



UNIVERSITÄT
KOBLENZ · LANDAU

Fachbereich 4: Informatik

Programm zur Erstellung von Mind Maps mit Hilfe der Netron Graph Library

Studienarbeit

im Studiengang Computervisualistik

vorgelegt von

Laura Dunkel

Betreuer: Dipl.-Inform. Marc Santos
(Institut für Wissensmedien)

Koblenz, im November 2006

Erklärung

	Ja	Nein
Mit der Einstellung dieser Arbeit in die Bibliothek bin ich einverstanden.	<input type="checkbox"/>	<input type="checkbox"/>
Der Veröffentlichung dieser Arbeit im Internet stimme ich zu.	<input type="checkbox"/>	<input type="checkbox"/>

Ort, Datum

Unterschrift

Inhaltsverzeichnis

1. Einführung	7
1.1 Ziel der Studienarbeit	7
1.2 Hilfsmittel und Herangehensweise	7
2. Kurze Einführung Mind Maps	9
3. Evaluation bereits vorhandener Programme	10
3.1 FreeMind	10
3.1.1 FreeMind Einführung	11
3.1.2 Fazit	16
3.2 MindManager	17
3.2.1 MindManager Einführung	18
3.2.2 Fazit	23
3.3 Vergleich	24
4. Design des Gesamtsystems	26
4.1 Systemeinsatz und Systemumgebung	26
4.2 Funktionale Anforderungen	26
4.2.1 Musskriterien	26
4.2.2 Wunschkriterien:	26
4.3 Nichtfunktionale Anforderungen	27
4.4 Benutzerschnittstellen und Fehlerverhalten	27
4.5 Dokumentationsanforderungen	28
4.6 Abnahmekriterien	28
5. Umsetzung	29
6. Fazit und Ausblick	44
7. Glossar	46
8. Quellenverzeichnis	48

Abbildungsverzeichnis

Abb. 1: In FreeMind erstellte Mind Map	11
Abb. 2: Startmodus FreeMind	11
Abb. 3: Leere Mind Map	11
Abb. 4: Rechtsklick auf freien Arbeitsbereich	12
Abb. 5: Rechtsklick auf Knoten	12
Abb. 6: Freemind Menüpunkt „Einfügen“ obere Leiste	13
Abb. 7: Freemind Menüpunkt „Einfügen“ bei Rechtsklick auf Knoten	13
Abb. 8: Freemind Menüpunkt „Format“ bei Rechtsklick auf Knoten	14
Abb. 9: Freemind Menüpunkt „Format“ obere Leiste	14
Abb. 10: Übersicht obere Leiste	14
Abb. 11: Übersicht Importmöglichkeiten	15
Abb. 12: Übersicht Exportmöglichkeiten	15
Abb. 13: Übersicht Extras	16
Abb. 14: Übersicht Hilfesystem	16
Abb. 15: In MindManager erstellte Mind Map	18
Abb. 16: Startzustand MindManager	19
Abb. 17: Menü durch Rechtsklicken auf Knoten	20
Abb. 18: Startzustand MindManager	20
Abb. 19: Powerfilter MindManager	21
Abb. 20: Präsentationsansicht MindManager	21
Abb. 21: Brainstorming – Option in MindManager	22
Abb. 22: Überarbeitung – Option in MindManager	22
Abb. 23: Untere Werkzeugleiste	23
Abb. 24: Linke Werkzeugleiste	23
Abb. 25: Startzustand MindMap	29
Abb. 26: Kontextmenü MindMap	30
Abb. 27: Schnellstartleiste MindMap	31
Abb. 28: Baumansicht links MindMap	31
Abb. 29: Layoutabfrage	32
Abb. 30: Veranschaulichung Berechnungen Layoutalgorithmus	35
Abb. 31: Korrespondierende Arbeitsfläche rechts MindMap	36
Abb. 32: Netron Tree – Layout	37
Abb.33: Datei – Menü MindMap	37
Abb.35: Einfügen – Menü MindMap	39
Abb.34: Datei – Menü MindMap	40
Abb.36: Baumansicht – Menü MindMap	40
Abb.37: Arbeitsfläche – Menü MindMap	41
Abb.38: Ansicht – Menü MindMap	43
Abb.39: Hilfe – Menü MindMap	43

1. Einführung

1.1 Ziel der Studienarbeit

Diese Arbeit hatte vor allen Dingen zwei wichtige Zielsetzungen, die im Rahmen von sechs Monaten bestmöglich realisiert werden sollten.

Zum Einen sollte eine Anwendung zur Erstellung von Mind Maps entwickelt werden, da es in diesem Sektor wenige gute Open Source – Produkte gibt. Hierzu muss allerdings erwähnt werden, dass die Entwicklung eines Programms, welches in einer sechsmonatigen Studienarbeit entwickelt werden soll, nicht das leisten kann, was bei anderen Mind Map – Programmen in einem Team mit erfahrenen Programmierern in einem längeren Zeitraum und mit anderen Mitteln entwickelt worden ist. Es soll vielmehr einen Anfang darstellen, dessen Hauptaugenmerk auf einem funktionierenden Gesamtsystem liegen soll.

Zum Anderen sollte diese Studienarbeit aber vor Allem auch dazu dienen, bessere Programmierkenntnisse in objektorientierter Programmierung, insbesondere bei der Erstellung einer grafischen Oberfläche zu erlangen und zu vertiefen. Außerdem sollte die Studienarbeit eine Einführung in eigenständiges, wissenschaftliches Arbeiten darstellen.

1.2 Hilfsmittel und Herangehensweise

Um die Aufgabenstellung erfolgreich lösen zu können, wurde sich dafür entschieden, die Open Source – Bibliothek „.Netron Graph Library“ zu verwenden, die dem Programmierer bei der Erstellung der Mind Map unterstützt. So werden hier bereits Grundformen von Knoten oder z.B. das Exportieren in Pixelgrafiken und noch vieles mehr zur Verfügung gestellt.

Des Weiteren wurde als objektorientierte Programmiersprache C Sharp verwendet, welche einem besonders die Erstellung von grafischen Oberflächen enorm erleichtert hat. Aber auch die Sprache an sich ist relativ leicht zu erlernen, wenn man Grundkenntnisse in der Programmiersprache Java und der objektorientierten Programmierung im Allgemeinen besitzt.

Microsoft Visual Studio .NET 2005 unterstützt das Entwickeln von Anwendungen durch die Vielfalt an Funktionen, die das Programm liefert. So lassen sich zum Beispiel sehr schnell einfache Setupprojekte erstellen.

Zu Beginn der Arbeit wurden zwei Programme zur Erstellung von Mind Maps evaluiert, um so ein besseres Verständnis für die Aufgabenstellung und das spätere Implementieren zu bekommen. Die Programme, welche für die Evaluation ausgewählt

wurden, sind das relativ bekannte Open Source – Produkt „FreeMind“ und das kostenpflichtige Programm „MindManager“ (was jedoch eine Testversion zur Verfügung stellt).

Nach erfolgreicher Evaluation wurden anschließend beide Programme im direkten Vergleich gegenübergestellt, um so ein Verständnis dafür entwickeln zu können, was an beiden Programmen gut umgesetzt wurde und was vielleicht besser umgesetzt werden könnte oder sogar an Funktionalität fehlt. Daraus resultierend konnte leicht ein Modell für die Entwicklung eines Mind Map – Programms im Rahmen dieser Studienarbeit entworfen werden.

Aufgrund der oben genannten Hilfsmittel und der Programmevaluation wurde ein grobes Designkonzept erarbeitet, welches im Kapitel 4 näher erläutert wird. Anhand dieses Entwurfes kann dann in Kapitel 5 die Umsetzung nachgelesen werden.

Die grafische Oberfläche ist direkt zu Beginn entstanden und wurde während der Implementierung ständig aktualisiert und um neue Funktionalität erweitert.

2. Kurze Einführung Mind Maps

Mind Maps, was auf deutsch soviel heißt wie Gedankenkarten, wurden 1960 durch den britischen Mentaltrainer und Autor Tony Buzan eingeführt und stellen seit dem eine beliebte Kreativitätstechnik dar. Sie können in vielen Bereichen des Arbeitslebens und der Schule eingesetzt werden, und sollen angeblich beide Gehirnhälften anregen und somit das Erfassen und das Erinnern von Konzepten erleichtern (vgl. [2]_A).

So können mit Hilfe dieser Kreativitätstechnik zum Beispiel Sachverhalte mit all seinen Unterthemen in einer Mind Map übersichtlich dargestellt werden, sei es zum Brainstorming in einer Gruppe oder auch alleine. Des Weiteren hat man die Möglichkeit, Mind Maps zur Präsentation eines Konzepts zu verwenden, um so anderen diese Idee leicht verständlich zu machen. Mit Mind Maps kann man also schnell und einfach eine Übersicht über ein Thema erlangen. Diesen Aspekt kann man sich natürlich auch für etwaige Prüfungsvorbereitungen zu Nutze machen, um so einen groben Überblick über den gesamten Lernstoff zu bekommen.

Grafisch aufgebaut wird eine Mind Map wie folgt. Es gibt in der Mitte einen zentralen Knoten, die Wurzel, der den Rahmen, also das eigentliche Thema der Mind Map, festlegt. Dann können von diesem Wurzelknoten „Kinder“ in Form von Ästen abgeleitet werden, die wiederum einen Knoten besitzen und thematisch einen Unterbereich des Wurzelknotens darstellen. So kann nach und nach eine hierarchische Struktur aufgebaut werden.

Die Struktur soll sinnvoll aufgebaut sein, das heißt, inhaltlich soll sich zum Beispiel der erste Kindsknoten von der Wurzel ableiten. Das Kind des ersten Kindknotens soll wiederum inhaltlich von seinem Vater abgeleitet werden können, so dass die Hierarchieebenen Sinn ergeben.

Zusätzlich hat man die Möglichkeit, die einzelnen Knoten durch Farben und Bilder weiter zu verfeinern und hervorzuheben.

Man muss aber auch erwähnen, dass Mind Maps in vielen Fällen nur dem oder den Verfassern nützlich sein können. Das liegt daran, dass Schlüsselbegriffe oder auch die Strukturierung sehr oft individuell und auf bestimmte Regionen oder Arbeitswelten abgestimmt sind, so dass es Personen außerhalb dieses Informationskreises ohne weitere Erklärungen schwer fallen könnte, eben diese Strukturierung und das Thema genau zu erfassen (vgl. [2]_A).

Außerdem werden Mind Maps ab einer gewissen Größe schnell unübersichtlich und sind nicht mehr geeignet, den Sachverhalt klar und verständlich zu repräsentieren (vgl. [2]_A).

3. Evaluation bereits vorhandener Programme

Dieses Kapitel soll eine Hilfestellung für die spätere Implementierung darstellen. Die Evaluation zweier bekannter Mind Mapping – Programme werden die Einschätzung über den erforderlichen Funktionsumfang der zu implementierenden Anwendung wesentlich vereinfachen. Man gewinnt so schnell einen Eindruck davon, was an notwendigen Funktionen gebraucht wird, was man eventuell sogar besser machen könnte oder was an Optionen gefehlt hat.

Die Programme, die evaluiert werden sollen, heißen FreeMind und MindManager. FreeMind stellt das wohl bekannteste Freeware – Produkt dar und wird oft mit dem kostenpflichtigen Programm MindManager verglichen. Da genau dieser Aspekt, die Gegenüberstellung eines kostenlosen und eines kostenpflichtigen Produkts, auch für diese Arbeit interessant ist, wurde sich für diese beiden Programme entschieden. Die Frage ist hierbei offensichtlich, wie stark die beiden Anwendungen sich voneinander unterscheiden und ob man einen merklichen Qualitätsunterschied feststellen kann.

Dazu werden zunächst die beiden Programme getrennt voneinander untersucht und dabei versucht die wichtigsten Funktionalitäten vorzustellen, um dann einen direkten Vergleich anzustreben. Aus diesen Erkenntnissen kann dann ein konkretes Programmdesign entwickelt werden, auf Grundlage dessen dann eine eigene Anwendung erstellt wird.

3.1 FreeMind

FreeMind ist ein in Java entwickeltes Freeware- Produkt, was zudem noch Open Source ist. Entwickelt wird FreeMind jetzt seit ungefähr fünf Jahren (am 18.06.2000 registriert), die momentane Version ist 0.8.0. Die Anwendung ist auf allen Plattformen lauffähig, allerdings muss die Java Laufzeitumgebung (Version 1.4 oder höher) auf dem Rechner installiert sein.

Das Programm ist in zwei Versionen erhältlich, einer etwas abgeschwächten Version, der Minimum – Version, und der Maximum – Version. Je nach gewünschtem Funktionsumfang kann man sich für die eine oder für die andere Version entscheiden. So sind z.B. in der Maximum – Version mehrere Import- bzw. Exportmöglichkeiten vorhanden.

Die nachfolgende Graphik (Abb. 1) konnte man ohne Probleme zu Beginn der Einarbeitung in FreeMind erstellen. Als Thema wurden die zu erreichenden Meilensteine dieser Studienarbeit gewählt und dessen gröbere Ziele, die im Laufe der Arbeit erreicht werden mussten. Natürlich zeigt die Abbildung nur einen kleinen Teil der Funktionen, die im Rahmen der Mind Map – Erstellung in dieser Anwendung möglich sind. Dieses Kapitel wird versuchen, die wichtigsten Funktionen im Laufe dieser Evaluation näher zu erläutern. Die oben erwähnte in FreeMind erstellte Mind

Map soll lediglich ein einführendes Beispiel darstellen und zeigen, wie die Kreativitätstechnik in diesem Programm prinzipiell umgesetzt ist.

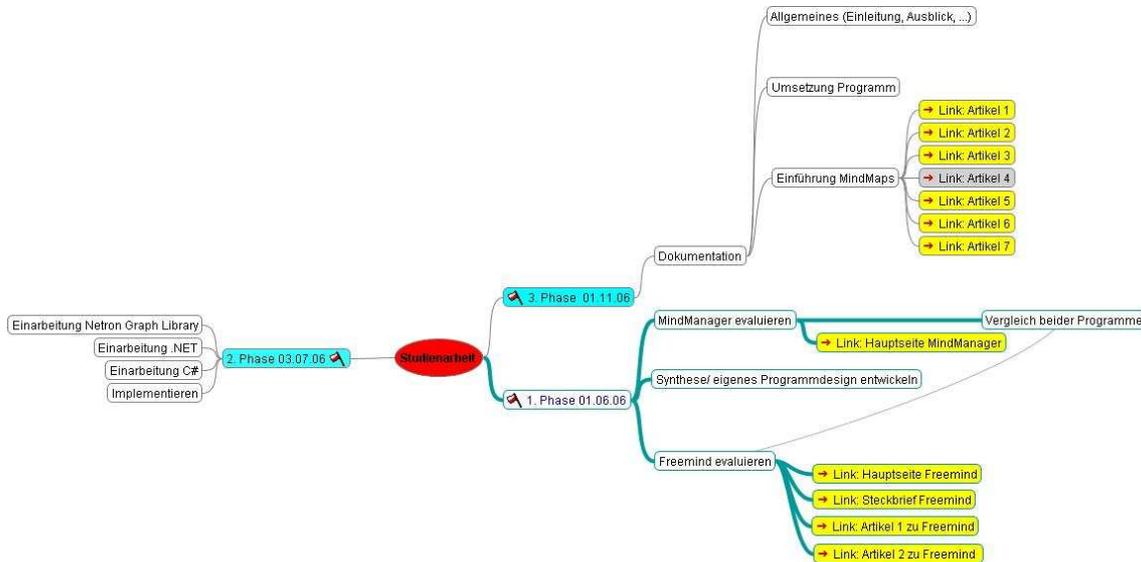


Abb. 1: In FreeMind erstellte Mind Map

3.1.1 FreeMind Einführung

Wenn FreeMind geöffnet wird, sind die meisten Funktionen grau ausgeblendet und werden erst bei Öffnen einer bestehenden Mind Map bzw. beim Klick auf das Erstellen einer neuen Mind Map freigestellt.

Auf der linken Seite befindet sich eine Leiste, die eine Vielzahl an Icons enthält, die man bei geöffneter Mind Map durch Anklicken in die jeweils markierte Blase einfügen kann (z.B. das „Fähnchen“ in den Kinderknoten der Wurzel in der Beispiel- Mind Map in Abb. 1). Des Weiteren stehen neben den gewohnten Menüpunkten (Datei, Bearbeiten, ...) eine Reihe von Buttons zur Verfügung, die man in der Regel häufiger benutzt (Speichern, Vorwärts, Zurück, ...) und deshalb für den schnellen Zugriff bereit stehen und im oberen Rand des Pogrammfensters zu finden sind.

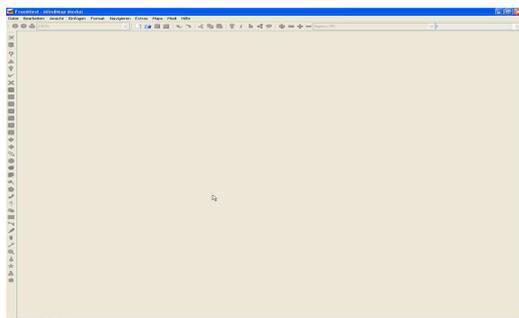


Abb. 2: Startmodus FreeMind

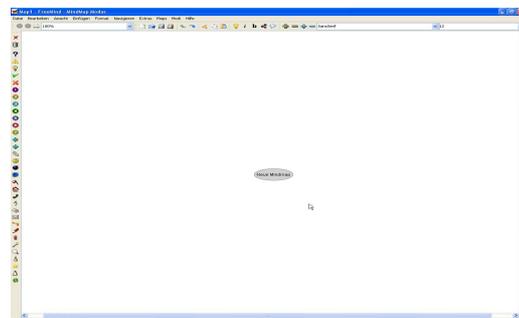


Abb. 3: Leere Mind Map

Einen Überblick über den Startzustand des Programms, sowie über den Zustand bei neu geöffneter Mind Map liefern Abbildung 2 und 3.

Der Wurzelknoten der neu geöffneter Mind Map lässt sich durch Doppelklick bzw. Linksklick mit der Maus zum Beschrifteten leicht öffnen. Weitere Funktionen zur Bearbeitung erlangt man mit einem Rechtsklick auf den Wurzelknoten bzw. Rechtsklick auf den freien Bereich. Welche Funktionen das im Einzelnen sind kann man gut in Abb. 4 und 5 erkennen.



Abb. 4: Rechtsklick auf freien Arbeitsbereich

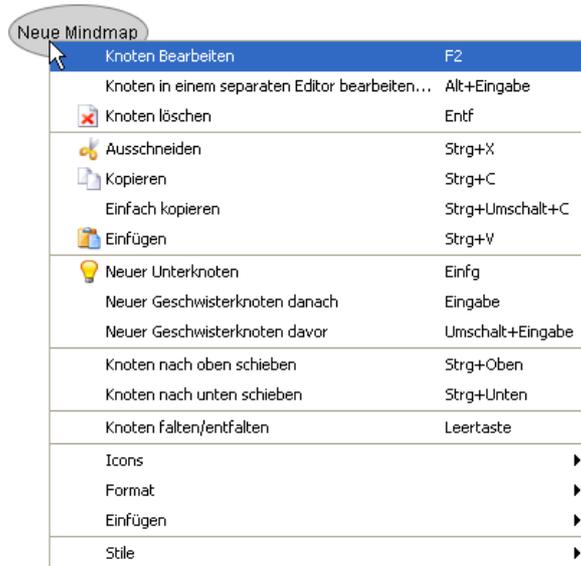


Abb. 5: Rechtsklick auf Knoten

In Abbildung 4 kann man einige Funktionen sehen, die generell bei geöffneter Mind Map zur Verfügung stehen. Das sind zum Einen das Wechseln der Ansicht zwischen verschiedenen geöffneten Mind Maps, wobei in dem Beispiel die Auswahl nur auf Map 1 beschränkt ist, und zum Anderen besteht die Möglichkeit die schon einmal erwähnten Leisten im oberen (häufig benutzte Buttons) und linken (Icons) Bildschirmrand ein- und auszublenden.

In Abbildung 5 sind weitaus mehr Funktionen bereitgestellt, die einem die Möglichkeit geben, die Mind Map pro Knoten ganz individuell gestalten zu können.

So hat man die Möglichkeit den Knoten entweder direkt in der Blase oder in einem separaten Editorfenster zu bearbeiten, wobei das Editorfenster keine neuen Funktionen bereitstellt und auch lediglich zum Ändern des Textes genutzt werden kann.

Dann gibt es noch gängige Funktionen, um den Knoten zu löschen, zu kopieren oder auszuschneiden, um dann an anderer Stelle wieder einzufügen. Ebenso ist es möglich Zweige aus- und einzublenden, die Positionen der Knoten untereinander zu ändern oder neue Knoten anzulegen (neuen Unterknoten oder zwei verschieden positionierte Geschwisterknoten).

Die letzten vier Menüpunkte stellen etwas weniger umfangreiche Verknüpfungen des oberen Programmmenüs (Format, Einfügen) dar. Hier kann man auch, wie an anderer Stelle schon einmal erwähnt, Icons in den jeweils markierten Knoten einfügen oder etwas an der Erscheinungsform des Knotens ändern. Die Einträge zu „Format“, „Stile“ und „Einfügen“ sind wieder Platzhalter für eine Menge an Funktionen. Der Punkt „Stile“ enthält z.B. eine Reihe an verschiedenen vordefinierten Formatvorlagen zur Knotenform und –farbe.

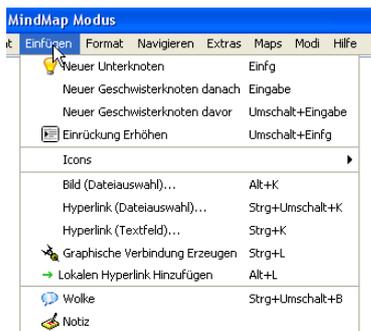


Abb. 6: Freemind Menüpunkt „Einfügen“ obere Leiste

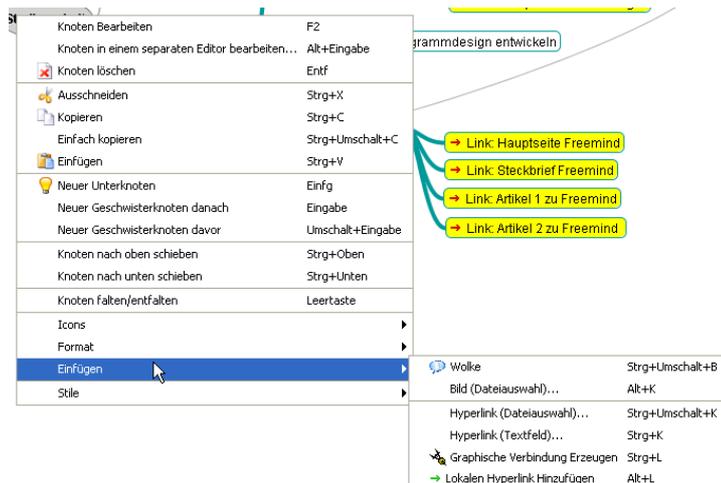


Abb. 7: Freemind Menüpunkt „Einfügen“ bei Rechtsklick auf Knoten

In Abbildung 6 sieht man den vollen Funktionsumfang des Menüpunktes „Einfügen“, wobei man in Abbildung 7 die schmalere Version für die schnelle Bearbeitung per rechten Mausklick sehen kann. Hier wurde im Ermessen der Entwickler auf einige wenige Optionen verzichtet.

Zu den Auswahlmöglichkeiten in diesem Menü zählt neben dem Anlegen von verschiedenen Knoten hauptsächlich das Einfügen verschiedener Elemente, wie etwa einer Wolke, die einen bestimmten Zweig abgrenzen kann. Außerdem besteht die Möglichkeit, Notizen oder Bilder an einen Knoten zu heften. Sehr nützlich ist die Funktion, einen Hyperlink im Knoten zu hinterlegen, der entweder einen Internetlink beinhaltet oder lokal auf eine Datei verweist. Zusätzlich kann man noch graphische Verbindungen zwischen zwei Knoten mittels Bézier – Kurven erzeugen, wie auch in Abbildung 1 zwischen den Knoten „FreeMind evaluieren“ und „MindManager evaluieren“ zu sehen ist.

Auch der Menüpunkt „Format“ ist bei direkter Bearbeitung durch den rechten Mausklick etwas abgeschwächt, was wiederum in den nachfolgenden Abbildungen 8 und 9 zu sehen ist. Allerdings handelt es sich beim Auslassen nur um kleinere Details. Im Wesentlichen umfasst der Menüpunkt die Gestaltung der eigentlichen Knoten. So kann man den Knoten blinkend hervorheben, ihm verschiedene Farben zuordnen, die Erscheinungsform ändern und eine Knotenhintergrundfarbe festlegen. Des Weiteren ist

es möglich, die Schrift durch Farbänderungen, Größenänderungen, etc. verschiedenartig darzustellen.

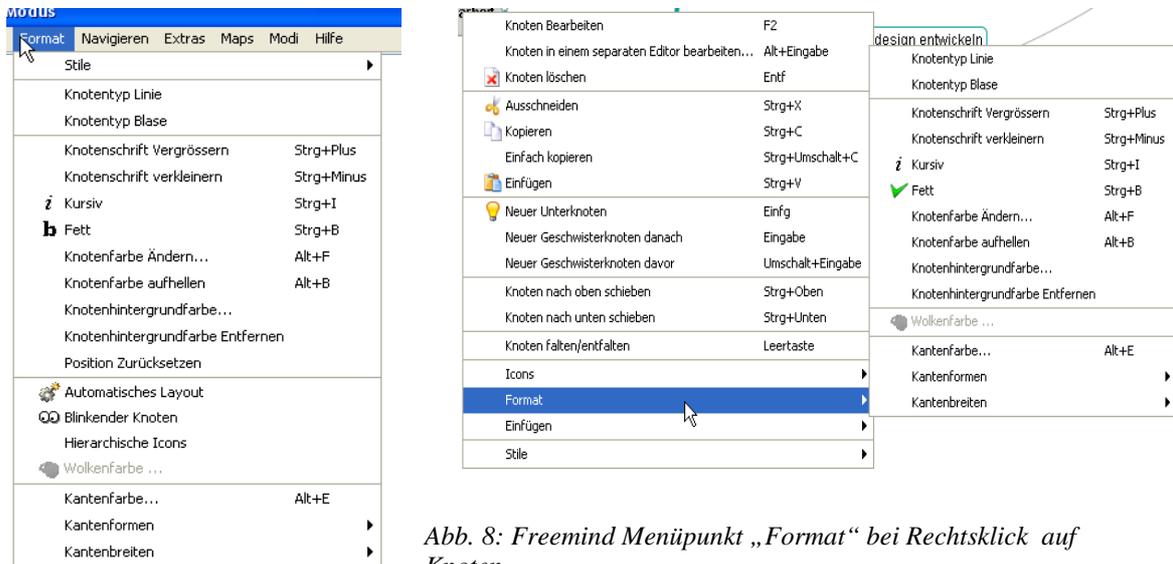


Abb. 8: Freemind Menüpunkt „Format“ bei Rechtsklick auf Knoten

Abb. 9: Freemind Menüpunkt „Format“ obere Leiste

Die restlichen Menüpunkte erreicht man über die obere Programmleiste, wie hier in der Übersicht (Abb. 10) zu sehen ist.

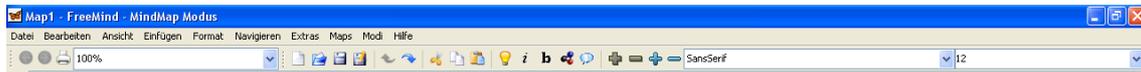


Abb. 10: Übersicht obere Leiste

Die Leiste beginnt mit dem „Datei“ – Menü und beinhaltet die gängigen Funktionen. Man kann neue Mind Maps anlegen, öffnen, speichern, schließen, drucken, sogar Mind Maps verschlüsseln. Interessant ist die Vielzahl an Import- bzw. Exportmöglichkeiten, die FreeMind dem Anwender bietet.

Bei den Importmöglichkeiten kann man einen anderen Zweig einer FreeMind Mind Map importieren, oder eine Datei die durch einen Link im Knoten bereits verzweigt ist, importieren. Ebenso ist es möglich entweder die Favoriten des Explorers, eine bestimmte Ordnerstruktur oder sogar eine mit MindManager erzeugte Mind Map zu importieren.

Die Exportmöglichkeiten bringen eine reiche Vielzahl an Dateiformaten mit sich. So kann man die Mind Map in die Graphikformate JPEG und PNG umwandeln oder in das plattformübergreifende Dateiformat PDF. Des Weiteren ist es möglich, die Mind Map

in ein Open Office – Dokument oder in eine skalierbare Vektorgraphik, SVG, umzuwandeln. Und man kann sogar die Mind Map in HTML oder XHTML exportieren.

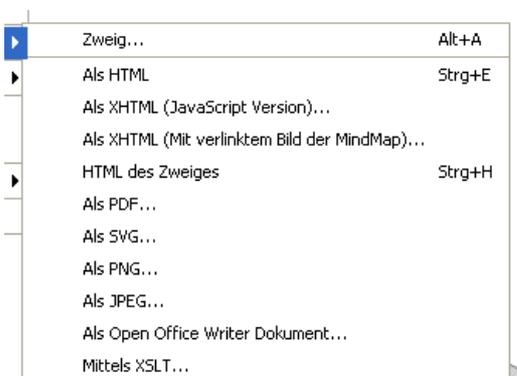


Abb. 11: Übersicht Importmöglichkeiten

Abb. 12: Übersicht Exportmöglichkeiten

Unter dem Menüpunkt „Bearbeiten“ findet man die gängigen Operationen, wie man sie auch von anderen Anwendungen kennt. Man kann schon gemachte Schritte wiederherstellen oder rückgängig machen, ebenso besteht die Möglichkeit die Mind Map oder einzelne Zweige zu markieren. Es ist möglich, Knoten auszuschneiden, zu kopieren oder einzufügen oder sogar die Formatierungen zu kopieren und einzufügen. Ebenso kann man wie erwartet, die Knoten bearbeiten und somit auch löschen. Schlussendlich kann man auch nach selbst eingegebenen Wörtern suchen.

Bei „Ansicht“ kann man Werkzeugleisten ein- und ausblenden oder die Mind Map verkleinern oder vergrößern.

Das Menü „Navigieren“ bietet einem hauptsächlich die Möglichkeit, Knoten zu verschieben oder bestimmte Zweige oder die komplette Mind Map soweit wie möglich ein- oder auszublenden. Diese Funktionen kann man aber auch leicht durch die Maus per Drag and Drop oder Klicken auf den Knoten benutzen.

Die „Extras“ stellen einige kleine Tools zur Verfügung. So kann man hier einstellen, dass geänderte Knoten gelb hinterlegt werden und/ oder deren Änderungsverlauf gespeichert wird. Außerdem kann man markierte Knoten miteinander verbinden, verschlüsselte Knoten erzeugen oder einen Kalender öffnen, mit dem man bei Bedarf auch Uhrzeiten und Daten in die Knoten eintragen lassen kann. In der Historie sieht man, wann man was erstellt und/ oder geändert hat.

Zusätzlich kann man in diesem Menü noch den Unterpunkt „Einstellungen“ auswählen, der dann ein eigenes Fenster mit mehreren Konfigurationen anzeigt, die man nach eigenem Ermessen auch ändern kann. Hier gibt es die folgenden Unterpunkte: Umgebung, Erscheinung, Standards, Tasten, Verhalten, Html.



Abb. 13: Übersicht Extras

In dem Menüpunkt „Maps“ kann man lediglich zwischen den geöffneten Maps hin- und herschalten.

Bei „Modi“ ist es dem Benutzer erlaubt, einem entweder das Hilfesystem mittels „Browse“ anzuzeigen, oder durch das eigene Dateisystem zu browsen. Beides ist mittels einer Mind Map visualisiert.

Das Hilfesystem besteht hauptsächlich aus einer Dokumentation in Form einer Mind Map (siehe Abbildung 14). Zusätzlich dazu gibt es noch eine andere Ansicht in Form einer Indexsuche und ein Verweis zu den FAQ's auf der Internetseite von FreeMind.

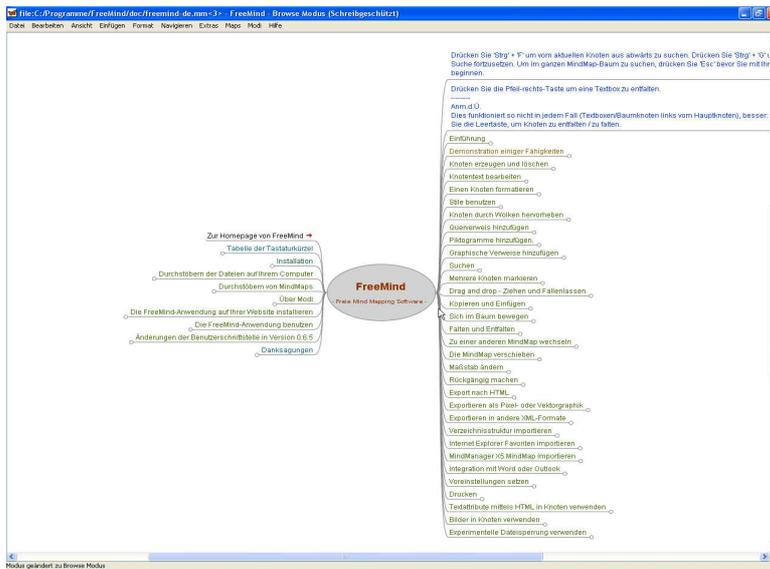


Abb. 14: Übersicht Hilfesystem

3.1.2 Fazit

Nachdem nun die wichtigsten Funktionen von FreeMind verwendet und vorgestellt wurden, kann man sagen, dass diese Anwendung eine durchaus gelungene Freeware darstellt. Auf den ersten Blick fehlen keine wichtigen Funktionen, um Mind Maps zu

erstellen. Allerdings muss man auch dazu sagen, dass FreeMind das erste Programm ist, das im Zusammenhang mit Mind Maps evaluiert wurde. Eventuell hat MindManager aufgrund der Kostenpflichtigkeit einen noch merklicheren Leistungsumfang und stellt Funktionen zur Verfügung, die einem bei der Erstellung von Mind Maps in FreeMind in Zukunft fehlen würden.

FreeMind ist sehr intuitiv zu bedienen. Man kann durch Rechtsklick mit der Maus auf den markierten Knoten alle wichtigen Funktionen nutzen, und sowohl die komplette Mind Map als auch einzelne Knoten leicht durch die Maus verschieben. Einzelne Zweige können je nach Bedarf auf- bzw. zugeklappt sein. Das Einzige, was in der Handhabung gestört hat, war die Tatsache, dass man zwei Knoten nicht wieder zu einem zusammen laufen lassen kann.

Des Weiteren waren die Gestaltungsmöglichkeiten der Mind Map ausreichend, wenn auch teilweise recht einfach gehalten. Das Programm war deswegen aber auch sehr übersichtlich.

Was sehr positiv aufgefallen ist, war die Vielzahl an Import- und Exportmöglichkeiten, um anderen die Mind Map zugänglich zu machen. So können Anwender, die eine Mind Map im MindManager – Format erhalten, die Mind Map mit FreeMind öffnen und daran weiterarbeiten. Außerdem kann man die Mind Map für sehr viele Präsentationsmöglichkeiten im Web oder in Folien, usw. exportieren.

Ebenso sehr gut war die Tatsache, dass man Hyperlinks in Knoten hinterlegen kann, das macht die Verwendung von Mind Maps noch nützlicher, um gewisse Arbeiten und Aufgaben gut zu strukturieren. Zusätzlich dazu kann man eigene Graphiken laden, allerdings besteht leider keine Möglichkeit, diese geeignet in FreeMind zu bearbeiten.

Das Hilfesystem ebenfalls als Mind Map zu strukturieren, ist eine interessante Idee. Allerdings ist die Hilfe nicht ganz so umfangreich, wobei man auch dazu sagen muss, dass vieles sehr intuitiv zu bedienen ist und man die Hilfe in der Regel recht wenig benötigen sollte. Außerdem gibt es zusätzliche Unterstützung im Internet auf der FreeMind – Homepage durch Foren, FAQ´s, usw.

Abschließend kann man sagen, dass FreeMind ein sehr gelungenes Tool zur Erstellung von Mind Maps darstellt und seine Stärken ganz klar darin liegen, dass es sowohl kostenlos als auch für unterschiedliche Plattformen nutzbar ist.

3.2 MindManager

MindManager ist ein kostenpflichtiges Programm, das ebenfalls der Erstellung von Mind Maps dient. Momentan ist es laut Herstellerseite Mindjet die führende Software auf diesem Gebiet. Ursprünglich konnte die Software nur von Anwendern genutzt werden, die das Betriebssystem Windows benutzten, mittlerweile ist es ebenso möglich als Mac- User Gebrauch von der Software zu machen. Das Programm kam erstmals 2001 auf den Markt und befindet sich zum jetzigen Zeitpunkt in der Version 6.

Das Programm ist ebenso wie FreeMind in zwei Versionen erhältlich, einer etwas abgeschwächten Version MindManager Basic, und einer Version mit vollem Funktionsumfang, MindManager Pro. Allerdings ist es an dieser Stelle nicht möglich, beide Versionen zu evaluieren, da nur eine der beiden Ausführungen als Testversion zur Verfügung steht, so dass in dieser Arbeit nicht explizit gesagt werden kann, wo genau die Unterschiede liegen.

Wie schon in FreeMind, wurde auch hier zu Beginn der Einarbeitung in das Programm versucht, die Arbeitsschritte, die im Laufe der Studienarbeit anfallen, ansatzweise in einer Mind Map darzustellen, ohne das Programm vorher genauer anzuschauen. Hier merkt man jedoch schnell einen relativ großen Unterschied zu der kostenfreien Alternative FreeMind. Das Programm ist viel umfangreicher gestaltet, so dass der Einstieg aufgrund der Fülle der Funktionen und Möglichkeiten etwas schwerer fällt. Dennoch kann man relativ schnell einen Überblick über die wichtigsten Optionen gewinnen.

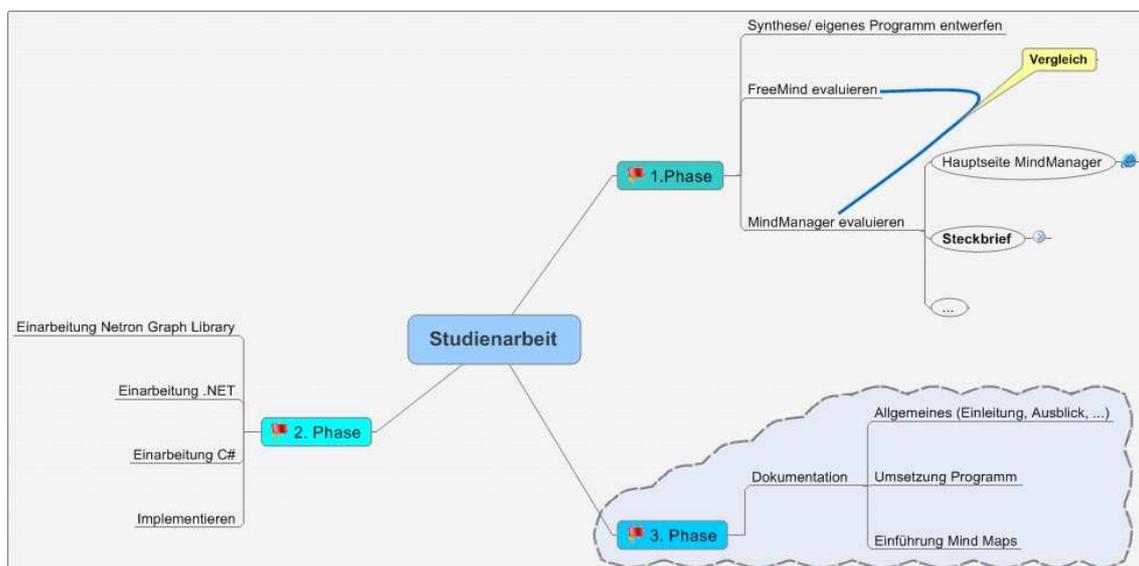


Abb. 15: In MindManager erstellte Mind Map

3.2.1 MindManager Einführung

Bei Programmstart sieht der Bildschirm wie in Abbildung 16 aus. Es wird einem eine Vielzahl an Möglichkeiten geboten. So kann man sich als „Beginner“ erst einmal eine Reihe von Lernvideos ansehen, um die Grundlagen im Umgang mit dem Programm zu lernen. Des Weiteren bietet sich dem Benutzer die Möglichkeit, die Unterschiede der neuen MindManager 6 Version im Gegensatz zu älteren Versionen anzeigen zu lassen. Außerdem kann man sich noch eine Präsentation über die Vorteile des Mind Mappings ansehen.

Neben diesen zusätzlichen Funktionen sind natürlich auch die üblichen Funktionen zum Erstellen von Mind Maps oder dem Öffnen bestehender Mind Maps gegeben.

Was man ebenfalls noch in der Abbildung erkennen kann, sind die Werkzeugleisten am oberen, unteren und rechten Bildschirmrand. In der oberen Leiste sind wieder, ähnlich wie in FreeMind, die gängigsten Funktionen für den schnellen Zugriff hinterlegt.

Zusätzlich dazu kann man einige wichtige Funktionen über die linke bzw. rechte Maustaste benutzen.

Die linke Maustaste dient dem Anlegen eines neuen Hauptzweigs, der dann an gewünschter Position mit der Wurzel verbunden wird.

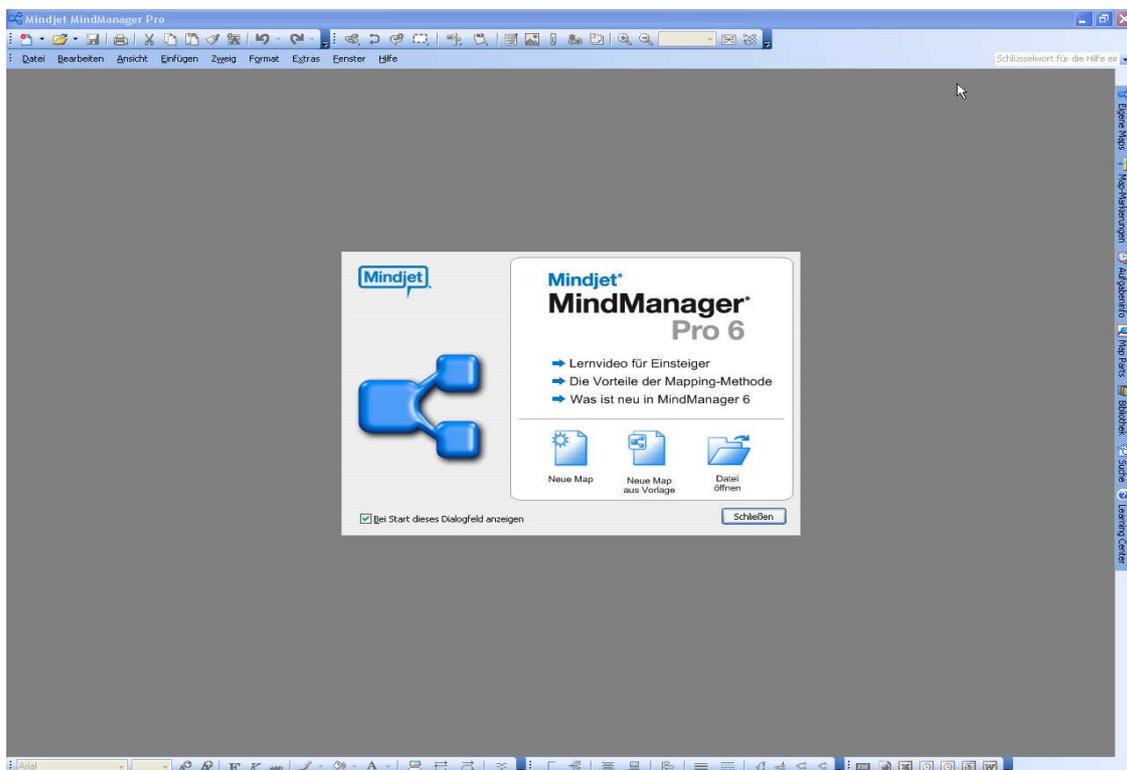


Abb. 16: Startzustand MindManager

Mit dem rechten Mausklick kann man zum Einen die Rechtschreibprüfung aufrufen, wenn sich der Cursor im Wort des Knotens befindet, oder zum Anderen mit einem Rechtsklick auf dem jeweiligen Knoten die in Abbildung 17 abgebildeten Funktionen nutzen.

Zu den Funktionen gehören, die Zweige je nach Bedarf zu verschieben, auszublenden, zu löschen und zu formatieren (Form, Farbe, Anordnung). Zusätzlich dazu kann man verschiedene Objekte einfügen, unter anderem z.B. Bilder/ Icons, Notizen, Anhänge und Hyperlinks oder Anhänge hinterlegen.

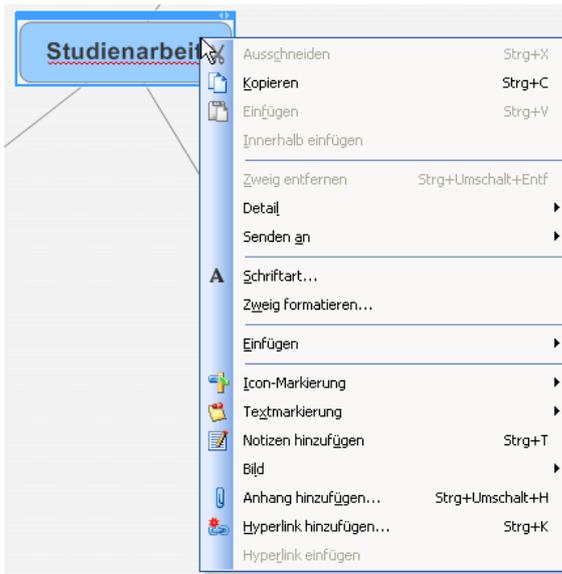


Abb. 17: Menü durch Rechtsklicken auf Knoten

In Abbildung 18 kann man die obere Werkzeugleiste und die Menüleiste von MindManager erkennen.

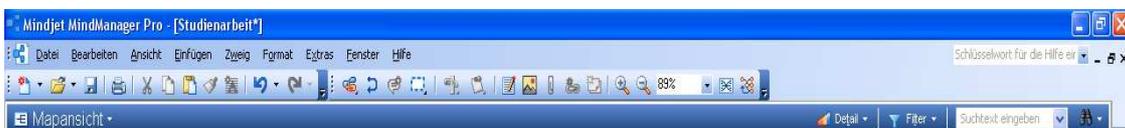


Abb. 18: Startzustand MindManager

In der Werkzeugleiste sind wieder Funktionen für den schnellen Zugriff abgelegt. Diese sind größtenteils die gleichen wie in dem zuvor erläuterten Menü, das man per Rechtsklick auf einen Knoten erreichen kann. Außerdem kann man hier in dieser Leiste noch die Funktion nutzen, die Mind Map herein- bzw. herauszuzoomen bzw. gängige Dateifunktionen (neue Datei anlegen oder eine bestehende öffnen oder z.B. eine Datei speichern bzw. ausdrucken) nutzen.

Die Menüleiste ist ähnlich wie die in FreeMind, bzw. wie in anderen Programmen auch. Die gängigsten Funktionen werden nun im Folgenden etwas näher erläutert.

Das Dateimenü enthält neben den bekannten Funktionen zur Dateiverwaltung auch die interessanten Menüpunkte Importieren, Exportieren. Ebenso besteht die Möglichkeit, Dateien zu verschlüsseln oder direkt per E-Mail zu verschicken.

Die Importmöglichkeiten beziehen sich in diesem Fall sehr stark auf das Betriebssystem Windows. Und zwar ist dem Anwender hier die Möglichkeit gegeben, aus einer Microsoft Word – Datei, aus dem Programm Microsoft Outlook oder aus einer MPX-Datei, Daten zu importieren.

MPX steht für Microsoft Project Exchange, diese Dateien beinhalten unformatierten Text, die für den Datenaustausch zwischen Microsoft Software und anderen Projektplanungsprogrammen genutzt werden können.

Bei den Exportmöglichkeiten gibt es relativ viele Möglichkeiten. So kann man zum Einen eine Mind Map wieder in das MPX-Format oder für eine Microsoft Anwendung (Outlook, Powerpoint, Word) umwandeln. Oder man hat zum Anderen noch die Möglichkeit gegeben, die Mind Map in ein PDF-, Bild- (sowohl Pixel- als auch Vektorgraphik) oder Html-Format zu exportieren. Außerdem kann man die Mind Map in ein älteres MindManager-Format umwandeln.

In dem „Bearbeiten“-Menü gibt es neben den Standardfunktionen (markieren, ersetzen, suchen, löschen, usw.) als interessanten Punkt noch den Power-Filter zu nennen. Dieser ermöglicht einem nach bestimmten Kriterien zu suchen, so z.B. nach einer Füllfarbe (vgl. Abb. 19).

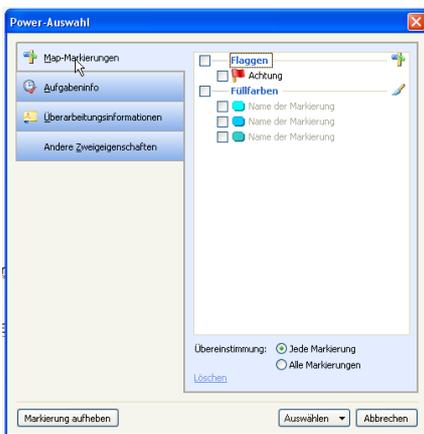


Abb. 19: Powerfilter MindManager

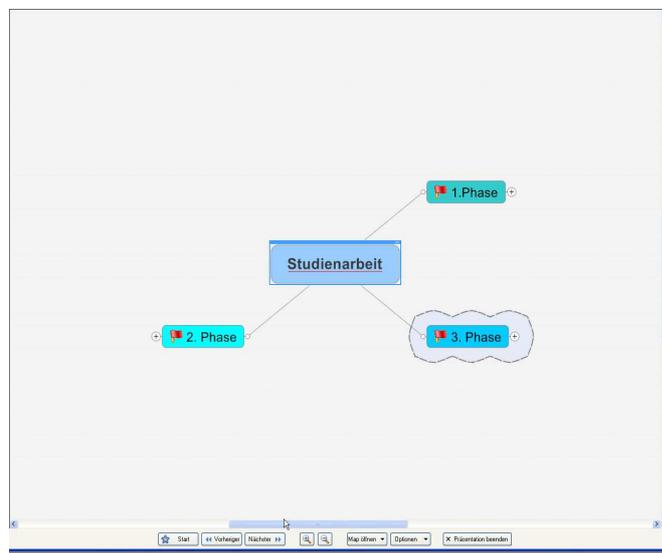


Abb. 20: Präsentationsansicht MindManager

Die interessanteste Funktion im Ansichtsmenü ist neben dem automatischen Anordnen der Map und dem Ein- und Ausblenden von Ebenen usw., der Unterpunkt „Präsentation“ (siehe Abb. 20). Diese Funktion erlaubt es einem, die Mind Map in Form einer Präsentation Schritt für Schritt vorzustellen. So wird z.B. in dieser Mind Map zuerst die nächste Ebene von der 1. Phase (vgl. Abb. 20) eingeblendet und dann jeder Knoten davon pro „Folie“ hervorgehoben. So wird die erstellte Graphik Schritt für Schritt durchlaufen.

In dem Menü Einfügen kann man entweder neue Zweige bzw. Unterzweige, Anmerkungen oder Verbindungen einfügen oder externe Mind Map – Teile usw. einbinden.

Auch das Zweigmenü bietet eine Reihe von Funktionen an, um bestimmte Objekte, wie Bilder, Tabellen, Erinnerungen einzufügen. Es besteht sogar die Möglichkeit, Tabellen in einen Knoten einzubinden.

Bei dem Formatmenü besteht neben den möglichen Knotenformatierungen sogar die Möglichkeit, aus der Mind Map eine Microsoft PowerPoint – Präsentation zu machen oder bestimmte MPX – Einstellungen vorzunehmen. Außerdem kann man Zweige nach dem Alphabet sortieren oder aus verschiedenen Stilen für Mind Maps wählen.

Im Extramenü werden sehr viele Möglichkeiten zur zusätzlichen Bearbeitung der Mind Map angeboten.

Hier gibt es eine Rechtschreibprüfung, zusätzlich kann man verschiedene Sprachen auswählen, die man im Dokument verwendet. Ebenso ist es möglich eine Autokorrektur vorzunehmen oder beschädigte Datei- Hyperlinks zu reparieren. Zusätzlich dazu kann man viele Modifikationen an den Programmeinstellungen vornehmen, z.B. welche Funktionen beim Start zur Verfügung stehen usw.

Gut ist auch der Brainstorming- (siehe Abb. 21), bzw. der Überarbeitungsmodus (siehe Abb. 22).



Abb. 21: Brainstorming – Option in MindManager



Abb. 22: Überarbeitung – Option in MindManager

Diese beiden Funktionen ermöglichen es einem in der Gruppe leichter Ideen zu visualisieren, bzw. einfacher Mind Maps durch andere Anwender modifizieren zu lassen und dann wieder weiterleiten zu können.

Im Fenstermenü kann man lediglich zwischen verschiedenen Ansichten wählen.

Das Hilfemenü wiederum stellt viel Funktionalität zur Verfügung. So hat man die Möglichkeit, einmal die sehr umfangreiche „Hilfe für Mindjet MindManager“ aufzurufen, die eine Indexsuche, Suche nach Stichwörtern und eine Reihe an vorgefertigten Infothemen bereitstellt. Außerdem gibt es noch das Learning – Center, welches eine Vielzahl von Lernvideos, Tipps und Tricks und Beispielgalerien bereitstellt. Des Weiteren kann der User Gebrauch vom Online – Support, Kunden – Feedback oder der Prüfung auf neue Updates machen.



Abb. 23: Untere Werkzeugleiste

Die untere Werkzeugleiste enthält einige sehr nützliche Funktionen um die Mind Map optisch zu verbessern. Man kann u. a. die Schriftarten und Schriftgröße auswählen, den Schrifthintergrund und die Schriftfarbe bestimmen bzw. auch die Knoten-, Pfeil- und Linienformen bzw. – muster verändern. Ebenso kann man die Ausrichtung von allen Linien, Texten und Zweigen bestimmen oder Abstände zwischen Zweigen/ Knoten vergrößern bzw. verkleinern. Schlussendlich stehen noch einige Exportfunktionen für den schnellen Zugriff bereit.



Abb. 24: Linke Werkzeugleiste

Abschließend wird noch die linke Werkzeugleiste kurz erwähnt. Und zwar kann man sich hier die eigenen Maps anzeigen lassen, eine Auswahl an Map – Markierungen (Smileys, Icons, ...) anschauen und auswählen, Aufgaben definieren oder vorgefertigte Map Parts zur aktuellen Mind Map hinzufügen. Der Unterpunkt Bibliothek umfasst alle vorhandenen Map- Markierungen, Bilder, Hintergrundbilder und Formen. Außerdem kann man eine ausführliche Suche in Ordnern bzw. in Mind Maps starten und zu guter letzt ist das bereits erwähnte Learning Center als Punkt aufgeführt.

3.2.2 Fazit

MindManager ist ein sehr umfangreiches Tool zur Erstellung von Mind Maps, das noch Einiges mehr kann. Die Einarbeitung in dieses Programm fiel trotz des Funktionsumfangs relativ leicht. Man hat schnell einen Überblick über die wichtigsten Funktionen. Selbst für einen Anfänger, für den der Umgang mit dem PC nicht alltäglich ist, wird durch die Einsteigervideos die Möglichkeit gegeben, sich schnell in MindManager zu Recht zu finden.

Gut war besonders die vielen Gestaltungsmöglichkeiten bzw. die vielen Vorlagen. Auch die Tatsache, dass man erstellte Mind Maps direkt in eine PowerPoint – Präsentation umwandeln kann, ist sehr nützlich. Das Programm stellt auch sonst eine Reihe von Extras zur Verfügung, die man besonders gut im späteren Arbeitsleben und in der Arbeit im Team nutzen kann, so zum Beispiel das Brainstorming - Modul.

Ein Nachteil ist mit Sicherheit die Tatsache, dass das Programm anfangs nur für Windows Betriebssysteme zur Verfügung stand. Mittlerweile ist es zwar auch für MAC verfügbar, aber wohl noch nicht in komplettem Funktionsumfang, wobei dieser Aspekt hier in dieser Arbeit nicht getestet wurde (vgl. [3]_M).

Das Programm ist allerdings für den „Privatmann“ relativ teuer und wohl auch nicht notwendig (Basic – Version ca. 230 €). Was mir hier wie auch zuvor in FreeMind, aufgefallen ist, ist die Tatsache, dass man keine direkte Verbindung zwischen zwei Knoten herstellen kann, wiederum ist es nur eine Bézier - Kurve, die eine Verbindung möglich macht.

Nichts desto trotz stellt MindManager ein gelungenes Produkt dar, dass gerade für das Geschäftsleben eine nützliche Anwendung darstellt.

3.3 Vergleich

Die beiden Programme haben überzeugt, und sowohl MindManager als auch FreeMind haben ihre Vor- und Nachteile.

FreeMind hat seine Stärken definitiv darin, plattformunabhängig und kostenlos zu sein. Außerdem ist der Zugang zum Programm recht intuitiv, so dass man keine langen Einarbeitungszeiten einberechnen muss. Die Funktionalität ist durchaus ausreichend, um Mind Maps in vollem Funktionsumfang darstellen zu können. Lediglich die Anzahl der verschiedenen Gestaltungsmöglichkeiten fällt im direkten Vergleich mit MindManager etwas kleiner aus.

Im Gegensatz zu FreeMind hat MindManager einen klaren Nachteil aus Sicht des Anwenders, da beide Versionen sowohl kostenpflichtig als auch nicht komplett plattformunabhängig sind. MindManager hat seinen Schwerpunkt nach wie vor ganz klar auf das Betriebssystem Windows ausgerichtet, was man auch an den sehr stark an Windows orientierten Import- und Exportmöglichkeiten sehen kann. Allerdings muss man auch sagen, dass die erste Version für den MAC bereits vorhanden ist, wenn auch noch nicht in vollem Funktionsumfang.

Der Programmumfang im Allgemeinen übersteigt den von FreeMind um ein Vielfaches. Das Hilfesystem ist weitaus ausführlicher und auch die schon erwähnten Gestaltungsmöglichkeiten lassen viel Raum. Des Weiteren werden noch einige zusätzliche Funktionalitäten zur Verfügung gestellt, so zum Beispiel der Brainstorming - oder der Präsentationsmodus, um nur einige zu nennen.

Auch die Einarbeitung wird einem durch zahlreiche Hilfestellungen (Videos, Tipps und Tricks usw.) erleichtert. Aufgrund seines größeren Programmumfangs ist es eben nicht ganz so schnell zu erschließen wie FreeMind.

Was man an beiden Programmen kritisieren könnte, ist die Tatsache, dass man zwei aus unterschiedlichen Zweigen stammende Knoten nicht mehr zusammenführen kann.

Beide Programme stellen hier aber zumindest die Option zur Verfügung, Knoten durch graphische Verbindungen, in beiden Fällen Bézier – Kurven, miteinander zu verbinden.

Abschließend kann man sagen, dass für den privaten Gebrauch ganz klar FreeMind zu empfehlen wäre, da es völlig ausreichend ist. MindManager hingegen wäre empfehlenswert, wenn es darum geht, ein Mind Mapping – Programm auf professioneller Ebene zu benutzen. Es überzeugt optisch, aber natürlich auch aufgrund seiner Funktionen und kann gut im Arbeitsleben, besonders beim Arbeiten im Team oder bei Präsentationen genutzt werden. Außerdem sollte der Preis in diesem Rahmen einigermaßen angemessen sein.

4. Design des Gesamtsystems

Im folgenden Kapitel soll die Idee des Programms und dessen Hilfsmittel, die für die Erstellung benötigt werden, etwas näher erläutert werden.

4.1 Systemeinsatz und Systemumgebung

Für die Entwicklung des Mind Mapping – Programms wird Visual Studio .NET 2005 verwendet. Die Programmiersprache wird C# sein. Der eigentliche Aufbau der Mind Map in der Anwendung wird durch ein Framework zur Entwicklung von Graphen unterstützt, und zwar wird hier dafür die Netron Graph Library verwendet.

Die Entwicklung des Programms findet auf Windows XP – Betriebssystemen statt.

4.2 Funktionale Anforderungen

Die folgenden beiden Unterkapitel beschreiben zum Einen die Punkte, die auf jeden Fall im Funktionsumfang der Anwendung enthalten sein müssen und zum Anderen die Punkte, die wünschenswert im Programmumfang wären. Allerdings ist zunächst das Hauptaugenmerk darauf gerichtet, ein funktionierendes Gesamtsystem zu entwickeln, dass je nach dem noch um einige Extras erweitert werden kann.

4.2.1 Musskriterien

- lauffähig auf Windows XP
- Bedienung über GUI
- Mind Maps neu erstellen können
- Wurzelknoten erstellen können
- Unterknoten erstellen können
- Mind Maps öffnen (Dateibrowser)
- Mind Maps bearbeiten (Knoten löschen, Knoteneigenschaften ändern)
- Mind Maps speichern (XML)

4.2.2 Wunschkriterien:

- Mind Maps drucken
- Importmöglichkeiten (z.B. Mindmanager, FreeMind)
- Exportmöglichkeiten (PDF, Vektorgrafiken, ...)

- Exportmöglichkeiten (Pixelgrafiken)
- Verbindungen zwischen 2 Knoten sollten auch möglich sein (vielleicht Bézier)
- Bilder / Icons sollten eingebunden werden können
- Bilder sollten auch bearbeitet werden können
- Knoten bearbeiten (suchen, Format kopieren/löschen)
- Werkzeugmenüs mit wichtigsten Funktionen anbieten
- Menüs ein-/ausblenden
- Mind Map verschieben per Drag and Drop

4.3 Nichtfunktionale Anforderungen

An das Mind Mapping - Programm werden die folgenden nichtfunktionalen Anforderungen gestellt, die bis zur Fertigstellung möglichst optimal realisiert worden sein sollen:

- Portabilität:
Das System muss auf Windowsrechnern lauffähig sein.
- Stabilität:
Durch eine weitestgehend geführte Eingabe seitens des Anwenders, werden Fehler bereits im Vorfeld abgefangen. Dadurch sollte das System stabil arbeiten können.
- Benutzerfreundlichkeit:
Das System wird leicht über eine graphische Benutzeroberfläche zu bedienen sein.
- Wartbarkeit:
Das System ist ausführlich dokumentiert und besitzt eine übersichtliche Programmstruktur, welche die Wartung erleichtert.

4.4 Benutzerschnittstellen und Fehlerverhalten

Die Bedienung des Programms erfolgt durch den Benutzer mittels eines Graphical User Interfaces (GUI), über das der komplette Ablauf gesteuert werden kann.

Das System soll bei nicht korrekter Handhabung eine Fehlermeldung an den Benutzer ausgeben. Eine nähere Fehlerbeschreibung lässt sich dem dokumentierten Source – Code entnehmen.

4.5 Dokumentationsanforderungen

Zur Projektdokumentation zählen diese Dokumentation und die dokumentierten Source – Codes. Diese können leicht aufgrund des verwendeten XML- Formats im Source – Code durch Visual Studio .NET in ein HTML – Dokument extrahiert werden.

Das Dokument wird in Microsoft Word erstellt und bei Fertigstellung als PDF – Dokument abgegeben. Die Abschlusspräsentation, falls nötig, wird mit Microsoft PowerPoint erstellt.

Die Arbeit erfolgt über einen SVN – Zugang, wo sowohl die Dokumente als auch die Source – Codes zur Bearbeitung abgelegt werden.

Die Dokumentation erfolgt ausschließlich in deutscher Sprache.

4.6 Abnahmekriterien

Die funktionalen Anforderungen, die hier im Dokument festgehalten sind, müssen erfüllt werden. Außerdem muss die Dokumentation die Arbeit und die Entwicklung dieser Studienarbeit vollständig dokumentieren und wie oben angegeben gestaltet sein. Die nichtfunktionalen Anforderungen sollten möglichst optimal gelöst werden.

5. Umsetzung

Zu Beginn der Implementierungsphase wurde auch direkt die grafische Oberfläche erstellt, die dann nach und nach um Funktionalitäten erweitert wurde. Im rechten Fenster wurde die sogenannte *Graph Control* von Netron eingebunden. Diese beinhaltet grundlegende Funktionen, wie etwa das Markieren und das automatische Verschieben von Knoten. Zusätzlich dazu gehören auch ein bereits implementierter Layoutalgorithmus und ein Kontextmenü.

Des Weiteren stellt Netron unter anderem auch eine Basisklasse *Shape* zur Verfügung, von der man dann eigene Knotenformen ableiten kann. Diese Knoten erben dann schon einige nützliche Eigenschaften, wie zum Beispiel das automatische Vergrößern/Verkleinern des Knotens oder die Eigenschaft, ob der Knoten verschoben werden kann oder ob er fest an einem bestimmten Punkt bleiben soll.

Außerdem kann die Farbe der Schrift oder des Hintergrundes, genauso die Schriftgrößen und Schriftarten verändert werden, Text hinzugefügt oder URL's hinterlegt werden, um nur einige von den zahlreichen mitgebrachten Features zu nennen.

Die mittlerweile fertiggestellte graphische Oberfläche ist in Abbildung 25 zu sehen. Das Fenster an sich wurde von der Oberklasse *System.Windows.Forms.Form* abgeleitet und trägt den Namen *MindMap*. Die konkreten Möglichkeiten und Implementationen, die diese Klasse *MindMap* mit sich bringt, werden nun im Folgenden explizit erläutert.

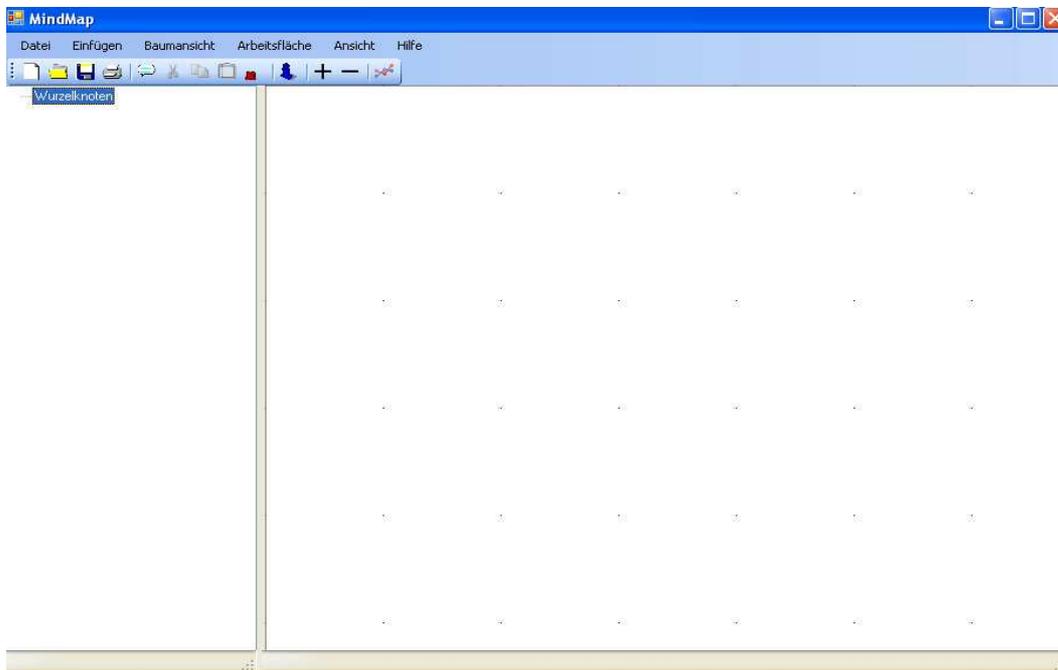


Abb. 25: Startzustand MindMap

Das bereits von Netron implementierte Kontextmenü wurde abgeschaltet und durch ein neues angepasstes Kontextmenü ersetzt. Dieses ist per rechtem Mausklick im Arbeitsfenster zu erreichen. Die dort angezeigten Funktionen (vgl. Abb. 26) sind die gleichen wie die des Menüpunktes Arbeitsfläche und werden dort auch erst erläutert.



Abb. 26: Kontextmenü MindMap

Für den schnellen Zugriff auf häufig verwendete Funktionen wurde ebenso eine Schnellstartleiste angelegt, die man nach seinen Wünschen im Programmfenster verschieben kann. Diese Funktionen sind im Einzelnen (vgl. Abb. 27 von links nach rechts):

- eine neue Mind Map anlegen
- eine bestehende Mind Map öffnen
- eine momentan bearbeitete Mind Map speichern
- eine momentan bearbeitete Mind Map drucken
- einen neuen Knoten auf der Arbeitsfläche (rechtes Fenster) hinzufügen
- markierte Knoten auf der Arbeitsfläche ausschneiden (leider noch nicht vollständig implementiert)
- markierte Knoten auf der Arbeitsfläche kopieren (leider noch nicht vollständig implementiert)
- markierte Knoten auf der Arbeitsfläche einfügen (leider noch nicht vollständig implementiert)
- markierte Knoten auf der Arbeitsfläche löschen
- die Hilfedatei zu dem Programm MindMap aufrufen
- die Mind Map auf der Arbeitsfläche vergrößern
- die Mind Map auf der Arbeitsfläche verkleinern
- den Netron Tree – Layout – Algorithmus auf die Mind Map anwenden



Abb. 27: Schnellstartleiste MindMap

Die Funktionen sind aus verschiedenen Menüs entnommen und werden im dazugehörigen Menü detailliert erläutert.

In Abbildung 28 sieht man die linke Fensterseite des Programms. Diese beinhaltet eine sogenannte *Treeview*. Dies ist eine von .NET bereitgestellte Baumansicht, die bereits einige vorgefertigte Implementationen mit sich bringt. Funktionen, die das Programm MindMap nutzt, sind unter anderem das Zugreifen auf Knotenelemente des erstellten Baumes oder das Hinzufügen oder Löschen von einzelnen Knoten.



Abb. 28: Baumansicht links MindMap

Mittels einer Traversierung können die Knoten, während der Baum durchlaufen wird, mit Hilfe eines eigens entwickeltem zirkularem Layoutalgorithmus, im rechten Fenster angeordnet werden. Für die oben gesehene Baumstruktur sieht die dazu gehörige Anordnung der Knoten auf der Arbeitsfläche wie in Abbildung 31 aus.

Das Programm läuft wie folgt ab. Will der Anwender die Baumstruktur auf die Arbeitsfläche übertragen und somit die Mind Map zeichnen, wird ein *Event* ausgelöst, welches zu zwei unterschiedlichen Möglichkeiten führt. Entweder kann der Nutzer sich dazu entschließen, das Layout von Netron nutzen oder eben den hier erwähnten Algorithmus. Entscheidet sich der Anwender für diesen Algorithmus, wird die Methode *UebertrageMethodeZwei()* aufgerufen, andernfalls *UebertrageMethodeEins()*. Ausgewählt werden die Methoden mittels einer *MessageBox*, wie in Abbildung 29 zu sehen ist.

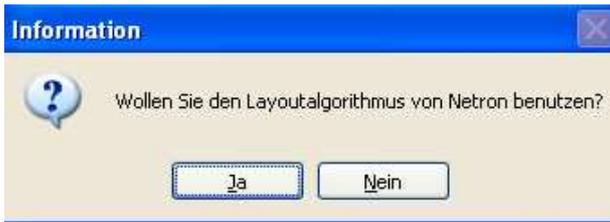


Abb. 29: Layoutabfrage

Folgender Code liegt dem Algorithmus zugrunde. Dieser wird nun Schritt für Schritt erklärt.

```
private void UebertrageMethodeZwei()
{
    if (this.treeView1.Nodes.Count != 0)
    {
        LauraForm wurzel = new LauraForm();
        this.graphControl1.AddShape(wurzel);
        wurzel.X = (int) posX;
        wurzel.Y = (int) posY;
        wurzel.Text = this.treeView1.Nodes[0].Text;
        this.graphControl1.Refresh();
        int geschwisterZaehler = this.treeView1.Nodes[0].Nodes.Count - 1;

        if (this.treeView1.Nodes[0].Nodes.Count > 0)
        {
            double layoutHilf =
                this.treeView1.Nodes[0].Nodes.Count;
            double platzKinderKreis = 360 / layoutHilf;

            uebertrageKinder(this.treeView1.Nodes[0].FirstNode, wurzel,
                geschwisterZaehler, platzKinderKreis, 1, 0, 360, 100);
        }
    }

    else
    {
        MessageBox.Show("Es müssen Knoten angelegt sein, um sie übertragen zu können!");
    }
}
```

Zuerst wird geprüft, ob die Baumstruktur (*treeView1*) überhaupt Knoten beinhaltet. Wenn ja, wird zuerst einmal in der Mitte der Arbeitsfläche ein Wurzelknoten vom Typ *LauraForm* angelegt. Anschließend erfolgt eine Abfrage, ob der Wurzelknoten Kinder hat. Wenn ja, wird berechnet, wieviele Kinder das sind und wieviel Platz diese im Umkreis von 360° jeweils einnehmen können. Das heißt zum Beispiel, dass bei einer Kinderanzahl von 4 jedes Kind 90° „Platz“ zur Verfügung steht. Danach wird die Methode *uebertrageKinder()* aufgerufen.

Diese Methode bekommt sehr viele Parameter übergeben. Zum Einen werden der jeweilige Baumknoten, der übertragen werden soll und der dazu neu angelegte korrespondierende *LauraForm* – Knoten mitgeliefert. Zum Anderen werden verschiedene Parameter für die Platzberechnung und die genaue Position der einzelnen Knoten benötigt, das sind die Anzahl der übrigen Geschwister, die der aktuelle Knoten hat, dann den Platz den die Kinder haben und die aktuelle Nummer des Kindes (Knoten werden durchnummeriert). Zusätzlich dazu werden noch der Anfangsgrad und der Endgrad, das heißt, der Bereich bzw. die Gradzahl, in dem dann die Kinder gezeichnet werden können und der Radius übergeben.

Beim ersten Aufruf der *uebertrageKinder()* – Methode werden also folgenden Werte übergeben:

- das erste Kind des Wurzelknotens (*this.treeView1.Nodes[0].FirstNode*)
- der neu angelegte Wurzelknoten auf der Arbeitsfläche (*wurzel*)
- die weiteren Geschwister des ersten Kindes, z.B. bei vier Kindern also noch drei (*geschwisterZaehler*)
- den Platz den die Knoten im Kreis haben, bei vier hätte jeder Knoten einen Bereich von 90° für sich, in dem er und seine Kinder wieder gezeichnet werden könnten (*platzKinderKreis*)
- die aktuelle Kindnummer, also hier die 1
- den Anfangsgrad, beim ersten Aufruf also 0°, noch sind ja keine Knoten bis auf die zentrierte Wurzel gezeichnet
- der Endgrad, also zuerst einmal 360°
- der Radius, in diesem Programm immer 100

```
private void uebertrageKinder(TreeNode knoten, LauraForm knotenName, int
anzahlUebrigerGeschwister, double layoutZaehler, int kindNummer, double anfangsGrad,
double endGrad, double radius)
{
    double layoutZaehler2 = (endGrad - anfangsGrad) / (1 + anzahlUebrigerGeschwister);

    halberWinkel = (layoutZaehler2 / 2) + (layoutZaehler2 * (kindNummer - 1)) + anfangsGrad;
    winkel = Math.PI * halberWinkel / 180.0;

    if (halberWinkel < 90.00005)
    {
        cosWinkel = Math.Cos(winkel)
        sinWinkel = Math.Sin(winkel);
        posX = (cosWinkel * radius) + 300;
        posY = (sinWinkel * radius) + 300;
    }

    else if (halberWinkel < 180.00005)
    {
        cosWinkel = - Math.Sin(winkel);
        sinWinkel = Math.Cos(winkel);
        posX = (sinWinkel * radius) + 300;
        posY = 300 - (cosWinkel * radius);
    }
}
```

```

else if (halberWinkel < 270.00005)
{
    cosWinkel = - Math.Cos(winkel);
    sinWinkel = - Math.Sin(winkel);
    posX = 300 - (cosWinkel * radius);
    posY = 300 - (sinWinkel * radius);
}

else if (halberWinkel < 360.00005)
{
    cosWinkel = Math.Sin(winkel);
    sinWinkel = - Math.Cos(winkel);
    posX = 300 - (sinWinkel * radius);
    posY = (cosWinkel * radius) + 300;
}

LauraForm kind = new LauraForm();
this.graphControl1.AddShape(kind);
this.graphControl1.AddConnection(knotenName.Connectors[0], kind.Connectors[0]);
kind.Text = knoten.Text;
kind.X = (int) posX;
kind.Y = (int) posY;
this.graphControl1.Refresh();

if (anzahlUebrigerGeschwister > 0)
{
    uebertrageGeschwister(knoten.NextNode, knotenName, anzahlUebrigerGeschwister - 1,
        layoutZaehler, kindNummer + 1, radius);
}

if (knoten.Nodes.Count > 0)
{
    radius = radius + 100;
    anzahlUebrigerGeschwister = knoten.Nodes.Count - 1;

    uebertrageKinder(knoten.FirstNode, kind, anzahlUebrigerGeschwister, layoutZaehler2,
        1, ((kindNummer - 1) * layoutZaehler), ((kindNummer) * layoutZaehler),
        radius + 100);
}
}

```

Zu Beginn des Aufrufes wird der neue Platz in Abhängigkeit vom Anfangs- und Endgrad für den übergebenen Knoten berechnet (*layoutZaehler2*). Die Variable *halberWinkel* berechnet den genauen Winkel, an dem der Knoten gezeichnet werden soll. Um bei dem Beispiel mit vier Knoten zu bleiben, hätte ja jeder Knoten 90° Platz, der erste Knoten würde dann also bei 45° gezeichnet werden (vgl. Abb. 30).

Nun muss nur noch der Winkel im Bogenmaß berechnet werden. Anschließend wird geschaut, in welchem Quadranten sich der Parameter *halberWinkel* befindet, um dementsprechend die richtige Position für die x- und y- Position berechnen zu können.

Berechnet werden die Positionen im rechtwinkligen Dreieck mit folgenden Formeln:

$$\cos \theta = \frac{\text{Ankathete}}{\text{Hypothemuse}}$$

$$\sin \theta = \frac{\text{Gegenkathete}}{\text{Hypothemuse}}$$

Um die Berechnungen des Algorithmus besser verstehen zu können, dient die nachfolgende Zeichnung zur Übersicht.

Bei einer Mind Map, deren Wurzelknoten (Koordinaten x und y liegen im Ursprung), vier Kinder hätte, würden die Positionen der einzelnen Kinder an den mit rot gekennzeichneten Kreuzen liegen.

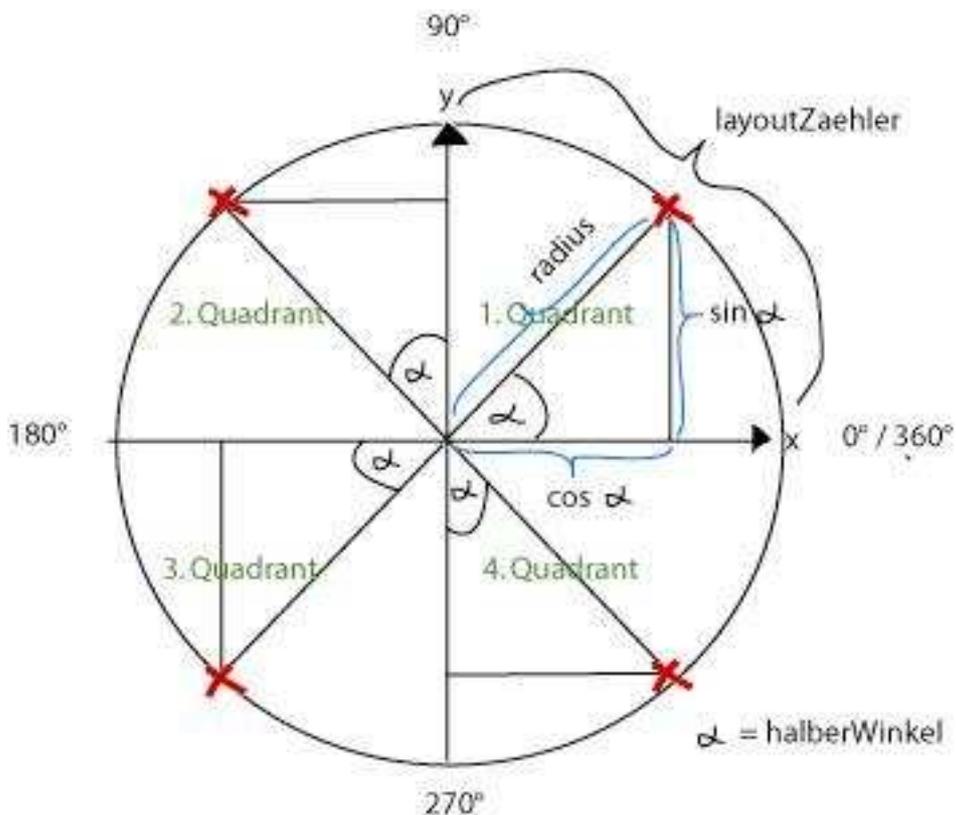


Abb. 30: Veranschaulichung Berechnungen Layoutalgorithmus

Da die Winkel aber auch 90° überschreiten und man dann die Formeln für Sinus und Cosinus nicht mehr direkt anwenden kann, da sie nur für rechtwinklige Dreiecke gelten, muss man zuerst folgende Reduktionsformeln anwenden:

$$\sin (90^\circ + \theta) = \cos \theta$$

$$\cos (90^\circ + \theta) = -\sin \theta$$

$$\sin (180^\circ + \theta) = -\sin \theta$$

$$\cos (180^\circ + \theta) = -\cos \theta$$

$$\sin (270^\circ + \theta) = -\cos \theta$$

$$\cos (270^\circ + \theta) = \sin \theta$$

Ist die Position berechnet, kann der neue Knoten *kind* gezeichnet werden und dieser mit dem übergebenen Knoten verbunden werden.

Danach wird überprüft, ob noch Kinder oder Geschwister gezeichnet werden müssen. Wenn noch Kinder vorhanden sind, wird die Funktion von neuem aufgerufen und der Radius um 100 erhöht. Ansonsten wird die *uebertrageGeschwister()* – Methode aufgerufen. Die Anzahl der Geschwister wird dann um eins erniedrigt, die Kindnummer um eins erhöht.

Die *uebertrageGeschwister()* – Methode ist größtenteils deckungsgleich wie die *uebertrageKinder()* – Methode und unterscheidet sich nur in dem Aspekt, dass kein neuer Platz berechnet werden muss, da dieser ja bereits vorher durch das erste Kind der neuen Ebene berechnet wurde und so der Platz für diese Geschwister feststeht. Dementsprechend werden der Anfangs- und Endgrad nicht mehr benötigt.

Das Ergebnis des Layout – Algorithmus bei 4 Kindern, die wiederum jeweils ein Kind haben, sieht folgendermaßen aus.

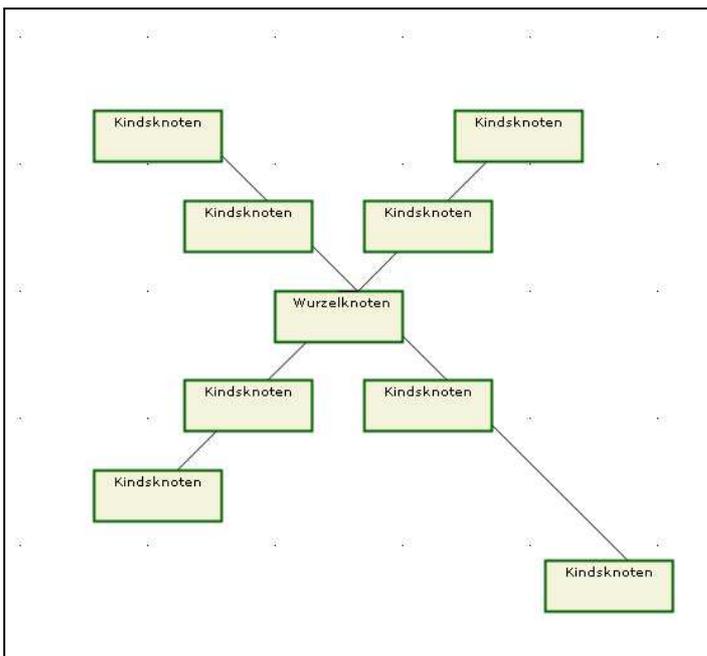


Abb. 31: Korrespondierende Arbeitsfläche rechts MindMap

Entscheidet man sich für das Netron Tree-Layout, wird die Baumstruktur wie in unten stehender Graphik (Abb. 32) angeordnet. Der Aufruf und das Zeichnen erfolgen ähnlich wie im gerade vorgestellten Algorithmus, lediglich die Positionsberechnungen entfallen hier natürlich, da sich darum in diesem Fall die Netron Graph Library und dessen TreeLayout – Algorithmus kümmert.

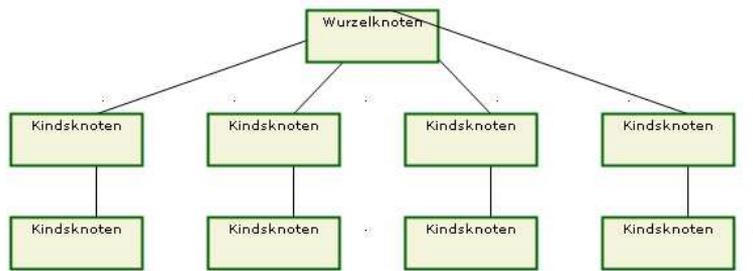


Abb. 32: Netron Tree – Layout

Im Folgenden werden nun die einzelnen Menüleisten näher erläutert. Hierzu gehören das Datei-, das Einfügen-, das Baumansicht-, das Arbeitsfläche-, das Ansicht- und das Hilfe – Menü.



Abb.33: Datei – Menü MindMap

Das Datei- Menü besitzt die üblichen Funktionen, die man auch von anderen Programmen gewöhnt ist. Man kann neue Mind Maps erstellen, bestehende Mind Maps öffnen, in zwei verschiedenen Formaten abspeichern und drucken, sowie das Programm beenden.

Bevor eine neue Mind Map erstellt werden kann, kommt zunächst eine Abfrage, ob die bestehende Mind Map gespeichert werden möchte. Wenn ja, wird zuerst die Methode `saveMM()` für das Speichern aufgerufen, ansonsten wird direkt mittels `NewDiagram(true)` ein leeres `graphControl` – Fenster aufgerufen.

Hier als Veranschaulichung der Source – Code, wie zum Beispiel mit Hilfe von Netron Mind Maps geöffnet werden und welche Methoden Netron dafür schon bereitstellt:

```
private void openMM()
{
    DialogResult res = MessageBox.Show("Wollen Sie die bestehende Mind Map vor dem Schließen speichern?", "Achtung", MessageBoxButtons.YesNoCancel, MessageBoxIcon.Question);
```

```

if (res == DialogResult.Yes)
{
    saveMM();
}

OpenFileDialog oeffnen = new OpenFileDialog();
oeffnen.Filter = "Netron diagram (*.nml)|*.nml";
oeffnen.RestoreDirectory = true;
oeffnen.InitialDirectory = "\\c:";
oeffnen.Title = "Öffnen...";

DialogResult ergebnis = oeffnen.ShowDialog();

string dateiname;

if (ergebnis == DialogResult.Cancel)
{
    return;
}

dateiname = oeffnen.FileName;

if (dateiname == "" || dateiname == null)
{
    MessageBox.Show("Ungültiger Dateiname", "Fehler", MessageBoxButtons.OK,
        MessageBoxIcon.Error);
}

else
{
    this.graphControl1.NewDiagram(true);
    this.graphControl1.OpenNML(dateiname);
}
}

```

Will man also eine bereits gespeicherte Mind Map öffnen, geschieht folgendes. Wieder wird eine Abfrage voran gestellt, die prüft, ob die momentan geöffnete Mind Map gespeichert werden soll. Danach wird ein *OpenFileDialog()* angelegt, der als *Filter* nur **.nml*, das heißt Netron – XML Dateien, anzeigt. Ist die vom Nutzer ausgewählte Datei eine gültige **.nml* – Datei wird mittels *NewDiagram(true)* ein neues Fenster angelegt, in dem dann mit Hilfe des Befehls *OpenNML(dateiname)* diejenige Datei auch geöffnet wird.

Der Speichervorgang einer **.nml* – Datei geht fast genauso vonstatten. Es wird ein Dateibrowser mittels *SaveFileDialog()* angelegt, in dem man dann **.nml* – Dateien abspeichern kann. Sind die Eingaben korrekt, kann der Speichervorgang mit Hilfe des Befehls *SaveNML(dateiname)* vorgenommen werden.

Auch das Exportieren in das **.jpg* – Dateiformat läuft genauso ab, nur werden jetzt durch einen *Filter* nur **.jpg* - statt **.nml* - Dateien zugelassen und der Befehl *SaveNML(dateiname)* wird durch *SaveImage(dateiname, true)* ersetzt. Der zweite

Parameter ist ein *bool* und bestimmt, ob der Hintergrund gezeichnet werden soll oder nicht.

Der letzte Befehl, der auf die Mind Map angewendet werden kann, ist das Ausdrucken.

```
private void printMM()
{
    PrintDocument dokument = new PrintDocument();
    dokument.PrintPage += new PrintPageEventHandler(this.graphControl1.PrintCanvas);

    PrintPreviewDialog vorschau = new PrintPreviewDialog();
    vorschau.Document = dokument;
    vorschau.ShowDialog(this);
    return;
}
```

Zuerst muss ein *PrintDocument* angelegt werden. Diesem *document* (also die zu druckende Mind Map) wird dann ein *PrintPageEventHandler()* mit dem aktuellen *graphControl* – Fenster (*this.graphControl1.PrintCanvas*) zugewiesen.

Dann braucht man nur noch eine Druckvorschau, die mittels *PrintPreviewDialog* angelegt wird und das *document* zugewiesen bekommt. Schon kann der Anwender eine Druckvorschau angezeigt bekommen und die Mind Map drucken.



Abb.35: Einfügen – Menü MindMap

Das Einfügen- Menü sollte ursprünglich zwei verschiedene Funktionen zur Verfügung stellen. Zum Einen sollte man die Möglichkeit haben, eine URL einem Knoten zuzuweisen, so dass man bei Doppelklick auf dem Knoten, die hinterlegte URL mit dem Standardbrowser aufrufen kann. Zum Anderen sollte es möglich sein, ein Bild in einem Knoten hinzuzufügen. Diese Funktion ist allerdings leider zum jetzigen Zeitpunkt noch nicht implementiert.

Wenn also ein Knoten selektiert ist und der Anwender den Punkt „URL einfügen“ anklickt, wird ein Eigenschaftsfenster aufgerufen, in dem der Nutzer die Möglichkeit dazu hat, eine URL zu dem ausgewählten Knoten hinzuzufügen.

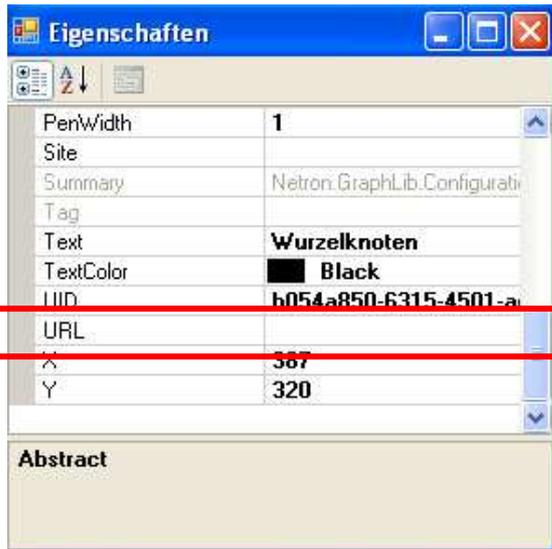


Abb.34: Datei – Menü MindMap

Die Möglichkeit, ausgewählte Eigenschaften eines Objekts, hier in dieser Arbeit eines Knotens, in einem eigenen Fenster anzeigen zu lassen, lässt sich durch das bereits vorgefertigte *PropertyGrid* von .NET realisieren. Hierzu wurde neben dem Hauptfenster MindMap eine neue *Form* - Klasse *Eigenschaften* angelegt. Mit Hilfe der Methode *Browsable(false)* bzw. *Browsable(true)* kann man angeben, welche Eigenschaften angezeigt werden sollen und der Nutzer somit auch ändern darf. Die Default- Einstellung liegt bei *Browsable(true)*, das heißt alle Eigenschaften werden angezeigt.

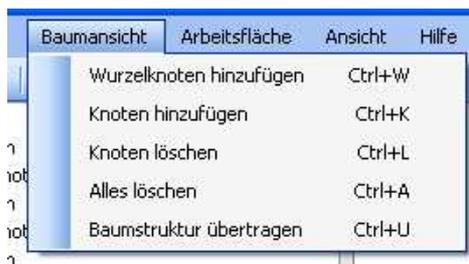


Abb.36: Baumansicht – Menü MindMap

In dem Menüpunkt Baumansicht können Knoten in der *TreeView* im linken Fenster hinzugefügt und gelöscht werden. Abschließend kann man die Baumstruktur auf das rechte Fenster in eine Mind Map übertragen.

Will man einen Wurzelknoten hinzufügen, geschieht dies durch den Befehl `this.treeView1.Nodes.Add("root", "Wurzelknoten")`. Falls bereits ein Knoten angelegt ist, kommt eine Fehlermeldung.

Genau der gleiche Befehl wird auch verwendet, wenn ein Kinderknoten hinzugefügt werden soll. Allerdings muss vorher geprüft werden, dass ein Knoten selektiert ist, zu dem der neue Kinderknoten auch hinzugefügt werden soll.

Beim Löschen eines Knotens muss dieser logischerweise selektiert sein und es werden alle Kinderknoten dieses Knotens selbstverständlich auch mit gelöscht. Realisiert wird dies durch den folgenden Befehl: `this.treeView1.SelectedNode.Nodes.Remove(this.treeView1.SelectedNode)`. Sollen alle Knoten gelöscht werden muss man einen anderen Befehl ausführen: `this.treeView1.Nodes.Clear()`.

Wie die Baumstruktur übertragen wird, wurde bereits vorher im Text zu dem Thema Layout erklärt.



Abb.37: Arbeitsfläche – Menü MindMap

Das Arbeitsfläche – Menü beinhaltet Methoden zum Hinzufügen, Bearbeiten und Löschen von Knoten. Außerdem kann das Netron Tree – Layout eingeschaltet werden. Die Funktionen zum Ausschneiden, Kopieren und Einfügen können momentan leider nicht benutzt werden, da die Implementierungen von Netron einen Bug hatten.

Beim Hinzufügen eines Knotens auf der Arbeitsfläche hat man wieder durch eine Abfrage die Möglichkeit, das Layout von Netron zu wählen.

Wenn noch keine Knoten auf der Arbeitsfläche vorhanden sind, wird zuerst ein Wurzelknoten in der Mitte angelegt, ansonsten wird ein Kinderknoten erzeugt. Falls im Vorfeld ein Knoten selektiert wurde, wird der neue Kinds-knoten automatisch damit verbunden, sonst wird er ohne Verbindung auf der Arbeitsfläche hinzugefügt. Dementsprechend wird kein Layoutalgorithmus angewendet.

```
private void knotenArbeitsflaecheToolStripMenuItem1_Click(object sender, EventArgs e)
{
    DialogResult res = MessageBox.Show("Wollen Sie den Layoutalgorithmus von Netron
        benutzen?", "Information", MessageBoxButtons.YesNo, MessageBoxIcon.Question);

    LauraForm wurzel = new LauraForm();

    if (this.graphControl1.Shapes.Count == 0)
    {
```

```

        this.graphControl1.AddShape(wurzel);
        wurzel.X = positionX;
        wurzel.Y = positionY;
        wurzel.Text = "Wurzelknoten";
        this.graphControl1.Refresh();
    }

    else
    {
        LauraForm kind = new LauraForm();
        this.graphControl1.AddShape(kind);
        positionX = positionX + 50;
        positionY = positionY + 50;

        try
        {
            this.graphControl1.AddConnection(this.graphControl1.SelectedShapes[0].Connectors[0], kind.Connectors[0]);
            kind.Text = "Kindsknoten";
            kind.X = positionX;
            kind.Y = positionY;
            this.graphControl1.Refresh();

            if (res == DialogResult.Yes)
            {
                this.graphControl1.StartLayout();
            }
        }

        catch (NullReferenceException nre)
        {
            kind.Text = "Kindsknoten";
            kind.X = positionX;
            kind.Y = positionY;
            this.graphControl1.Refresh();
        }
    }
}

```

Wenn man den Menüpunkt „Knoten bearbeiten“ auswählt, wird das schon einmal erwähnte Eigenschaftsfenster geöffnet, um hier bestimmte Attribute des Knotens ändern zu können, das sind unter anderem die Textfarbe, die Schriftfarbe und Hintergrundfarbe usw.

Will man alle Knoten auswählen, wird dies durch die Funktion *this.graphControl1.SelectAll(true)* realisiert, alle Knoten löschen kann man durch den Befehl *this.graphControl1.Delete()*.

Entscheidet der Anwender sich zu einem späteren Zeitpunkt dafür, dass Layout von Netron einzuschalten, kann man dies ganz leicht dadurch erreichen, in dem man alle Knoten selektiert und dann den Befehl *this.graphControl1.StartLayout()* ausführt.

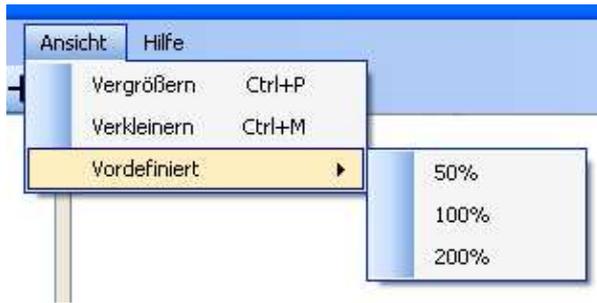


Abb.38: Ansicht – Menü MindMap

Das Ansicht- Menü bietet dem Anwender die Möglichkeit, entweder die Mind Map auf der Arbeitsfläche zu vergrößern, oder zu verkleinern. Außerdem kann man direkt einige vorgefertigte Größen als Zoomfaktor aufrufen.

Die Umsetzung ist in diesem Fall sehr leicht, da Netron eine *Zoom* - Funktion bereitstellt, die einfach auf das bestehende *graphControl* – Fenster angewendet werden muss.

Lediglich ein Bug musste behoben werden, und zwar kann die MindMap nur bis zu einer bestimmten Größe verkleinert werden und stürzt ansonsten ab. Der Fehler konnte aber ganz leicht abgefangen werden, in dem eine Variable *zoomZaehler* angelegt wurde, die anfangs mit null initialisiert ist. Im Anfangszustand, das heißt bei 100%, kann man die Mind Map lediglich einmal verkleinern und erniedrigt den *zoomZaehler* anschließend um eins. Ansonsten wird eine *if* – Abfrage, die darauf achtet, dass der *zoomZaehler* mindestens null ist, verhindern, dass die Funktion aufgerufen werden kann. Wird allerdings zwischenzeitlich die Vergrößern – Methode aufgerufen, wird der *zoomZaehler* um eins erhöht. Auf diese Weise, dass der *zoomZaehler* entweder erhöht oder erniedrigt wird, kann kein Programmabsturz zustande kommen.

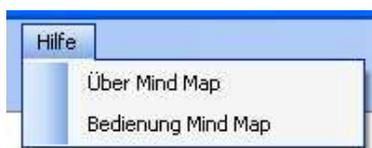


Abb.39: Hilfe – Menü MindMap

Das Hilfe- Menü öffnet einem entweder eine *MessageBox*, welche einem Informationen über das Programm MindMap gibt oder ruft ein PDF – Dokument auf, in dem einige Dinge in Bezug auf die Bedienung der beiden Fensterseiten (Baumansicht und Arbeitsfläche) erklärt werden.

6. Fazit und Ausblick

Das Programm MindMap stellt zum Zeitpunkt der Abgabe ein funktionierendes System zur Erstellung von Mind Maps dar. Allerdings muss man dazu sagen, dass noch sehr viele Funktionalitäten, die zum momentanen Zeitpunkt nicht implementiert bzw. nur ansatzweise implementiert sind, wünschenswert wären.

So sollte es zum Beispiel möglich sein, Bilder der MindMap hinzufügen zu können, sei es ganze Bilder oder auch nur kleine Graphiken und Icons zur Kennzeichnung bestimmter Sachverhalte. Diese Funktion war ja zum Beispiel in FreeMind verfügbar. Schlecht war daran allerdings, dass größere Bilder auch genau so in den Knoten der Mind Map eingefügt wurden und man diese nicht mehr skalieren konnte. Diesen Aspekt sollte eine spätere Implementierung auf jeden Fall berücksichtigen.

Des Weiteren wäre es nützlich und optisch ansprechender gewesen, mehrere verschiedene Knoten- und Linienformen zu entwickeln, um so eine gewisse Vielfalt zu gewährleisten. Dies ist also auch ein erstrebenswertes Ziel für die Zukunft.

Das schwierigste Problem im Laufe der Umsetzung war das Layout des Graphen. Es wurde versucht einen eigenen Algorithmus dafür zu entwickeln, der aber leider noch alles andere als eine optimale Lösung für die Anordnung der Mind Map darstellt. Ab einer bestimmten Anzahl an Knoten überlappen sich diese teilweise, das ist natürlich ein Aspekt, den es zu vermeiden gilt. Hier müsste also noch eine Implementierung erfolgen, die darauf achtet, dass sich die Knoten nicht berühren und automatisch den benötigten Platz freigibt. Außerdem funktioniert der Algorithmus nur bis zu einer gewissen Ebene, danach wirkt die Anordnung leider etwas chaotisch.

Die Layoutalgorithmen, die Netron hier zur Verfügung stellt sind allerdings auch nur suboptimal. Das Tree – Layout stellt eine ansatzweise annehmbare Lösung für das Problem dar, wünschenswert und viel passender wäre allerdings eine zirkulare Anordnung der Knoten um die Wurzel.

Die Netron Graph Library ist sehr umfangreich und es gibt sicherlich noch zahlreiche Funktionen, die man mit Hilfe dieser Bibliothek umsetzen könnte. In dieser Arbeit wurden die Funktionen bei Weitem nicht alle benutzt. Es gibt einige Papers zu Netron, die ansatzweise zeigen, wie mächtig diese Bibliothek ist.

Es war auf jeden Fall eine gute Wahl, sich für dieses Open Source – Graphenframework zu entscheiden. Die Nachteile von Netron liegen allerdings ganz klar in der mangelhaften Dokumentation. Aufgrund von Problemen mit dem ursprünglichen Entwickler von Netron wurden alle Dateien von der offiziellen Netron – Homepage entfernt und somit auch die recht spärlichen Dokumentationen. Lediglich Nutzer der Netron Graph Library versuchten ihre auf der eigenen Festplatte enthaltenen Dokumentationen wieder online zu stellen, so gingen aber leider einige vielleicht interessante Dokumentationen verloren.

Abschließend sollte man noch sagen, dass der Einsatz von Microsoft Visual Studio .NET 2005 in Verbindung mit der von Microsoft entwickelten objektorientierten Programmiersprache C# eine richtig gute Wahl gewesen ist. Diese Kombination hat gerade die Entwicklung eines Clients erheblich erleichtert und wäre in anderen Programmiersprachen mit Sicherheit nicht so einfach realisierbar gewesen.

7. Glossar

Bézierkurven	Kurve, dessen Kurvenverlauf sich leicht mittels Kontrollpunkten und sog. Bernsteinpolynomen auf mathematische Art und Weise berechnen lässt
Brainstorming	Kreativitätstechnik, die bei der Suche in einer Gruppe nach neuen Ideen zu einem gegebenem Thema hilft
FreeMind	Freeware – Programm zur Erstellung von Mind Maps
JPEG	Pixelgrafik – Format, sowohl verlustfreie als auch verlustbehaftete Kompression von Bildern
Kreativitätstechnik	eine Technik, um in einer Gruppe zu einem gegebenen Problem oder Aufgabenstellung eine Lösung zu finden
Mindjet	Firma, die das Programm MindManager entwickelt hat
MindManager	kostenpflichtige Programmlösung der Firma Mindjet zur Erstellung von Mind Maps
MindMap	der Name des von mir in dieser Studienarbeit entwickelten Programms zur Erstellung von Mind Maps
Mind Maps	Kreativitätstechnik, bei der um einen zentralen Begriff systematisch Ideen weiter verfeinert und strukturiert werden
MPX	Dateien, die unformatiertem Text beinhalten und für den Datenaustausch zwischen Microsoft Software und anderen Projektplanungsprogrammen genutzt werden
Netron Graph Library	Open Source – Bibliothek für Graphen
Open Source	frei zugänglicher Code im Internet, den andere benutzen und weiterentwickeln dürfen
PDF	plattformübergreifendes Format für Dokumente, welches sowohl Texte, als auch Bilder und Graphiken enthalten kann
Pixelgrafik	ist ein Bild in computerlesbaren Daten, besteht aus einer Tabelle von farbigen Pixeln, die eine bestimmte Größe und Breite haben (die Auflösung)

PNG	Pixelgrafik, verlustfreie Komprimierung
SVG	Vektorgraphik, XML – basiertes Dateiformat
SVN	ein Open – Source Tool, das der Versionskontrolle bei Softwareentwicklungen dient
Vektorgrafik	ist ein Bild in computerlesbaren Daten, das aus geometrischen Formen wie z.B. Linien, Kreisen und Polygonen gebildet ist
XHTML	Sprache zur Darstellung von Dokumenten, Bildern, Hyperlinks etc., Erweiterung von HTML in XML - Syntax
XML	standardisierte Sprache, legt Regeln für den Aufbau von Dokumenten fest

8. Quellenverzeichnis

Mind Maps allgemein:

[1]_A <http://www.e-teaching.org/glossar/mindmap>, abgerufen am 18.06.2006

[2]_A http://de.wikipedia.org/wiki/Mind_Map, abgerufen am 18.06.2006

FreeMind:

[1]_F <http://sourceforge.net/projects/freemind>, abgerufen am 11.06.2006

[2]_F http://freemind.sourceforge.net/wiki/index.php/Main_Page, abgerufen am 11.06.2006

[3]_F <http://www.e-teaching.org/technik/produkte/freemindsteckbrief>, abgerufen am 11.06.2006

[4]_F http://phlow.net/mag/netzkultur_spielerei/freemind_mindmapping_strukturiere_deinen_alltag_vernetze_deine_gedanken.php, abgerufen am 11.06.2006

[5]_F http://www.openbytes.de/Artikel/freemind_html, abgerufen am 11.06.2006

MindManager:

[1]_M <http://www.mindjet.com/de/>, abgerufen am 21.06.2006

[2]_M <http://www.e-teaching.org/technik/produkte/mindmanagersteckbrief>, abgerufen am 21.06.2006

[3]_M <http://www.macnews.de/news/76648>, abgerufen am 23.06.2006