

Diagen

Vortrag im Seminar *Visuelle Sprachen*

Wolfgang Machert

Universität Bremen - 01.07.2004

Betreuer: Dr. Berthold Hoffmann

Diagram Generator...

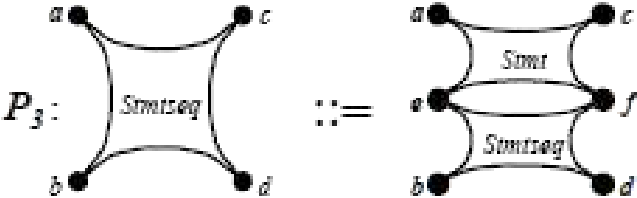
- ... enthält einen Generator der aus einer formalen Sprachspezifikation einen Diagrammeditor erzeugt ...
- ... enthält ein Editor Framework das die Erstellung und Bearbeitung von syntaktisch korrekten Diagrammen ermöglicht ...
- ... ist ein Java-Framework das allgemeine Funktionalität zur Erzeugung, Bearbeitung und Analyse von Diagrammen und Diagrammeditoren bereit stellt.

- Begriffsdefinitionen
- Entstehung
- Architektur
- Einordnung
- Quellen

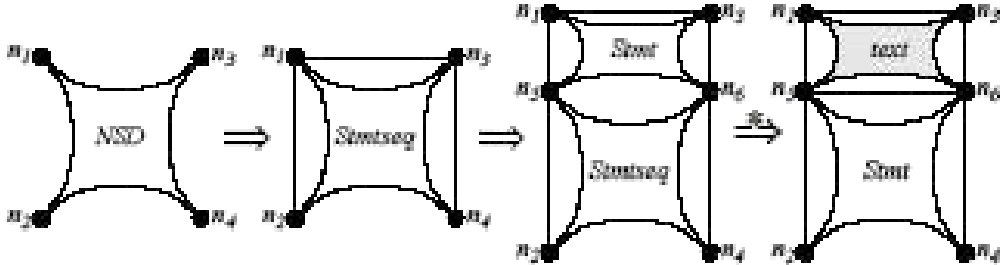
Begriffsdefinitionen

Begriffsdefinitionen

- Hypergraphen
 - Gerichteter Graph: jede Kante besucht (verbindet) 2 Knoten
 - Hypergraph: Hyperkanten besuchen beliebig viele Knoten
 - Hyperkanten besitzen Typ sowie Anzahl Verbindungspunkte (Arität)
- Kontextfreie Hypergraphgrammatiken
 - analog zu Kontextfreien Chomsky-Grammatiken
 - können direkt mehrdimensionale Beziehungen repräsentieren
 - einzelne Hyperkanten mit verbundenen Knoten werden überführt

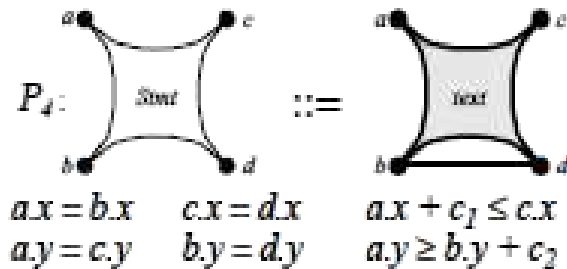


Hypergraph Grammar
(Quelle: [Minas 93])



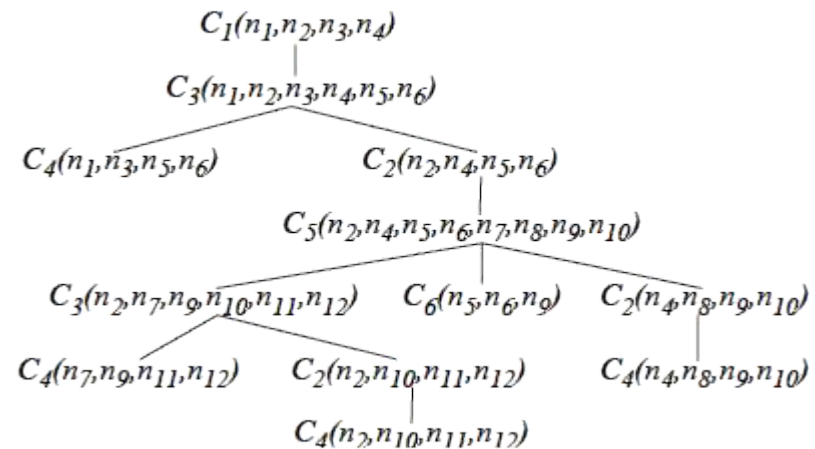
Derivation Tree
(Quelle: [Minas 93])

- Constraint Hypergraphgrammatik
 - Hypergraph Grammatik besitzt Attribute
 - Verwendung der Attribute für Constraints
 - Ableitungsbaum einer Hypergraphgrammatik muss Constraintbaum berücksichtigen



Constraint Hypergraph Grammar

(Quelle: [Minas 93])



Constraint Tree

(Quelle: [Minas 93])

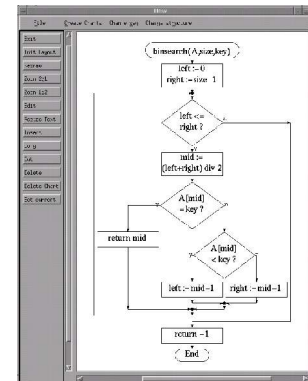
Entstehung von Diagen



- 1993: inkrementeller Algorithmus zur Constraintbewertung
 - Minimalisierung von Änderungen
 - Grundlage der Generierung und Validierung von Constraint Hypergraphgrammatiken
- 1994: Implementation von DEG
- 1995: DiaGen (C++ Implementation)
 - Generator für Diagramm Editoren aus Kontextfreien Hypergraphgrammatik
 - Syntaktische Manipulation der Diagramme möglich

Aber:

- inkonsistente Diagramm Manipulationen werden verweigert
- Manipulationen von spezifizierten Transformationen abhängig



Flowchart Editor
(Quelle: [Minas 95])

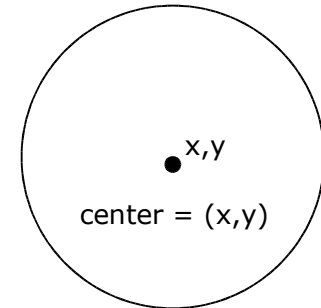
- 1997: DiaGen 2 (Java Implementation)
 - Parser zur Erkennung korrekter und inkorrektter Subgraphen
 - Freihand Bearbeitungsmodus erlaubt inkonsistente Graphen
 - nur noch Grammatik und Diagramm Blöcke notwendigAber:
 - keine Bearbeitung konsistenter Graphen im Editor
- 1999: DiaGen SRHG
 - Unterstützung räumlicher Relationen bei Verbindung von Knoten durch Editor (Spatial Relations)
 - Unterstützung der Java2D API: Zoomen, Bezierkurven ...
- 2000: DiaGen Designer
 - Visueller Designer als GUI zum erstellen der Spezifikation
 - Visuelle Erstellung der Regeln im Designer
 - vollständige Spezifikation des Editors in XML

Architektur

□ Komponenten (Hyperkanten)

Bitte Koordinaten des Mittelpunktes eingeben (1,1):

- Komponententyp -> Typ der Hyperkante
- Anzahl an Attachment Areas -> Arität
- Parameter
 - Beispiel: $x = 10$; $y = 10$; $center1 = (x,y)$
- Visuelle Repräsentation
 - Framework / eigene Implementation
- Initialisierungsparameter
 - Vorgabe von Standardwerten
 - Abfrage beliebiger Parameter
- Attachment Areas
 - Verknüpfungspunkte mit anderen Komponenten
 - Ausgangspunkt für Räumliche Relationen
- Handle
 - Interaktionsmöglichkeit im Editor
 - Standardhandles im Framework enthalten

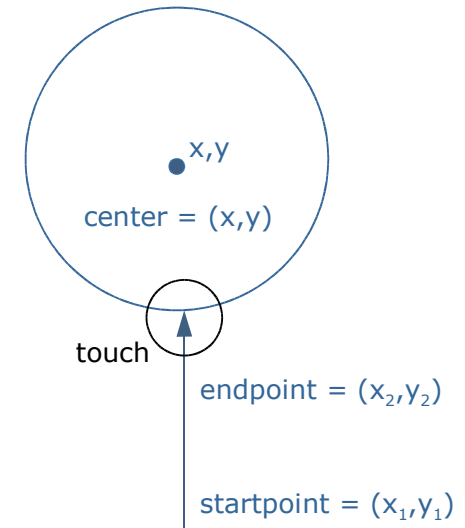


□ Relationen (Knoten)

- beschrieben durch räumliche Relation
 - Beispiel:
 - touch: distance (center, endpoint) = radius
 - touch: connection (attachment1, attachment2)
- derzeit nur Binäre Relationen
- Relationsprüfung nur bei aufeinander liegenden Berührungspunkten

□ Constraints

- bestimmen Layout
- separate Constraints für Reducer, Parser, Attribute
- Constraint-Manager frei wählbar, Standard: Qoca

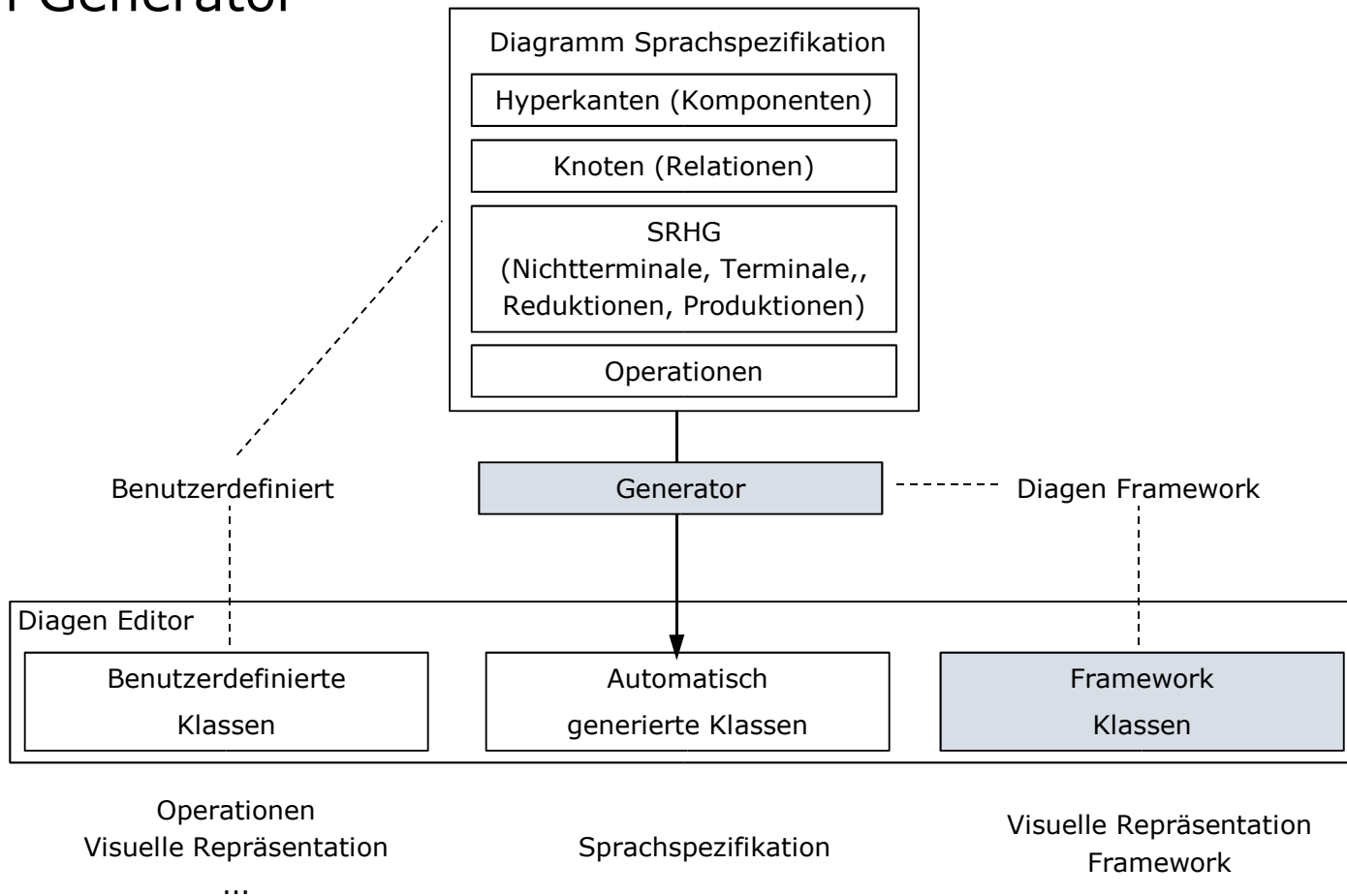


- Grammatik
 - attributiert
 - Nichtterminale/Terminale Hypergraphen
 - Starthypergraph
 - Reduktionen
 - nur einmal ausgeführt
 - keine Beschränkung auf Kontextfreiheit
(Regeln können auch "Nicht-Existenz" Bedingungen enthalten)
 - Produktionen
 - rekursive Ausführung auf Subgraphen
 - Kontextfrei
 - Beispiel:
 - $\langle \text{CircleSequence} \rangle \text{ l:S(a)} ::= [\text{r1:Circle(a)} \text{ r2:S(b)}]$
 - $\{ \text{constraints: l.x} = \text{r1.x}; \text{l.y} = \text{r1.y};$
 $(\text{r2.x} < \text{l.x} - \text{radius} \ || \ \text{l.x} + \text{radius} < \text{r2.x})$
 $\ || \ (\text{r2.y} < \text{l.y} - \text{radius} \ || \ \text{l.y} + \text{radius} < \text{r2.y}); \}$

□ Operationen

- erlauben automatisiertes ausführen komplexer Schritte
- bestehen aus einzelnen Teilregeln
- Beispiel:
 - rule connectCircle: circle(a) circle(b) do +arrow(a,b);
 - rule removeCircle: circle(a) do -circle(a);
 - rule addCircle: circle(a) do + circle(a);
 - operation replaceCircle "replace circle" "replaceimg.gif" :
 specify Circle c "specifiy circle to replace"
 do removeCircle(c)
 addCircle(a);

Diagen Generator



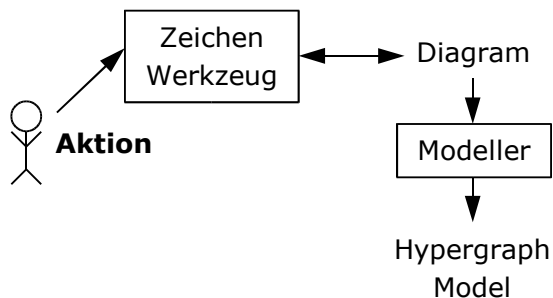
Editor

The screenshot shows the 'Generic DiaGen test editor' window. The interface includes a menu bar (File, Edit, View), a toolbar with icons for file operations and editing, and a left-hand sidebar with several sections: 'select mode' (with a mouse cursor icon), 'pan mode' (with a hand icon), 'zoom mode' (with a magnifying glass icon), 'Edit Properties:' (with an 'edit statement' button), 'Create Components:' (with buttons for 'Statement', 'Alternatives', and 'While loop'), and 'Operations:' (with buttons for 'insert text before', 'insert cond before', 'insert while before', 'insert text after', 'insert cond after', 'insert while after', 'remove text', and 'Move subdiagram after text').

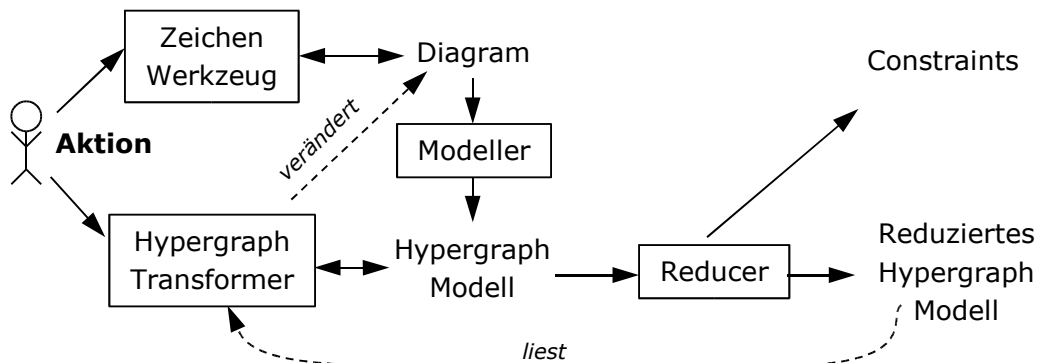
The main workspace displays a diagram with two 'Statement' nodes. The top node is a rectangular box with a red border and a red 'Statement' label. It contains a conditional structure with a 'y' branch, a 'cond' label, and an 'n' branch. The bottom node is a blue rectangular box with a blue 'Statement' label. Lines connect the diagram elements to labels on the right: 'Handles' points to the top corners of the red box, 'inkorrektter Subgraph' points to the red box, and 'korrekter Subgraph' points to the blue box. On the left, 'verfügbare Komponenten' points to the 'Create Components' section, and 'auf aktuelle Komponente anwendbare Operationen' points to the 'Operations' section. On the right, 'Bearbeitungs Modus' points to the toolbar.

- unterschiedliche Bearbeitungsmodi
 - Freihand Modus
 - Grammatik unabhängige Bearbeitung möglich
 - Syntax-Directed-Editing Modus
 - in korrekten Subgraphen nur Syntaktisch korrekte Bearbeitungsschritte möglich
 - Änderungen werden Reduziert und per Produktionen validiert
 - CYK ähnlicher Parser - Erkennung von korrekten Subgraphen
 - visuelle Anzeige korrekter und inkorrekt Subgraphen

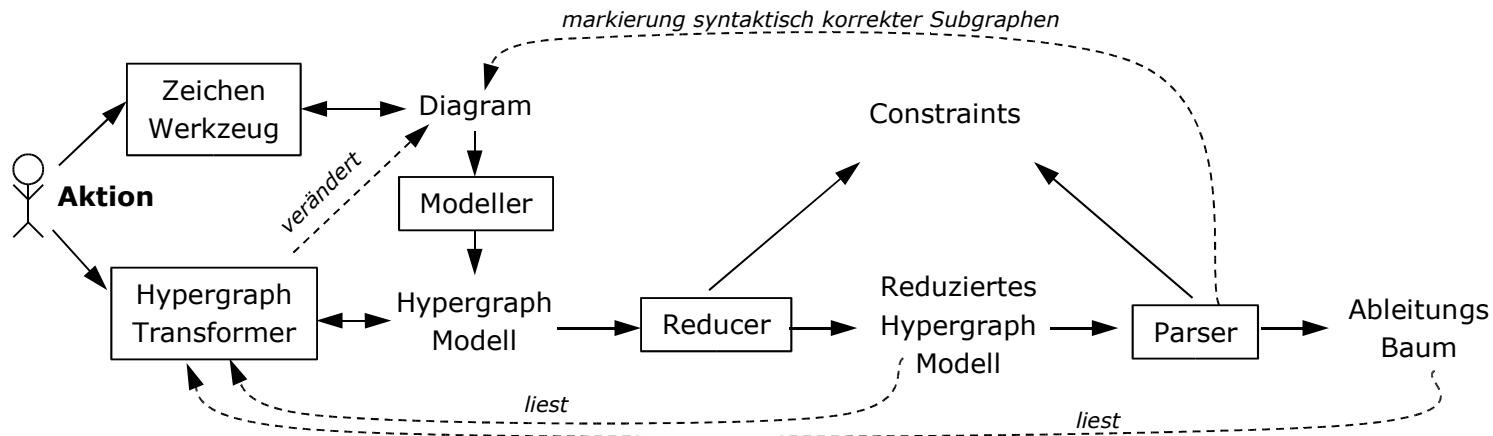
- Editor Programmablauf nach Änderung
 - Änderung propagiert durch Diagramm (Freihand Bearbeitung)
 - Modeler
 - Generierung eines (verschiedener) Hypergraphen für aktuellen Diagramm
 - Komponenten -> Hyperkanten, Attachment Areas -> Knoten
 - Spatial Relations -> Relations-Kanten zwischen Knoten



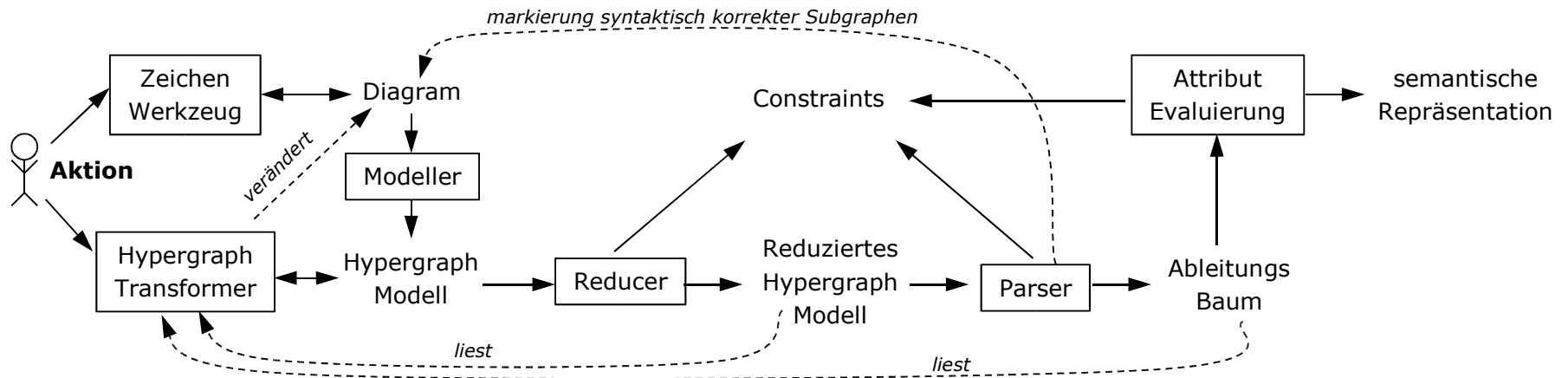
- Editor Programmablauf nach Änderung
 - Änderung propagiert durch Hypergraph Transformer (Syntaktisch Direkte Bearbeitung)
 - direkte Auswirkung auf Hypergraph (Komponenten hinzufügen/entfernen)
 - Reducer
 - Generierung eines Reduzierten Hypergraphen
 - äquivalent zu Lexikalischer Analyse im Compilerbau
 - Sammlung der Constraints reduzierter Knoten
 - Eliminierung nicht in Reduktion enthaltener Knoten



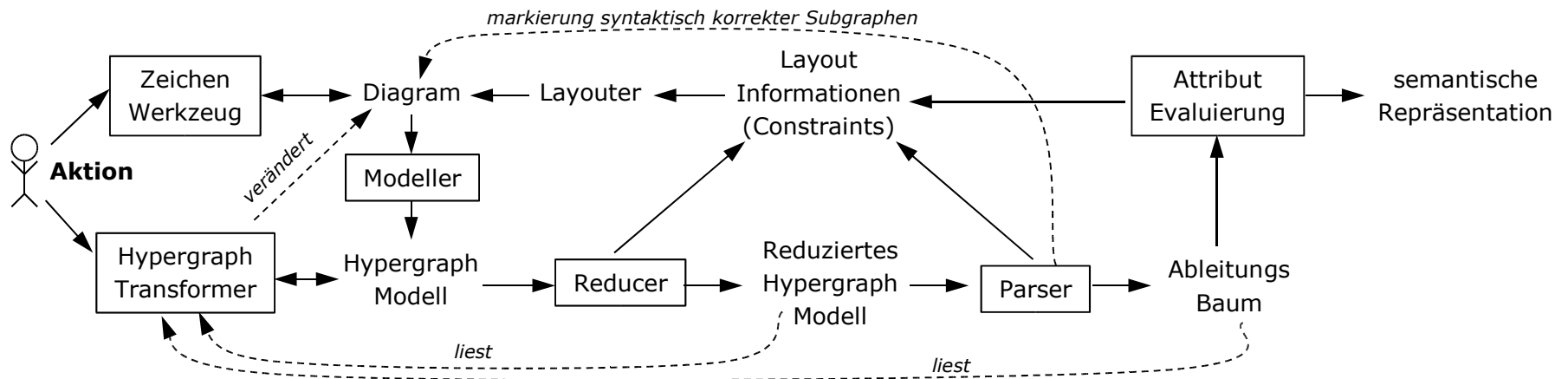
- Parser
 - Generierung eines gerichteten Azyklischen Ableitungsbaumes
 - äquivalent zu Syntaktischer Analyse im Compilerbau
 - visuelle Anzeige syntaktisch korrekter und inkorrektter Subgraphen im Diagramm
 - Sammlung der Constraints produzierter Knoten

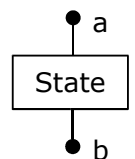
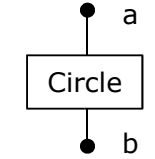
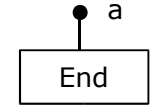
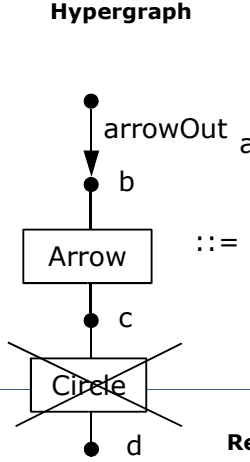
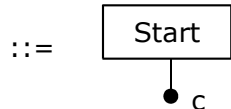
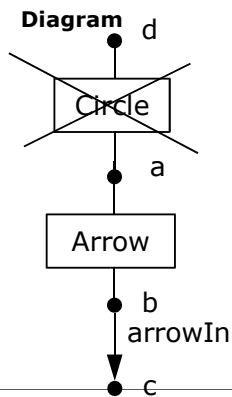
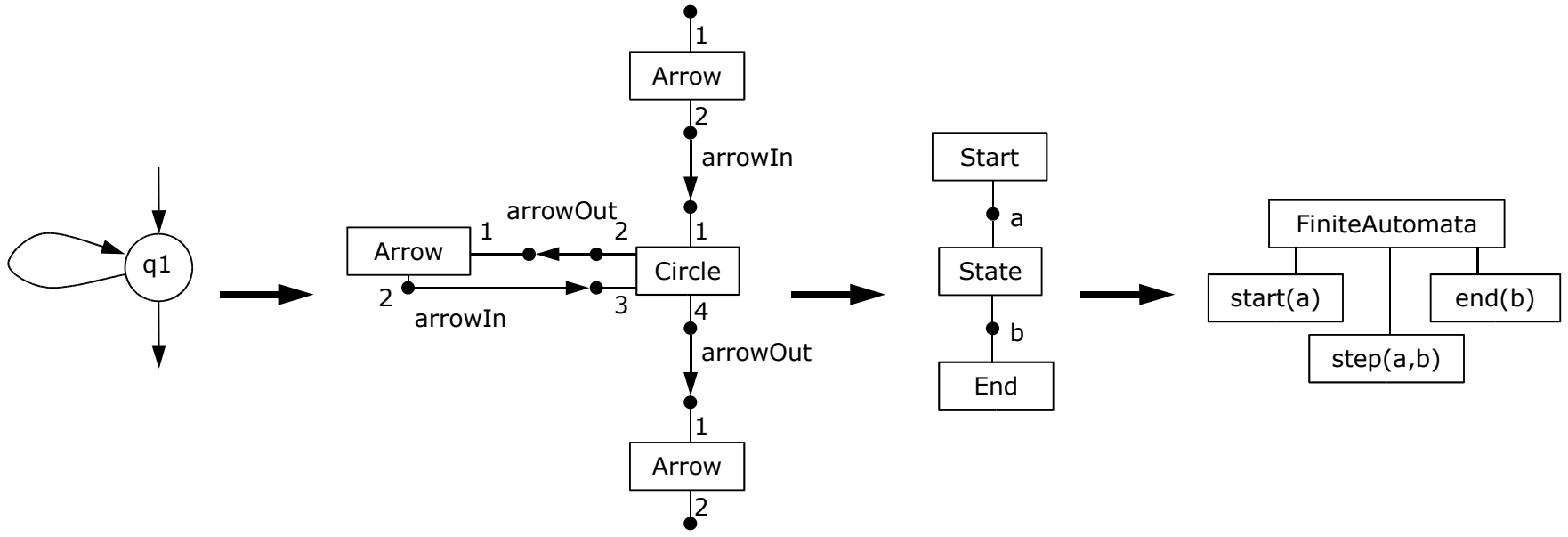


- **Attribut Evaluation**
 - Generierung einer gewünschten Datenstruktur
 - Berechnung der Werte anhand Produktionsregeln
 - Sammlung der Constraints attributierter Knoten



- Layouter
 - wird bei Freihand Bearbeitung nicht durchgeführt
 - Verwendung von Constraints aus Reducer, Parser und Attribut Evaluierung
 - Berechnung mit Hilfe eines Constraint Managers (derzeit Qoca)





Reduktionen



Einordnung

- persönliche Bewertung
 - Erzeugung allgemeiner Visueller Diagramm Editoren
 - Individuelle Anpassung des "Visuellen Grades" des Editors durch Programmierer
 - schnelle Erzeugung einfacher (Syntaxloser) Editoren
 - schrittweise Erweiterung zu komplexeren Editoren möglich
 - vorhandene Komponenten Bibliothek
 - Erweiterungsmöglichkeit dank Modularer Architektur und Editor als Sammlung von Javaklassen
 - "Self-hosted"
 - fortlaufende Entwicklung
 - Forschungsprojekt

- Aber
 - Sprachspezifikation Komplex
 - keine eingebaute Möglichkeit (Sub-)Graphen auszublenden (vorhandener Ansatz muss per Transformation implementiert werden)
 - wenig (aktuelle) Dokumentation
 - versteckte Features (Visuelle Regelerzeugung) nicht dokumentiert
 - nur ein Hauptentwickler
 - Forschungsprojekt

Quellen

- [Berge 89] Berge,C.(1989): Hypergraphs . Amsterdam
- [Drewes 91] Drewes,F.;Kreowski,H.-J.(1991): A note on hyperedge replacement .
In: Proc. 4th Int. Workshop on Graph Grammars and their Application
to Computer Science, S.1-11
- [DiaGen] DiaGen Homepage .
<http://www2-data.informatik.unibw-muenchen.de/DiaGen/>
- [Koeth 99] Köth,O.(1999): Design and Implementation of a Generic Graphical
Editor
- [Minas 93] Minas,M.; Viehstaedt,G.(1993): Specification of Diagram Editors
Providing Layout Adjustment with Minimal Change . In: Proc. 1993
IEEE Symposium on Visual Languages : IEEE Computer Society Press,
S.324-329
- [Minas 94] Minas,M.; Viehstaedt,G.(1994): Interaction in Really Graphical User
Interfaces . In: Proc. 1994 IEEE Symposium on Visual Languages :
IEEE Computer Society Press, S.270-277
- [Minas 95] Minas,M.; Viehstaedt,G.(1995): *A Generator for Diagram Editors
Providing Direct Manipulation and Execution of Diagrams* . In: Proc.
1995 IEEE Symposium on Visual Languages : IEEE Computer Society
Press, S.203-210

- [Minas 97] Minas,M.(1997): Diagram Editing with Hypergraph Parser Support . In: Proc. 1997 IEEE Symposium on Visual Languages : IEEE Computer Society Press, S.230-237
- [Minas 02] Minas,M.(2002): Concepts and realization of a diagram editor generator based on hypergraph transformation . In: *Science of Computer Programming*, 2002(44), S.157-180
- [Qoca] Qoca Homepage . <http://www.csse.monash.edu.au/projects/qoca/>

Bildquellen (wenn nicht anders angegeben):

- Seite 7 (von l. u. nach r. o.): [*Minas 93*], [*Minas 94*], [*Minas 95*], [*Minas 97*]

Diagen

