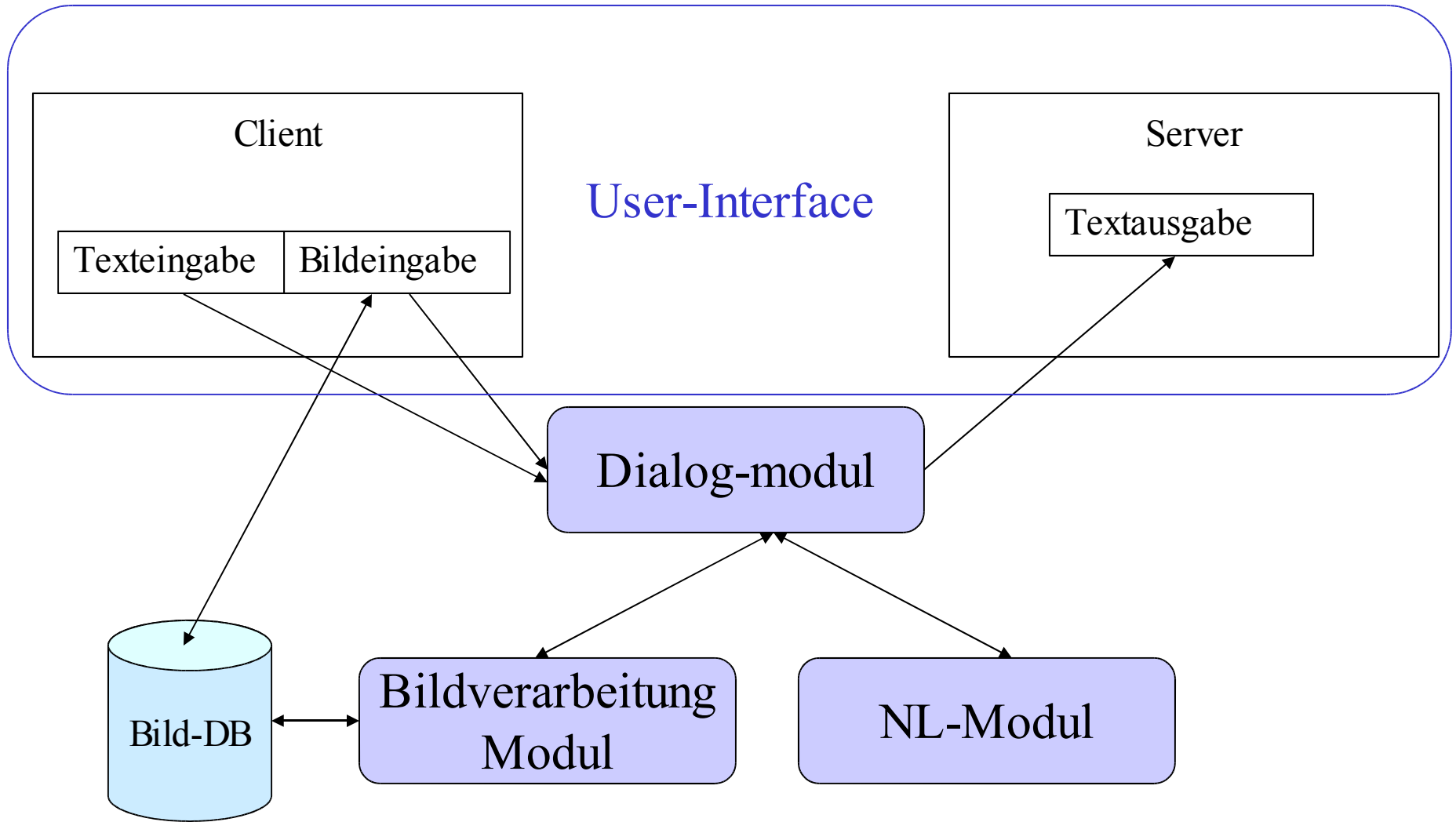


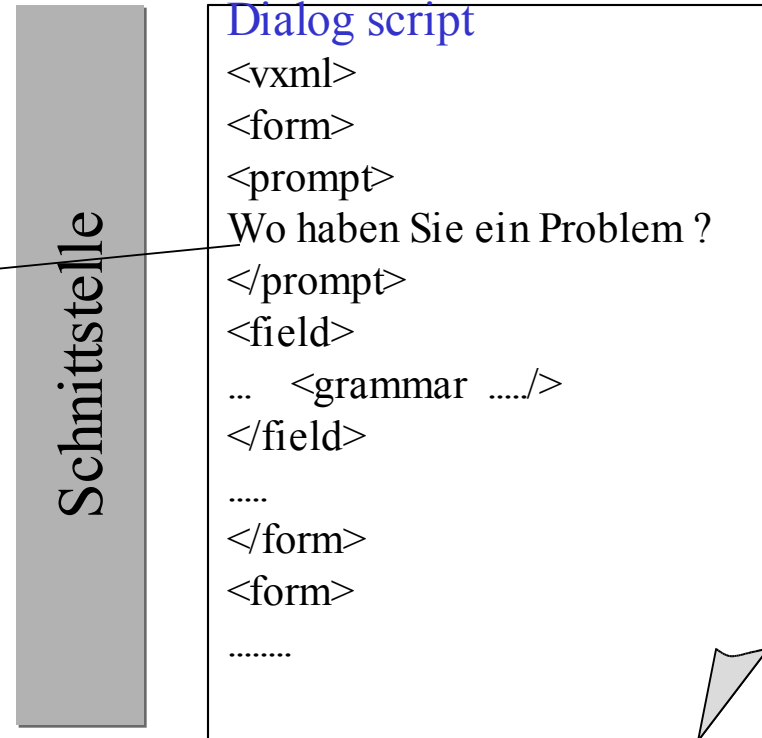
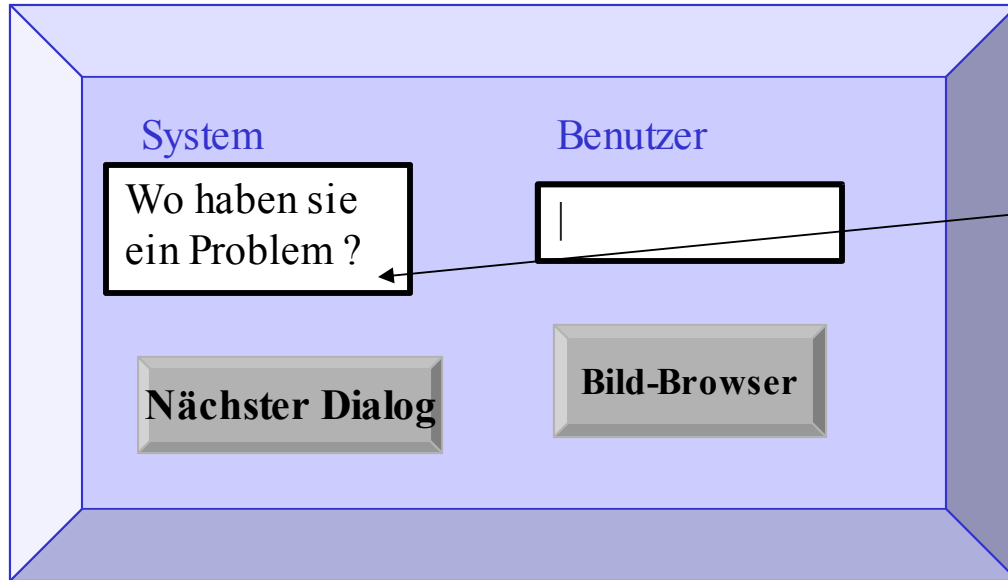
System-Architektur

Cristina Vertan
Walther v. Hahn

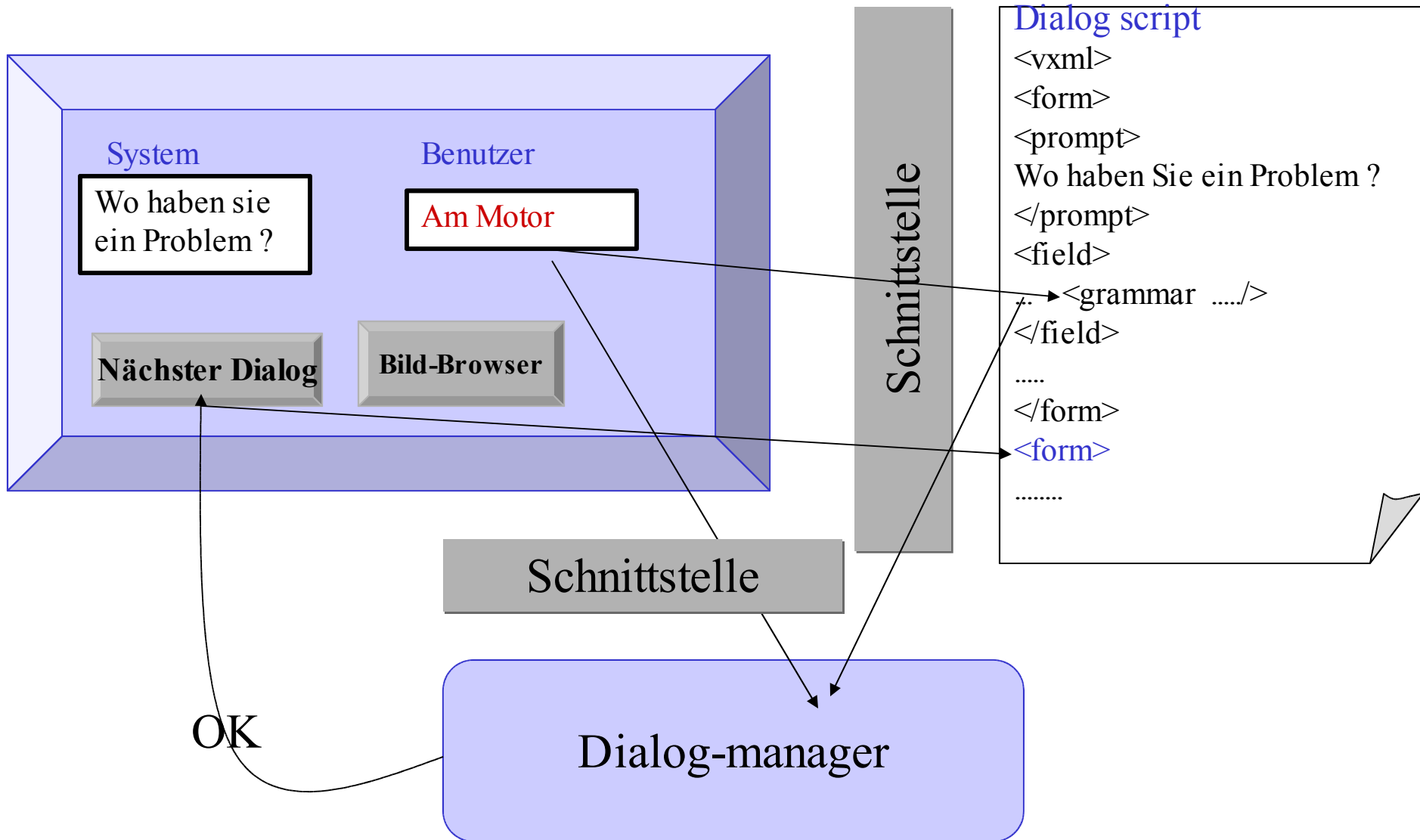
Grobarchitektur



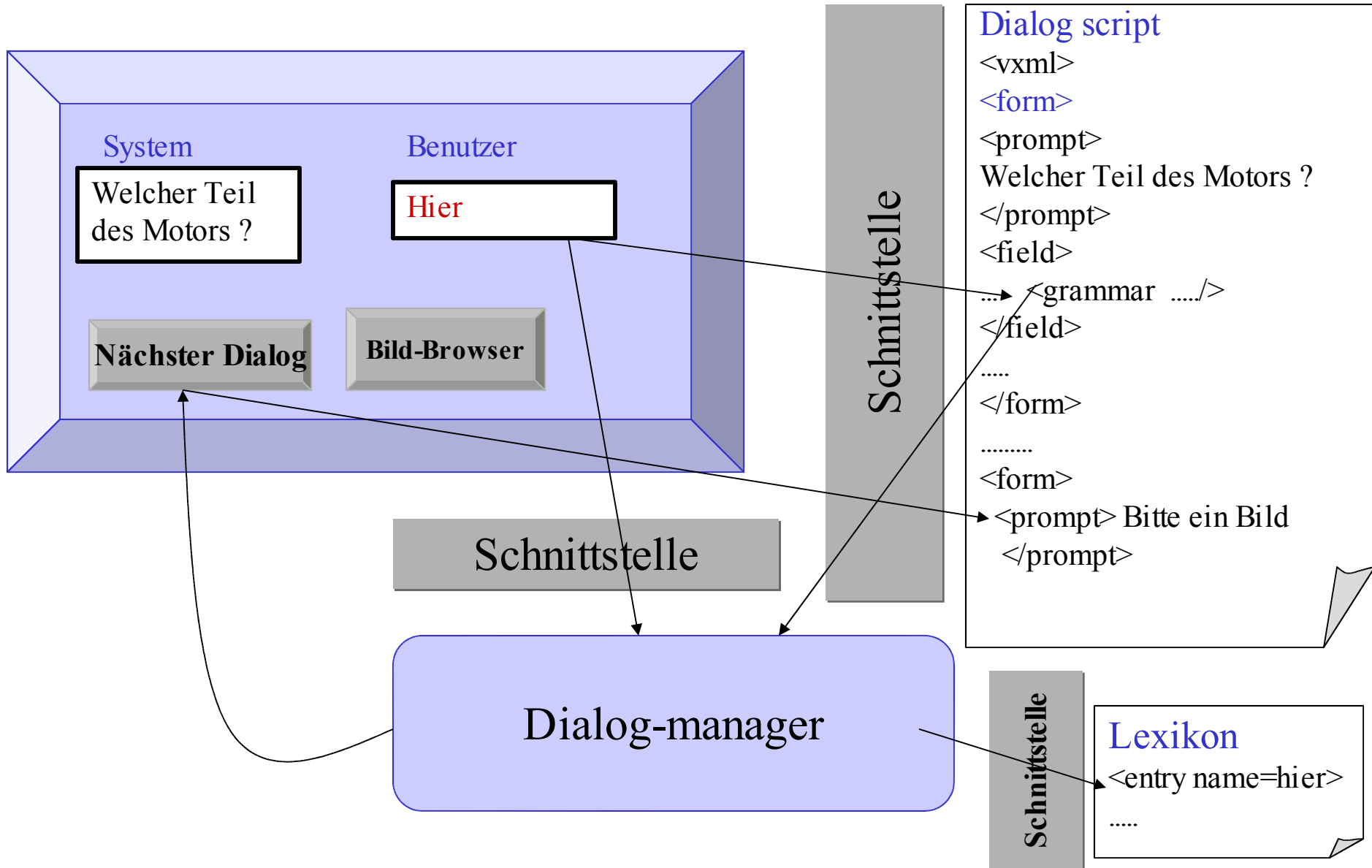
Dialogablauf - Prompts



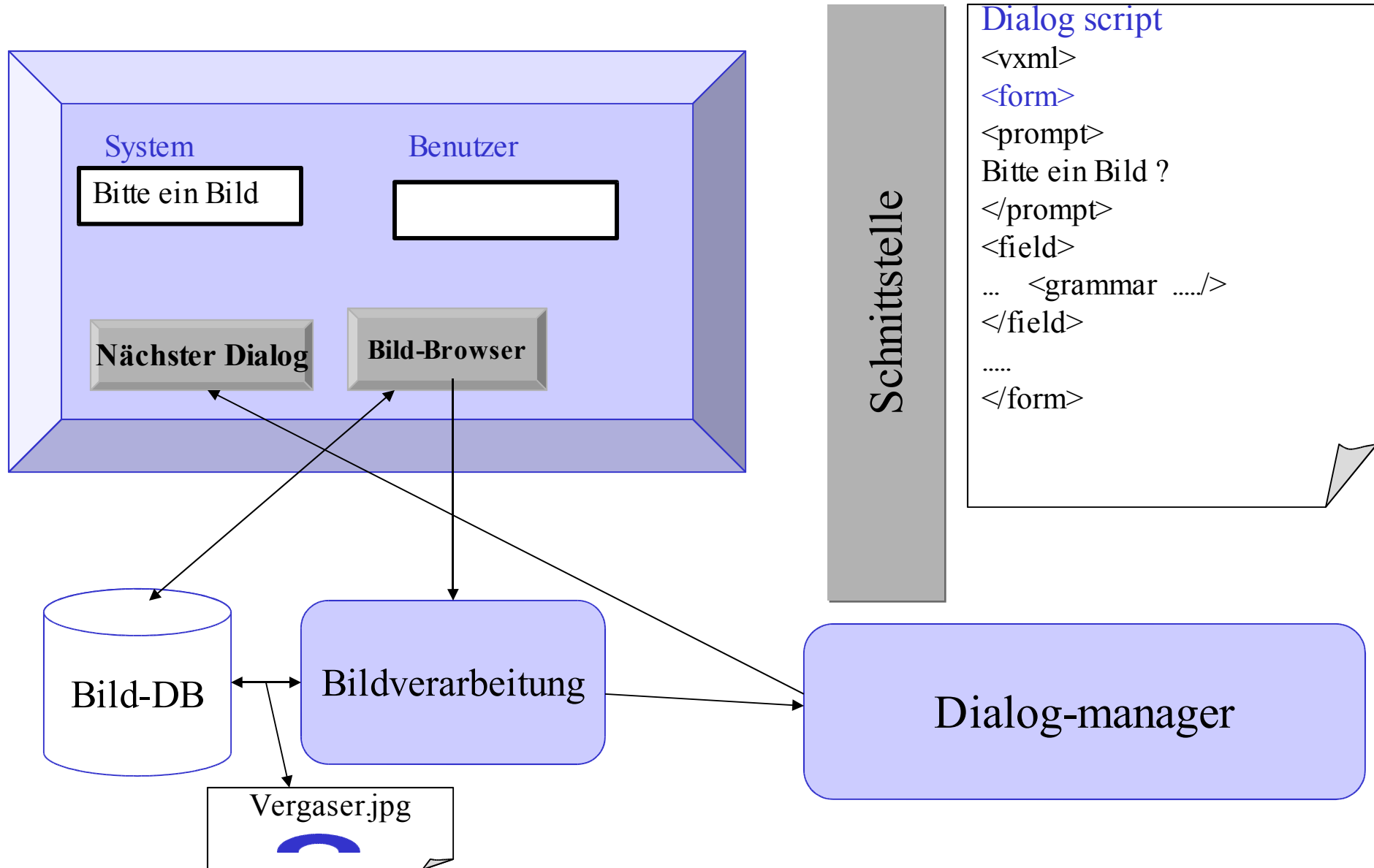
Dialogablauf - VoiceXML -like Sätze



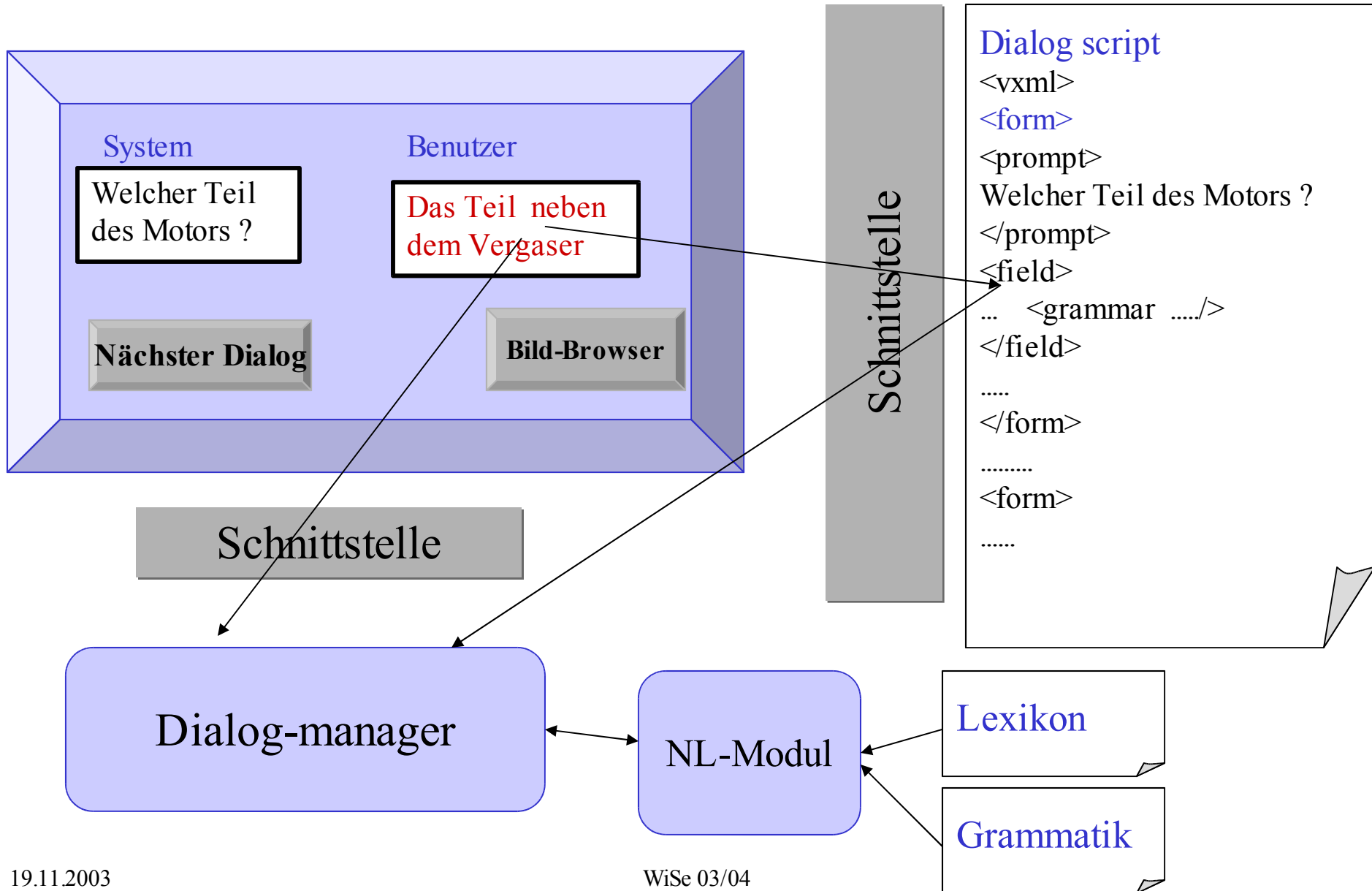
Dialogablauf - Bildunterstützung -1-



Dialogablauf - Bildunterstützung 2



Dialogabalauf - Höhere Sprachverarbeitung



User Interface

- Entwurf der graphischen Benutzungs-Schnittstelle
- Entwurf der Schnittstelle mit dem Dialog-Manager und Dialog-Skript:
 - Die Daten vom Dialog-Skript werden in eine Datenstruktur geladen
 - Diese Datenstruktur ermöglicht den Sprung zu weiteren Teilen des Skripts (des Dialogs). Am besten benutzt man Hash-Tables oder verkettete Listen
- Entwurf des Bild-Browsers

Der Dialog-Manager

- bearbeitet die Benutzer-Texteingabe :
 - für den ersten Dialog verifiziert er die Sprache
 - für jeden Dialog vergleicht er die Benutzereingabe gegenüber der spezifizierten Voice-Xml-Grammatik für diesen Schritt.
 - Falls ja, geht der Dialog weiter
 - Falls nein, prüft er, ob spezifische Wörter wie “hier”, “da”, “neben” usw. erscheinen.
 - Falls ja, fordert er ein Bild an
 - Falls nein, wird der Satz zu dem NL-Modul geschickt.
- Schnittstellen des Dialog-Managers mit dem Bildverarbeitungsmodule und dem NL-Modul sowie mit dem Lexikon müssen vereinbart werden

Das Bildverarbeitungs-Modul

- extrahiert die Kanten für das eingegebene Bild und identifiziert diese Kanten mit einem der existierenden Muster
- Wenn eine solche Identifizierung erfolgreich ist, wird der Name der passenden Muster-Datei an den Dialog-manager geschickt.

Das NL-Modul

- analysiert entsprechend der Grammatik und dem Lexikon komplexere Sätze, die von dem Dialog-Manager geschickt wurden
- identifiziert Pronomen.
- Schickt nur die Teile zurück, die für den Dialog relevant sind

Dateien und Datenstrukturen -1-

- Die Daten, die in XML / Voice-XML spezifiziert sind, müssen von Java-prozessen interpretiert werden.
- Es ist extrem rechneraufwändig, für jeden Verarbeitungsschritt eine Leseoperation zu machen.
- Deswegen müssen die Daten einmal am Anfang des Programms in Datenstrukturen hochgeladen werden.
- Danach arbeitet das Programm nur mit diesen Datenstrukturen
- Die Datenstrukturen sind nur während des Programablauf benutzbar, die Dateien bleiben auf der Platte.

Datenstrukturen - Dialog-Form

- Am besten benutzt man eine verkettete Liste oder ein Hashtable .

```
public class Forms {  
    public Hashtable dialog;  
    Document document;  
  
    public Forms (int adresse, float load, document doc){  
        dialog = new Hashtable(adresse, load)  
        document=doc  
        document.turn_number++;  
    }  
}
```

Datenstrukturen - Dialogskript

```
public class Document implements Tags {
    File file;
    public Forms FirstDialog;
    public Forms CurrentDialog;
    public int dialog_number;
    public EventListenerList changeListeners = new EventListenerList();
    // hier sollen Methoden, die auf verschiedene Teile des Dokuments
    // zugreifen können, implementiert werden
    // Dazu muß man die "Listeners" spezifizieren so daß jede Änderung im
    // Dokument für alle andere Programme, die das Dokument benutzen,
    // sichtbar wird
}
```

Identifizierung von Tags

```
public interface Tags{
    final String tag [] ={
        "Form",
        "/Form"
        "Field"
        "/Field"
        .....
    };
    public boolean isTag(String value);
    // die Methode wird von der Klasse die das Dokument manipuliert, implementiert
}
```

Lesen von Dateien

```
public void ReadFile (Document d) throws FileNotFoundException{
    FileReader in = new FileReader (d.file);
    StreamTokenizer tok= new StreamTokenizer(in);
    boolean tagopen=false;
    Forms CurrentDialog=d.currentDialog;
    tok.eolIsSignificant(true);
    tok.wordChars(63,127);
    //weitere Charaktere...
try{
while (tok.nextToken()!=tok.TT_EOF){
    if(tok.ttype==tok.TT_WORD){
        if(d.isTag(tok.sval)){
            if(tagopen) {
                if (tok.sval.equals("/")+currenttag){tagopen=false;currenttag=null;}
                else System.out.println ("Error");
            }
            else{
                currenttag=tok.sval;
                CurentDialog.dialog.put(currenttag) .....
            }
        }
    }
}
} catch(IOException e) }
```