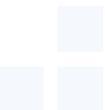
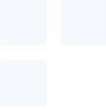


Gruppe 6 SW2: Konfigurations-System

Christian Deutschmann, cappiman@cs.tu-berlin.de
Kay Massow, mallcom@cs.tu-berlin.de



Aufgabenstellung

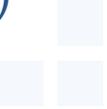


Entwicklung eines Konfigurations-Systems für die Entwicklungsumgebung.

Ziel:

Der Programmierer / Anwender soll die Möglichkeit bekommen Parameter, die zur Verwendung der Orthese von entscheidender Bedeutung sind und sich häufig ändern, extern in einem Dateisystem zu speichern.

Grund:

- Orthesen unterscheiden sich von den Abmessungen
 - Sensor-Daten unterscheiden sich von Nutzer zu Nutzer
 - Daten sind abhängig vom Betriebsmodus (Simulation, Stromversorgung)
 - ...
- 

Aufgabenstellung



- Objektorientiertes Programmieren mit C++
- Bereitstellung einer Benutzerschnittstelle
 - für den Programmierer (Funktionen)
 - für den Anwender (QT-Widget)
- Funktionen sollen Rückgabewerte besitzen
- Verwaltung der Daten in einem zentralen Dateisystem

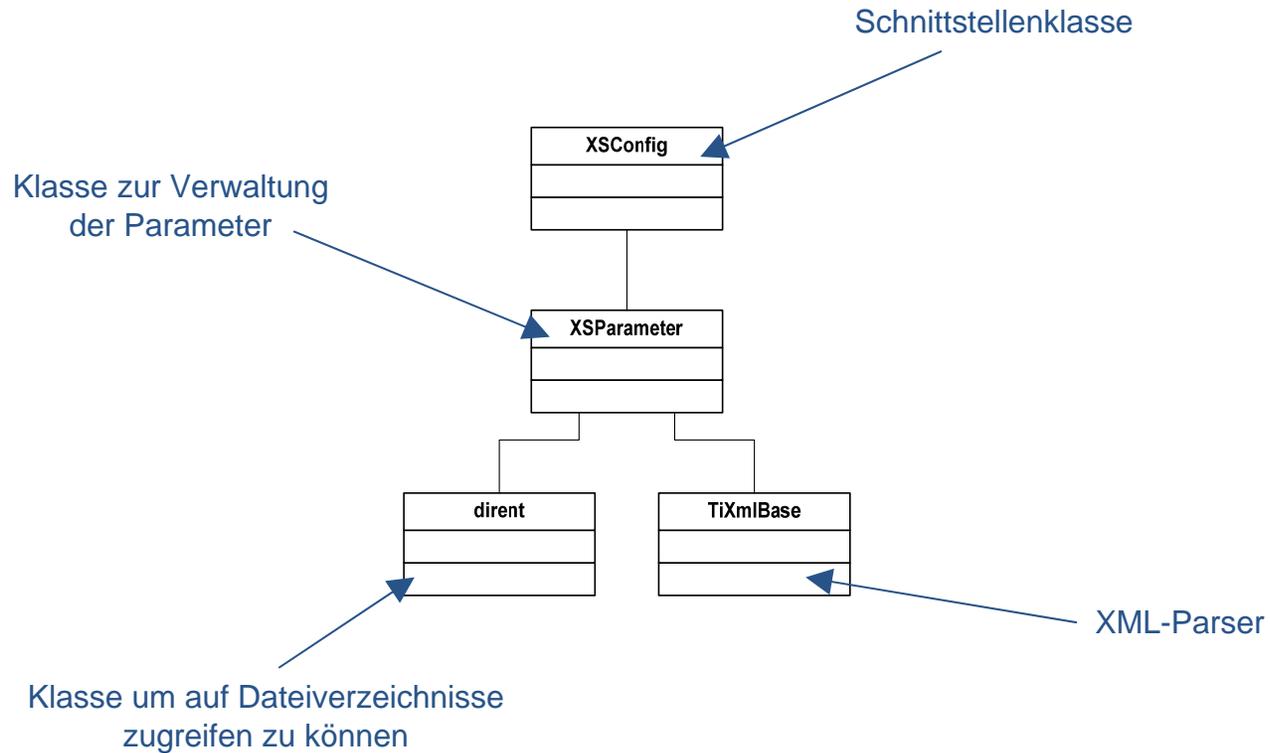


Aufgabenstellung

- **Objektorientiertes Programmieren mit C++**
- Bereitstellung einer Benutzerschnittstelle
 - für den Programmierer (Funktionen)
 - für den Anwender (QT-Widget)
- Funktionen sollen Rückgabewerte besitzen
- Verwaltung der Daten in einem zentralen Dateisystem

Objektorientiertes Programmieren mit C++

Klassendiagramm



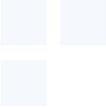
TinyXml Class List

Here are the classes, structs, unions and interfaces with brief descriptions:

TiXmlAttribute	<i>An attribute is a name-value pair</i>
TiXmlBase	<i>TiXmlBase is a base class for every class in TinyXml</i>
TiXmlComment	<i>An XML comment</i>
TiXmlDeclaration	<i>In correct XML the declaration is the first entry in the file</i>
TiXmlDocument	<i>Always the top level node</i>
TiXmlElement	<i>The element is a container class</i>
TiXmlHandle	<i>A TiXmlHandle is a class that wraps a node pointer with null checks; this is an incredibly useful thing</i>
TiXmlNode	<i>The parent class for everything in the Document Object Model</i>
TiXmlText	<i>XML text</i>
TiXmlUnknown	<i>Any tag that tinyXml doesn't recognize is saved as an unknown</i>

Generated on Wed Oct 19 07:51:53 2005 for TinyXml by  1.4.4

Aufgabenstellung

- 
- Objektorientiertes Programmieren mit C++
 - **Bereitstellung einer Benutzerschnittstelle**
 - **für den Programmierer (Funktionen)**
 - **für den Anwender (QT-Widget)**
 - Funktionen sollen Rückgabewerte besitzen
 - Verwaltung der Daten in einem zentralen Dateisystem

Bereitstellung einer Benutzerschnittstelle

für den Programmierer (Funktionen)

```
/* gathers parameters from all configurationfiles in this subdirectory */
void    loadConfig(char* p_Path);
/* getter functions for Parameters overloaded by default value in case of not existing parameter */
int     getPmInt    (char* p_Parameter);
int     getPmInt    (char* p_Parameter, int p_Default);
float   getPmFloat  (char* p_Parameter);
float   getPmFloat  (char* p_Parameter, int p_Default);
bool    getPmBool   (char* p_Parameter);
bool    getPmBool   (char* p_Parameter, int p_Default);
char    getPmChar   (char* p_Parameter);
char    getPmChar   (char* p_Parameter, int p_Default);
char*   getPmString (char* p_Parameter);
char*   getPmString (char* p_Parameter, int p_Default);
/* gets userflags */
bool    getUserFlag (char* p_Parameter);
bool    getUserFlag (char* p_Parameter, int p_Default);
/* sets userflags */
void    setUserFlag (char* p_Parameter, bool p_Flag, char* p_Path);
void    setUserFlag (char* p_Parameter, bool p_Flag, char* p_Path, char* p_File);
/* setter functions */
void    setPmInt    (char* p_Parameter, int p_Value, char* p_Path);
void    setPmInt    (char* p_Parameter, int p_Value, char* p_Path, char* p_File);
void    setPmFloat  (char* p_Parameter, float p_Value, char* p_Path);
void    setPmFloat  (char* p_Parameter, float p_Value, char* p_Path, char* p_File);
void    setPmBool   (char* p_Parameter, bool p_Value, char* p_Path);
void    setPmBool   (char* p_Parameter, bool p_Value, char* p_Path, char* p_File);
void    setPmChar   (char* p_Parameter, char p_Value, char* p_Path);
void    setPmChar   (char* p_Parameter, char p_Value, char* p_Path, char* p_File);
void    setPmString (char* p_Parameter, char* p_Value, char* p_Path);
void    setPmString (char* p_Parameter, char* p_Value, char* p_Path, char* p_File);
/* prints all parameter to screen*/
void    screen();
```

Bereitstellung einer Benutzerschnittstelle

Anwendungsbeispiel

```
#include <xconfig.h>

int main (int argc, char** argv) {

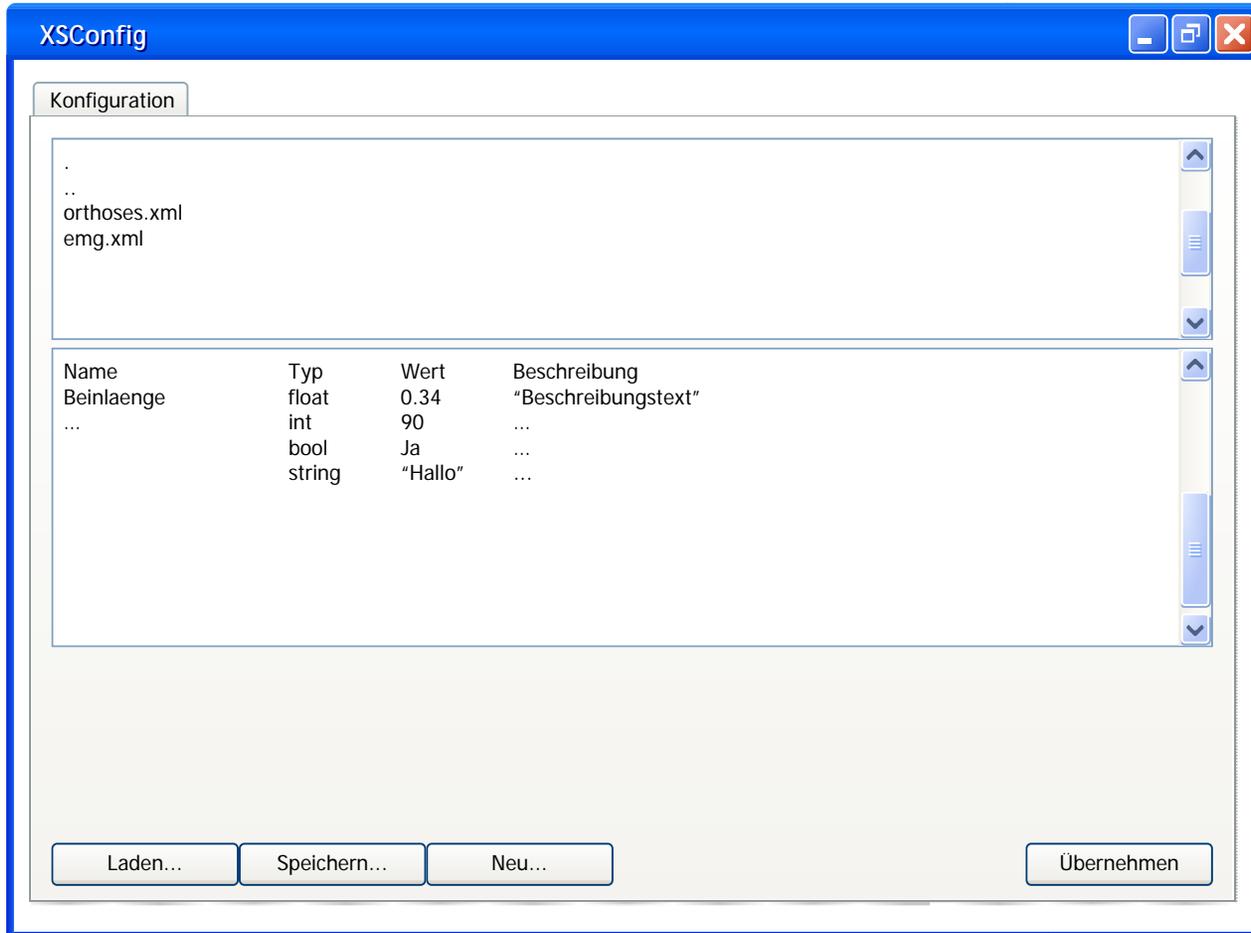
    XConfig A,B;
    A.loadConfig("orthese/A");
    B.loadConfig("patient/Meier/rehaphase2.xml");

    float Beinl;
    string farbe;

    beinl = B.getPmFloat("Beinlaenge", 88.8);
    farbe = A.getPmString("Farbe");

};
```

Bereitstellung einer Benutzerschnittstelle für den Anwender (QT-Widget)

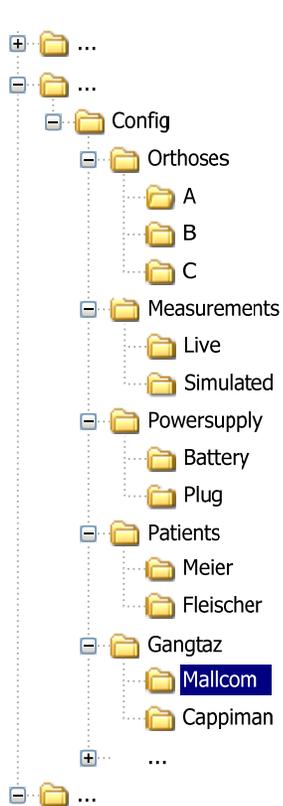


Aufgabenstellung

- 
- Objektorientiertes Programmieren mit C++
 - Bereitstellung einer Benutzerschnittstelle
 - für den Programmierer (Funktionen)
 - für den Anwender (QT-Widget)
 - Funktionen sollen Rückgabewerte besitzen
 - **Verwaltung der Daten in einem zentralen Dateisystem**



Verwaltung der Daten in einem zentralen Dateisystem



Verwaltung der Daten in einem zentralen Dateisystem

Beispiel XML-Datei

```
<!-- Speichert die Daten der Orthese -->
<?xml version = "1.0">
<parameter name="Beinlaenge" typ="float" wert="0.83" beschreibung="..." flag="1"/>
<parameter name="Winkel" typ="int" wert="90" beschreibung="..." flag="1"/>
<parameter name="Unterstützung" typ="bool" wert="1" beschreibung="..." flag="1"/>
<parameter name="Text" typ="char" wert="Hast`e nicht gesehen!" beschreibung="..." flag="1"/>
```