

# **Belegarbeit im Fach Software Engineering**

## **Sommersemester 2012**

### **Thema:**

**Erarbeitung von unterschiedlichen Modellierungsmethoden einer bestehenden Software am Beispiel des E-Mailverteilers der HTW („Mailman“)**

### **Angefertigt von:**

Thorsten Klein (543372, th.klein@yahoo.de)  
Robert Ledwig (533316, robert.ledwig@gmx.de)  
Max Wahl (533326, maxwahl91@gmail.com)

### **Abgabe:**

15.6.2012

### **Dozent:**

Prof. Dr. Thomas Baar

**Inhaltsverzeichnis:**

- 1. Einführung und Ziel der Arbeit**
- 2. Textuelle Beschreibung**
- 3. Modellierung**
  - 3.1. Use-Case-Modell**
    - 3.1.1. Use-Case-Szenarien (Storys)**
    - 3.1.2. Glossar**
  - 3.2. Domänenmodell**
  - 3.3. Ausgewählte Zustandsdiagramme**
- 4. Erfahrungen aus der Teamarbeit**
- 5. Schlussbetrachtung**

## **1. Einführung**

Heutige Software ist oft ausgesprochen komplex. Mit dem Siegeszug des Computers und immer leistungstärkerer Hardware ist es Softwareentwicklern heute möglich, schier unendlich aufwendige Programme zu schreiben. Je größer ein Projekt wird, desto zeitaufwändiger und somit teurer wird die Entwicklung. Unüberlegt zusammengeschriebene Software wird dadurch ab einer gewissen Größe unübersichtlich und somit zum Risiko für andere beteiligte Programmierer oder gar den User.

Um dem Endprodukt von Anfang an eine sinnvolle Struktur zu geben und somit Zeit und Geld zu sparen, versucht deshalb die Wissenschaft des „Software Engineering“, mit Hilfe verschiedener Vorgehensweisen und Modellen, die Gestaltung und den Ablauf bestimmter Teilprozesse der zukünftigen Software im Vorfeld zu beschreiben.

Ziel dieser Arbeit soll es sein, diesen Beschreibungs- und Modellierungsprozess nachzuvollziehen und selbst für eine bestehende Software umzusetzen.

Unsere Wahl fiel in die diesem Zusammenhang auf die E-Mailverteilungssoftware der HTW Berlin. Mit Hilfe dieser über das Internet erreichbaren Applikation ist es jedem Studenten, Dozent sowie anderen Mitarbeitern möglich, E-Mail-Listen anzulegen, zu verwalten und zu nutzen. Wir wählten das Programm, da es in seinen Funktionen leicht verständlich ist (auch ohne den Quellcode zu kennen) und somit eine ideale Modellierungs- und Abstraktionsgrundlage darstellt. Zu Testzwecken wurden sämtliche Funktionen der Liste mit einer bereits bestehenden Liste (ce-sose@lists.htw-berlin.de, die Liste der Computer-Ingenieure des SS 11) getestet.

## 2. Textuelle Beschreibung

### Nutzen:

Das zu beschreibende System dient als Verteiler von E-Mails an diverse Adressen, die unter einer gemeinsamen Adresse gebündelt werden. Somit wird es überflüssig 50-mal dieselbe E-Mail zu verschicken, oder lange Adressbücher zu pflegen.

In der Praxis heißt das, ein Anwender schickt eine Nachricht an die Gemeinschaftsadresse und diese wird an alle Mitglieder der Liste weitergeleitet.

Nachrichten von Nicht-Mitgliedern werden an den Admin geleitet, der daraufhin entscheiden kann, ob die Nachricht zugestellt oder verworfen werden soll. Der Absender bekommt eine Nachricht, dass die Mail bearbeitet wird und das Angebot, die Nachricht zurückzuziehen. Der Admin kann die Nachricht dann nicht mehr veröffentlichen.

Jeder Anwender kann eine Liste abonnieren (mit Zustimmung des jeweiligen Admins). Er kann sich außerdem selbst austragen (bedarf keiner Zustimmung) und eigene Einstellungen (z.B. über Inhalt und Darstellungsform der Mails, die er von den Listen erhält) festlegen.

Der Administrator kann mehrere User austragen und beliebig neue User hinzufügen. Des Weiteren legt er Einstellungen fest, die z.B. Archivierung, Passwörterinnerungen oder Filter betreffen.

Zusätzlich gibt es sogenannte Moderatoren, die zwar Anwender hinzufügen und löschen dürfen, aber keine Details des Mailvertailers verändern dürfen.

Allen Listen übergeordnet steht das Rechenzentrum der HTW, die alle Listen verwalten und einsehen können. Diese Instanz wird in unseren Modellierungen nicht betrachtet. Weiterhin haben wir uns dazu entschlossen, nur eine Mailingliste zu modellieren, in der Realität sind es beliebig viele.

### Hardware-Voraussetzungen:

Ein internetfähiges Gerät, Mailprovider



### 3.1.1 Use-Case-Szenarien (Storys)

Nachfolgend die textuellen Teile der Use-Cases.

**Use Case:** Verteiler-Mail an Liste versenden

**Kurzbeschreibung:** Eine E-Mail wird von einem Mailclient an die Liste gesendet.

**Primärer Aktor:** Mailclient

**Erfolgsszenario:**

1. Mailclient versendet Mail an Mailinglisten-Server
2. System validiert Zugehörigkeit des Versenders zur Liste
3. Weiter mit Use-Case „Mail weiterleiten“
4. Use-Case endet erfolgreich

**Erweiterungen:**

2.a. System stellt fest, dass der Versender nicht der Liste angehört:

2.a.1 System schickt Moderationsaufforderung (und empfangene E-Mail) an Administrator  
gesendet

2.a.2 System schickt Absender eine Mitteilung, dass E-Mail bearbeitet wird

2.a.3 Admin bestätigt das Versenden innerhalb von 3 Tagen

2.a.4 Fortsetzung bei 3.

2.a.3.a Admin bestätigt das Versenden nicht innerhalb von 3 Tagen:

2.a.3.a.1 System löscht die Anfrage

2.a.3.a.2 System informiert Absender, dass die E-Mail nicht gesendet wurde

2.a.3.a.3 Use-Case endet erfolglos

2.a.3.b Absender zieht seine Anfrage zurück:

2.a.3.b.1 System löscht Anfrage

2.a.3.b.2 Use-Case endet erfolglos

2.a.3.c Admin verweigert das Versenden innerhalb von 3 Tagen:

2.a.3.c.1 System löscht die Anfrage

2.a.3.c.2 System informiert Absender, dass die E-Mail nicht gesendet wurde

2.a.3.c.3 Use-Case endet erfolglos

**Use-Case:** Mail weiterleiten

**Kurzbeschreibung:** Eine Mail wird an alle Mitglieder der Liste gesendet.

**Primärer Akteur:** System (Mailverteiler)

**Erfolgsszenario:**

1. System schickt Mail an alle Mitglieder der Liste
2. Use-Case endet erfolgreich

Erweiterungen:

1.a Das Überprüfen der Größe ist aktiviert:

1.a.1 Mail hat eine erlaubte Größe:

1.a.1.1 Mail wird an Verteiler-User geschickt

1.a.1.2 Fortsetzung bei 2.

1.a.1.a Mail ist zu groß:

1.a.1.a.1 Use-Case endet erfolglos

**Use-Case:** Liste abonnieren

**Kurzbeschreibung:** Die Mailadresse eines externen Users soll in die Liste aufgenommen werden.

**Primärer Akteur:** Mailclient-User

**Erfolgsszenario:**

1. User öffnet zentrale Seite der Mailverteiler
2. User wählt Liste aus
3. User gibt seine Mail-Adresse ein
4. System sendet E-Mail an Administrator und bittet um Bestätigung der neuen Adresse
5. System sendet Bestätigungs-Mail an User
6. Administrator bestätigt neue Adresse innerhalb von drei Tagen
7. User sendet korrekten Code der Bestätigungs-Mail innerhalb von drei Tagen an System
8. Mailadresse des Users wird in Liste aufgenommen
9. Use-Case endet erfolgreich

*Hinweis:*

5. Und 7. Können in beliebiger Reihenfolge, oder Zeitgleich ablaufen

Erweiterungen:

- 6.a Admin bestätigt neue Adresse nicht innerhalb von drei Tagen:
  - 6.a.1 Use-Case endet erfolglos
- 6.b Admin lehnt Bestätigung ab:
  - 6.b.1 Use-Case endet erfolglos
- 7.a User sendet korrekten Code nicht innerhalb von drei Tagen an System zurück:
  - 7.a.1 Use-Case endet erfolglos
- 9.a Das Versenden einer Begrüßungsmail ist aktiviert:
  - 9.a.1 weiter mit Use-Case „Interne Mail versenden“
  - 9.a.2 Use-Case endet erfolgreich

**Use-Case:** Interne Mail versenden

**Kurzbeschreibung:** Das System versendet eine interne Mail an alle User.

**Primärer Akteur:** System (Mailverteiler)

**Vorbedingung:** Das Versenden einer internen Mail ist eingestellt.

**Erfolgsszenario:**

- 1. System generiert automatische Mail
- 2. Weiter mit Use-Case „Mail weiterleiten“
- 3. Use-Case endet erfolgreich

**Use-Case:** Abo kündigen

**Kurzbeschreibung:** Ein User möchte seine Mailadresse aus der Liste austragen.

**Primärer Akteur:** Verteiler-User

- 1. User besucht online die zentrale Schnittstelle seiner Liste
- 2. User meldet sich mit seiner Mail-Adresse an
- 3. User bestätigt die Kündigung
- 4. Das System entfernt die Mailadresse aus der Liste
- 5. Use-Case endet erfolgreich

Erweiterungen:

- 5.a Das Versenden einer Abschiedsmail ist aktiviert:
  - 5.a.1 weiter mit Use-Case „interne Mail versenden“



#### 5.a.2 Use-Case endet erfolgreich

**Use-Case:** Moderationsanfragen bearbeiten

**Kurzbeschreibung:** Das Versenden einer Mail von einem Nichtmitglied soll bestätigt werden.

**Primärer Aktor:** Moderator

1. System meldet dem Admin per E-Mail einen neuen Moderationsauftrag
2. Admin öffnet die zentrale Administrationsschnittstelle seiner Liste
3. Admin meldet sich mit seinem Passwort an
4. System validiert die Zugangsdaten
5. Admin erlaubt das Versenden einer Nachricht eines externen Versenders
6. Use-Case endet erfolgreich

Erweiterungen:

- 4.a System stellt fest, dass das Passwort falsch ist:
  - 4.a.1 System meldet dem Admin die falsche Eingabe
  - 4.a.2 System bittet um erneute Eingabe des Passworts
  - 4.a.3 Fortsetzung bei 3.

### **3.1.2 Glossar**

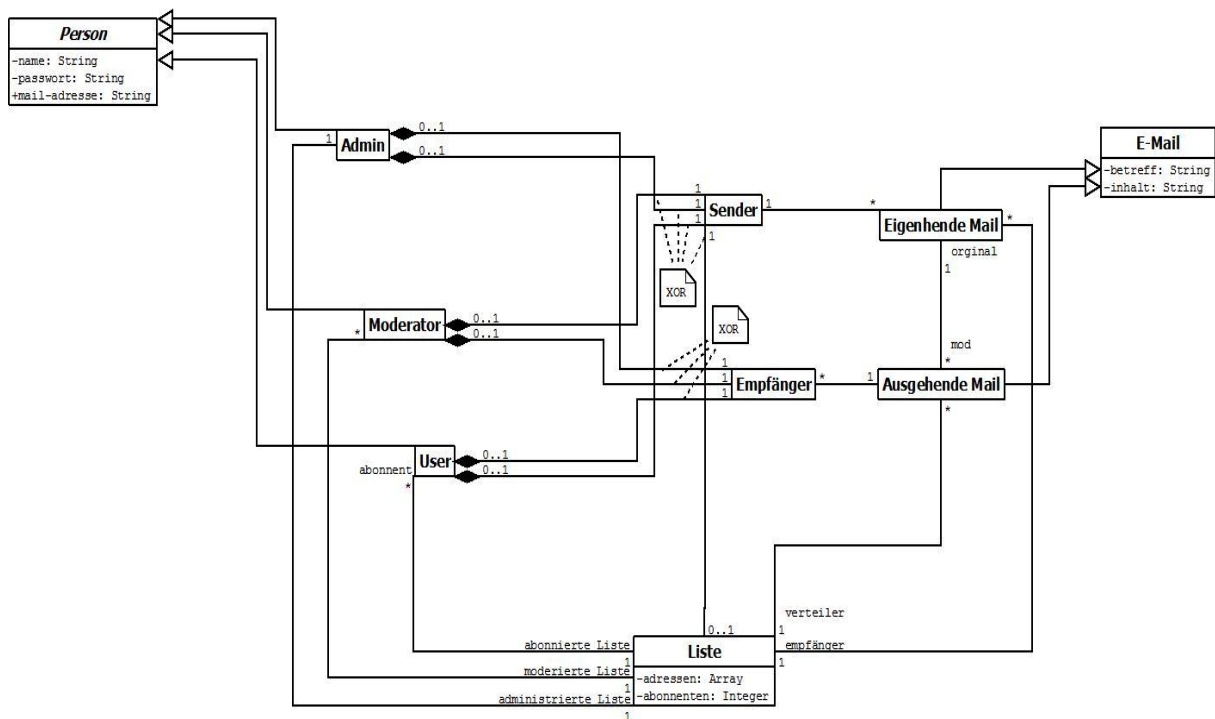
Das Glossar dient der Begriffsklärung und ist als Hilfsmittel zu verstehen, um Fachbegriffe und Kunstworte aus Use-Case-Diagramm und Storys zu beschreiben. Alle hier definierten Begriffe gelten auch für alle anderen Modellierungsformen, die in dieser Arbeit vorgestellt werden.

Aufgrund der von uns gewählten Modellierungen haben wir uns entschlossen, keine Begriffe ins Glossar zu übernehmen. Die Erklärung des Begriffs „E-Mail“ erschien uns für unsere Software beispielsweise als nicht notwendig.

### 3.2 Domänenmodell

In einem Domänenmodell werden die einzelnen Klassen, deren Eigenschaften, sowie die komplexen Beziehungen zwischen deren Instanzen in UML-Notation beschrieben. Es ist eine leistungsstarke Technik, sich frühzeitig (also vor der Implementierung) mit der Problem-Domäne vertraut zu machen und ein umfassendes System mit einfachen Mitteln zu beschreiben. In einem Domänenmodell tauchen noch keine Aktionen der Instanzen auf, nur deren Beziehungen.

In der folgenden Abbildung ist unser Entwurf eines Domänenmodells zu sehen.



Unser Domänenmodell zeigt insgesamt zehn Klassen. Die bereits auf vorgehenden Seiten beschriebenen Akteure User, Admin und Moderator sind hier als jeweils eine Klasse abgebildet. Alle drei erben von der Vaterklasse Person. Jede von ihnen (aber immer nur eine => XOR) kann als Sender eine E-Mail verschicken, die als eingehende Mail der Liste (in der Rolle Empfänger) zugestellt wird. Die Liste leitet diese Mail dann weiter, indem sie aus der eingehenden Original-Mail eine modifizierte Mail erstellt und an den oder die Empfänger schickt. Das sind wiederum User, Moderatoren und/oder der Admin.

Jeder Liste ist desweiteren genau ein Admin, aber beliebig viele Moderatoren und User zugeordnet.

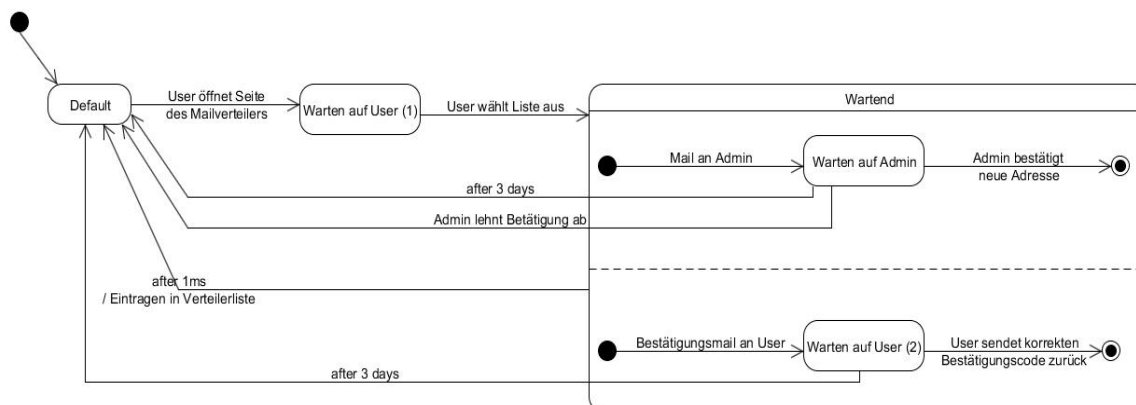
Sollen automatisierte E-Mails verschickt werden, so fungiert die Liste selbst als Sender.

### 3.3 Ausgewählte Zustandsdiagramme

Um das Verhalten einer Software zur Laufzeit zu modellieren, sowie Ein- und Ausgabeverhalten zu untersuchen, bedient man sich sogenannter Zustandsdiagramme. Dieses System ähnelt in vielen Punkten der Beschreibung eines Automaten, der von einem zum anderen Zustand wechselt, sobald ein bestimmtes Ereignis auftritt.

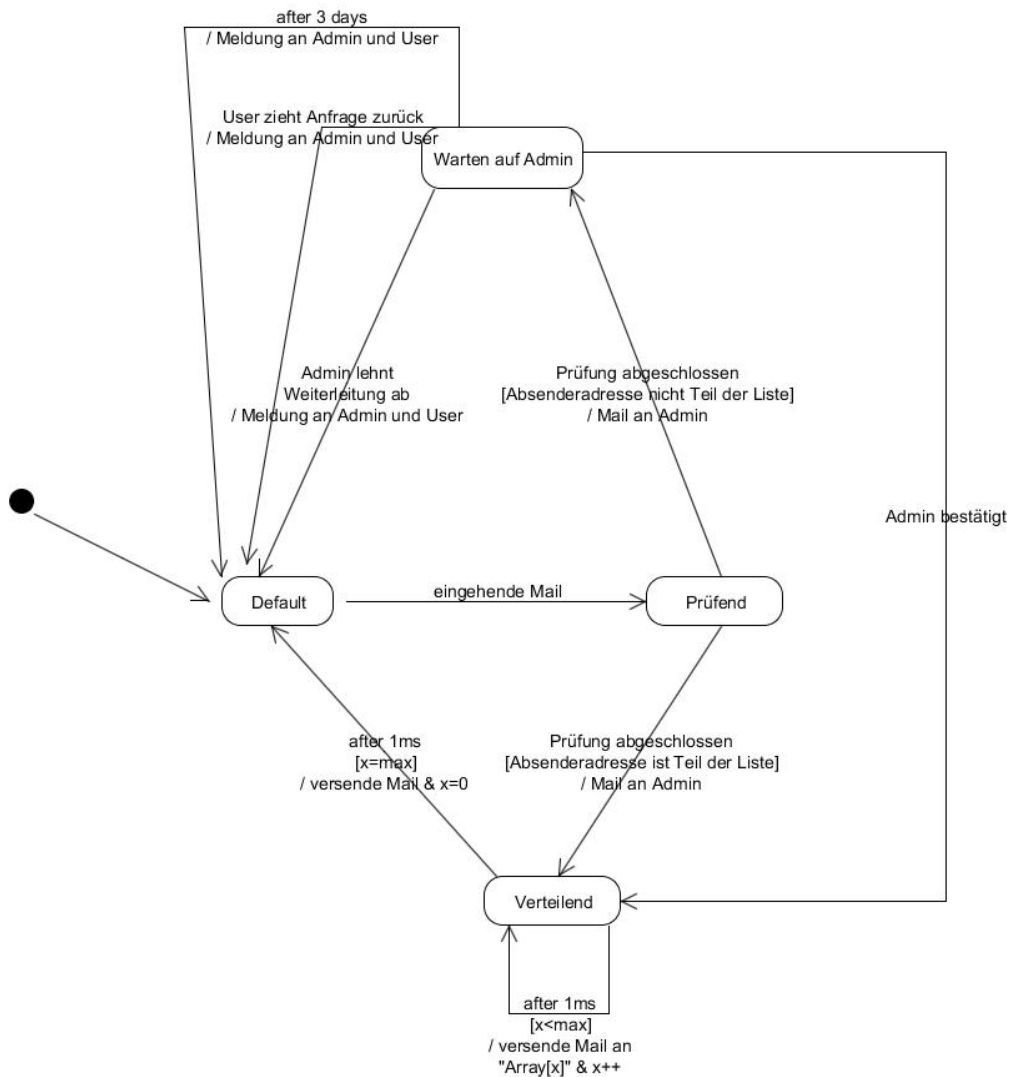
Im Rahmen unseres Projektes haben wir uns entschlossen, beispielhaft die Use-Cases „Liste abonnieren“ und „Mail an Verteiler versenden“ darzustellen. Darin enthalten sind außerdem die Use-Cases „Interne Mail versenden“ und „Moderationsanfrage bearbeiten“.

#### Liste abonnieren:



Der Initialzustand ist der „Defaultzustand“. Der User kann nun über die Webseite des Mailverteilers sich eine Liste aussuchen, in die er aufgenommen werden möchte. Hat er sich für eine Liste entschieden, wird eine Benachrichtigungsmail an den Admin und eine Bestätigungsmail an den User geschickt. Wenn der Admin die neue Email-Adresse bestätigt und der User den korrekten Bestätigungscode zurücksendet, wird der User mit seiner neuen Email-Adresse in die Liste aufgenommen. Erfolgt eine der Bestätigungen nicht innerhalb von 3 Tagen, erfolgt keine Aufnahme in die Liste. Alternativ kann der Admin die Aufnahme in die Liste auch direkt ablehnen.

## Mail an Verteiler versenden:



Der Initialzustand ist der „Defaultzustand“. Der Mailclient versendet eine Mail an den Mailinglisten-Server. Diese Mail wird anschließend auf Zugehörigkeit zur Liste überprüft. Ist die Zugehörigkeit gegeben, wird sie an alle Mitglieder der Liste verschickt. Ist die Zugehörigkeit nicht gegeben, kann der Admin entweder die Mail zum Verteilen freigeben oder eine Weiterleitung ablehnen. Andernfalls kann auch der User seine Anfrage zurückziehen. Wurde die Mail nicht versandt, erhalten User und Admin eine entsprechende Nachricht.

### *Anmerkungen:*

„x“ ist eine Prüfvariable, die mit jeder versendeten Mail inkrementiert wird. Als Index des Arrays, in dem alle Adressen gespeichert sind, ermöglicht sie, alle User der Liste zu erreichen.

„max“ sei die Anzahl aller gespeicherten Mail-Adressen (die Größe des Arrays).

#### **4. Erfahrungen aus der Teamarbeit**

Rückwirkend betrachtet kann die Teamarbeit insgesamt als positiv bewertet werden! Da wir selbst eine E-Mail-Verteiler-Liste nutzen konnten, um alle Funktionen als Admin und User auszuprobieren, konnten sämtliche Funktionen problemlos erarbeitet werden und in die entsprechenden Diagramme überführt werden. Die Mailverteilungssoftware der HTW wartet nicht mit übermäßigen Funktionen auf, ist aber übersichtlich und durchdacht. Sie stellt eine zeitgemäße Art der Kommunikation dar, ist selbsterklärend und auch für den Laien schnell kontrollierbar. Einzig die Position des Rechenzentrums, welches als eine Art „Über-Admin“ fungiert konnten wir nicht gut veranschaulichen.

Zunächst wurde das Programm von allen Teammitgliedern studiert und getestet. Anschließend wurde die textuelle Beschreibung gemeinsam erarbeitet. Daraufhin übernahm ein Kommilitone die Use-Case-Diagramme, einer das Domänenmodell und einer die Zustandsdiagramme. Anschließend wurden die Ideen gemeinsam ausgewertet und aneinander angepasst. Die Zusammenarbeit klappte dabei sehr gut, da einzelne Aufgabenbereiche ineinander übergingen und so ein reger Austausch stattfand.

## **5. Schlussbetrachtung**

Abschließend kann das Ziel dieser Arbeit, unterschiedliche Modellierungsmethoden kennen zu lernen, als erfüllt betrachtet werden. In relativ kurzer Zeit entstanden komplexe Darstellungsformen nahezu aller Bereiche unserer Software und ermöglichten uns so einen Einblick in ein Thema, dass uns so (trotz bereits umfangreicher Kenntnisse in der OOP) nicht geläufig war und sicherlich zukünftig zu einem festen Bestandteil in eigenen Softwareprojekten werden wird.