

Inhalt

- Motivation und Zielstellung
- Anforderungen und Stakeholder
- Theoretische Grundlagen
- Softwarearchitektur und -design
- Technologieentscheidungen
- Demonstration der Anwendung
- Reflexion und Ausblick

Motivation und Zielstellung

Motivation

Ziel

Formale sprachen sind schwer zu greifen

SW Tool zur Unterstützung in der Lehre

Anforderungen und Stakeholder

Stakeholder

Primäre

- Mitarbeitende der DHBW
- Lehrende an der DHBW
- Entwicklungsteam

Sekundäre

- Partnerunternehmen
- DHBW
- Zukünftige Entwicklungsteams
- Studierende

Anforderungen

Grundlegende Funktionalität

- Grammatik anlegen und verwalten
- Worte mit CYK testen
- Visualisierung
 - CYK
 - Ableitungsbaum

Anforderungen

Umgebung

- Geringe Hardwareanforderungen
- Unterstützung gängiger Betriebssysteme

Zuverlässigkeit

- Keine Beeinträchtigung des Vorlesungsbetriebs
- Gute User Experience, intuitive Bedienung

Anforderungen

Zukünftiges Potential

- Wiederverwendbarkeit
- Erweiterbarkeit
- Wartbarkeit
- Dokumentation

Theoretische Grundlagen

Begriffe

(Formale) Sprache

Menge von Wörtern über einem Alphabet

Formale Grammatik

Menge von Regeln zur Erzeugung formaler Sprachen

Begriffe

Terminale (T)

Folge von Terminalen bildet ein Wort

Nichtterminal (N)

Symbole aus denen (Nicht)Terminal(e) abgeleitet werden können
Startsymbol der Grammatik ist ein Nichtterminal

Kontextfreie Grammatiken

- Hier: Beschränkung auf kontextfreie (Chomsky-2) Grammatiken
- Regeln der Form $X \rightarrow \alpha$
 - $X \in N$
 - $\alpha \in (N \cup T)^*$

CYK

Chomsky Normalform (CNF)

- Regeln der Form
 - $S \rightarrow \varepsilon$
 - $X \rightarrow YZ$
 - $X \rightarrow a$
- $S, X, Y, Z \in N$; $a \in T$
 ε : Leeres Wort; S : Startsymbol
- Chomsky-2 Grammatiken können in die CNF transformiert werden

CYK

Grundsätzliche Arbeitsweise

- Betrachtung der möglichen Kombinationen von Teilworten
- Ergebnis: CYK Dreiecks-Matrix
- Wortproblem: Entscheidung durch Gegenwart des Startsymbols in letzter Zelle
- Ableitungsbäume: Rückverfolgung der Produktionen vom Startsymbol ausgehend

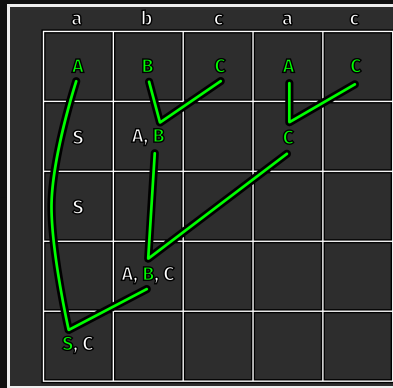
CYK

	a	b	c	a	c

CYK

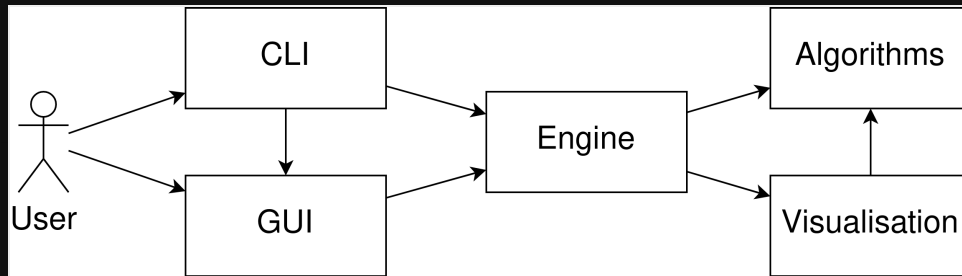
	a	b	c	a	c
	A	B	C	A	C
	S	A, B		C	
	S				
		A, B, C			
	S, C				

CYK

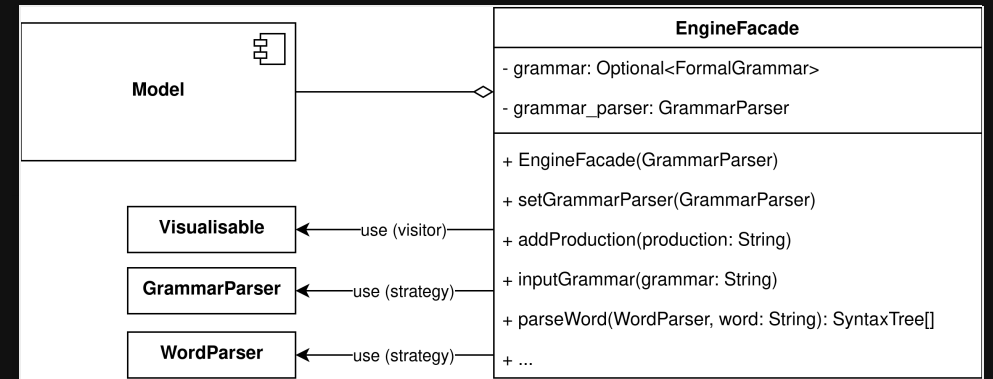


Softwarearchitektur und -design

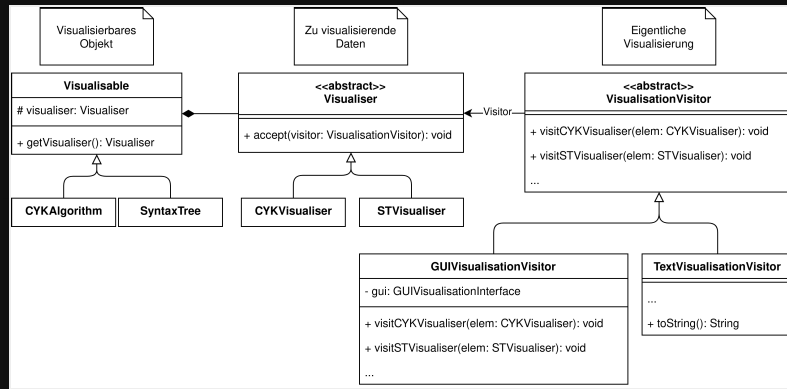
Überblick



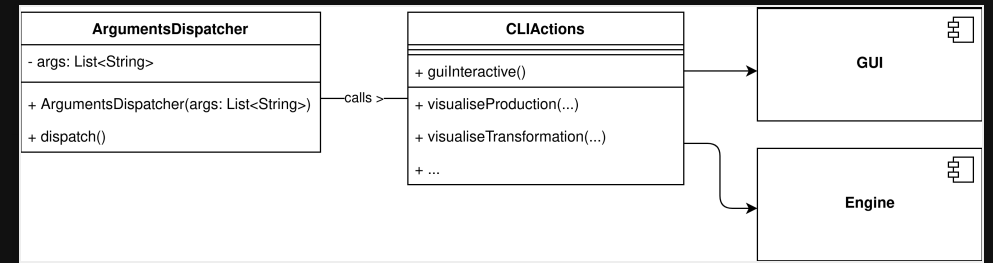
Engine



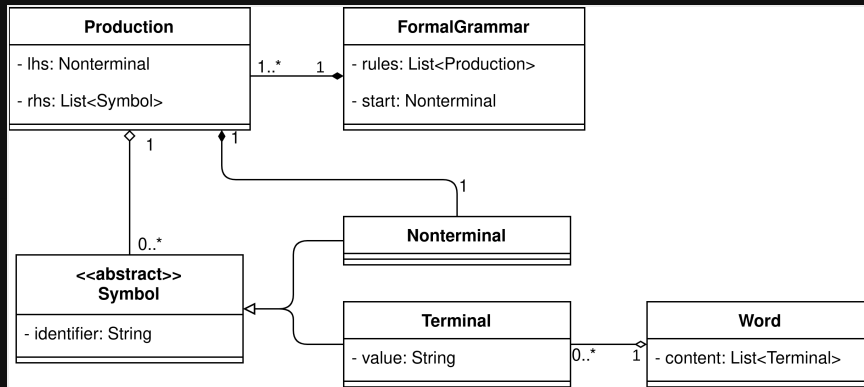
Visualisierung



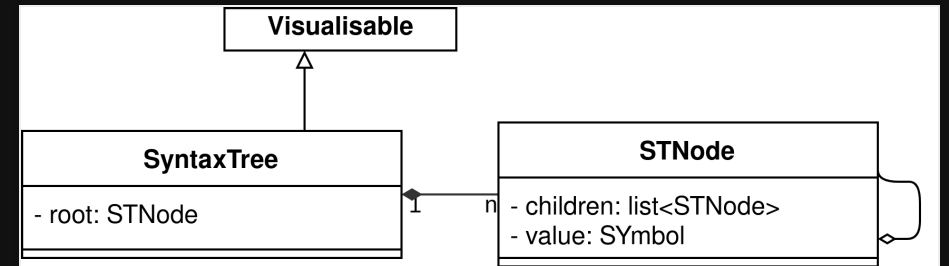
CLI



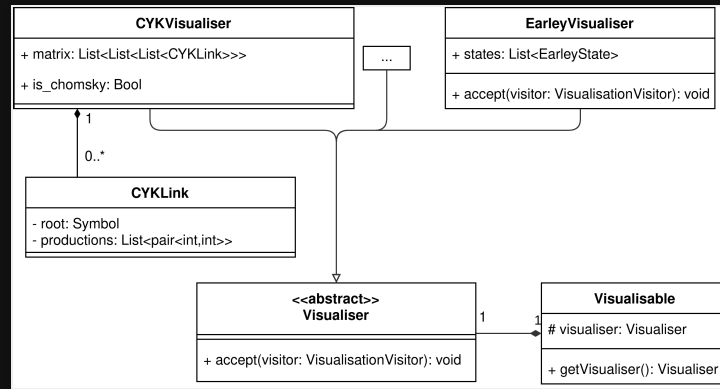
Datenmodell



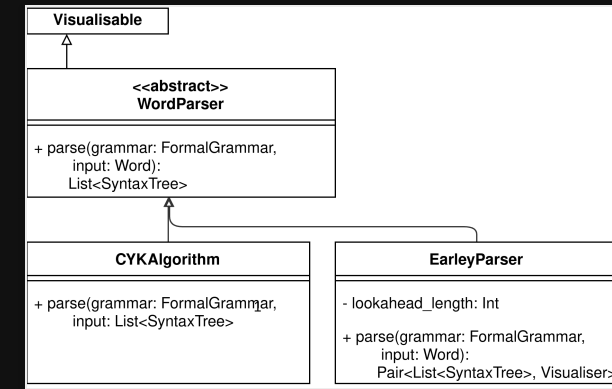
Datenmodell



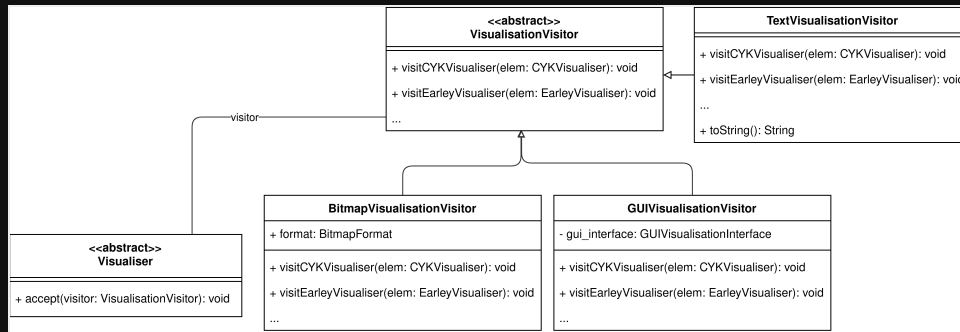
Algorithmik



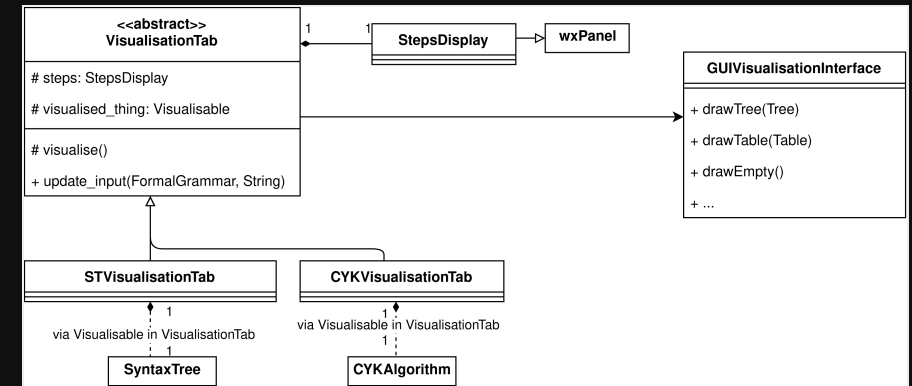
Algorithmik



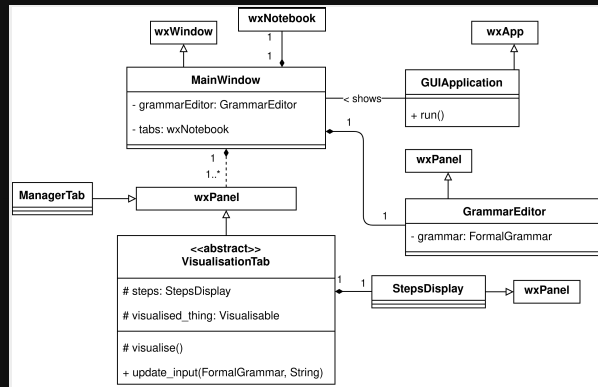
GUI



GUI



GUI



Technologieentscheidungen

Programmiersprache

C++

- Entwicklererfahrung vorhanden
- Verbreitet
- Prinzipiell hochperformant
- Unterstützung von OO-Programmierung
- Portabler Quellcode möglich

Buildsystem

CMake

- Plattformunabhängig
- Weitverbreitet bei OSS
 - Einbinden von Abhängigkeiten
 - IDE Integration

GUI

wxWidgets

- Plattformunabhängig
- Entwicklererfahrung vorhanden
- Verfügbarkeit von RAD Tools
- Verwendung nativer GUI-Elemente

Demonstration der Anwendung

Reflexion und Ausblick

Reflexion

Performance CYK

- Keine optimale Implementierung
- Overhead für Visualisierung
 - ~30% für Generierung der Syntaxbäume
 - ~60% für Handling der CYKLinks
 - ~20% für Speichern der einzelnen CYK Schritte

Reflexion

Technologie

- Einschränkungen der Implementierung
 - Dateien
 - Länge von Regeln und Wörtern
- Wiederverwendbarkeit
- Auf Windows momentan kleine Darstellungsprobleme

Ausblick

Funktionalität

- Unterstützte Algorithmen
- CLI
 - Textuelle Grammatikeingabe
 - Automatisiert verarbeitbare Ausgabe

Ausblick

Bedienung (GUI)

- Verfeinern des Bedienkonzepts
- Visuelle Verbesserungen
- Konfiguration

Vielen Dank für Ihre Aufmerksamkeit!

Gibt es noch Fragen?