp/haeusepp/giant-1.1.0
./giant.sh
```

gestartet.
Wir erstellen mittels `mkdir /projects/tmp/haeusepp/giant-1.1.0/first_example` gleich ein neues Verzeichnis für unser erstes Projekt.

5.2. Erstellen eines neuen Projektes

Nun erscheint das Hauptfenster. Im Menü `Project` ist zum Erstellen eines neuen Projektes der Punkt `New` anzuwählen.

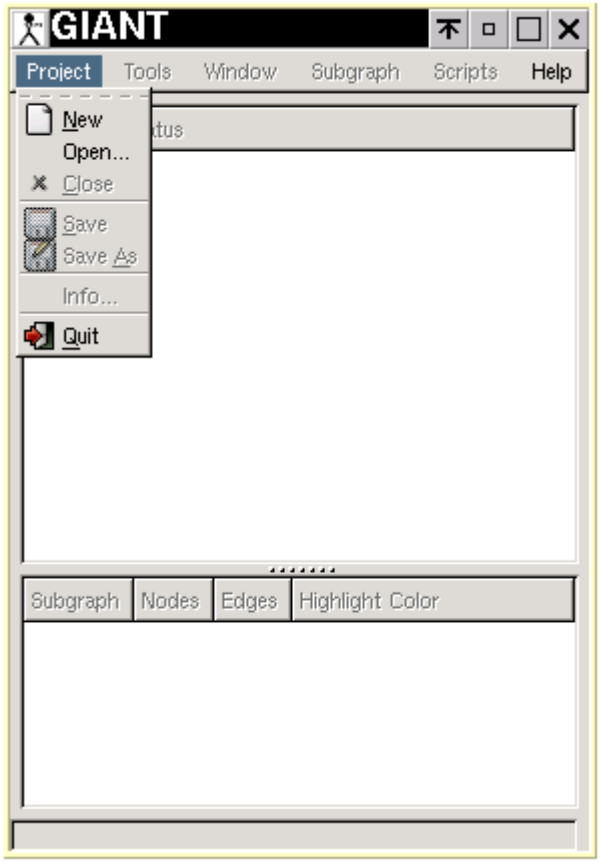


Abbildung 5.1.: Main-Window

GIANT verlangt über den Dateiauswahl-Dialog nach Pfad und Namen der neu zu erstellenden Projektdatei. Wir wählen hier
`/projects/tmp/haeusepp/giant-1.1.0/first_example/first_example`.
Dann fragt GIANT durch einen Dateiauswahl-Dialog nach dem IML-Graphen, auf dem das Projekt basieren soll. Wir wählen hier
`/projects/tmp/haeusepp/giant-1.1.0/graphs/rfg_examp.xml`.

GIANT meldet nun auf der Statuszeile links unten „Project Initialized“, rechts unten steht der Name der IML-Datei, `rfg_examp.iml`.

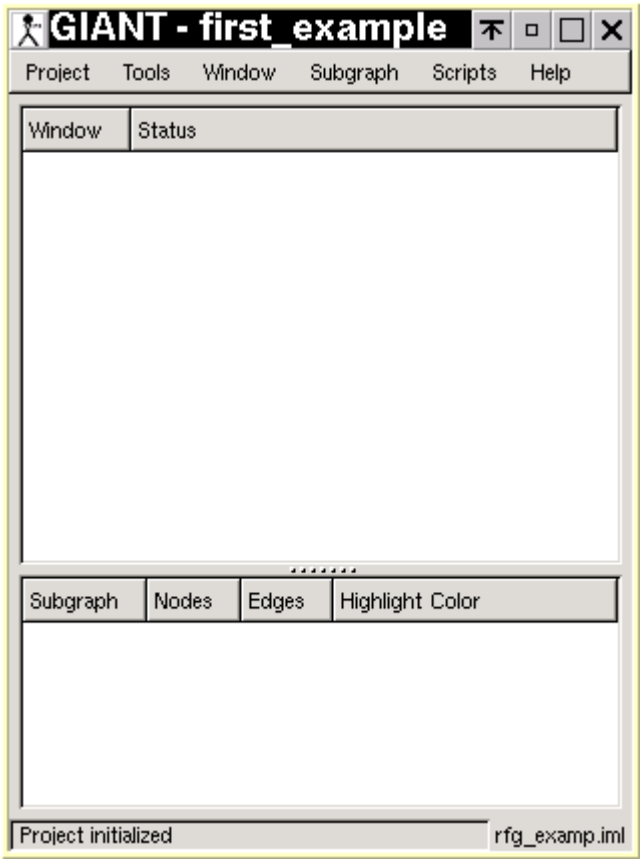


Abbildung 5.2.: Main-Window Bild 2

5.3. Öffnen eines Fensters

Wir wollen nun ein erstes Graphenfenster öffnen. Das erledigen wir durch Auswählen des Menüpunkts `Open Entire Graph` unter `Scripts`.

Nun erscheint ein Graphenfenster.

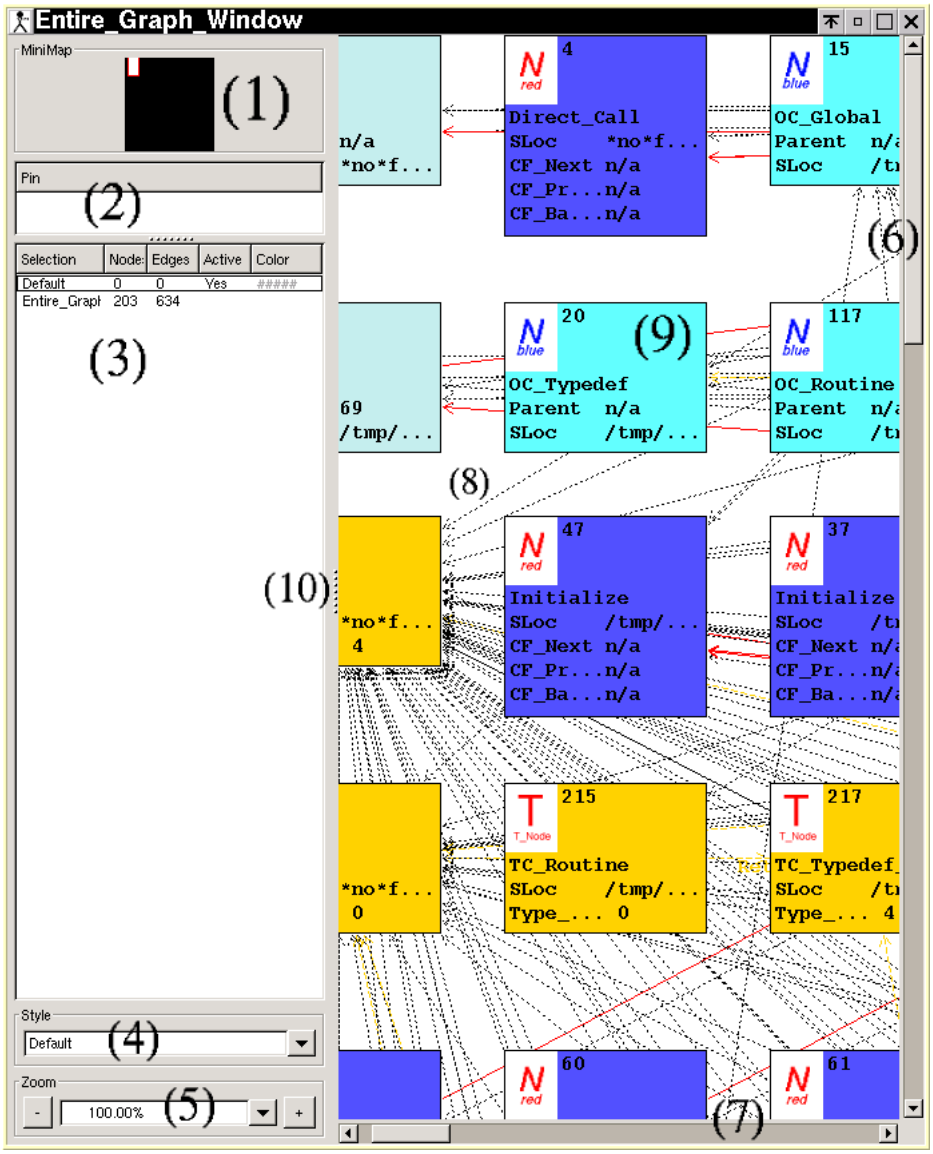


Abbildung 5.3.: Graphenfenster

5.4. Zoomen

Mittels der Zoomkontrolle (4) können wir die Ansicht im Fenster vergrößern oder verkleinern. Dazu können die Buttons + und - verwendet werden, mittels der Combobox voreingestellte werte gesetzt werden oder ein Wunschwert per Tastatur eingetragen werden. Wir tragen `45.00` Prozent ein (und drücken Return), um einen guten Überblick zu erhalten.
Wir wählen eine beliebige Kante mit der linken Maustaste an und wählen `Zoom to Edge`, worauf GIANT so scrollt und zoomt, daß diese Kante komplett auf dem Bildschirm sichtbar ist. Wir tragen nun wieder 45

5.5. Scrollen im Fenster

Nun wollen wir den sichtbaren Anzeigeinhalt verschieben. Dies geht durch Verschieben des weißen Kästchens in der MiniMap (1) mit der Maus. Wir verschieben das Kästchen ganz in die linke obere Ecke, wo wir einige Knoten sehen können.

Selbstverständlich kann die Fenstergröße in der von anderen Programmen gewohnten Weise verändert werden. Zusätzlich kann die Breite der linken Hälfte des Fensters durch ziehen von Element (10) verändert werden. Durch Betätigen der Scrollbars (6) und (7) kann wiederum der sichtbare Anzeigeinhalt verschoben werden.

5.6. Knoteninformation anzeigen

Wir suchen nun durch zoomen und scrollen den Knoten mit der Nummer 10, welcher sich links Oben im Block der Knoten befindet (es kann auch ein beliebiger anderer Knoten verwendet werden). Durch Klick mit der rechten Maustaste auf den Knoten erscheint ein Popup-Menü, wir wählen „Show Info“. Es werden hier einige Informationen zum Knoten angezeigt (Bild 5.4). Durch Klicken von „Pick“ und Klick auf einen anderen Knoten kann nunmehr die Information dieses anderen Knoten angezeigt werden.

Wir schließen das Nodeinfo-Fenster wieder.

5.7. Pin festlegen / ändern / löschen

Wir wollen uns nun eine Ansicht des Graphen für ein späteres schnelles Wiederfinden merken. Dazu scrollen und zoomen wir in eine für uns interessante Position (z.B. auf Knoten 176 rechts oben im Graphen), klicken mit der rechten Maustaste auf eine weiße Fläche in der Ansicht und wählen im Menü „New Pin...“ an. Bei der Frage nach dem Namen des Pins geben wir als Namen z.B. `Kn-176` ein. Wenn wir nun in eine andere Position scrollen und zoomen, können wir durch Aktivieren von `Show` im Popup-Menü, welches bei Auswahl des Knotennamens mit der Rechten Maustaste in der Pinliste (2) erscheint, wieder zu genau dieser Ansicht zurückspringen.
Mit `Rename...` kann der Name des Knotens geändert werden. Wir ändern den Knotennamen auf

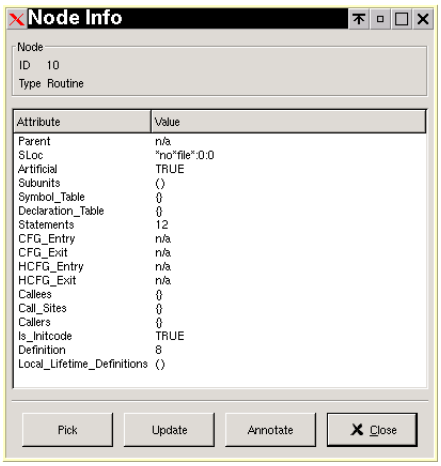


Abbildung 5.4.: Node-Info Fenster

`Interessanter_Knoten`. Falls wir den Knoten später einmal löschen wollen, geht dies mit `Delete` im Kontextmenü.

5.8. Visualisierungsstile ändern

Mittels der Visualisierungsstil-Auswahl (4) können wir die Art der Darstellung der Knoten ändern. Probeweise ändern wir die Darstellung in der Combobox einmal auf `Plain`. Wir können nun einmal im Graph herumzoomen und scrollen, um die Unterschiede zu sehen. Nun wechseln wir aber wieder zurück zum Default-Stil.

5.9. Selektionen bearbeiten und markieren

Wir springen nun zu unserem Pin `Interessanter Knoten`, und zoomen und scrollen ggf. so, daß wir ca. 8 Knoten bei einer Zoomstufe von 70 Prozent auf dem Bildschirm haben.

Nun klicken wir auf den Knoten Nr. 176 (rechts oben in der Matrix) mit der linken Maustaste.

Der Knoten wird farbig markiert. Wir können sehen, daß in der Selektionsliste (3) in der Zeile `Default` der angewählte Knoten auftaucht. Wann immer wir mit der Maus Selektionen manipulieren, so bezieht sich diese Änderung auf diejenige Selektion in der Liste, bei der in der Spalte `Active` ein `Yes` steht, die aktive Selektion.

Mit gedrückter Shift-Taste wählen wir noch zwei andere Knoten aus. Nun halten wir die Shift-Taste und ziehen mit gedrückter linker Maustaste einen Rahmen über alle bis auf die unteren zwei Knoten. Wenn wir loslassen, können wir feststellen, daß GIANT den momentanen Status (Selektiert oder

nicht) aller Knoten im Rahmen umgekehrt hat.

Wenn wir einen Rahmen ohne gedrückte Tasten ziehen, werden alle bisherigen Knoten in der Default-Selektion entselektiert. Wir ziehen nun einen Rahmen über die oberen beiden Knoten. Diesen wollen wir nun die unteren beiden Knoten hizufügen. Dies geschieht durch Anklicken dieser beiden Knoten mit gedrückter Shift-Taste. Alternativ könnte auch ein Rahmen mit gedrückter Shift-Taste gezogen werden.

Nun ziehen wir einen Rahmen mit der Maus über alle Knoten im Fenster bei gedrückter Ctrl-Taste. Wir können sehen, daß GIANT hier alle Knoten im Rahmen markiert, im Gegensatz zur Umkehrung des Selektionsstatus bei Verwendung der Shift-Taste.

Zu guter Letzt ziehen wir nochmals einen Rahmen ohne gedrückte Tasten über die oberen vier Knoten.

Das Selektieren von Kanten funktioniert völlig analog zum Selektieren der Knoten.

Durch den Rahmen haben wir nun eine gewisse Anzahl von Knoten und Kanten in der Default-Selektion sichtbar in (3). Wir duplizieren nun diese Auswahl durch Auswählen von `Duplicate` im Kontextmenü nach Drücken der rechten Maustaste über `Default` in der Selektionsliste (3). Als Namen wählen wir `Neu`.

Nun wollen wir die Selektion `Neu` farbig markieren. Wir wählen dazu in (3) im Kontext-Menü dieser Selektion `Highlight - Color 1` an. Wir können nun sehen, daß die Selektion farbig markiert wurde. (Siehe Bild 5.5)

Durch Auswählen von `Select All` oder `Select Nothing` bei Rechtsklick auf ein Stück von Knoten oder Kanten leeren Hintergrund des Anzeigefensterinhaltes können alle Knoten und Kanten markiert oder entmarkiert werden.

Wenn wir nun im Kontextmenü der Selektion `Zoom to Selection` auswählen, zoomt und scrollt GIANT so in den Graph hinein, daß alle Knoten der Selektion sichtbar sind.

Mit unserem neu erworbenen Wissen erstellen wir nun eine Selektion mit der Vierergruppe Knoten unten rechts im Graphen und nennen sie `vier`. Nun wählen wir im Kontextmenü dieser Selektion `Delete` aus und wir sehen, daß die Knoten noch da sind, die Selektion aber gelöscht wurde. Nun erstellen wir nochmals eine Selektion mit dem Namen `neu2`, wählen diesmal `Delete with Content` aus. GIANT hat nun die Knoten nebst ihrer Kanten aus dem Graph gelöscht.

Nun zu einer Mengenoperation: Auf der Selektion `Neu` in (3) wählen wir nun `Set Operation` aus. Es erscheint der Set Operation Dialog. Wir wählen (siehe Bild 5.6)

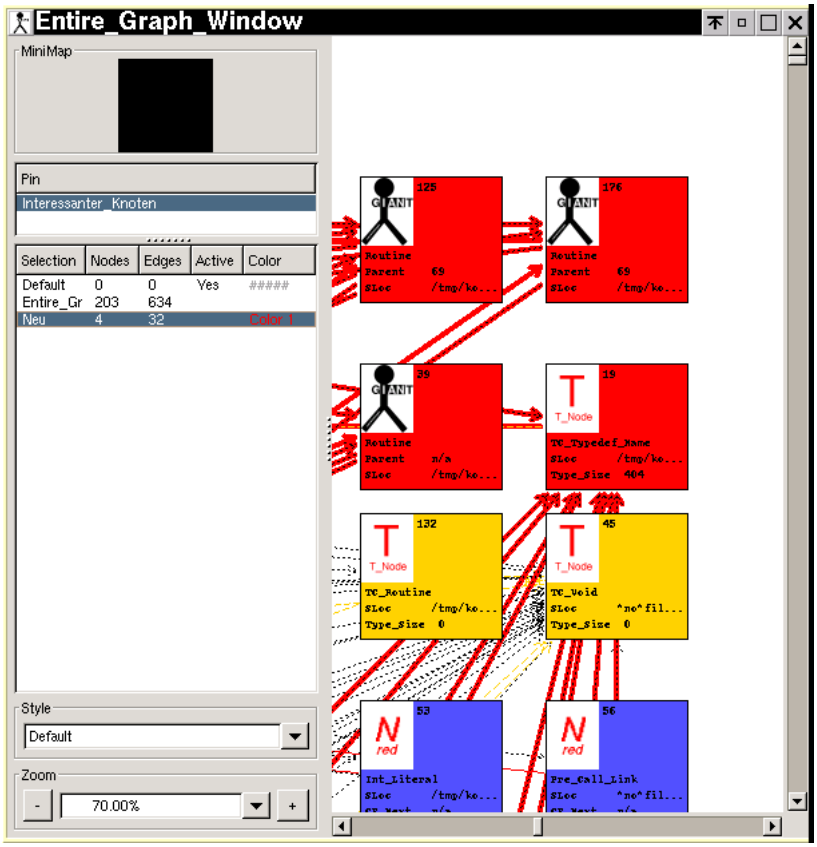


Abbildung 5.5.: Graphenfenster mit rot hervorgehobenem Graph

```
Left Source = Entire_Graph
Operation = Difference
Right Source = Neu
Target = Differenzmenge
```
aus.

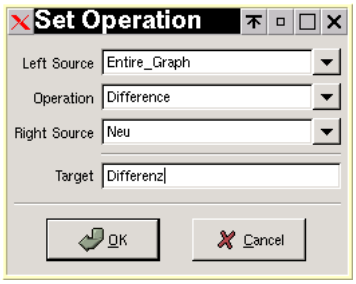


Abbildung 5.6.: Set Operation Dialog für Mengenoperationen

Nun bildet GIANT die Differenz zwischen den beiden Mengen `Entire_Graph` (also dem Gesamtgraphen) und der Menge `Neu`, die neue Menge heisst `Differenzmenge`. Man kann sehen, wie sich die Kantenzahlen unterscheiden.
Nun wählen wir auf `Differenzmenge` den Menüpunkt `Set Active` aus. Diese Menge ist nun die aktive Selektion. Wir können nun sehen, daß alle Knoten im Graphen, die zur Menge gehören, markiert sind. Mit gedrückter Shift-Taste können wir nun z.B. Knoten zur Selektion hinzufügen oder aus ihr entfernen.

Nachdem wir einige Knoten ausgewählt haben, wählen wir wieder die `Default` Selektion als Aktiv an.

Unsere Sekektion `Differenzmenge` mit ihren ausgewählten Knoten fügen wir nun als eigenständigen Subgraphen in GIANT ein, dies passiert durch Auswählen von `Insert as Subgraph` in ihrem Kontextmenü. Wie wir sehen, erscheint sie nun im Hauptfenster in der Subgraph-Liste.

5.10. Verschieben von Knoten, Platz schaffen

Wir können mit der linken Maustaste durch Drag and Drop Knoten verschieben. Es kann vorkommen, daß Knoten aufeinander liegen, sie können dann damit auseinandergeschoben werden.

Durch Klick mit der rechten Maustaste auf einen freien Platz im Anzeigeinhalt des Graphenfensters können wir im Popup-Menü `Make Room` anwählen. Durch diese Operation werden die Knoten rund um den freien Platz auseinandergeschoben, je näher desto weiter. Bild 5.7 zeigt das Ergebnis einer solchen Operation.

5.11. Mehrere Fenster

Durch Auswählen von `Window - New` erschaffen wir ein neues leeres Fenster. Dieses wird in der Fensterliste im Hauptfenster angezeigt. Das Fenster heisstr `Unknown`, wir ändern seinen Namen durch Anwählen von `Rename` auf im dazugehörigen Popup-Menü in der Liste auf `Testfenster`.
Wir schließen nun unser eingangs erstelltes Fenster `Entire_Graph_Window` durch Klick auf den Close-Button rechts oben und speichern es bei der erscheinenden Sicherheitsabfrage. Mittels `Open` in seinem Popup-Menü können wir es bei Bedarf wieder öffnen.

Nun wollen wir unseren Teilgraphen in das neue Testfenster einfügen. Dazu wählen wir im Hauptfenster auf dem erstellten Subgraphen im Popup-Menü `Insert as Selection`. GIANT fordert uns auf, in den leeren Anzeigeinhalt des Fensters zu klicken, dort wird der Subgraph als Selektion eingefügt.

Wir öffnen nun wieder unser erstes Fenster `Entire_Graph_Window`durch Auswählen von `Open` auf seinem Popup-Menü.

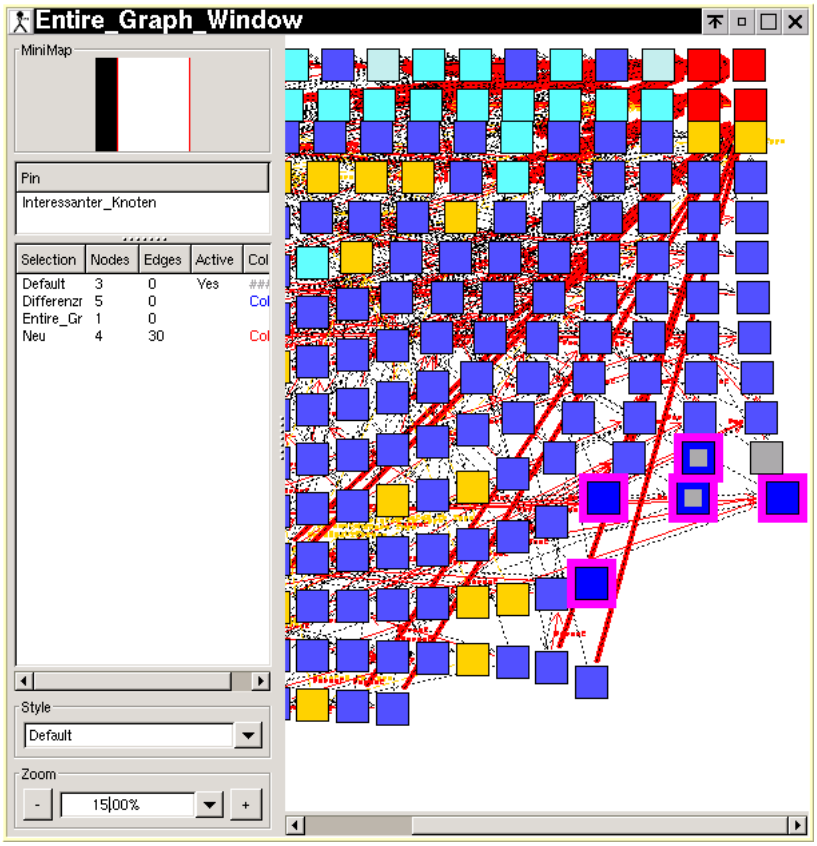


Abbildung 5.7.: Resultat von Platz schaffen

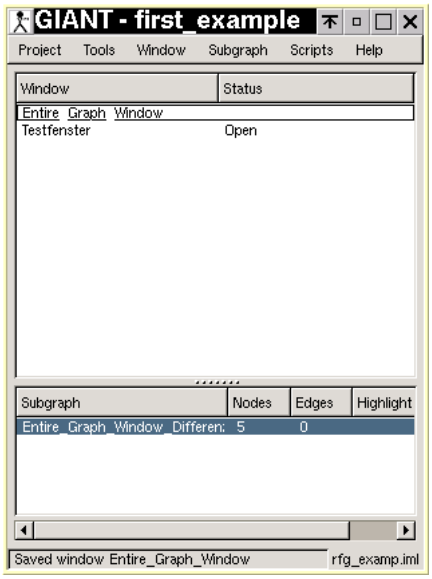


Abbildung 5.8.: Hauptfenster

Wir wählen im Kontextmenü des Subgraphen im Hauptfenster aus
`Highlight - Color 2` und können sehen, daß unsere Selektion in beiden Fenstern hervorgehoben wurde. Mittels `Unhighlight in alle Windows` läßt sich die Markierung rückgängig machen.

Wir schließen und speichern das `Testfenster`.

5.12. Layouts

Mittels des oben erwähnten `Select All` entmarkieren wir alle Knoten in unserem verbleibenden Fenster. Wir sorgen dafür, daß im Fenster ca. die Hälfte des Anzeigeinhaltes frei ist. Wir wählen durch Ziehen mit der Maus ca. 10-15 Knoten im linken oberen Teil des Graphen aus und wählen dann in der Selektionsauswahlliste auf der aktiven `Default` Selektion `Apply Layout` aus.

Mittels `Matrix Layout` läßt sich der Inhalt des Fensters in eine Matrixform bringen. Wir wählen diese Option im Tab aus (siehe Bild 5.9).

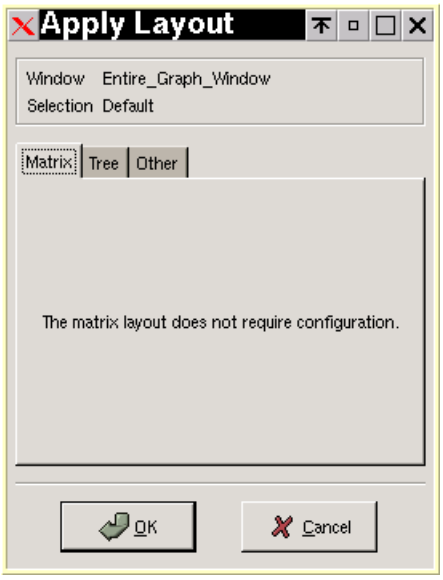


Abbildung 5.9.: Apply Layout

Nach Klick auf `OK` klicken wir an eine leere Stelle im Anzeigefenster- Inhalt, dies zeigt GIANT, wo die umorganisierten Knoten platziert werden sollen. Ein mögliches Resultat wird in Bild 5.10 gezeigt.

Wir wählen im Popup-Menü des Knoten rechts oben `Show Info` aus und merken uns den oben im Fenster angezeigten numerischen Wert `ID` des Knotens.

Nun wählen wir ca. 5 * 5 Knoten in der rechten oberen Ecke des Graphen durch Ziehen mit der Maus aus, einer dieser Knoten sollte der mit der bekannten ID sein. Vorher sollten wir dafür sorgen, daß auch noch genügend freier Platz im Fenster sichtbar ist.

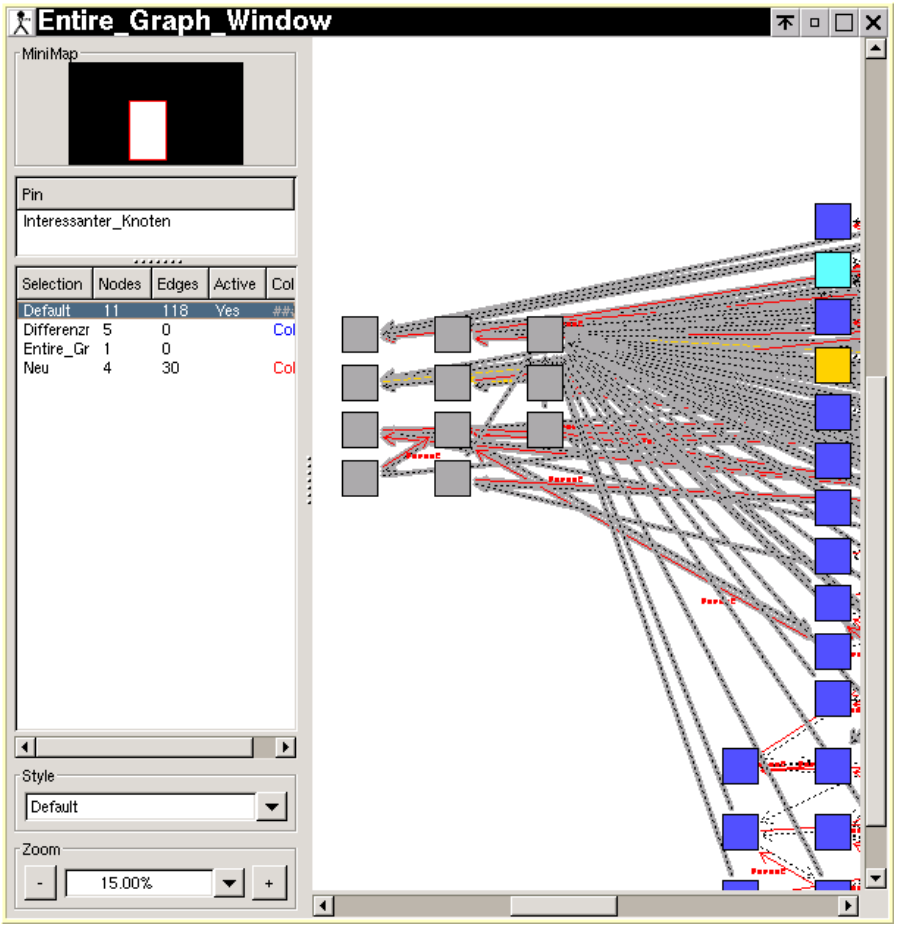


Abbildung 5.10.: Ergebnis des Matrix-Layouts

Wir wählen dann in der Selektionsauswahlliste auf der aktiven `Default` Selektion `Apply Layout` aus. In das Feld `Root Node ID` tragen wir die ID des Knotens ein. In der Liste `Class Sets` wählen wir `parent_edge_class_set`, `empty_class_set` und `normal_class_set` aus. Dies sind die Klassenmengen, die für das Layout berücksichtigt werden. Nun ist noch ein Klick an die Position des Wurzelknotens im Anzeigefenster nötig, und das Layout wird berechnet und angezeigt.

5.13. Benachbarte Knoten

Mittels des Menüpunktes `Scripts - Get Adjacent Nodes` können die benachbarten Knoten eines Knotens in eine Knotenmenge eingefügt werden. Wir führen dies mit einem beliebigen Knoten aus und können dies auch anhand der Knotenzahlen überprüfen.

5.14. Annotieren

Knoten können durch Klick auf `Annotate` im Knoteninformationsfenster (siehe Bild 5.4) oder durch Auswahl des Punktes `Annotate` im Knoten-Popup-Fenster annotiert werden.

Nach der Auswahl von `Annotate` öffnet sich das Knoten-Annotationsfenster (siehe Bild 5.11). Hier kann ein Kommentar zum Knoten eingefügt werden, der mit gespeichert wird. Durch Delete kann der Kommentar wieder gelöscht werden.
Annotationen werden nur dann auf Festplatte gespeichert, wenn wir im Hauptmenü `Project - Save` oder `Project - Save As..` auswählen, nicht jedoch, wenn nur einzelne Fenster gespeichert werden.

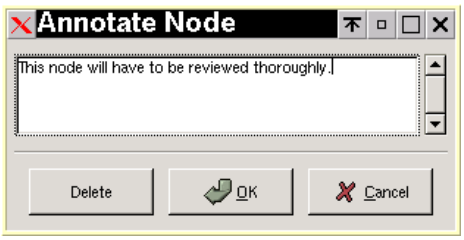


Abbildung 5.11.: Knoten-Annotationsfenster

Nach dem Annotieren wird der Benutzer bei genügend großer Zoomstufe durch ein kleines „Dokument“-Symbol auf die Annotation aufmerksam gemacht. (siehe Bild 5.12)

5.15. Fenster speichern

Wir wollen nun alle unsere Änderungen im Fenster speichern. Dazu wählen wir im Hauptfenster im Popup-Menü des Fensters `Save` aus, das Fenster ist nun gespeichert und erscheint beim nächsten Laden der Projektdatei auch in der Liste in genau dem Zustand (Knotenpositionen etc.), in dem wir es gespeichert haben. Lediglich Annotationen werden nicht gespeichert.

5.16. HPG darstellen

Durch Anwählen von `Scripts - Open HPG` im Menü des Hauptfensters von GIANT können wir ein Graphenfenster mit dem hierarchischen Programmgraphen öffnen.

5.17. Einbinden der GSL in Menüpunkte

Zur Verknüpfung von GSL-Ausdrücken mit Menüpunkten in den Popup-Menüs von Knoten bzw. Kanten oder dem Hauptmenü sind folgende Schritte nötig:

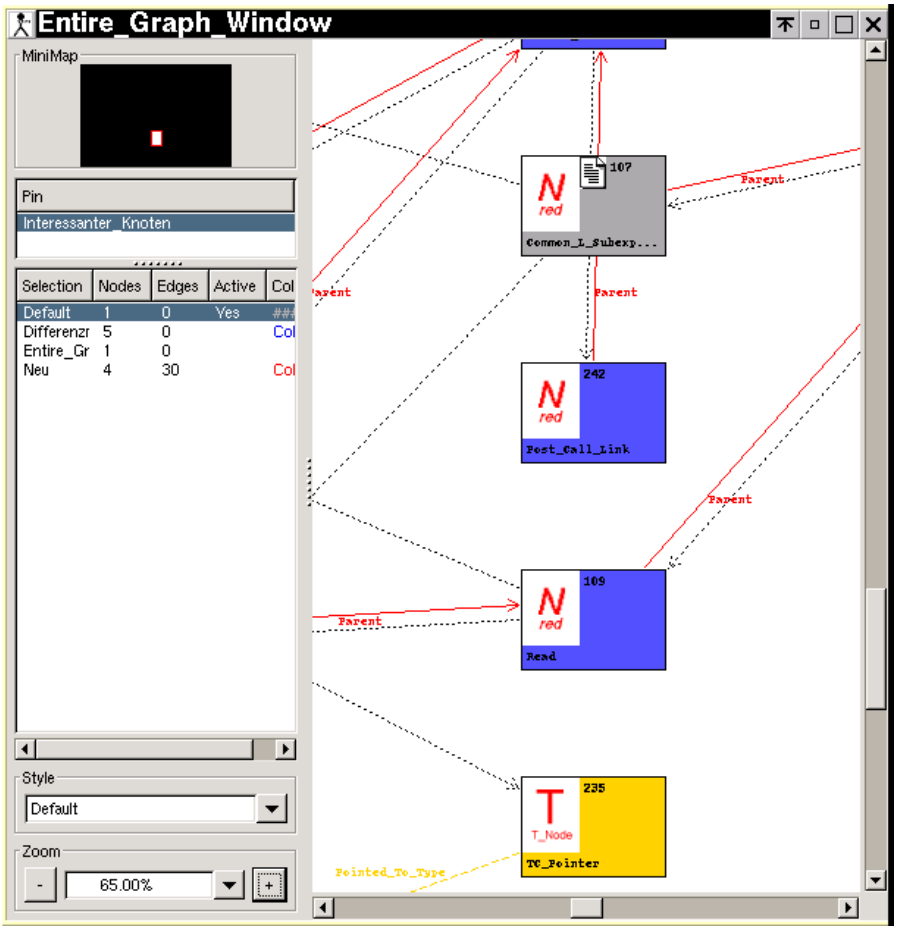


Abbildung 5.12.: Annotierter Knoten Nr. 107

Zuerst erstellen wir die GSL-Datei, welche ggf. den Knoten oder die Kante als „Parameter“ übergeben bekommt, und speichern diese im GIANT-Verzeichnis `shared/gsl` ab.

Nötig ist auch der Eintrag des Skriptes in der Config-Datei `etc/global-config.xml` im GIANT-Verzeichnis, hierzu muß diese in einen `GSL.` Setting-Knoten eingetragen werden, nähere Informationen können wir leicht dem mitgelieferten Config-File oder Abschnitt entnehmen, Trennzeichen ist der Doppelpunkt (`:`).
Der Eintrag im Menü muß dabei dem Filenamen entsprechen, ggf. ergeben sich dadurch auch Leerzeichen im Filenamen.

Soll das Skript im Hauptmenü aufrufbar sein, so fügen wir seinen Namen in Knoten `GSL.No_Param` ein. Für einen kontextabhängigen Aufruf aus einem Fenster (im Popup-Menü auf dem leeren Hintergrund des Fensters) ist `GSL.No_Param_Context` richtig, und für Knoten- Kanten- Teilgraph- oder Selektionssensitive Aufrufe sind `GSL.Node_Id_Param`, `GSL.Edge_Id_Param`, `GSL.Subgraph_Param` oder `GSL.Selection_Param` zuständig.

6. Beschreibung der Benutzeroberfläche

Dieses Kapitel beschreibt, geordnet nach den in GIANT auftretenden Fenstertypen, alle Buttons, Menüs und Funktionen. Dies ist als Nachschlagewerk über alle Elemente der graphischen Benutzeroberfläche (GUI) gedacht.

6.1. Kommandozeilenaufruf

Giant kann wie folgt von der Kommandozeile aus gestartet werden:

1. Ohne Parameter:

   ```
   ./giant
   ```

2. Mit Parametern, `[]` bedeutet „optional“, `|` bedeutet „oder“:

   ```
   ./giant  project-file [-g graph-file] [-e script-file]
        [-c config-file] | -h | -v
   ```

   `project-file` ist der Filename der Projektdatei, `graph-file` der Filename des IML-Graphen. `script-file` ist eine Datei, die ein GSL Script enthält, daß nach dem Laden bzw. Erzeugen des Projektes Ausgeführt wird. `config-file` ist die optional anzugebende globale Konfigurationsdatei, siehe Abschnitt 9.1.

   Die Option `-h` steht für Hilfe, mit `-v` kann die Programmversion von GIANT angezeigt werden.
   Wenn ein Graph-File angegeben wird und `project-file` nicht existiert, wird ein neues Projekt mit diesem Namen erzeugt.

   Ein Beispiel für den Kommandozeilenaufruf von GIANT ist:

   ```
   ./giant projects/sample.project -g graphs/mygraph.xml
   ```

6.2. Main Window

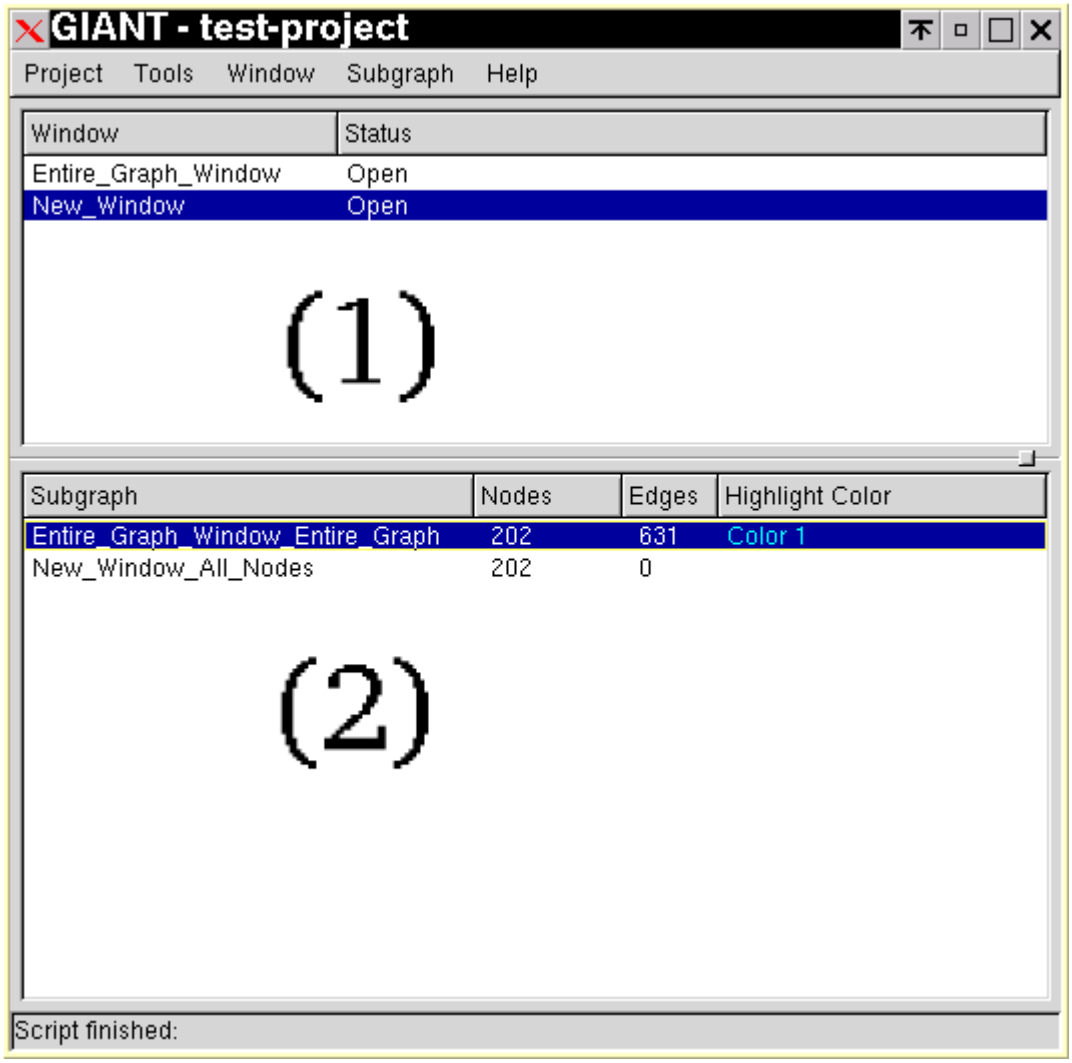


Abbildung 6.1.: Main-Window

Jede Instanz von GIANT hat genau ein Hauptfenster. Im Hauptfenster (siehe Abbildung 6.1) werden die Anzeigefenster und IML-Teilgraphen des aktuell geöffneten Projektes (siehe 8.1) in Listen angezeigt. Über Popup-Menüs können IML-Teilgraphen und Anzeigefenster manipuliert werden.

Im Fenster befinden sich zwei Listen, Window List (1) und Subgraph List (2).

6.2.1. Titelzeile

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

6.2.2. Menüleiste (Main Window)

In der Menüleiste befinden sich folgende Einträge:

6.2.2.1. Untermenü Project

1. New

   Hiermit wird ein neues Projekt angelegt. Ein eventuell bereits geöffnetes Projekt wird dabei geschlossen, wobei Änderungen auf Nachfrage vorher gespeichert werden.

   - GIANT zeigt den Standard-Filechooser-Dialog und fordert den Benutzer auf, eine vorhandene IML-Graph Datei auszuwählen.
   - GIANT zeigt erneut den Standard-Filechooser-Dialog und fordert den Benutzer zur Eingabe des Namens der Projektdatei auf. 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. Existiert die eingegebene Projektdatei bereits, erscheint eine Fehlermeldung. Existiert in dem Projektverzeichnis bereits eine andere Projektdatei, so erscheint ebenfalls eine Fehlermeldung.

2. Open

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

   - War vorher bereits ein Projekt offen, erscheint vorher eine Abfrage, ob dieses gespeichert werden soll.
   - GIANT zeigt den Standard-Filechooser-Dialog und fordert den Benutzer zur Auswahl einer vorhandenen GIANT-Projektdatei auf.

3. Save

   Speichert ein Projekt in die Verwaltungsdateien im Projektverzeichnis der neuen Projektdatei.

4. Save As...

   Speichert ein Projekt in die Verwaltungsdateien im Projektverzeichnis der neuen Projektdatei unter neuem Namen.

   - Der Benutzer gibt im Standard-Filechooser-Dialog das neue Projektverzeichnis und den Namen für die neue Projektdatei ein, die Dateiendung wird später von GIANT automatisch gesetzt.

5. Info

   Zeigt einen Informationsdialog über den Graphen an.

6. Quit

   Beendet das Programm GIANT.

6.2.2.2. Untermenü Tools

1. GSL Editor...

   Mit diesem Menüpunkt kann ein GSL Script ausgeführt werden. Es erscheint der Skriptdialog, in den das Skript eingegeben werden kann. Es ist auch möglich, Skripte zu laden oder zu speichern. Details hierzu finden sich unter 6.5.

6.2.2.3. Untermenü Window

1. New
   Mit diesem Menüpunkt kann ein neues leeres Fenster geöffnet werden.

6.2.2.4. Untermenü Subgraph

1. New
   Mit diesem Menüpunkt kann ein neuer leerer IML-Teilgraph erstellt werden.
2. Set Operation
   Mit diesem Menüpunkt kann eine Mengenoperation auf IML-Teilgraphen durchgeführt werden, GIANT zeigt dazu den „Set Operation Dialog“, siehe Abschnitt 6.7.

6.2.2.5. Untermenü Help

1. About..
   Dieser Menüpunkt gibt ein Fenster mit Informationen zur GIANT Programmversion und zu den Autoren aus.

6.2.3. Statuszeile

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

6.2.4. Window List (1)

Im Hauptfenster befindet sich zuoberst eine Liste „Window List“, welche alle Anzeigefenster des Projektes (offen und geschlossen anzeigt). Eine Beschreibung, was alles mit einem Fenster gespeichert wird, befindet sich in Abschnitt 8.1.

6.2.4.1. Inhalt Window List

Window List hat die Spalten „Window“ und „Status“. In „Window“ wird der Name aller Anzeigefenster dargestellt, in „Status“ befindet sich der Text Öpen", wenn das Fenster geöffnet ist.

6.2.4.2. Popup-Menü Window List

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

1. Open
   Öffnet ein im Speicher vorhandenes, momentan geschlossenes Fenster.
2. Close
   Schließt ein vorhandenes Fenster, es kann wieder geöffnet werden. Falls das Fenster noch nicht gespeichert wurde, wird eine Sicherheitsabfrage durchgeführt, ob gespeichert werden soll.
3. Save Window
   Speichert ein Fenster im Projektverzeichnis.
4. Rename Window
   Ändert den Namen eines Fensters, ein entsprechender Dialog erscheint.
5. Delete Window
   Löscht ein Fenster aus dem Projektverzeichnis, es wird auch aus der Window List gelöscht.

6.2.5. Subgraph List (2)

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

6.2.5.1. Inhalt Subgraph List

Subgraph List hat die folgenden Spalten:

1. Subgraph

2. Nodes
3. Edges
4. Color

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. Bei nicht hervorgehobenen IML-Teilgraphen wird dieses Kästchen nicht angezeigt.

6.2.5.2. 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
   Mit diesen drei Menüpunkten kann der angewählte IML-Teilgraph in allen Fenstern in der ausgewählten Farbe markiert werden.
2. Unhighlight In All Windows
   Mit diesen drei Menüpunkten kann der angewählte IML-Teilgraph in allen Fenstern wieder entfärbt werden, die Markierung also entfernt werden.
3. Insert as Selection
   Mit diesem Menüpunkt kann ein IML-Teilgraph als Selektion in ein Fenster kopiert werden. Nach Auswählen dieser Funktion ist mit der Maus (linke Taste) in das Fenster zu klicken, in das der Subgraph eingefügt werden soll, an eine passende Stelle zu klicken.
4. Rename Mit dieser Funktion kann der Name eines IML-Teilgraphen geändert werden, ein entsprechender Dialog erscheint nach der Auswahl dieses Punktes.
5. Duplicate Mit dieser Funktion kann ein IML-Teilgraph dupliziert werden, ein Dialog zur Auswahl eines neuen Namens erscheint nach der Auswahl dieses Punktes.
6. Delete Mit diesem Menüpunkt kann ein IML-Teilgraph nach Sicherheitsabfrage gelöscht werden.

Als weitere Menüpunkte im Popup-Menü erscheinen die im GIANT-Config-File definierten kontextsensitiven GSL-Skripte. Mehr Informationen hierzu finden sich in Abschnitt 9.2.

6.3. Anzeigefenster

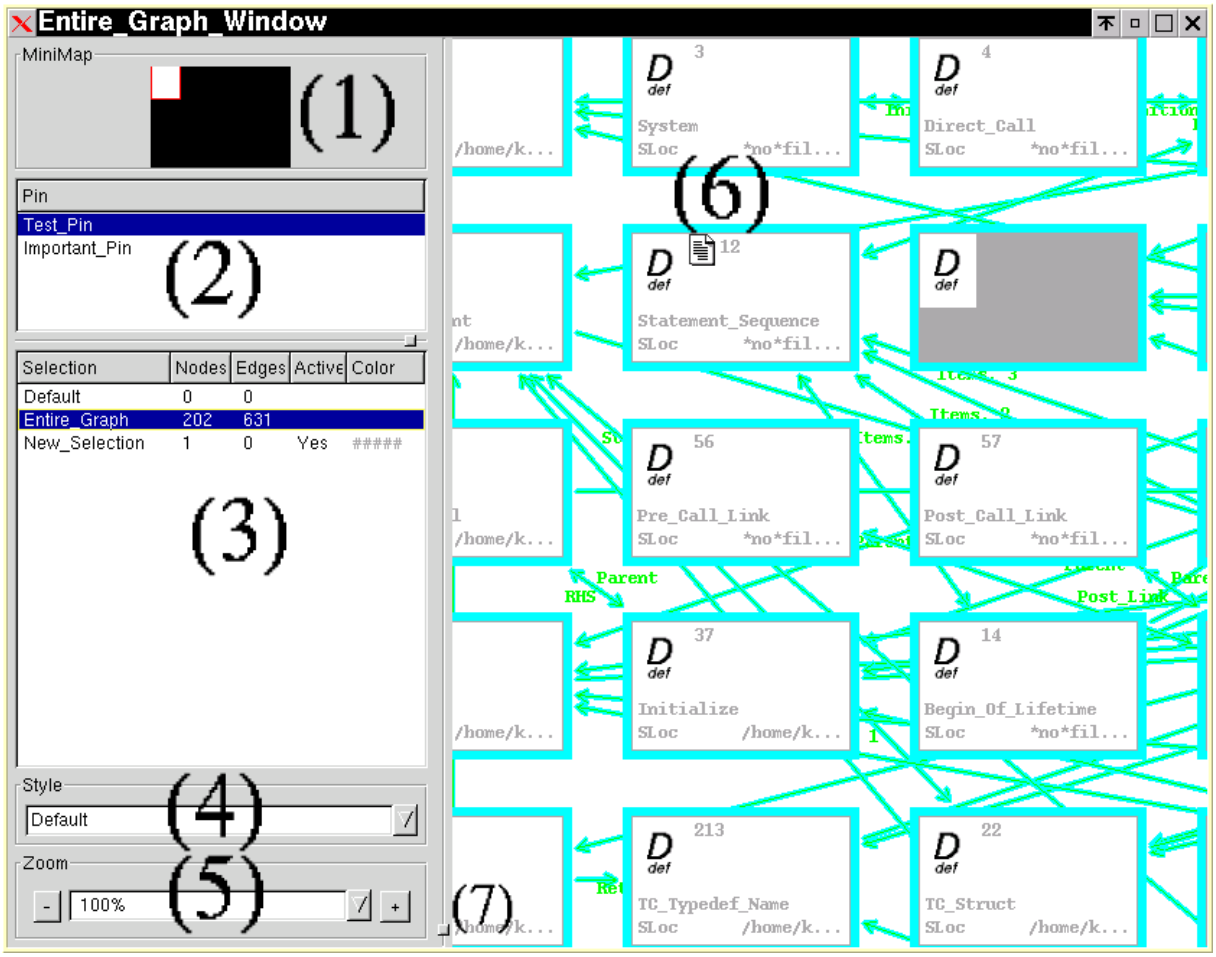


Abbildung 6.2.: Graph-Window

Im Programm kann es mehrere Anzeigefenster (siehe Abbildung 6.2) geben.

In Anzeigefenstern (Visualization Windows) werden Teile des IML-Graphen visualisiert. Sie erscheinen im großen quadratischen rechten Teil („Vis Pane“ (6)) des Fensters. Die Vis Pane enthält den Anzeigeinhalt des Fensters. Im linken Teil des Fensters, der Toolbar („Vis Toolbar“) befinden sich untereinander die MiniMap (1), Pinliste (2), Selektionsliste (3), die Stilauswahl-Combobox (4) und die Zoom-Kontrolle (5). Dieser Teil kann durch Verschieben des entsprechenden Buttons (7) unten an der Trennleiste in der Größe verändert werden.

6.3.1. MiniMap (1)

In der MiniMap wird der Anzeigeinhalt 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 Anzeigeinhalt 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 Anzeigeinhalt in der Vis Pane entsprechend angepasst.
Vergrößert sich die von Knoten bedeckte Fläche im Fenster, z.B. durch Einfügen von Teilgraphen, so verändert sich auch die Minimap entsprechend.

6.3.2. Visualisierung der Knoten und Kanten (6)

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

6.3.2.1. Node-Popup-Menü

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

1. Show Info
   Zeigt ein Fenster mit Informationen zum Knoten
2. Show Source
   Zeigt den zum Knoten gehörenden Source Code in einem externen Editorfenster
3. Annotate... Mit diesem Menüpunkt läßt sich eine Annotation zu einem Knoten hinzufügen, ändern oder löschen. Der Knoten ist dann ggf. graphisch als annotiert gekennzeichnet.
   - Der Nutzer führt einen Rechtsklick auf den Knoten aus, den er annotieren will und wählt im Popup-Menü „Annotate...“ an
   - Im erscheinenden Knotenannotations-Dialog kann der Text der Annotation eingegeben werden und mit OK bestätigt werden. Mittels der Buttons Delete kann die Annotation gelöscht werden, durch Cancel kann die bisherige Annotation beibehalten werden.
   - Nach dem nächsten Speichern des Projektes ist die Annotation dauerhaft erstellt.

Als weitere Menüpunkte im Popup-Menü erscheinen die im GIANT-Config-File definierten kontextsensitiven GSL-Skripte. Mehr Informationen hierzu finden sich in Abschnitt 9.2. Mit diesem Menü können die GSL-Skripte im Untermenü schnell auf den Knoten bezogen ausgeführt werden, d.h. ihnen wird die ID des Knotens als Parameter übergeben.

6.3.2.2. Background-Popup-Menü

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.

1. New Pin
   Mittels dieses Menüpunktes kann ein neuer Pin (siehe 6.3.3)angelegt werden.
   - Der Benutzer führt einen Rechtsklick auf den Anzeigeinhalt eines Anzeigefensters durch und wählt aus dem Popup-Menü (siehe 6.3.2.2) den Eintrag „New Pin“ aus. Der später erstellte Pin verweist dann auf die Stelle im Anzeigeinhalt, auf die der Rechtsklick durchgeführt wurde.

- Der Benutzer gibt im aufgehenden Texteingabedialog einen zulässigen Namen für den neuen Pin ein und bestätigt mit OK
- GIANT speichert die aktuelle Zoomstufe und die Position des sichtbaren Anzeigeinhaltes in einem neuen Pin.

2. Make Room
   Mittels dieses Menüpunktes können Knoten auseinandergeschoben werden, um Platz zu schaffen.
   - Der Nutzer führt einen Rechtsklick auf eine leere Stelle des sichtbaren Anzeigeinhalts in einem Anzeigefenster durch und wählt im erscheinenden Popup-Menü „Make Room“ an
   - 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.
   - GIANT zeigt einen Dialog an, in dem der Benutzer auswählt, um welchen Betrag die Fenster-Knoten auseinander geschoben werden sollen. Der Benutzer wählt einen geeigneten Betrag aus und bestätigt mit OK.
   - GIANT schiebt die Knoten entsprechend auseinander.
3. Select All
   Bei Auswahl dieses Menüpunktes werden alle Knoten im Fenster selektiert.
4. Select Nothing
   Bei Auswahl dieses Menüpunktes wird bei allen Knoten im Fenster die Selektierung aufgehoben. Als weitere Menüpunkte im Popup-Menü erscheinen die im GIANT-Config-File definierten kontextsensitiven GSL-Skripte. Mehr Informationen hierzu finden sich in Abschnitt 9.2.

6.3.2.3. Kanten-Popup-Menü

Wenn in der Vis Pane mit der rechten Maustaste auf eine Kante geklickt wird, öffnet sich ein Popup-Menü: Als Menüpunkte im Popup-Menü erscheinen die im GIANT-Config-File definierten kontextsensitiven GSL-Skripte. Mehr Informationen hierzu finden sich in Abschnitt 9.2. Mit diesem Menü können die GSL-Skripte im Untermenü schnell auf die Kante bezogen ausgeführt werden, d.h. ihnen wird die ID der Kante als Parameter übergeben.

6.3.3. Pins (2)

In der Pinliste („Pin List“) können Pins angesprungen oder verändert werden.

Im Popup-Menü der Pinliste befinden sich folgende Menüpunkte:

1. Show
   Mittels dieses Menüpunktes kann der sichtbare Anzeigeinhalt gemäß den im Pin gespeicherten Informationen gesetzt werden.

2. Rename
   Mittels dieses Menüpunktes kann der Name des Pins geändert werden, ein entsprechender Dialog erscheint.

3. Delete
   Mittels dieses Menüpunktes kann der Pin aus der Pinliste gelöscht werden.

6.3.4. Selektionsauswahlliste (3)

Die Selektionsauswahlliste hat die Spalten „Name“ mit dem Namen der Selektion, „Color“ mit der Farbe der Selektion und „Active“ mit „Yes“ bei der aktiven Selektion.

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

1. Set Active
   Mittels dieser Funktion kann eine Selektion zur aktiven Selektion gemacht werden.

2. Highlight (Menü)
   Diese Funktion dient zum Hervorheben von Selektionen innerhalb eines Anzeigefensters. 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, so ist diese Selektion nicht mehr hervorgehoben.

   a) Color 1
      Hebt in der im Config-File definierten Farbe 1 hervor

   b) Color 2
      Hebt in der im Config-File definierten Farbe 2 hervor

   c) Color 3
      Hebt in der im Config-File definierten Farbe 3 hervor

   Der Benutzer startet die Funktion mit einem Rechtsklick auf die gewünschte Selektion in der Selektionsauswahlliste und wählt im zugehörigen Popup-Menü das Untermenü „Highlight Selection“ und dort die gewünschte Farbe aus. GIANT hebt die Selektion mit der ausgewählten Farbe hervor.

3. Unhighlight Selection
   Diese Funktion dient dazu, die Hervorhebung von Selektionen innerhalb eines Anzeigefensters aufzuheben. Der Benutzer startet die Funktion mit einem Rechtsklick auf die gewünschte Selektion in der Selektionsauswahlliste und wählt im zugehörigen Popup-Menü den Eintrag „Unhighlight Selection“ aus.

4. Apply Layout
   Mittels dieser Funktion können Selektionen innerhalb eines Anzeigefensters layoutet werden. Die verfügbaren Layoutalgorithmen sind in Kapitel „Layoutalgorithmen“ beschrieben.

   - Der Benutzer wählt aus der Selektionsauswahlliste (siehe 6.3.4) 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.

- GIANT zeigt einen Dialog zur Auswahl und Konfiguration des gewünschten Layoutalgorithmus (siehe 6.8). Der Benutzer wählt in diesem Dialog den gewünschten Layoutalgorithmus aus.

5. Set Operation
   Zusätzlich zu den Möglichkeiten der Anfragesprache GSL (siehe Kapitel 7) kann der Benutzer die gängigen Mengenoperationen, wie Mengenvereinigung, Schnitt und Differenz, auch direkt über einen entsprechenden Dialog (siehe 6.7) ausführen, welcher mit dieser Funktion aufgerufen werden kann.

6. Insert As Subgraph
   Dieser Menüpunkt leitet einen neuen IML-Teilgraphen aus einer Quell-Selektion ab.

   - Der Benutzer führt einen Rechtsklick auf die Quell-Selektion in der Selektionsauswahlliste aus und wählt im Popup-Menü den Eintrag „Create New IML Subgraph from This Selection“ aus.
   - GIANT zeigt den allgemeinen Texteingabedialog an. Der Benutzer gibt einen Namen für den neu zu erstellenden IML-Teilgraphen ein und bestätigt mit OK. Gibt der Benutzer hier keinen Namen ein, so vergibt GIANT automatisch einen Namen.

7. Rename
   Mit dieser Funktion kann der Name einer Selektion geändert werden, GIANT zeigt einen entsprechenden Dialog.

8. Duplicate
   Mit dieser Funktion kann eine Selektion dupliziert werden, GIANT fragt in einem entsprechenden Dialog nach dem Namen der neuen Selektion.

9. Zoom To Selection Der im Fenster sichtbare Anzeigeinhalts wird durch automatisches Zoomen und Scrollen so verändert, dass die ausgewählte Selektion vollständig sichtbar ist.

10. Delete
    Diese Funktion dient zum Löschen von Selektionen innerhalb eines Anzeigefensters. Hierdurch bleiben die Fenster-Knoten und Fenster-Kanten unverändert. Die Standard-Selektion (siehe 4.3.1) kann nicht gelöscht werden. Wurde die aktuelle Selektion gelöscht (siehe 4.3.2), so wird die Standard-Selektion zur aktuellen Selektion. Die Fenster-Knoten und Fenster-Kanten, die zur gelöschten Selektion gehören, werden nicht gelöscht. War die Selektion hervorgehoben, so wird die Hervorhebung der Fenster-Knoten und Fenster-Kanten aufgehoben.

    - Der Benutzer startet die Funktion durch Rechtsklick auf die zu löschende Selektion in der Selektionsauswahlliste (siehe 6.3.4) und wählt aus dem Popup-Menü den Eintrag „Delete Selection“ aus. Falls der Benutzer die Standard-Selektion (siehe 4.3.1) ausgewählt hat, ist der Eintrag „Delete Selection“ deaktiviert.
    - GIANT löscht die entsprechende Selektion.

11. Delete With Content
    Mittels dieser Funktion können alle Fenster-Knoten und Fenster-Kanten einer Selektion aus einem Anzeigefenster gelöscht werden (siehe auch 4.6).
    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. Wurde die Standard-Selektion (siehe 4.3.1) gewählt, so wird diese nicht aus dem Anzeigefenster gelöscht, sondern ist jetzt leer.

- Der Benutzer führt einen Rechtsklick auf die entsprechende Selektion in der Selektionsauswahlliste durch (siehe 6.3.4) und wählt im Popup-Menü den Eintrag „Delete Nodes and Edges of Selection“ aus.
- GIANT zeigt die Sicherheitsabfrage (siehe 6.11) 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?“). Der Benutzer bestätigt mit Yes.
- GIANT löscht die Selektion samt allen zugehörigen Fenster-Knoten und Fenster-Kanten aus dem entsprechenden Anzeigefenster. Wurde die Funktion für die Standard-Selektion (siehe 4.3.1) ausgeführt, so werden die Fenster-Knoten und Fenster-Kanten aus dem Anzeigeinhalt gelöscht, die Standard-Selektion selbst wird geleert aber nicht gelöscht.

6.3.5. Stilauswahl-Combobox (4)

In der Stilauswahl-Combobox („Style Chooser“) kann ein Visualisierungsstil (siehe auch 9.3) ausgewählt werden.

6.3.6. Zoom-Kontrolle (5)

Die Zoom-Kontrolle besteht aus folgenden Elementen:

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

Mittels der Buttons „+“und „-“ kann die Zoomstufe nach oben bzw. unten verändert werden, pro Klick verändert sie sich um einen bestimmten Betrag.

6.3.7. Scrollen

Der sichtbare Anzeigeinhalt kann folgendermassen gescrollt werden:

- Durch Bewegen des sichtbaren Bereiches in der Minimap mit der Maus
- Durch Klicken mit der linken Maustaste auf eine leere Stelle des sichtbaren Anzeigeinhalts, festhalten der Maustaste und Bewegen des Mauszeigers aus dem Fenster heraus in die entsprechende Richtung (oben, unten, links oder rechts).

6.4. Knoten-Informationsfenster

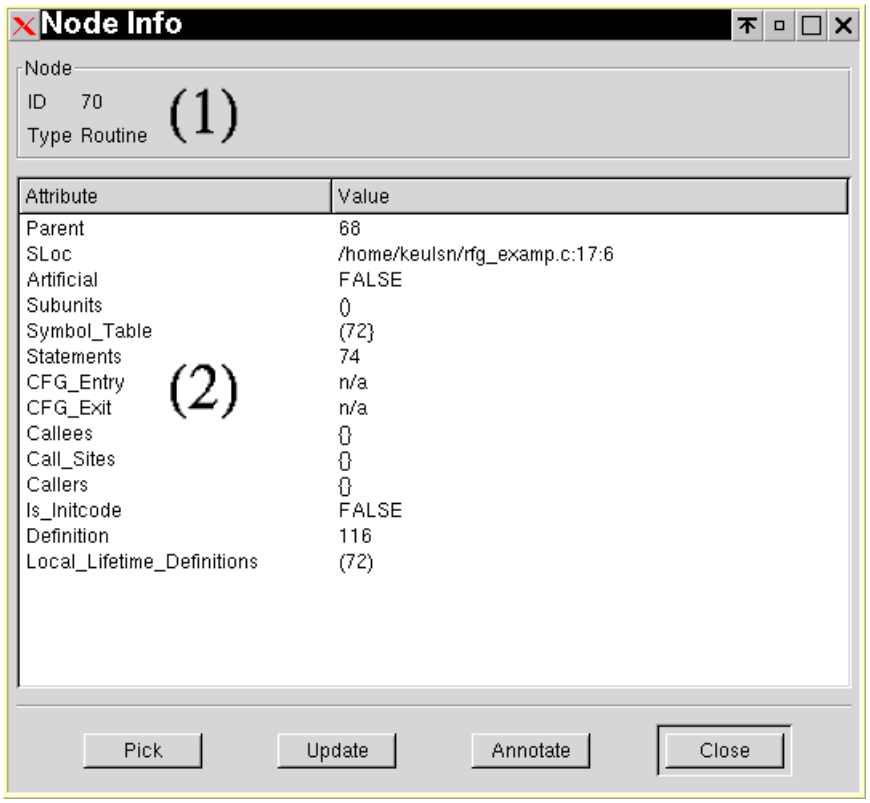


Abbildung 6.3.: Node-Info-Window

In Knoten-Informationsfenstern (siehe Abbildung 6.3) 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 (1), darunter befindet sich die Liste „Attributes“ mit den Attributen des Knotens (2).

Unter der Liste (2) befinden sich nebeneinander die Buttons „Pick“, „Update“, „Annotate“ und „Close“. „Pick“ dient zur Auswahl eines anderen Fenster-Knotens mittels des Fadenkreuzcursors, dazu wird nach Klicken von „Pick“ ein Fensterknoten angeklickt.

„Update“ dient zur Aktualisierung der Liste, Annotate öffnet das Knoten-Annotationsfenster (siehe Abschnitt 6.9.1).

Die Informationen zu dem ausgewählten Fenster-Knoten werden dann im Knoten-Informationsfenster dargestellt.

Das Knoten-Informationsfenster enthält, neben der Anzeige von Knoten-ID und Knoten-Typ, die Liste „Attributes“:

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

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

6.5. Skriptdialog

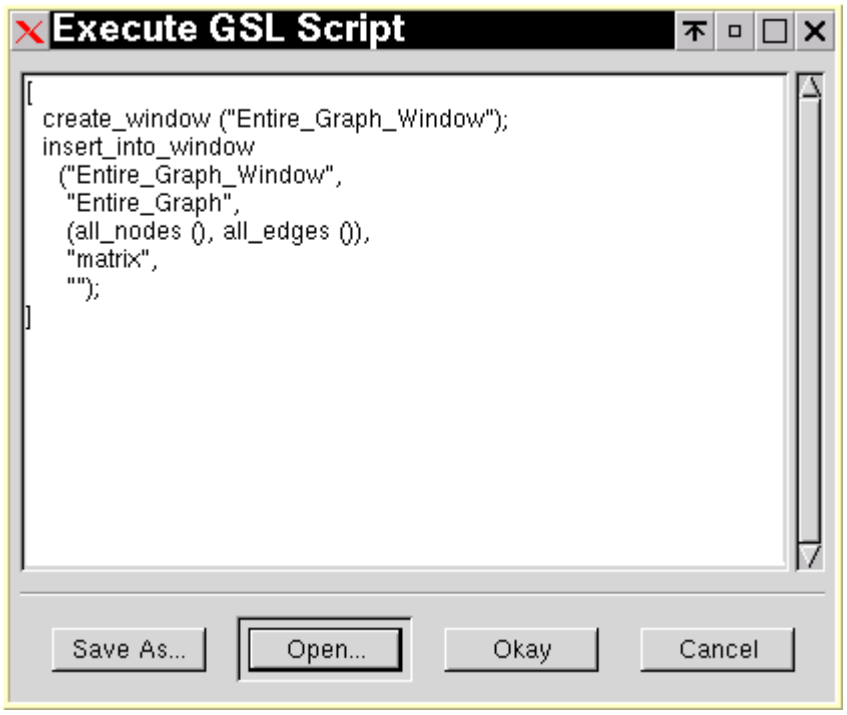


Abbildung 6.4.: Skript-Dialog

Der Skriptdialog (siehe Abbildung 6.4) hat zuoberst ein Textfeld, in das ein GSL Skript eingegeben werden kann.

Unter Query Text befinden sich folgende Buttons:

1. Open...
   Mit diesem Button kann ein auf Festplatte gespeichertes Skript geladen werden. Nach Klick auf diesen Button erscheint ein Dateiauswahlfenster, nach Auswahl eines Skriptes wird dieses in das Textfeld eingefügt.
2. Save As...
   Mit diesem Button kann das im Textfeld eingetippte Skript gespeichert werden. Nach dem Klick auf diesen Button erscheint ein Dateiauswahlfenster, nach Festlegen eines Dateinamens wird das Skript unter diesem Namen gespeichert.
3. Okay
   Führt das eingegebene Skript aus.
4. Cancel Schließt das Fenster ohne Ausführung oder Speicherung des Skriptes.

6.6. Allgemeiner Texteingabedialog

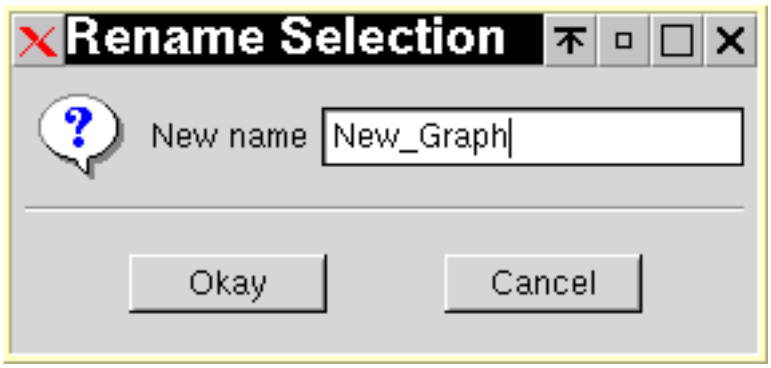


Abbildung 6.5.: Dialog-Window

Ein allgemeiner Texteingabedialog ist ein Fenster Dialog Window (siehe Abbildung 6.5) mit dem Titel „Input“. Im Fenster ist ein einzeiliges Textfeld mit einem Prompt-Text, zwei Buttons mit den Beschriftungen „OK“ und „Cancel“.

6.7. Set-Operation-Dialog

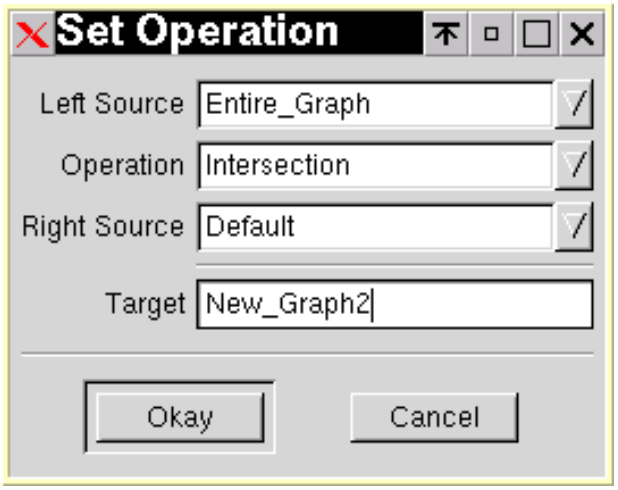


Abbildung 6.6.: Set-Operation-Dialog

Der Set-Operation-Dialog (siehe Abbildung 6.6) dient dazu, Mengenoperationen über Selektionen und IML-Teilgraphen durchzuführen. Die Mengenoperation lässt sich wie folgt beschreiben:
TARGET := LEFT_SOURCE <Operation> RIGHT_SOURCE

Bestandteile des Dialoges sind:

1. Left Source Combobox
   Soll eine Mengenoperation für Selektionen ausgeführt werden, so kann hier eine der Selektio-

nen 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. Operation 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.

6.8. Layoutalgorithmen Dialog

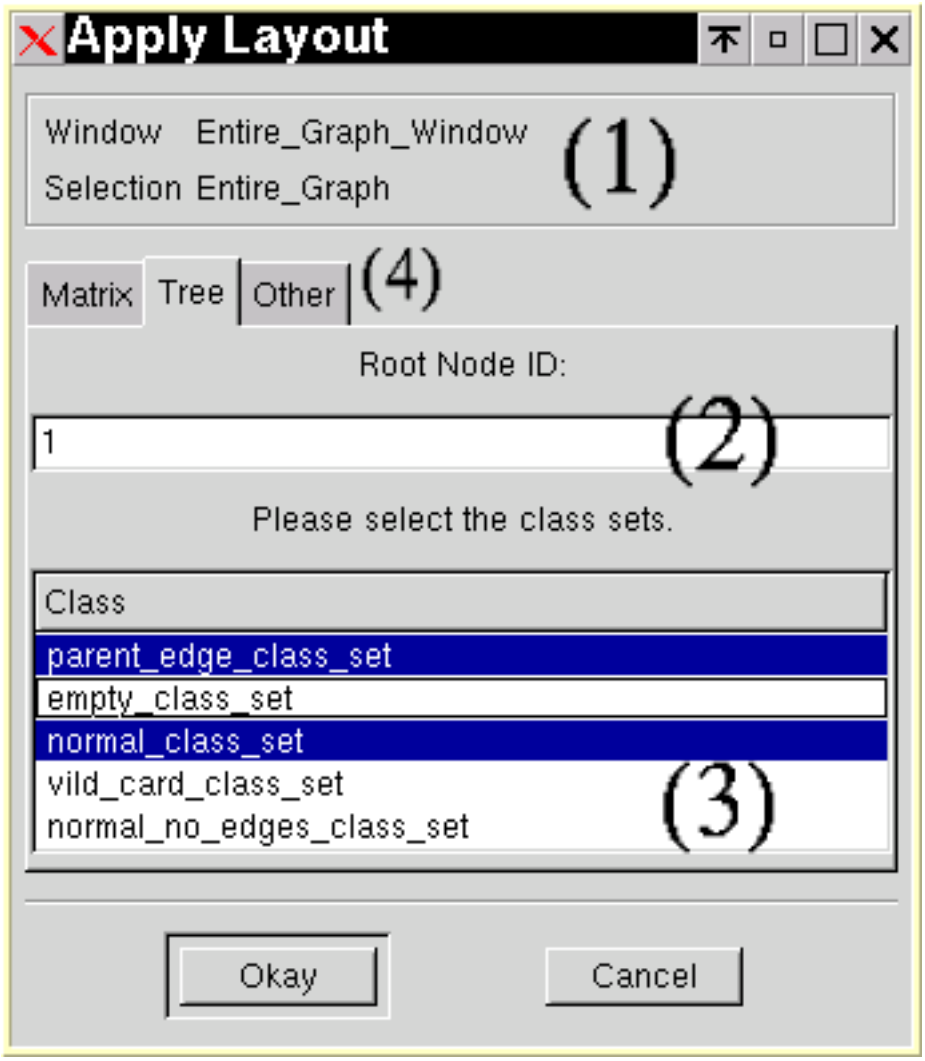


Abbildung 6.7.: Layout-Algorithmen-Dialog

Der Layoutalgorithmen Dialog (siehe Abbildung 6.7) wird über Popup-Menüs auf Selektionen aufgerufen und bietet folgende Möglichkeiten:

1. Anzeige der zu layoutenden Selektion mit dem zu layoutenden Fenster (1)

2. Auswahl des Layoutalgorithmus
   Über die Tabs (4) im Fenster kann der Layoutalgorithmus ausgewählt werden, in der aktuellen Version stehen zur Auswahl: Matrix, Tree Layout, Other (für spätere Erweiterungen).
   Das Tree-Layout ordnet die Knoten als Baum an, dessen Wurzel anhand der ID des Knotens (welche u.a. mit dem Knoten-Informationsfenster, siehe 6.4, festgestellt werden kann) festgelegt wird.
   Das Matrix-Layout ordnet die Knoten in einer Matrix an.

3. Ggf. Eingabefeld für die ID des Wurzelknotens bei Treelayouts (2)
4. Ggf. Auswahl der für das Layout der Knoten zu berücksichtigenden Klassenmengen (3) (class sets, siehe 9.3.6 und 4.1) bei semantischen Tree-Layouts.
   Nur Klassenmengen, die in der Liste per Mausklick markiert worden sind (Mehrfachmarkierungen sind möglich), werden für das Layout berücksichtigt. Falls keine Klassenmengen angeben wurden, werden alle Kanten, *auch diejenigen, die nicht im Fenster sichtbar sind,* für das Layout verwendet.
5. OK Button
6. Cancel Button

Nach dem Klicken von OK ist mit der linken Maustaste in ein Fenster auf den dargestellten Anzeigeinhalt zu klicken (an einer leere Stelle), an dieser Position wird dann die neu layoutete Selektion eingefügt.

6.9. Dateneingabe

Beschreibung von verschiedenen GUI Elementen zur Eingabe von Daten.

6.9.1. Knoten-Annotations-Dialog

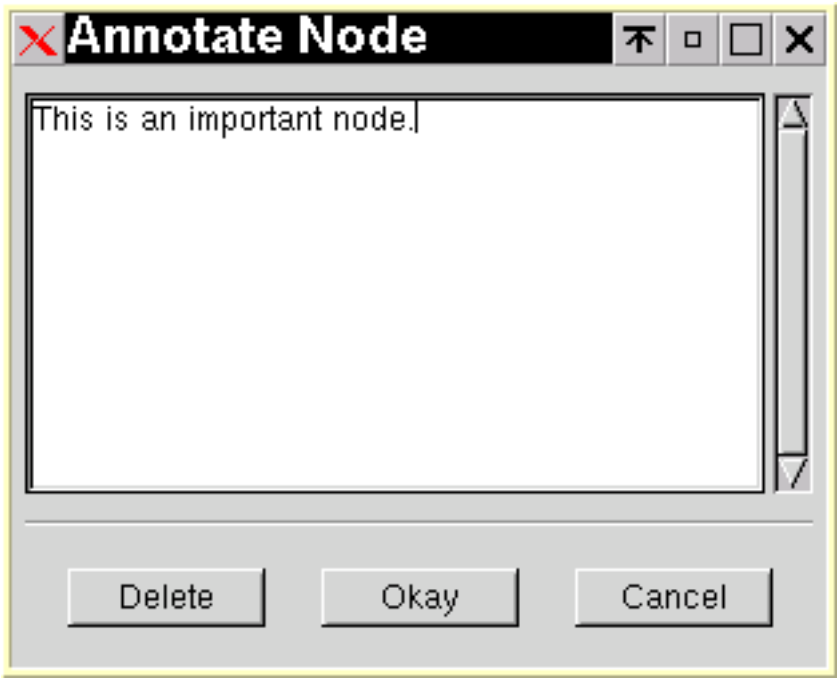


Abbildung 6.8.: Node-Annotation-Dialog

Der Knoten-Annotations-Dialog (siehe Abbildung 6.8) besteht aus einem Fenster mit einem Label, welches die ID des Knotens anzeigt, und einem Textfeld für den Annotationstext. Mittels der beiden Buttons Okay und Cancel können die Änderungen im Textfeld übernommen oder verworfen werden. Der Button Delete kann dazu verwendet werden, das Textfeld zu leeren.

6.9.2. Platz Schaffen-Dialog

Im Dialogfenster Make Room (siehe Abbildung 6.9) 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 Okay kann bestätigt werden, mit dem Button Cancel abgebrochen werden.

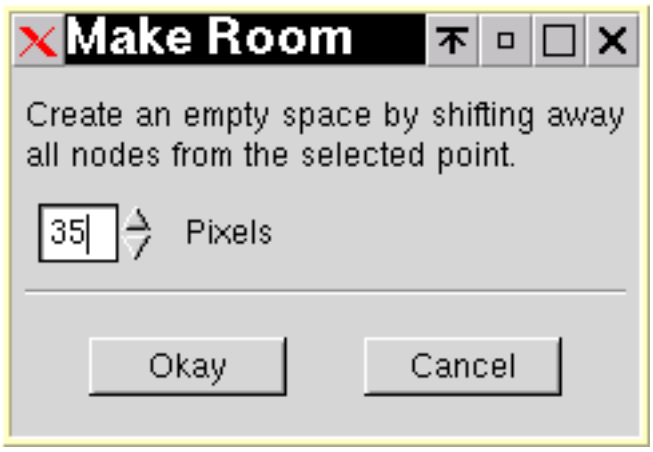


Abbildung 6.9.: Make-Room-Dialog

6.10. Ausgabe von Fehlermeldungen

Abbildung 6.10.: Error-Window

Im Fenster eines allgemeinen Fehlerdialoges (siehe Abbildung 6.10) befindet sich die Fehlermeldung und ein „OK“-Button, mit dem das Fenster geschlossen werden kann..

6.11. Sicherheitsabfrage

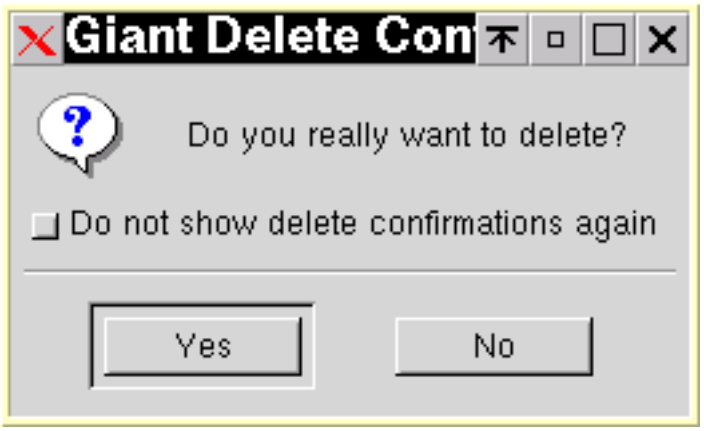


Abbildung 6.11.: Confirmation-Window

Eine allgemeine Sicherheitsabfrage (siehe Abbildung 6.11) besitzt einen Fragetext und die zwei Buttons „Yes“ und „No“. Ggf. kann mit einem Radiobutton festgelegt werden, daß Sicherheitsabfragen des jeweiligen Types in Zukunft nicht mehr angezeigt werden (z.B. „Do not show delete confirmations again“).

6.12. Auswahl von Dateien

6.12.1. Der „Standard-Filechooser-Dialog“

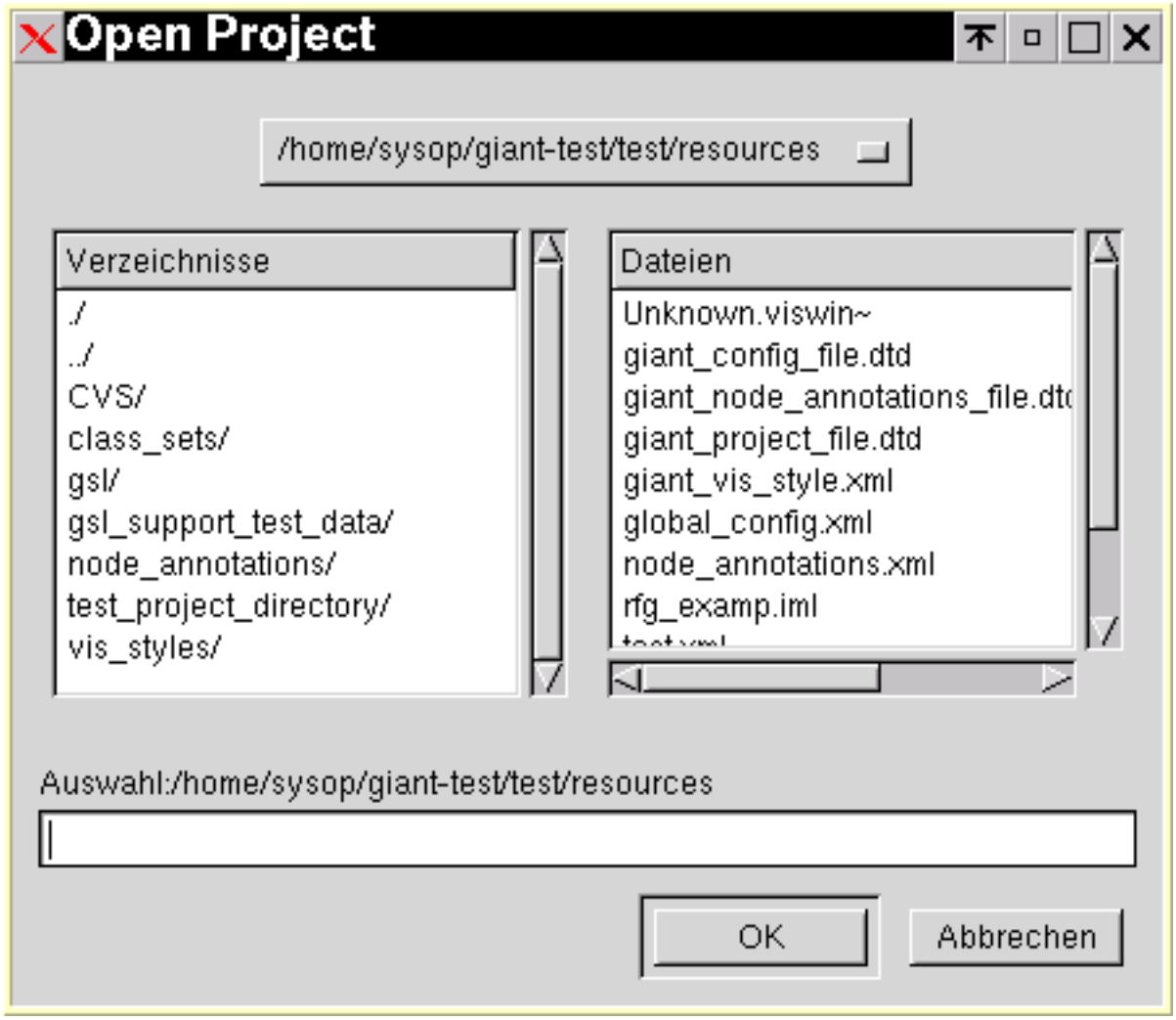


Abbildung 6.12.: Filechooser-Window

Die Auswahl von Dateien (Laden/Speichern) passiert bei GIANT mittels des Standard-Filechooser-Dialogs (siehe Abbildung 6.12). Dieser Dialog ist Bestandteil der Widget-Bibliothek von GTK.

6.13. Fadenkreuz-Cursor

Der Cursor verwandelt sich nach Auswahl bestimmter Funktionen (siehe z.B. Funktion 6.9.2) in ein Fadenkreuz. Durch Linksklick auf einen Fenster-Knoten oder eine Fenster-Kante in einem Anzeigefenster 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.

7. GIANT Scripting Language

Die Fähigkeiten der GIANT Scripting Language (GSL) werden im Anhang „GSL“, der Teil der GIANT Spezifikation ist, beschrieben.

7.1. Parameter der Layoutalgorithmen

Bei der Funktion `insert_into_window` auf Seite 34 der GSL-Spezifikation ist als Parameter `parameters` für den gewählten Layoutalgorithmus zu übergeben. Diese werden 1:1 dem Parameter `additional_parameters` der Routine `Create` in `src/vis/giant-layout_factory.ads` übergeben. Die möglichen Parameter sind dort dokumentiert.

Für das Matrix-Layout sind keine Parameter vorgesehen.

Für das Treelayout sind es folgende (Auszug aus dem Kommentar):

```
Format:  [<Root_Node_ID>]; <List_Of_Class_Set_Names>[; Reverse_Edges]
Example: "5; Aber, Hallo"
         "; Ja, genau"
         It is not possible to use " or ; or , in a classsetname
         Brackets are possible.

Meaning:
  Root_Node_Id    : The root-node of the tree to layout
                    If not given, the root-node is searched
                    If there's more than one possible root-node-id,
                    the result is random.
  Class_Set_Names : Names of ClassSet containing node-classes
                      and edge-classes to layout
  Reverse_Edges   : If given, edges are reversed
    Normally, an outgoing edge indicates the target to be the child.
    If this parameter is given, the source of an incoming edge is regarded
    as child
    Useful for parent-edges
```

7.2. GSL-FAQ

Hier finden sich einige oft gestellte Fragen zur GSL.

7.2.1. Wie starte ich GSL?

Zwei Möglichkeiten:

- Über Kommandozeile
- Über Query-Dialog

Darüberhinaus gibt es noch die Möglichkeit, vorgefertige Skripte über die Menüs zu starten.

7.2.2. Mein Script bei if funktioniert nicht als condition

So ist das auch nicht spezifiziert. Abhilfe:

```
// ...
+start_cond;
if
  (is_script (b),
   {() set ('start_cond, b ())},
   {() set ('start_cond, b   )});
if
  (start_cond,
  // ...
```

Wichtig ist hier, dass beide Zweige als Skripte implementiert sind. Siehe auch Frage 7.2.3.

7.2.3. If führt immer beide Zweige aus

Beispiel:

```
if
  (true,
   set ('a, 10),
   set ('a, 20))
```

a ist jetzt 20 und nicht 10.

Der Grund liegt in der Auswertung: if ist auch ein GSL-Skript. Um ein Skript auszuführen, werden zuerst die Parameter für dieses Skript interpretiert und das Ergebnis dieser Interpretation dem Skript übergeben. Bei obigem Beispiel sind das true, 10 und 20, da die Skripte `set ('a, 0)`1 und `set ('a, 20)` „ausgeführt“ wurden. Nachdem true true ist, wird 10 ausgewertet, was zu 10 führt.

Richtig ist deshalb:

```
if
  (true,
   {() set ('a, 10)},
   {() set ('a, 20)})
```

Hier in den Zweigen nach der Auswertung und vor der „Ausführung“ von if Aktivierungsinformationen liegen. Und so wird nach der Entscheidung, welcher Zweig ausgewertet werden soll, die Aktivierungsinformationen zu Scripten ausgewertet.

7.2.4. Wie erzeuge ich einen neuen Subgraphen?

Folgendes Script erzeugt einen leeren Subgraphen NAME

```
[
  set (+target_edges, empty_edge_set ());
  set (+target_nodes, empty_node_set ());

  set
    ('subgraph.NAME,

     (target_nodes,
      target_edges));
]
```

8. 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. Soweit für das Verständnis von GIANT nötig, sind auch Interna des Programms festgehalten.

8.1. Projekte in GIANT

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.

Wird während des Betriebs von GIANT ein neues Projekt angelegt, so erhält es automatisch eine Projektdatei. 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.

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).

Der Benutzer kann beliebig viele Projekte anlegen. In GIANT kann immer nur ein Projekt gleichzeitig geöffnet sein.

Während der Arbeit mit GIANT kann ein vorhandenes Projekt geladen oder ein neues Projekt angelegt werden.

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 8.1.3) modifizieren und so dem Projekt z.B. neue Verwaltungsdateien (siehe 8.1.7, 8.1.5 und 8.1.6) 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, es ist also bei Manuellen Änderungen mit großer Sorgfalt vorzugehen.

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.

8.1.1. 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 8.1.3).
2. Verwaltungsdateien für Anzeigefenster.
   Jedes zum Projekt gehörende Anzeigefenster hat, sobald es zum ersten mal durch Benutzen der „Project - Save“ oder „Save As..“ Funktion im Pulldown Menü des Hauptfensters gespeichert wurde, seine eigene Verwaltungsdatei. Siehe hierzu auch 8.1.5.
3. Verwaltungsdateien für IML-Teilgraphen.
   Jeder IML-Teilgraph des Projektes hat seine eigene Verwaltungsdatei (siehe 8.1.6). 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 („Project - Save“ oder „Save As..“ im Pulldown Menü des Hauptfensters).
4. Genau einer Verwaltungsdatei für alle Knoten-Annotationen (siehe 8.1.7).

8.1.2. Das Projektverzeichnis

GIANT speichert alle Verwaltungsdateien (siehe 8.1.7, 8.1.5 und 8.1.6) automatisch im Projektverzeichnis.

In einem Projektverzeichnis darf nur ein Projekt (insb. nur eine Projektdatei) abgelegt werden.

8.1.3. Die Projektdatei

Die Projektdatei liegt als XML-Datei vor. Sie befindet sich im Projektverzeichnis und enthält Informationen, die zur Identifikation des zu einem Projekt gehörenden IML-Graphen nötig sind.

Der Name der Projektdatei entspricht dem Namen des Projektes.

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.

Der Pfad zu der Datei, die den IML-Graphen enthält, ist in der Projektdatei gespeichert.

8.1.4. 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. Die IML-Graph Datei muß unbedingt zu dem IML-Graphen passen, der dem Projekt zugrundeliegt.

8.1.5. Verwaltungsdateien für Anzeigefenster

Die Verwaltungsdateien für Anzeigefenster sind Binärdateien. Zu jedem Anzeigefenster gibt es eine Verwaltungsdatei. 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 Anzeigeinhalt (alle visualisierten Fenster-Knoten und Fenster-Kanten mit Position).
2. Der Zustand (Zoomstufe und Position) des sichtbaren Anzeigeinhaltes.
3. Alle Pins (gespeicherte Zustände des sichtbaren Anzeigeinhaltes).
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 9.3).

Der Name des für das Anzeigefenster gewählten Visualisierungsstils wird zwar gespeichert, da die Visualisierungsstile aber völlig unabhängig von den Projekten über XML-Dateien (siehe 9.3) konfiguriert bzw. definiert werden, kann nicht garantiert werden, dass der gewünschte Stil beim Laden des Projektes auch wieder gefunden wird. Sollte der Stil nicht gefunden werden, wird dann ein „Standard-Stil“ verwendet.

8.1.6. Verwaltungsdateien für IML-Teilgraphen

Die Verwaltungsdatei für IML-Teilgraphen ist eine Binärdatei. Für jeden IML-Teilgraphen gibt es eine eigene Verwaltungsdatei. Sämtliche erzeugten IML-Teilgraphen werden in Verwaltungsdateien in binärer Form gespeichert.

8.1.7. Die Verwaltungsdatei für Knoten-Annotationen

Die Verwaltungsdatei für Knoten-Annotationen liegt als XML Datei vor.

Alle Knoten-Annotationen des Projektes werden in genau einer solchen Verwaltungsdatei gespeichert. Weitere Informationen zu dieser Datei befinden sich in Abschnitt 9.3.8.

8.2. Grundlegendes Verhalten von GIANT beim Speichern von Projekten

8.2.1. „Project - Save / Project - Save As.. Pulldown-Menü“

Diese Funktionalität wird von entsprechenden Funktionen 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 „Project - Save (As..)“ exakt dem Zustand des innerhalb von GIANT geöffneten Projektes.

8.2.2. Persistenz von Anzeigefenstern

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.

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.

Sämtliche dem Projekt bekannten Anzeigefenster (alle Anzeigefenster zu denen es eine entsprechende Verwaltungsdatei gibt) werden in der Liste über die Anzeigefenster angezeigt.

Zu jedem Anzeigefenster eines Projektes gibt es eine Verwaltungsdatei. Bei neu erzeugten Anzeigefenstern wird diese Verwaltungsdatei beim ersten Speichern angelegt.

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). 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).

8.2.3. Persistenz von IML-Teilgraphen

Alle in einem Projekt bereits vorhandenen IML-Teilgraphen werden in einer entsprechenden Liste angezeigt. Neu erzeugte IML-Teilgraphen und Änderungen an bestehenden IML-Teilgraphen können nur über „Project - Save (As..)“ (siehe oben) gespeichert werden.

Wird das Programm beendet, ohne das zuvor „Project - Save (As..)“ 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.

Modifikationen an bestehenden IML-Teilgraphen werden nicht automatisch gespeichert. Zu neu erzeugten IML-Teilgraphen wird nicht automatisch eine Verwaltungsdatei erzeugt.

IML-Teilgraphen können gelöscht werden. Falls vorhanden, wird dann auch die entsprechende Verwaltungsdatei ebenfalls sofort gelöscht.

8.2.4. 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 „Project - Save (As..)“ (siehe 8.2.1) in die Verwaltungsdatei für Knoten-Annotationen (siehe 8.1.7) 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.

9. Konfiguration von GIANT

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

9.1. Allgemeines

Die Konfiguration von GIANT wird in XML-Dateien vorgenommen.

Es gibt eine „globale Konfigurationsdatei“, die sich im GIANT-Stamm-Verzeichnis
unter etc/global_config.xml befindet.

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, shared/styles/. Die Datei des standardmäßig verwendeten Visualisierungsstils muss existieren, sie heisst Default.xml.

Die unter Punkt 9.1 beschriebenen Dateien zu 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.

9.2. Die globale Konfigurationsdatei

Diese Datei ist in XML verfasst. In Ihr können die anschließend beschriebenen Einstellungen vorgenommen werden. Die Datei heisst global_config.xml und befindet sich unter etc/ im GIANT Verzeichnis.

Hier das File giant_config_file.dtd:

```
<!ELEMENT giant_config_file (absolute_path_root?, (setting)*)>

  <!ELEMENT absolute_path_root EMPTY>
  <!ATTLIST absolute_path_root
    root_directory  CDATA #REQUIRED
  >

  <!ELEMENT setting EMPTY>
  <!ATTLIST setting
    name   CDATA #REQUIRED
    value  CDATA #REQUIRED
  >
```

Generell hat in dieser Datei jede Einstellung einen setting-Knoten mit name und value.
Im Folgenden sind die möglichen Einstellungen, die im Config-File sein können, aufgelistet, jeweils mit ihrem Defaultwert. Wenn die betreffenden Daten im Config-File nicht existieren, so wird der im Folgenden als Beispiel angegebene Wert von GIANT als default verwendet. Farben können entweder im Format `#XXYYZZ` (wobei XX , YY und ZZ jeweils die Rot- Grün- und Blau-Anteile zweistellig Hexadezimal angeben) oder mit den GTK bekannten Namen (z.B. `red`) angegeben werden.

Einstellungsdaten, die einen Pfad darstellen, müssen mit einem Pfadseperator (z.B: / bei Linux) enden. Bei relativen Pfaden versucht GIANT, den Pfad nacheinander bis zum Erfolg wie folgt durch Pfad-Erweiterung zu finden:

1. Erweiterung um das root_directory Attribut des <absolute_path_root> - Knotens aus dem Config-File.
2. Wenn kein absolute_path_root - Knoten im Config-File existiert oder kein gültiger Pfad zusammengesetzt werden konnte, so wird versucht, vom Directory aus, in dem sich das Config-File befand, die Datei bzw. das Directory zu finden.

Das folgende Directory gibt an, wo sich das GIANT-Verzeichnis der Installation von GIANT mit allen Unterverzeichnissen befindet. Wird dieses weggelassen, wird das Verzeichnis, in dem sich der Nutzer in der Shell des Betriebssystemes bei Aufruf von GIANT befindet, verwendet.

```
<absolute_path_root root_directory  = "." />
```

Das folgende Directory gibt an, in welchem Verzeichnis die Bilder von GIANT gesucht werden, wenn andere Pfad-Erweiterungen (Expansionen) fehlschlagen:

```
<setting name = "Resources_Directory" value="." />
```

Das folgende Icon wird verwendet, graphisch um anzuzeigen, daß ein Knoten annotiert ist:

```
<setting name = "Icon_For_Node_Annotations" value="my_icon.xpm" />
```

Die Farbe der aktuellen Selektion:

```
<setting name = "Current_Selection_Highlight_Color" value="#AAAAAA" />
```

Die Farbe Nr. 1 für die Hervorhebung von Selektionen:

```
<setting name = "Selection\_Highlight_Color_1" value="#AAAAAA" />
```

Die Farbe Nr. 2 für die Hervorhebung von Selektionen:

```
<setting name = "Selection_Highlight_Color_2" value="#AAAAAA" />
```

Die Farbe Nr. 3 für die Hervorhebung von Selektionen:

```
<setting name = "Selection_Highlight_Color_3" value="#AAAAAA" />
```

Die Farbe Nr. 1 für die Hervorhebung von IML-Teilgraphen:

```
<setting name = "IML_Subgraph_Highlight_Color_1" value="#AAAAAA" />
```

Die Farbe Nr. 2 für die Hervorhebung von IML-Teilgraphen:

```
<setting name = "IML_Subgraph_Highlight_Color_2" value="#AAAAAA" />
```

Die Farbe Nr. 3 für die Hervorhebung von IML-Teilgraphen:

```
<setting name = "IML_Subgraph_Highlight_Color_3" value="#AAAAAA" />
```

Hier kann eine Menge von Pfaden, ggf. getrennt durch den Path Seperator des Betriebssystems, angegeben werden, diese Pfade geben die Positionen der vordefinierten GSL Skripte an, eingesetzt werden kann hier z.B. shared/gsl/ :

```
<setting name = "GSL.Include_Paths" value="." />
```

Hier kann die Höhe des Hauptfensters angegeben werden:

```
<setting name = "Main_Window.Height" value="400" />
```

Hier kann die Breite des Hauptfensters angegeben werden:

```
<setting name = "Main_Window.Width" value="200" />
```

Hier kann angegeben werden, nach wieviel Pixeln der untere Bereich des Hauptfensters mit der unteren Liste anfängt:

```
<setting name = "Main_Window.Separator" value="230" />
```

Hier kann festgelegt werden, ob bei Löschoperationen eine Sicherheitsabfrage erfolgen soll:

```
<setting name = "Confirm.Delete" value="true" />
```

Hier kann der Editor zum Anzeigen der Quelltexte angegeben werden, mit seinem Aufruf. %l wird dabei von GIANT beim Aufruf der Funktion zum Anzeigen des Quelltextes durch die Zeilennummer ersetzt, %c durch die Spaltennummer und %f durch den Dateinamen der betreffenden Datei. So kann bei den meisten Editoren dafür gesorgt werden, daß durch den Aufruf die Datei geladen wird und der Editor mit dem Cursor an die richtige Stelle in der Datei springt. GIANT spawned einen neuen Prozeß, in dem der Editor abläuft.

```
<setting name = "Editor.Source" value="/usr/bin/emacs +%l:%c %f" />
```

Nun folgen einige Einstellungen zu den Kontextmenüs von GIANT. Mit den folgenden Einstellungen können die GSL-Skripte, die über die Kontext ausgewählten Knoten/Kante/Subgraph angestoßen werden können, festgelegt werden. Jedem Skript wird dabei ggf. auch ein Parameter übergeben.

Die Skripte werden anhand ihres Dateinamens (ggf. auch mit Leerzeichen im Dateinamen) mit der hinzugefügten Endung .gsl in den weiter oben schon definierten GSL.Include_Paths gesucht.

Bei Angabe von mehreren Skripten sind diese durch das Zeichen : voneinander zu trennen, dies gilt für alle folgenden „GSL.“-Knoten

Mit der folgenden Zeile kann festgelegt werden, welche GSL-Scripts im Pulldown-Menü „Script“ des GIANT-Hauptfensters auftauchen. Diese Scripts werden beim Auswählen des jeweils dazugehörigen Menüpunktes ohne Parameter gestartet.

```
<setting name = "GSL.No_Param" value="" />
```

Mit der folgenden Zeile kann festgelegt werden, welche GSL-Scripts im Kontext-Menü bei Drücken der rechten Maustaste, wenn der Mauszeiger sich im Inhalt des Anzeigefensters, jedoch nicht über einem Knoten, befindet, ausgeführt werden. Jedes Script ergibt eine mögliche Auswahl im Kontextmenü.

Diese Scripts werden beim Auswählen des jeweils dazugehörigen Menüpunktes jeweils ohne Parameter gestartet.

```
<setting name = "GSL.No_Param_Context" value="" />
```

Mit der folgenden Zeile kann festgelegt werden, welche GSL-Scripts im Kontext-Menü bei Drücken der rechten Maustaste, wenn der Mauszeiger sich im Inhalt des Anzeigefensters über einem Knoten, befindet, ausgeführt werden. Jedes Script ergibt eine mögliche Auswahl im Kontextmenü.

Den Scripts wird dabei die ID des Knotens als Parameter übergeben.

```
<setting name = "GSL.Node_Id_Param" value="" />
```

Mit der folgenden Zeile kann festgelegt werden, welche GSL-Scripts im Kontext-Menü bei Drücken der rechten Maustaste, wenn der Mauszeiger sich im Inhalt des Anzeigefensters über einer Kante befindet, ausgeführt werden. Jedes Script ergibt eine mögliche Auswahl im Kontextmenü.

Den Scripts wird dabei die ID der Kante als Parameter übergeben.

```
<setting name = "GSL.Edge_Id_Param" value="" />
```

Mit der folgenden Zeile kann festgelegt werden, welche GSL-Scripts im Kontext-Menü bei Drücken der rechten Maustaste, wenn der Mauszeiger sich in der Liste der IML-Teilgraphen („Subgraph“-Liste) im GIANT-Hauptfenster befindet, ausgeführt werden. Jedes Script ergibt eine mögliche Auswahl im Kontextmenü.

Den Scripts wird dabei der IML-Teilgraph als Parameter übergeben.

```
<setting name = "GSL.Subgraph_Param" value="" />
```

Mit der folgenden Zeile kann festgelegt werden, welche GSL-Scripts im Kontext-Menü bei Drücken der rechten Maustaste, wenn der Mauszeiger sich in der Liste der Selektionen („Selections“) in einem GIANT-Anzeigefenster befindet, ausgeführt werden. Jedes Script ergibt eine mögliche Auswahl im Kontextmenü.

Den Scripts wird dabei die Selektion als Parameter übergeben.

```
<setting name = "GSL.Selection_Param" value="" />
```

Hier nun ein Beispiel-Configfile (global_config.xml). Wie man sieht, müssen nicht alle möglichen Einstellungen zugewiesen werden. Wo eine Zuweisung fehlt, werden die oben aufgelisteten Defaultwerte eingesetzt.

```
<?xml version="1.0" encoding="ISO-8859-1" ?>
<!DOCTYPE giant_config_file
  SYSTEM "giant_config_file.dtd">

<giant_config_file>

  <absolute_path_root root_directory  = "resources_dir" />

  <setting name="Resources_Directory" value=".." />

  <setting name  = "Icon_For_Node_Annotations"
           value = "annotations_icon_user.xpm" />

  <setting name="Actual_Selection_Highlight_Color" value="RGB:00/00/80" />

  <setting name="Selection_Highlight_Color_1" value="RGB:FF/00/00" />
  <setting name="Selection_Highlight_Color_2" value="RGB:00/FF/00"/>
  <setting name="Selection_Highlight_Color_3" value="RGB:00/00/FF" />

  <setting name="IML_Subgraph_Highlight_Color_1" value="RGB:00/FF/FF" />
  <setting name="IML_Subgraph_Highlight_Color_2" value="RGB:FF/00/FF" />
  <setting name="IML_Subgraph_Highlight_Color_3" value="RGB:FF/FF/00" />

  <setting name="GSL.Include_Path" value="./gsl" />
</giant_config_file>
```

9.3. 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 6.3.5).

Für die Visualisierungsstile gelten die folgenden Konventionen:

1. Für jeden Visualisierungsstil gibt es eine entsprechende XML-Datei geben, die Visualisierungsstil-XML-Dateien befinden sich im GIANT-Verzeichnis unter shared/styles.
2. Ein Visualisierungsstil beschreibt, wie die Knoten und Kanten des IML-Graphen innerhalb eines Anzeigefensters graphisch dargestellt werden.
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. Vererbung im IML-Baum kann zur leichteren Auswahl von Knoten- und Kantenklassen benutzt werden.

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.

9.3.1. Name des Visualisierungsstils

Jeder Visualisierungsstil erhält einen Namen. Unter diesem Namen ist der Visualisierungsstil für jedes Anzeigefenster einzeln auswählbar. Der Name des Visualisierungsstil ist der Dateiname der XML-Datei ohne Endung .xml.

9.3.2. Generelle Einstellungen innerhalb des Visualisierungsstils

1. Die Hintergrundfarbe im Anzeigefenster.

9.3.3. 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“.

9.3.4. 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“.

9.3.5. Config-Dateien für Visualisierungsstile

Jeder Visualisierungsstil wird durch eine XML-Datei beschrieben. Der Dateiname sollte dem Namen des Visualisierungsstils mit der Endung .xml entsprechen.

GIANT lädt zuerst die im GIANT_VIS_DIRECTORY (dieses ist in der globalen Config-Datei definiert) gespeicherten Visualisierungsstile, dann die Visualisierungsstile, die sich im User_VIS_DIRECTORY befinden.
Bei Namensgleichheit zweier solcher XML-Dateien in beiden Verzeichnissen wird nur die Datei im User_VIS_DIRECTORY beachtet, genauer gesagt ersetzen diese Einstellungen die zunächst aus GIANT_VIS_DIRECTORY geladenen Einstellungen.
Wird bei der Auswahl von Knoten oder Kanten durch Wildcards oder Namen eine oder mehrere Kanten doppelt definiert, so gilt die zuletzt festgelegte Definition.

Der Aufbau der XML-Dateien ist wie folgt durch die Datei giant_vis_style_file.dtd spezifiziert:

```
<!ELEMENT giant_vis_style_file
  (global_settings,
   default_node_setting, (node_class_specific_setting)*,
   default_edge_setting, (edge_class_specific_setting)*)>

  <!ELEMENT global_settings EMPTY>
  <!ATTLIST global_settings
    vis_window_background_color CDATA #REQUIRED
  >

  <!ELEMENT default_node_setting (node_attribute)*>
  <!ATTLIST default_node_setting
    icon         CDATA #REQUIRED
    text_color   CDATA #REQUIRED
    border_color CDATA #REQUIRED
```

```
    fill_color   CDATA #REQUIRED
  >

  <!ELEMENT node_class_specific_setting
    ((node_attribute)*, (node_class)*, (super_node_class)*)>
  <!ATTLIST node_class_specific_setting
    icon         CDATA #REQUIRED
    text_color   CDATA #REQUIRED
    border_color CDATA #REQUIRED
    fill_color   CDATA #REQUIRED
  >

    <!ELEMENT node_attribute EMPTY>
    <!ATTLIST node_attribute
      attribute_name CDATA #REQUIRED
    >

    <!-- A node class that is affected by this setting. -->
    <!ELEMENT node_class EMPTY>
    <!ATTLIST node_class
      node_class_name CDATA #REQUIRED
    >

    <!ELEMENT super_node_class EMPTY>
    <!ATTLIST super_node_class
      super_node_class_name CDATA #REQUIRED
    >

  <!ELEMENT default_edge_setting EMPTY>
  <!ATTLIST default_edge_setting
    line_color CDATA                                        #REQUIRED
    text_color CDATA                                        #REQUIRED
    line_style (continuous_line|dotted_line|dashed_line) #REQUIRED
    show_label (yes|no)                                     #REQUIRED
  >

  <!ELEMENT edge_class_specific_setting ((edge_class)*, (super_edge_class)*)>
  <!ATTLIST edge_class_specific_setting
     line_color CDATA                                        #REQUIRED
     text_color CDATA                                        #REQUIRED
     line_style (continuous_line|dotted_line|dashed_line) #REQUIRED
     show_label (yes|no)                                     #REQUIRED
  >

    <!ELEMENT edge_class EMPTY>
    <!ATTLIST edge_class
      start_node_class CDATA #REQUIRED
```

```
    attribute_name   CDATA #REQUIRED
  >

  <!ELEMENT super_edge_class EMPTY>
  <!ATTLIST super_edge_class
    super_start_node_class CDATA #REQUIRED
    attribute_name         CDATA #REQUIRED
  >
```

Hier ein konkretes Beispiel einer solchen Datei (Beispiel 1):

```
<!--  Beginn des Beispiels 1 -->
```

In jeder Datei wird die Hintergrundfarbe durch
`<global_settings vis_window_background_color="#202020"/>`
festgelegt.

Jede Knotendefinition, ob für einen oder mehrere Knoten, besteht aus folgenden Elementen:

- icon
  Der Dateiname des Icons, durch das der Knoten dargestellt wird
- text_color
  Die Farbe des Textes
- border_color
  Die Farbe des Randes des Knoten
- fill_color
  Die Farbe des Hintergrundes des Knoten

Alle Knoten, die nicht später behandelt werden, werden durch

```
<default_node_setting>
  icon         = "test_node_icon_default_1.xpm"
  text_color   = "#AAAAAA"
  border_color = "#AAAAAA"
  fill_color   = "white">
</default_node_setting>
```

genauer festgelegt.

Im Folgenden können einzelne Knoten vom default_node_setting abweichend festgelegt werden. Dies geht durch folgendes Konstrukt.

```
<node_class_specific_setting
  icon         = "test_node_icon_blue_1.xpm"
  text_color   = "blue"
  border_color = "blue"
  fill_color   = "white">
```

Bei der Suche nach in Config-Files mit dem Dateinamen angegebenen Dateien gelten die in Abschnitt 9.2 genannten Grundsätze.

Dies sind soweit die schon bekannten Einstellungen zur Darstellung. Nun kommt noch optional dazu, welche Attribute darzustellen sind. Hier können auch mehrere Attribute angegeben werden.

```
  <node_attribute attribute_name  = "SLoc"/>
```

Es wird hier also das Attribut SLoc im Knoten sichtbar dargestellt.

Nun können wir festlegen, für welche Knoten diese Einstellungen gelten. Die Einstellungen gelten für jegliche Knoten, die im Folgenden durch eine oder mehrere Beschreibungen getroffen werden (ODER-Verknüpfung). Hier zunächst einige Auswahlen anhand des Knotenklassennamens:

```
  <node_class     node_class_name = "IML_Root"/>
  <node_class     node_class_name = "HPGNode"/>
  <node_class     node_class_name = "OC_Parameter"/>
  <node_class     node_class_name = "TC_Boolean"/>
```

Mittels
`  <super_node_class super_node_class_name = "T_Node"/>`
können alle Knoten, die im IML-Baum unterhalb des angegebenen Knoten liegen (und der Knotentyp selbst), in die Auswahl mit aufgenommen werden. Hierdurch kann u.U. das File kurz und übersichtlich gehalten werden.

Nun ist der <node_class_specific_setting> Block zuende und wird mittels
`</node_class_specific_setting>`
geschlossen.

Es können weitere solche Blöcke folgen, z.B. wie folgt:

```
<node_class_specific_setting
  icon         = "test_node_icon_red_1.xpm"
  text_color   = "red"
  border_color = "red"
  fill_color   = "white">

  <node_attribute attribute_name  = "SLoc"/>
  <node_attribute attribute_name  = "Type_Size"/>
  <node_class     node_class_name = "TC_Boolean"/>
</node_class_specific_setting>
```

Nun kommen wir zu den Darstellungen der Kanten. Zunächst wird in einem Block festgelegt, wie alle Kanten, die nicht später festgelegt werden, dargestellt werden.

```
<default_edge_setting
    line_color = "#AAAAA1"
    text_color = "#AAAAA2"
    line_style = "continuous_line"
    show_label = "no" />
```

Mit line_color und text_color werden die Farben der Linien der Kanten und der Beschriftung der Kanten nach dem bekannten RGB-Farbschema oder den GTK-Abkürzungen festgelegt.
show_label legt fest, ob die Beschriftung der Kante angezeigt werden soll.
line_style gibt die Art der Zeichnung der Kanten an.
Zulässig sind hier

- continuous_line für eine durchgezogene Linie
- dotted_line für eine gepunktete Linie
- dashed_line für eine strichlierte Linie

Nun kommen wir zur Festlegung von individuellen Kantendesigns je nach Kantentyp, abweichend von default_edge_setting.

```
<edge_class_specific_setting
  line_color = "blue"
  text_color = "blue"
  line_style = "dotted_line"
  show_label = "yes">
  <edge_class
     start_node_class = "Routine_Call"
     attribute_name   = "*" />
  <edge_class
     start_node_class = "*"
     attribute_name   = "CF_Next" />
</edge_class_specific_setting>
<!--  Ende des Beispiels 1 -->
```

Durch diesen Block werden mit den oben angegebenen Farben und Stil alle Kanten ausgehend von Routine_Call Knoten angezeigt, ausserdem alle Kanten des Typs CF_Next.

Es gibt folgende Möglichkeiten, Kanten festzulegen:

Das im o.G. Beispiel innerhalb eines <edge_class_specific_setting> Blocks verwendete Konstrukt

```
<edge_class
     start_node_class = "O_Node"
```

```
     attribute_name   = "SLoc" />
```

Inkludiert alle Kanten, die von den genannten Knoten ausgehen und genanntes Attribut haben. Dabei ist bei einem der beiden Parameter auch das Zeichen * erlaubt, dies wählt damit alle Knotenklassen bzw. Kantennamen, die dieses Attribut besitzen bzw. von diesem Knoten ausgehen, aus.

Auch die Verwendung von Vererbung ist möglich:

```
<super_edge_class
     super_start_node_class = "T_Node"
     attribute_name   = "Prog_Unit" />
```

Inkludiert alle Kanten, die von den genannten Knoten ausgehen und genanntes Attribut haben.

Ausserdem werden alle passenden Kanten mit aufgenommen, die von allen Knotentypen unterhalb des angegebenen Knotentyps im IML-Baum ausgehen.
Dies wäre also dasselbe, wie wenn für jeden Knotentyp unterhalb von T_Node im IML-Baum ein einzelner edge_class-Eintrag mit dem angegebenen attribute_name Prog_Unit existieren würde. Durch dieses Konstrukt können also ggf. Einträge in der Datei gespart werden.

Hier ist wiederum auch der * Operator verwendbar, z.B.:

```
<super_edge_class
     super_start_node_class = "TC_Boolean"
     attribute_name   = "*" />
```

Beispiel 2:

Hier nun noch ein zweites Beispiel, die einzelnen Elemente wurden schon im ersten Beispiel erklärt.

```
<?xml version="1.0" encoding="ISO-8859-1" ?>
<!DOCTYPE giant_vis_style_file
  SYSTEM "giant_vis_style_file.dtd">

<giant_vis_style_file>

  <global_settings
    vis_window_background_color="RGB:20/20/20"/>

  <default_node_setting
    icon         = "test_node_icon_default_1.xpm"
    text_color   = "black"
    border_color = "black"
    fill_color   = "white">

  </default_node_setting>
```

```
<node_class_specific_setting
  icon         = "test_node_icon_red_1.xpm"
  text_color   = "red"
  border_color = "red"
  fill_color   = "white">

  <node_attribute attribute_name  = "Type_Size"/>
  <node_attribute attribute_name  = "SLoc"/>

  <node_class     node_class_name = "IML_Root"/>
  <node_class     node_class_name = "HPGNode"/>

  <super_node_class super_node_class_name = "T_Node"/>
  <super_node_class super_node_class_name = "Prog_Unit"/>

</node_class_specific_setting>


<default_edge_setting
  line_color = "RGB:00/FF/FF"
  text_color = "RGB:00/FF/FF"
  line_style = "dotted_line"
  show_label = "no" />

<edge_class_specific_setting
  line_color = "green"
  text_color = "green"
  line_style = "dashed_line"
  show_label = "yes">

  <super_edge_class
    super_start_node_class = "Prog_Unit"
    attribute_name   = "Subunits" />

  <super_edge_class
    super_start_node_class = "Op"
    attribute_name   = "*" />

</edge_class_specific_setting>

<edge_class_specific_setting
  line_color = "blue"
  text_color = "blue"
  line_style = "dotted_line"
  show_label = "yes">

<edge_class
```

```
      super_start_node_class = "*"
      attribute_name   = "SLOC" />

  </edge_class_specific_setting>

</giant_vis_style_file>
```

9.3.6. Config-Dateien für Class-Sets (Klassenmengen)

Ein Class Set ist eine durch ein XML-File festgelegte Menge von Knotenklassen und Kantenklassen. Der Dateiname des XML-Files gibt den Namen des Class Sets an.

Hier das giant_class_set_file.dtd File:

```
<!ELEMENT giant_class_set_file
  (associated_node_classes, associated_edge_classes)>

  <!ELEMENT associated_node_classes ((node_class)*, (super_node_class)*)>

    <!ELEMENT node_class EMPTY>
    <!ATTLIST node_class
      node_class_name CDATA #REQUIRED
    >

    <!ELEMENT super_node_class EMPTY>
    <!ATTLIST super_node_class
      super_node_class_name CDATA #REQUIRED
    >

  <!ELEMENT associated_edge_classes ((edge_class)*, (super_edge_class)*)>

    <!ELEMENT edge_class EMPTY>
    <!ATTLIST edge_class
      start_node_class CDATA #REQUIRED
      attribute_name   CDATA #REQUIRED
    >

    <!ELEMENT super_edge_class EMPTY>
    <!ATTLIST super_edge_class
      super_start_node_class CDATA #REQUIRED
      attribute_name         CDATA #REQUIRED
    >
```

Hier ein Beispiel eines GIANT Class Set Files:

Zunächst der Header:

```
<?xml version="1.0" encoding="ISO-8859-1" ?>
<!DOCTYPE giant_class_set_file
  SYSTEM "giant_class_set_file.dtd">


<giant_class_set_file>
```

Nun werden der Menge Knotenklassen hinzugefügt. Knotenklassen werden anhand ihres Namens im IML-Baum identifiziert.

```
  <!-- All node classes being a member of the class set -->
  <associated_node_classes>
```

In den folgenden zwei Zeilen wird die Knotenklasse T_Node hinzugefügt mittels node_class , außerdem mittels super_node_class alle Knoten unterhalb der Klasse Prog_Unit im IML-Baum (und Prog_Unit selbst).

```
    <node_class           node_class_name="T_Node" />
    <super_node_class     super_node_class_name = "Prog_Unit"/>


  </associated_node_classes>
```

Danach werden Kantenklassen hinzugefügt. Eine Kantenklasse ist festgelegt durch ihren Namen (attribute_name) und die Knotenklasse, von der sie ausgeht (start_node_class).

```
  <!-- All edge classes being a member of the class set -->
  <associated_edge_classes>
```

Hier wird die Kantenklasse Parent, abgehend von T_Node Knoten, hinzugefügt.

```
    <edge_class start_node_class="T_Node"
      attribute_name="Parent" />
```

Hier werden alle Kanten, auf die die Wildcard zutrifft, abgehend von der Knotenklasse O_Node hinzugefügt.

```
    <edge_class
      start_node_class="O_Node"
      attribute_name="*" />
```

Hier werden alle Kanten mit Namen Its_Type abgehend von allen Knotenklassen, die eine solche Kante haben, hinzugefügt.

```
    <edge_class
      start_node_class="*"
      attribute_name="Its_Type" />
```

Bei Verwendung von super_edge_class werden außer der bezeichneten Kante des bezeichneten Knoten auch alle gleichnamigen Kanten von Knoten, die sich in der IML-Hierarchie unter dem bezeichneten Knoten befinden, aufgenommen.

```
    <super_edge_class
      super_start_node_class = "Prog_Unit"
      attribute_name         = "Subunits" />
```

Wird bei attribute_name eine Wildcard verwendet, so werden alle Kanten, die zutreffen, vom bezeichneten Knotentyp und von allen erbenden Knoten ausgewählt.

```
    <super_edge_class
      super_start_node_class = "Op"
      attribute_name         = "*" />
  </associated_edge_classes>

</giant_class_set_file>
```

9.3.7. GIANT Project File

Jedes Projekt in GIANT besitzt eine Projektdatei, der Dateiname ist der Projektname erweitert durch die Endung .xml.
Die Datei ist wie folgt aufgebaut:

```
<!ELEMENT giant_project_file (global_data, visualisation_windows, subgraphs)>

  <!ELEMENT global_data EMPTY>
  <!ATTLIST global_data
    iml_graph_file_path        CDATA  #REQUIRED
    iml_graph_checksum         CDATA  #REQUIRED
    node_annotations_file_name CDATA  #REQUIRED
  >

  <!ELEMENT visualisation_windows (a_vis_window_file)*>

    <!ELEMENT a_vis_window_file EMPTY>
    <!ATTLIST a_vis_window_file
      file_path  CDATA  #REQUIRED
    >

  <!ELEMENT subgraphs (a_subgraph_file)*>

    <!ELEMENT a_subgraph_file EMPTY>
    <!ATTLIST a_subgraph_file
      file_path  CDATA  #REQUIRED
    >
```

Hier ein Beispiel beispiel.xml einer Projektdatei eines Projektes mit Namen Beispiel, Abschnittsweise aufgetrennt mit Erklärungen:

```
<?xml version="1.0" encoding="ISO-8859-1"?>
<!DOCTYPE giant_project_file
  SYSTEM "giant_project_file.dtd">


<giant_project_file>
```

Hier der Name (mit vollständigem, absolutem Pfad) der IML-Datei und eine Checksumme, anhand derer GIANT mit recht großer Wahrscheinlichkeit erkennt, falls die IML-Datei beschädigt oder in einem angelegten Projekt unzulässig verändert wurde.
Es folgt der Dateiname der Knotenannotationsdatei. Nach Anlage eines Projektes sind Änderungen der IML-Datei nicht zulässig und könnten GIANT zum Abstürzen bringen.

```
  <global_data
    iml_graph_file_path = "/home/giant/test/resources/rfg_examp.iml"
    iml_graph_checksum = " 503"
    node_annotations_file_name = "node_annotations.xml" />
```

Für jedes Anzeigefenster in GIANT gibt es eine interne Datei, in der GIANT die nötigen Daten wie Positionen der Knoten etc. ablegt. Dateinamen sind jeweils mit relativem Pfad (relativ zur Projektdatei) angegeben.

```
  <visualisation_windows>
    <a_vis_window_file file_path = "./My_Window.viswin" />
    <a_vis_window_file file_path = "./Unknown 1.viswin" />
  </visualisation_windows>
```

Auch jeder Subgraph hat eine interne Datei, in der die Daten des jeweiligen Subgraphen abgelegt sind. Dateinamen sind ebenfalls jeweils mit relativem Pfad (relativ zur Projektdatei) angegeben.

```
  <subgraphs>
    <a_subgraph_file file_path = "./Test_1.subgraph" />
    <a_subgraph_file file_path = "./Super_Test_2.subgraph" />
  </subgraphs>
</giant_project_file>
```

9.3.8. Annotations-Datei

In der Annotations-Datei von GIANT, welche sich im Projektverzeichnis des dazugehörigen Projektes befindet, stehen alle Annotationen von Knoten. Der Dateiname ist node_annotations.xml.

In dieser Datei sind alle Annotationen von Knoten anhand der Knoten-ID abgelegt. Es ist selbstverständlich auch möglich, diese Datei durch andere Programme automatisiert zu erstellen und sie dann vor dem Laden des Projektes in GIANT an die richtige Stelle zu kopieren. Einträge von nicht existierenden Knoten werden ignoriert.

Hier eine Beispieldatei:

```
<?xml version="1.0" encoding="ISO-8859-1"?>
<!DOCTYPE giant_node_annotations_file
  SYSTEM "giant_node_annotations_file.dtd">

<giant_node_annotations_file>

<node_annotation node_id    = "1"
                 annotation = "This is a very important node" />

<node_annotation node_id    = "3"
                 annotation = "This is the key node of the graph" />

</giant_node_annotations_file>
```