

Belegarbeit zum Thema

Entwicklung  
einer interaktiven grafischen Benutzeroberfläche  
für den Prozesssimulator Dupsim

Martina Burghagen

01.11.06

## **Gliederung:**

1. Benutzeroberflächen
  - 1.1 Softwareergonomie – Definition
  - 1.2 Softwareergonomie – Disziplinen
2. Die europäische Norm
  - 2.1 Grundsätze der Dialoggestaltung
  - 2.2 Anforderungen an die Gebrauchstauglichkeit
3. Das Programm
  - 3.1 Der Funktionsumfang
  - 3.2 Entwurfsentscheidungen und Implementierung
  - 3.3 Verwendete Klassen
  - 3.4 Die Simulation
4. Zusammenfassung

### 1. Benutzeroberflächen

#### **Definition:**

Eine Benutzerschnittstelle, eigentlich Benutzungsschnittstelle, ist der Teil eines Programms, der den Datenaustausch mit dem Benutzer durchführt.

#### **Aufgaben:**

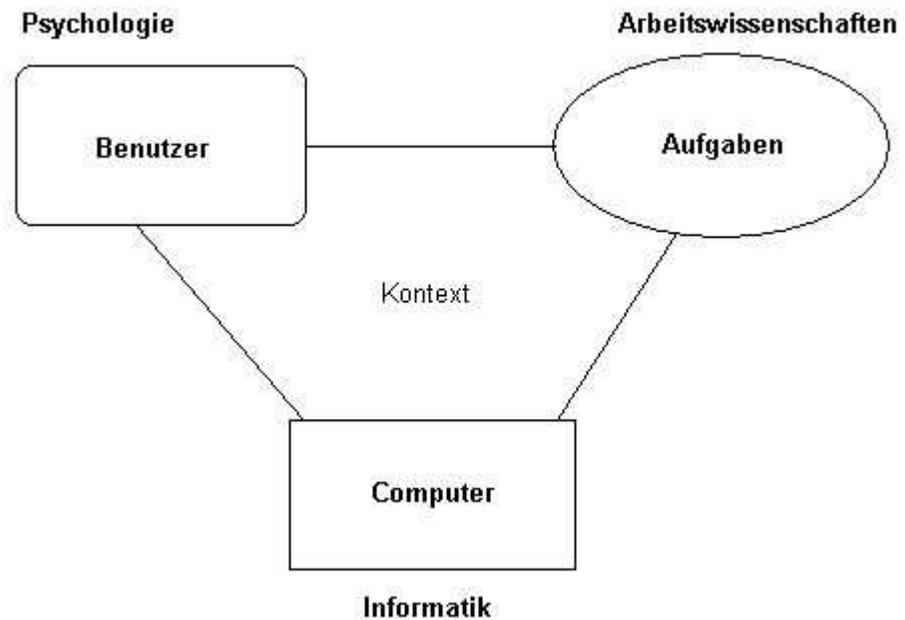
- Informationen müssen angezeigt werden.
- Eingabemöglichkeiten für Befehle müssen bereitgestellt werden.
- Eingaben müssen verarbeitet werden.

### **1.1 Softwareergonomie - Definition**

Die Ergonomie ist die Wissenschaft von der Gesetzmäßigkeit menschlicher Arbeit. Zentral ist dabei die Verbesserung der Mensch-Maschine-Schnittstelle zwischen Benutzer und Objekt in einem Mensch-Maschine-System.

Gegenstand der Software Ergonomie ist die Anpassung der Arbeitsbedingungen bei der Benutzer - Computer - Interaktion an die sensomotorischen und kognitiven Fähigkeiten und Prozesse des Menschen.

## 1.2 Softwareergonomie - Disziplinen



## 2.1 Grundsätze der Dialoggestaltung

### 1. Aufgabenangemessenheit:

Ein Dialog ist aufgabenangemessen, wenn er den Benutzer unterstützt, seine Arbeitsaufgabe effektiv und effizient zu erledigen.

## 2.1 Grundsätze der Dialoggestaltung

### 2. Selbstbeschreibungsfähigkeit:

Ein Dialog ist selbstbeschreibungsfähig, wenn jeder einzelne Dialogschritt durch Rückmeldung des Dialogsystems unmittelbar verständlich ist oder dem Benutzer auf Anfrage erklärt wird.

## 2.1 Grundsätze der Dialoggestaltung

### 3. Steuerbarkeit:

Ein Dialog ist steuerbar, wenn der Benutzer in der Lage ist, den Dialogablauf zu starten sowie seine Richtung und Geschwindigkeit zu beeinflussen, bis das Ziel erreicht ist.

## 2.1 Grundsätze der Dialoggestaltung

### 4. Erwartungskonformität:

Ein Dialog ist erwartungskonform, wenn er konsistent ist und den Merkmalen des Benutzers entspricht, z.B. seinen Kenntnissen aus dem Arbeitsgebiet, seiner Ausbildung und seiner Erfahrung sowie den allgemein anerkannten Konventionen.

### 2.1 Grundsätze der Dialoggestaltung

#### 5. Fehlertoleranz:

Ein Dialog ist fehlertolerant, wenn das beabsichtigte Arbeitsergebnis trotz erkennbar fehlerhafter Eingaben entweder mit keinem oder mit minimalem Korrekturaufwand seitens des Benutzers erreicht werden kann.

## 2.1 Grundsätze der Dialoggestaltung

### 6. Individualisierbarkeit:

Ein Dialog ist individualisierbar, wenn das Dialogsystem Anpassungen an die Erfordernisse der Arbeitsaufgabe sowie an die individuellen Fähigkeiten und Vorlieben des Benutzers zulässt.

## 2.1 Grundsätze der Dialoggestaltung

### 7. Lernförderlichkeit:

Ein Dialog ist lernförderlich, wenn er den Benutzer beim Erlernen des Dialogsystems unterstützt und anleitet.

### **2.2 Anforderungen an die Gebrauchstauglichkeit**

Das Ausmaß, in dem ein Produkt durch bestimmte Benutzer in einem bestimmten Nutzungskontext genutzt werden kann, um bestimmte Ziele effektiv, effizient und zufrieden stellend zu erreichen.

## 3.1 Der Funktionsumfang



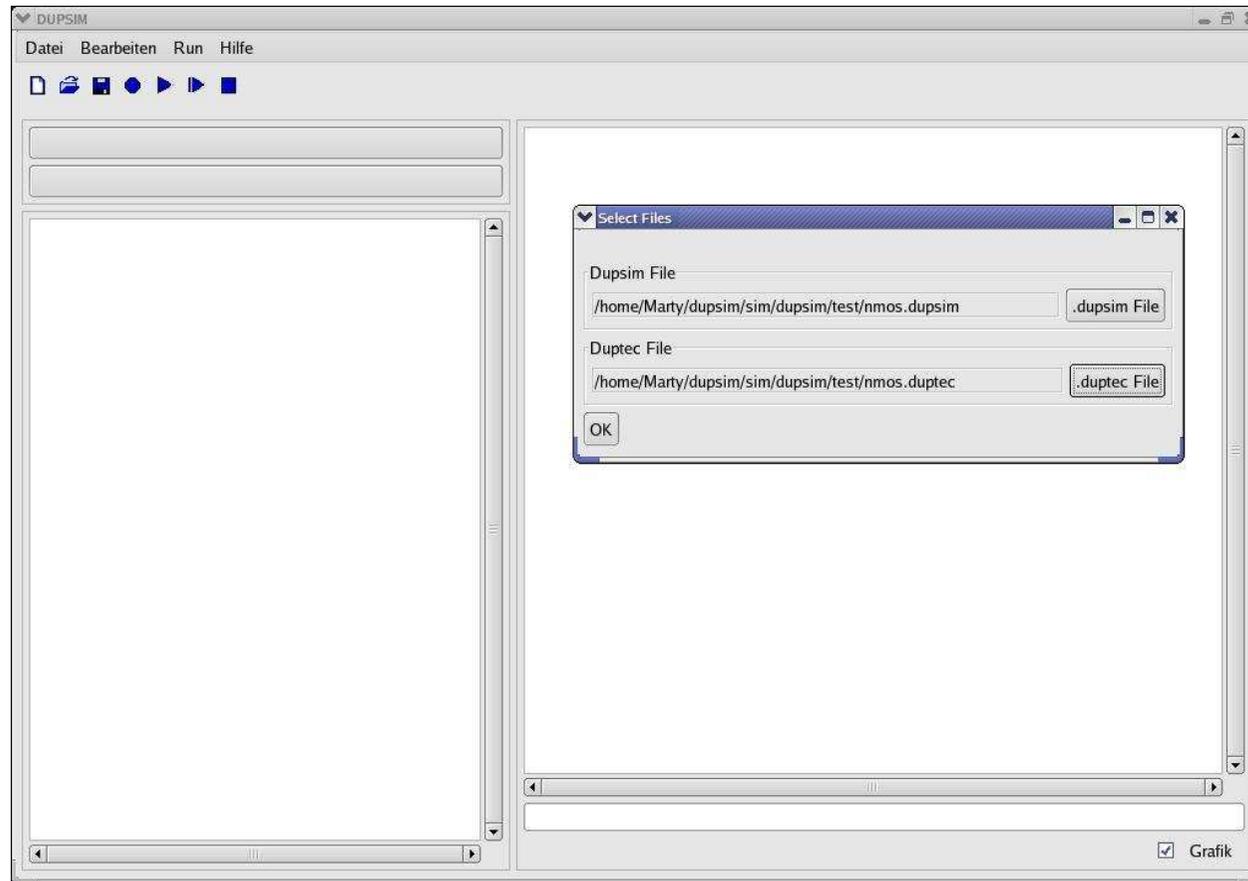
### 3.2 Entwurfsentscheidungen und Implementierung

- Implementierung in Java
- Grafikbibliothek SWT
- Steuerung von Dupsim über Runtime Klasse
- Zentrale Klasse: GUI
- Weitere wichtige Klassen: SimulationOutput, Editor und FileDisplay

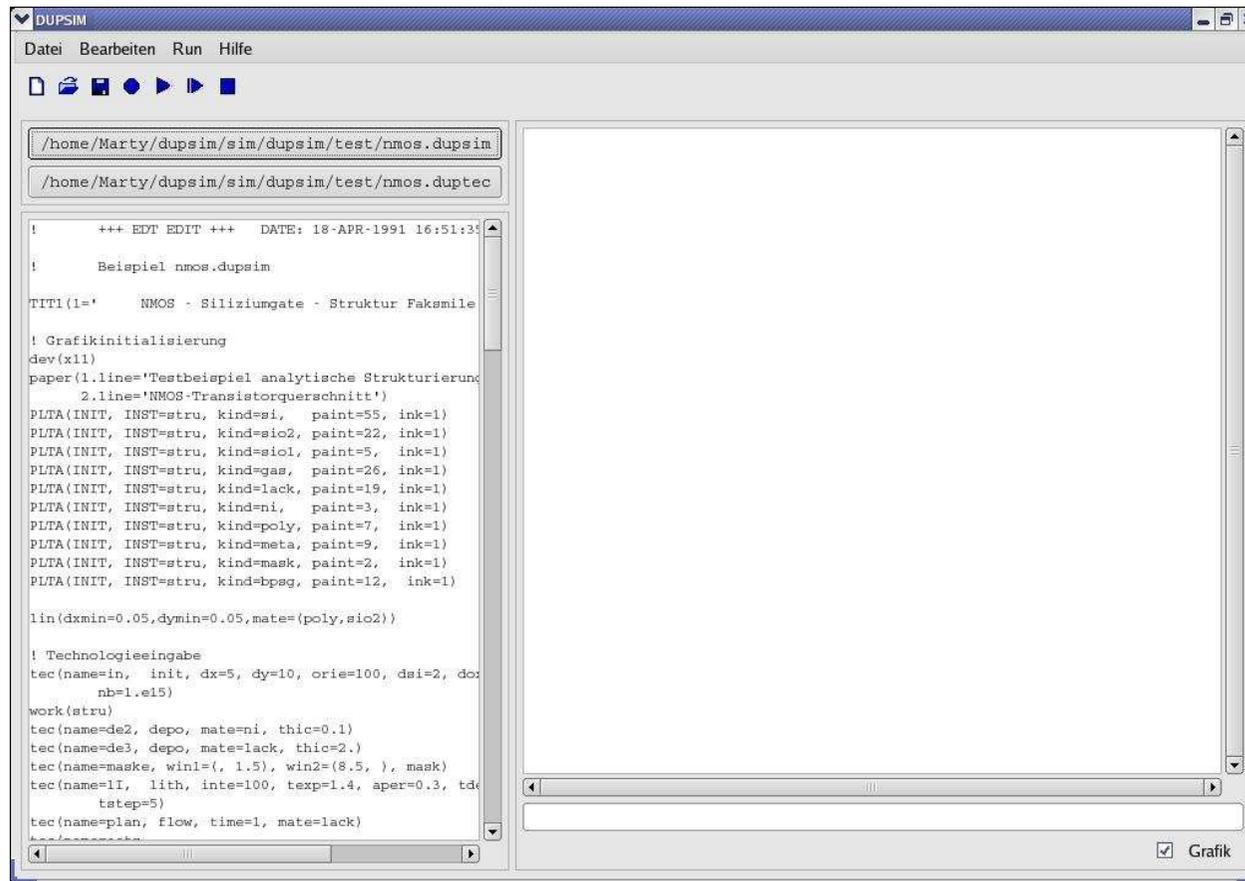
### 3.3 Verwendete Klassen

GUI:	→	main-Methode
Editor:	→	Anzeigen des Dateiinhalts
	→	Haltepunkte
	→	Stepmodus
SimulationOutput:	→	Anzeigen der Ausgabe
	→	Starten von Dupsim
FileDisplay:	→	Anzeigen der Dateinamen
CommandHelp	→	Befehlshilfe
TechnologyHelp	→	Technologiehilfe
FileSelect	→	Dateiauswahl
FileBrowse	→	Dateidialog
TransformFile	→	Veränderte Ausgabe
TextArea	→	Textanzeige

### 3.4 Simulation



## 3.4 Simulation



The screenshot shows the DUPSIM software interface. The main window is titled "DUPSIM" and contains a menu bar with "Datei", "Bearbeiten", "Run", and "Hilfe". Below the menu bar are several icons for file operations and simulation control. The interface is split into two main panes. The left pane is a text editor showing the following code:

```
+++ EDT EDIT +++ DATE: 18-APR-1991 16:51:35
!
! Beispiel nmos.dupsim
TIT1(1=' NMOS - Siliziumgate - Struktur Fakemile
!
! Grafikinitalisierung
dev(x11)
paper(1.line='Testbeispiel analytische Strukturierung
2.line='NMOS-Transistorquerschnitt')
PLTA(INIT, INST=stru, kind=sl, paint=55, ink=1)
PLTA(INIT, INST=stru, kind=sio2, paint=22, ink=1)
PLTA(INIT, INST=stru, kind=sio1, paint=5, ink=1)
PLTA(INIT, INST=stru, kind=gas, paint=26, ink=1)
PLTA(INIT, INST=stru, kind=lack, paint=19, ink=1)
PLTA(INIT, INST=stru, kind=ni, paint=3, ink=1)
PLTA(INIT, INST=stru, kind=poly, paint=7, ink=1)
PLTA(INIT, INST=stru, kind=meta, paint=9, ink=1)
PLTA(INIT, INST=stru, kind=mask, paint=2, ink=1)
PLTA(INIT, INST=stru, kind=bspag, paint=12, ink=1)
lin(dxmin=0.05, dymin=0.05, mate={poly,sio2})
!
! Technologieeingabe
tec(name=in, init, dx=5, dy=10, orie=100, dsi=2, do
nb=1.e15)
work(stru)
tec(name=de2, depo, mate=ni, thic=0.1)
tec(name=de3, depo, mate=lack, thic=2.)
tec(name=maske, win1=(, 1.5), win2=(8.5, ), mask)
tec(name=lI, lith, inte=100, texp=1.4, aper=0.3, tde
tstep=5)
tec(name=plan, flow, time=1, mate=lack)
```

The right pane is a graphics window, currently blank. At the bottom right of the graphics window, there is a checkbox labeled "Grafik" which is checked.