



# Multimediale Algorithmen und Datenstrukturen Assessments

Entwicklung eines Assessment-Systems zur  
Generierung, Durchführung und Auswertung von  
Aufgaben im Bereich Algorithmen und  
Datenstrukturen

Markus Krebs



# Gliederung des Vortrags

## ■ Theorie

- Grundlagen
- Editierung (Fibonacci-Heap)
- Automatische Generierung von Aufgaben
- Feedback
- Evaluation
- Plug-In-Konzept

## ■ Praxis

- Vorstellung der Programme

## ■ Diskussion



# Assessment: 3 Phasen

- Vorbereitung
  - Generierung durch Person vs. Automatisch
- Durchführung
  - Bearbeitung der Aufgaben durch den Studenten
- Evaluation
  - Auswertung durch Person vs. Automatisch



# Technologische Realisierungen

- Web-basierte Systeme

- HTML

- CGI

- Java-Script

- Plug-Ins (Shockwave, AuthorWare)

- Java-Applets

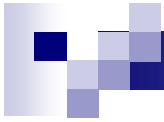
- Stand-Alone Applikationen

- Höhere Programmiersprachen (Java, C++, etc.)



# Kategorien von Assessment-Systemen

- Systeme die den Work-Flow unterstützen
  - WebAssign
- Klassische Self-Assessments
  - CUE, Perception, TSA
- Komplexe Self-Assessments
  - Exorciser, TRAKLA2, MAUDA
- Intelligente tutorielle Systeme (ITS)
  - QuizPACK (nur teilweise)



# Aufgabentypen

- Normal-Mode

- Vorgegebene Grundoperationen korrekt durchführen

- Fault-Mode

- Finden eines Fehlers in einer Aufgabe



# Editierung (Fibonacci-Heap)

- Grundoperationen aufgeteilt in Teiloperationen:
  - newfheapmeld x
  - setkey x y
  - cut x
  - mark x
  - unmark x
  - link x y
  - updatemin x
  - remove x
- Beispiel: delete-min
  - remove 3, link 8 71, link 8 16, link 8 19, updatemin 8



# Automatische Generierung (1)

## ■ Init

- Erzeugen einer Ausgangsstruktur

## ■ Scramble

- Durcheinanderbringen der Ausgangsstruktur
- Beachtung des Schwierigkeitsgrades

## ■ AutoGen

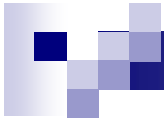
- Erzeugen der Aufgabenstellung
- Bestimmter Schwierigkeitsgrad





# Automatische Generierung (2)

- Suche im Zustandsraum mit Bewertung
- Jede Teiloperation erhält ein definierbares Gewicht
  - Schwerpunkt einer Aufgabe definierbar
- Menge von Grundoperationen (nur ID's) vorgegeben
- Mischen
- 4er Blöcke bilden
- Permutationen der Blöcke berechnen
- Jede Permutation mit allen verschiedenen Parametern durchrechnen
- Permutation bestimmten Schwierigkeitsgrades auswählen



# Feedback-Typen

- Immediate Feedback & Error Correction
- Error-Flagging
- Demand-Feedback
- No-Tutor
- Fault-Mode Feedback



# Angepasstes Feedback

- Knowledge-Base
  - Formeln
    - Randbedingungen
    - Typische Fehlersituationen
  - Feedback-Sätze



# Knowledge-Base

## → Einträge

- FORMULA: `firstincorrect AND subop[current].id = link AND op[current].id = delete_min AND subop[current].param1.pre.rank != subop[current].param2.rank`
- FORMULA: `firstincorrect AND subop[current].id = link AND op[current].id = delete AND correctsubop[0].id = remove AND subop[current].param1.pre.rank != subop[current].param2.rank`
- MESSAGE: Two nodes from the rootlist were linked, only when they have the same outdegree. And this is here not the case.



# Knowledge-Base

## → Allgemeine Auswertung

- FORMULA: TRUE AND link = link AND delete = delete\_min AND 5[2].pre.rank != 6[2].rank
- FORMULA: TRUE AND link = link AND delete = delete AND remove = remove AND 5[2].pre.rank != 6[2].rank
- MESSAGE: Two nodes from the rootlist were linked, only when they have the same outdegree. And this is here not the case.



# Knowledge-Base

## → Plug-In-spezifische Auswertung

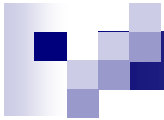
- FORMULA: TRUE AND link = link AND delete = delete\_min AND 4 != 5
- FORMULA: TRUE AND link = link AND delete = delete AND remove = remove AND 4 != 5
- MESSAGE: Two nodes from the rootlist were linked, only when they have the same outdegree. And this is here not the case.



# Knowledge-Base

## → Allgemeine Auswertung

- FORMULA: TRUE AND TRUE AND FALSE AND TRUE → **FALSE**
- FORMULA: TRUE AND TRUE AND TRUE AND TRUE AND TRUE → **TRUE**
- MESSAGE: Two nodes from the rootlist were linked, only when they have the same outdegree. And this is here not the case.



# Evaluation

- automatisch

- Grundlage: Bewertung der Teiloperationen
- Jeder Fehler wird protokolliert

- tutoriell

- JEDAS-Annotations-System





# PlugIn-Konzept

- JEDAS-Animations-Klassen
- Analyse-Modul
- Interaktionskomponente (Maus-gesteuert)
- Abbildungs-Modul
- Formel-Auswertung
- Datenverzeichnis



# Synchronisation mit JEDAS

## ■ Messages

- SPECIAL
- OPERATION\_EXECUTED
- FILENAME\_CHANGED
- CLEAR\_PRECEDING
- CLEAR\_FOLLOWING
- FINISHED\_LOADING
- DSOBJECT\_HIDE
- DSOBJECT\_SHOW
- UNDO
- REDO
- REDO\_ANIMATED
- RESET
- PLUGIN\_CHANGED
- JUMP
- END\_OF\_EXECUTION
- START\_OF\_EXECUTION

## ■ Gesteuerte Objekte

- OperationRecorder
- UndoRedo
- Exercise-Klassen
- TreeOperationView
- GenEditor: Reiter
  - Exercise
  - Operations
- WorkEditor: Feedback
  - Immediate Feedback & Error Correction
  - Error-Flagging
  - Demand-Feedback
  - No-Tutor
  - Fault-Mode Feedback
- EvalEditor+EvalPlayer
  - CommentView
  - TimeStampRecorder
- Etc.



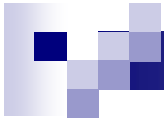
# JEDAS: Bemerkungen

- Animationsklassen

- Trennung Logik von Darstellung

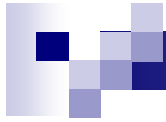
- Bug

- Aufnahmen sind nur abspielbar, wenn die Aufnahme sofort nach Programmstart gestartet wird.
- JEDAS-Player



# Praxis

- Vorstellung der Applikations-Pakete
  - GenEditor
  - WorkEditor
  - EvalEditor
  - EvalPlayer



# Diskussion

Fragen ?