Pflichtenheft zum erweiterten Programm (mit Teil 3)

Version 1.0

Firma: Brainstrom mkH
Mitarbeiter: Brigitte Bonder, Dominik Grüning, Wolfgang Hoppe, Holger Picker,
Ludmilla Scheinker, Martin Swobodzinski, Knut Willems
Projekt: U-Topp (UML-Tool Programmier-Praktikum)
Kunde: Dietmar Lammers
Version: 1.0
Datum: 25.01.01

1 Zielbestimmung

Es soll ein Case – Tool programmiert werden, das eine Hilfe bei der Softwareentwicklung mit UML sein soll.
Das Programm soll es ermöglichen, über Eingabemasken Use – Cases, Klassen, Objekte, Verbindungen bzw.
Verteiler zwischen den einzelnen Klassen bzw. Use – Cases und Pflichtenhefte eingeben zu können. Wichtig ist hierbei,
dass alle erstellten Objekte einzeln abgespeichert werden können im Datenaustauschformat XML. Die
Komponenten sollen graphisch flexibel dargestellt werden, daher sollen unterschiedliche Möglichkeiten der
Detaillierung darstellbar sein. Klassen sind z.B. sowohl mit als auch ohne Attribute und Operationen
darstellbar. Im Anschluss an die Erstellung der einzelnen Komponenten sollen diese wahlweise in ein
Use-Case-Diagramm oder Klassendiagramm (keine weiteren Diagrammarten) eingefügt werden,
um so die graphische Visualisierung eines Projektes zu ermöglichen. Des weiteren soll die Möglichkeit
der Aufnahme eines zum Projekt gehörenden Pflichtenheftes gegeben werden. Alle Projektdaten
(Diagramme, Pflichtenheft) sollen gemeinsam gespeichert und geladen werden können.

1.1 Musskriterien

1. Eingabe der UML – Komponenten "Klasse", "Objekt" und "Use – Case“, eines Pflichtenheftes und
   Verbindungen / Verteiler zwischen Komponenten über Erfassmasken
2. graphische Darstellung / Ausgabe dieser Komponenten
3. variierender Detaillierungsgrad bei der Anzeige der Komponenten
4. Editiermöglichkeit der Komponenten
5. Einfügen der Komponenten in ein Use-Case- bzw. Klassendiagramm
6. graphische Darstellung / Ausgabe der Diagramme
7. variierender Detaillierungsgrad der Diagramme
8. Editiermöglichkeit der Diagramme

1.2 Wunschkriterien

1. Möglichkeit der Eingabe von frei wählbaren Datenzusätzen
2. Die Diagramme sollen in ihrer Auflösung frei skalierbar sein und verschiedene Detailstufen aufweisen
3. Der Verlauf der Linie einer Verbindung soll automatisch berechnet werden
4. Die Elemente sollen im Diagramm-Fenster frei selektiert und bearbeitet werden können
5. Das Diagram soll so modular gestaltet werden, dass einzelne Klassen zur Modellierung der Komponenten
   getrennt vom Case-Tool verwendet werden können.

1.3 Abgrenzungskriterien

1. Erstellung der Diagramme soll auf Use-Case- und Klassendiagramme beschränkt sein.

2 Produkteinsatz

2.1 Anwendungsbereiche

1. Softwareengineering (im Bereich der Planung, Analyse und Design mit UML)
2. Präsentationen von Softwareentwicklungsideen

2.2 Zielgruppen

1. Programmierer
2. Softwareentwickler, die UML verwenden

2.3 Betriebsbedingungen:

1. Benutzer sollte Erfahrungen über Softwareengineering haben

3 Produktumgebung

3.1 Software

1. JDK – Environment

3.2 Hardware

1. eine hinreichend leistungsfähige Hardware, auf der Java läuft

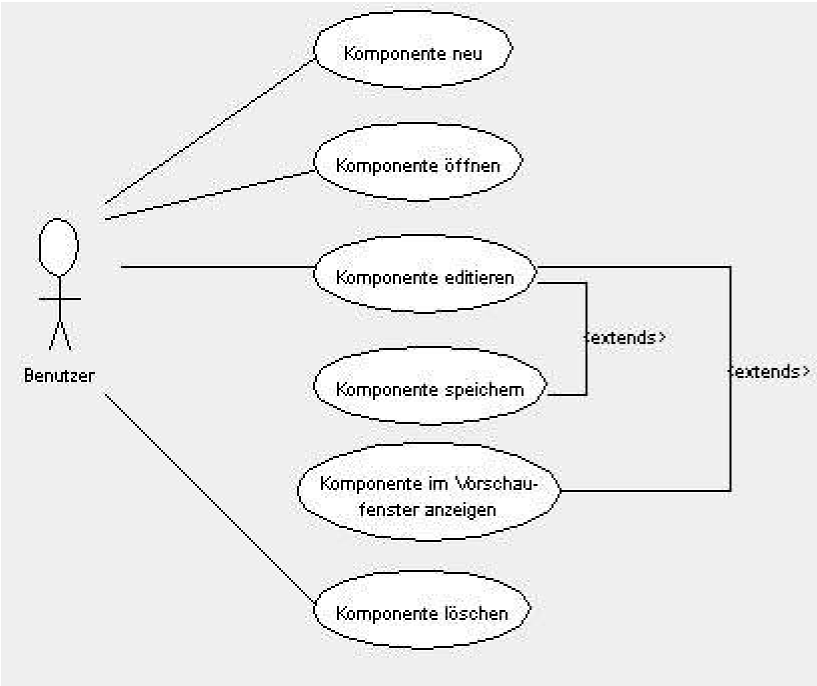
3.3 Schnittstellen

1. XML als Datenaustauschmöglichkeit

4 Produktinformation

4.1. Use Case Diagramme

4.1.1 Für die einzelnen Komponenten



1. Komponente (= Use - Case, Objekt, Klasse, Verbindung, Verteiler) neu

Akteur	Benutzer
Vorbedingung	Es muss ein Projekt existieren, man befindet sich entweder im Modus: "Use-Case-Diagramm" oder "Klassendiagramm"
Nachbedingung	Komponente wird zur Projektliste hinzugefügt
Ablauf	Anlegung der Komponente in der Liste mit Default - Werten (Name), Übergang in den Edit - Vorgang
Ausnahmen	-

2. Komponente öffnen

Akteur	Benutzer
Vorbedingung	Es muss ein Projekt existieren, man befindet sich entweder im Modus: "Use-Case-Diagramm" oder "Klassendiagramm" und die zu ladende Komponente darf nicht in der Liste enthalten sein.
Nachbedingung	Daten geladen, Name in der Liste des Diagramms angezeigt
Ablauf	zu öffende Komponente wählen, Daten werden geladen, der Name wird angezeigt
Ausnahmen	Verbindung / Verteiler im Projekt schon vorhanden: Fehler

3. Komponente editieren

Akteur	Benutzer
Vorbedingung	Komponente in der Liste selektiert
Nachbedingung	Datenveränderungen übernommen, Ein Update der Bildschirmanzeige wurde erzeugt
Ablauf	Daten verändern, übernehmen und neue Daten anzeigen
Ausnahmen	Bei Abbruch werden geänderte Daten nicht übernommen

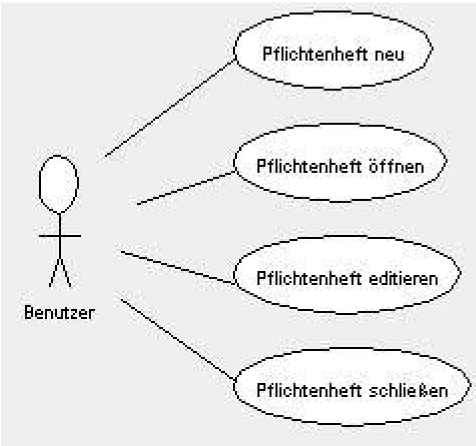
4. Komponente löschen

Akteur	Benutzer
Vorbedingung	Es muss ein Projekt existieren, man befindet sich entweder im Modus: "Use-Case-Diagramm" oder "Klassendiagramm" und die zu löschende Komponente muss am Bildschirm selektiert sein.
Nachbedingung	Komponente wurde aus Liste und Diagramm entfernt Bei Komponente = Klasse, Objekt, Use-Case und Verteiler werden darauf verweisende Verbindungen ebenfalls aus dem Diagramm entfernt
Ablauf	Die markierte Komponente wird aus der Liste entfernt und intern gelöscht
Ausnahmen	keine Komponente markiert

5. Komponente im Vorschaumodus anzeigen

Akteur	Benutzer
Vorbedingung	Eine Komponente vorhanden
Nachbedingung	Die Komponente ist angezeigt.
Ablauf	Die Komponente wird im Vorschaufenster angeschaut.
Ausnahmen	Keine Komponente vorhanden.

4.1.2 Für das Pflichtenheft



1. Pflichtenheft neu

Akteur	Benutzer
Vorbedingung	Es ist ein Projekt geöffnet, es gibt kein Pflichtenheft im Projekt.
Nachbedingung	Es gibt ein Pflichtenheft im Projekt.
Ablauf	Es wird ein Pflichtenheft mit Defaultwerten erzeugt, Übergang in den Editier - Modus.
Ausnahmen	-

2. Pflichtenheft öffnen

Akteur	Benutzer
Vorbedingung	Es gibt kein Pflichtenheft im Projekt, das zu öffnende Pflichtenheft befindet sich auf externem Medium.
Nachbedingung	Es gibt ein Pflichtenheft im Projekt.
Ablauf	Das Pflichtenheft wird geladen.
Ausnahmen	-

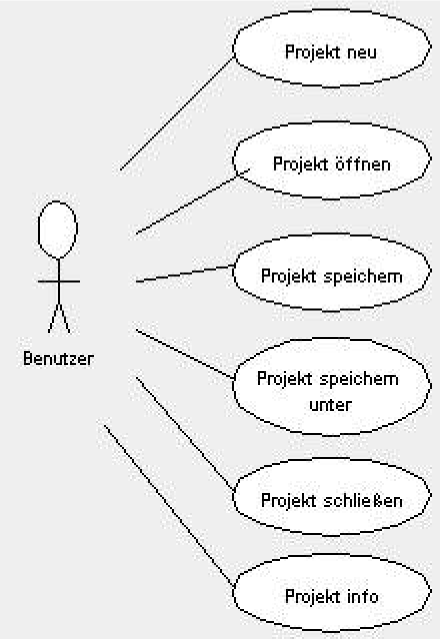
3. Pflichtenheft editieren

Akteur	Benutzer
Vorbedingung	Es gibt ein Pflichtenheft im Projekt.
Nachbedingung	Es gibt ein Pflichtenheft mit eventuellen Änderungen im Projekt.
Ablauf	Die Editiermethode der Komponente Pflichtenheft wurd aufgerufen.
Ausnahmen	-

4. Pflichtenheft schließen

Akteur	Benutzer
Vorbedingung	Es gibt ein Pflichtenheft im Projekt.
Nachbedingung	Es gibt kein Pflichtenheft im Projekt.
Ablauf	Das Pflichtenheft wird entfernt.
Ausnahmen	-

4.1.3 Für das Gesamtprojekt



1. Projekt neu

Akteur	Benutzer
Vorbedingung	Es gibt kein geöffnetes Projekt
Nachbedingung	Es gibt ein Projekt.
Ablauf	Es wird ein leeres Projekt mit Defaultwerten erzeugt.
Ausnahmen	-

2. Projekt öffnen

Akteur	Benutzer
Vorbedingung	Das gewählte Projekt existiert auf einem externen Medium.
Nachbedingung	Es gibt ein Pflichtenheft im Projekt.
Ablauf	1. Dialogfenster für die Auswahl eines vorhandenen Projekts wird geöffnet. 2. Die Komponenten und Diagramme des Projektes werden geladen.
Ausnahmen	-

3. Projekt speichern

Akteur	Benutzer
Vorbedingung	Es gibt ein geöffnetes Projekt.
Nachbedingung	Das Projekt ist extern gespeichert.
Ablauf	Jede Komponente wird in die entsprechende Datei gespeichert. Existiert für eine Komponente kein Dateiname, so wird dieser abgefragt.
Ausnahmen	-

4. Projekt speichern unter

Akteur	Benutzer
Vorbedingung	Es gibt ein geöffnetes Projekt.
Nachbedingung	Das Projekt ist extern unter dem gewünschten Pfad gespeichert.
Ablauf	Dialogfenster speichern unter wird geöffnet. Jede Komponente wird in der entsprechenden Datei gespeichert. Existiert für eine Komponente kein Dateiname, so wird dieser abgefragt.
Ausnahmen	-

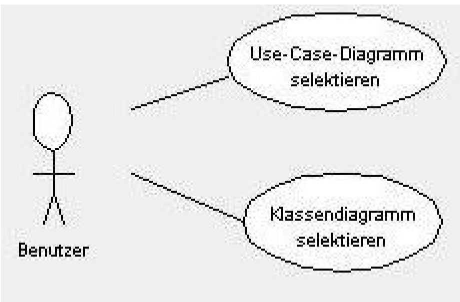
5. Projekt schließen

Akteur	Benutzer
Vorbedingung	Es gibt ein geöffnetes Projekt.
Nachbedingung	Das Projekt ist geschlossen.
Ablauf	Das Projekt wird intern gelöscht.
Ausnahmen	-

6. Projekt Info

Akteur	Benutzer
Vorbedingung	Es gibt ein geöffnetes Projekt.
Nachbedingung	Das Info-Fenster ist geöffnet.
Ablauf	Das Info-Fenster zum Projekt wird angezeigt.
Ausnahmen	-

4.1.4 Für die Diagramme



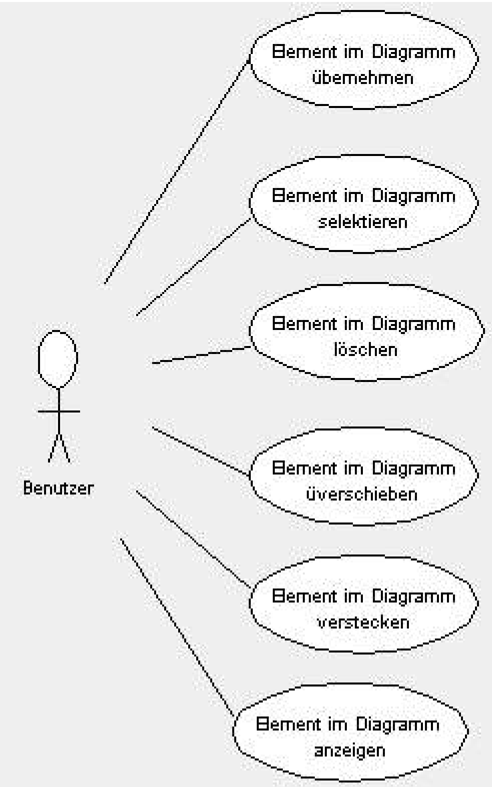
1. Use-Case-Diagramm auswählen

Akteur	Benutzer
Vorbedingung	Es gibt ein geöffnetes Projekt.
Nachbedingung	Das Use-Case-Diagramm wird auf dem Bildschirm angezeigt, eventuell mit schon vorhandenen Daten.
Ablauf	Das Use-Case-Diagramm wird auf dem Bildschirm angezeigt, eventuell werden schon vorhandene Daten in den Listen angezeigt.
Ausnahmen	-

2. Klassendiagramm auswählen

Akteur	Benutzer
Vorbedingung	Eis gibt ein geöffnetes Projekt.
Nachbedingung	Das Klassen-Diagramm wird auf dem Bildschirm angezeigt, eventuell mit schon vorhandenen Daten.
Ablauf	Das Klassen-Diagramm wird auf dem Bildschirm angezeigt, eventuell werden schon vorhandene Daten in den Listen angezeigt.
Ausnahmen	-

4.1.5 Für die Darstellung im Diagramm



Die Abläufe der Anzeigeeigenschaft des Projektes sind bei allen Aktionen, die sich auf die graphische Ausgabe beziehen, integriert.

1. Element (im Diagramm) übernehmen

Akteur	Benutzer
Vorbedingung	Die selektierte Komponente befindet sich noch nicht in der Canvas.
Nachbedingung	Die Komponente ist angezeigt.
Ablauf	Die selektierte Komponente wird wird in der Canvas wie folgt angezeigt: 1. Bei Verbindung: Durch das Anklicken der entsprechenden Elemente in der Canvas werden diese durch die ausgewählte Verbindung verbunden. 2. Sonst werden die Komponenten durch Klick auf die Canvas an dieser Stelle angezeigt. 3. Der Button "Übernehmen" wechselt zu "Löschen".
Ausnahmen	keine Komponente markiert

2. Element (im Diagramm) selektieren

Akteur	Benutzer
Vorbedingung	Mindestenst ein Element wird in der Canvas angezeigt.
Nachbedingung	Das Element wurde in der Canvas selektiert.
Ablauf	1. Der Benutzer selektiert das Element durch Klicken. 2. Der Button "Übernehmen" wechselt zu "Löschen". 3. Der Benutzer hat die Möglichkeit, das Element zu verschieben.
Ausnahmen	keine Komponente vorhanden

3. Element (im Diagramm) löschen

Akteur	Benutzer
Vorbedingung	Das selektierte Element befindet sich in der Canvas.
Nachbedingung	Das Element wurde in der Canvas gelöscht.
Ablauf	Das Element wird aus der Canvas gelöscht.
Ausnahmen	Keine Komponente in der Canvas selektiert.

4. Element (im Diagramm) verschieben

Akteur	Benutzer
Vorbedingung	Das selektierte Element befindet sich in der Canvas.
Nachbedingung	Das Element wurde verschoben.
Ablauf	Das Element wird samt aller Verbindungen im Diagramm verschoben.
Ausnahmen	Keine Komponente in der Canvas selektiert.

5. Element (im Diagramm) verstecken

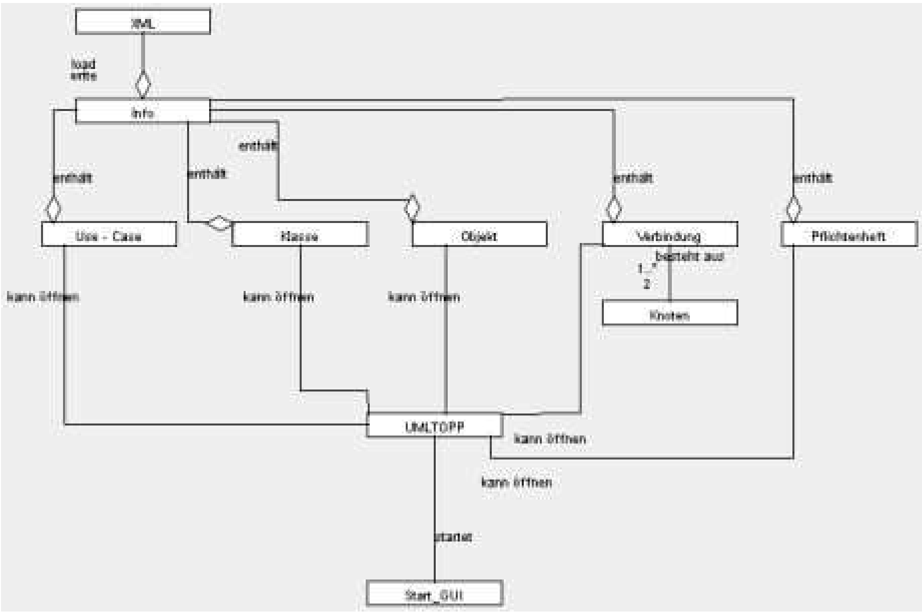
Akteur	Benutzer
Vorbedingung	Eine Komponente ist selektiert, ist in der Canvas angezeigt.
Nachbedingung	Das Element wurde versteckt..
Ablauf	Die Komponente ist in der Canvas nicht mehr sichtbar, befindet sich aber noch im Diagramm.
Ausnahmen	Keine Komponente in der Canvas.

6. Element (im Diagramm) anzeigen

Akteur	Benutzer
Vorbedingung	Eine Komponente ist selektiert, ist in der Canvas bereits versteckt.
Nachbedingung	Das Element wurde angezeigt.
Ablauf	Die Komponente wird in der Canvas wieder sichtbar.
Ausnahmen	Keine Komponente in der Canvas oder keine Komponente versteckt.

5. Konzeptklassen und Daten

5.1. Klassendiagramm



5.2 Klasse XML

XML
Char safelastchar int internalcount int maxbytes Vector vectorarray
loadXML(String filename) saveXML(String filename) getChar() : Char lastChar() : Char writeChar(Char c) getTag(int maxlength) : String writeTag(String tag, Int leerzeichen) getString(Int maxlength, String tag) : String writeString(String st) skipTag(String tag)

5.3 Klasse Info

Info
String info_name String filepath String info_version String erstellungsdatum String aenderungsdatum String info_autor String info_firma String kommentar
edit(String fenstertitel, String aufrufer) read(Xml xmlobjekt) write(Xml xmlobjekt) getFilepath : String setFilepath(String filepath)

5.4 Die Klassen für Objekt

5.4.1 Klasse Methode

Methode
String methode_name String methode_gueltigkeit String methode_rueckgabetyp String methode_parameter String methode_zusicherung

5.4.2 Klasse Attribut

Attribut
String attribut_name String attribut_gueltigkeit String attribut_datentyp String attribut_attributwert String attribut_zusicherung

5.4.3 Klasse Objekt

Objekt
Info objekt_info String objekt_stereotyp String objekt_klasse String objekt_eigenschaftswerte Vector objekt_methoden (Methode) Vector objekt_attribute (Attribut) detail
edit() load(String filepath) save(String filepath) draw(Float xcoo, Float ycoo, Float scale, Int detail)

5.5 Die Klassen der Klasse

5.5.1 Klasse Methode

Methode
String methode_name String methode_gueltigkeit String methode_rueckgabetyp String methode_parameter String methode_zusicherung

5.5.2 Klasse Attribute

Attribute
String attribut_name String attribut_gueltigkeit String attribut_datentyp String attribut_initialwert String attribut_zusicherung

5.5.3 Klasse Klasse

Klasse
Info klasse_info String klasse_stereotyp String klasse_eigenschaftswerte Vector klasse_methoden (Methode) Vector klasse_attribute (Attribut) detail
edit() load(String filepath) save(String filepath) draw(Float xcoo, Float ycoo,Float scale, Int detail)

5.6 Die Klassen der Verbindung

5.6.1 Klasse Knoten

Knoten
String pfeiltyp String multiplikator String zusicherung_knoten

5.6.2 Klasse Verbindung

Verbindung
Info header String beschriftung String stereotyp String zusicherung Knoten knoten1 Knoten knoten2 detail
edit() load(String filepath) save(String filepath) draw(Float xcoo, Float ycoo, Float scale, Int detail)

5.7 Klasse Verteiler

Verteiler
Info header String beschriftung detail
edit() load(String filepath) save(String filepath) draw(Float xcoo, Float ycoo, Float scale, Int detail)

5.8 Klasse Use-Case

Use-Case
Info usecase_info String usecase_nummer Vector usecase_attribute Vector usecase_attribute_text detail
edit() load(String filepath) save(String filepath) draw(Float xcoo, Float ycoo, Float scale, Int detail)

5.9 Klasse Pflichtenheft

Pflichtenheft
Info pflichtenheft_info String pflichtenheft_text
edit() load(String filepath) save(String filepath)

5.10 Die Klassen für das Projekt

5.10.1 Klasse Koordinate

Koordinate
Float xcoo Float ycoo

5.10.2 Klasse Element

Element
String elementtyp vector coordinate (Koordinate) display

5.10.3 Klasse Projekt

Projekt
Info projekt_info diagramm Pflichtenheft projekt_pflichtenheft vector klassen_elemente vector klassenv_elemente vector klassen_elementdata (Element) vector klassenv_elementdata (Element) vector usecase_elemente vector usecasev_elemente vector usecase_elementdata (Element) vector usecasev_elementdata (Element) masterusedetail masterclassdetail Float masterusescale Float masterclassscale
load(String filepath) save(String filepath)

5.11 DDT

```
<?xml version="1.0"?>
<!--
====================================================================== -->
<!-- Header DTD       -->
<!--
====================================================================== -->
<!-- wird jeweils eingebunden in die DTDs von       -->
<!-- Objekt, Klasse, Verbindung, UseCase und Pflichtenheft       -->
<!ELEMENT header (name?, filepath?, version?,
date?, lastchanged?,
author?, company?, comment?) >
<!ELEMENT name (#PCDATA) >
<!ELEMENT filepath (#PCDATA) >
<!ELEMENT version (#PCDATA) >
<!ELEMENT date (#PCDATA) >
<!ELEMENT lastchanged (#PCDATA) >
<!ELEMENT author (#PCDATA) >
<!ELEMENT company (#PCDATA) >
<!ELEMENT comment (#PCDATA) >

<!--
====================================================================== -->
<!-- Objekt DTD       -->
<!--
====================================================================== -->
<!ELEMENT objekt (header, body) >
<!ELEMENT body (stereotyp?, klasse?, eigenschaftswerte?,
methode*, attribut*, detail?) >
<!ELEMENT stereotyp (#PCDATA) >
<!ELEMENT klasse (#PCDATA) >
<!ELEMENT eigenschaftswerte (#PCDATA) >
<!ELEMENT methode (name, gueltigkeit?, rueckgabetyp?
parameter?, zusicherung?) >
<!ELEMENT attribut (name, gueltigkeit?, datentyp?
wert?, zusicherung?) >
<!ELEMENT gueltigkeit (#PCDATA) >
<!ELEMENT rueckgabetyp (#PCDATA) >
<!ELEMENT datentyp (#PCDATA) >
<!ELEMENT parameter (#PCDATA) >
<!ELEMENT wert (#PCDATA) >
<!ELEMENT zusicherung (#PCDATA) >
<!ELEMENT detail (#PCDATA) >
```

```
<!--
==================================================================== -->
<!-- Klasse DTD       -->
<!--
==================================================================== -->
<!ELEMENT klasse (header, body) >
<!ELEMENT body (stereotyp?, eigenschaftswerte?,
methode*, attribut*, detail?) >
<!ELEMENT stereotyp (#PCDATA) >
<!ELEMENT eigenschaftswerte (#PCDATA) >
<!ELEMENT methode (name, gueltigkeit?, rueckgabetyp?
parameter?, zusicherung?) >
<!ELEMENT attribut (name, gueltigkeit?, datentyp?
initialwert?, zusicherung?) >
<!ELEMENT gueltigkeit (#PCDATA) >
<!ELEMENT rueckgabetyp (#PCDATA) >
<!ELEMENT datentyp (#PCDATA) >
<!ELEMENT parameter (#PCDATA) >
<!ELEMENT initialwert (#PCDATA) >
<!ELEMENT zusicherung (#PCDATA) >
<!ELEMENT detail (#PCDATA) >

<!--
==================================================================== -->
<!-- Verbindung DTD       -->
<!--
==================================================================== -->
<!ELEMENT verbindung (header, body) >
<!ELEMENT body (beschriftung?, stereotyp?,
zusicherung?, knoten*, detail?) >
<!ELEMENT beschriftung (#PCDATA) >
<!ELEMENT stereotyp (#PCDATA) >
<!ELEMENT zusicherung (#PCDATA) >
<!ELEMENT knoten (pfeiltyp?, multiplikator?,
zusicherungs?) >
<!ELEMENT pfeiltyp (#PCDATA) >
<!ELEMENT multiplikator (#PCDATA) >
<!ELEMENT zusicherung (#PCDATA) >
<!ELEMENT detail (#PCDATA) >

<!--
==================================================================== -->
<!-- Verteiler DTD       -->
<!--
==================================================================== -->
<!ELEMENT verteiler (header, body) >
<!ELEMENT body (beschriftung?, detail?) >
<!ELEMENT beschriftung (#PCDATA) >
<!ELEMENT detail (#PCDATA) >
```

```
<!--
===================================================================== -->
<!-- UseCase DTD       -->
<!--
===================================================================== -->
<!ELEMENT usecase (header, body) >
<!ELEMENT body (nummer?, eintrag*, zusicherung?,
detail?) >
<!ELEMENT eintrag (bezeichnung, text) >
<!ELEMENT bezeichnung (#PCDATA) >
<!ELEMENT text (#PCDATA) >
<!ELEMENT zusicherung (#PCDATA) >
<!ELEMENT detail (#PCDATA) >

<!--
===================================================================== -->
<!-- Pflichtenheft DTD       -->
<!--
===================================================================== -->
<!ELEMENT pflichtenheft (header, body) >
<!ELEMENT body (text) >
<!ELEMENT text (#PCDATA) >

<!--
===================================================================== -->
<!-- Projekt DTD       -->
<!--
===================================================================== -->
<!ELEMENT projekt (header, body) >
<!ELEMENT body (usecasediag?, klassendiag?,
pflichtenheft?) >
<!ELEMENT usecasediag (scale?, detail?,
usecaseelement*,
lineelement*) >
<!ELEMENT klassendiag (scale?, detail?,
klasseelement*,
lineelement*) >
<!ELEMENT scale (#PCDATA) >
<!ELEMENT detail (#PCDATA) >
<!ELEMENT usecaseelement (elementdata, [usecase|akteur]) >
<!ELEMENT klasseelement (elementdata, [klasse|objekt]) >
<!ELEMENT lineelement (elementdata, [verbindung|verteiler]) >
<!ELEMENT akteur (#PCDATA) >
<!ELEMENT elementdata (display, coordinate*) >
<!ELEMENT display (#PCDATA) >
<!ELEMENT coordinate (xcoo, ycoo) >
<!ELEMENT xcoo (#PCDATA) >
<!ELEMENT ycoo (#PCDATA) >
```

6. Leistungen / Funktionen

6.1 Komponenten laden

- Laden von Use-Cases, Klassen, Objekten, Verbindungen und Pflichtenheften

6.2 Komponenten editieren

- Editieren von Komponenten über Editiermasken, die auf die jeweilige Komponente zugeschnitten sind

6.3 Komponenten anzeigen

- Die unterschiedlichen Komponenten können in einem im Editierfenster enthaltenden Vorschaufenster angezeigt werden, hierbei kann bei Klasse und Objekt ein Detaillierungsgrad gewählt werden.

6.4 Komponenten speichern

Die Daten der unterschiedlichen Komponenten werden mittels XML gespeichert.

6.5 Use-Case-Diagramm darstellen

Use-Cases und Verbindungen können in ein Diagramm geladen werden und beliebig angeordnet werden.

Unterschiedliche Detaillierungsgrade sind möglich.

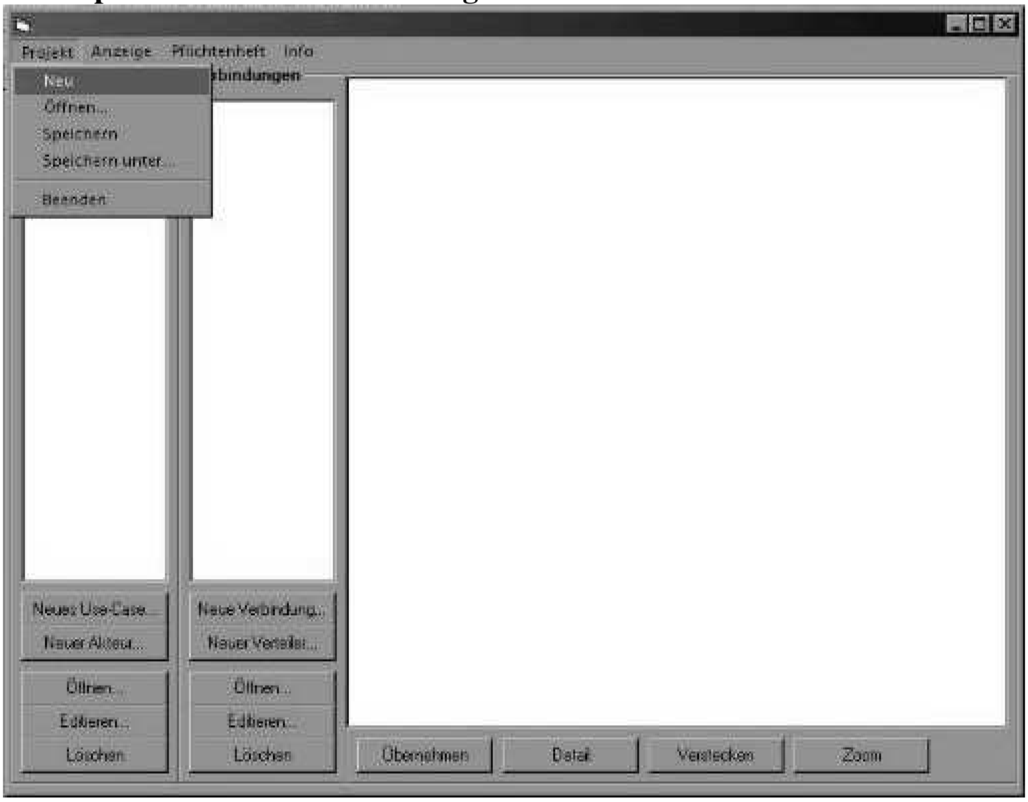
6.5 Klassen-Diagramm darstellen

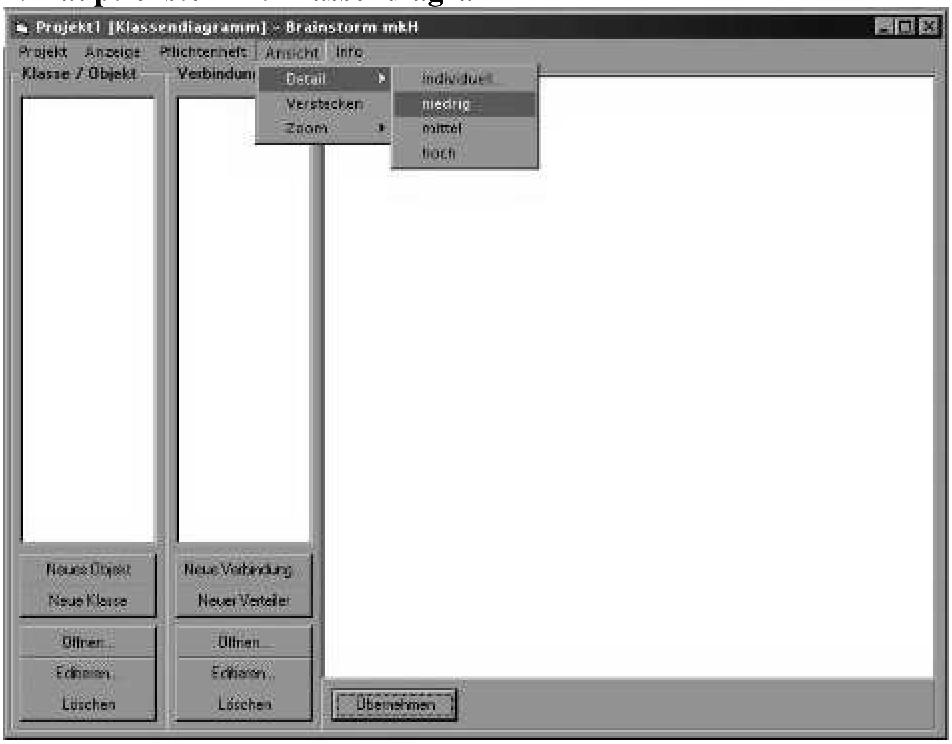
Klassen, Objekte und Verbindungen können in ein Diagramm geladen werden und beliebig angeordnet werden.

Unterschiedliche Detaillierungsgrade sind möglich.

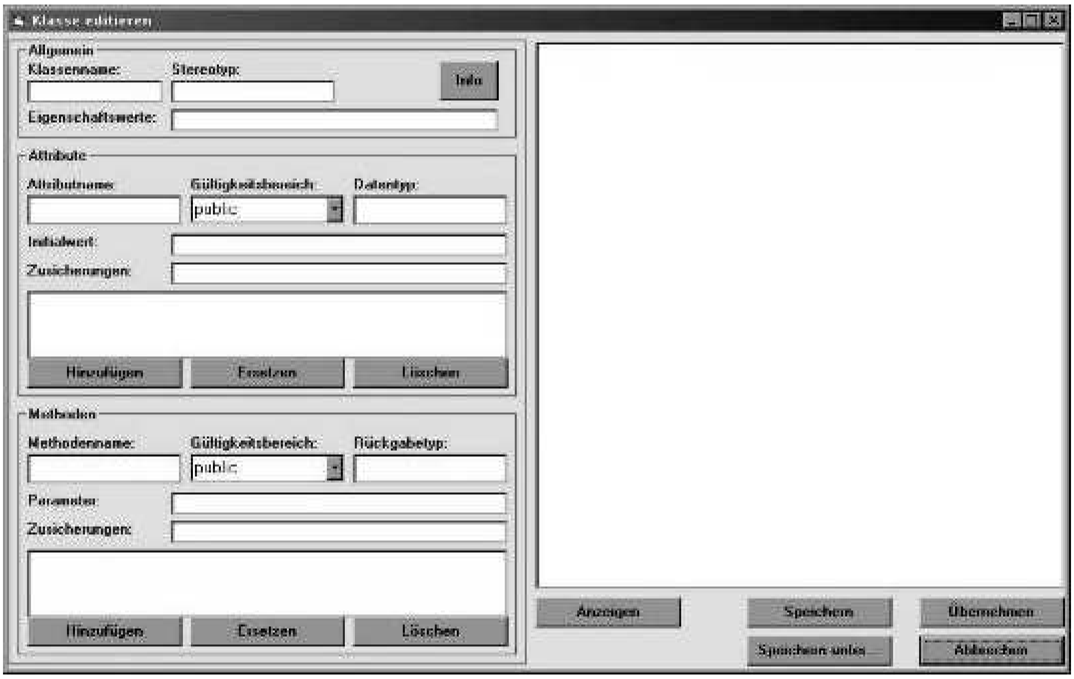
7. Benutzeroberfläche

7.1 Grundkonzeption
Das Programm enthält eine Menüleiste mit den wichtigsten Grundfunktionen. .

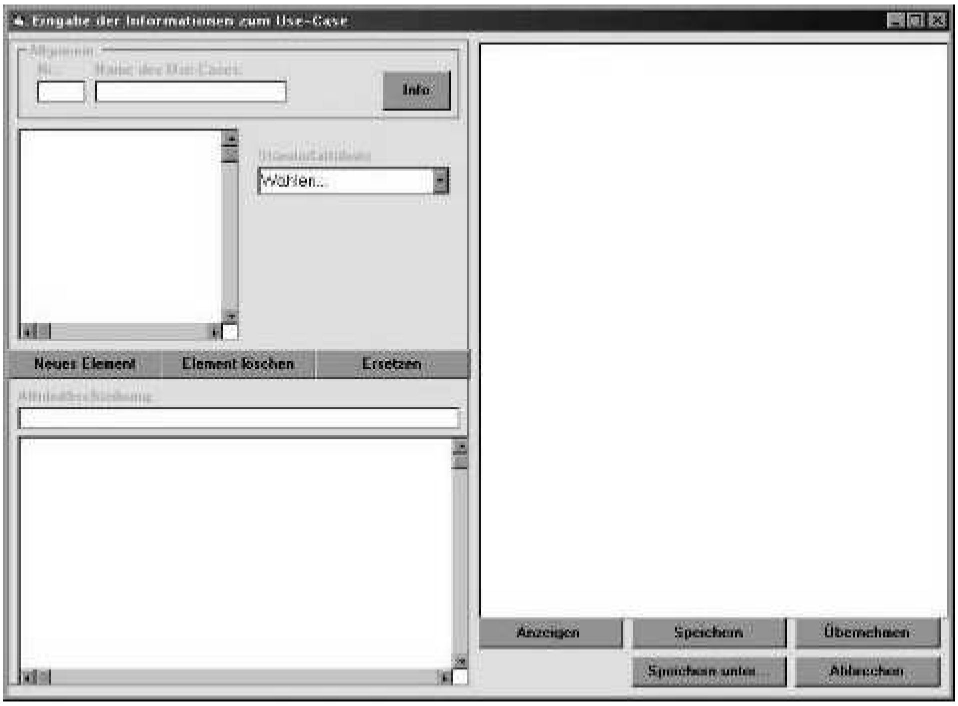
7.2 Screenshots
1. Hauptfenster mit Use-Case-Diagramm


2. Hauptfenster mit Klassendiagramm


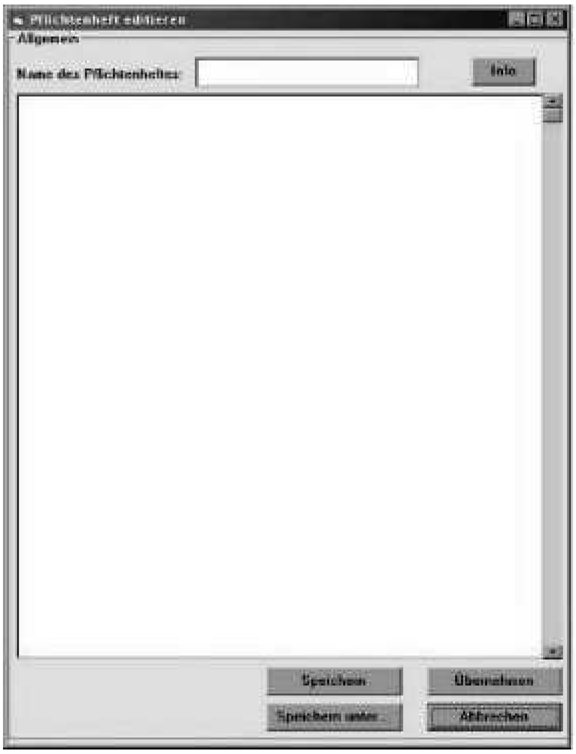
3. Klasseneditierfenster



4. Use-Case-Editierfenster



5. Pflichtenheft-Editierfenster



Letzte Änderung: 7.02.01