

SimuVis, netSim & SMILE

Grafisch gestützte Simulation in Netzwerken

Jörg Rädler, dezentral gbr
Andreas Gerber, FH Biberach

Bereits vorgestellt: SMILE und das SMILE-Gebäudemodell

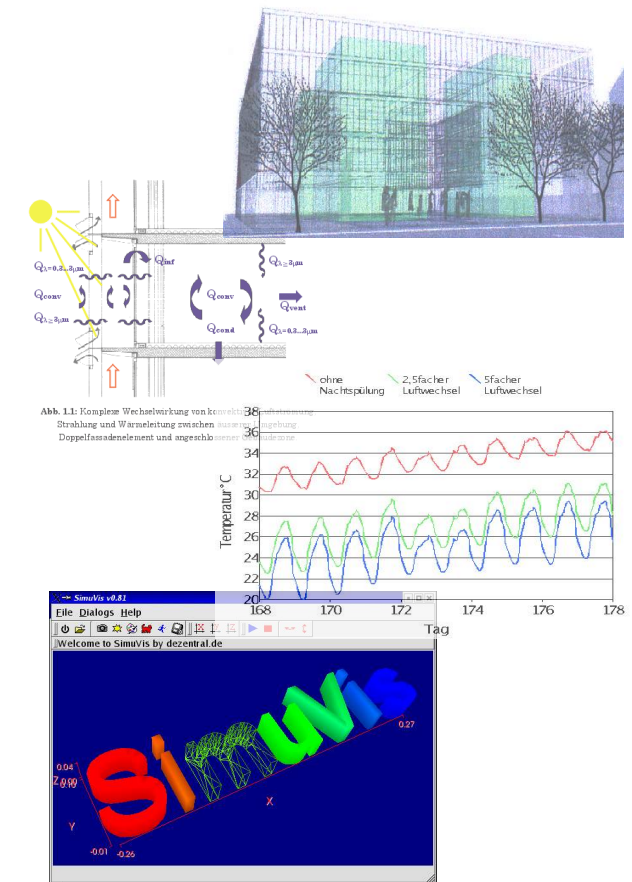
1. Vortrag

- Vorstellung dezentral
- Ausgangslage und Motivation
- Konzept der allgemeinen Bausteine
- Werkzeuge und Bibliotheken
- SimuVis – allgemeines GUI und Visualisierung
- netSim – Simulationsverwaltung im Netzwerk
- die Verwendung im Lernnetz Bauphysik - Kühllastberechnung

2. Software-Demonstration



- Energiekonzepte und Beratung
- Simulationen von Gebäuden und Anlagentechnik
- Software SMILE, SimuVis, ...
- Lehre



Befehlsfenster 2 - Konsole

IW	[time_rows.new]	Row 3	Col 1	4:1
time	environment.T_air_ref	zone_west.T		
0	266.35	293.15	293.15	0
3600	271.05	293.153	291.151	289.233
7200	270.25	293.15	291.15	289.011
10800	269.55	293.15	291.15	288.828
14400	269.149	293.15	291.15	288.689
18000	268.25	293.15	291.15	288.479
21600	267.45	293.15	291.15	288.274
25200	267.15	293.15	291.15	288.135
28800	267.049	293.15	291.15	288.032
32400	265.958	293.15	291.15	287.814
36000	265.95	293.15	291.15	287.706
39600	266.25	293.15	291.15	287.672
43200	266.05	293.15	291.15	287.563
46800	266.05	293.15	291.15	287.476
50400	266.15	293.15	291.15	287.386
54000	265.85	293.15	291.15	287.252
57600	265.35	293.15	291.15	287.079
61200	264.652	293.15	291.15	286.887
64800	264.35	293.15	291.15	286.748
68400	263.75	293.15	291.15	286.549
72000	262.95	293.15	291.15	286.34
75600	261.75	293.15	291.15	286.07
79200	261.85	293.15	291.15	285.971

emacs@rollmops.dezentral.de

```
File Edit Options Buffers Tools C++ Help

@eq cont thickness_wall(thickness_layer, n_layers)
{
    int j;
    double sum = 0.0;

    for(j = 0; j < (int)n_layers; j++)
        sum = sum + thickness_layer[j];
    return sum;
}

// Nettowandschichtmasse
@eq cont m_layer[j](n_layers, A, thickness_layer, rho_layer, thickness_wall)
{
    if (j < (int)n_layers) return A * thickness_layer[j] * rho_layer[j];
    return 0.0;
}

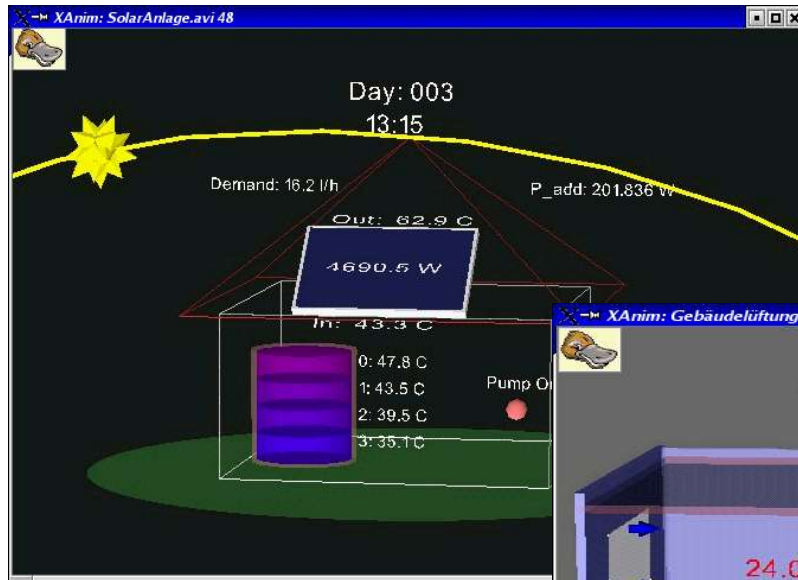
// Gesamtmasse aller Wandschichten (Nettowandmasse)
@eq cont m_wall(m_layer, n_layers)
{
    int j;
    double value = 0.0;

    for(j = 0; j < (int)n_layers; j++)
        value = value + m_layer[j];
    return value;
}

// Berechnung der inneren Oberflächentemperatur der Wand
@eq cont qdot_lw_ins, qdot_sw_ins, qdot_cv_ins, qdot_cd_ins
{
    return qdot_lw_ins + qdot_sw_ins + qdot_cv_ins + qdot_cd_ins;
}

// Berechnung der inneren Oberflächentemperatur der Wand
@eq cont T_lw_ins
{
    return T_lw_ins;
}
```

- computernahes Arbeiten (Texteditor, Kommandozeile, Compiler)
- dadurch maximale Flexibilität
- kaum Anschaulichkeit, besonders für Fachfremde
- wenig geeignet für schnelle Anwendungen



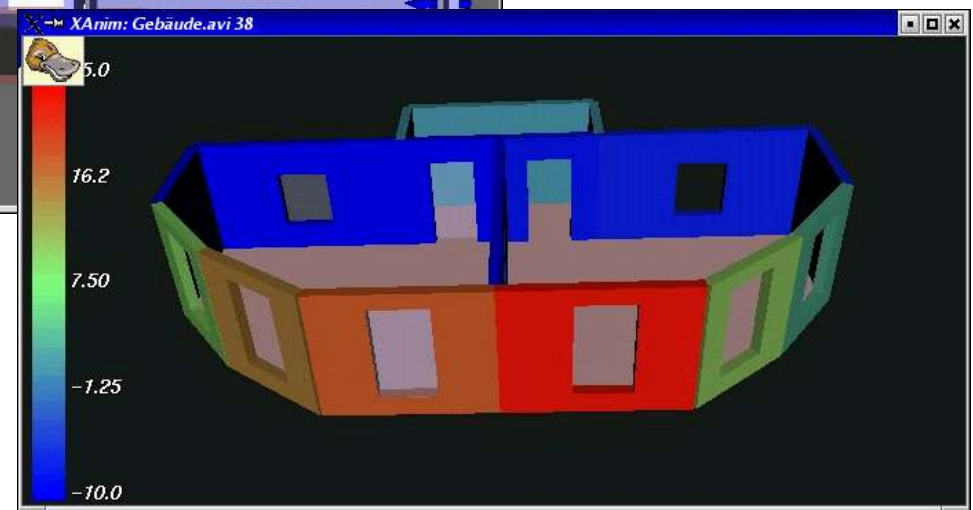
Solarthermische Anlage

Büroriegel mit Nachtauskühlung



Gebäude
mit TWD

mehrere einzelne
Demoprogramme
zur Visualisierung bestimmter
Simulationsmodelle und
-ergebnisse



- viele wiederkehrende Module ohne echtes Framework (z.B. Interaktionsmöglichkeiten, Synchronisation zwischen Datenaustausch und Anzeige)
 - Interaktion nur zur Darstellung
 - keine Eingabemöglichkeiten für Simulationsparameter
 - Simulationssystem muss fest konfiguriert sein
 - nur rudimentäre Netzwerkfähigkeiten
- **eingeschränkte Wiederverwendbarkeit**

■ Ansatz: Framework aus Bausteinen



1. Simulations-Backend:
SMILE mit Gebäudemodell



2. Verwaltungsebene:
netSim und angepasste
Task-Klassen

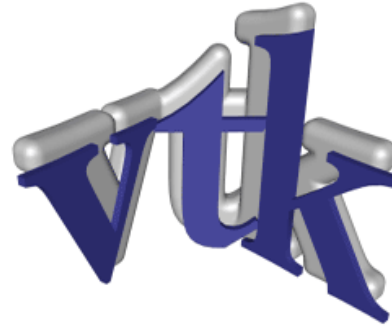


3. Frontend / GUI
Ein-/Ausgabe über SimuVis
und angepasste Szenen



Python

- Programmiersprache vergleichbar mit Perl, Java oder Tcl
- verbindet alle bekannten Plattformen
- Anbindung an C (+Dialekte) und Java
- unzählige Module
- perfekter „Kleber“ für bestehenden Code



- Toolkit für 3D- und 2D-Visualisierung
- Pipeline-Konzept
- interaktiv
- verschiedene Sprachbindungen
- unzählige Anwendungen in vielen Bereichen



- modernes, umfangreiches GUI-Toolkit
- Win32/UNIX/Mac
- verschiedene Sprachbindungen
- Erweiterungen (z.B. Qwt mit Plotfunktionen)

- Applikation zur Darstellung von 3D-Objekten



- Framework mit oft benötigten Funktionen:
Import / Export / Animation / Darstellung / Interaktion etc.
- ladbare Szenen bestimmen den Inhalt
- Szenen beschreiben Objekte und deren Eigenschaften
- Szenen können beliebigen Python-Code beinhalten und die Oberfläche beliebig erweitern und (um-)gestalten:
 - Ein- und Ausgabe
 - Netzwerkkommunikation
 - weitere GUI-Elemente

Szenen sind normale Python-Klassen (Erben einer Standardszene).

Spezielle Methoden werden zur Anpassung überschrieben.

Das Beispiel zeigt eine simple Farbanimation.

```
import Scene, Objects
import math, vtkpython
```

```
class ColorTable(Scene.Scene):
    sceneName='ColorTable'
    def setup(self):
        self.cube = Objects.ObjectBase(vtkpython.vtkCubeSource())
        self.cBar = Objects.ColorBar(-1.0, 1.0)
        self.addObject(self.cube)
        self.addObject(self.cBar)
        self.counter = 0.0
    def videoStepForward(self):
        r, g, b = self.cBar.RGB(math.sin(self.counter))
        self.cube._actor.GetProperty().SetColor(r, g, b)
        self.setHeadline('Value is %06.3g' % (self.counter,))
        self.counter = self.counter+0.005
```

**Einfache Oberflächen sind mit
wenigen Zeilen Code machbar!**

System zur allgemeinen Verwaltung von (Simulations-)Aufträgen mit:

- Prioritätswarteschlange
- Netzwerkfähigkeit
- Benutzerverwaltung
- Sicherheitskonzept
- änderbaren Auftragsklassen

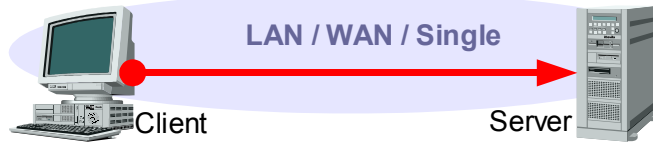




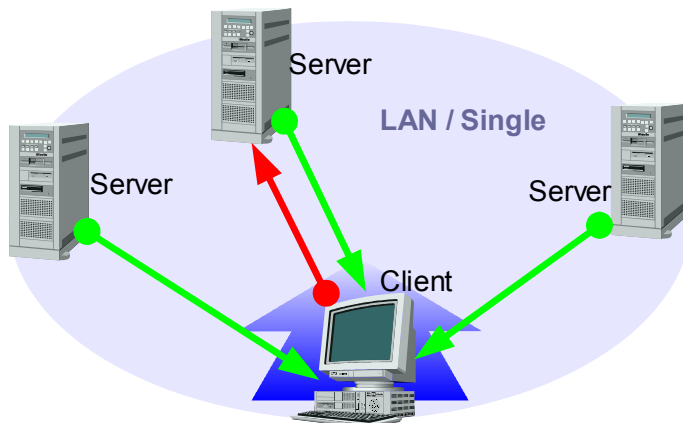
- Bevorzugte Nutzung von OpenSource-Lösungen und offenen Standards
- Kommunikation über XMLRPC - (HTTP(S) & XML)
- 100% pure Python - 95% Plattformunabhängigkeit
- Statusbeobachtung im Browser über HTML möglich



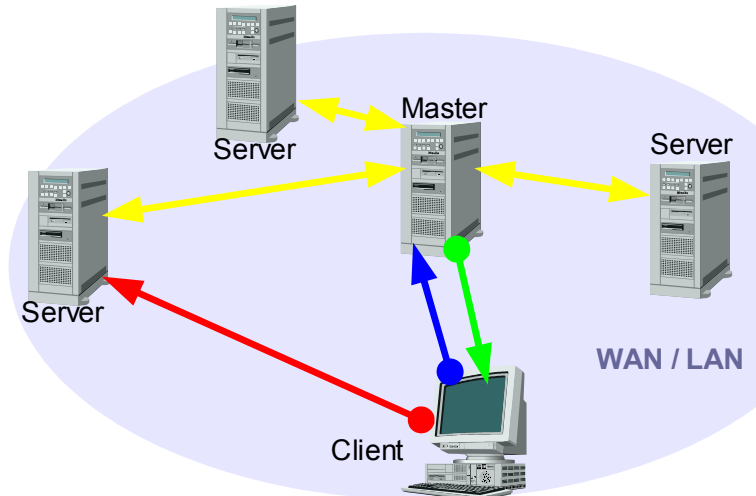
Beispiel: Verbindungsaufbauphase



1. client contacts server directly when knowing its address by sending its request



1. the client sends a broadcast to the whole network asking for a suitable server
2. all matching server answer by sending their address and workload
3. the client selects the best server by evaluating the workload
4. the client contacts the selected server by sending its request



1. the master is monitoring a list of servers, a single server may register and unregister at the master
2. the client asks the master for a suitable server
3. the master suggests the server with the lowest workload
4. the client contacts the selected server by sending its request

netSim – finding and addressing the server

LAN – local area network

WAN – wide area network (internet)

Single – client and server may be the same machine



1. Aufgabe automatisierbar (batch-fähig) gestalten
2. Aufgabenklasse beschreiben:
 - vordefiniert PythonTask, ShellTask und ProcessTask
 - durch Vererbung leicht anpassbar
3. Übertragung von strukturiertem Input und Output:
 - einfache Datenstrukturen durch XMLRPC
 - beliebige Datenstrukturen möglich durch verschiedene Zusatzmodule





alles zusammen – z.B. das Kühllastmodul



- dynamisches Gebäudemodell
- 1,2 oder 3 Zonen vorkonfiguriert
- keine Codegenerierung, keine Kompilation



- Pool von Simulationsservern
- Simulationen sind langlaufend (ProcessTask)
- Parametrisierung der Simulation aus mitgelieferten Datenmodellklassen



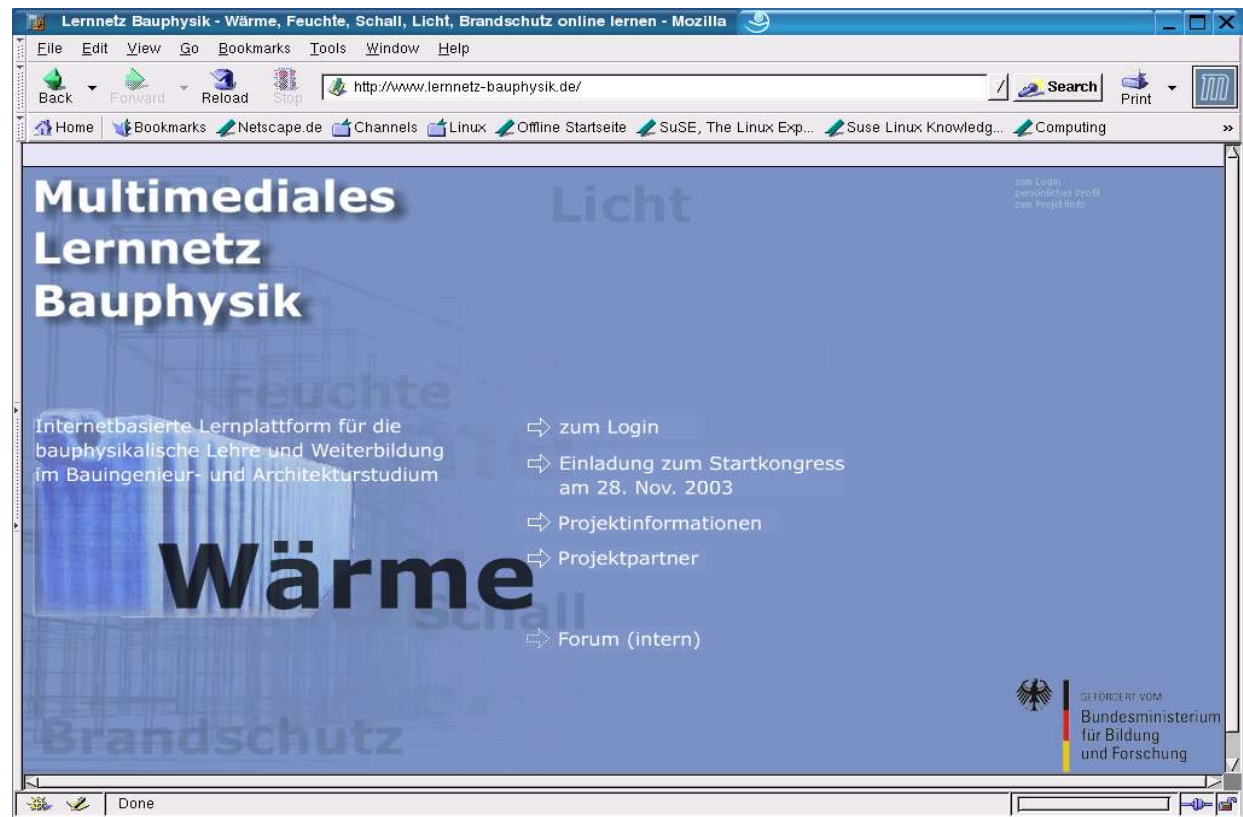
- angepasste Szene bietet Gebäudedarstellung in 3D und die komplette Funktionalität
- Parametereingabe und Ergebnisvisualisierung
- Gebäudeelemente führen zu den Dialogen



E-Learning für die Bauphysik

Partner im Verbundprojekt:

- Universität Karlsruhe
- Universität Stuttgart
- Universität Kassel
- Universität Weimar
- Fachhochschule Biberach

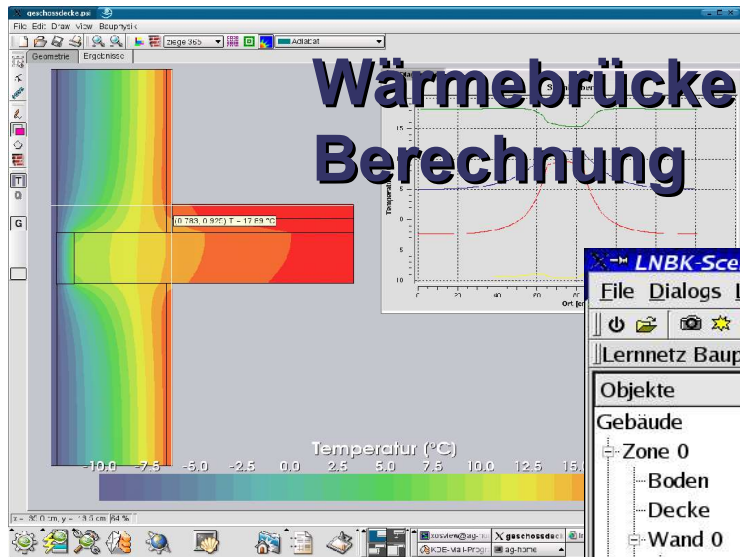




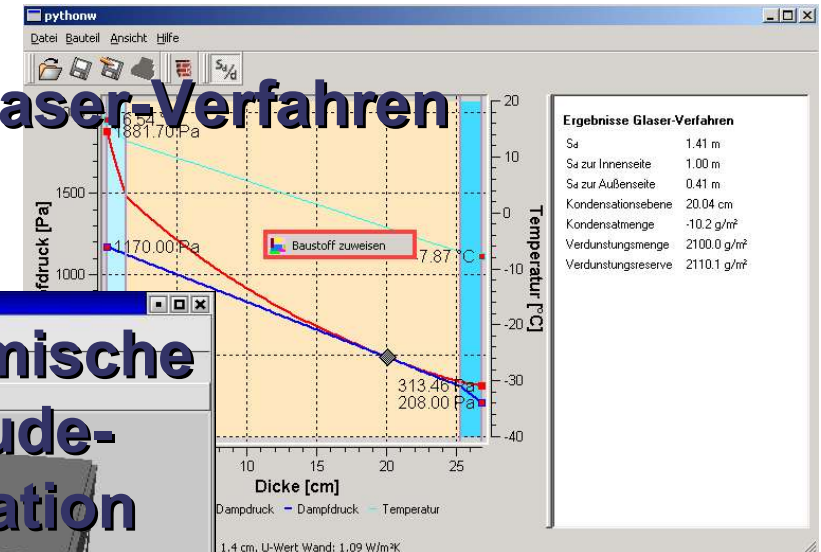
- Applikationen haben gemeinsames Datenmodell
- Datenaustausch über Datenmodell
- XML-basiert
- Datenstrukturen werden aus XML-Schema erzeugt
- Offen (Open Source)
- Erweiterbar
- Spezialisierung

```
</frameType>
</frameData>
<spacerData>
  <spacerType id='pw3'>
    <psi>0.040</psi>
  </spacerType>
</spacerData>
<windowPattern id='pw4' glassTypeId='pw2' frameTypeId='pw1' spacerTypeId='pw3' />
<windowType id='pw5' name='Fenster, 2-flügelig, dreh-kipp, mit Unterlicht' patternId='pw4'>
  <constructionType>Fenster, 1-flügelig, dreh-kipp, mit Oberlicht</constructionType>
  <width>1.5</width>
  <height>1.25</height>
  <frameRatio>0.273493333333</frameRatio>
  <frameSequence>
    <frame index='1' width='0.050' />
    <frame index='2' width='0.050' />
    <frame index='3' width='0.070' />
    <frame index='4' width='0.070' />
    <frame index='5' width='0.090' />
  </frameSequence>
  <frameSplittingHorizontal>0.32</frameSplittingHorizontal>
  <spacerLength>9.48</spacerLength>
```

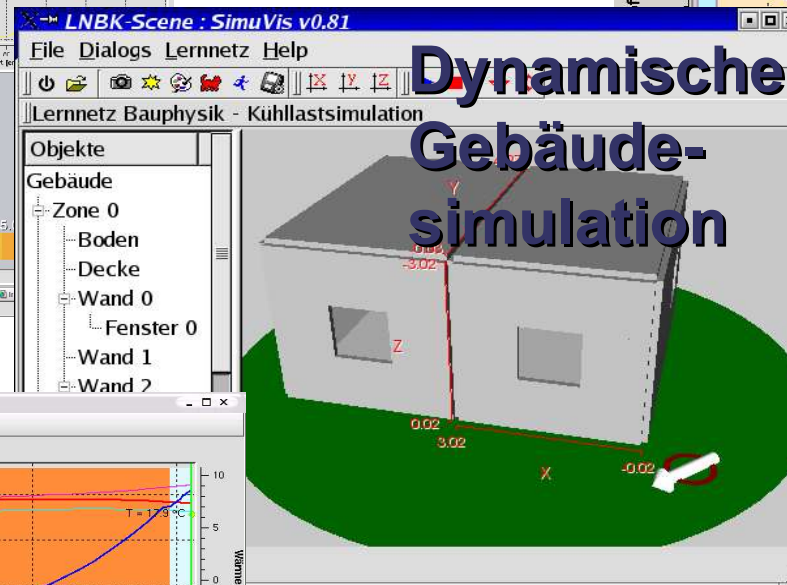
**Wärmebrücken-
Berechnung**



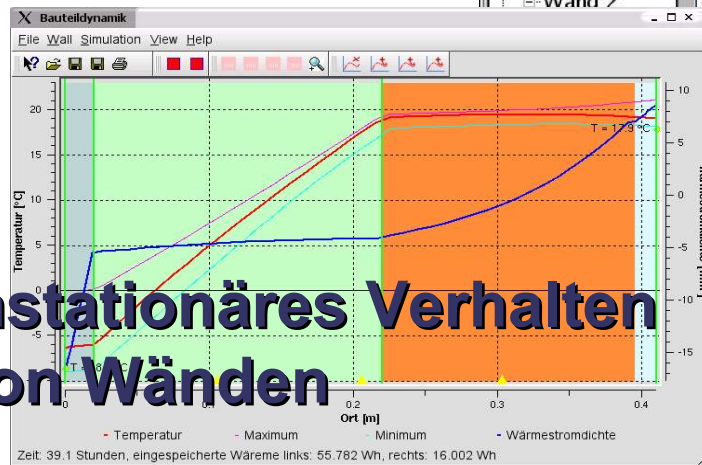
Glaser-Verfahren



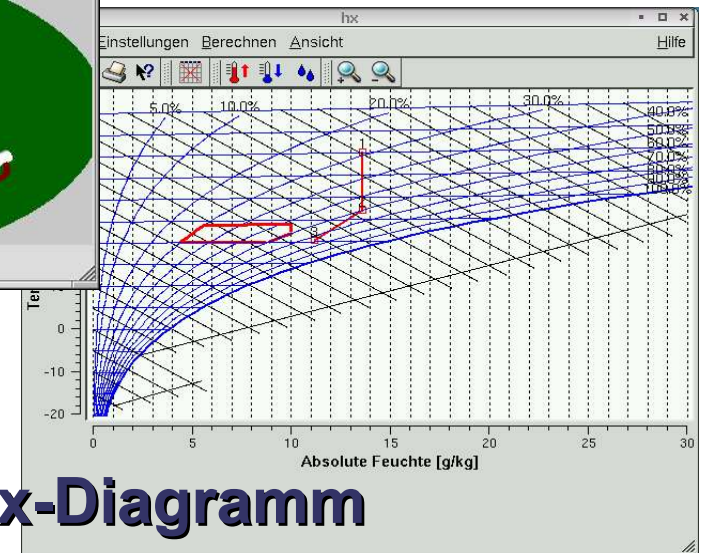
**Dynamische
Gebäude-
simulation**



**Instationäres Verhalten
von Wänden**



hx-Diagramm



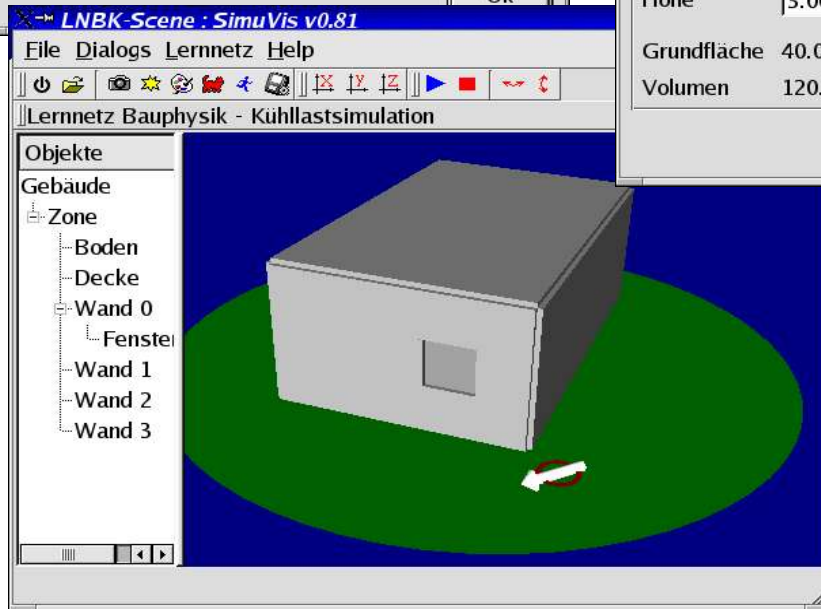
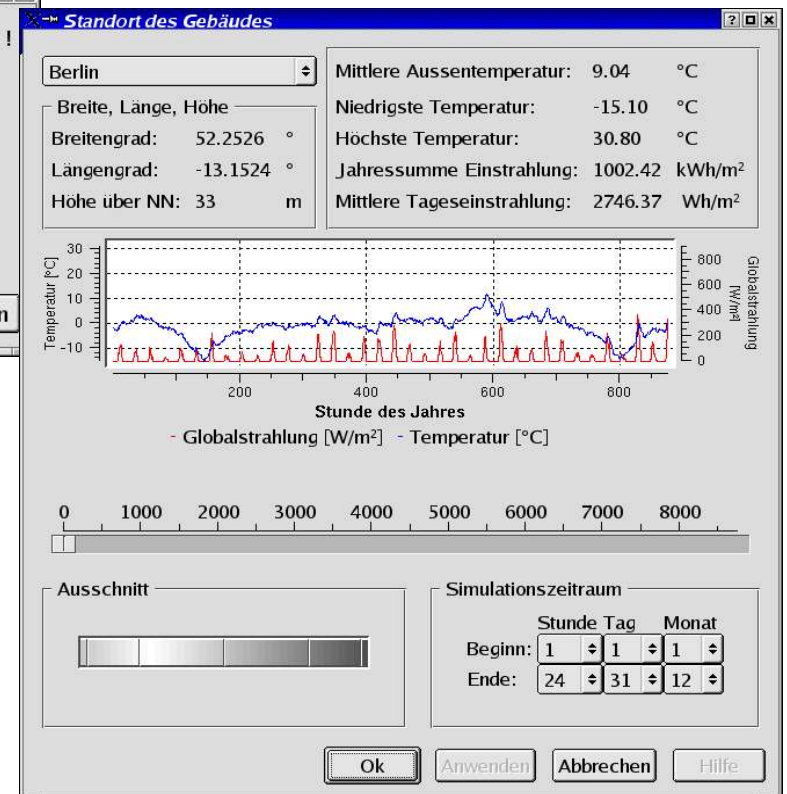
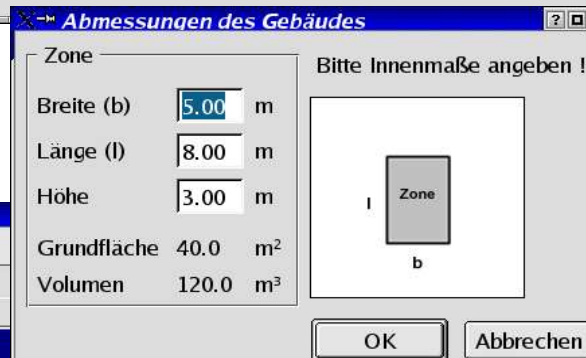
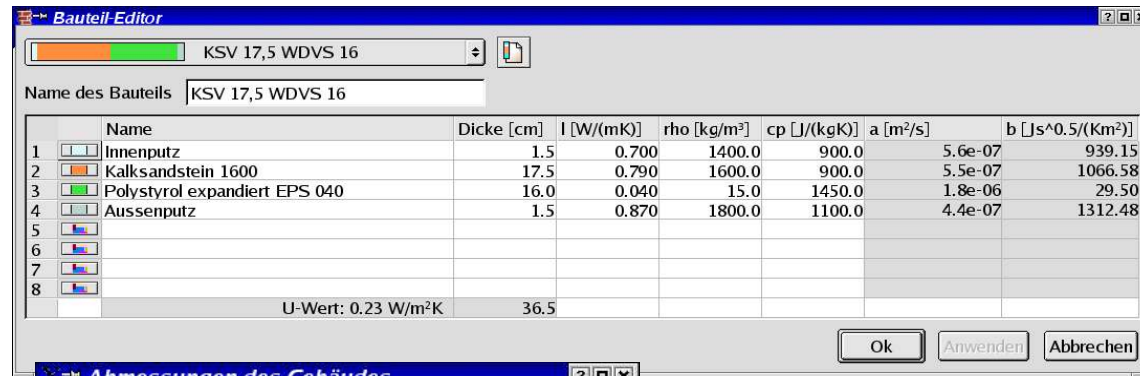
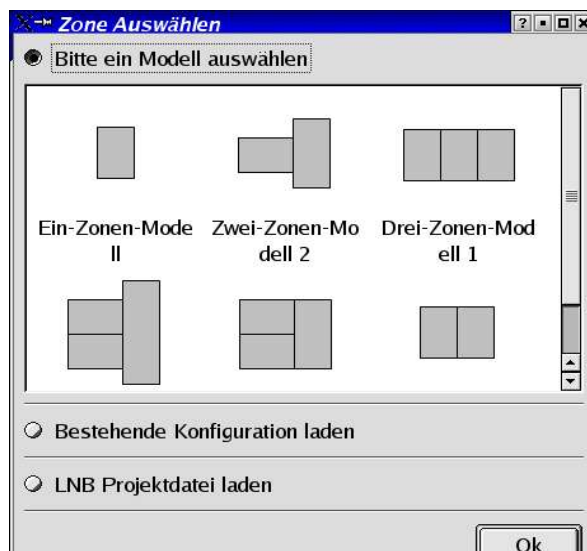


- dezentral/SimuVis/netSim: **www.dezentral.de**
- FH Biberach: **www.fh-biberach.de**
- Lernnetz: **www.lernnetz-bauphysik.de**
- SMILE: **www.smilenet.de**
- Werkzeuge / Bibliotheken
 - www.python.org
 - www.vtk.org
 - www.trolltech.com



Kontakt:
gerber@fh-biberach.de
raedler@dezentral.de







Anhang | Screenshots



The screenshot displays the SimuVis software interface, which is used for simulating building energy performance. The main window is titled "SimuVis" and contains several panels for configuring a simulation.

Main Window Panels:

- keine Fenster geladen:** A dropdown menu showing "keine Fenster geladen" and buttons for "Laden" and "Speichern".
- Bauart:** A dropdown menu showing "Fenster, 1-flügelig, dreh-kipp, mit Unterlicht".
- Fenstergröße:** Input fields for "Breite [m]" (1.00) and "Höhe [m]" (1.25).
- Rahmen:** A section for selecting the frame type and material. It includes a radio button for "Auswahl" and a text field for "Uf [W/m²K]" (2.1). The selected material is "Hartholz (Rahmenstärke = 66 mm), (Uf = 2.1 W/(m²K))".
- Rahmenbreite 1 [m]:** 0.05
- Rahmenbreite 2 [m]:** 0.07
- Rahmenbreite 3 [m]:** 0.07
- Rahmenbreite 4 [m]:** (empty)
- Rahmenbreite 5 [m]:** (empty)
- Rahmenbreite 6 [m]:** (empty)
- Teilung horizontal [m]:** 0.32
- Teilung vertikal [m]:** (empty)
- Längenbezogener Wärmedurchgangskoeffizient:** Input field for "Psi [W/mK]" (1.3).
- Verglasung:** A radio button for "Auswahl" and a dropdown menu showing "2-Scheiben-Is".
- ... oder:** A button for alternative configurations.

Geometrie Panel:

- Fensterfläche:** 1.25 m²
- Rahmenfläche:** 0.33 m² (26%)
- Glasfläche:** 0.92 m²

netSim/TaskWatch Panel:

- Server:** http://localhost:4242/
- Task ID:** lnbk_0
- User:** testuser
- Password:** *****
- Spoiled:** Wed Oct 29 11:09:20 2003
- Started:** Wed Oct 29 11:09:21 2003
- Finished:** ---
- Status:** Running
- Update:** 10 s
- Close:** Button

Start Simulation Panel:

- Simulation:** netSim Settings
- Find netSim/Server by:**
 - ☐ netSim/Master: http://localhost:4244/
 - ☐ Broadcast at Port: 4243
 - ☒ Fixed Address: http://localhost:4242/
- User information:**
 - Username:** testuser
 - Password:** *****
 - Priority:** Default
- Status:** Cancel, Start/Queue

Simulationsergebnisse Panel:

- Variablen:**
 - T_env
 - profile_0.time_transformer.c
 - profile_0.time_transformer.t
 - profile_0.weekend
 - time
 - zone_0.T_air[0]
 - zone_0.T_op
 - zone_0.floor.T_node[0]
 - zone_0.floor.T_node[1]
 - zone_0.floor.T_node[2]
 - zone_0.floor.T_node[3]
- Ergebnisdiagramm:** A line graph showing temperature over time. The x-axis is "Zeit [s]" (1 to 4) and the y-axis is "Temperatur [K]" (270 to 295). The graph shows a red line for "zone_0.T_air[0]" and a green line for "T_env".
- Datei:** /var/tmp/lnbk_res_2078
- 0 - 0**
- Einstellungen, Drucken, Schliessen:** Buttons