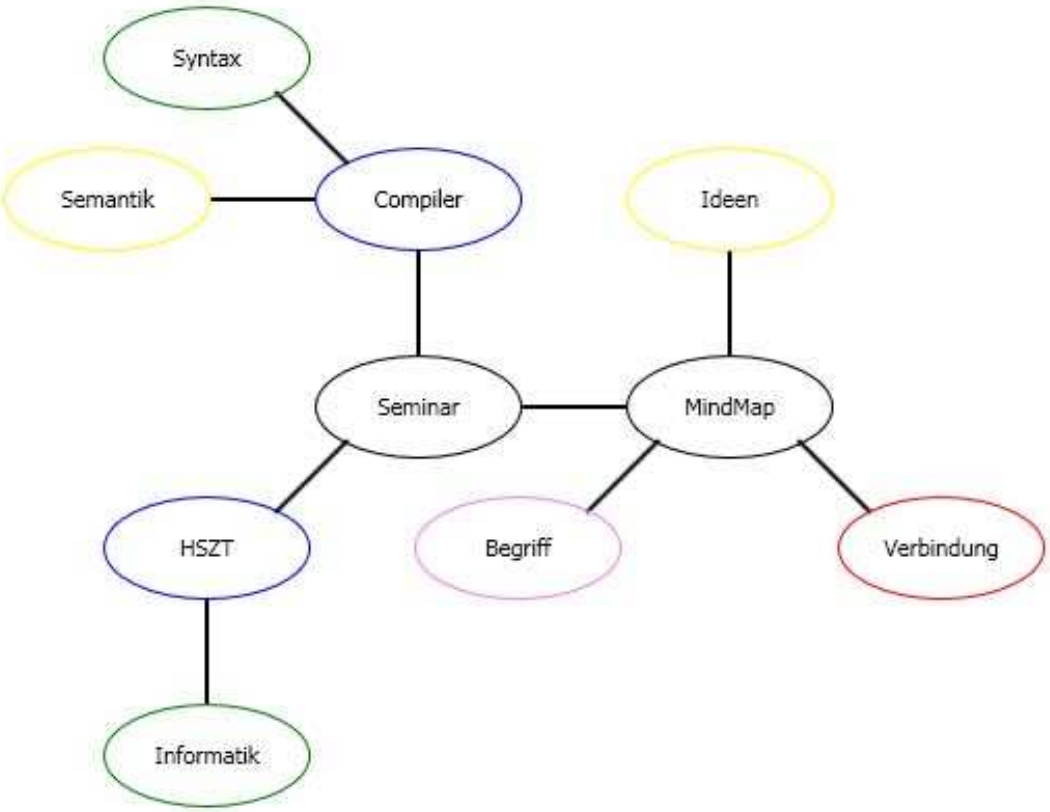
Seminar Syntax und Semantik



Reto Rezzonico

Dozent: Lukas Eppler

Hochschule für Technik Zürich

Frühjahrssemester 2009

Inhaltsverzeichnis

Abbildungsverzeichnis

1 Einleitung

1.1 Aufgabenstellung

Mit der Arbeit sollen die theoretischen Grundlagen zum Sprachaufbau / Parsing (Niklaus Wirth: Grundlagen und Techniken des Compilerbaus) angewendet werden. Kernstück der Arbeit ist die Entwicklung einer Software, mit der anhand einer eigenen Sprache ein grafisches Mind Map generiert wird.

1.2 Ziel der Arbeit

- Erstellen einer eigenen Sprach-/Syntaxdefinition mit Befehlen zur Erstellung eines Mind Maps.
- Entwickeln einer Software in C# .NET
- Parsen der definierten Sprache
- Erstellen einer geeigneten Datenstruktur für das Mind Map
- Grafisches Aufbereiten der Datenstruktur zur Anzeige des Mind Maps

1.3 Zusammenfassung

Bevor ich mit der eigentlichen Seminararbeit begann, las ich das Buch „Grundlagen und Technicken des Compilerbaus“ von Niklaus Wirth. Das Buch gab mir einige Inspirationen. Vor allem die Kapitel über Sprache, Syntax, Grammatik und Semantik beinhalten Aspekte, die ich in meiner Arbeit verwenden konnte. Die Definition meiner Sprache nahm ich anhand der im Buch beschriebenen Technik vor. Die Kapitel, die sich konkret mit dem Compilerbau beschäftigen, waren interessant zu lesen, brachten mir für die Seminararbeit jedoch wenig.

Als ersten Schritt meiner Arbeit definierte ich die Sprache. Da ich bereits eine ziemlich genaue Vorstellung hatte wie mein Programm aufgebaut werden sollte, konnte ich die Sprache anhand der Anforderungen des Programms entwickeln. Anfangs wollte ich eine C# ähnliche Syntax verwenden, als ich jedoch die Anwendungsfälle durch spielte entschied ich mich einen anderen Ansatz zu wählen. Nachdem ich die Sprache definiert hatte, erstellte ich eine Liste von Befehlen, die in einer ersten Version verfügbar sein sollten.

Da ich keine Zeit für das Erlernen einer neuen Sprache/Umgebung oder wegen nebensächlichen Problemen verschwenden wollte, verwendete ich für die Entwicklung des Mindmappers die mir bereits bekannte Spache C# .NET (Framework 3.5) und die Entwicklungsumgebung Visual Studio 2008. Damit konnte ich mich auf die Kernaufgabe der Arbeit konzentrieren. Als erstes konzipierte und programmierte ich die Datenstruktur für das Mind Map. Anschliessend erstellte ich den Parser mit dem die Befehle des Benutzers umgesetzt werden. Zu einer ersten Version fehlte nur noch die Anzeige des Mind Maps. Für das GUI verwendete ich die neue Technologie WPF (Windows Presentation Foundation), die mich vor allem beim Zeichnen des Mind Maps sehr komfortabel unterstütze. Obwohl ich WPF zuvor noch kaum verwendete, konnte ich bereits nach kurzer Zeit ein Mind Map anzeigen.

Mit der aktuellen Version des Mindmappers kann, mit den zur Verfügung stehenden Befehlen, ein einfaches Mind Map gezeichnet werden. Damit konnte ich die gesteckten Ziele der Seminararbeit erreichen.

2 Syntax

2.1 Definition der Sprache

Inspiriert durch das Buch Grundlagen und Techniken des Compilerbaus von Niklaus Wirth habe ich meine Sprache mittels der Backus Naur Form (BFN) Notatien definiert.

syntax	=	{production}
production	=	instruction attributelist
attributelist	=	attributename=attributevalue|attributevalue {attributename=attributevalue|attributevalue}
attributename	=	letter|digit {letter|digit}
attributevalue	=	letter|digit {letter|digit}
instruction	=	letter|digit {letter|digit}
letter	=	a | … | z
digit	=	0 | … | 9

Ein Faktor der Form {x} ist gleichbedeutend mit einer beliebig langen Folge von x, inklusive der leeren Folge.

Eine Codezeile beginnt immer mit einer Instruktion (Befehl) gefolgt von der dazugehörigen Attributliste. Instruktionen, Attributnamen und Attributwerte bestehen aus kleinen Buchstaben oder Zahlen. Das einzige verwendete Sonderzeichen ist das Gleichheitszeichen, es wird für die Attributübergabe per Name (Attribute by name) verwendet.

2.2 Befehlsübersicht

Um das Mind Map zu erstellen stehen die folgenden Befehle zur Verfügung:

Befehl	Attribute	Beschreibung
mind	name, caption, color, activation	Erstellt einen Begriff im Mind Map.
north	siehe mind	Erstellt oberhalb des aktiven Begriffes einen neuen Begriff.
south	siehe mind	Erstellt unterhalb des aktiven Begriffes einen neuen Begriff.
east	siehe mind	Erstellt rechts vom aktiven Begriff einen neuen Begriff.
west	siehe mind	Erstellt links vom aktiven Begriff einen neuen Begriff.
northeast	siehe mind	Erstellt rechts oberhalb des aktiven Begriffes einen neuen Begriff.
southeast	siehe mind	Erstellt rechts unterhalb des aktiven Begriffes einen neuen Begriff.
northwest	siehe mind	Erstellt links oberhalb des aktiven Begriffes einen neuen Begriff.
southwest	siehe mind	Erstellt links unterhalb des aktiven Begriffes einen neuen Begriff.
forget	name	Löscht einen Begriff und alle seine Verbindungen.
center	name	Aktiviert einen Begriff.

Attribute:

- name: Variablenname des Begriffs
- caption: Bezeichner des Begriffs (Anzeige)
- color: Farbe des Randes
- activation: Definiert ob der neue Begriff gleich zum neuen Ausgangspunkt wird

2.3 Ablauf Parsing

Im folgenden Ablaufdiagram wird der Ablauf des Parsings einer Befehlszeile beschrieben. Die einzelnen Schritte werden in den folgenden Unterkapiteln erläutert.

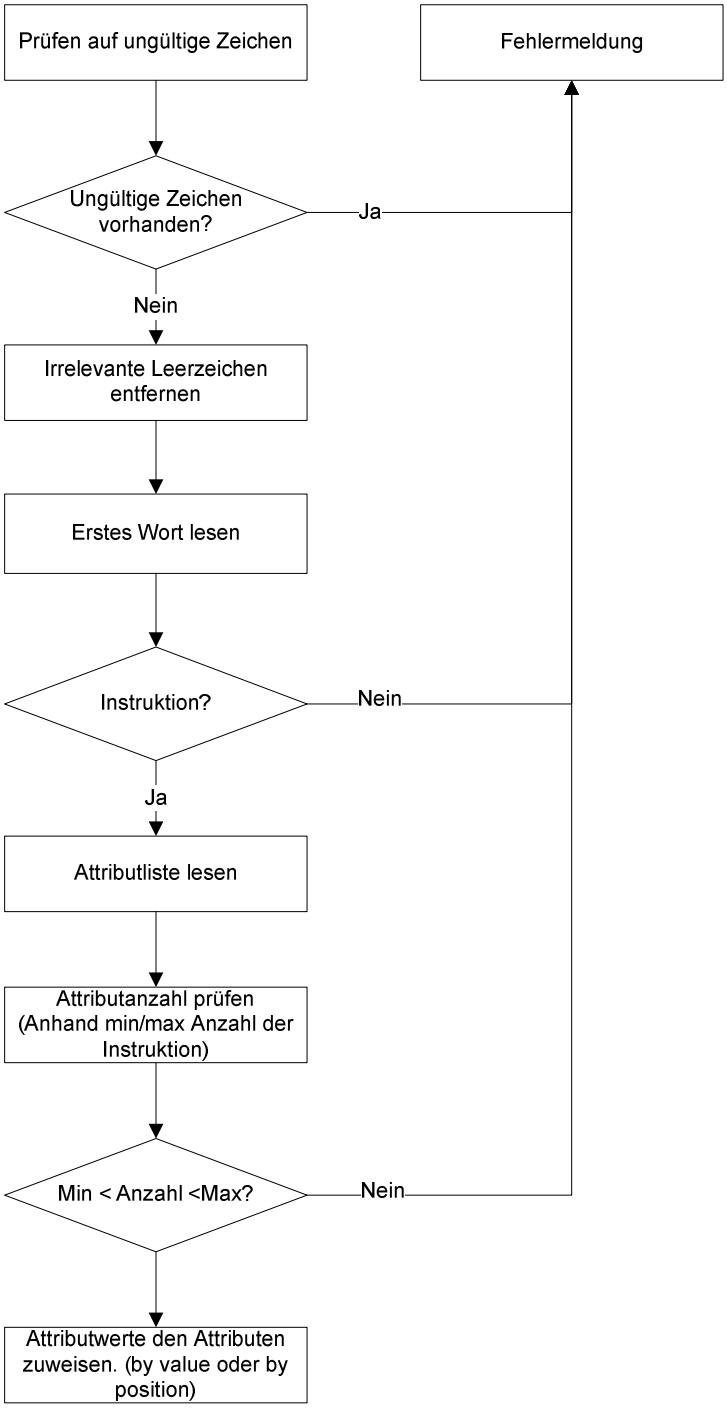


Figure 1 - Ablaufdiagram Parsing

2.3.1 Prüfen auf ungültige Zeichen

Eine Codezeile darf nur Buchstaben, Zahlen, Leerzeichen sowie Gleichheitszeichen enthalten.

2.3.2 Irrelevante Leerzeichen entfernen

Leerzeichen werden verwendet um die einzelnen Instruktionen und Attribute zu trennen. Leerzeichen die offensichtlich keine Bedeutung haben (keine Instruktionen oder Attribute trennen) werden entfernt.

Beispiele:
{space}{space} wird zu {space}
{space}={space} wird zu =

2.3.3 Erstes Wort/Instruktion lesen

Das erste Wort der Codezeile muss immer eine Instruktion sein. Es wird geprüft, ob das erste Wort eine Instruktion ist.

2.3.4 Attributliste lesen

Die Attributliste der aktuellen Instruktion wird ermittelt.

2.3.5 Prüfung Attributanzahl

Anhand der minimal und maximal erwarteten Attributanzahl der Instruktion wird validiert, ob eine gültige Anzahl Attribute übergeben wurde.

2.3.6 Zuweisung Attributwerte

Die übergebenen Attributwerte werden den Attributen der Instruktion zugewiesen. Die Werte können über die Position (by position) oder über den Attributnamen (by name) übergeben werden. Bei der

Beispiele:
By position: mind m1 schweiz red
By name: mind name=m1 color=red caption=schweiz

3 Architektur

3.1 Kernarchitektur

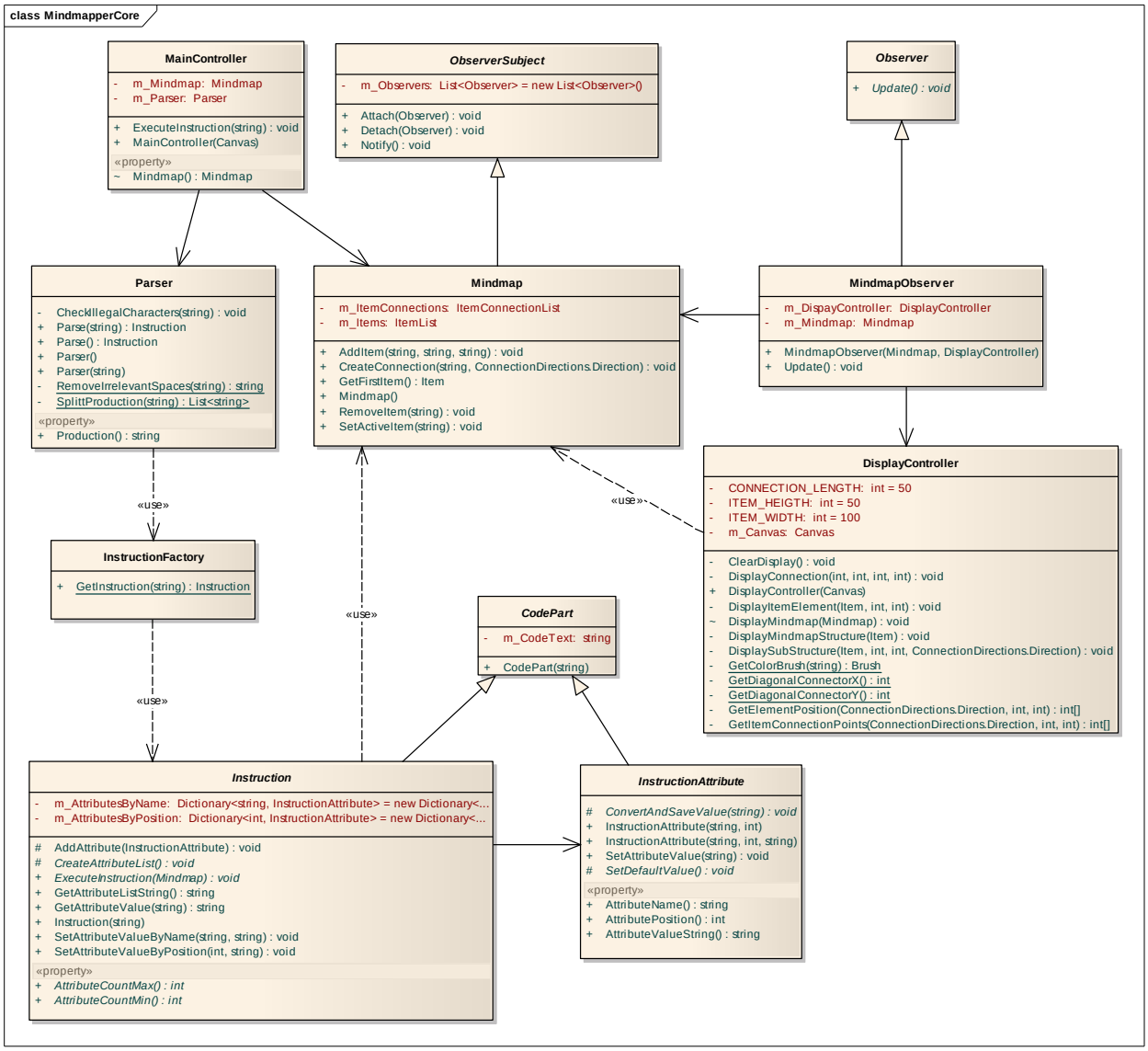


Figure 2 - Klassendiagramm Kernarchitektur

3.1.1 InstructionFactory

Die InstructionFactory liefert anhand des Befehls (Instruction der BFN) eine Instanz der entsprechenden Instruction Klasse.

3.1.2 Mindmap Observer

Änderungen am Mindmap Objekt werden mit dem Observer Pattern abgehandelt. Mindmap ist von ObserverSubject abgeleitet und benachrichtigt mit der Notify Methode alle angemeldeten Observer (in meinem Fall der MindmapObserver) über die Änderung.

3.1.3 Ablauf ExecuteInstruction

Mit dem folgenden Sequenzdiagramm wird grob der Ablauf der Befehlsausführung beschrieben. Anhand dieses Ablaufs kann die Funktionalität der Kernarchitektur am besten nachvollzogen werden.

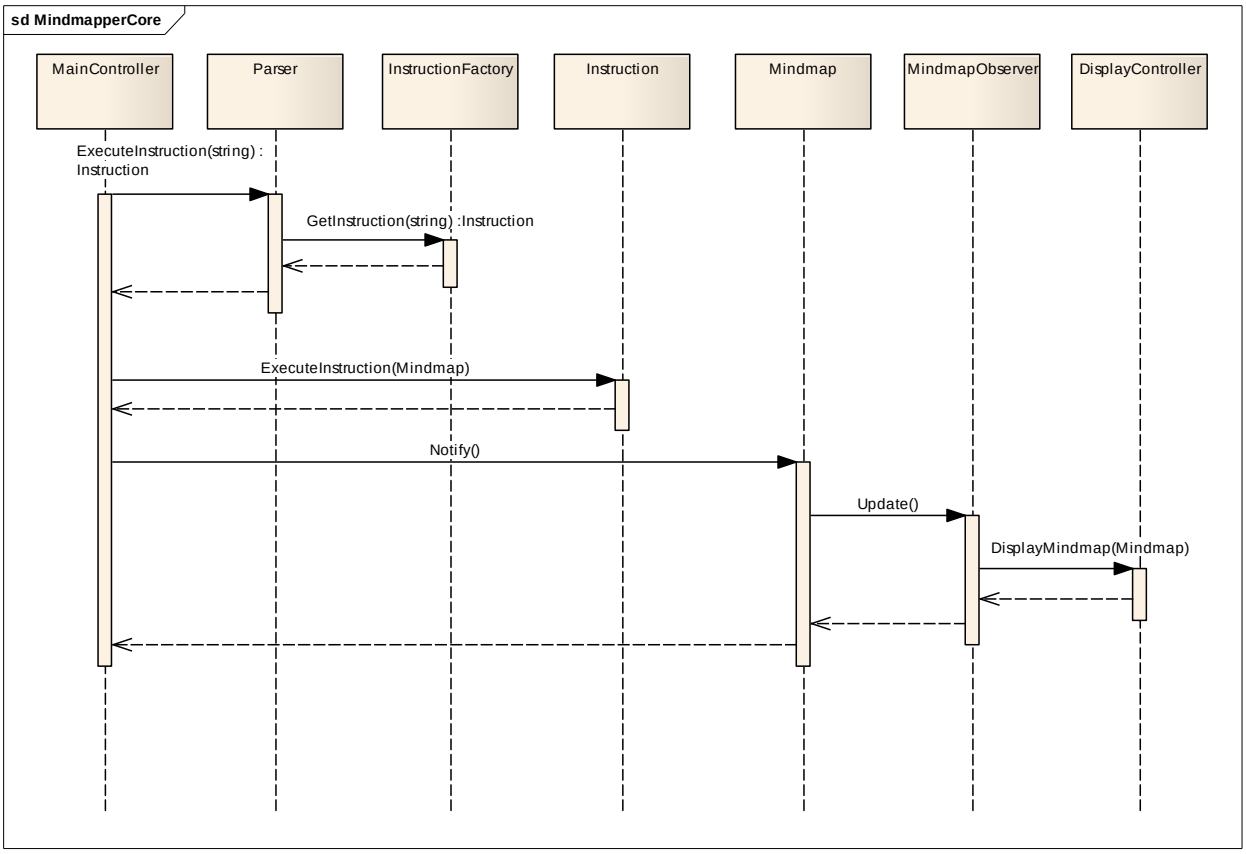


Figure 3 - Ablauf ExecuteInstruction

3.2 Architektur Befehle

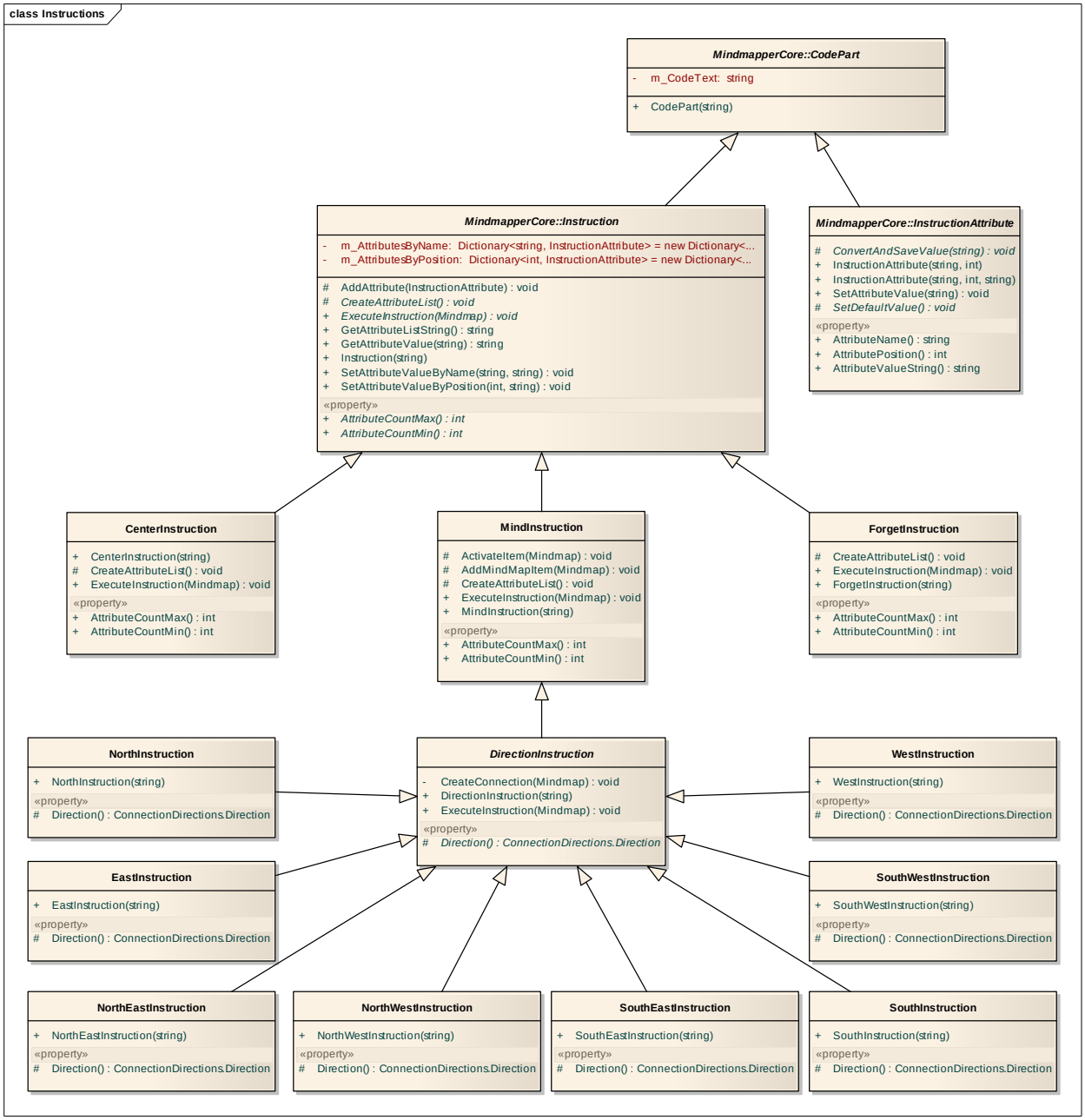


Figure 4 - Klassendiagramm Befehle

Alle spezifischen Instructionklassen sind von der Basisklasse Instruction abgeleitet, diese wiederum ist von CodePart abgeleitet. In der Basisklasse Instruction ist die Interaktion mit den InstructionAttributes implementiert. In den spezifischen Implementationen müssen lediglich die verfügbaren Attribute definiert werden. Weiter wird in den spezifischen Klassen natürlich die Ausführungslogik des entsprechenden Befehls implementiert.

3.3 Datenstruktur Mind Map

Das Mind Map wird als Items und ItemConnections gespeichert. Für jeden Begriff im Mind Map wird ein Item erstellt und für jede Verbindung eine ItemConnection. Da für die verschiedenen Operationen (erstellen, löschen, anzeigen, etc.) unterschiedlich durch die Datenstruktur navigiert werden muss, werden alle Items und Connections in einer entsprechenden Liste abgelegt. Jede Connection hat eine Direkte Referenz auf die betroffenen Items.

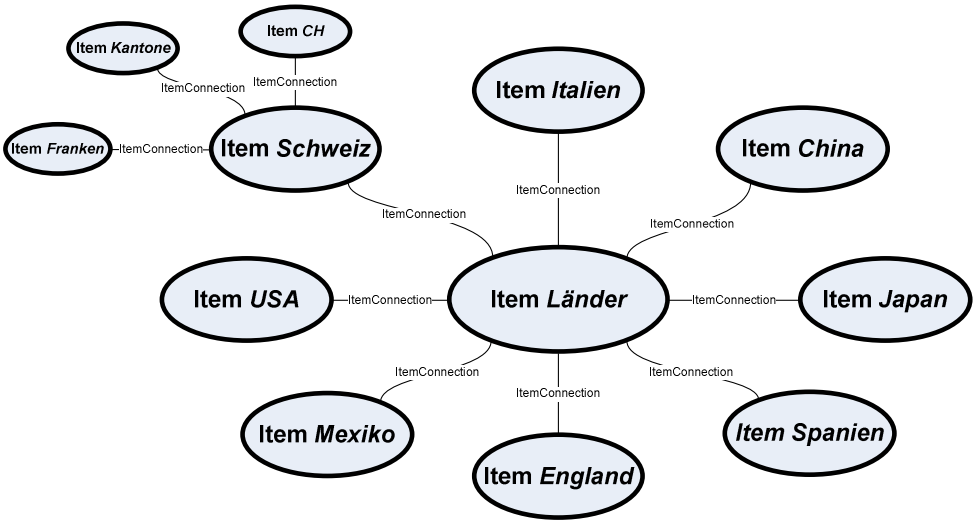


Figure 5 - Beispiel Mind Map für Datenstruktur

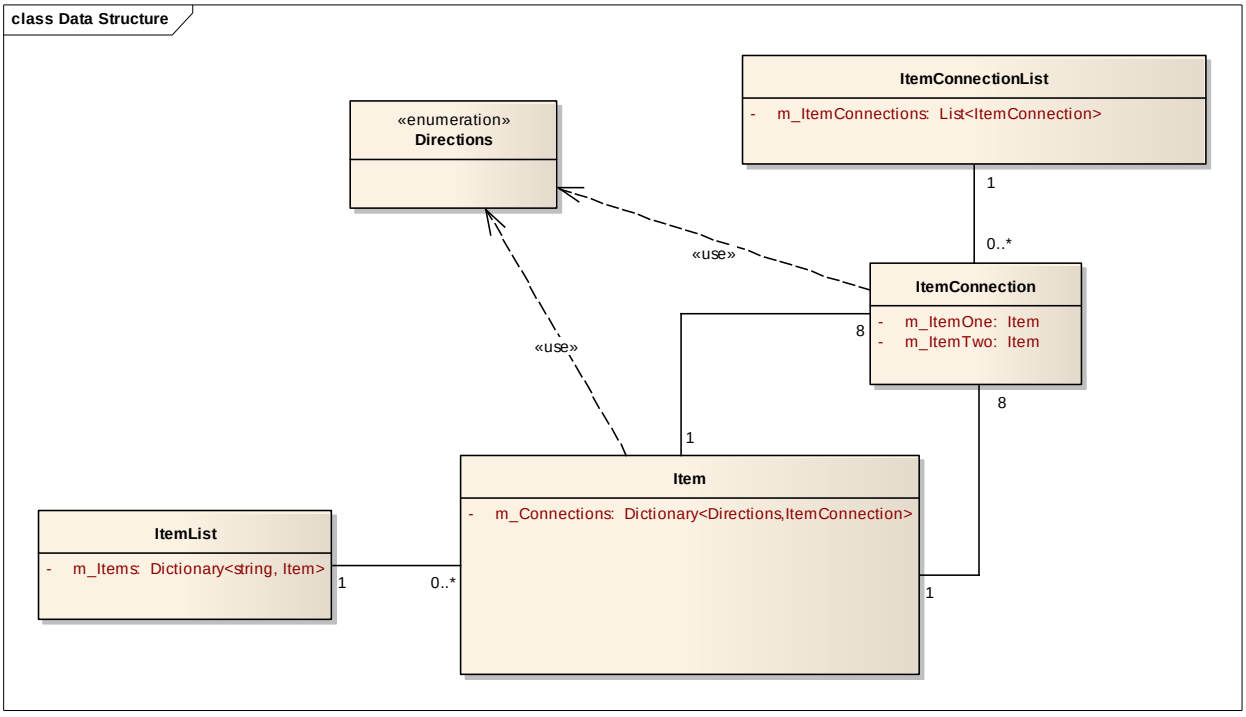


Figure 6 - Datenstruktur Mind Map

4 Bedienungsanleitung

Das folgende Kapitel beschreibt die Bedienung der Anwendung. Da die Anwendung noch nicht vollständig fertig entwickelt ist, sollte sich der Benutzer bewusst sein, dass gewisse Fehler auftreten können. Die folgende Version implementiert vor allem die notwenige Logik, auf Benutzerführung und allgemeine Fehlerbehandlung wurde noch wenig Wert gelegt.

4.1 Mindmapper Oberfläche

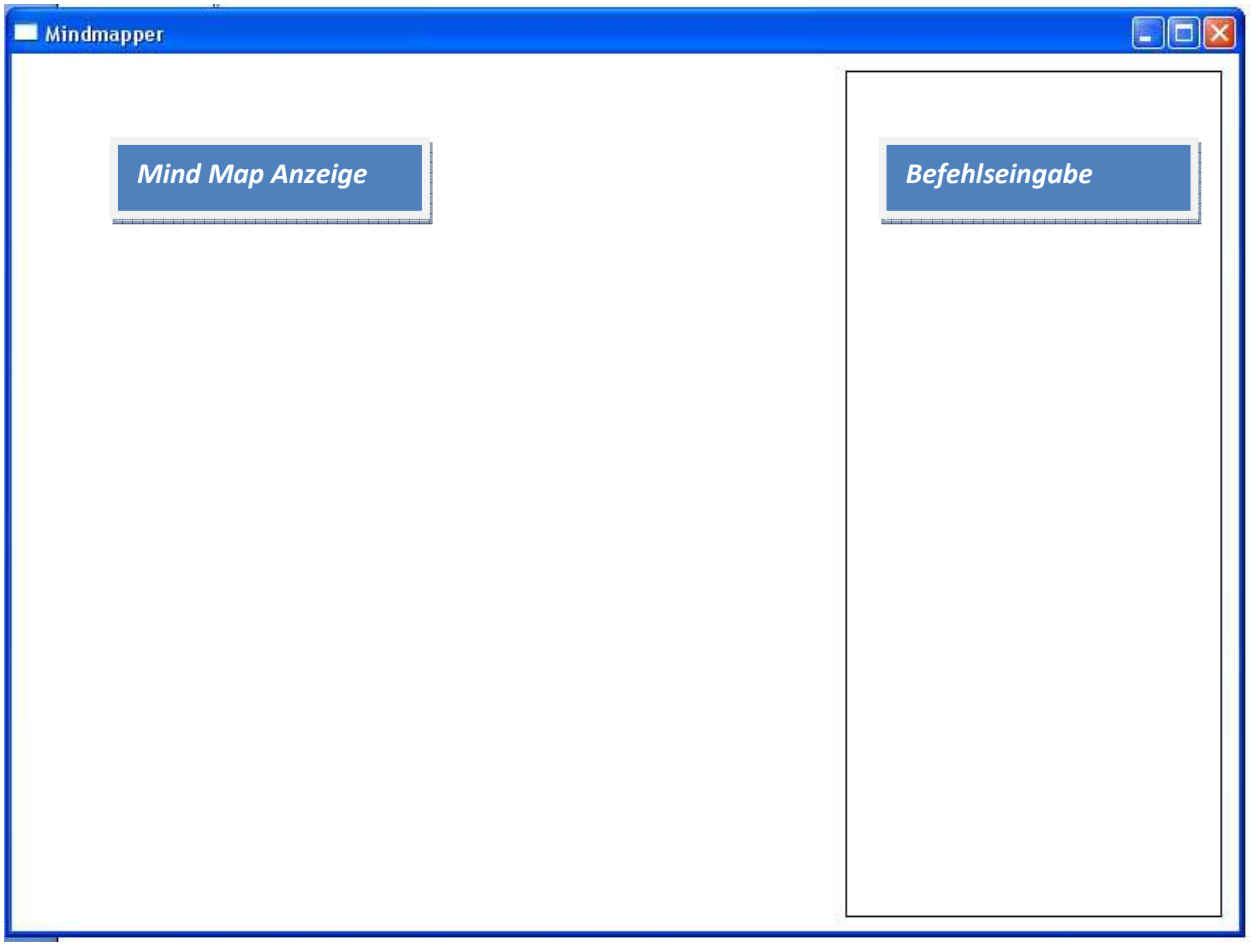


Figure 7 - Oberfläche Mindmapper

Die Oberfläche ist sehr einfach gehalten. Sie besteht aus einem Anzeigebereich rechts und dem Eingabefeld für Befehle links.

4.2 Befehle

4.2.1 Mind Befehl

4.2.1.1 Beschreibung

Der Mind Befehl wird als erster Befehl verwendet um ein Mind Map zu beginnen und ein Begriff zu zeichnen.

4.2.1.2 Parameter

name

Beschreibung: Der Variablenname des Begriffs.

caption

Beschreibung: Der Anzeigename des Begriffs.

color

Beschreibung: Die Randfarbe des Begriffs.
Mögliche Werte: black, red, blue, green, yellow, violet
Standardwert: black

activation

Definiert ob der neue Begriff als neuer Ausgangspunkt gesetzt wird.
Mögliche Werte: true, false
Standardwert: true

4.2.1.3 Beispiele

mind l1 schweiz red true
mind name=l1 caption=schweiz color=red

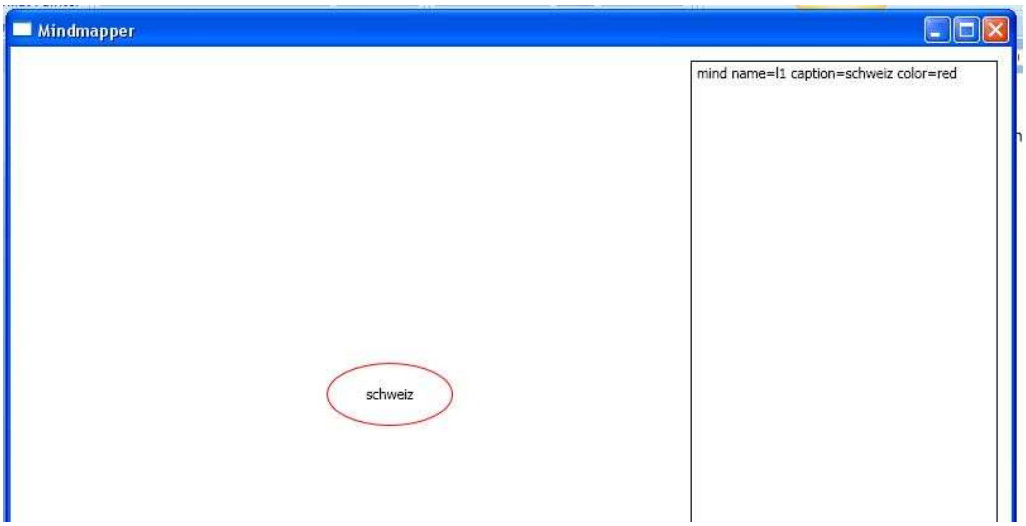


Figure 8 - Mind Befehl

4.2.2 Richtungbefehle

4.2.2.1 Beschreibung

Die Richtungsbefehle (north ,south, east, west, northeast, southeast, northwest, southwest) erstellen einen neuen Begriff in der entsprechenden Richtung, ausgehend vom aktuell aktiven Begriff.

4.2.2.2 Parameter

name

Beschreibung: Der Variablenname des Begriffs.

caption

Beschreibung: Der Anzeigename des Begriffs.

color

Beschreibung: Die Randfarbe des Begriffs.
Mögliche Werte: black, red, blue, green, yellow, violet
Standardwert: black

activation

Definiert ob der neue Begriff als neuer Ausgangspunkt gesetzt wird.
Mögliche Werte: true, false
Standardwert: true

4.2.2.3 Beispiele

north l2 deutschland yellow false
south name=l3 caption=italien color=green
northwest l4 frankreich blue

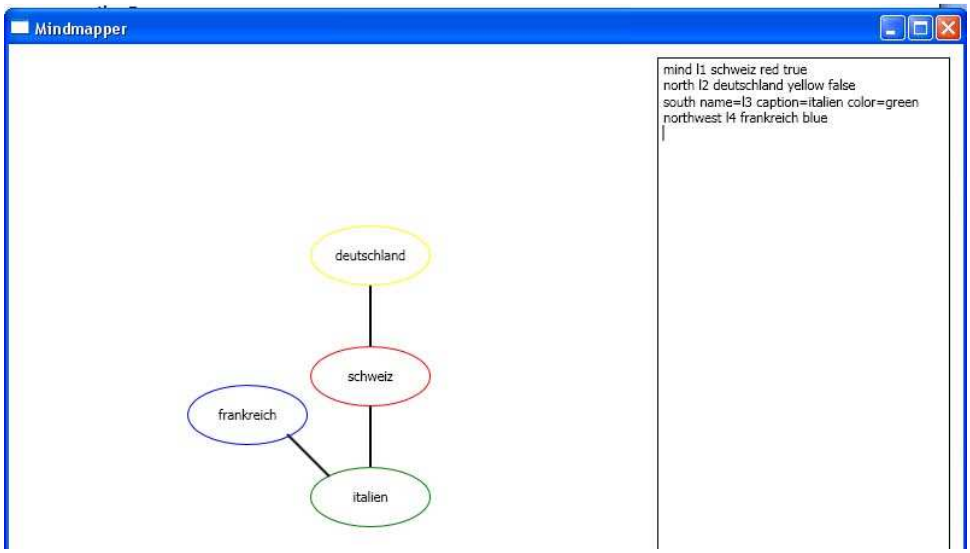


Figure 9 – Richtungsbefehle

4.2.3 Forget Befehl

4.2.3.1 Beschreibung

Mit dem Foget Befehl können bestehende Begriffe vom Mind Map entfernt werden.

4.2.3.2 Parameter

name

Beschreibung: Der Variablenname des Begriffs.

4.2.3.3 Beispiele

forget l2

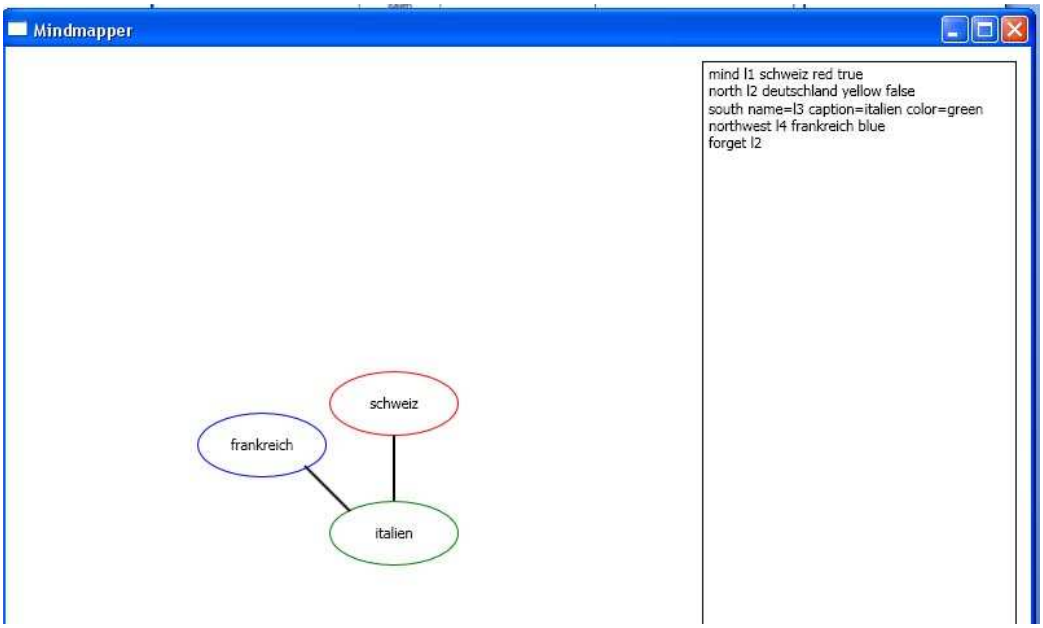


Figure 10 - Forget Befehl

4.2.4 Center Befehl

4.2.4.1 Beschreibung

Mit dem Center Befehl kann ein bestehender Begriff als Ausgangspunkt für neue Begriffe gewählt werden.

4.2.4.2 Parameter

name

Beschreibung: Der Variablenname des Begriffs.

4.2.4.3 Beispiele

center l1

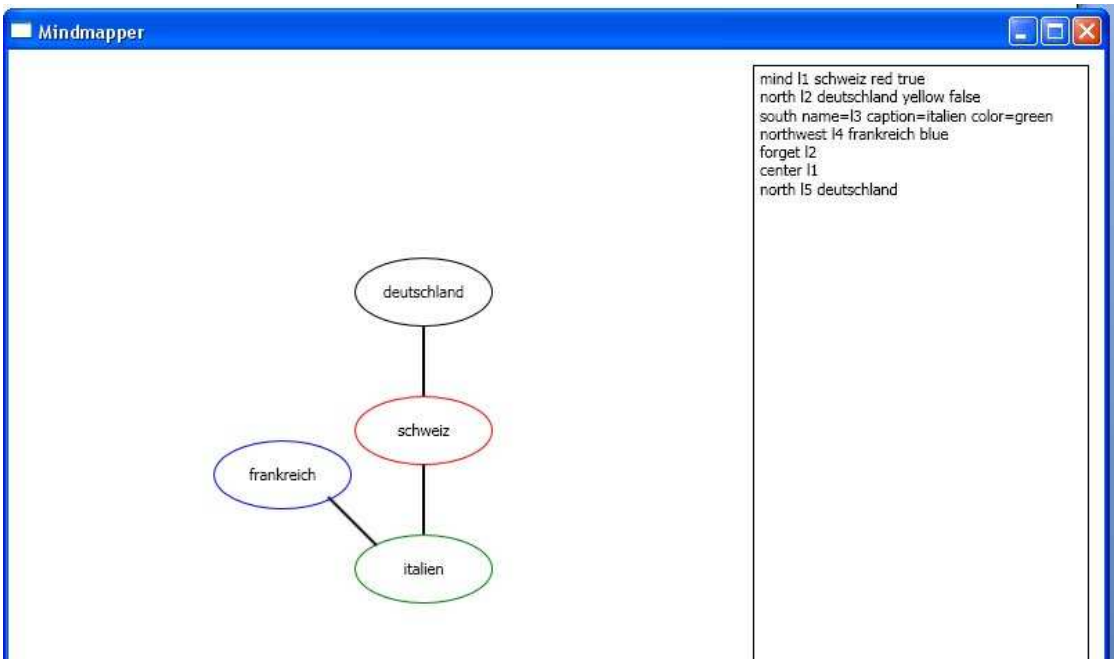


Figure 11 - Center Befehl

Bevor der Befehl center l1 ausgeführt wurde, war l4 (frankreich) der aktive Begriff. Nach der Ausführung ist l1 (schweiz) der aktive Begriff, deshalb wird l5 (deutschland) nördlich von l1 (schweiz) platziert.

5 Benutze Hilfsmittel/Tools

- Microsoft Visual Studio 2008
- Enterprise Architect
- Github

6 Literatur

- Niklaus Wirth: Grundlagen und Techniken des Compilerbaus, ISBN 978-3-486-58581-0
- Gamma, Helm, Johnson, Vlissides: Design Patterns, ISBN 0-201-63361-2