Der ProcessViewBrowser unter

Linux/Unix OpenVMS und Windows

2003

http://pvbrowser.org

Lehrig Software Engineering
Dr.-Ing. Rainer Lehrig
lehrig@t-online.de

ProcessViewBrowser

- OpenSource
- Prinzip des Browsers
- Beispiele für Gestaltungsmöglichkeiten
- Design von Masken mit Qt Designer
- Erstellen des Sourcecodes mit ui2pvc
- 3D Graphiken mit OpenGL / VTK
- Online Hilfe
- Multithreaded Server / inetd / ucx service
- Lizenzmodell

OpenSource

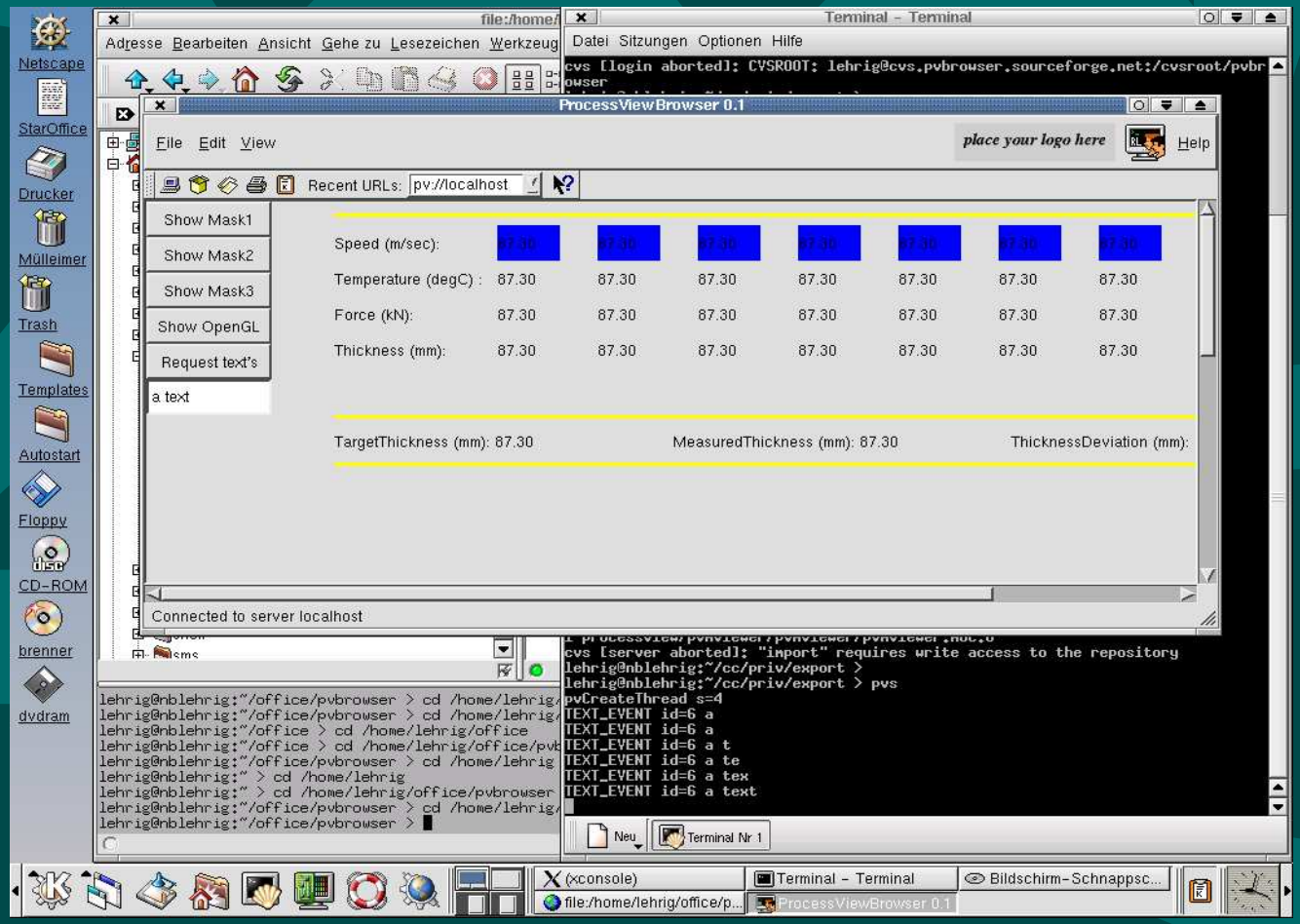
Der ProcessViewBrowser besteht aus OpenSource Komponenten:

- ProcessViewBrowser http://pvbrowser.org
- Qt http://www.trolltech.com
- OpenGL http://www.opengl.org
- Tcl/Tk http://www.tcl.tk
- VTK http://www.kitware.com/vtk/

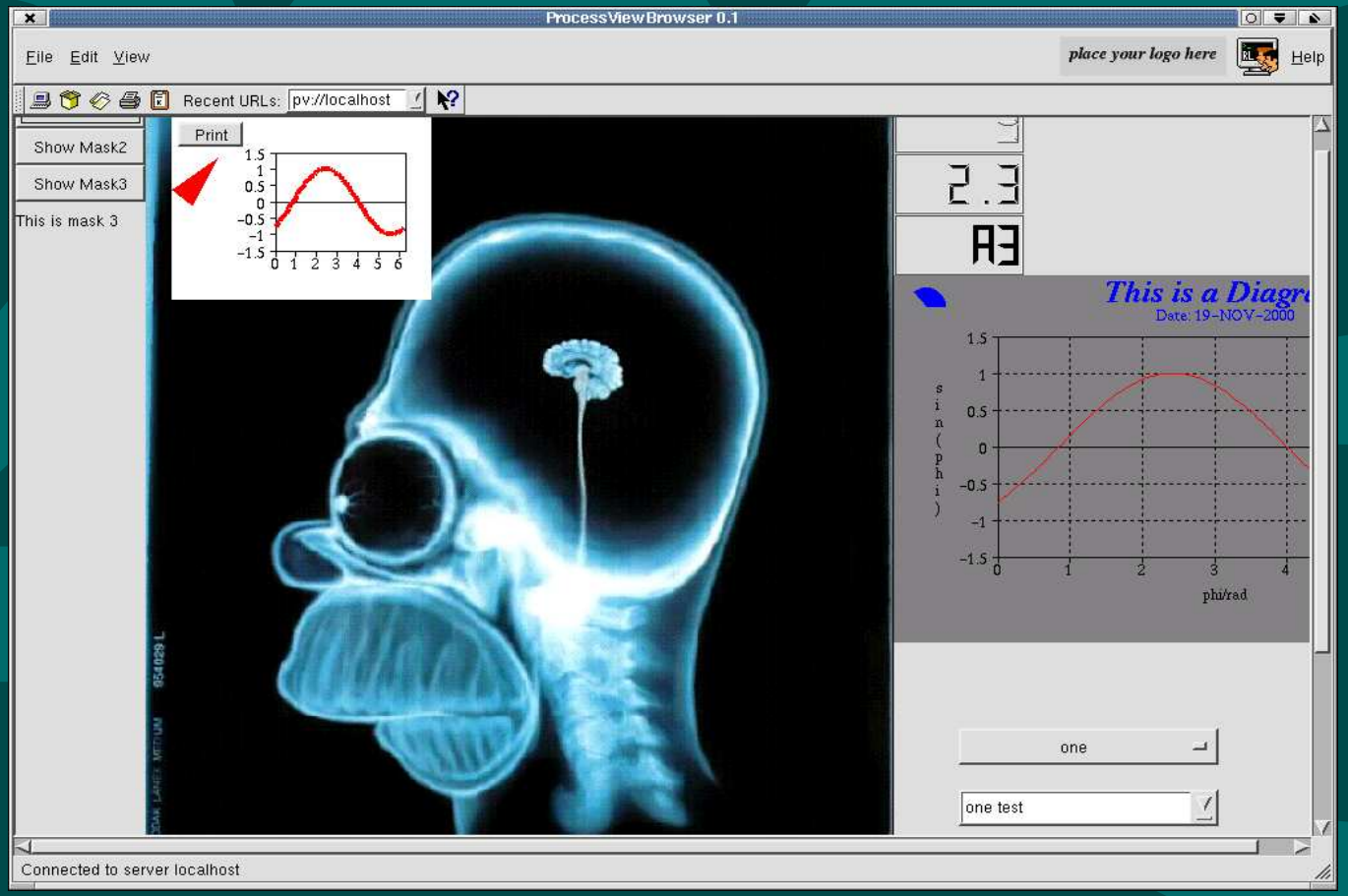
Prinzip des Browsers

- Client/Server - Modell
- Vergleichbar mit einem Internet Browser
- Schwerpunkt Prozessvisualisierung
- Der Browser bleibt unverändert
- Der Server wird vom Anwender entworfen
- Der Server wird graphisch mit dem Qt Designer entworfen
- Der Server schickt Befehle zum Browser
- Der Browser interpretiert diese Befehle
- Der Browser schickt Ereignisse zum Server
- Objektorientiert (Qt Widgets, eigene Widgets)

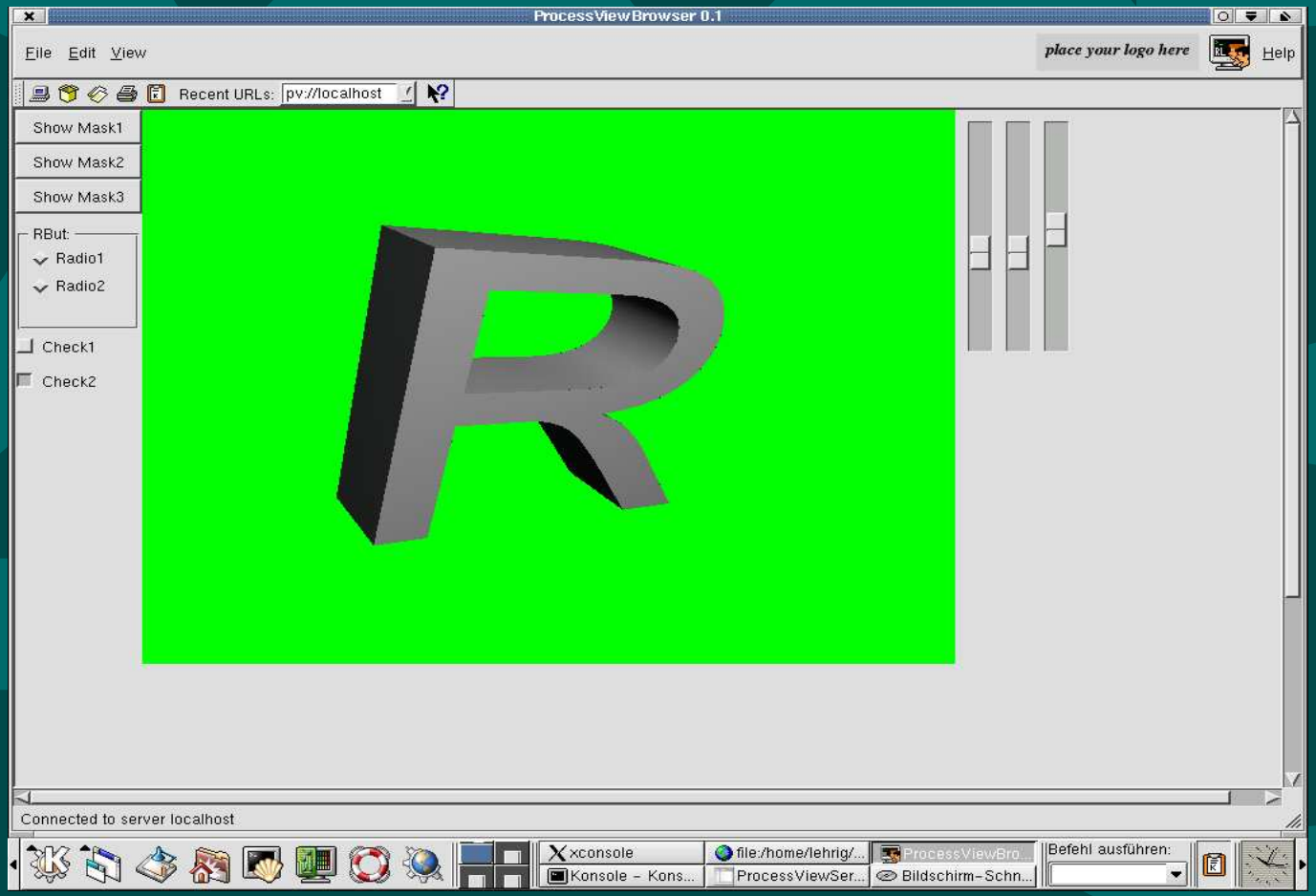
Beispiele für Gestaltungsmöglichkeiten



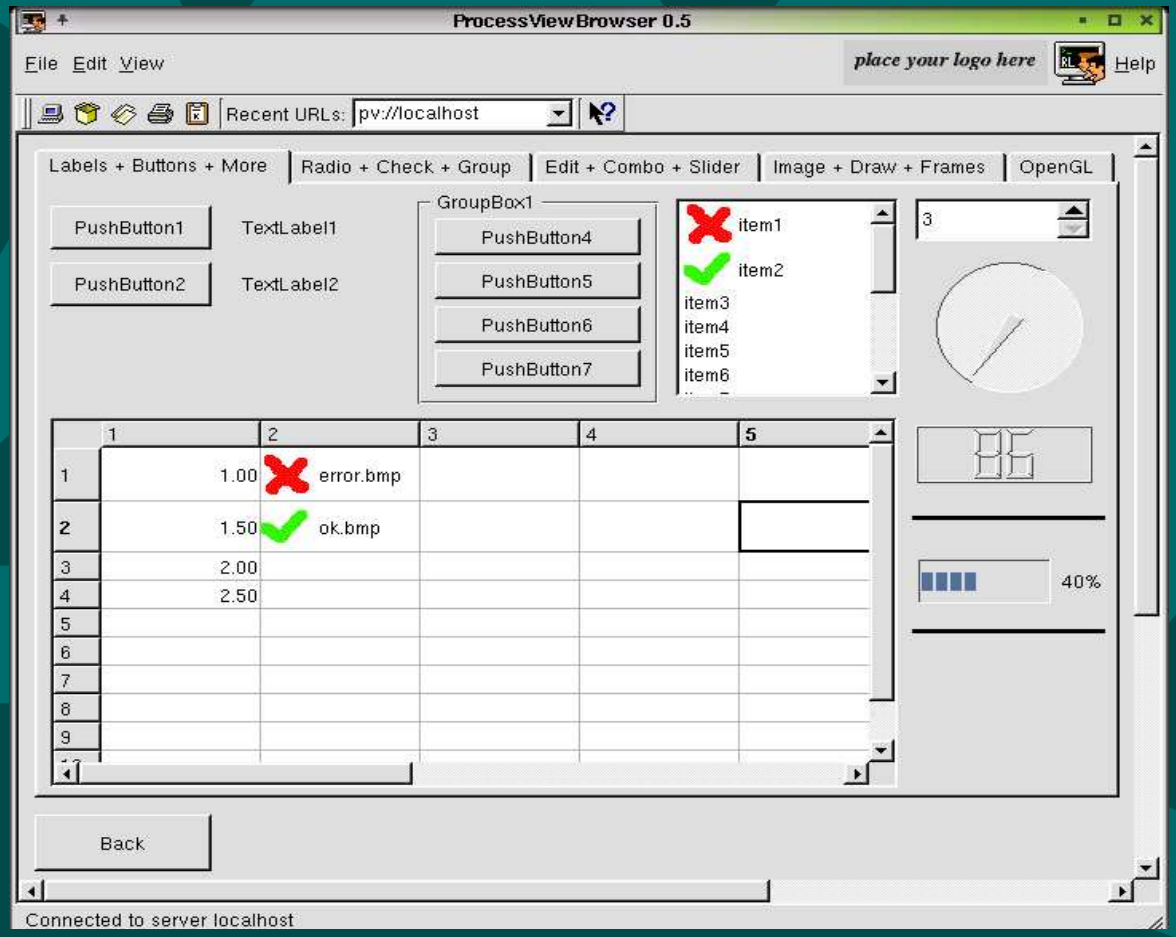
Beispiele für Gestaltungsmöglichkeiten



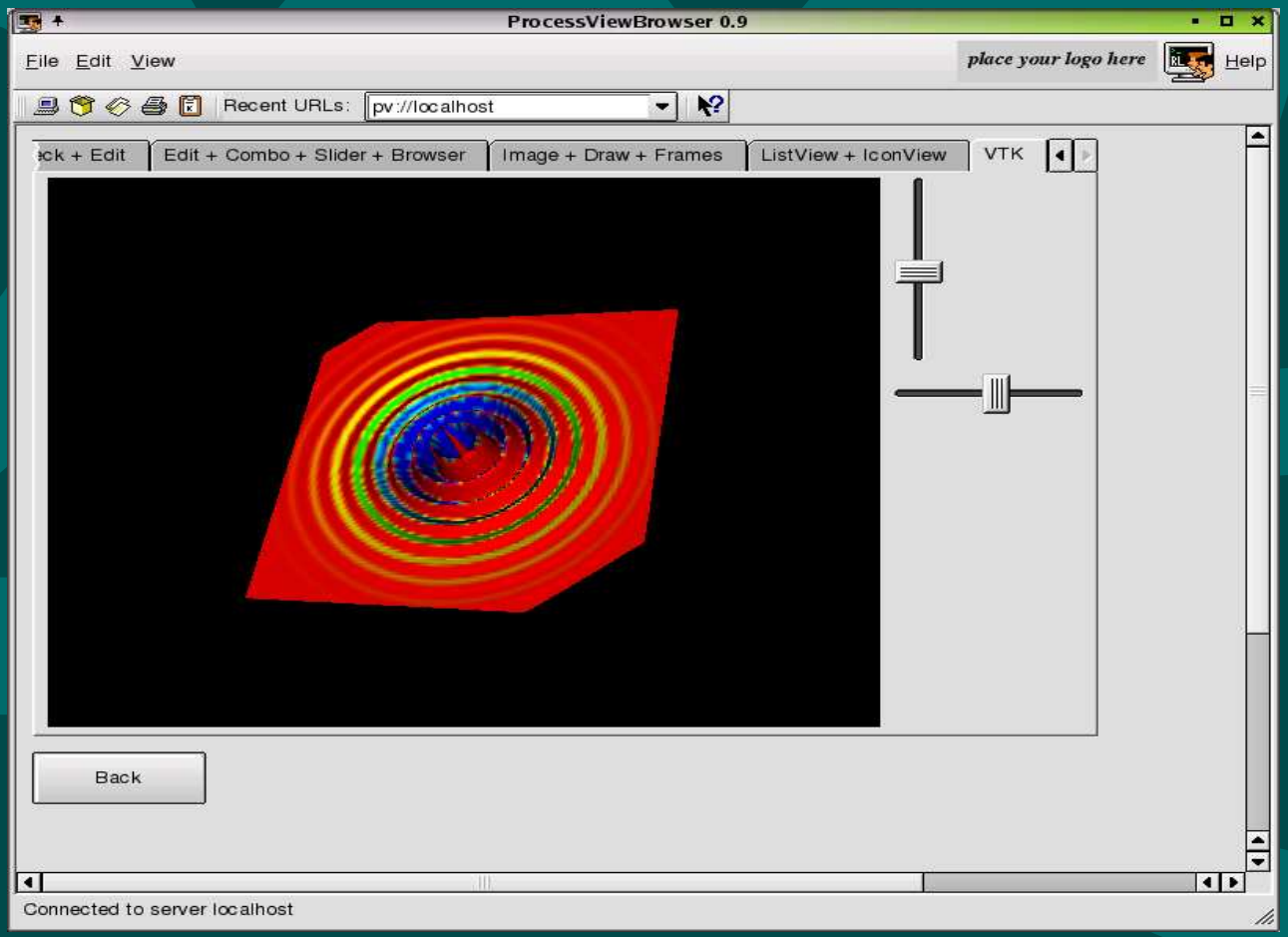
Beispiele für Gestaltungsmöglichkeiten



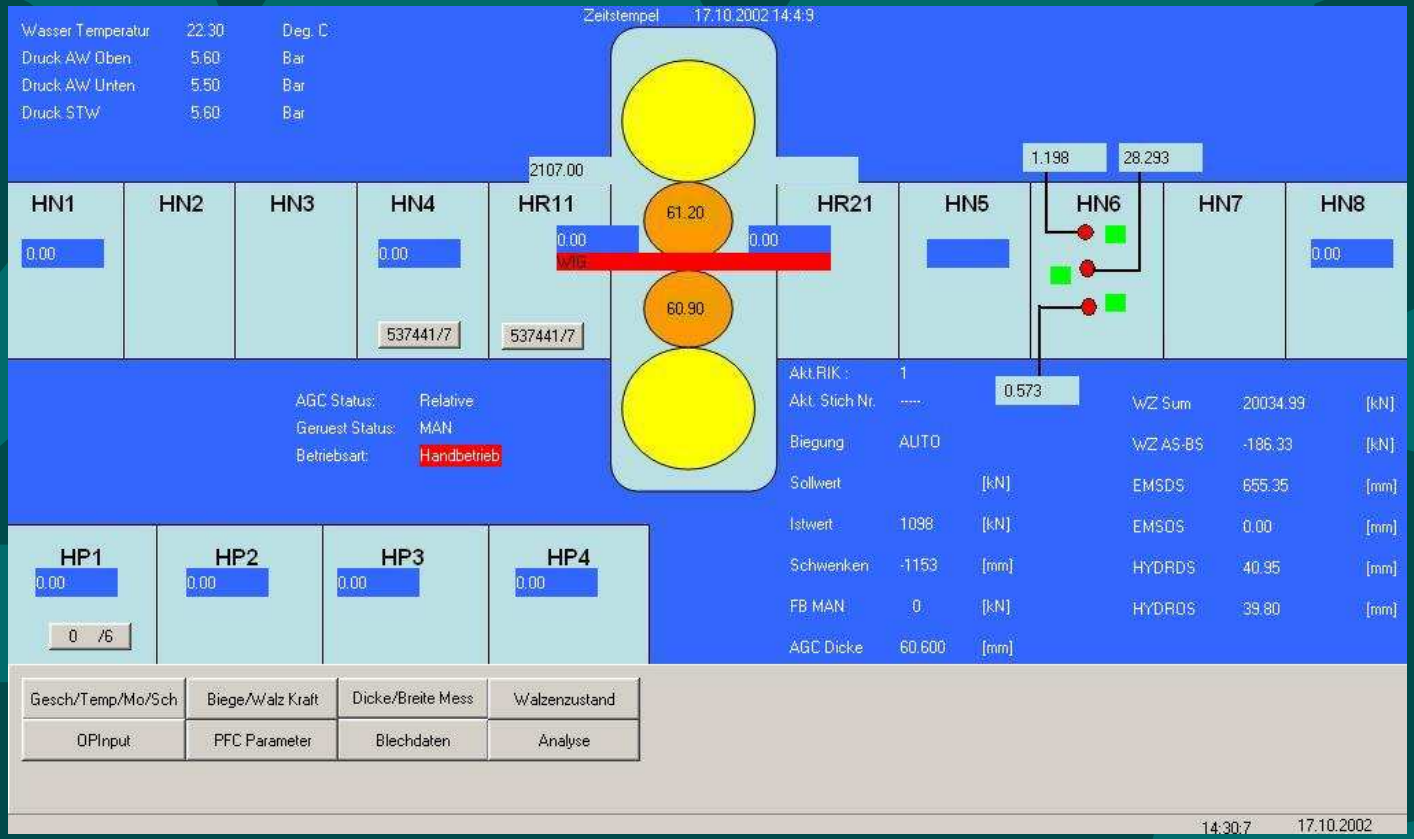
Beispiele für Gestaltungsmöglichkeiten



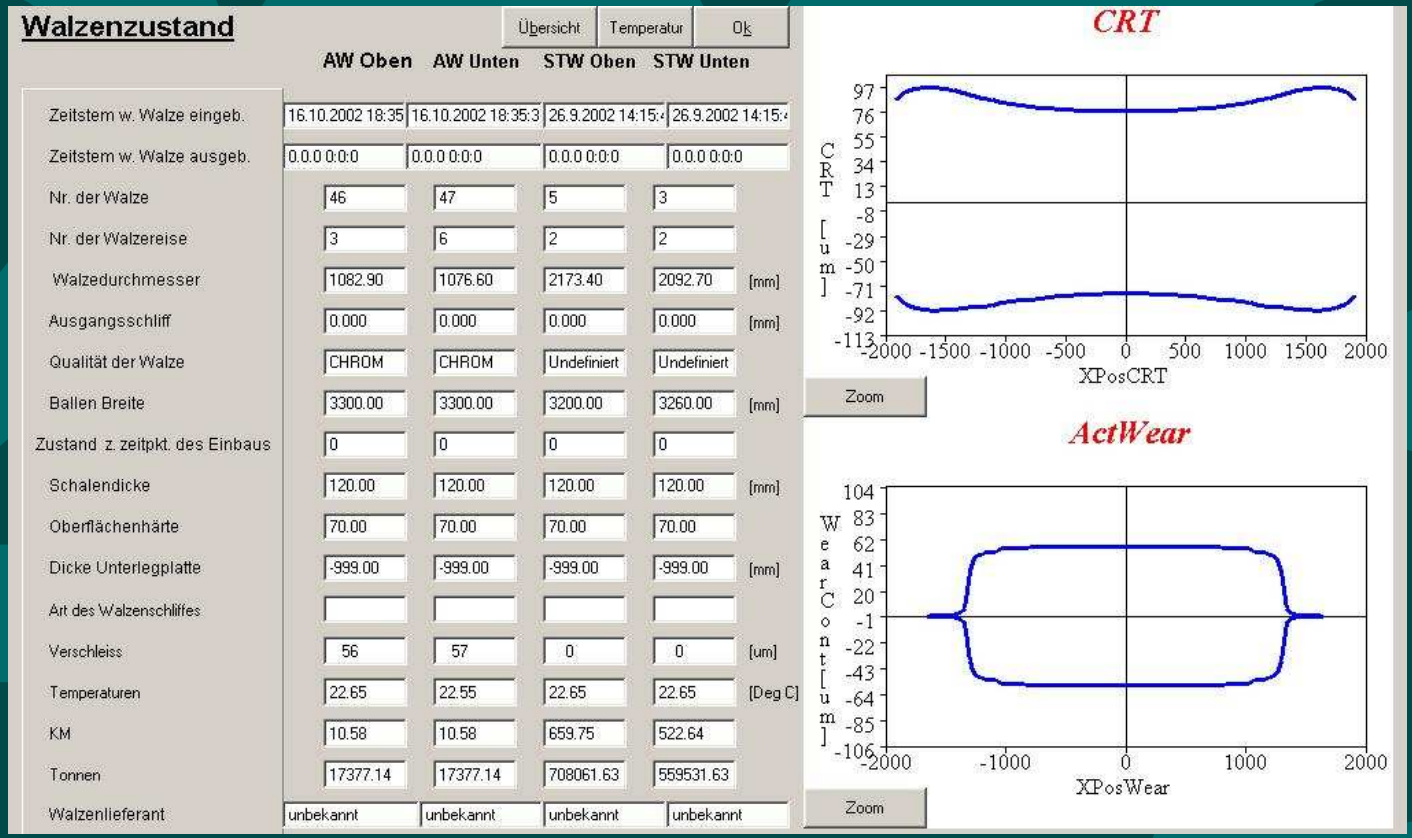
Beispiele für Gestaltungsmöglichkeiten



Beispiele für Gestaltungsmöglichkeiten



Beispiele für Gestaltungsmöglichkeiten



Beispiele für Gestaltungsmöglichkeiten

MASKEN | GRAFIKEN

BLECHDATEN

Blech ID	537441	Berechn. zeit	17.10.2002 13:12:20
RIK	7	Berechn. modus	Setup
Qalität	CMV0W	Online	Online
Walzart	TG mit intensiv Kühlung		

STICHPLAN

BRAMME

Dicke	245.00	[mm]
Breite	2200.00	[mm]
Länge	2300.00	[m]

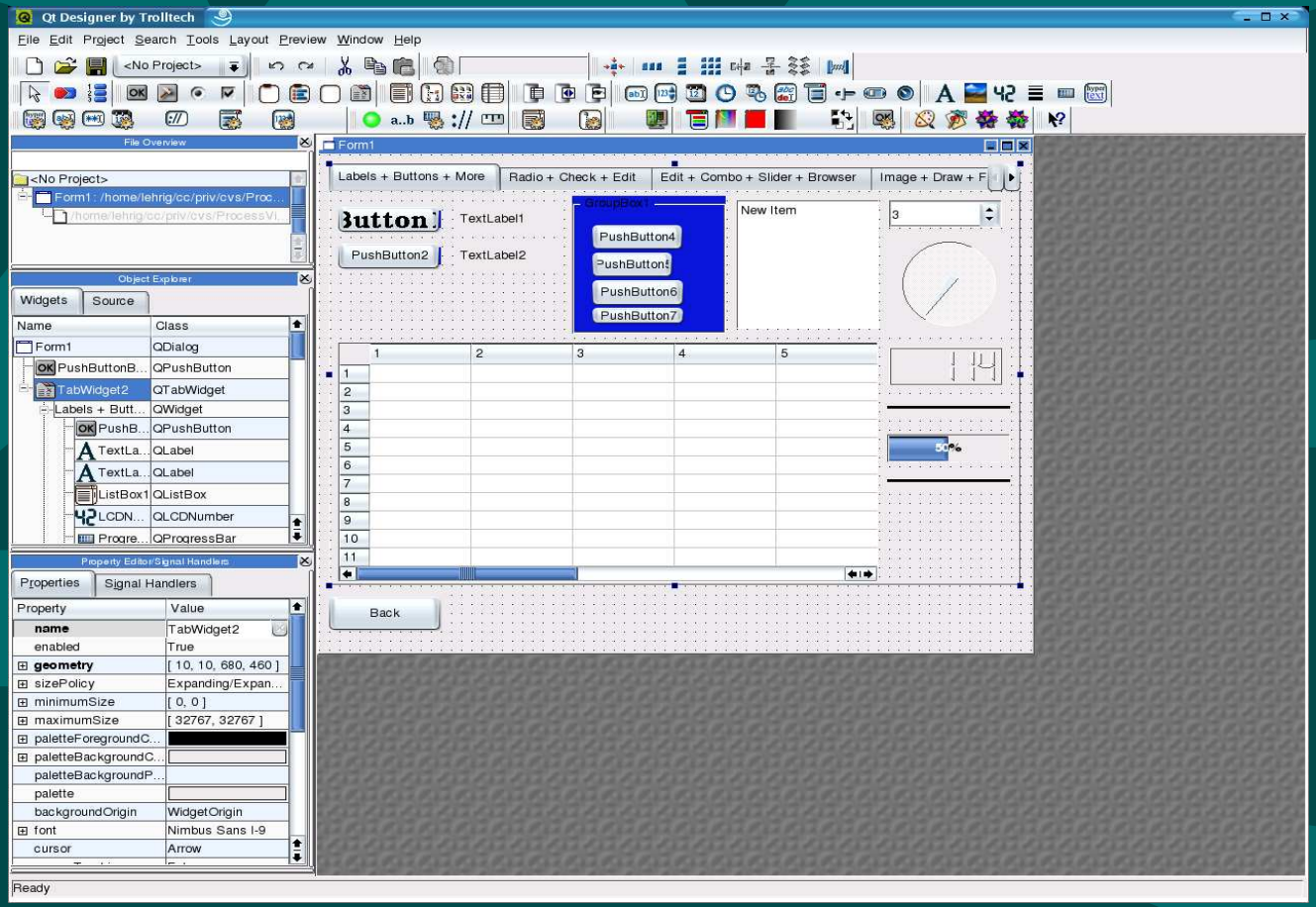
BLECH

Dicke	12.50	[mm]
Breite	2750.00	[mm]
Profile	-99.00	[um]

Akt. Stich Nr 1

	Stichnummer	1	2	3	4	5	6	7	8
1	Ber. Zeit	10.2002 13:12:2(	17.10.2002 13:12:2(	17.10.2002 13:12:2(	17.10.2002 13:12:2(	17.10.2002 13:12:2(	17.10.2002 13:12:2(	17.10.2002 13:12:2(	17.10.2002 13:12:2(
2	Blechbreite [mm]	2238.00	2238.00	2750.00	2750.00	2821.00	2821.00	2821.00	2821.00
3	Blechdicke Auslauf [mm]	226.06	212.12	184.96	168.27	138.29	109.17	83.06	60.26
4	Abnahme [mm]	23.21	13.93	27.16	16.69	29.97	29.12	26.11	22.79
5	Blechlänge [m]	2.58	2.75	2.57	2.82	3.35	4.24	5.57	7.68
6	erwartetes Walzmoment [kNm]	2137.85	1363.12	3216.58	1951.52	3952.08	4216.86	4216.86	4216.86
7	erwartete Walzkraft[kN]	16151.55	13327.24	22594.52	17514.68	26654.47	29047.30	30998.82	33685.84
8	Pos. mech. Anstellung [mm]	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
9	Pos. Hyd. Anstellung Anstich [mm]	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
10	Pos. Hyd. Anstellung [mm]	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
11	Geschw. Kopf [m/s]	0.65	0.65	0.65	0.65	0.65	0.65	0.65	0.65
12	Geschw. Filet [m/s]	0.65	0.65	0.65	0.65	0.65	0.97	0.97	0.97
13	Geschw. Ende [m/s]	0.65	0.65	0.65	0.65	0.65	0.65	0.65	0.65
14	Leerzeit [s]	4.00	4.00	15.00	4.00	15.00	4.00	4.00	4.00
15	Walzzeit [s]	1.00	1.00	1.00	1.00	1.00	1.00	1.00	2.00
16	Verlängerung	1.11	1.11	1.11	1.11	1.11	1.11	1.11	1.11
17	WZ Phase	ecken	strecken	Breitung	Breitung	Zwischenphase	Zwischenphase	Zwischenphase	Zwischenphase
18	Walzrichtung	hlbett	Ofen	Kühlbett	Ofen	Kühlbett	Ofen	Kühlbett	Ofen
19	Drehen	in	Ja	Nein	Ja	Nein	Nein	Nein	Nein
20	Abspritzen	in	Nein	Ja	Nein	Ja	Nein	Nein	Nein
21	AGC Modus	lativ	Relativ	Relativ	Relativ	Relativ	Relativ	Relativ	Relativ
22	Pendelort	1	HN1	HN1	HN1	HN1	HN1	HN1	HN1

Design von Masken mit Qt Designer



Erstellen des Sourcecodes mit ui2pvc

```
////////////////////////////////////////////////////////////////////////
//
// show_form1 for ProcessViewServer created: Fri Nov 22 13:55:15 2002
//
////////////////////////////////////////////////////////////////////////
#include "pvapp.h"

typedef struct // (todo: define your data structure here)
{
}
DATA;

// _begin_of_generated_area_ (do not edit -> use ui2pvc) -------------------

// our mask contains the following objects
enum {
  ID_MAIN_WIDGET = 0,
  PushButton1,
  LineEdit1,
  ID_END_OF_WIDGETS
};

static int generated_defineMask(PARAM *p)
{
  int w,h,depth;

  w = h = depth = 0;
  if(w==h) depth=0; // fool the compiler
  pvStartDefinition(p,ID_END_OF_WIDGETS);


  pvQPushButton(p,PushButton1,0);
  pvSetGeometry(p,PushButton1,20,20,90,41);
  pvSetText(p,PushButton1,"PushButton1");

  pvQLineEdit(p,LineEdit1,0);
  pvSetGeometry(p,LineEdit1,120,31,91,21);
  pvSetText(p,LineEdit1,"");

  pvEndDefinition(p);
  return 0;
}

// _end_of_generated_area_ (do not edit -> use ui2pvc) ----------------------

static int defineMask(PARAM *p)
{
  generated_defineMask(p);
  // (todo: add your code here)
  return 0;
}

static int showData(PARAM *p, DATA *d)
{
  // (todo: add your code here)
  return 0;
}
```

```
int show_form1(PARAM *p)
{
 DATA d;
 char event[MAX_EVENT_LENGTH];
 char text[MAX_EVENT_LENGTH];
 char str1[MAX_EVENT_LENGTH];
 int  i,w,h,val,x,y,button;

 defineMask(p);
 memset(&d,0,sizeof(DATA));
 readData(&d); // from shared memory, database or something else
 showData(p,&d);
 while(1)
 {
   pvPollEvent(p,event);
   switch(pvParseEvent(event, &i, text))
   {
     case NULL_EVENT:
       readData(&d); // from shared memory, database or something else
       showData(p,&d);
       break;
     case BUTTON_EVENT:
       printf("BUTTON_EVENT id=%d\n",i);
       break;
     case TEXT_EVENT:
       printf("TEXT_EVENT id=%d %s\n",i,text);
       break;
     case SLIDER_EVENT:
       sscanf(text,"(%d)",&val);
       printf("SLIDER_EVENT val=%d\n",val);
       break;
     case CHECKBOX_EVENT:
       printf("CHECKBOX_EVENT id=%d %s\n",i,text);
       break;
     case RADIOBUTTON_EVENT:
       printf("RADIOBUTTON_EVENT id=%d %s\n",i,text);
       break;
     case GL_INITIALIZE_EVENT:
       printf("you have to call initializeGL()\n");
       break;
     case GL_PAINT_EVENT:
       printf("you have to call paintGL()\n");
       break;
     case GL_RESIZE_EVENT:
       sscanf(text,"(%d,%d)",&w,&h);
       printf("you have to call resizeGL(w,h)\n");
       break;
     case GL_IDLE_EVENT:
       break;
     case TAB_EVENT:
       sscanf(text,"(%d)",&val);
       printf("TAB_EVENT(%d,page=%d)\n",i,val);
       break;
     case TABLE_TEXT_EVENT:
       sscanf(text,"(%d,%d,",&x,&y);
       pvGetText(text,str1);
       printf("TABLE_TEXT_EVENT(%d,%d,\"%s\")\n",x,y,str1);
       break;
```

3D Graphiken mit OpenGL / VTK

- OpenGL Subroutinen in jedem Server verfügbar
- OpenGL Befehle werden vom Browser interpretiert
- 3D Graphik ist mit einem Browser unter OpenVMS nicht verfügbar
- Server unter OpenVMS ist aber voll einsetzbar
- VTK wird mit Tcl/Tk angesteuert
- Der Server sendet dem Browser Tcl/Tk Befehle bzw. Scripte zur Nutzung von VTK

Online Hilfe

Multithreaded Server / inetd / ucx service

- Auf allen Betriebssystemen kann der Server als Multithreaded Server laufen
- Ein Thread pro Client
- Unter Linux/Unix kann der Server auch über inetd betrieben werden
- Unter OpenVMS kann der Server auch über
  $ ucx set service
  bekannt gemacht werden
- Windows besitzt leider keinen eingebauten inetd

Lizenzmodell

This program is free software; you can redistribute it and/or modify
it under the terms of the GNU General Public License as published by
the Free Software Foundation; either version 2 of the License, or
(at your option) any later version.

QT, VTK, Tcl/Tk, OpenGL are licensed elsewhere.

There is one exception to the above statement:

If you don't want your ProcessViewServer become GPL software.
E.g. you don't want to publish your ProcessViewServer as GPL Software
with all your code. It is possible to order a commercial license.
In this case your ProcessViewServer may stay closed source.
In order to get a license contact lehrig@t-online.de
Commercial licenses are only available from the original autor of the program(s).