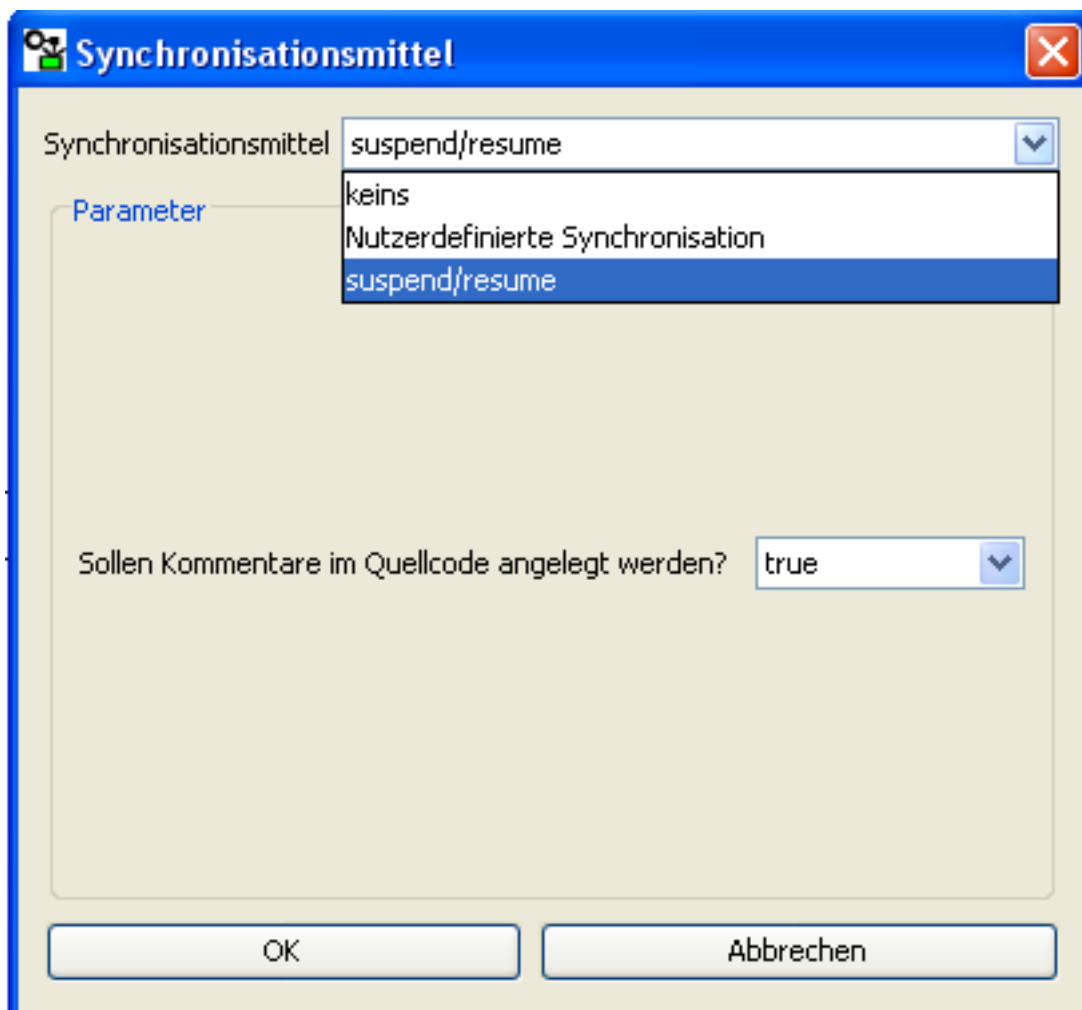


# PND Tool

## Quellcodegenerierung

### Handbuch



Autor: Heiko Weiß

# Inhaltsverzeichnis

<b>1 Allgemeines</b>	<b>3</b>
<b>2 Systemanforderungen</b>	<b>3</b>
<b>3 Anforderungen an ein Petrinetz</b>	<b>3</b>
<b>4 Bibliotheken</b>	<b>3</b>
<b>5 Petrinetzdatei und Bibliothek laden</b>	<b>3</b>
5.1 Quellcodemodus direkt aus dem Designmodus . . . . .	4
5.2 Quellcodemodus nach dem Programmstart . . . . .	4
<b>6 Programmaufbau im Quellcodemodus</b>	<b>4</b>
6.1 Fenster zur Visualisierung und Eingabe von Daten . . . . .	4
6.2 Symbolik . . . . .	5
<b>7 Menüpunkte</b>	<b>6</b>
<b>8 Eingabefenster</b>	<b>7</b>
<b>9 Vorgehensweise</b>	<b>7</b>
9.1 Allgemeine Daten eingeben . . . . .	7
9.2 Prozesse erzeugen/löschen und Objekte zuordnen . . . . .	7
9.3 Prozessdaten eingeben . . . . .	7
9.4 Daten für Synchronisationsplätze eintragen . . . . .	8
9.5 Schleifen/Verzweigungen zuordnen . . . . .	8
9.6 Aktionen zuordnen . . . . .	9
9.7 Testlauf starten . . . . .	10
9.8 Quellcode erstellen . . . . .	10

# 1 Allgemeines

Das hier vorgestellte Programm ist eine Erweiterung des Petrinetz-Designtools, kurz PND. Es behandelt dabei die Erstellung von Quellcodegerüsten auf Basis im PND-Tool erstellter Petrinetze. Dabei können je nach Bedarf Bibliotheken für unterschiedliche Zielsysteme/Programmiersprachen nachgerüstet werden.

## 2 Systemanforderungen

Das vorgestellte Programm inklusive des PND Tools ist getestet unter folgenden Mindestanforderungen:

- Prozessor mit 1.2 GHz
- 512 MB RAM
- Java Runtime Environment ab Version 1.5

## 3 Anforderungen an ein Petrinetz

Folgende Anforderungen müssen an ein Petrinetz gestellt werden, damit es sich zur Quellcodeerstellung eignet:

- Das Petrinetz ist fertig erstellt und braucht nicht mehr strukturell geändert zu werden. Eine nachträgliche Änderung ist während oder nach der Quellcodeerstellung nicht mehr möglich!
- Die einzelnen Petrinetzobjekte lassen sich eindeutig einzelnen Prozessen zuordnen.
- Alle Objekte, die keinem Prozess zugeordnet sind, dürfen nur Plätze sein, welche als Synchronisationsmittel/Betriebsmittel dienen.
- Das Petrinetz enthält keine Aufspaltung in Unterprozesse.

Für die Quellcodeerstellung sind der Typ eines Petrinetzes sowie entsprechende Marken und die Formeln innerhalb der Transitionen unerheblich. Quellcode wird nur auf Grund des Aufbaus eines Netzes sowie aller weiteren nutzerdefinierten Informationen erstellt.

## 4 Bibliotheken

Um Quellcode für unterschiedliche Systeme zur Verfügung zu stellen, müssen sich entsprechende Bibliotheksdateien im Unterordner `..\lib` befinden.

## 5 Petrinetzdatei und Bibliothek laden

Zur Erstellung von Quellcode muss ein existierendes Petrinetz geladen werden, dies kann auf 2 Wegen geschehen:

## 5.1 Quellcodemodus direkt aus dem Designmodus

1. Das Petrinetz wurde im Design-Programm erstellt und geladen. Über die Menüpunkte Datei->Quellcodeerstellungsmodus kann in den Erstellungsmodus gewechselt werden.
2. Im nachfolgenden Bibliotheksfenster wird die Bibliothek, welche dem Zielsystem/der Zielsprache entspricht, ausgewählt.
3. Mit "Bibliothek laden" wird die Bibliothek geladen und der Quellcodemodus gestartet.

## 5.2 Quellcodemodus nach dem Programmstart

1. Nach dem Programmstart wird der Menüpunkt "Quelltexterstellung aus Petrinetz" ausgewählt.
2. Im nachfolgenden Fenster können nun zwei Dateitypen gewählt werden:
  - \*.pnf Dateien: Quellcodeerstellung auf Grund eines erstellten Petrinetzes.
  - \*.pns Dateien: Erneutes Laden einer erstellten Quellcodeerzeugungsdatei.
3. Wurde eine \*.pnf Datei geladen, erscheint ein Fenster, in dem die Bibliothek auszuwählen ist. Dazu muss die entsprechende Bibliothek markiert und mit "Bibliothek laden" aktiviert werden.
4. Wurde eine \*.pns Datei ausgewählt, wird automatisch die Bibliothek, mit der die \*.pns Datei erzeugt wurde, nachgeladen.
5. Das Programm schaltet in den Quellcodemodus.

# 6 Programmaufbau im Quellcodemodus

## 6.1 Fenster zur Visualisierung und Eingabe von Daten

Der Quellcodemodus enthält zwei Hauptfenster, um Informationen bereitzustellen und zu visualisieren. Im **Datenfenster** (Abbildung 2), werden alle Zuordnungen im Petrinetz in einem Baumdiagramm dargestellt. Dabei werden nicht zugeordnete Transitionen im Ast "nicht zugeordnet" des Baumes eingefügt. Nicht zugeordnete Plätze werden im Ast "Synchronisationsplätze" angezeigt.

Im **Petrinetzfenster** (Abbildung 1), wird der strukturelle Aufbau des Petrinetzes dargestellt. In diesem können Zuordnungen von Synchronisationsmitteln, Aktionen, Schleifen und Verzweigungen vorgenommen werden.

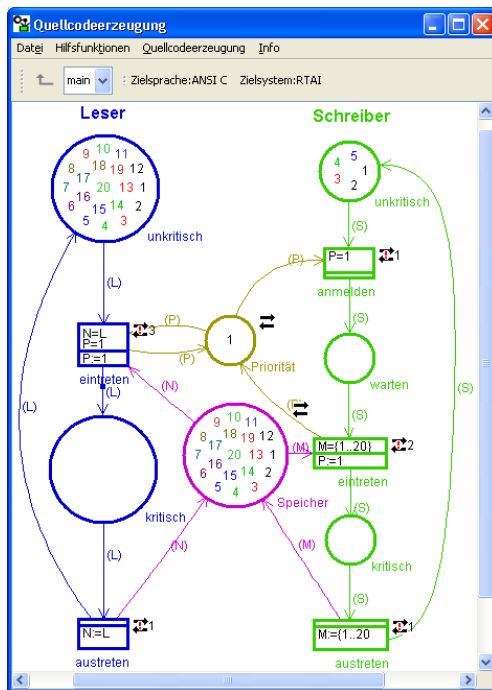


Abbildung 1: Das Petrinetzfenster

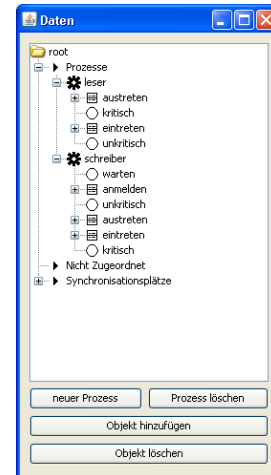


Abbildung 2: Das Datenfenster

## 6.2 Symbolik

Folgende Symbole finden Anwendung:

Symbol	Beschreibung
○	Das Objekt ist ein Platz
□	Das Objekt ist eine Transition
✳	Das Objekt ist ein Prozess
↑↓	Das Objekt ist ein Synchronisationsmittel
→	Das Objekt ist eine Aktion
↘	Das Objekt ist eine Verzweigung
↻	Das Objekt ist eine Schleife

Tabelle 1: Symbolik

Innerhalb des Datenfensters bedeuten die Symbole, dass ein Objekt entsprechenden Typs zugeordnet ist, z. B. dass eine Aktion zu einer bestimmten Transition gehört. Die Abb. 3 zeigt so eine Zuordnung. Dem Prozess "schreiber" ist der Platz "warten" und die Transition "anmelden" zugeordnet. Zu der Transition "anmelden" gehört eine Aktion.

Innerhalb des Petrinetzfensters sind solche Symbole einzelnen Objekten zugeordnet. Dies bedeutet, dass ein entsprechendes Objekt vom Typ des Symbols ist, z. B. Abb. 4 zeigt den Platz "park". Dabei stellen die Symbole dar, dass dieser Platz gleichzeitig eine Schleife und eine Verzweigung repräsentiert.

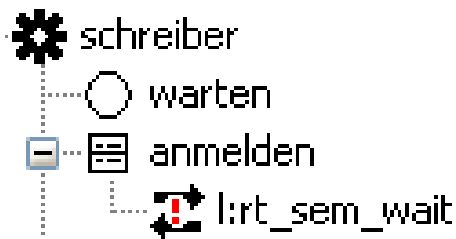


Abbildung 3: Beispiel Datenfenster



Abbildung 4: Beispiel Petrinetzfenster

## 7 Menüpunkte

Menüpunkt	Beschreibung
Neu	Neue Datei anlegen. Dazu wird wieder die Auswahl des Petrinetztypes bzw. der Quellcodeerstellung angezeigt.
Quellcodeerzeugungsdatei laden	Lädt eine schon vorhandene Quellcodeerzeugungsdatei mit Anhang *.pns.
Quellcodeerzeugungsdatei speichern	Speichert die aktuelle Quellcodeerzeugungsdatei unter dem letzten verwendeten Dateinamen.
Quellcodeerzeugungsdatei speichern unter	Speichert die aktuelle Quellcodeerzeugungsdatei unter einem neuen Dateinamen.
Beenden	Schließt das komplette Programm

Tabelle 2: Menüpunkt Datei

Menüpunkt	Beschreibung
Prozesse aus Farben erzeugen	Erzeugt anhand der Farbcodierung innerhalb des Petrinetzes neue Prozesse. Dabei werden alle Objekte mit der gleichen Farbe einem Prozess zugeordnet.
Alle Quellcodeerzeugungsdaten löschen	Setzt alle Daten, welche für die Quellcodeerzeugung eingegeben wurden zurück. Danach existiert nur das reine Petrinetz ohne Informationen über Prozesse, Schleifen, Aktionen und Synchronisationsmittel.
Alle Objekte aus Prozessen entfernen	Entfernt alle Objektzuordnungen von Prozessen, so dass hinterher kein Objekt mehr einem Prozess zugeordnet ist.

Tabelle 3: Menüpunkt Hilfsfunktionen

Menüpunkt	Beschreibung
Allgemeine Daten	Zeigt ein Fenster an, in dem die allgemeinen Daten, welche für die Quellcodeerzeugung wichtig sind, eingegeben werden können. Diese Daten sind bibliotheksabhängig, daher kann der Aufbau des Fensters unterschiedlich sein.
Quellcodeerzeugung starten	Startet die Quellcodeerzeugung.
Testlauf starten	Startet einen Testlauf, welcher analysiert, ob alle nötigen Informationen für eine Quellcodeerstellung vorhanden sind. Diese Funktion ist bibliotheksabhängig und kann für jede Bibliothek unterschiedlich ausfallen.
Fehlerfenster anzeigen	Der Testlauf erzeugt ein Fenster mit allen, innerhalb des Petrinetzes aufgetretenen Fehlern. Sofern dieses Fenster geschlossen wurde, kann es mit dieser Menüfunktion erneut angezeigt werden.

Tabelle 4: Menüpunkt Quellcodeerzeugung

## 8 Eingabefenster

Fast alle Fenster, welche zur Eingabe von Daten auffordern, sind bibliotheksabhängig. Sofern eine Eingabe verlangt wird, gibt ein Textfeld eine kurze Information, welche Daten eingegeben werden sollen. Wird der Mauszeiger längere Zeit über dem Eingabefeld gehalten, wird in einem längeren Tooltiptext eine konkretere Information über den Zweck dieser Eingabe dargestellt. Folgende Besonderheiten gelten bei der Eingabe von Daten:

**<id>** Für die Eingabe von Zahlenwerten (auch für Prozess-IDs bei Prozessen) kann der Wert `<id>` statt einer Zahl eingegeben werden. Dieser Wert stellt bei der Quellcodeerzeugung die ID des Prozesses dar, zu dem das Objekt gehört, dessen Daten gerade eingegeben werden.

**<var>** Für die Eingabe von Zahlenwerten (auch für Prozess-IDs bei Prozessen) stellt der Wert `<var>` irgendeinen Variablenwert dar. Das heißt, der Wert ist kein konkreter numerischer Wert, sondern wird im Quelltext durch eine Variable repräsentiert.

Die Umsetzung dieser beiden Eingaben in Quellcode kann von Bibliothek zu Bibliothek unterschiedlich ausfallen.

## 9 Vorgehensweise

Nach dem Laden eines Petrinetzes und der gewünschten Bibliothek sollten folgende Punkte abgearbeitet werden, um Quellcode für ein Petrinetz zu erstellen:

### 9.1 Allgemeine Daten eingeben

Über den Menüpunkt Quellcodeerzeugung->Allgemeine Daten wird ein Fenster geöffnet, welches die Eingabe der allgemeinen Daten wie Pfade der Quellcodeerstellung und ähnliches eingetragen werden können. Der konkrete Aufbau des Fensters ist dabei von der geladenen Bibliothek abhängig.

### 9.2 Prozesse erzeugen/löschen und Objekte zuordnen

Prozesse werden über die Buttons "neuer Prozess" angelegt. Dabei entsteht ein leerer Prozess mit einem vom Programm vorgegebenen Namen. Der aktuell im Baum ausgewählte Prozess kann per Klick auf "Prozess löschen" entfernt werden. Dabei werden alle, dem Prozess zugeordneten Objekte wieder freigegeben. Der Button "Objekt hinzufügen" fügt alle im Petrinetzfenster ausgewählten Objekte dem im Datenfenster aktuell gewählten Prozess hinzu. Der Button "Objekt löschen" entfernt alle Zuordnungen, der im Petrinetzfenster ausgewählten Objekte.

Hinweis: Ein einfacher Weg, effektiv Prozesse mit deren Objekten zu erstellen, ist der Menüpunkt Hilfsfunktionen->Prozess aus Farben erzeugen, welche anhand der Farbcodierung der Petrinetzobjekte Prozesse anlegt und diesen die Objekte zuordnet.

### 9.3 Prozessdaten eingeben

Per Doppelklick auf einen Prozess im Datenfenster öffnet sich das Fenster für die Prozessdaten. In dieses müssen grundlegend folgende 3 Werte eingegeben werden:

- Der Bezeichner des Prozesses.

- Die Anzahl der konkret zu erzeugenden Prozesse.
- Der Startplatz des Prozesses, bei dem die Quellcodeerzeugung beginnt.

Wird der Startplatz durch eine Transition erneut erreicht, deutet dies auf einen zyklischen Prozess hin. Wird der Startplatz nicht erneut erreicht, wird der Prozess nur einmalig ausgeführt und ist nicht zyklisch. Je nach Bibliothek können noch andere Daten von der Bibliothek angefordert werden. Diese sollten in diesem Schritt ebenfalls eingegeben werden.

## 9.4 Daten für Synchronisationsplätze eintragen

Per Doppelklick auf einen Synchronisationsplatz (Platz, welcher nicht zu einem Prozess gehört) im Petri-netzfenster bzw. Datenfenster wird ein Fenster wie in Abbildung 5 geöffnet. In diesem kann in dem entsprechenden Dropdown-Feld "Synchronisationsmittel" ein Synchronisationsmittel gewählt werden, welches für die Quellcodeerstellung für diesen Platz verwendet werden soll. Die angebotenen Arten der Synchronisationsmittel hängen dabei von der Bibliothek ab. Zusätzlich könnten je nach Synchronisationsmittel im Bereich "Parameter" weitere Daten abgefragt werden. Diese sollten in diesem Schritt ebenfalls eingegeben werden.



Abbildung 5: Fenster Synchronisationsmittel

## 9.5 Schleifen/Verzweigungen zuordnen

Ein Platz, zu dem mehr als **eine** eingehende Kante führt, deutet auf einen Schleifenplatz hin. Ein Platz, der mehr als **eine** ausgehende Kante besitzt, deutet auf einen Verzweigungsplatz hin. Dabei kann ein und derselbe Platz gleichzeitig Schleifenplatz und Verzweigungsplatz sein. Per Doppelklick auf einen Schleifenplatz wird das in Abbildung 6 dargestellte Fenster für die Schleifenbearbeitung geöffnet. In diesem ist per Dropdown-Feld "Schleifentyp" auswählbar, welche Art der Schleife verwendet werden soll. Ist der entsprechende Platz trotz mehrerer eingehender Kanten keine Schleife, muss der Typ "keins" ausgewählt werden. Je nach Schleifentyp können im Parameterfenster weitere Daten angefordert werden. Diese sollten ebenfalls in diesem Schritt vervollständigt werden.

Per Doppelklick auf einen Verzweigungsplatz wird das in Abbildung 7 dargestellte Fenster geöffnet. In diesem kann per Dropdown-Feld der Verzweigungstyp ausgewählt werden. In der Liste "Pfade" muss die Reihenfolge eingestellt werden, in welcher die Pfade erzeugt werden sollen. Das heißt: Der erste Eintrag wird der If-Teil einer Verzweigung. Der zweite Eintrag wird zum Else-If-Teil. Der letzte Eintrag wird der Else-Teil.

Sofern ein Platz gleichzeitig Schleifenplatz und Verzweigungsplatz sein kann, werden die beiden genannten Fenster nacheinander geöffnet.



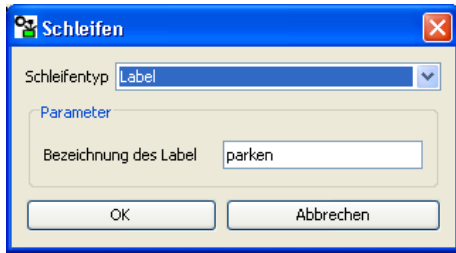


Abbildung 6: Beispiel Schleifenfenster

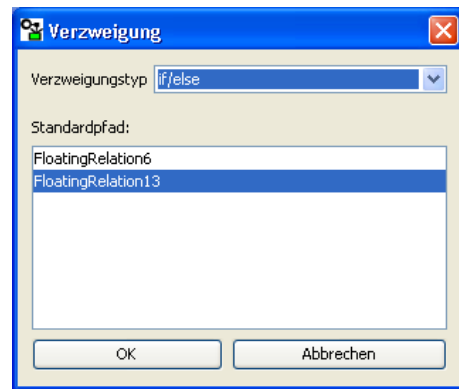


Abbildung 7: Beispiel Petrinetzfenster

## 9.6 Aktionen zuordnen

Per Doppelklick auf eine Transition, welche Verbindung mit einer oder mehreren Synchronisationsplätzen aufweist, wird das in Abbildung 7 dargestellte Fenster geöffnet. In diesem sind im zentralen Bereich alle Verbindungen zu entsprechenden Synchronisationsplätzen aufgelistet. Die Reihenfolge der Tabelleneinträge entspricht dabei der späteren Abarbeitungsreihenfolge der Befehle. Per Rechtsklickmenü können Funktionen ausgewählt werden, um die aktuell markierte Aktion nach oben oder unten zu schieben (Dies funktioniert nur, wenn wirklich eine konkrete Aktion zugeordnet wurde, andernfalls wird beim Schließen die Position nicht gespeichert). Im Petrinetzfenster wird die aktuell markierte Transition/Aktion zur besseren Visualisierung markiert. Per Doppelklick auf einen Eintrag wird das Fenster zum Eintragen einer konkreten Aktion geöffnet. Sofern eine Transition direkt nachfolgend auf eine Verzweigung ist, kann noch eine Bedingung für den Verzweigungspfad definiert werden. Dazu muss der Haken bei "Bedingung aus Rückgabewert erstellen" gesetzt und ein String, welcher die Bedingung repräsentiert, eingetragen sein. Dabei wird die Bedingung aus dem Rückgabewert der ersten Aktion in der Tabelle und dem Bedingungsstring erzeugt. Folgendes Beispiel auf Abbildung 8 verdeutlicht dies:

"wartet" ist eine Verzweigung, deren Reihenfolge "fährt aus","fährt zurück" ist. Fährt aus ist ein bedingter Pfad, bei dem eine Bedingung vorgegeben ist. Der Vergleichsterm ist `==1` und die erste Aktion wäre `rt_receive_until`. Dieser Befehl gibt 1 zurück, wenn innerhalb einer definierten Zeit etwas empfangen wurde. Andernfalls wird 0 zurückgegeben. In diesem Fall würde eine Verzweigung folgenden Typs erzeugt werden:

```
if ( rt_receive_until(...) == 1 )
{
    //Weitere Aktionen für fährt aus
}
else
{
    //ggf Aktionen für fährt zurück
}
```

Jede weitere Aktion wird innerhalb des entsprechenden Verzweigungspfades eingliedert.



Abbildung 8: Beispiel für Verzweigungen mit bedingten Aktionen

Die Abbildung 9 zeigt das Fenster, welches beim Doppelklick auf einen Eintrag in der Liste für Flussrelationen/Aktionen geöffnet wird. In diesem kann nun per Dropdown-Feld eine passende Aktion ausgewählt werden. Dabei sind nur solche Aktionen vorgegeben, welche zur Richtung der Flussrelation passen (Zu einem Synchronisationsplatz hinführende Flussrelation gilt als Lock-Aktion. Von einem Synchronisationsplatz wegführende Flussrelation gilt als Unlock-Aktion). Im Bereich Parameter sollten in diesem Schritt alle weiteren Parameter der Aktion eingetragen werden.

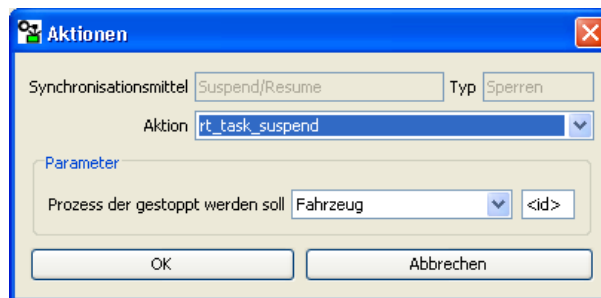


Abbildung 9: Beispiel Aktionenfenster

## 9.7 Testlauf starten

Per Menüpunkt Quellcodeerstellung->Testlauf starten wird geprüft, ob alle eingegebenen Informationen in einer Form vorliegen, welche für eine Quellcodeerzeugung nötig ist. Dabei ist zu beachten, dass die Funktionsweise dieses Menüpunktes von der geladenen Bibliothek abhängt. Ggf. entfällt eine Auswertung, ob das Petrinetz strukturell korrekt ist (Bei der mitgelieferten RTAI/C-Bibliothek ist dies der Fall). Entsprechend gefundene Fehler oder Hinweise werden in einem Fehlerfenster angezeigt. Per Doppelklick auf einen Eintrag in diesem Fenster wird das Objekt, welches zu dem Fehler führte, markiert.

## 9.8 Quellcode erstellen

Durch einen Klick auf den Menüpunkt Quellcodeerstellung->Quellcodeerzeugung starten wird die Quellcodeerzeugung ausgeführt. Dabei wird ein Statusfenster angezeigt. In diesem sind Informationen über den Fortschritt der Codeerzeugung aufgelistet. Kritische Fehler werden direkt mit einer Warnung in einer Nachrichtenbox dargestellt sowie das auslösende Objekt markiert. Je nach Implementation der Bibliothek befindet sich die erzeugte(n) Quellcodedatei(en) nun in einem bestimmten Ordner.