1. Zielbestimmungen

1.1 Mußbestimmungen

Das Programm soll Erfassmasken für die UML-Komponenten Klassen, Objekte, Use-Cases, Verbindungen und Pflichtenheft enthalten. Diese Komponenten sollen in unterschiedlichem Detaillierungsgrad graphisch darstellbar und auch dauerhaft speicherbar sein. Desweiteren gibt das Programm dem Benutzer die Möglichkeit die Komponenten zu editieren, d.h. die gespeicherten Daten zu verändern

1.2 Kannbestimmungen

Das Programm könnte bestimmte Arten von Diagrammen erstellen, wie z.B. Klassen-, Use-Case- und Sequenzdiagramme. Wünschenswert wäre unter Umständen, diese Diagramme in verbreitete Formate exportieren und/oder drucken zu können. Außerdem könnte man die einzelnen Komponenten so vernetzen, dass zum Beispiel automatisch festgestellt würde, ob die eingegebenen Daten konsistent sind.

1.3 Abgrenzungskriterien

Im Gegensatz zu anderen Systemen (wie z.B. ArgoUML) wird nicht die Möglichkeit bestehen, vorhandene Daten zur Generation von Quellcode (z.B. "Rahmen" von Javaklassen) zu verwenden. Die Darstellung der Komponenten erfolgt einzeln, d.h. nicht in einem Gesamtbild. Weiterhin wird man vorhandene Information nicht graphisch (d.h. durch direkten Zugriff auf eine Ansicht)ändern können.

2 Produkteinsatz

2.1 Zielgruppe

Das Programm U-Topp ist für Benutzer einer objektorientierten Softwareentwicklung gedacht. Softwareentwicklung wird immer komplexer und anspruchsvoller. Aus diesen Gründen ist es sinnvoll nach Methoden zu suchen, welche die Komplexität beherrschbar machen und dem Anwender eine zuverlässige Übersicht über die bei der Softwareentwicklung zu berücksichtigen Komponenten zu geben, um so die Qualität und die Zuverlässigkeit der Software aufrechtzuerhalten. Das Programm U-Topp soll dem Entwickler (in Ahnlehnung an Argo-UML) ein wenig hilfreich in der Planungsphase zur Seite stehen, um den Blick für das Wesentliche nicht zu verlieren.

2.1 Betriebsbedingungen

Das Programm kann sicherlich später von folgender Webseite heruntergeladen werden: wwwmath.uni-muenster.de/cs/u/lammers/Programmierpraktikum. Ansprechpartner sind die für dieses Programm Verantwortlichen.

3. Produktumgebung

3.1 Software

Das Produkt benötigt eine virtuelle Java-Maschine, z.B. das Java-Runtime-Environment von Sun, das auch bei der Entwicklung verwendet wird.

3.2 Hardware

Die Software sollte im wesentlichen Plattform-unabhängig sein. Zumindest sollte sie unter folgenden zwei Hardware-Umgebungen laufen:

- Genügend ausgestatteter PC
- Sun-Workstations im mathematischen Institut

Darüber hinaus wird keine besondere Hardware benötigt.

3.3 Orgware

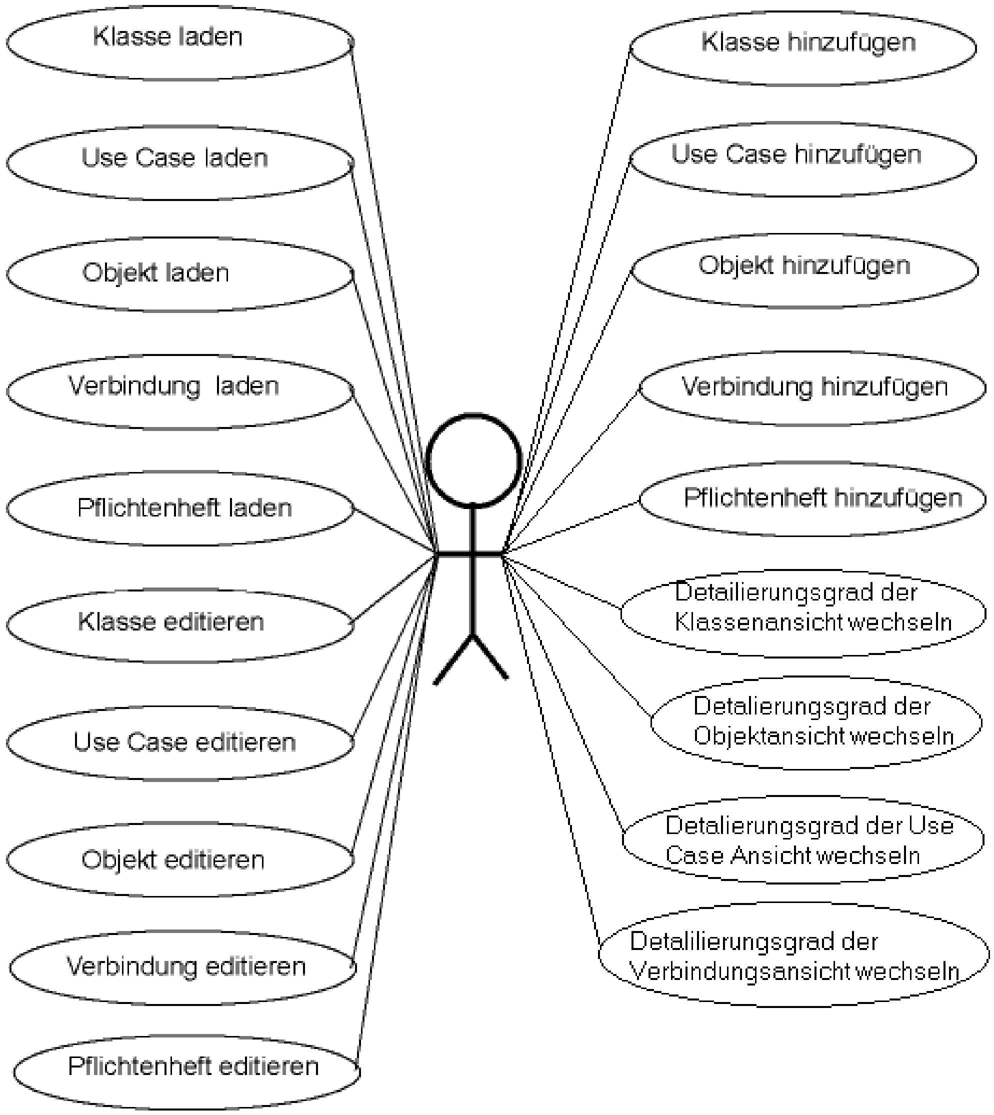
Es wird keine besondere Orgware benötigt.

3.4 Schnittstellen

Es werden keine besonderen Schnittstellen benötigt.

4. Produktfunktionen

4.1 Use Case Diagramm



4.2 Akteure

Die alleinigen Akteure sind die Benutzer des Programms.

4.3 Anwendungsfälle:

Use-Case 1: Klasse hinzufügen

- **Akteure:** Benutzer
- **Vorbedingungen:**keine
- **Nachbedingungen:** Die hinzugefügte Klasse wird angezeigt
- **Ablaufbeschreibung:**
    1. Aufruf des Menüpunktes Klasse --> Klasse hinzufügen
    2. Eine Eingabemaske für Stereotyp, Paket, Name der Klasse, Eigenschaftswerte, Attribute

   und Operationen erscheint. Die eingegeben Attribute bzw. Operationen können einer Attribut- bzw. Operationsliste hinzugefügt werden. Elemente dieser Listen können noch mit Typ, Intialwert und Zusicherung, bzw. Parameter und Zusicherung versehen werden. Desweitern besteht die Wahl zwischen public/private/protected und static/nonstatic
3. Der Benutzer trägt die gewünschten Daten ein und bestätigt diese.
4. Es erscheint eine Maske, die dem Benutzer die Möglichkeit gibt, die Klasse abzuspeichern (Typ: *.kla)
5. Nach der Speicherung erscheint der Anzeigeframe in dem eine graphische Darstellung der hinzugefügten Klasse zu sehen ist. Der Anzeigemodus ist Standard

Use-Case 2: Use-Case hinzufügen

- **Akteure:** Benutzer
- **Vorbedingungen:**keine
- **Nachbedingungen:**Die hinzugefügte Klasse wird angezeigt
- **Ablaufbeschreibung:**
    1. Aufruf des Menüpunktes Use-Case--> Use-Case editieren
    2. Eine Eingabemaske für Nummer, Name, Akteur, Vorbedingung, Nachbedingung, Invariante, Ablaufbeschreibung und Variation erscheint
    3. Der Benutzer kan "erweitert" wählen, dort kan er Nicht funktionale Anforderungen, Ausnahmen, Regeln, Services, Ansprechpartner, Anmerkungen und Dialogbeispiele eintragen, auf Bestätigung werden diese den Daten des Use-Cases hinzugefügt
    4. Der Benutzer trägt die gewünschten Daten ein und bestätigt diese.
    5. Es erscheint eine Maske, die dem Benutzer die Möglichkeit gibt, den Use-Case abzuspeichern (typ: *.use)
    6. Nach der Speicherung erscheint ein Anzeigeframe in dem eine graphische Darstellung des hinzugefügten Use-Cases zu sehen ist. Der Ansichtsmodus ist auf Standard gestellt

Use-Case 3: Objekt hinzufügen

- **Akteure:** Benutzer
- **Vorbedingungen:**keine
- **Nachbedingungen:** Das hinzugefügte Objekt wird angezeigt
- **Ablaufbeschreibung:**
    1. Aufruf des Menüpunktes Objekt --> Objekt hinzufügen
    2. Eine Eingabemaske für Objektname, Objektzugehörige Klasse, und Attribut erscheint. Die eingegebenen Attribute können einer Attributliste hinzugefügt werden. Elemente dieser Liste können mit einem Initialwert versehen werden
    3. Der Benutzer trägt die gewünschten Daten ein und bestätigt diese.
    4. Es erscheint eine Maske, die dem Benutzer die Möglichkeit gibt, das Objekt abzuspeichern (Typ: *.obj)
    5. Nach der Speicherung erscheint ein Anzeigeframe in dem eine graphische Darstellung des UseCases zu sehen ist. Der Anzeigemodus ist Standard

Use-Case 4: Verbindung hinzufügen

- **Akteure:** Benutzer
- **Vorbedingungen:**keine
- **Nachbedingungen:** Die hinzugefügte Verbindung wird angezeigt

- **Ablaufbeschreibung:**
  1. Aufruf des Menüpunktes Verbindung --> Verbindung hinzufügen
  2. Eine Eingabemaske für Name, Stereotyp, Beziehungsname, Zusicherung, Eigenschaftswere, Multiplizität links/rechts erscheint. Es kann eine Leserichtung gewählt werden. Unter "(Pfeil-)spitze kann gewählt werden: Linie durchgezogen/unterbrochen, Pfeilspitze links/rechts/keine, Pfeilspitze offen/dreieckig unausgefüllt/dreieckig ausgefüllt/Raute ausgefüllt/ Raute unausgefüllt
  3. Der Benutzer trägt die gewünschten Daten ein und bestätigt diese.
  4. Es erscheint eine Maske, die dem Benutzer die Möglichkeit gibt, die Verbindung abzuspeichern (Typ: *.ver)
  5. Nach der Speicherung wird die Verbindung im Modus Standard in einem Frame angezeigt

Use-Case 5: Pflichtenheft hinzufügen

- **Akteure:** Benutzer
- **Vorbedingungen:**keine
- **Nachbedingungen:** Falls gewünscht wird das hinzugefügte Pflichtenheft abgespeichert, sonst gehen die eingegebenen Daten verloren
- **Ablaufbeschreibung:**
  1. Aufruf des Menüpunktes Pflichtenheft --> Pflichtenheft hinzufügen
  2. Eine Eingabemaske für die im Pflichtenheft zu tätigenden EInträge erscheint.
  3. Der Benutzer trägt die gewünschten Daten ein. Er kann nun ein Pflichtenheft in HTML erzeugen (dieses wird automatisch unter dem Namen "Plichtenheft.html" abespeichert) oder bestätigen.Auf zweiteres erscheint eine Maske, die dem Benutzer die Möglichkeit gibt, das Pflichtenheft abzuspeichern. Im Gegensatz zu den anderen Fällen gibt es keine vom Programm generierte Anzeige, da diese schon durch den von UTopp generierten HTML code gegeben ist. (Typ: *.pfl)

Use-Case 6-8: Klasse/Objekt/Use-Case laden

- **Akteure:** Benutzer
- **Vorbedingungen:**keine
- **Nachbedingungen:** Es erscheint ein Rahmen in dem das gewählte Element vom Typ Klasse/Objekt/Use-Case dargestellt wird. In diesem kann es bearbeitet werden. Zur Verfügung stehen die Menüpunkte editiern, sowie Modus-->simpel/standard/detailiert bei der Anzeige eines UseCases wird stattdessen eliptisch/standard/nur Ergänzungen zur Auswahl gestellt
- **Ablaufbeschreibung:**
  1. Aufruf des Menüpunktes Klasse/Objekt/Use-Case --> Klasse/Objekt/Use-Case laden
  2. Es erscheint eine Maske, die dem Benutzer die Möglichkeit gibt, ein bereits gespeichertes Element zu laden
  3. Der oben beschriebene Rahmen erscheint

Use-Case 9: Pflichtenheft laden

- **Akteure:** Benutzer
- **Vorbedingungen:**keine
- **Nachbedingungen:** Die in UseCase 5 beschriebene Eingabemaske für das Pflichtenheft mit den gespeicherten Daten ist sicht- und nutzbar

- **Ablaufbeschreibung:**
  1. Aufruf des Menüpunktes Pflichtenheft --> Pflichtenheft laden
  2. Es erscheint eine Maske, die dem Benutzer die Möglichkeit gibt, ein bereits gespeichertes Element zu laden
  3. Die oben beschriebene Maske erscheint

Use-Case 10-12: Klasse/Objekt/Use-Case editieren

- **Akteure:** Benutzer
- **Vorbedingungen:** Ein darstellender Rahmen für Klasse/Objekt/Use-Case existiert
- **Nachbedingungen:** Der Benutzer ist zur Anzeige der/des Klasse/Objekt/Use-Case zurückgekehrt
- **Ablaufbeschreibung:**
  1. Aufruf des Menüpunktes editieren
  2. Die entsprechende Eingabemaske mit den bereits gegebenen Daten erscheint. Diese können nun nach Belieben geändert werden. Auf Bestätigung kann die Änderung abgespeichert werden
  3. Nach der Speicherung kehrt der Benutzer zum Anzeigeframe zurück.

Use-Case 13: Detalierungsgrad der Klassenansicht wechseln

- **Akteure:** Benutzer
- **Vorbedingungen:** Ein darstellender Rahmen für Klasse existiert
- **Nachbedingungen:** Der Benutzer ist zur Anzeige der Klasse zurückgekehrt
- **Ablaufbeschreibung:**
  1. Aufruf des Menüpunktes Modus
  2. 1.Fall: der Benutzer wählt simpel:
  3. 2.Fall: der Benutzer wählt normal:
  4. 3.Fall: der Benutzer wählt detailiert:

Use-Case 14: Detalierungsgrad der Objektansicht wechseln

- **Akteure:** Benutzer
- **Vorbedingungen:** Ein darstellender Rahmen für Objekt existiert
- **Nachbedingungen:** Der Benutzer ist zur Anzeige des Objektes zurückgekehrt
- **Ablaufbeschreibung:**
  1. Aufruf des Menüpunktes Modus
  2. 1.Fall: der Benutzer wählt simpel:
  3. 2.Fall: der Benutzer wählt normal:
  4. 3.Fall: der Benutzer wählt detailiert:

Use-Case 15: Detalierungsgrad der Verbindungsansicht wechseln

- **Akteure:** Benutzer
- **Vorbedingungen:** Ein darstellender Rahmen für Verbindung existiert
- **Nachbedingungen:** Der Benutzer ist zur Anzeige der Verbindung zurückgekehrt
- **Ablaufbeschreibung:**
  1. Aufruf des Menüpunktes Modus
  2. 1.Fall: der Benutzer wählt simpel:
  3. 2.Fall: der Benutzer wählt normal:

4. 3.Fall: der Benutzer wählt detailiert:

Use-Case 16: Detalierungsgrad der UseCaseansicht wechseln

- **Akteure:** Benutzer
- **Vorbedingungen:** Ein darstellender Rahmen für UseCase existiert
- **Nachbedingungen:** Der Benutzer ist zur Anzeige des UseCase zurückgekehrt
- **Ablaufbeschreibung:**
    1. Aufruf des Menüpunktes Modus
    2. 1.Fall: der Benutzer wählt simpel:
    3. 2.Fall: der Benutzer wählt normal:
    4. 3.Fall: der Benutzer wählt detailiert:

5. Produktdaten

5.1 Namen

- Emin Bahdir bahadir@math.uni-muenster.de
- Ralph Carrie carrie@uni-muenster.de
- Verena Kutschker veku@uni-muenster.de
- Marc Rogge mrogge@muenster.de
- Birga Schwartz birgasch@uni-muenster.de
- Karsten Wantia wantiak@math.uni-muenster.de
- Carolin Woltermann carowo@uni-muenster.de

5.2 Titel

C.V.K. Mbert - U-Topp

6. Produktleistungen

Noch nicht ausgearbeitet

7. Benutzeroberfläche

7.1 Struktur der Benutzeroberfläche

Nach dem Start des Programms erscheint eine Menüleiste, der Frame1, die den Ausgangspunkt zu allen anderen Funktionen, die UTop ermöglicht, darstellt. Die darin enthaltenen Meüpunkte sind Klasse, Objekt, Verbindungen, UseCases und Pflichtenheft, die jeweils die Unterpunkte "hinzufügen" und "laden" enthalten. Der Ablauf der weiteren Funktionen ist bei allen diesen Möglichkeiten, das Pflichtenheft ausgenommen, der gleiche: "hinzufügen" lässt eine leere Eingabemaske des jeweils angewählten Menüpunktes erscheinen, mit den Optionsbuttons "abbrechen" und "speichern". Wird "abbrechen" angeklickt, wird die Eingabemaske geschlossen. Bei "speichern" wird die Eingabemaske durch einen Dateimanager ersetzt, worauf, falls wieder bestätigt wird, auch dieser verschwindet und durch eine Anzeige des hinzugefügten Objektes ersetzt wird. Bei "laden" wird ein Dateimanager aufgerufen, um den User die gewünschte Datei auswählen zu lassen. Bei Bestätigung verschwindet der Dateimanager und die Anzeige des gewählten Objektes erscheint. Auf dieser AnzeigenOberfläche exisitiert eine weitere Menüleiste mit den Menüpunkten

"Ansicht" und "Editieren". Ersteres ändert den Detailliertheitsgrad der Anzeige, dh verringert die Anzahl der angezeigten Informationen oder vergrössert sie. "Editieren" lässt die Anzeige verschwinden und ruft dafür die Eingabemaske mit den Daten dieses speziellen Objektes auf. Werden diese durch "Speichern" bestätigt, erscheint die modifizierte Anzeigeoberfläche.

7.2 Screenshots

Noch nicht erstellt

8. Qualitätsziele

Funktionalitätsziele	++			
Zuverlässigkeit	++			
Benutzbarkeit			+-	
Effizienz		+		
Änderbarkeit			+-	

9. Testszenarien

9.0 Testfall: Hinzufügen

/T010/ Abbruch durch den Benutzer.

/T020/ Hinzufügen bestätigen: Der Komponentenname muss unbesetzt sein.

9.1 Testfall: Laden

/T110/ Der Dateimanager wird angezeigt. Es dürfen nur Dateinamen mit dem zur gewählten Komponente gehörenden Dateiende auf der DateiAnzeige erscheinen.

/T120/ Datei anzeigen bestätigen: Eine Datei muss ausgewählt sein.

9.2 Testfall: Editieren

(Testfall: Laden /T110/ Der Dateimanager wird angezeigt. Es dürfen nur Dateinamen mit dem zur gewählten Komponente gehörenden Dateiende auf der DateiAnzeige erscheinen.

/T120/ Datei anzeigen bestätigen: Eine Datei muss ausgewählt sein.)

/T210/ Abbruch durch den Benutzer.

/T220/ Editieren bestätigen:

Der Komponentenname darf von keiner Datei ausser der zu editierenden besetzt sein.

10. Entwicklungsumgebung

1. Software: Symantic VisualCafé wurde zur Erstellung eines Teils der Benutzerdialoge

verwendet, ArgoUML um Diagramme (ZB das Klassendiagramm) zu zeichnen. Für Funktionstest (außerhalb des VisualCafes) wurde JPad bzw. Kawa verwendet. Desweiteren wurde ein Texteditor benutzt.
2. Hardware: Üblicher PC, Unix-Rechner
3. Orgware: Es wurde keine besondere Orgware benötigt
4. Schnittstellen Es wurden keine besonderen Schnittstellen benötigt.

11. Ergänzungen

Noch nicht ausgearbeitet