Ein Transportsystem mit passiven Elementen modellieren



Wenn Sie Transportsysteme mit passiven Elementen modellieren, setzen Sie den *Weg* und das *Fahrzeug* ein.

Da der *Weg* ein passives Objekt ist, können wir ihn nicht einfach mit zwei Stationen verbinden. Wir müssen vielmehr sicherstellen, daß ein *Fahrzeug* zur Verfügung steht. Außerdem müssen wir dieses *Fahrzeug* beladen und entladen. Um zu verhindern, daß BEs automatisch von der vorangehenden Station auf den *Weg* umlagern, in unserem Beispiel von *ES1*, haben wir sie nicht mit einer *Kante* verbunden.



Um ein Fahrzeug zu erzeugen und einzusetzen, können Sie eine der folgenden Aktionen ausführen:

• Die *Quelle* benutzen: Wählen Sie Attribute > Erzeugungszeitpunkt > Anzahl einstellbar und tragen Sie eine Anzahl von 1 ein. Damit sie Fahrzeuge erzeugt, wählen Sie BE > .BEs.Transporter.

→.Modelle.Transportsysteme2.Quelle ? 🛛
Navigieren Ansicht Extras ?
Name: Quelle 🔽 🗖 Gestört Etikett: 🔽 🗖 Geplant 💌 🗖 Ausgang gesperrt 🗖
Attribute Störungen Steuerungen Ausgangsverhalten Statistik Benutzerdefinierte
Betriebsart: Image: Blockierend Erzeugungszeitpunkt: Anzahl einstellbar Anzahl: 1
Abstand: Konst 💌 1:00.0000
BE Auswahl: Konstant BE:
OK Abbrechen Übernehmen

• Die SimTalk-Methode *erzeugen* benutzen und diesen Befehl in die *init*-Methode Ihres Simulationsmodells eintragen: .BEs.Fahrzeug.erzeugen(Weg);

Dies setzt ein *Fahrzeug* am Ende des *Wegs* ein. Wenn Sie ein *Fahrzeug* an einer anderen Stelle auf dem *Weg* einsetzen möchten, tragen Sie eine Position in die Methode ein. Dies Beispiel erzeugt ein *Fahrzeug* an der Position 5,5 Meter des *Wegs*.

M.Modelle.Transportsysteme2.init *	
]] <u>D</u> atei <u>B</u> earbeiten <u>N</u> avigieren A <u>u</u> sführen	Templ <u>a</u> te <u>Ansicht</u> <u>Extras</u> <u>?</u>
🕅 🙆 X 🖻 🛍 그 으 🗚 🅯	4 # 🗚 🎋 🖗 🗄 🗄
is	
do	
.BEs.Fahrzeug.erzeugen	(Weg, 5.5);
end;	

Da das *Fahrzeug* Teile von Station zu Station transportieren soll, *ES1* und *ES2* in unserem Beispiel, müssen wir diese Teile an der Station *ES1* auf das *Fahrzeug* laden und es an der Station *ES2* wieder davon abladen. Dafür programmieren wir eine Steuerung in einer Methode und tragen deren Namen in das entsprechende Textfeld auf der Registerkarte **Steuerungen** ein.

	ysteme2.Weg ? 🗙
Navigieren Ansicht Extr	ras ?
Name: Weg Etikett:	Geplant Carporal Gestort
Attribute Zeiten Stör	ungen Steuerungen Ausgangsverhalten Statistik Kurve 💶 🕨
Eingang: Ausgang: Eingang rückwärts:	Image: Bug: Im
Ausgang rückwärts:	RückwärtsAusStrg
Schichtkalender:	
	Sensoren Anzahl der Sensoren: 0
	OK Abbrechen Übernehmen

Wenn Sie keine Steuerung eintragen, bewegt sich das *Fahrzeug* nur bis zum Ende des *Wegs* und hält dort an. Damit das *Fahrzeug* an den Anfang des *Wegs* zurückfährt, wenn es die Teile abgeladen hat, aktivieren Sie das Kontrollkästchen **Bug** und tragen Sie diese Anweisungen in die **Ausgang**-Steuerung ein, *RückwärtsAusStrg* in unserem Beispiel:

```
if @.leer then
    @.rueckwaerts := true;
end;
```

Sobald das *Fahrzeug* rückwärts an den Anfang des *Wegs* gefahren ist, aktiviert eM-Plant die heckgesteuerte Ausgangssteuerung, wenn Sie das Kontrollkästchen **Heck** aktivieren und den Namen der Methode in das Textfeld **Ausgang rückwärts** eintragen. Diese Steuerung muß diese Aufgaben erfüllen:

- Warten, bis ein BE bereit ist aus der Station, ESI in unserem Beispiel, auszutreten.
- Die BEs auf das *Fahrzeug* aufladen.
- Das *Fahrzeug* ans Ende des Wegs fahren lassen.

In unser Beispiel haben wir diese Anweisungen eingetragen.

M.Mo	odelle.Transportsysteme2.RückwärtsAusStrg
∬ <u>D</u> ate	i <u>B</u> earbeiten <u>N</u> avigieren A <u>u</u> sführen Templ <u>a</u> te <u>A</u> nsicht <u>E</u> xtras <u>?</u>
99	🙆 % 🖻 🛍 그 으 🏘 🎧 縃 🔺 % % % 🗉 🖿 🔅 🗲 🗸
1:	is
2:	do
3:	waituntil ES1.belegt and ES1.inhalt.fertig prio 1;
4:	ES1.inhalt.umlagern(0); das BE aufladen
5:	<pre>@.rueckwaerts := false;</pre>
6:	end;

Zeile 3 teilt eM-Plant mit zu warten, bis sich ein BE auf der Station *ES1* aufhält und ganz bearbeitet ist. Zeile 4 lädt das BE auf das *Fahrzeug*. Beachten Sie, daß wir den anonymen Bezeichner @ benutzen, um das *Fahrzeug* anzusprechen. Zeile 5 veranlaßt das *Fahrzeug* wieder vorwärts zum Ende des *Wegs* zu fahren.

Wenn das Fahrzeug das Ende des Wegs erreicht hat, müssen Sie diese Aufgaben lösen:

- Warten bis die Station, ES1 in unserem Beispiel, leer ist.
- Das *Fahrzeug* entladen.
- Das *Fahrzeug* an den Anfang des *Wegs* fahren lassen.

Da das *Fahrzeug* anfangs ans Ende des *Wegs* fährt und dann die folgende Ausgangssteuerung aktiviert, während es entladen wird, fragen wir ab, ob es beladen oder leer ist. In unser Beispiel haben wir diese Anweisungen eingetragen.

M.Moo	lelle.Transportsysteme2.VorwärtsAusStrg *
∬ <u>D</u> atei	<u>B</u> earbeiten <u>N</u> avigieren A <u>u</u> sführen Templ <u>a</u> te <u>A</u> nsicht <u>E</u> xtras <u>2</u>
	👌 % 🖻 🛍 그 으 🗛 🎧 縃 🔺 % % % 🗄 💕 🔆 🗲 🗸
1:	is
2:	do
3:	if 0.leer then
4:	0.rueckwaerts := true;
5:	else
6:	waituntil ES2.leer prio 1;
7:	<pre>@.inhalt.umlagern(ES2);</pre>
8:	<pre>@.rueckwaerts := true;</pre>
9:	end;
10:	end;

Gehe zu Einen Querverschiebewagen modellieren

Zurück zu Transportsysteme modellieren

\$ ⇒