

VISUELLE SPRACHEN

Einführung und Organisatorisches

Berthold Hoffmann
Seminar *Visuelle Sprachen*, Sommer 2004
Donnerstag, den 21. April 2004

Gestatten, ...

- ✿ ... Berthold Hoffmann
 - Diplom (78) und Promotion (83) an der TU Berlin
 - wissenschaftlicher Mitarbeiter (AG BKB, seit 82)
 - Lehre:
 - Praktische Informatik 3 (FUN)
 - Programmiersprachen
 - Übersetzer
- ✿ MZH 8130, Tel. 218-2450, hof@tzi.de

meine Motivation

- ✿ **mein Hintergrund**
 - Übersetzer- Werkzeuge
 - Programmiersprachen (funktional, objektorientiert)
 - Graphtransformation
- ✿ **Synthese**
 - Graphen sind „visuelle Daten“
 - Transformationsregeln sind „visuelle Programme“
- ✿ **Forschungsrichtung**
 - Implementierung visueller Sprachen
 - Eine Programmiersprache für die Diagramm-Manipulation

Lehrziele

- ✿ Konzepte und Implementierungstechniken von visuellen Sprachen studieren
- ✿ lernen, Vorträge auszuarbeiten
- ✿ lernen, Texte zu schreiben
- ✿ Anteile: 70% - 20% - 10%

und Ihre / Eure Motivation?

Motto

- ✿ *Ein Bild sagt mehr als 10 000 Worte*
[Konfuzius]
- ✿ *An algorithm must be **seen** to be believed, and the best way to learn what an algorithm is all about is to **try** it*
[D.E. Knuth, *Art of Computer Programming*]

Hoffnung in Bilder statt Texte

Bilder sind ...

- intuitiver
- internationaler
- konkreter

... als Text

🐼 Stimmt das wirklich?

Geschichte

Die Sechziger

- 60: Flussdiagramme aus Fortran-Programmen
- 63: Zeichenprogramm *Sketchpad* (I.E. Sutherland)
- 66: *Graphical Program Editor* (W.R. Sutherland)
- 69: *Grail* (Ellis et al.)
- 68-71: *AMBIT/G* und *AMBIT/L* (Christensen)

Die Siebziger

- 🐼 Pygmalion (D.C. Smith, Stanford)
 - *programming by example*
- 🐼 Programming Support System (Frei u.a., IBM San José)
 - Erstellen von Struktogrammen und Übersetzen nach PL/1
- 🐼 PIGS (Pong und Ng, Hong Kong)
 - Erstellen und von Struktogrammen und Animation ihrer Ausführung
- 🐼 QBE (Zloof, IBM Yorktown Heights)
 - Büroautomation
- 🐼 VisiCalc
 - Tabellenkalkulation

Die Achtziger

- 🐼 "Visuelles Programmieren" als Begriff (MacDonald 1983)
- 🐼 Prograph (Matwin und Pietrzykowski, Nova Scotia)
 - datenflussorientierte funktionale objektorientierte Programmiersprache
- 🐼 "Visual Language Community" und Konferenz (1984)
- 🐼 PICT (Glinert, Tanimoto 1984)
 - tastaturloses Programmieren auf Pascal-Basis
- 🐼 LabVIEW (Kodosky, National Instruments, 1986)
 - *virtuelle Instrumente* grafisch programmieren

Die Neunziger

- 🐼 Visuelles objektorientiertes Programmieren
 - VIPR
- 🐼 Visuelle Softwareentwicklung
 - Unified Modeling Language (UML)
- 🐼 Reasoning with Diagrams
 - Diagrams-Konferenzserie
- 🐼 Tools für visuelle Editoren
 - VLCC, ProGreS, DiaGen

aktuelle Forschung

- 🌟 Visuelle Sprachen
 - Syntax, Piktogramme, Theorie
- 🌟 Visuelle Programmierung
 - *programming by example*
 - end user programming
- 🌟 Softwarevisualisierung
 - Daten- und Programm-Visualisierung und -Animation
- 🌟 Mensch-Maschine-Kommunikation (*Human-Computer-Interaction, HCI*)
 - Diagrammgestaltung, Benutzungsschnittstellen, Programmiersysteme
- 🌟 Anwendungsgebiete
 - geographische und Bild-Informationssysteme

visuell: *das Sehen betonend*

- 🌟 visuelle Benutzungsschnittstelle
- 🌟 visuelle Programmierumgebung
- 🌟 Visualisierung von Algorithmen
- 🌟 visueller Entwurf von Software
- 🌟 visuelles Programmieren
- 🌟 visuelle Spezifikationsprache
- 🌟 visuelle Programmiersprache

visuelle Benutzungsschnittstelle

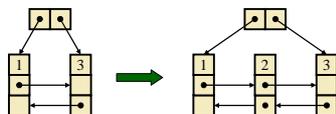
- 🌟 Fenster, Tastatur und Maus
- 🌟 grafische Bedienelemente
 - Knöpfe, Textflächen, Rollbalken, Bäume
 - aber *keine* Diagramme oder Bilder! (außer Ikonen)
- 🌟 Schnittstelle zu Texten, Dateien usw.

visuelle Programmierumgebung

- 🌟 Visuelle Benutzungsschnittstelle für klassische textuelle Programmiersprachen
- 🌟 die semantischen Objekte sind Texte und Dateien

Algorithmen visualisieren

- 🌟 Daten werden grafisch dargestellt
- 🌟 textuelle Programme werden *animiert*
- 🌟 Ziel: Verstehen von Algorithmen



visueller Entwurf von Software

- 🌟 UML: Klassendiagramme, Aktivitätsdiagramme
- 🌟 Prozesse: Statecharts, Petrinetze
- 🌟 Kontrolle: Flussdiagramme, Struktogramme

visuelles Programmieren

- ✦ Programmieren mit Bildern (Diagrammen)
- ✦ eine neue Syntax für existierende Paradigmen
- ✦ Beispielsprachen
 - spezielle Sprachen (Labview)
 - objektorientierte Sprachen (VIPR)
 - funktionale Sprachen (Prograph)

Sprachen ...

- ✦ ... dienen der Kommunikation
 - gesprochen / gehört
 - geschrieben / gelesen
 - gezeichnet / betrachtet
- ✦ ... zwischen ...
 - Menschen
 - Menschen und Computern
 - Computern

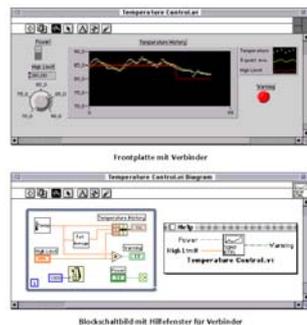
visuelle versus textuelle Sprachen

Visuell	textuell
Ein Bild sagt mehr als 1000 Worte	1000 Worte brauchen weniger Platz als ein Bild
Gestaltungsprinzipien unklar	Gestaltungsprinzipien klar
aufwändige Implementierung	einfache (bekannte) Implementierungstechnik

Beispiele

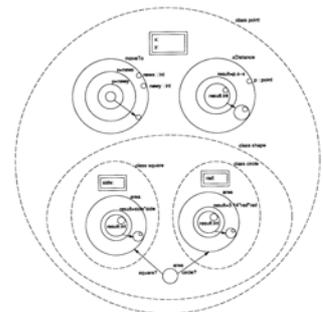
Labview

- ✦ Virtuelle Messgeräte "programmieren"



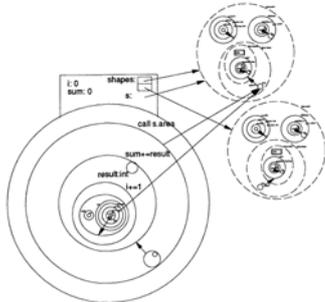
VIPR (Visual Programming)

- ✦ Visuelle Darstellung der Klassenstruktur



VIPR (Visual Programming)

- 🌟 Animation des Kontrollflusses



Prograph

- 🌟 vorübergehender Abstieg zu oligomedialer Vortragsweise
- auf Deutsch: *Folien*

Organisation und Ablauf

Vorkenntnisse?

- 🌟 **Seminarschein**
 - wieviele wollen einen?
- 🌟 **Vorkenntnisse**
 - Graphtransformation?
 - Programmiersprachen
 - Übersetzer
 - Erfahrung mit visuellen Sprachen?

Ablauf eines Seminartermins

- 🌟 Vortrag von ca. einer Stunde
 - Tafel, Folien oder Bildschirmpräsentation
 - Rückfragen zulassen
- 🌟 Diskussion über den Inhalt
 - Bezug zu den anderen Vorträgen herstellen
- 🌟 Tipps zum Vortrag
 - Verbesserungsvorschläge ...

Was ist zu tun?

- 🌟 Recherche zum Thema
- 🌟 Material studieren
- 🌟 Vortrag ausarbeiten
- 🌟 Vortrag halten
- 🌟 Zusammenfassung schreiben
 - ggf. überarbeiten

Termine

- ✿ bis Montag: Auswahl eines Themas
- ✿ sofort danach: Recherche
- ✿ so bald wie möglich: Material studieren
- ✿ 3 Tage vorher: Vortrag mit mir besprechen
- ✿ spätestens Ende Juli: Zusammenfassung
- ✿ ggf. Überarbeitung bis September

Recherche

- ✿ LV-Seite www.tzi.de/~hof/Lehre/VS/
 - Themenliste und Zeitplan des Seminars
 - Liste der teilnehmenden Studierenden
 - Quellen und Links zu den Themen des Seminars
 - Stefan Schiffers Buch *Visuelle Programmierung online*
 - *Folien und schriftliche Zusammenfassungen (nach und nach)*
- ✿ Seminarordner
- ✿ Bibliothek
- ✿ WWW

studienbegleitende Leistungsnachweise

- ✿ einen Vortrag halten
 - 60 Minuten
- ✿ eine Zusammenfassung schreiben
 - 6-10 Seiten
- ✿ regelmäßige aktive Teilnahme
- ✿ Hintergrundwissen aneignen (wenn nötig)
 - Programmiersprachen und Implementierung
 - Schiffer: *Visuelle Programmierung*
- ✿ allein oder zu zweit

Wichtig für die Beurteilung

- ✿ **Vortrag**
 - sorgfältig einarbeiten
 - Thema vollständig und verständlich darstellen
 - Bezug zu anderen Vorträgen herstellen
 - Medien adäquat einsetzen
- ✿ **Zusammenfassung**
 - inhaltlich korrekt beschreiben
 - verständlich darstellen (Beispiele, ...)
 - Beschreibung von der Bewertung abgrenzen
 - Text angemessen gestalten

Allgemeine Literatur

- ✿ Burnett, Goldberg, Lewis (eds.):
Visual Object-Oriented Programming,
Manning, Greenwich, CT 1995
- ✿ Marriott, Meyer (eds.): *Visual Language Theory*,
Springer 1996
- ✿ Schiffer: *Visuelle Programmierung*,
Addison-Wesley 1998

Themen

- ✿ Einführung
Vortragstechnik
- ✿ Sprachen / Systeme
 - Labview
 - Forms/3
 - Prograph
 - VIPR
 - Vista
- ✿ Werkzeuge
 - ProGreS
 - VLCC / VLDesk
 - **DiaGen**
 - **Diaplan**
- ✿ Schlussbesprechung