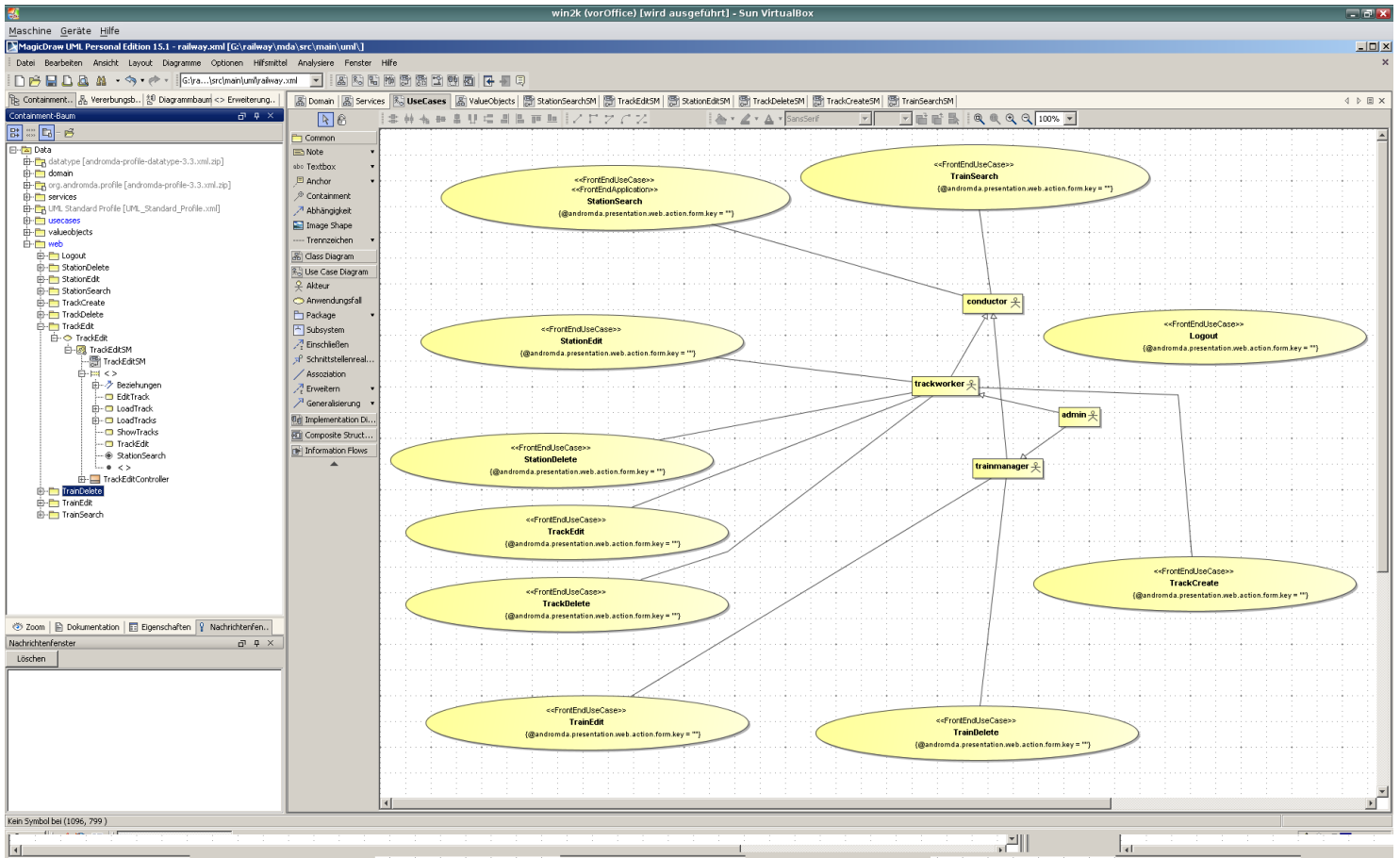


# Modellgetriebene Softwareentwicklung - Blatt 10

Daniel Aigner, Sebastian Hösel, Daniel Gibson

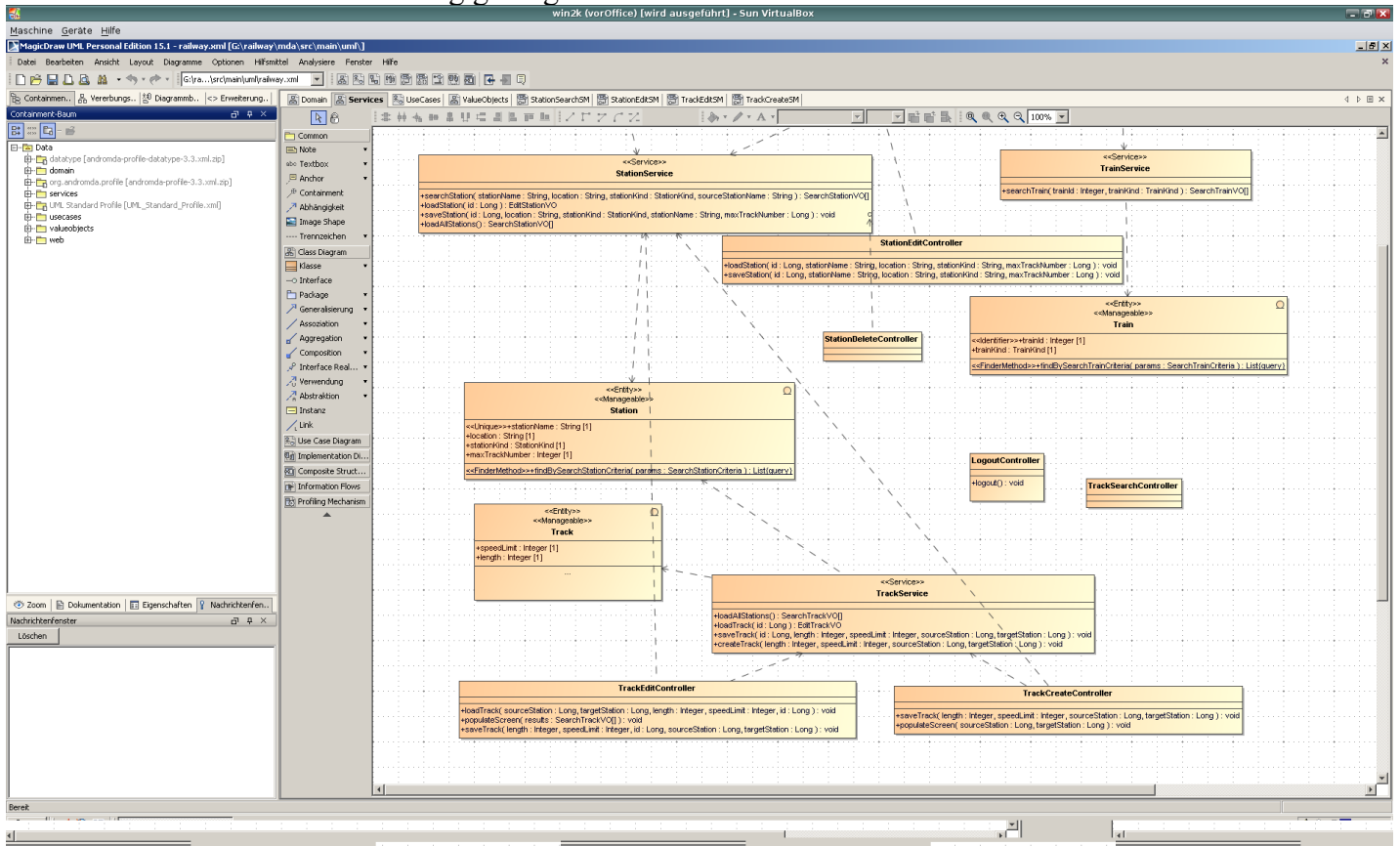
Zunächst haben wir einen Anwendungsfall „TrackEdit“ angelegt und der trackworker-Rolle zugewiesen.

Dann haben wir das entsprechende Package in Data/web/ angelegt, darin einen Anwendungsfall angelegt, in den wir eine Zustandsmaschine mit Zustandsautomaten und Controller-Klasse gepackt haben.

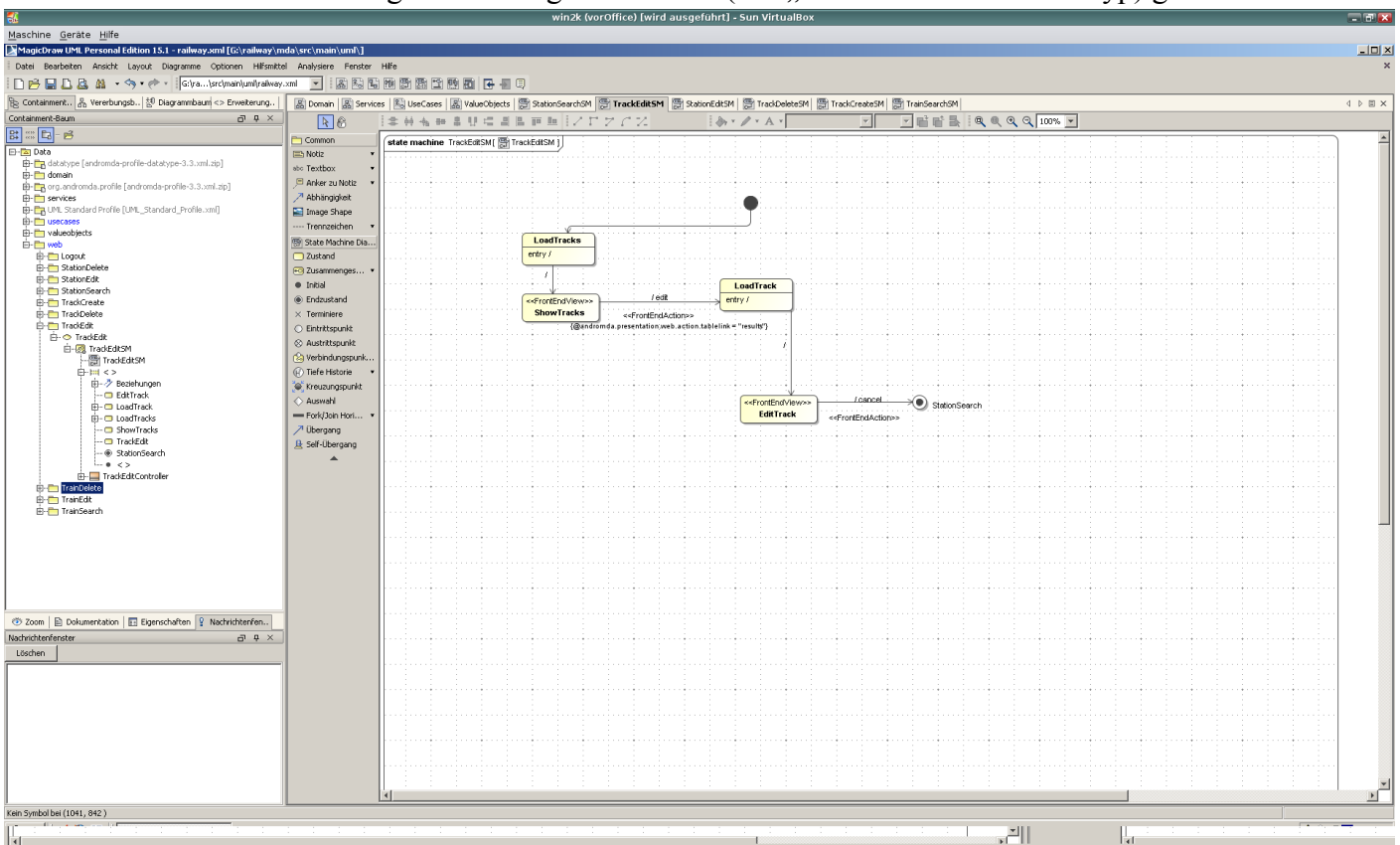


Danach haben wir Value Objects angelegt.

Dann haben wir zum Laden der Stationen einen TrackService mit einer entsprechenden Methode (loadAllStations) angelegt und mit Track verknüpft. Mit diesem Service haben wir dann unseren TrackEditController in Abhängigkeit gesetzt.



Schließlich haben wir in den Zustandsautomaten die nötigen Zustände eingefügt und mittels Aktivitäten in den fürs Daten Laden zuständigen Zuständen und Effekten in den Transitionen die Daten in die fürs Anzeigen zuständigen Zustände (mit „FrontEndView“-Stereotyp) geleitet.



win2k (vorOffice) [wird ausgeführt] - Sun VirtualBox

MagicDraw UML Personal Edition 15.1 - railway.uml [C:\railway\mdu\_src\main.uml]

state machine TracEditSM TracEditSM

```

stateDiagram-v2
    [*] --> entry
    state entry
    state state {
        ShowTracks
        LoadTracks
    }
    entry --> state
    state --> state : edit
  
```

Parameter (results: SearchTrackVO[]) [web:TracEdit:TracEditSM:...]
 

Name	Typ	Defaultwert	Richtung
results	SearchTrackVO[]		in

win2k (vorOffice) [wird ausgeführt] - Sun VirtualBox

MagicDraw UML Personal Edition 15.1 - railway.uml [C:\railway\mdu\_src\main.uml]

state machine TracEditSM TracEditSM

```

stateDiagram-v2
    [*] --> entry
    state entry
    state state {
        ShowTracks
        LoadTracks
    }
    entry --> state
    state --> state : edit
    state --> LoadTrack : call
  
```

Call Operation Aktion - loadTrack

Name	Typ	Defaultwert	Richtung
loadTrack	SearchTrackVO[]		in

loadTrack

Call Operation Aktion

Name	Typ	Defaultwert	Richtung
loadTrack	SearchTrackVO[]		in

Operation

is synchron

Altiver Hyperlink

Zusicherungen

To Do

Name

The name of the NamedElement.

Jetzt mussten noch die Implementierungen des Services und des Controllers von Hand angepasst werden:

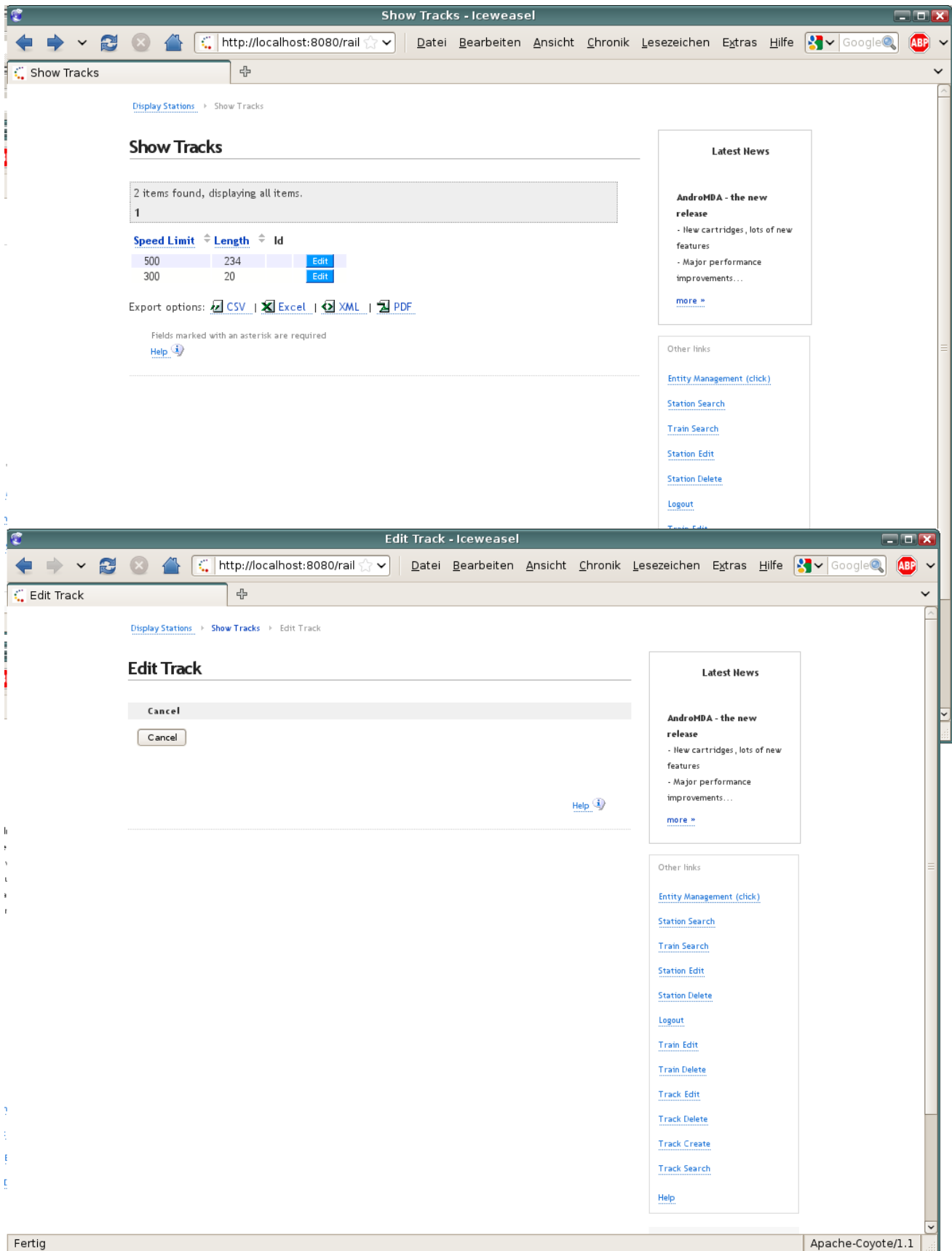
```
TrackServiceImpl.java - SciTE
File Edit Search View Tools Options Language Buffers Help
TrackServiceImpl.java TrackEditControllerImpl.java TrackCreateControllerImpl.java
/opt/MDD/railway/core/src/main/java/services/TrackServiceImpl.java

24  * @see services.TrackService
25  */
26  public class TrackServiceImpl
27  extends services.TrackServiceBase
28  {
29
30      /**
31       * @see services.TrackService#loadAllStations()
32       */
33      protected valueobjects.SearchTrackVO[] handleLoadAllStations()
34      throws java.lang.Exception
35      {
36
37          Collection<TrackDao> results = getTrackDao().loadAll(TrackDao.TRANSFORM_SEARCHTRACKVO);
38          SearchTrackVO[] tracks = new SearchTrackVO[results.size()];
39          return (SearchTrackVO[]) results.toArray(tracks);
40      }
41
42
43      @Override
44      protected EditTrackVO handleLoadTrack(Long id) throws Exception {
45          Track track = getTrackDao().load(id);
46          EditTrackVO editTrackVO = new EditTrackVO(track.getSpeedLimit(), track.getLength(), track.getId(), track.getSourceStation())
47          return editTrackVO;
48      }
49
50      @Override
51      protected void handleSaveTrack(Long id, Integer length, Integer speedLimit,
52          Long sourceStation, Long targetStation) throws Exception {
53          Track track = getTrackDao().load(id);
54          track.setId(id);
55          track.setLength(length);
56          track.setSpeedLimit(speedLimit);
57          track.setSourceStation(getStationDao().load(sourceStation));
58          track.setTargetStation(getStationDao().load(targetStation));
59          getTrackDao().update(track);
60      }
61
62      @Override
63      protected void handleCreateTrack(Integer length, Integer speedLimit,
64          Long sourceStation, Long targetStation) throws Exception {
65          getTrackDao().create(length, getStationDao().load(sourceStation), speedLimit, getStationDao().load(targetStation));
66      }
67
68  }
```

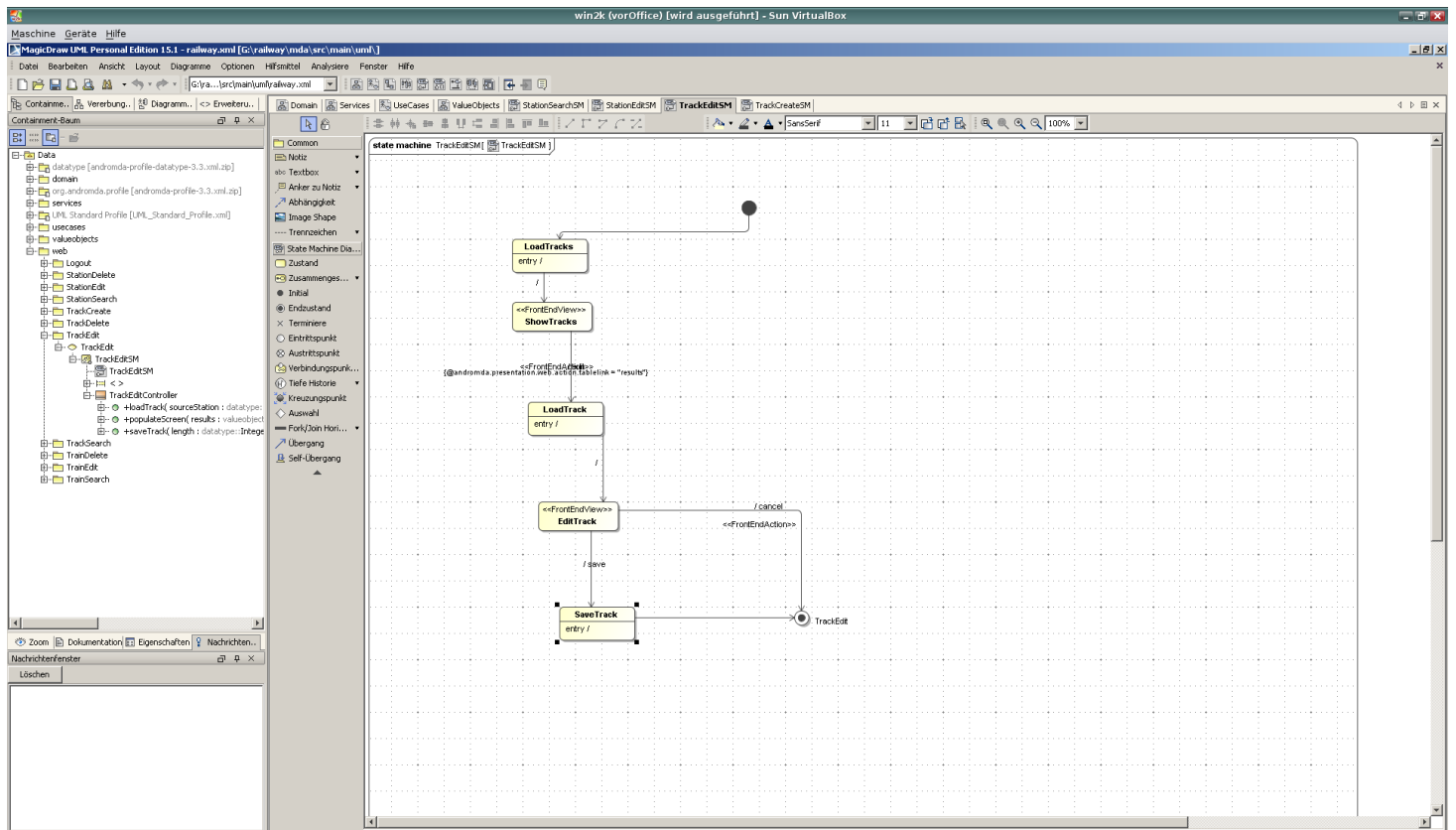
```
TrackEditControllerImpl.java - SciTE
File Edit Search View Tools Options Language Buffers Help
TrackServiceImpl.java TrackEditControllerImpl.java TrackCreateControllerImpl.java
/opt/MDD/railway/web/src/main/java/web/TrackEdit/TrackEditControllerImpl.java

16  /**
17  * @see web.TrackEdit.TrackEditController
18  */
19  public class TrackEditControllerImpl extends TrackEditController
20  {
21
22      /**
23       * @see web.TrackEdit.TrackEditController#loadTrack(org.apache.struts.action.ActionMapping, web.TrackEdit.LoadTrackForm,
24       javax.servlet.http.HttpServletRequest, javax.servlet.http.HttpServletResponse)
25       */
26      public final void loadTrack(ActionMapping mapping, web.TrackEdit.LoadTrackForm form, HttpServletRequest request,
27          HttpServletResponse response) throws Exception
28      {
29          EditTrackVO trackVO = this.getTrackService().loadTrack(form.getId());
30          // this property receives a default value, just to have the application running on dummy data
31          form.setSpeedLimit(trackVO.getSpeedLimit());
32          // this property receives a default value, just to have the application running on dummy data
33          form.setTargetStation(trackVO.getTargetStation());
34          // this property receives a default value, just to have the application running on dummy data
35          form.setSourceStation(trackVO.getSourceStation());
36          // this property receives a default value, just to have the application running on dummy data
37          form.setLength(trackVO.getLength());
38      }
39
40      /**
41       * @see web.TrackEdit.TrackEditController#populateScreen(org.apache.struts.action.ActionMapping,
42       web.TrackEdit.PopulateScreenForm, javax.servlet.http.HttpServletRequest, javax.servlet.http.HttpServletResponse)
43       */
44      public final void populateScreen(ActionMapping mapping, web.TrackEdit.PopulateScreenForm form, HttpServletRequest request,
45          HttpServletResponse response) throws Exception
46      {
47          form.setResults(getTrackService().loadAllStations());
48      }
49
50      public void saveTrack(ActionMapping mapping, SaveTrackForm form,
51          HttpServletRequest request, HttpServletResponse response)
52          throws Exception {
53          // TODO Auto-generated method stub
54          this.getTrackService().saveTrack(form.getId(), form.getLength(), form.getSpeedLimit(), form.getSourceStation(), form.
55          getTargetStation());
56      }
57
58  }
```

Nun hatten wir „schon“ eine lauffähige Version, in der man rechts auf „Track Edit“ klicken konnte, wonach einem eine Liste der vorhandenen Verbindungen angezeigt wurde inkl. Edit-Button, der einen dann auf eine (noch nicht funktionierende) Seite zum Editieren einer Verbindung bringt:

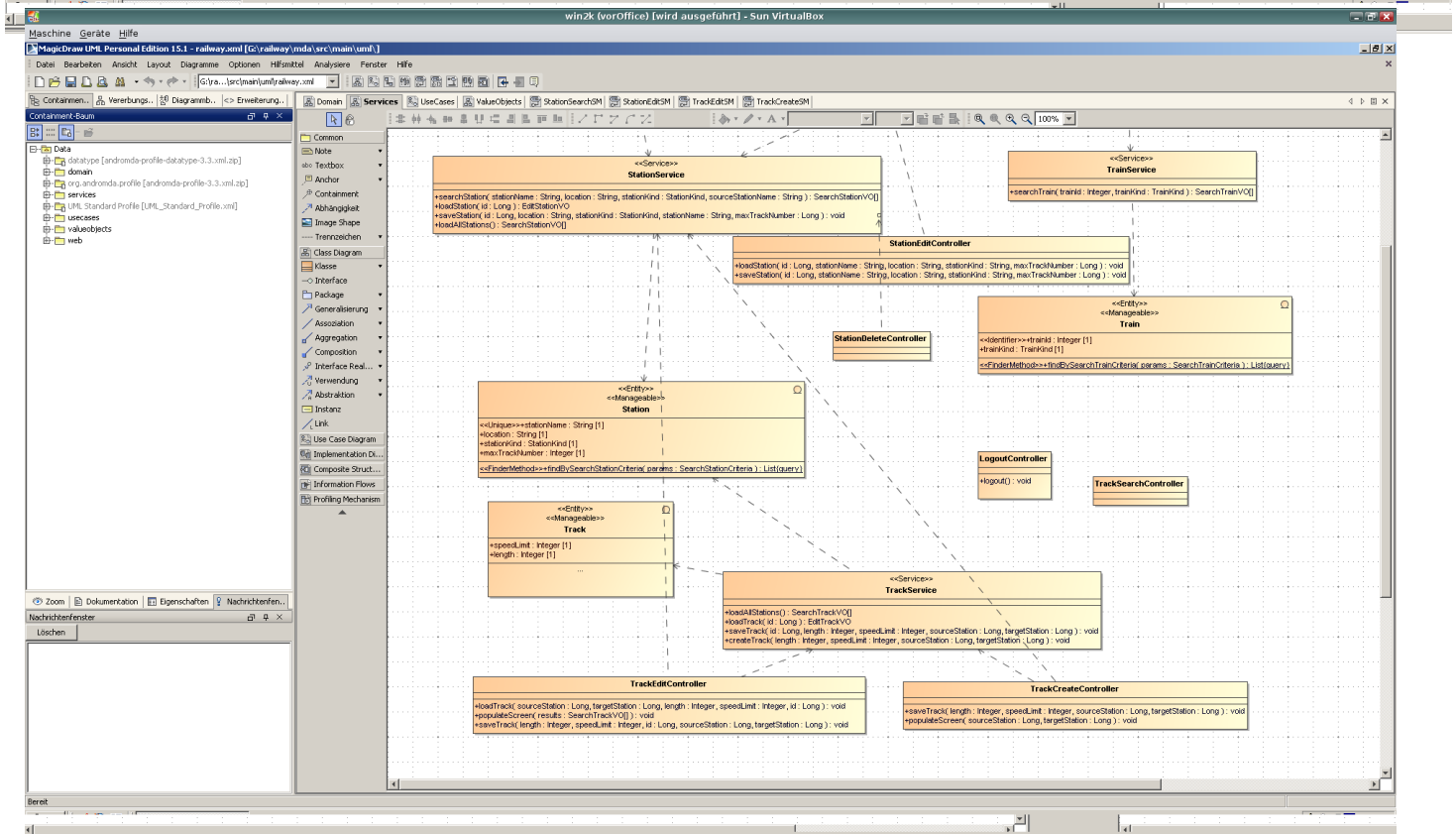
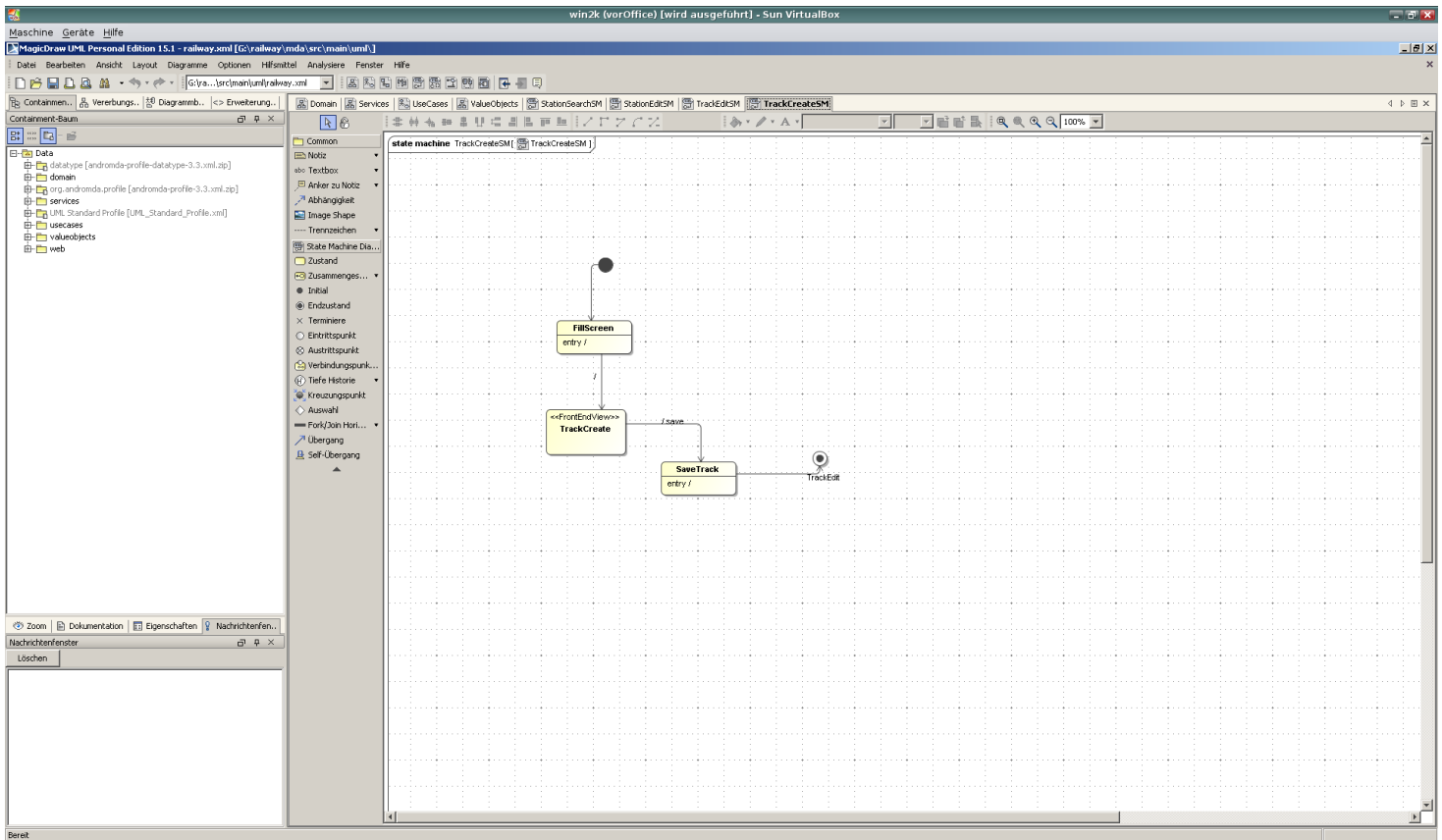


Mit einer weiteren Methode im Controller zum Laden einer bestimmten Verbindung (loadTrack()) und einer angepassten Transition zwischen dem LoadTrack- und dem EditTrack-Zustand werden nun, nachdem man auf „Edit“ geklickt hat, die Eigenschaften einer Verbindung angezeigt, sodass man sie verändern und mittels der saveTrack-Methode (ebenfalls aus dem Controller) speichern kann.



The screenshot shows the 'Edit Track' web application interface. The browser address bar shows 'http://localhost:8080/rail'. The page title is 'Edit Track'. The breadcrumb navigation is 'Display Stations > Track Create > Show Tracks > Edit Track'. The main content area has a 'Save' section with the following fields: Length (234), Speed Limit (500), Id (1), Source Station (Marburg), and Target Station (Gießh?en). There are 'Save' and 'Cancel' buttons. Below the form, it says 'Fields marked with an asterisk are required' and has a 'Help' link. On the right side, there is a 'Latest News' section with a link to 'AndromDA - the new release' and a list of features. Below that is an 'Other links' section with various navigation links like 'Entity Management', 'Station Search', 'Train Search', etc. The footer shows 'Fertig' and 'Apache-Coyote/1.1'.

Jetzt fehlte uns noch der Anwendungsfall zum erstellen einer neuen Verbindung:  
 Nach bekanntem Schema haben wir einen Anwendungsfall gebaut und dem zugehörigen Zustandsautomaten gemalt.  
 Zum anlegen neuer Verbindungen wurde im TrackCreateController eine saveTrack-Methode notwendig, der aus den ihm übergebenen Daten eine neue Verbindung erstellt. Er ist vom TrackService abhängig, der auch eine entsprechende Methode beinhalten muss (gut, dass man mit Modellgetriebener Softwareentwicklung Arbeit und Redundanzen vermeiden kann).  
 Der TrackCreateController musste außerdem von Hand erweitert werden.



```
TrackCreateControllerImpl.java - SciTE
File Edit Search View Tools Options Language Buffers Help
/opt/MDD/railway/web/src/main/java/web/TrackCreate/TrackCreateControllerImpl.java
1 // license-header java merge-point
2 package web.TrackCreate;
3
4 import org.apache.struts.action.ActionMapping;
5
6 import javax.servlet.http.HttpServletRequest;
7 import javax.servlet.http.HttpServletResponse;
8
9 /**
10  * @see web.TrackCreate.TrackCreateController
11  */
12 public class TrackCreateControllerImpl extends TrackCreateController
13 {
14     /**
15      * @see web.TrackCreate.TrackCreateController#saveTrack(org.apache.struts.action.ActionMapping,
16      web.TrackCreate.SaveTrackForm, javax.servlet.http.HttpServletRequest, javax.servlet.http.HttpServletResponse)
17      */
18     public final void saveTrack(ActionMapping mapping, web.TrackCreate.SaveTrackForm form, HttpServletRequest
19 request, HttpServletResponse response) throws Exception
20 {
21     this.getTrackService().createTrack(form.getLength(), form.getSpeedLimit(), form.getSourceStation(), form.
22 getTargetStation());
23 }
24 }
```

Track Create - Iceweasel

http://localhost:8080/rail

Track Create

Display Stations > Track Create > Show Tracks > Edit Track > Track Create

### Track Create

**Save**

Length

Speed Limit

Target Station

Source Station

Fields marked with an asterisk are required

[Help](#)

#### Latest News

**AndromDA - the new release**

- New cartridges, lots of new features
- Major performance improvements...

[more >>](#)

#### Other links

- [Entity Management \(click\)](#)
- [Station Search](#)
- [Train Search](#)
- [Station Edit](#)
- [Station Delete](#)
- [Logout](#)
- [Train Edit](#)
- [Train Delete](#)
- [Track Edit](#)
- [Track Delete](#)
- [Track Create](#)
- [Track Search](#)
- [Help](#)

Fertig Apache-Coyote/1.1

Zum Schluss haben wir der Übersicht halber in die Tabelle zur Auswahl der zu bearbeitenden Strecke noch Felder für die SourceStation und TargetStation hinzugefügt.

## Show Tracks

5 items found, displaying all items.

1

Speed Limit	Length	Source Station Name	Target Station Name	Id	
3	5	Giessen	asfas		Edit
2	7	Giessen	asfas		Edit
2	1	asfas	asfas		Edit
5	5	Giessen	Giessen		Edit
5	5	Giessen	Giessen		Edit

Export options:  CSV |  Excel |  XML |  PDF

Fields marked with an asterisk are  
required

[Help](#) 

Dazu musste die Klasse SearchStationVO um die 2 Felder sourceStationName und targetStationName erweitert werden, sowie die Funktion toSearchTrackVO in der Datei TrackDaoImpl.java angepasst werden.

```
/**
 * @see domain.TrackDao#toSearchTrackVO(domain.Track, valueobjects.SearchTrackVO)
 */
public void toSearchTrackVO(
    domain.Track source,
    valueobjects.SearchTrackVO target)
{
    // @todo verify behavior of toSearchTrackVO
    super.toSearchTrackVO(source, target);

    if(source.getSourceStation() != null) {
        target.setSourceStationName(source.getSourceStation().getStationName());
    }
    if(source.getTargetStation() != null) {
        target.setTargetStationName(source.getTargetStation().getStationName());
    }
}
```