Das Handbuch zu Rocs

Tomaz Canabrava
Andreas Cord-Landwehr
Übersetzung: Burkhard Lück

Das Handbuch zu Rocs

Inhaltsverzeichnis

1 Einführung **6**

1.1 Ziel, Zielgruppe und Arbeitsabläufe 6

1.2 Rocs kurz zusammengefasst 7

1.2.1 Graphendokumente 7

1.2.2 Kantentypen 7

1.2.3 Knotentypen 7

1.2.4 Eigenschaften 8

1.3 Anleitung 8

1.3.1 Graphen erstellen 8

1.3.2 Elementtypen erstellen 8

1.3.3 Der Algorithmus 9

1.3.4 Ausführung des Algorithmus 9

2 Die Benutzerschnittstelle von Rocs **10**

2.1 Hauptelemente der Benutzeroberfläche 10

2.2 Werkzeugleiste für den Graphen-Editor 11

3 Skripte **12**

3.1 Ausführen von Algorithmen in Rocs 12

3.1.1 Kontrolle der Skriptausführung 12

3.1.2 Skriptausgabe 12

3.1.3 Skriptmodul-API 12

4 Importieren und Exportieren **14**

4.1 Rocs-Projekte austauschen 14

4.1.1 Import und Export von Graphendokumenten 14

4.1.1.1 „Trivial Graph“-Dateiformat 14

4.1.1.1.1 Beschreibung des Formats 14

4.1.1.1.2 Ein Beispiel 15

4.1.1.2 DOT-Sprache / Graphviz-Graphendateiformat 15

4.1.1.2.1 Nicht unterstützte Fähigkeiten 15

4.1.1.2.2 Ein Beispiel 15

5 Graphenlayout **16**

5.1 Automatisches Layout für Graphen in Rocs 16

5.1.1 Kraftbasiertes Layout 16

5.1.1.1 Radiales Baumlayout 17

6 Danksagungen und Lizenz **18**

Zusammenfassung

Rocs ist ein Graphentheorie-Programm von KDE.

Kapitel 1

Einführung

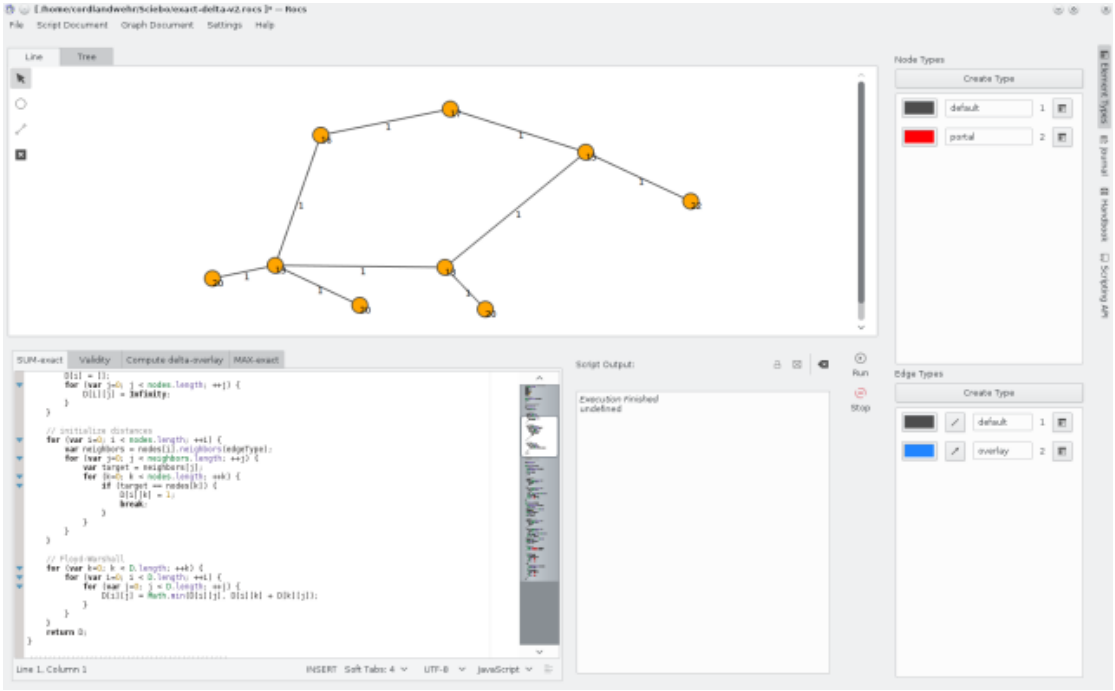
In diesem Kapitel finden Sie einen Überblick über die wichtigsten Funktionen und den typischen Arbeitsablauf. Die wichtigsten Informationen finden Sie in Abschnitt 1.2 und Kapitel 3. Damit sollten neue Benutzer problemlos mit Rocs arbeiten können.

1.1 Ziel, Zielgruppe und Arbeitsabläufe

Rocs ist eine Anwendung für Graphen-Theorie für Alle, die am Entwurf und Untersuchung von Graphen-Algorithmen arbeiten, unter anderem:

- Lehrer, die Ihren Studenten Algorithmen demonstrieren möchten,
- Studenten und Forscher, die verstehen und sehen möchten, wie ein Algorithmus funktioniert,
- Alle, die an Datenstrukturen und Algorithmen interessiert sind

Für alle diese Benutzer stellt Rocs einen einfach zu bedienenden Editor für die Erstellung von Graphen, ein leistungsfähiges Skriptmodul zur Ausführung von Algorithmen und mehrere Hilfswerkzeuge für Simulationen, Experimente und den export von Graphen bereit. Beim typischen Arbeitsablauf erstellen Sie als erstes einen Graphen entweder von hand, indem Sie Knoten und Kanten zur Zeichenfläche hinzufügen, oder mit einem der Graphen-Generatoren. Graphen-Algorithmen können dann implementiert und mit dem erzeugten Graphen ausgeführt werden. Alle Änderungen durch den Algorithmus sind sofort auf der Zeichenfläche im Editors sichtbar.



1.2 Rocs kurz zusammengefasst

Jede Rocs-Sitzung ist ein Projekt: Wenn Sie Rocs starten, wird ein leeres Projekt erstellt, wenn Sie ein Projekt laden, wird es das aktive Projekt. Ein Projekt selbst besteht aus *Graphendokumenten*, *Skripten/Algorithmen* und einem *Protokoll*.

1.2.1 Graphendokumente

Ein Graphendokument repräsentiert den Inhalt der Zeichenfläche im Graphen-Editor. Es enthält Informationen über die benutzerdefinierten Knoten- und Kantentypen. über deren Eigenschaften und über die bereits erstellten Knoten und Kanten. Daher kann Rocs aus den Knoten und Kanten eines Graphendokuments einen nicht notwendigerweise verbundenen Graphen erzeugen. Mit dem Skriptmodul ist der Zugriff mit dem globalen Objekt **`Document`** auf alle Elemente in einem Graphendokument möglich.

1.2.2 Kantentypen

Graphen können aus verschiedenen Kantentypen bestehen wie zum Beispiel aus einem ungerichteten Graphen und Baumkanten, die mit einem Algorithmus zur Breitensuche berechnet wurden. Um Kantentypen unterschiedlich zu behandeln und darzustellen, können Sie außer dem Standardtyp beliebige weitere Kantentypen definieren. Jeder Kantentyp hat eine eigene visuelle Darstellung, dynamische Eigenschaften und kann entweder gerichtet oder ungerichtet sein. Im Skriptmodul finden Sie nützliche Methoden, um nur auf bestimmte Kantentypen zuzugreifen.

1.2.3 Knotentypen

Wie bei den Kantentypen können Sie verschiedene Knotentypen für einen Graphen definieren, d. h. um einigen Knoten besondere Rollen zu geben. Jeder Knotentyp hat seine eigene visuelle Darstellung und dynamische Eigenschaften.

1.2.4 Eigenschaften

Alle Elemente wie Knoten und Kanten können Eigenschaften haben, sie müssen beim zugehörigen Knoten- oder Kantentyp definiert werden. Eigenschaften werden durch den Namen definiert und der Zugriff ist über diesen Namen möglich. Sie können beliebige Werte enthalten. Um neue Eigenschaften zu definieren oder vorhandene Eigenschaften zu bearbeiten, öffnen Sie die Seitenleiste **Elementtypen** und klicken auf den Knopf **Eigenschaften**, um den Dialog anzuzeigen.

Sie können auch das Skriptmodul zum Zugriff auf registrierte Eigenschaften zu erhalten und deren Werte zu ändern. Im folgenden Beispiel wird vorausgesetzt, dass die Eigenschaft „weight“ für den Standard-Kantentyp registriert ist.

```
var nodes = Document.nodes()
for (var i = 0; i < nodes.length; ++i){
    nodes[i].weight = i;
}
for (var i = 0; i < nodes.length; ++i){
    Console.log("weight of node " + i + ": " + nodes[i].weight);
}
```

1.3 Anleitung

In diesem Abschnitt wird ein Beispielprojekt erzeugt, um die wichtigsten Funktionen von Rocs zu zeigen. Als Ziel soll ein Graph und ein Skript erstellt werden, um damit einen einfachen Approximationsalgorithmus für das *Minimum-Knotenüberdeckungproblem* zu zeigen, der eine Knotenüberdeckung mit Güte 2 berechnet. Mit dem Minimum-Knotenüberdeckungsproblem wird eine Untermenge von Graphenknoten C mit minimaler Größe gesucht, so dass jede Graphenkante mit mindestens einem Knoten in C verbunden ist. Dies Problem ist NP-vollständig. Es soll gezeigt werden, wie eine Approximation mit dem Faktor 2 durch Suchen nach Matching in den angegebenen Graphen berechnet wird.

Ziel ist die Visualisierung der Beziehung von Matching-Knotenüberdeckungen und Minimum-Knotenüberdeckungen. Dazu werden zwei Kantentypen definiert, einer für die Matching-Kanten und einer zur Darstellung normaler Kanten, außerdem noch zwei Datentypen, die zur Unterscheidung von in C enthaltenen bzw. nicht enthaltenen Knoten verwendet werden.

1.3.1 Graphen erstellen

Zum Erstellen eines Graphen verwenden Sie in der Voreinstellung den Graphen-Generator von Rocs. Wählen Sie dazu im Menü **Graphendokument** → **Extras** → **Graph erstellen**. Wählen Sie einen **Zufälligen Graph** mit 30 Knoten, 90 Kanten und mit Seed = 1. Seed ist der Startwert für den Generator des zufälligen Graphen. Wird der gleiche Seed verwendet, ergibt das auch reproduzierbar den gleichen Graphen.

1.3.2 Elementtypen erstellen

Öffnen Sie die Seitenleiste **Elementtypen** und erstellen Sie einen zweiten Knotentyp und auch einen zweiten Kantentyp. Klicken Sie bei beiden neuen Typen auf den Knopf **Eigenschaften**, um einen Dialog zu öffnen. Setzen Sie die Kennung der neuen Typen auf `2` und geben Sie ihnen eine andere Farbe, um si von den Standardtypen zu unterscheiden. Dann setzen Sie alle Kantentypen auf Bidirektional und stellen die Kennung der Standardtypen auf `1`.

1.3.3 Der Algorithmus

Dann muss der Algorithmus der Approximation wie im folgenden implementiert werden:

```
for (var i=0; i < Document.nodes.length; i++) {
    Document.nodes[i].type = 1;
}
for (var i=0; i < Document.edges.length; i++) {
    Document.edges[i].type = 1;
}

var E = Document.edges(); // Menge der nicht bearbeiteten Kanten
var C = new Array();            // Matching-Kanten
while (E.length
> 0) {
    var e = E[0];         // die erste Kante e={u,v}
    var u = e.from();
    var v = e.to();
    e.type = 2;         // als Matching-Kanten setzen
    E.shift();             // e (i.e., E[0]) aus der Liste der Kanten ↩
        entfernen
    C.push(u);             // u zu C hinzufügen
    C.push(v);             // v zu C hinzufügen


    // u,v als Knoten von C markieren C
    u.type = 2;
    v.type = 2;

    // von E alle Kanten entfernen, die inzident zu u oder v sind
    var adjacent = u.edges();
    for (var i=0; i < adjacent.length; i++) {
        var index = E.indexOf(adjacent[i]); // Index suchen
        if (index != -1) {
            E.splice(index, 1); // Löschen, wenn gefunden
        }
    }
    var adjacent = v.adj_edges();
    for (var i=0; i < adjacent.length; i++) {
        var index = E.indexOf(adjacent[i]); // Index suchen
        if (index!=-1) {
            E.splice(index, 1); // Löschen, wenn gefunden
        }
    }
}
Console.log("Knotenüberdeckung enthält " + C.length + " Knoten.");
```

1.3.4 Ausführung des Algorithmus

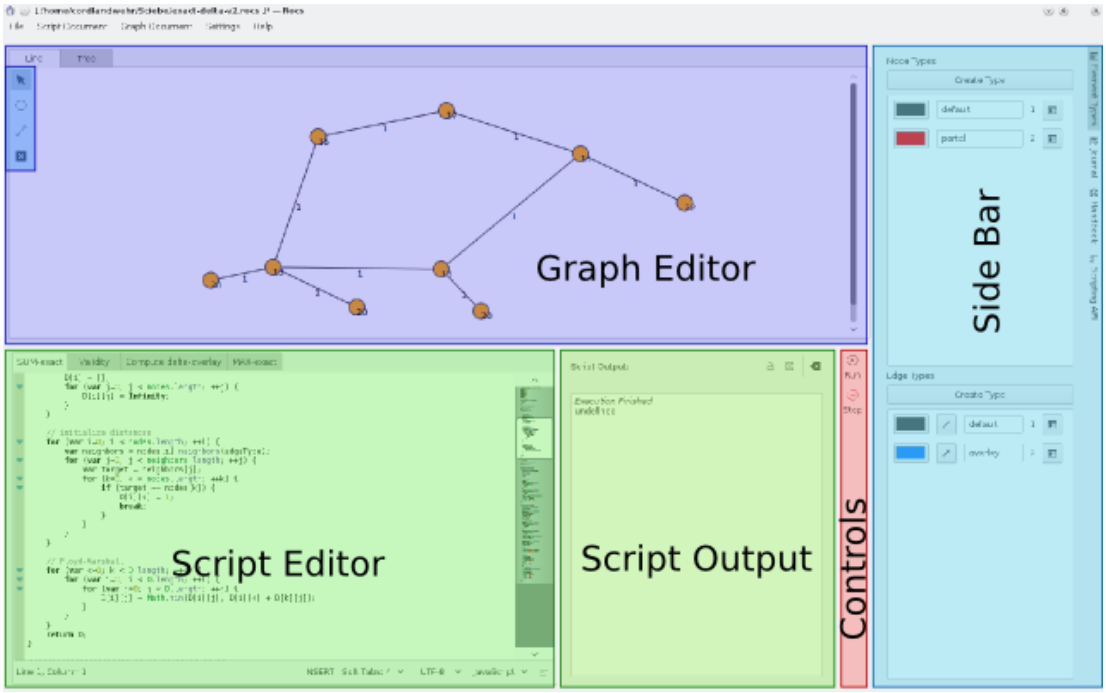
Der Algorithmus kann ausgeführt werden, indem Sie auf den Knopf **Ausführen** der Bedienelemente für Skripte drücken:

Kapitel 2

Die Benutzerschnittstelle von Rocs

2.1 Hauptelemente der Benutzeroberfläche

Die Benutzerschnittstelle besteht aus mehreren logischen Teilen, die im folgenden Bildschirmfoto gezeigt werden.



Graphen-Editor

Im Editor gibt es eine Zeichenfläche, auf der Knoten und Kanten erstellt werden können. Mit Doppelklick auf jedes dieser Elemente öffnen Sie das zugehörige Eigenschaftenmenü. Sie können auch die Aktionen aus der Werkzeugleiste benutzen, um Graphen zu erstellen und zu bearbeiten.

Werkzeugleiste für den Graphen-Editor

In dieser Werkzeugleiste finden Sie die Aktionen **Knoten erstellen** oder **Kante erstellen**, damit können Sie neue Elemente auf der Zeichenfläche erzeugen. Beachten Sie die zusätzliche Werkzeugleiste zur Auswahl des entsprechenden Knoten- oder Kantentyps, die

angezeigt wird, wenn diese Aktionen ausgewählt sind. Hier finden Sie auch Aktionen zur Auswahl, zum Verschieben und zum Löschen von Elementen. Weitere Details finden Sie im Abschnitt Abschnitt 2.2.

Seitenleiste

Rechts finden Sie die Seitenleiste mit mehreren Werkzeugen zur Unterstützung Ihres Arbeitsablaufs.

- Elementtypen: Hier haben Sie direkten Zugriff auf alle vorhandenen Kanten- und Knotentypen.
- Journal: Für jedes Projekt gibt es ein Journal, um z. B. Aufgaben, Ergebnisse oder Beobachtungen festzuhalten.
- Skript-API: Benutzen Sie dieses Fenster zu Zugriff auf das Handbuch und dadurch auch zur Skriptdokumentation.

Skripteditor

In diesem Texteditor können Sie Algorithmen schreiben, das wird im Kapitel 3 erläutert. Sie können gleichzeitig an mehreren Skriptdokumenten arbeiten, indem Sie Unterfenster verwenden.

Skriptausgabe

In diesem Textfeld werden Fehlermeldungen oder Skriptausgaben Ihres Algorithmus angezeigt, abhängig von der gewählten Einstellung der Ausgabe oben. Tritt ein Fehler in Ihrem Skript auf, wird automatisch die Fehlerausgabe angezeigt.

Bedienelemente

In diesem Bereich finden Sie die Bedienelemente, um die Ausführung von Skripten zu steuern. Mit dem Knopf **Ausführen** starten Sie das aktuell im Editor geöffnete Skript. Wird ein Skript ausgeführt, können Sie es mit **Anhalten** stoppen.

2.2 Werkzeugleiste für den Graphen-Editor

Die Werkzeugleiste enthält die im Folgenden genannten Aktionen. Klicken Sie auf den Knopf für eine Aktion, dann können Sie diese Aktion anschließend mit dem Mauszeiger auf die Elemente auf der Zeichenfläche anwenden.

- **Auswählen und Verschieben**: Um Elemente auszuwählen, klicken Sie entweder mit der linken Maustaste auf eine freie Stelle auf der Zeichenfläche, halten die Maustaste gedrückt und ziehen ein Rechteck um Datenelemente und/oder Kanten, um sie auszuwählen oder klicken Sie sonst zur Auswahl auf ein nicht ausgewähltes Element. Klicken Sie dann auf ein oder mehrere ausgewählte Elemente und verschieben sie mit gedrückter Maustaste. Ausgewählte Elemente können auch mit den Pfeiltasten verschoben werden.
- **Knoten erstellen**: Klicken Sie mit der linken Maustaste auf eine beliebige Position auf der Zeichenfläche, um ein neues Datenelement für die gerade ausgewählte Datenstruktur zu erstellen. Halten Sie die Maustaste gedrückt, dann erscheint ein Menü, aus dem der Datentyp des neu erstellten Datenelements ausgewählt werden kann, wenn es mehrere Datentypen gibt.
- **Kante erstellen**: Klicken Sie mit der linken Maustaste auf ein Datenelement, halten Sie die Maustaste gedrückt und ziehen Sie eine Linie zu einem anderen Datenelement. Diese Aktion kann nur dann ausgeführt werden, wenn beim aktuellen Graphen diese Kante hinzugefügt werden kann. In einem ungerichteten Graphen dürfen Sie z. B. nicht mehrere Kanten zwischen zwei Datenelementen einfügen. Halten Sie die Maustaste gedrückt, dann erscheint ein Kontextmenü, aus dem der Datentyp der neu erstellten Kante ausgewählt werden kann, wenn es mehrere Kantentypen gibt.
- **Löschen**: Klicken Sie auf ein Element, um es zu löschen. Löschen Sie einen Knoten, dann werden alle anliegenden Kanten ebenfalls gelöscht.

Kapitel 3

Skripte

3.1 Ausführen von Algorithmen in Rocs

Rocs verwendet intern das QtScript-/Javascript-Modul. Daher müssen Sie für alle implementierten Algorithmen Javascript benutzen. In diesem Abschnitt wird erklärt, wie der Zugriff und Änderung von Elementen eines Graphendokuments mit dem Skriptmodul erfolgt. Beachten Sie, dass Änderungen durch das Skriptmodul direkt die Eigenschaften der Elemente im Graphen-Editor beeinflussen.

3.1.1 Kontrolle der Skriptausführung

Es gibt verschiedene Ausführungsmodi für Ihre Algorithmen.

- **Ausführen**: Führt das Skript bis zum Ende aus.
- **Anhalten**: Hält die Ausführung eines Skripts an und ist nur während der Ausführung aktiviert.

3.1.2 Skriptausgabe

Bei der Ausführung eines Algorithmus werden Meldungen in der *Fehler- & Skriptausgabe* angezeigt. Erkennt das Skriptmodul einen Syntaxfehler in Ihrem Skript, wird der Fehler als Debug-Nachricht angezeigt. Auch alle Programmausgaben werden in der Fehlerausgabe als fett gedruckter Text angezeigt.

Sie können den angezeigten Text in der Skriptausgabe mit folgenden Funktionen steuern:

```
Console.log(string message);            // zeigt message als Skriptausgabe
    Console.debug(string message);              // zeigt message als Debug- ↩
        Ausgabe
Console.error(string message);          //  zeigt message als Fehlerausgabe
```

3.1.3 Skriptmodul-API

Die einzelnen Bestandteile von Rocs stellen ein statisches Element bereit, auf das mit dem Skriptmodul zugegriffen werden kann:

- **`Document`** für das Graphendokument
- **`Console`** für die Protokollausgabe der Konsole

. Informationen über die Programmschnittstelle und eine Referenz der Methoden finden Sie in der Seitenleiste von Rocs.

Kapitel 4

Importieren und Exportieren

4.1 Rocs-Projekte austauschen

Rocs-Projekte können als Archivdateien im Format `.tar.gz` importiert und exportiert werden. Diese Archive können zum Austausch von Projekten benutzt werden. Der Import und Export erfolgt mit **Graphendokument** → **Graphen importieren** beziehungsweise **Graphendokument** → **Graphen exportieren unter ...**

4.1.1 Import und Export von Graphendokumenten

Rocs unterstützt zurzeit den Import und Export folgender Dateiformate:

- DOT-Dateien, auch als Graphviz-Dateien bekannt.
- GML-Dateien
- „Trivial Graph"-Dateien
- Keyhole-Markup-Sprachformat

4.1.1.1 „Trivial Graph"-Dateiformat

Das *„Trivial Graph"-Format* (TGF) ist ein einfaches textbasiertes Dateiformat zur Beschreibung von Graphen. Eine TGF-Datei besteht aus einer Liste von Knotendefinitionen mit der Zuordnung von Kennungen zu Beschriftungen, gefolgt von einer Liste von Kanten. In diesem Format kann es nur eine Beschriftung für einen Knoten und einen Wert für eine Kante geben. Rocs interpretiert importierte Graphen als ungerichtet. Exportierte Graphen enthalten zwei Kanten für jede zweiseitige Verbindung.

4.1.1.1.1 Beschreibung des Formats

- Die Datei beginnt mit einer Liste der Knoten, jeder Knoten auf einer einzelnen Zeile. Dann folgt eine Zeile nur mit dem Zeichen „#", gefolgt von einer Liste von Kanten in einer eigenen Zeile.
- Ein Knoten wird durch eine ganze Zahl als Kennung, dann ein Leerzeichen und ein beliebiger Text beschrieben.
- Die Beschreibung einer Kante besteht aus zwei ganzen Zahlen (Kennungen von Knoten), durch Leerzeichen getrennt, dann folgt nach einem Leerzeichen ein beliebiger Text. Es wird angenommen, dass die gerichtete Kante von der ersten zur zweiten Kennung zeigt.

4.1.1.1.2 Ein Beispiel

```
1 Startknoten
2 Sender
3 Endknoten
#
1 2 blau
2 1 rot
2 3 grün
```

4.1.1.2 DOT-Sprache / Graphviz-Graphendateiformat

„DOT“ ist eine Beschreibungssprache für Graphen, die ein einfaches Textformat verwendet. Damit ist die Darstellung von Graphen für den Benutzer gut lesbar und sie kann effizient durch Anzeigeprogramme für Graphen verarbeitet werden. „DOT“ ist das Standardformat für das Visualisierungsprogramm Graphviz, wird aber auch häufig von anderen Graphenprogrammen verwendet. „DOT“-Dateien haben üblicherweise die Erweiterung *.gv* und *.dot*.

4.1.1.2.1 Nicht unterstützte Fähigkeiten

Rocs kann jede Graphendatei einlesen und verarbeiten, die in der Sprache „DOT“[1] spezifizierte Graphen enthält. Die Funktionen der Sprache werden vollständig mit folgenden Ausnahmen unterstützt:

- Untergraph: Wegen des fehlenden Konzepts für Untergraphen in Rocs werden Untergraphen nur als Menge von Datenelementen und Verbindungen importiert. Besonders Verbindungen zu oder von Untergraphen werden nicht importiert.
- HTML und XML Attribute: Attribute wie Beschriftungen mit HTML- oder XML-Syntax werden unverändert eingelesen. Insbesondere werden keine Anpassungen für Schriften und Stile aus diesen Attributen eingelesen.

4.1.1.2.2 Ein Beispiel

```
digraph myGraph {
    a -> b -> c;
    b -> d;
}
```

[1] http://www.graphviz.org/content/dot-language

Kapitel 5

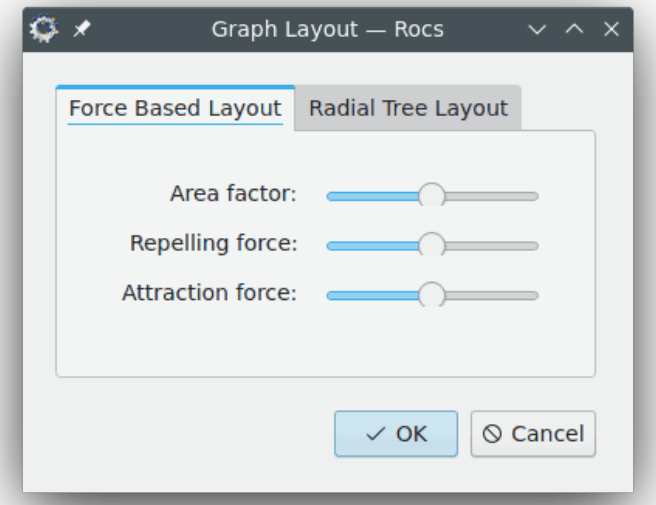
Graphenlayout

5.1 Automatisches Layout für Graphen in Rocs

Rocs kann Layouts von Graphen automatisch anlegen. Benutzen Sie dazu im Hauptmenü **Graphendokument** → **Extras** → **Graphenlayout**. Es gibt zwei verschiedene Layout-Algorithmen, die benutzt werden können: **Kraftbasiertes Layout** und **Radiales Baumlayout**. Wählen Sie die die entsprechende Registerkarte im Dialog, stellen Sie die gewünschten Parameter ein und führen Sie den Algorithmus aus, indem Sie auf **OK** klicken. Weitere Informationen zu den Layout-Algorithmen werden werden in den nächsten Abschnitten beschrieben.

5.1.1 Kraftbasiertes Layout

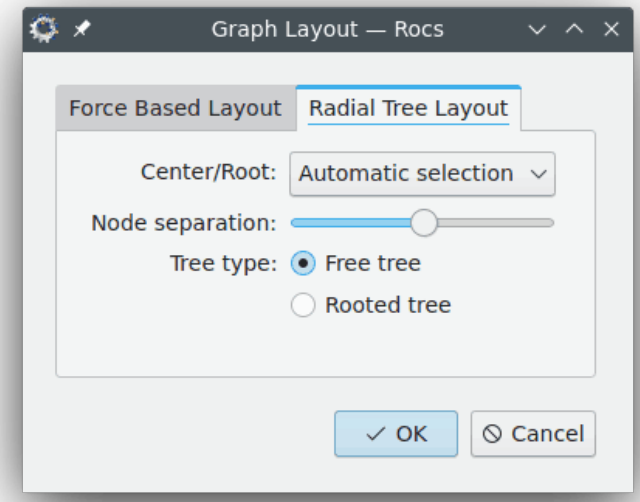
Das **Kraftbasiertes Layout** kann auf jeden Graphen angewendet werden. Dieser Algorithmus simuliert Kräfte, die zwischen den Knoten wirken. Es gibt abstoßende und anziehende Kräfte zwischen benachbarten Knotenpaaren. Die Größe dieser Kräfte kann mit Schiebereglern in der Benutzeroberfläche festgelegt werden.



Ein weiterer Parameter zur Anpassung des Layouts ist der Flächenfaktor. Dieser Parameter bestimmt die Verteilung der Knoten. Hohe Werte des Flächenfaktors führen zu großen Abständen zwischen den Knoten.

5.1.1.1 Radiales Baumlayout

Das „Radiale Baumlayout“ kann nur auf Bäume angewendet werden. Bei anderen Arten von Graphen, führt die Anwendung dieses Layout zu einer Fehlermeldung. Die Parameter für das „Radiale Baumlayout“ können über die Benutzeroberfläche eingestellt werden.



Als **Baumtyp** können Sie zwischen einem freien Baum und einem Basisbaum wählen. In einem freien Baumlayout werden die Knoten frei ohne eine offensichtliche Hierarchie zwischen ihnen angeordnet. In einem Basisbaum-Layout wird der Basisknoten oben und die Unterbäume darunter angeordnet, so dass eine Hierarchie zwischen den Knoten entsteht.

Der Parameter **Zentrum/Basis** legt fest, welcher Knoten als Basis für das Baumlayout oder als Zentrum für das freie Baumlayout verwendet wird. Das Zentrum eines eines freien Baumlayouts ist der erste Knoten, der vom Algorithmus platziert wird. Alle anderen Knoten werden auf Kreisen platziert, die um den Basisknoten zentriert sind. Ein Zentrum oder eine Basis kann vom Layout-Algorithmus automatisch ausgewählt werden.

Der Parameter „Knotenabstand“ steuert den Abstand zwischen den Knoten. Wenn Sie den Wert dieses Parameters erhöhen, wird der Abstand zwischen den Knoten vergrößert. Bei kleineren Werten dieses Parameters wird der Abstand zwischen den Knoten verringert.

Kapitel 6

Danksagungen und Lizenz

Rocs
Copyright für das Programm:

- Copyright 2008 Ugo Sangiori (ugorox AT gmail.com)
- Copyright 2008-2012 Tomaz Canabrava (tcanabrava AT kde.org)
- Copyright 2008-2012 Wagner Reck (wagner.reck AT gmail.com)
- Copyright 2011-2015 Andreas Cord-Landwehr (cordlandwehr AT kde.org)

Copyright der Dokumentation:

- Dokumentation Copyright 2009 Anne-Marie Mahfouf annma@kde.org
- Dokumentation Copyright 2009 Tomaz Canabrava (tcanabrava AT kde.org)
- Dokumentation copyright 2011-2015 Andreas Cord-Landwehr (cordlandwehr AT kde.org)

Übersetzung Burkhard Lück lueck@hube-lueck.de

Diese Dokumentation ist unter den Bedingungen der GNU Free Documentation License veröffentlicht.
Dieses Programm ist unter den Bedingungen der GNU General Public License veröffentlicht.