

Tutorial

Ein Rangierbeispiel mit XML-Scripten umgesetzt

Vorwort

In diesem Tutorial wird die Steuerung eines Rangier-Szenarios nur mit Hilfe von XML-Scripten beschrieben. Dieses Beispiel kann man sicherlich auch konventionell mit Aktionen umsetzen. Allerdings ist der Aufwand entsprechend hoch und man kann dabei sehr schnell den Überblick bei den Aktionssteuerungen und Bedingungen verlieren. Dieses Beispiel soll zeigen, daß man sich wiederholende Prozeduren übersichtlich in ein Script auslagern kann. Damit kann man dann den eigentlichen Plan übersichtlich halten. In diesem Tutorial wird nicht auf die Grundfunktionen von Rocrail eingegangen und es wird vorausgesetzt, daß der Leser das Verwenden von Aktionen und Variablen beherrscht.

Mein besonderer Dank gilt Rob Versluis, der mit seiner Bereitschaft das XML-Scripting in Rocrail zu implementieren und sein unermüdlicher Einsatz für Erweiterungen, dieses Beispiel ermöglicht hat.

Unter www.rocrail.net können weitere Informationen zur Software Rocrail aufgerufen werden.

Dieses Tutorial wurde von Gerrit Schulze erstellt und erhebt keinen Anspruch auf Vollständigkeit.

Das Szenario

Ein Güterzug soll, wenn er beladen ist, im Bahnhof halten. Die Lok löst sich dann vom Güterzug und fährt auf ein Abstellgleis. Eine weitere Lok (Rangierlok) soll dann an den Güterzug fahren und die einzelnen Wagen aus dem Güterzug auf entsprechende Abstellgleise verteilen. Wenn die Wagen entladen sind, soll die Rangierlok die Wagen wieder zu einem Güterzug zusammenstellen. Die Güterzuglok übernimmt dann wieder den Güterzug und setzt seine Reise fort.

Anmerkungen

Das Beispiel beschränkt sich auf drei Güterwagen, die alle gleichzeitig beladen oder entladen werden. Und es wird jeder Wagen einzeln bewegt. Es ist aber auch denkbar und umsetzbar, daß mehrere Wagen der gleichen Untergruppe gleichzeitig aus dem Güterzug gelöst werden, oder nur einzelne leere Wagen aus den Abstellgleisen wieder zu einem Güterzug zusammengeführt werden.

Damit dieses Beispiel aber auch für Einsteiger nachvollziehbar bleibt, habe ich mich für dieses Minimalbeispiel entschieden. Wichtig ist, daß man sich im Vorfeld Gedanken darüber macht, was soll an welcher Stelle passieren und unter welchen Bedingungen. Es dann anschließend in ein Script zu packen, kann einfacher sein, als es herkömmlich mit Aktionen zu lösen. Gerade wenn die Aufgabenstellung komplexer ist und sich Abläufe wiederholen. Für die Erstellung der Scripte habe ich einen externen Editor verwendet, der auch die Syntax hervorhebt, um Syntaxfehler zu vermeiden.

Das beschriebene Szenario habe ich mit drei Scripten und 7 Aktionen umgesetzt. Drei Aktionen für das Aufrufen der Scripte und 4 Aktionen um die zwei benötigten Zugkennungen in eine Variable zu schreiben.

Vorarbeiten

Als erstes einmal benötigt man in der Fahrdienstleitertabelle zwei Züge. Einen „Güterzug“ und einen für's „Rangieren“. Außerdem müssen drei Wagen dem Güterzug zugeordnet werden. Wichtig ist, daß bei den Wageneinstellungen eine Untergruppe ausgewählt ist. Über diese wird später ausgewählt, in welches Abstellgleis der Wagen geschoben wird (z.B. ein „Kesselwagen“ zum Tanklager, ein offener „offener Güterwagen“ zur Kohlehandlung, ein „Containerwagen“ an den Kai).

Der Zug „Rangieren“ wird der Rangierlok „loco2-R“ zugeordnet, der Zug „Güterzug“ der Güterzuglok „loco1-G“. Dann werden noch zwei Variablen für die Zug-Kennung benötigt. Eine für den Block „BK5“ (Ich habe sie „BK5-ZugID“ genannt) und eine für den Block „BK1“ (diese habe „BK1-ZugID“ genannt). Sie werden im Prinzip nur dafür benötigt, daß man die Züge dann später auch anders benennen kann und so immer die richtige Kennung verwendet wird. Diese Variablen werden dann über den Block BK5, bzw. BK1 bei „Enter“ aktualisiert.

Der Schalter „Car-Status“ steuert das be- und entladen der Wagen und hat drei Stati (grün = beladen, rot=entladen und active= der Güterzug dreht seine Runden, ohne das etwas passiert)

Die Scripte

Wagen-Ladung

```
<?xml version="1.0" encoding="UTF-8"?>
<xmlscript>
<if state="co Car-Status = on">
<then>
<car id="Containerwagen" cmd="loaded"/>
<car id="Kesselwagen" cmd="loaded"/>
<car id="Kohlenwagen" cmd="loaded"/>
</then>
</if>
<if state="co Car-Status = off">
<then>
<car id="Containerwagen" cmd="empty"/>
<car id="Kesselwagen" cmd="empty"/>
<car id="Kohlenwagen" cmd="empty"/>
</then>
</if>
</xmlscript>
```

Dieses Script ist relativ unspektakulär. Es setzt die Wagen bei Status „on“ von Car-Status auf beladen und bei Status „off“ auf entladen und wird aufgerufen, wenn der Schalter „Car-Status“ verändert wird. Ich habe es deshalb auch nicht weiter auskommentiert.

Gueterzug

Dieses Script steuert den Güterzug. Gehen wir die einzelnen Schritte, was das Script können soll, einmal durch. Es soll den Güterzug anhalten, wenn er beladen ist und die Lok auf ein Abstellgleis fahren. Nehmen wir also diesen ersten Teil:

```
<!-- Wenn der Güterzug im Block BK1 angekommen ist und er ist beladen, wird der Güterzug getrennt und die Lok fährt auf ein Nebengleis -->
<if condition="%lccargo% # goods" state="lc loco1-G = BK1|lc loco2-R = BK5|co Car-Status = on">
<then>
<lc id="%lclid%" cmd="stop"/>
```

```

<lc id="%lcid%" cmd="releasetrain"/>
<lc id="%lcid%" cmd="gotoblock" blockid="BK6"/>
<lc id="%lcid%" cmd="go"/>
<exit/>
</then>
</if>

```

In den Bedingungen wird also geprüft, ob es sich um einen Güterzug %lccargo% # goods handelt. Als Status wird geprüft, ob die Güterzuglok sich in BK1 lc loco1-G = BK1 befindet, die Rangierlok in BK5 lc loco2-R = BK5 und der Schalter auf on co Car-Status = on steht. Wenn alle Bedingungen wahr sind, also „true“ im Trace zurückgeliefert wird, dann kommen die Befehle:

```

Güterzuglok stoppen <lc id="%lcid%" cmd="stop"/>
Güterzug von der Lok lösen <lc id="%lcid%" cmd="releasetrain"/>
Ziel für die Güterzuglok setzen <lc id="%lcid%" cmd="gotoblock" blockid="BK6"/>
Die Lok starten <lc id="%lcid%" cmd="go"/>
Das Script verlassen, damit nicht auf die nächste Bedingung gesprungen wird <exit/>

```

Der zweite Teil – Die Güterzuglok soll, wenn sie wieder im Block BK1 ist mit dem Güterzug verbunden werden und starten

```

<!-- Wenn die Güterlok wieder im Block BK1 ist, wird der Güterzug mit der Lok verbunden, die Platzierung der Lok gedreht und der Zug fährt los. Der Schalter wird auf "active" gesetzt, damit der Zug seine Runden drehen kann, ohne wieder direkt in BK1 zu stoppen -->
<if state="lc loco1-G = BK1|co Car-Status = off">
<then>
<co id="Car-Status" cmd="active"/>
<lc id="%lcid%" cmd="assigntrain" train="Gueterzug"/>
<lc id="%lcid%" cmd="swap"/>
<lc id="%lcid%" cmd="go"/>
<exit/>
</then>
</if>

```

In den Bedingungen wird also geprüft ob die Güterzuglok in BK1 ist und ob der Schalter auf off (entladen) steht. Dann wird der Schalter auf active gesetzt, der Güterzug der Lok zugeordnet, die Platzierung der Lok gewechselt, damit sie in die richtige Richtung fährt und gestartet. Am Ende wieder ein <exit/>.

Der dritte Teil – Die Güterzuglok soll wieder an den Güterzug fahren, wenn die entladenen Wagen wieder zum Güterzug zusammengestellt sind. Hier ist das Problem, daß nicht die Lok oder der Block in dem die Lok steht (BK6) selbst dieses Ereignis auslösen kann, das kann nur am Ende von „Rangieren“ erfolgen. Die Rangierlok sagt sozusagen, daß sie fertig ist und die Güterzuglok starten soll. Deshalb wird dieser Teil zwar in das Script Güterzug gepackt, aber als eine Funktion, die dann später aus dem Script rangieren.xml aufgerufen wird.

```

<!-- Diese Funktion wird aus dem script rangieren.xml aufgerufen -->
<function id="Ziel_BK1">
<lc id="loco1-G" cmd="gotoblock" blockid="BK1"/>
<lc id="loco1-G" cmd="go"/>
</function>

```

Wichtig ist an dieser Stelle zu wissen und zu beachten, daß dieser Teil des Scripts muß unbedingt **am Anfang** des Scripts stehen.

Das ganze Script sieht dann so aus:

```

<?xml version="1.0" encoding="UTF-8"?>
<xmlscript>
<!-- Diese Funktion wird aus dem script rangieren.xml aufgerufen -->
<function id="Ziel_BK1">
<lc id="loco1-G" cmd="gotoblock" blockid="BK1"/>
<lc id="loco1-G" cmd="go"/>

```

```

</function>
<!-- Wenn der Güterzug im Block BK1 angekommen ist und er ist beladen, wird der Güterzug getrennt und
die Lok fährt auf ein Nebengleis -->
<if condition="%lccargo% # goods" state="lc loco1-G = BK1|lc loco2-R = BK5|co Car-Status = on">
<then>
  <lc id="%lclid%" cmd="stop"/>
  <lc id="%lclid%" cmd="releasetrain"/>
  <lc id="%lclid%" cmd="gotoblock" blockid="BK6"/>
  <lc id="%lclid%" cmd="go"/>
  <exit/>
</then>
</if>
<!-- Wenn die Güterlok wieder im Block BK1 ist, wird der Güterzug mit der Lok verbunden, die Platzierung
der Lok gedreht und der Zug fährt los. Der Schalter wird auf "active" gesetzt, damit der Zug seine Runden
drehen kann, ohne wieder direkt in BK1 zu stoppen -->
<if state="lc loco1-G = BK1|co Car-Status = off">
<then>
  <co id="Car-Status" cmd="active"/>
  <lc id="%lclid%" cmd="assigntrain" train="@BK1-ZugID"/>
  <lc id="%lclid%" cmd="swap"/>
  <lc id="%lclid%" cmd="go"/>
  <exit/>
</then>
</if>
</xmlscript>

```

Jetzt haben wir schon zwei von drei Scripten zusammen.

Rangieren

Das entscheidende und daher „umfangreichste“ Script ist aber das Rangieren. Hier muß die Rangierlok unter verschiedenen Bedingungen gesteuert werden. Aber genau hier liegt auch der Vorteil von Scripten. Da sich hier Abläufe wiederholen, sind diese in einem Script einfacher integriert und kopiert. Das Script wird über die einzelnen Blöcke gesteuert, also bei „occupied“ in einem Block gestartet, denn nur im jeweiligen Block weiß man, wie die Rangierlok zu steuern ist. Deshalb beginnt auch jeder Abfrageteil im Script damit, das geprüft wird, wo sich die Rangierlok befindet. Auch hier nehmen wir uns die einzelnen Bestandteile vor.

Fangen wir mit einem einfachen Teil an. Die Abstellblöcke für die Wagen. Hier soll geprüft werden, ob die Wagen beladen sind und von der Rangierlok getrennt, oder entladen sind und mit der Rangierlok verbunden werden sollen.

```

<!-- Prüfen ob die Rangierlok im Block BK2 steht -->
<if state="lc loco2-R = BK2">
<then>
  <!-- DANN, Prüfen ob die Wagen auf beladen stehen -->
  <if state="co Car-Status = on">
  <then>
    <!-- DANN, Prüfen welcher Wagen im Block BK2 steht und beladen ist -->
    <foreach table="carlist" state="car %oid% = BK2|car %oid% = loaded">
    <!-- Wagen aus dem Rangierzug lösen -->
    <operator id="@BK5-ZugID" cmd="removecar" carids="%oid%"/>
    </foreach>
    <!-- Fahrbefehl Richtung BK5 -->
    <lc bkid="BK2" cmd="gotoblock" blockid="BK5"/>
    <lc bkid="BK2" cmd="go"/>
    <!-- script verlassen -->
    <exit/>
  </then>
</if>

```

```

<!-- DANN, Prüfen ob die Wagen auf entladen stehen -->
<if state="co Car-Status = off">
<then>
<!-- DANN, Prüfen welche Wagen im Block BK2 stehen und entladen sind -->
<foreach table="carlist" state="car %oid% = BK2|car %oid% = empty">
<!-- Wagen in den Rangierzug einfügen -->
<operator id="@BK5-ZugID" cmd="addcar" carids="%oid%"/>
</foreach>
<!-- Fahrbefehl Richtung BK5 -->
<lc bkid="BK2" cmd="gotoblock" blockid="BK5"/>
<lc bkid="BK2" cmd="go"/>
<!-- script verlassen -->
<exit/>
</then>
</if>
</then>
</if>

```

Die Prüfung des Schalters „Car-Status“ gibt als erstes vor, ob nach be- oder entladenen Wagen gesucht werden soll `<if state="co Car-Status = on">` und diese dann an die Rangierlok gebunden oder gelöst werden sollen. Im zweiten Schritt wird nach passenden Wagen über eine foreach-Schleife im Block BK2 `<foreach table="carlist" state="car %oid% = BK2|car %oid% = loaded">` gesucht. %oid% beinhaltet die Wagenkennung, wenn die Wagenliste durchgegangen wird. Da hier nach beladen gesucht wird, muß der Wagen anschließend von der Rangierlok gelöst werden `<operator id="@BK5-ZugID" cmd="removecar" carids="%oid%"/>`. Anschließend wird die Lok wieder Richtung BK5 geschickt, damit sie den nächsten Wagen holen kann. Auch hier wird am Ende mit einem `<exit/>` ausgestiegen, damit keine weiteren Bedingungen oder Schleifen durchlaufen werden. Das Ganze dann noch mal für Car-Status „off“. Den Aufbau dieses Script-Teils benötigen wir jetzt noch mal für BK3 und BK4. Hier liegt schon der erste Vorteil. Einfach kopieren und die Parameter von BK2 auf BK3 und BK4 anpassen – fertig. Der Script-Teil sieht dann komplett so aus:

```

<!-- Prüfen ob die Rangierlok im Block BK2 steht -->
<if state="lc loco2-R = BK2">
<then>
<!-- DANN, Prüfen ob die Wagen auf beladen stehen -->
<if state="co Car-Status = on">
<then>
<!-- DANN, Prüfen welcher Wagen im Block BK2 steht und beladen ist -->
<foreach table="carlist" state="car %oid% = BK2|car %oid% = loaded">
<!-- Wagen aus dem Rangierzug lösen -->
<operator id="@BK5-ZugID" cmd="removecar" carids="%oid%"/>
</foreach>
<!-- Fahrbefehl Richtung BK5 -->
<lc bkid="BK2" cmd="gotoblock" blockid="BK5"/>
<lc bkid="BK2" cmd="go"/>
<!-- script verlassen -->
<exit/>
</then>
</if>
<!-- DANN, Prüfen ob die Wagen auf entladen stehen -->
<if state="co Car-Status = off">
<then>
<!-- DANN, Prüfen welche Wagen im Block BK2 stehen und entladen sind -->
<foreach table="carlist" state="car %oid% = BK2|car %oid% = empty">
<!-- Wagen in den Rangierzug einfügen -->
<operator id="@BK5-ZugID" cmd="addcar" carids="%oid%"/>
</foreach>
<!-- Fahrbefehl Richtung BK5 -->
<lc bkid="BK2" cmd="gotoblock" blockid="BK5"/>
<lc bkid="BK2" cmd="go"/>
<!-- script verlassen -->
<exit/>
</then>
</if>

```

```

        </then>
    </if>
</then>
</if>
<!-- Prüfen ob die Rangierlok im Block BK3 steht -->
<if state="lc loco2-R = BK3">
<then>
    <!-- DANN, Prüfen ob die Wagen auf beladen stehen -->
    <if state="co Car-Status = on">
    <then>
        <!-- DANN, Prüfen welcher Wagen im Block BK3 steht und beladen ist -->
        <foreach table="carlist" state="car %oid% = BK3|car %oid% = loaded">
        <operator id="@BK5-ZugID" cmd="removecar" carids="%oid%"/>
        </foreach>
        <!-- Fahrbefehl Richtung BK5 -->
        <lc bkid="BK3" cmd="gotoblock" blockid="BK5"/>
        <lc bkid="BK3" cmd="go"/>
        <!-- script verlassen -->
        <exit/>
    </then>
    </if>
    <!-- DANN, Prüfen ob die Wagen auf entladen stehen -->
    <if state="co Car-Status = off">
    <then>
        <!-- DANN, Prüfen welche Wagen im Block BK3 stehen und entladen sind -->
        <foreach table="carlist" state="car %oid% = BK3|car %oid% = empty">
        <operator id="@BK5-ZugID" cmd="addcar" carids="%oid%"/>
        </foreach>
        <!-- Fahrbefehl Richtung BK5 -->
        <lc bkid="BK3" cmd="gotoblock" blockid="BK5"/>
        <lc bkid="BK3" cmd="go"/>
        <!-- script verlassen -->
        <exit/>
    </then>
    </if>
</then>
</if>
<!-- Prüfen ob die Rangierlok im Block BK4 steht -->
<if state="lc loco2-R = BK4">
<then>
    <!-- DANN, Prüfen ob die Wagen auf beladen stehen -->
    <if state="co Car-Status = on">
    <then>
        <!-- DANN, Prüfen welcher Wagen im Block BK4 steht und beladen ist -->
        <foreach table="carlist" state="car %oid% = BK4|car %oid% = loaded">
        <operator id="@BK5-ZugID" cmd="removecar" carids="%oid%"/>
        </foreach>
        <!-- Fahrbefehl Richtung BK5 -->
        <lc bkid="BK4" cmd="gotoblock" blockid="BK5"/>
        <lc bkid="BK4" cmd="go"/>
        <!-- script verlassen -->
        <exit/>
    </then>
    </if>
    <!-- DANN, Prüfen ob die Wagen auf entladen stehen -->
    <if state="co Car-Status = off">
    <then>
        <!-- DANN, Prüfen welche Wagen im Block BK4 stehen und entladen sind -->
        <foreach table="carlist" state="car %oid% = BK4|car %oid% = empty">
        <operator id="@BK5-ZugID" cmd="addcar" carids="%oid%"/>
        </foreach>
        <!-- Fahrbefehl Richtung BK5 -->
        <lc bkid="BK4" cmd="gotoblock" blockid="BK5"/>

```

```

        <lc bkid="BK4" cmd="go"/>
        <!-- script verlassen -->
        <exit/>
    </then>
</if>
</then>
</if>

```

(Zur besseren Verdeutlichung sind die einzelnen Schritte mit „TAB“ eingerückt, daher nicht zum kopieren verwenden, sondern das original Script!)

Jetzt haben wir vom dritten Script schon 1/3. Sehen wir uns jetzt den Block BK1 an. Hier soll die Rangierlok entweder die Wagen vom Güterzug herauslösen und übernehmen oder in den Güterzug wieder einfügen. Es muß also geprüft werden, ob der Schalter Car-Status auf be- oder entladen steht und dann, ob ein Wagen be- oder entladen ist.

```

<!-- Prüfen ob die Rangierlok im Block BK1 steht -->
<if state="lc loco2-R = BK1">
<then>
    <!-- DANN, Prüfen ob die Wagen auf beladen stehen -->
    <if state="co Car-Status = on">
    <then>
        <!-- DANN, Prüfen welche Wagen im Block BK1 stehen und beladen sind -->
        <foreach table="carlist" state="car %oid% = BK1|car %oid% = loaded">
        <!-- Den Wagen aus dem Güterzug lösen und in den Rangierzug einfügen -->
        <operator id="@BK1-ZugID" cmd="removecar" carids="%oid%"/>
        <operator id="@BK5-ZugID" cmd="addcar" carids="%oid%"/>
        <break/>
        </foreach>
        <!-- Fahrbefehl Richtung BK5, hier ist ein Platzierungswechsel notwendig -->
        <lc bkid="BK1" cmd="swap"/>
        <lc bkid="BK1" cmd="gotoblock" blockid="BK5"/>
        <lc bkid="BK1" cmd="go"/>
        <!-- script verlassen -->
        <exit/>
    </then>
    </if>
    <!-- DANN, Prüfen ob die Wagen auf entladen stehen -->
    <if state="co Car-Status = off">
    <then>
        <!-- DANN, Prüfen welcher Wagen sich im Zug Rangieren befindet und entladen sind -->
        <foreach table="carlist" condition="@BK5-ZugID = %operatorid%">
        <!-- Wagen aus dem Zug Rangieren lösen und in den Güterzug einfügen -->
        <operator id="@BK5-ZugID" cmd="removecar" carids="%oid%"/>
        <operator id="@BK1-ZugID" cmd="addcar" carids="%oid%"/>
        <break/>
        </foreach>
        <!-- Fahrbefehl Richtung BK5, hier ist ein Platzierungswechsel notwendig -->
        <lc bkid="BK1" cmd="swap"/>
        <lc bkid="BK1" cmd="gotoblock" blockid="BK5"/>
        <lc bkid="BK1" cmd="go"/>
        <!-- script verlassen -->
        <exit/>
    </then>
    </if>
</then>
</if>

```

Auch hier beginnt es damit, daß erst einmal geprüft wird, ob die Rangierlok sich im Block BK1 befindet <if state="lc loco2-R = BK1">. Dann wird geprüft, ob der Schalter Car-Status auf be- oder entladen steht <if state="co Car-Status = on">. Dann werden passende Wagen im Block BK1 gesucht. Dann wird der Wagen entweder aus dem Güterzug gelöst und der Rangierlok hinzugefügt <foreach table="carlist" state="car %oid% = BK1|car %oid% = loaded">, oder eben umgekehrt. Anschließend ein

Platzierungswechsel für die Rangierlok und das Ziel BK5 mit einem „go“. Damit hätten wir schon 2/3 vom dritten Script.

Fehlt jetzt noch der Block BK5. Hier müssen mehrere Dinge abgeprüft werden. Zum besseren Verständnis gehen wir es Schritt für Schritt durch. Als Erstes wird wieder geprüft, ob die Rangierlok sich in BK5 befindet.

```
<!-- Prüfen ob die Rangierlok im Block BK5 steht -->
```

```
<if state="lc loco2-R = BK5">
```

```
<then>
```

Dann wird geprüft, ob der Schalter Car-Status auf „beladen“ steht.

```
<!-- DANN, Prüfen ob die Wagen auf beladen stehen -->
```

```
<if state="co Car-Status = on">
```

```
<then>
```

Dann wird geprüft, welcher Wagen im Block BK5 steht und welche Untergruppe er hat, damit er auf das richtige Abstellgleis gestellt wird. Hier gilt es jetzt darauf zu achten, in welcher Reihenfolge die Wagen aus dem Güterzug gelöst werden. Nach dieser Reihenfolge werden die Wagen jetzt wegsortiert. Später müssen sie dann wieder in umgekehrter Reihenfolge zusammengestellt werden, damit die Reihenfolge wieder stimmt. Im Beispiel gibt es drei Wagen (Containerwagen, Kesselwagen, Kohlenwagen). Die Reihenfolge ergibt sich aus der Reihenfolge in der Plan-Datei, also so, wie die Wagen in Rocrail angelegt wurden. Um also die richtige Reihenfolge zu erhalten, empfiehlt es sich, eine Dokumentenliste aus der Wagenliste zu generieren. Dort hat man dann die Reihenfolge, in der die Wagen in Rocrail angelegt wurden. BK4 ist das Tanklager, BK2 das Kohlenlager und BK3 der Hafenkai.

```
<!-- DANN, Prüfen welcher Wagen noch im Block BK5 steht und welche Untergruppe er hat -->
```

```
<foreach table="carlist" condition="%carsubtype% # tankcar" state="car %oid% = BK5">
```

```
<!-- Fahrbefehl Richtung BK4 -->
```

```
<lc bkid="BK5" cmd="gotoblock" blockid="BK4"/>
```

```
<lc bkid="BK5" cmd="go"/>
```

```
<!-- script verlassen -->
```

```
<exit/>
```

```
</foreach>
```

```
<foreach table="carlist" condition="%carsubtype% # gondola" state="car %oid% = BK5">
```

```
<!-- Fahrbefehl Richtung BK2 -->
```

```
<lc bkid="BK5" cmd="gotoblock" blockid="BK2"/>
```

```
<lc bkid="BK5" cmd="go"/>
```

```
<!-- script verlassen -->
```

```
<exit/>
```

```
</foreach>
```

```
<foreach table="carlist" condition="%carsubtype% # wellcar" state="car %oid% = BK5">
```

```
<!-- Fahrbefehl Richtung BK3 -->
```

```
<lc bkid="BK5" cmd="gotoblock" blockid="BK3"/>
```

```
<lc bkid="BK5" cmd="go"/>
```

```
<!-- script verlassen -->
```

```
<exit/>
```

```
</foreach>
```

Die Untergruppe wird mit %carsubtype% ermittelt. Das Ergebnis sind die englischen Bezeichnungen. Also für Kesselwagen = tankcar, offener Güterwagen = gondola und Containerwagen = wellcar. Wenn kein Wagen in Block BK5 steht, muß noch geprüft werden, ob sich noch weitere Wagen in BK1 befinden und Die Lok Richtung BK1 geschickt werden.

```
<!-- DANN, Prüfen welche Wagen noch im Block BK1 stehen und und noch beladen sind -->
```

```
<foreach table="carlist" state="car %oid% = BK1|car %oid% = loaded">
```

```
<!-- Fahrbefehl Richtung BK1 -->
```

```
<lc bkid="BK5" cmd="gotoblock" blockid="BK1"/>
```

```
<lc bkid="BK5" cmd="go"/>
```

```
<!-- script verlassen -->
```

```
<exit/>
```

```
</foreach>
```

Jetzt kommt noch der umgekehrte Weg, also die leeren Wagen wieder zum Güterzug

zusammenstellen. Hier wird als erstes der Schalter Car-Status auf „off“ geprüft.

```
<!-- DANN, Prüfen ob die Wagen auf entladen stehen -->
```

```
<if state="co Car-Status = off">
```

```
<then>
```

Jetzt wird als erstes geprüft, ob die Rangierlok bereits einen leeren Wagen hat und damit ein Wagen in Block BK5 steht.

```
<!-- DANN, Prüfen welcher Wagen noch im Block BK5 steht und entladen ist -->
```

```
<foreach table="carlist" state="car %oid% = BK5|car %oid% = empty">
```

```
<!-- Fahrbefehl Richtung BK3 -->
```

```
<lc bkid="BK5" cmd="gotoblock" blockid="BK1"/>
```

```
<lc bkid="BK5" cmd="go"/>
```

```
<!-- script verlassen -->
```

```
<exit/>
```

```
</foreach>
```

Wenn das nicht der Fall ist kann jetzt nach leeren Wagen in BK2,BK3 und BK4 gesucht werden.

Hier gilt es die umgekehrte Reihenfolge zum Abstellen einzuhalten. Es wird jetzt also erst der Containerwagen, dann der Kohlenwagen, dann der Kesselwagen geholt.

```
<!-- DANN, Prüfen welcher Wagen im Block BK3 stehen und entladen ist -->
```

```
<foreach table="carlist" state="car %oid% = BK3|car %oid% = empty">
```

```
<!-- Fahrbefehl Richtung BK3 -->
```

```
<lc bkid="BK5" cmd="gotoblock" blockid="BK3"/>
```

```
<lc bkid="BK5" cmd="go"/>
```

```
<!-- script verlassen -->
```

```
<exit/>
```

```
</foreach>
```

```
<!-- DANN, Prüfen welcher Wagen im Block BK2 steht und entladen ist -->
```

```
<foreach table="carlist" state="car %oid% = BK2|car %oid% = empty">
```

```
<!-- Fahrbefehl Richtung BK3 -->
```

```
<lc bkid="BK5" cmd="gotoblock" blockid="BK2"/>
```

```
<lc bkid="BK5" cmd="go"/>
```

```
<!-- script verlassen -->
```

```
<exit/>
```

```
</foreach>
```

```
<!-- DANN, Prüfen welcher Wagen im Block BK4 steht und entladen ist -->
```

```
<foreach table="carlist" state="car %oid% = BK4|car %oid% = empty">
```

```
<!-- Fahrbefehl Richtung BK3 -->
```

```
<lc bkid="BK5" cmd="gotoblock" blockid="BK4"/>
```

```
<lc bkid="BK5" cmd="go"/>
```

```
<!-- script verlassen -->
```

```
<exit/>
```

```
</foreach>
```

Damit haben wir den 3/3 Teil vom dritten Script fast komplett. Jetzt kommt nur noch der Aufruf der Funktion im Script Güterzug, damit die Güterzuglok wieder an den Güterzug fährt. Dieser Funktionsaufruf kommt an das Ende vom Script, denn wenn keiner der vorhergehenden Bedingungen erfüllt ist, hat die Rangierlok ihren Job getan und bleibt in BK5 stehen. Das Einzige, was jetzt noch passiert ist die Güterzuglok nach BK1 zu bringen, wenn der Car-Status off ist.

```
<if state="co Car-Status = off">
```

```
<then>
```

```
<sub file="gueterzug.xml" id="Ziel_BK1"/>
```

```
</then>
```

```
</if>
```

```
</xmlscript>
```

Das Script rangieren.xml wird jetzt in die Blöcke BK1, BK2,BK3,BK4 und BK5 mit Status „occupied“ eingebunden. Zusätzlich empfiehlt es sich, beim Block BK1 das Script „rangieren.xml“ zusätzlich mit der Bedingung „Zug Rangieren“ zu versehen und das Script „güterzug.xml“ mit der Bedingung „Zug Güterzug“. Dann wird das jeweilige Script in dem Block auch nur vom passenden Zug aufgerufen.

DAS WAR'S :)