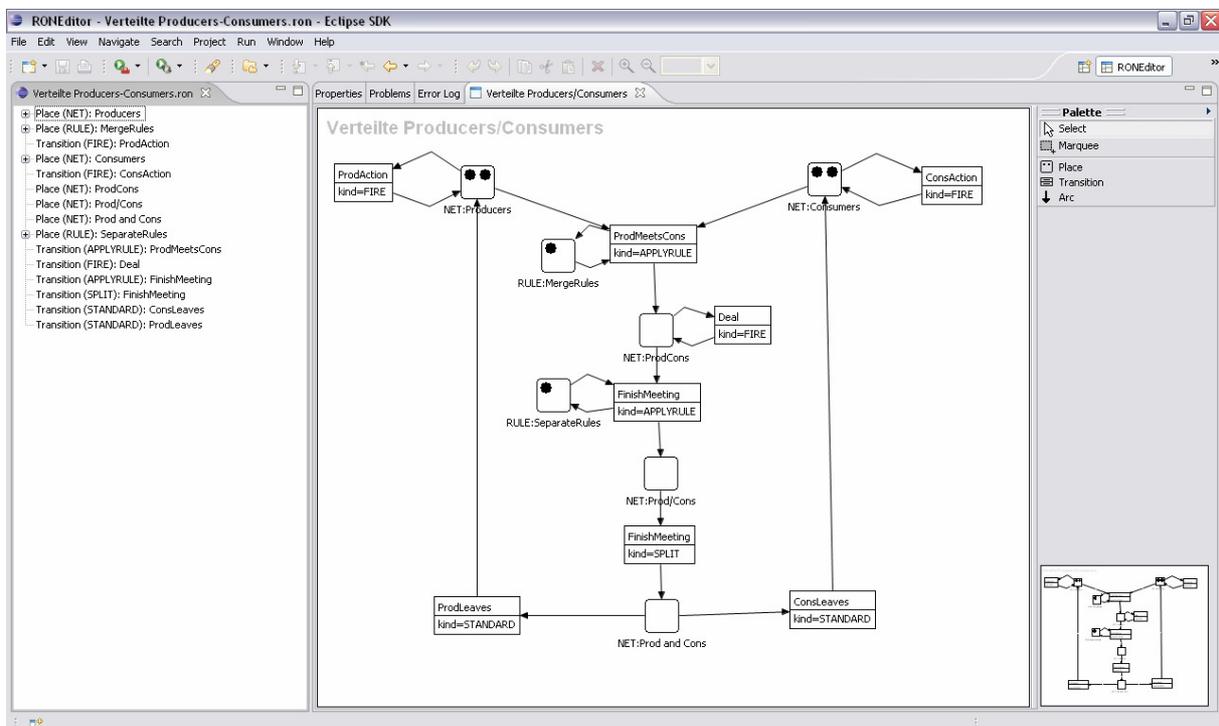


RONEditor

Handbuch

Autoren: Gayass Daher
Louay Bassbous
Waldemar Lichtner
Artem Kähm



Hauptstudiumsprojekt

Visuelle Sprachen – Projekt SS07

Veranstalter

Claudia Ermel
Frank Hermann

Inhaltsverzeichnis

1 Einführung.....	3
2 Editor.....	4
2.1 Erstellen einer RON-Datei.....	4
2.2 TreeEditor.....	6
2.3 Views im Überblick.....	6
3 ObjectNetView.....	8
3.1 ObjectNet erzeugen.....	8
3.2 Places.....	9
3.2.1 Erstellen von Places.....	9
3.2.2 Editieren von Places.....	9
3.3 Transitions.....	10
3.3.1 Erstellen von Transitions.....	10
3.3.2 Editieren von Transitions.....	10
3.3.3 Schalten von Transitions.....	11
3.4 Arcs.....	11
3.4.1 Erstellen von Arcs.....	11
3.4.2 Editieren von Arcs.....	12
3.4.3 BEndpoints.....	12
4 RuleView.....	13
4.1 Komponenten einer Regel.....	13
4.2 Neue Regel erzeugen.....	14
4.3 Kopieren von NAC/LHS/RHS.....	15
5 RONView.....	16
5.1 HLPlaces.....	16
5.1.1 HLPlaces erzeugen.....	16
5.1.2 Tokens in HLPlaces erzeugen.....	17
5.2 HLTransitions.....	17
5.2.1 HTRansitions erzeugen.....	17
5.2.2 Schalten von HLTransitions.....	18

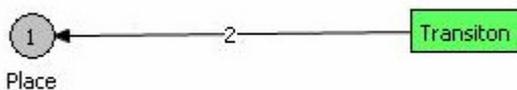
1 Einführung

Dieser Editor ist im Rahmen des Hauptstudiumsprojekts **Visuelle Sprachen-Projekt** an der Technischen Universität Berlin im SS2007 entstanden. Es handelt sich um einen visuellen Editor für eine regelbasierte Petrinetz-Modellierungssprache (Rekonfigurierbare Objektnetze, kurz RONS).

Die Modellierungssprache für RONS, für die wir einen Editor entworfen haben, umfasst High-Level-Petrinetze (RONS) mit zwei Arten von Stellen. Die eine Stellensorte wird mit Stellen-Transitions-Petrinetzen (Objektnetzen) markiert, die zweite Stellensorte mit Netz-Transformations-Regeln. Beim Schalten einer RON-Transition können so Objektnetze durch Anwendung von Netz-Transformations-Regeln manipuliert werden. RONS eignen sich daher hervorragend zur Modellierung verteilter Vorgänge, bei denen einzelne Prozessdefinitionen im laufenden System angepasst / verändert / synchronisiert werden müssen.

Jetzt aber zum Begriff Objektnetze (Petrinetze):

Ein Petri-Netz ist ein bipartiter und gerichteter Graph. Er besteht aus Stellen (Places) und Übergängen bzw. Transitionen (Transitions). Stellen und Transitionen sind durch gerichtete Kanten verbunden. Es gibt keine direkten Verbindungen zwischen zwei Stellen oder zwei Transitionen.



Stellen werden als Kreise, Transitionen als Rechtecke dargestellt (siehe Bild). Jede Stelle hat eine Kapazität und kann entsprechend viele Token enthalten. Ist keine Kapazität angegeben, steht das für unbegrenzte Kapazität. Jeder Kante ist ein Gewicht zugeordnet, das die Kosten dieser Kante festlegt. Ist einer Kante kein Gewicht zugeordnet, wird der Wert eins verwendet. Transitionen sind aktiviert bzw. schaltbereit, falls sich in allen Eingangsstellen mindestens so viele Marken befinden, wie die Transitionen Kosten verursacht und alle Ausgangsstellen noch genug Kapazität haben, um die neuen Marken aufnehmen zu können. Schaltbereite Transitionen können zu einem beliebigen Zeitpunkt schalten. Beim Schalten einer Transition werden aus allen Eingangsstellen entsprechend der Kantengewichte Marken entnommen und bei allen Ausgangsstellen entsprechend der Kantengewichte Marken hinzugefügt.

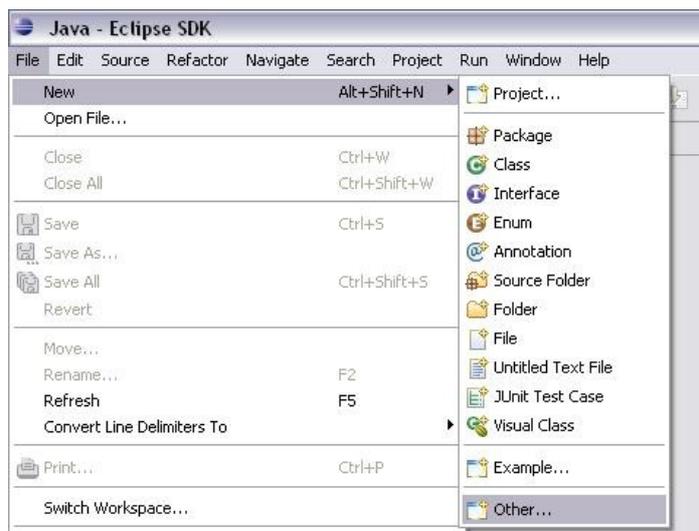
2 Editor

Zu Beginn wollen wir erklären, wie man eine RON-Datei erzeugt und wie man in dem sich dann öffnenden TreeEditor die Bestandteile der Dateien erstellt. Es wird ein kurzer Überblick über die verschiedenen Views gegeben, die gebraucht werden, um den RONEditor zu benutzen.

2.1 Erstellen einer RON-Datei

Jede Datei kann nur ein RONnetz besitzen, deswegen ist die Erstellung eines neuen RONs gleichbedeutend mit dem Erzeugen einer neuen .ron Datei. Zum Erstellen einer neuen .ron Datei und somit eines neuen RONs muss wie folgt vorgegangen werden.

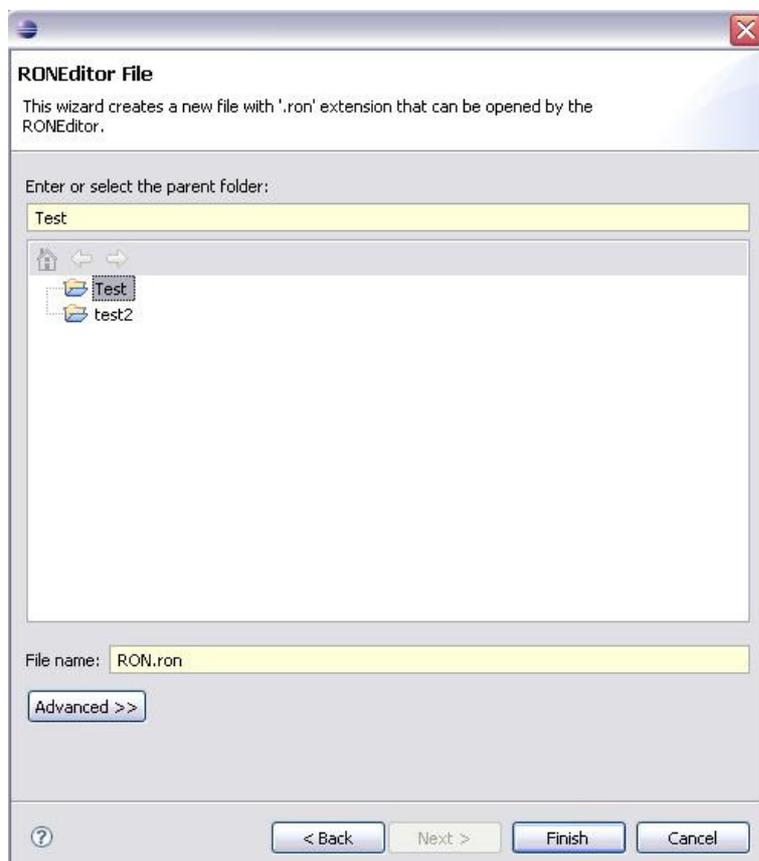
Auswahl des RON File Creation Wizard im Menü unter File -> New -> Other



dann Other -> RON File Creation Wizard -> Next



dann Parent-Directory und Name der ron-Datei auswählen und Finish drücken



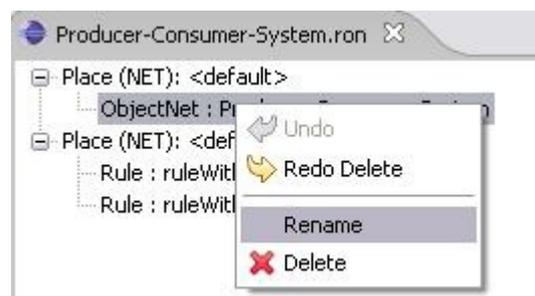
Danach wird eine neu erstellte Datei im RONEditor geöffnet:



2.2 TreeEditor

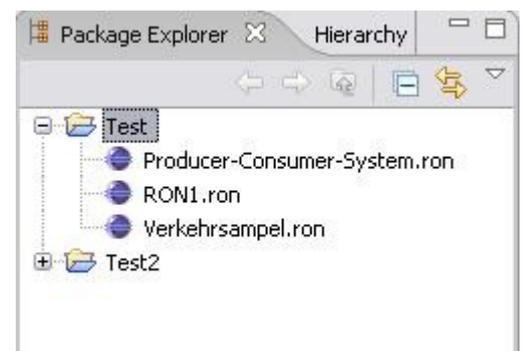
Der TreeEditor bezeichnet das Fenster in dem alle Teile der geöffneten RON-Datei angezeigt werden, das heißt, es werden HLPlaces, die vom Typ NET (d.h. die beinhalten Objektnetze als Token) oder vom Typ RULE (d.h. die beinhalten Regeln als Token) sein können, und HLTransitions, die vom Typ Standard, Fire, Split oder Applyrule sein können, aufgelistet. Durch einen Doppelklick auf eines der Bestandteile wird ein Fenster geöffnet bzw. in den Vordergrund gebracht, das die dementsprechende Darstellung der Auswahl beinhaltet.

Wenn man mit der rechten Maustaste auf ein Bestandteil klickt, dann kann man es auch umbenennen bzw. löschen.



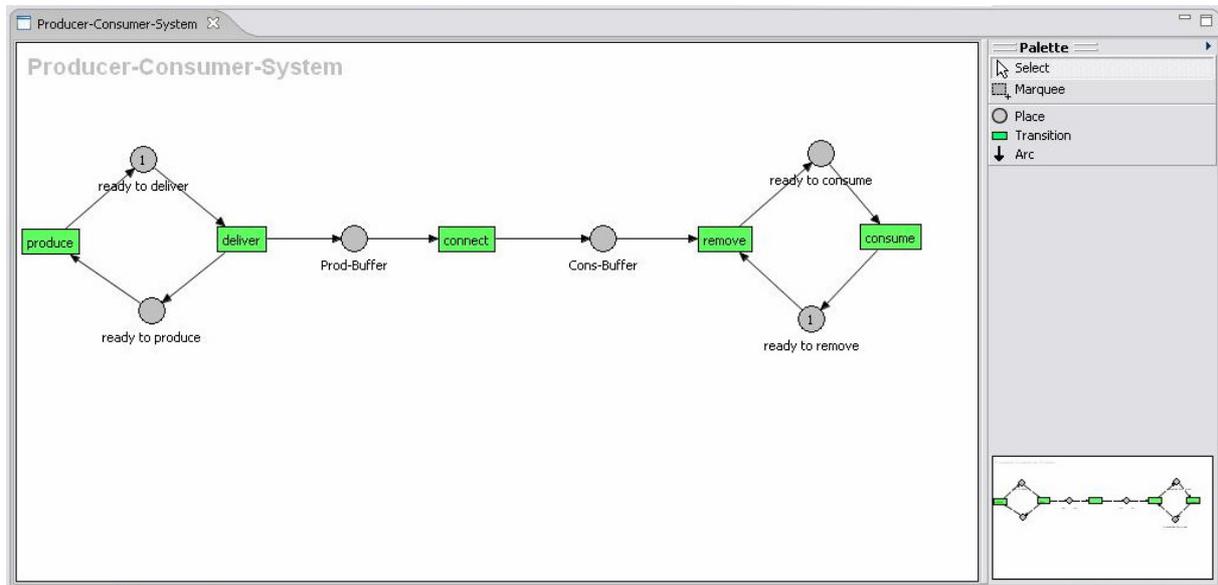
2.3 Views im Überblick

- *Navigator*: hier werden die RON-Datei und die angelegten Verzeichnisse angezeigt. Eine bestehende Datei kann man durch einen Doppelklick öffnen. Eine neue RON-Datei lässt sich hier auch erstellen (siehe Abschnitt 2.1). Sollte dieser View nicht offen sein, lässt er sich öffnen über die Menüauswahl Windows->Show View-> Navigator.



- *TreeEditor*: Der TreeEditor zeigt alle Bestandteile einer geöffneten RON-Datei an. Er öffnet sich, wenn man eine Datei im Navigator öffnet. In ihm kann man den ObjectNetView und RuleView öffnen um einen Teil der Datei zu editieren. Mehr dazu in Abschnitt 2.2.
- *ObjectNetView*: In diesem View wird ein Objektnetz angezeigt und kann bearbeitet werden. Er wird über einen Doppelklick im TreeEditor auf das entsprechende ObjectNet geöffnet. Mehr zu diesem View in Abschnitt 3.
- *RuleView*: Hier werden die Elemente der Regel angezeigt und editiert. Siehe dazu Abschnitt 4. Er wird über einen Doppelklick im TreeEditor auf die entsprechende Rule geöffnet.
- *Properties*: In diesem Fenster werden die Eigenschaften zu einem ausgewählten Objekt angezeigt. Diese sind unterschiedlich, je nachdem getroffenen Auswahl; Von der Anzeige des Namens beim Selektieren eines Eintrages im TreeEditor bis zum Anzeigen des Namens, der Anzahl der Tokens und der Position in dem Graphen bei Auswahl eines Knotens im ObjectNetView. Dieser View ist standardmäßig immer geöffnet. Sollte dieser View nicht offen sein, ist er zu öffnen über die Menüauswahl Windows -> Show View -> Properties.
- *ProblemView*: Hier werden mögliche Probleme angezeigt, die dadurch entstehen, dass z.B.eine Regel falsch editiert wurde. Mehr dazu bei der Beschreibung von RuleView. Er ist über die folgende Menüauswahl zu öffnen: Windows -> Show View -> Problems.

3 ObjectNetView

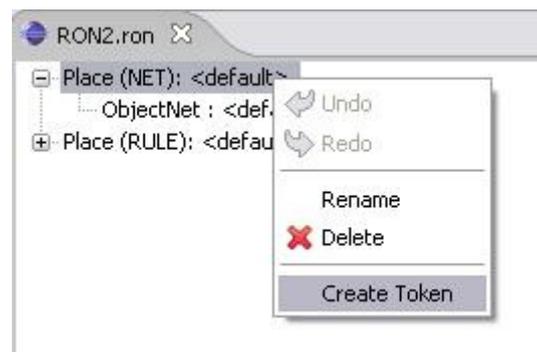


Hier werden die Places und Transitions, sowie die Verbindungen (Arcs) zwischen ihnen, erzeugt und bearbeitet.

Um die Palette zu öffnen, so wie in der obigen Abbildung, muss man mit der linken Maustaste auf den Streifen am rechten Rand des ObjectNetViews klicken.

3.1 ObjectNet erzeugen

Um neues Objektnetz zu erzeugen muss man im TreeEditor ein HLPlace vom Typ NET auswählen und dann in dem mit Rechtsklick geöffneten Menü „Create Token“ auswählen. Man gibt dann im geöffneten Menü einen Namen ein und das neue Objektnetz ist erzeugt.



3.2 Places

3.2.1 Erstellen von Places

Places werden erstellt indem man mit der linken Maustaste im Menü den Eintrag Place auswählt und irgendwo auf der freien Fläche des ObjectNetViews wieder mit der linken Maustaste klickt.



Dann erscheint auf dem Bildschirm ein Fenster mit der Aufschrift „Create ONPlace“. In diesem Fenster kann man Name und Anzahl der Tokens des Places festlegen.

Bemerkung: Name darf nicht leer sein.



Wenn man dann mit der linken Maustaste „OK“ angeklickt hat, erscheint auf dem Bildschirm ein Place mit vorher festgelegten Namen und Anzahl der Tokens. Mit „Cancel“ wird die Aktion abgebrochen.



3.2.2 Editieren von Places

Man klickt mit der rechten Maustaste auf einen Place und wählt in angezeigtem Kontextmenü „Properties Dialog“.



Dann erscheint auf dem Bildschirm ein Fenster mit der Aufschrift „Edit ONPlace“. In diesem Fenster kann man Name und Anzahl der Tokens des Places neu festlegen. Man klickt dann mit der linken Maustaste „OK“ und schon sind Name und Anzahl der Tokens geändert. Mit „Cancel“ wird die Aktion abgebrochen.



Wenn man einen Place mit der linken Maustaste anklickt und nicht losläßt, dann kann man ihn frei bewegen. Wenn man die linke Maustaste wieder losläßt, dann wird er auf neuer Position gespeichert.

3.3 Transitions

3.3.1 Erstellen von Transitions

Places werden erstellt indem man mit der linken Maustaste im Menü den Eintrag Transition auswählt und irgendwo auf der freien Fläche des ObjectNetViews wieder mit der linken Maustaste klickt.

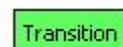


Dann erscheint auf dem Bildschirm ein Fenster mit der Aufschrift „Create ONTransition“. In diesem Fenster kann man Name der Transition festlegen.



Bemerkung: Name darf nicht leer sein.

Wenn man dann mit der linken Maustaste „OK“ angeklickt hat, erscheint auf dem Bildschirm eine Transition mit vorher festgelegten Namen. Mit „Cancel“ wird die Aktion abgebrochen.



3.3.2 Editieren von Transitions

Man klickt mit der rechten Maustaste auf eine Transition und wählt in angezeigtem Kontextmenü „Properties Dialog“.



Dann erscheint auf dem Bildschirm ein Fenster mit der Aufschrift „Edit ONTransition“. In diesem Fenster kann man Name der Transition neu festlegen. Man klickt dann mit der linken Maustaste „OK“ und schon ist Name geändert. Mit „Cancel“ wird die Aktion abgebrochen.



Wenn man eine Transition mit der linken Maustaste anklickt und nicht losläßt, dann kann man sie frei bewegen. Wenn man die linke Maustaste wieder losläßt, dann wird sie auf neuer Position gespeichert.

3.3.3 Schalten von Transitions

Man klickt mit der rechten Maustaste auf eine freie Fläche und wählt in angezeigtem Kontextmenü „Fire“. Wenn die Bedingungen für einen Schaltvorgang gegeben sind, dann wird ein Schaltvorgang simuliert. Wenn nicht, dann wird „Fire“ einfach ausgeblendet.



3.4 Arcs

3.4.1 Erstellen von Arcs

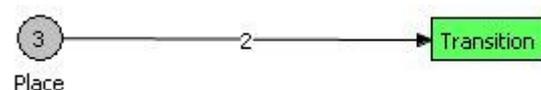
Arcs werden erstellt indem man mit der linken Maustaste im Menü den Eintrag Arc auswählt. Dann klickt man mit der linken Maustaste den Startknoten des Arcs an. Weiterhin klickt man den Zielknoten des Arcs.



Dann erscheint auf dem Bildschirm ein Fenster mit der Aufschrift „Create ONArc“. In diesem Fenster kann man die Anzahl der Tokens, die ein Arc durchlassen kann, festlegen.



Wenn man dann mit der linken Maustaste „OK“ angeklickt hat, erscheint auf dem Bildschirm ein Arc mit vorher festgelegter Anzahl der Tokens. Mit „Cancel“ wird die Aktion abgebrochen.



3.4.2 Editieren von Arcs

Man klickt mit der rechten Maustaste auf ein Arc und wählt in angezeigtem Kontextmenü „Properties Dialog“.



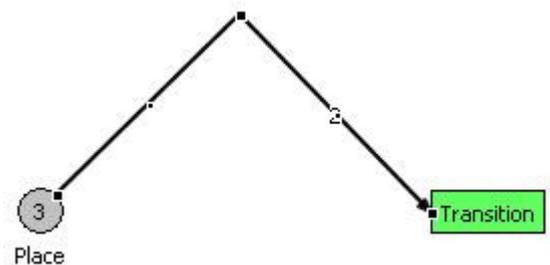
Dann erscheint auf dem Bildschirm ein Fenster mit der Aufschrift „Edit ONArc“. In diesem Fenster kann man Anzahl der Tokens neu festlegen. Man klickt dann mit der linken Maustaste „OK“ und schon ist Anzahl geändert. Mit „Cancel“ wird die Aktion abgebrochen.



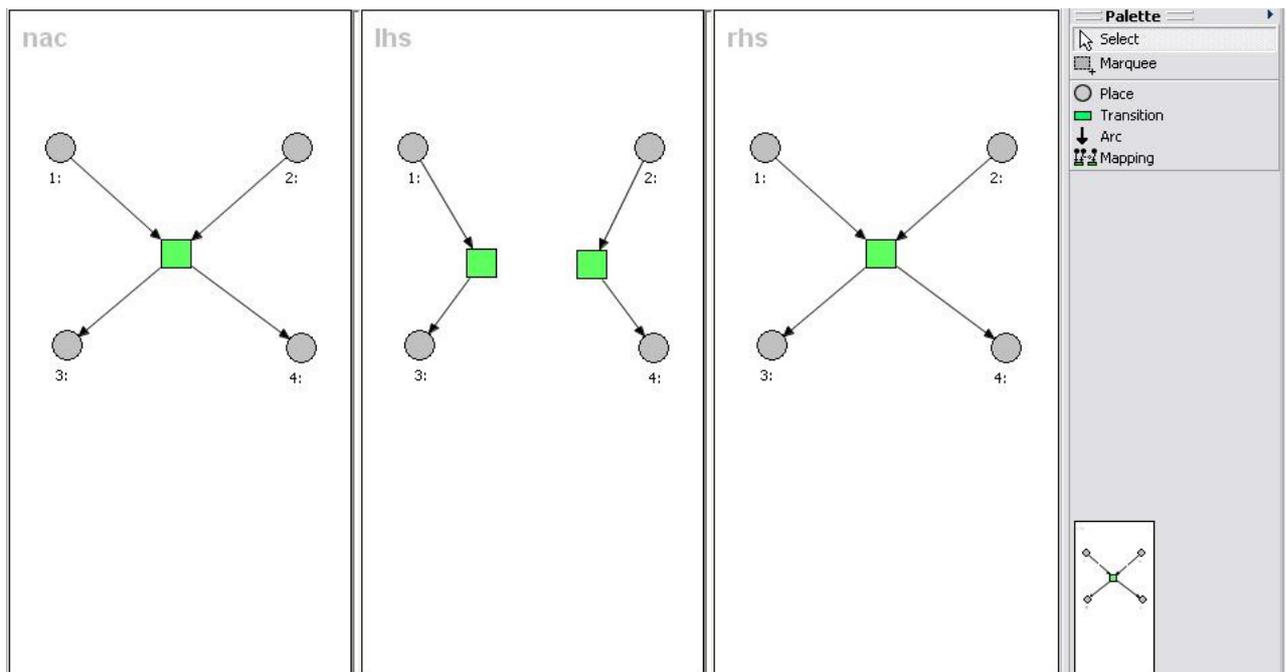
Wenn man das Ende eines Arcs mit der linken Maustaste anklickt und nicht losläßt, dann kann man Ziel bzw. Quelle des Arcs neu festlegen. Wenn man die linke Maustaste über einen Ziel bzw. einer Quelle wieder losläßt, dann wird sie auf neuer Position gespeichert.

3.4.3 BEndpoints

Die Arcs können nicht nur über die Strecke zwischen den beiden Knoten realisiert werden, sondern auch durch weitere Wegpunkte spezialisiert werden. Diese Wegpunkte werden auch BEndpoints genannt. Ein Bendpoint kann gesetzt werden, indem die Markierung auf der Mitte eines jeden geraden Arcs gesetzt wird, wodurch auf den dadurch entstehenden Arc-Hälften wiederum das setzen von BEndpoints eingerichtet wird. Zum löschen eines BEndpoints muss er einfach nur auf die Position eines benachbarten BEndpoints gezogen werden.



4 RuleView



In diesem Abschnitt wollen wir erklären wie man eine Regel erzeugt und weisen auf einige Dinge hin, die bei der Erstellung einer Regel wichtig sind. Das generelle Verständnis von Regeln wird vorausgesetzt.

4.1 Komponenten einer Regel

Eine Regel besteht aus verschiedenen Teilen, die in separaten Abschnitten eines Views dargestellt werden.

LHS (Left-hand side)

Die LHS definiert den Initialzustand vor dem Anwenden einer Regel. In der LHS wird ein Ausschnitt beschrieben, der sich in einem Objektnetz (ObjectNet) befinden muss, damit die Regel überhaupt angewendet werden kann.

RHS (Right-hand side)

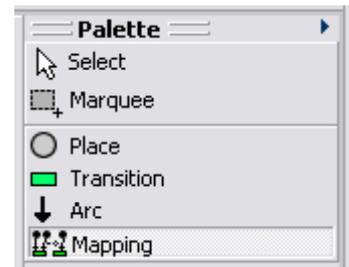
Die RHS beschreibt den Zustand nach Anwendung der Regel. Er zeigt an, wie man die LHS durch Anwendung der Regel verändern will.

NAC (Negative application condition)

Die NAC ist eine negative Anwendungsbedingung. Sie stellt einen Ausschnitt dar, der vor der Anwendung der Regel in Objektnetz (ObjectNet) nicht vorhanden sein darf.

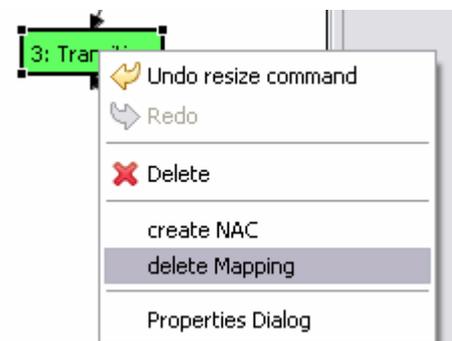
Mapping:

Ein Mapping bezeichnet eine Zuordnung von einer LHS der Regel auf einen anderen Teil. Dahinter verbirgt sich nicht mehr als die Aussage, dass man zwei Places oder Transitions, die in verschiedenen Teilen (NAC/LHS/RHS/) dargestellt sind, dem gleiche Symbol in einer Instanz entsprechen. Es kann aber keine Mappings zwischen den Places oder Transitions der RHS und einer NAC geben.



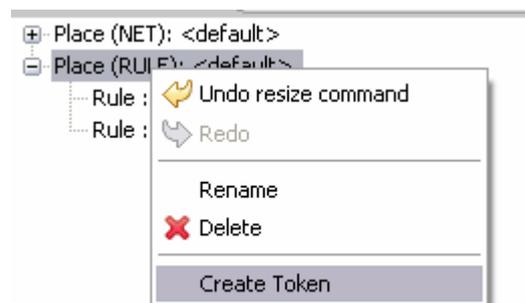
Um ein Mapping zu erzeugen wählt man „Mapping“ in der Palette aus und danach per Mausklick einen der Mapping- Partner, dann den anderen Mapping- Partner und damit ist das Mapping erstellt.

Durch einen Rechtsklick auf ein gemapptes Element können über den Punkt „delete Mapping“ sämtliche Mappings des selektierten Elements gelöscht werden.

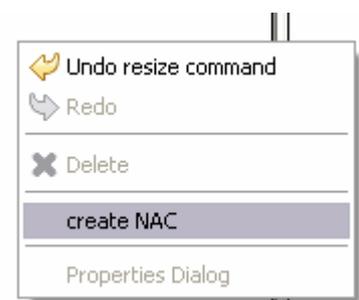


4.2 Neue Regel erzeugen

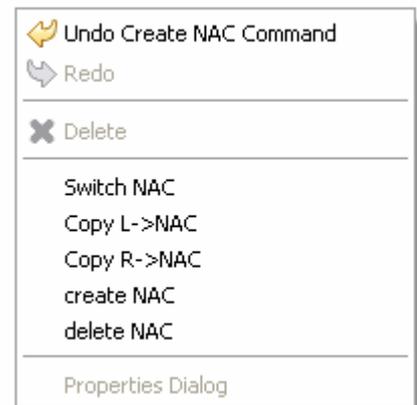
Um neue Regel zu erzeugen muss man im TreeEditor ein HLPlace vom Typ RULE auswählen und dann in dem mit Rechtsklick geöffneten Menü „Create Token“ auswählen. Man gibt dann im geöffneten Menü einen Namen ein und die neue Regel ist erzeugt.



Will man eine neue NAC zu einer Regel erzeugen, klickt man mit der rechten Maustaste irgendwo auf die freie Fläche der Regel, wählt dann im geöffneten Menü „create NAC“ aus.



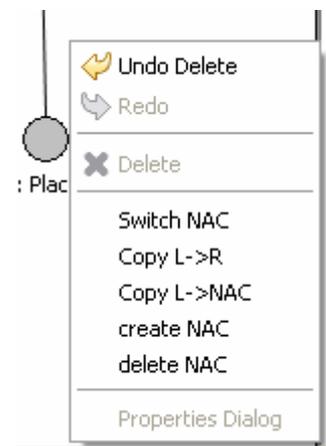
Um zwischen NACs zu blättern, klickt man mit der rechten Maustaste irgendwo auf die freie Fläche der Regel, wählt dann im geöffneten Menü „Switch NAC“ aus.



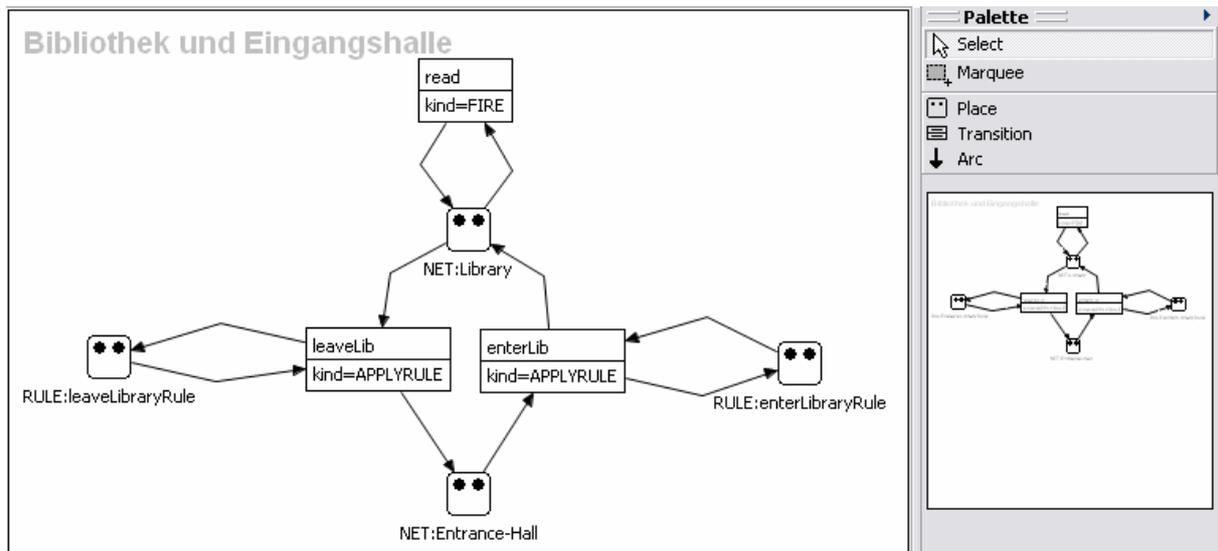
4.3 Kopieren von NAC/LHS/RHS

Aufgrund der generell relativ hohen Übereinstimmung von LHS und RHS ist es über das Kontextmenü der Regel möglich, die Elemente der LHS in die RHS zu kopieren, dabei werden unmittelbar alle Elemente miteinander gemappt.

Analog dazu befindet sich im Kontextmenü die Möglichkeit, alle Elemente der LHS oder RHS in die NAC zu kopieren. Die Mappings werden auch mitkopiert.



5 RONView



Hier werden die HLPlaces und HLTransitions, sowie die Verbindungen (HLArcs) zwischen ihnen, erzeugt und bearbeitet.

Um diese View zu öffnen, muss man im TreeEditor mit der linken Maustaste auf einen HLPlace (das sind Place(NET) oder Place(RULE)) oder auf eine HLTransition doppelklicken.

Um die Palette zu öffnen, so wie in der obigen Abbildung, muss man mit der linken Maustaste auf den Streifen am rechten Rand des RONViews klicken.

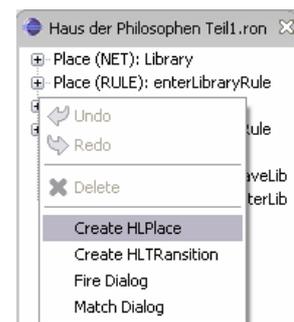
5.1 HLPlaces

5.1.1 HLPlaces erzeugen

HLPlaces werden erstellt indem man mit der linken Maustaste im Menü den Eintrag Place auswählt und irgendwo auf der freien Fläche des RONViews wieder mit der linken Maustaste klickt. Man wählt dann im geöffneten Menü Typ des HLPlaces (NET oder RULE) aus, gibt einen Namen ein und HLPlace ist erzeugt.

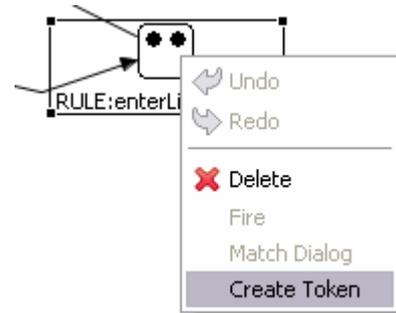


Man kann auch über TreeEditor HLPlaces erzeugen. Man klickt mit der rechten Maustaste auf einen Pfad im TreeEditor und wählt dann im geöffneten Kontextmenü „Create HLPlace“. Dann wählt man im geöffneten Menü Typ des HLPlaces aus, gibt einen Namen ein und HLPlace ist erzeugt.



5.1.2 Tokens in HLPlaces erzeugen

Man klickt mit der rechten Maustaste auf ein HLPlace und wählt dann im geöffneten Kontextmenü „Create Token“. Dann gibt man im geöffneten Menü einen Namen ein und schon hat HLPlace einen Token (siehe auch Abschnitte 3.1 und 4.2). Typ des Tokens hängt vom Typ des HLPlaces ab. HLPlace vom Typ NET kann nur Objektnetze als Tokens haben, HLPlace vom Typ RULE entsprechend nur Regeln.



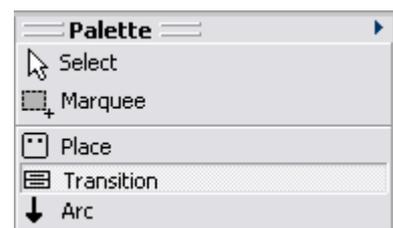
Mit einem Doppelklick auf einen Token öffnet sich die entsprechende View (für Objektnetze ObjectNetView, siehe Abschnitt 3, und für Regeln RuleView, siehe Abschnitt 4), wo man Tokens editieren kann (ObjektNetze siehe Abschnitte 3.2 bis 3.4 und Regeln siehe Abschnitte 4.1 und 4.3).

Man kann auch über TreeEditor Tokens erzeugen. Man klickt mit der rechten Maustaste auf einen HLPlace im TreeEditor und wählt dann im geöffneten Kontextmenü „Create Token“. Dann gibt man im geöffneten Menü einen Namen ein und schon hat HLPlace einen Token.

5.2 HLTransitions

5.2.1 HLTransitions erzeugen

HLTransitions werden erstellt indem man mit der linken Maustaste im Menü den Eintrag Transition auswählt und irgendwo auf der freien Fläche des RONViews wieder mit der linken Maustaste klickt. Man wählt dann im geöffneten Menü Typ der Transition (STANDARD, FIRE, APPLYRULE oder SPLIT) aus, gibt einen Namen ein und HLTransition ist erzeugt.

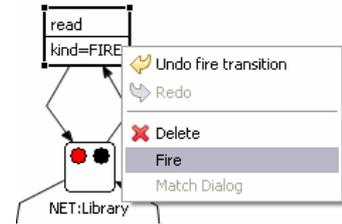


Man kann auch über TreeEditor HLTransitions erzeugen. Man klickt mit der rechten Maustaste auf einen Pfad im TreeEditor und wählt dann im geöffneten Kontextmenü „Create HLTransition“. Dann wählt man im geöffneten Menü Typ der HLTransition aus, gibt einen Namen ein und HLTransition ist erzeugt.

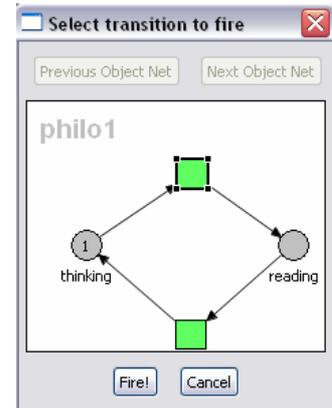


5.2.2 Schalten von HLTransitions

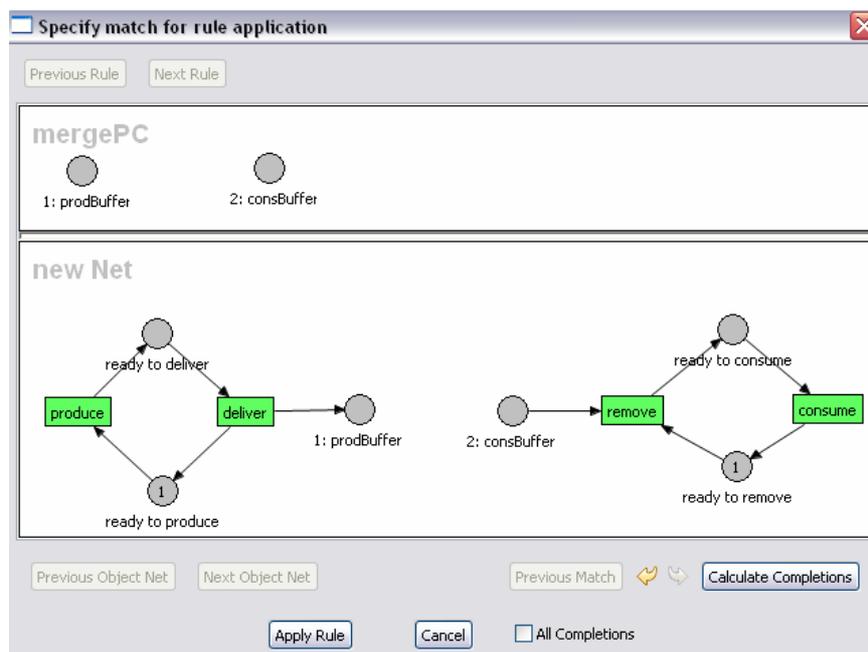
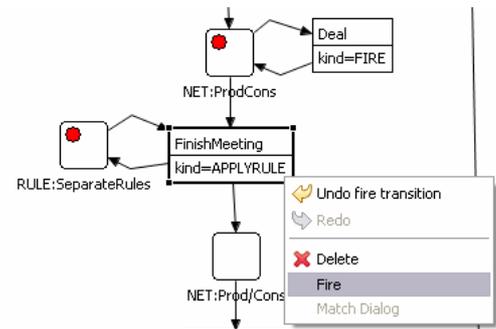
Man markiert zuerst einen Token, indem man mit der linken Maustaste auf ihn klickt, dann klickt man mit der rechten Maustaste auf eine HLTransition und wählt dann im geöffneten Kontextmenü „Fire“. Wenn die Voraussetzungen für einen Schaltvorgang gegeben sind, dann wird ein Schaltvorgang ausgeführt.



Wenn es sich um eine FIRE-Transition handelt, dann öffnet sich noch zusätzlich ein Dialogfenster „Select transition to fire“, wo man die Transition, die geschaltet werden sollen, noch auswählen kann. Nach der Auswahl klickt man auf „Fire!“, dann wird der Schaltvorgang ausgeführt.



Wenn es sich um eine APPLYRULE-Transition handelt, dann sollte man natürlich auch Regel, die angewendet werden soll, markieren. Es öffnet sich dann ein Match-Dialog, wo man Elemente auf die die ausgewählte Regel angewendet werden soll matchen kann. Dann klickt man auf „Apply Rule“ und Regel wird ausgeführt.



STANDARD HLTransition

bewegt Objektnetze durch das HL-Netz.

Funktionen: delete, copy, move, union

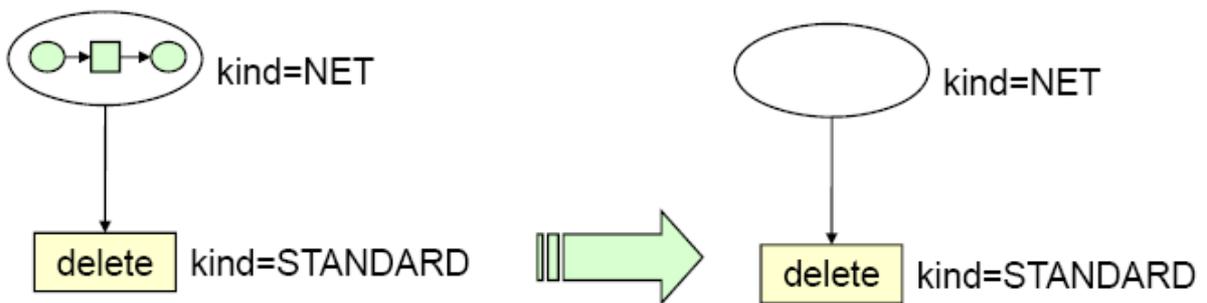
Vorbereich: höchstens eine RULE-Stelle, beliebig viele NET-Stellen

Nachbereich: beliebig viele RULE (NET) -Stellen, wenn mindestens eine RULE(NET) -Stelle im Vorbereich ist (sonst 0).

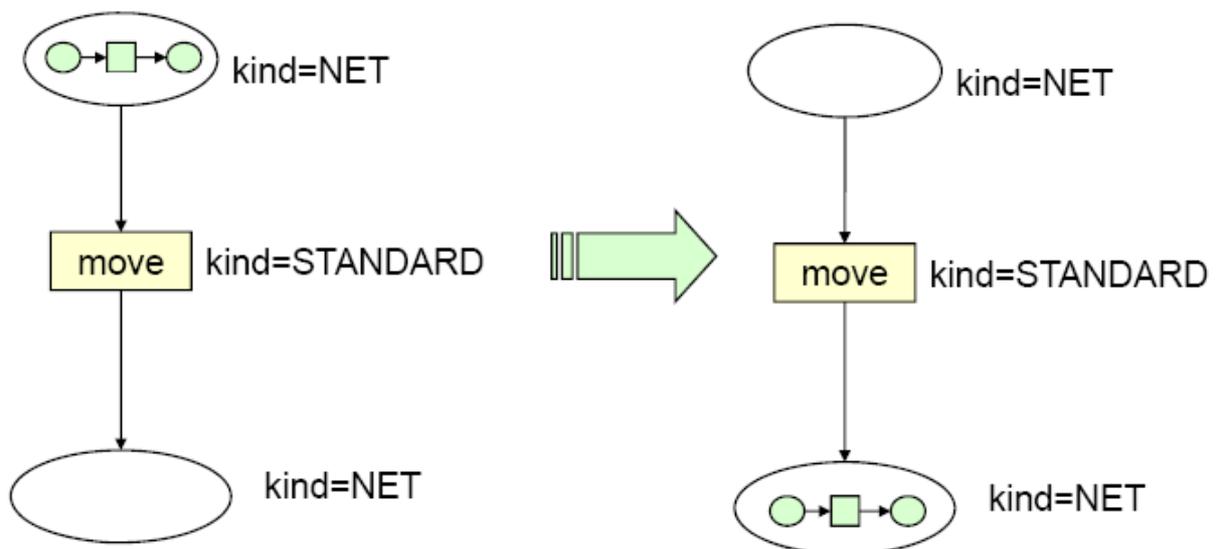
Aktivierung: mindestens ein Token auf jeder Vorbereichsstelle

Schaltverhalten: Von jeder Vorbereichsstelle wird ein Token entfernt. Auf alle NET-Stellen im Nachbereich wird die disjunkte Vereinigung aller Objektnetze, die dem Vorbereich entnommen wurden, gelegt. Auf alle RULE-Stellen im Nachbereich wird die Regel, die dem Vorbereich entnommen wurde, gelegt.

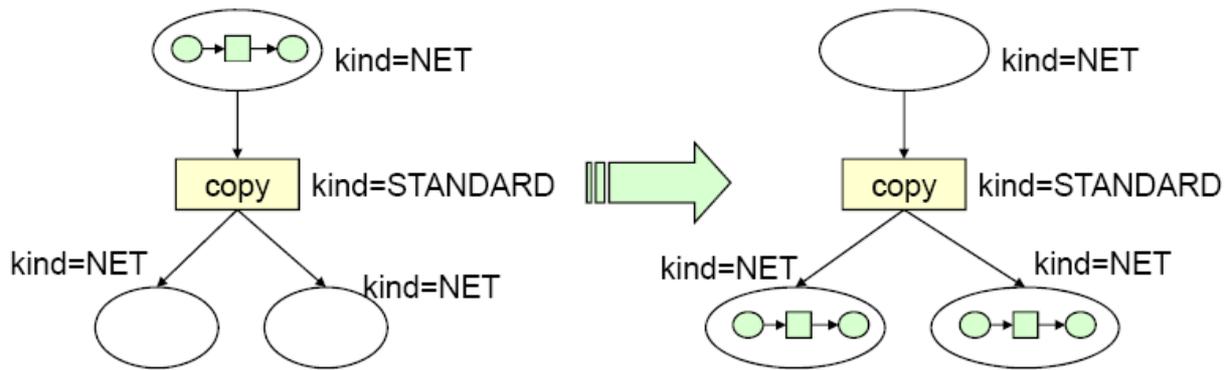
Beispiel Delete Net:



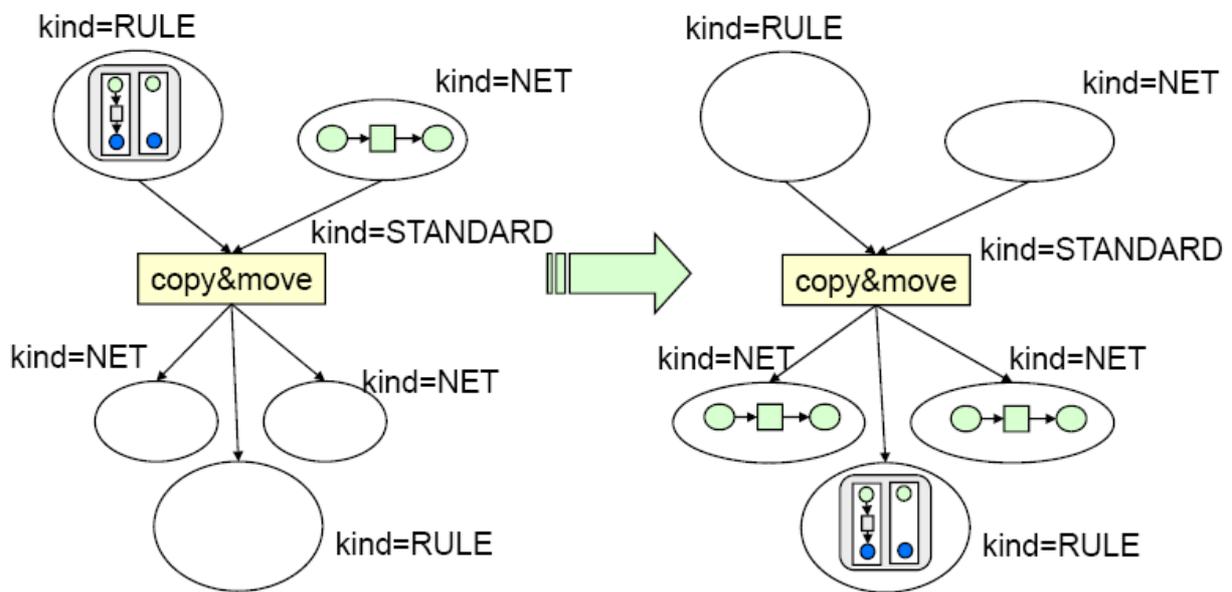
Beispiel Move Net:



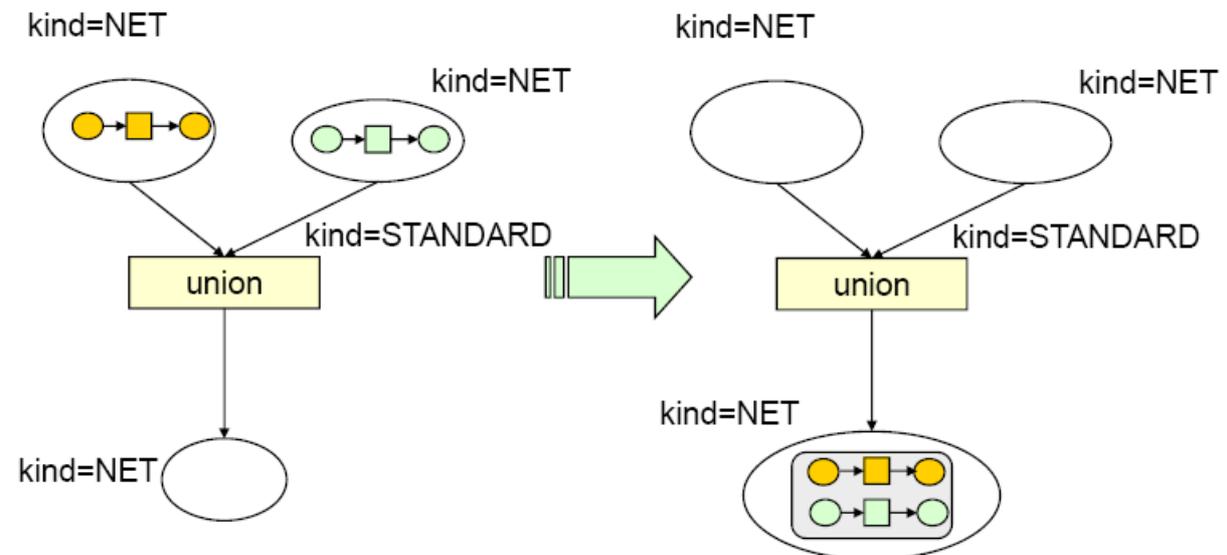
Beispiel Copy Net:



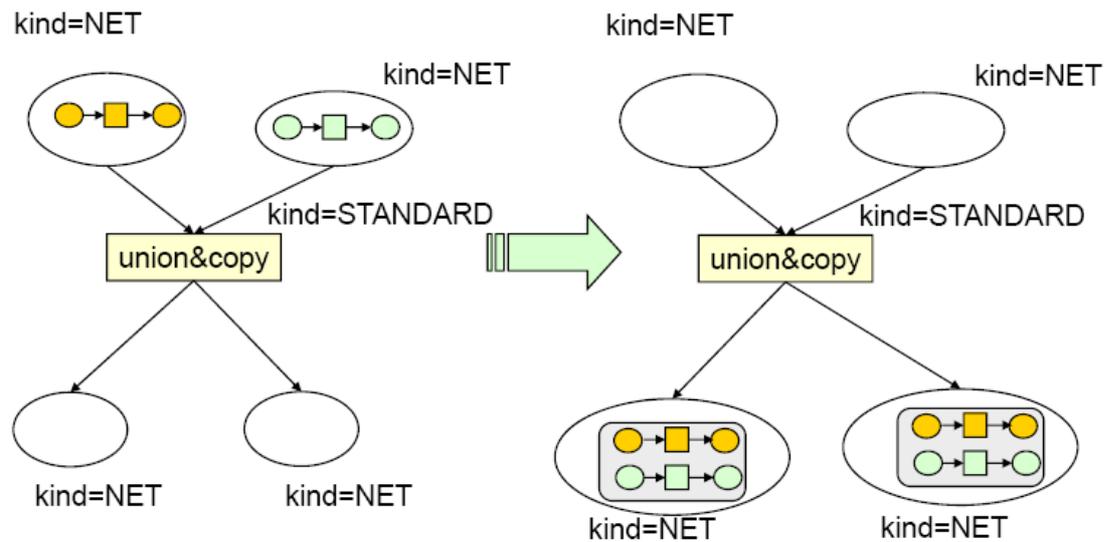
Beispiel Copy Net and Move Rule:



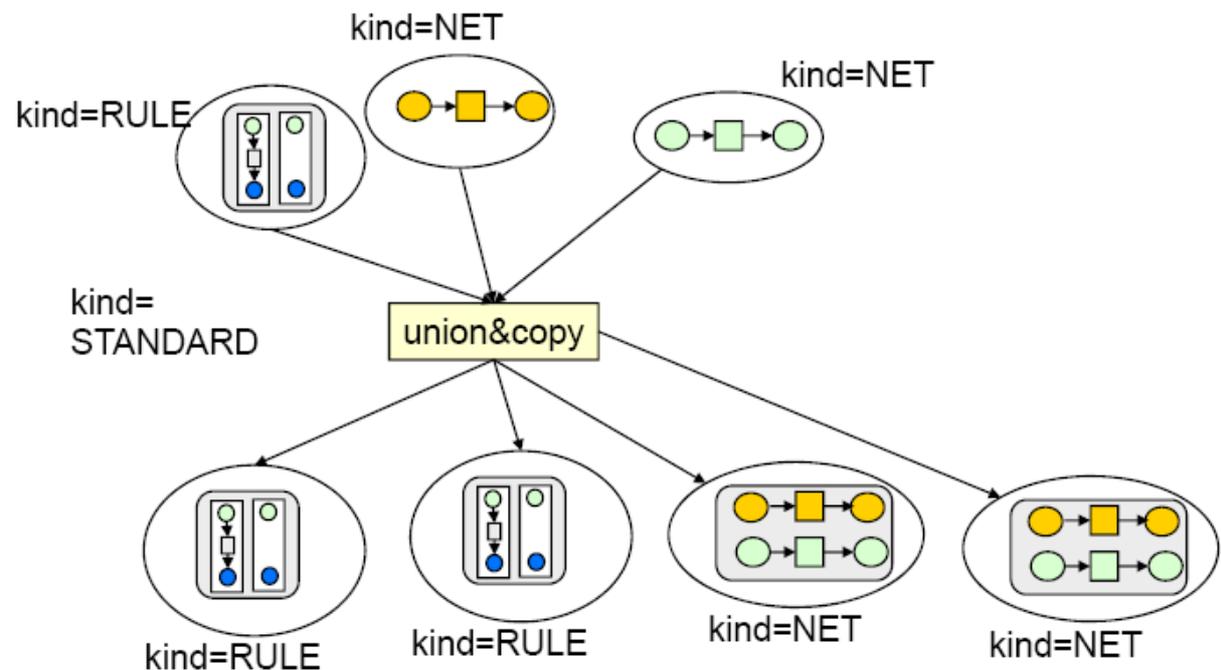
Beispiel: Union of Nets



Beispiel: Copy Union of Nets



Beispiel: Union and Copy of Nets and Copy of Rule



FIRE HLTransition

Schaltet eine Transition in einem Objektnetz

Vorbereich: eine NET-Stelle

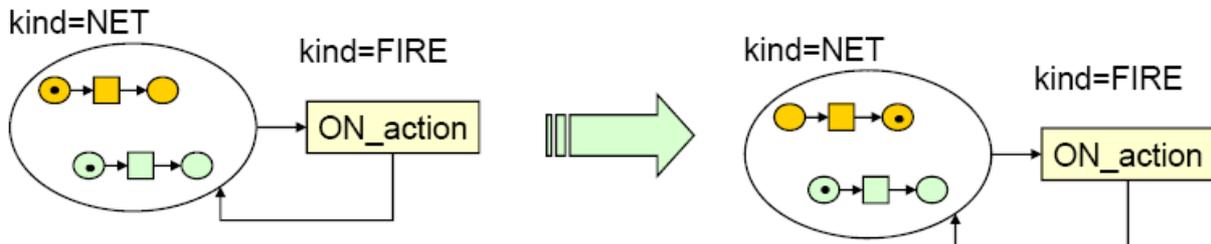
Nachbereich: eine NET-Stelle (muss nicht unbedingt genau die gleiche NET-Stelle sein, die im Vorbereich ist)

Aktivierung: mindestens ein ON-Token liegt auf der Vorbereichsstelle, und mindestens ein Transition des Objektnetzes ist aktiviert

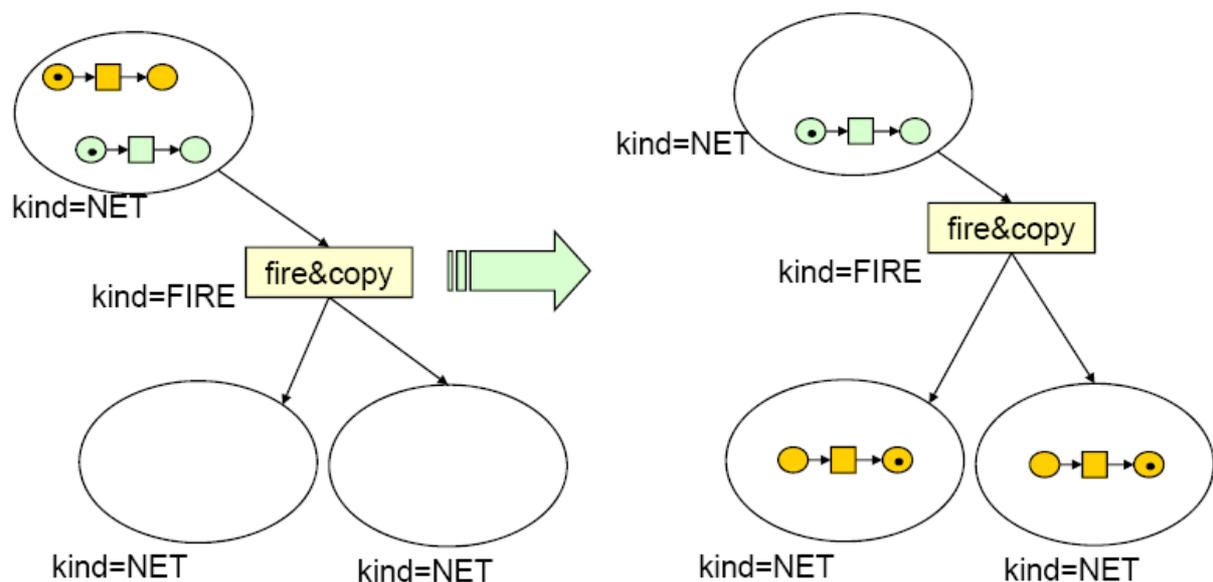
Schaltverhalten: bei mehreren ON-Token: Auswahl eines ONs durch Benutzeraktion; bei mehreren aktivierten Transitionen: Auswahl durch Benutzeraktion; Beim ausgewählten ON

wird die ausgewählte Transition geschaltet. Das resultierende ON wird auf alle NET-Stellen im Nachbereich gelegt.

Beispiel: Select Net, Fire enabled Transition



Beispiel: Fire and Copy



APPLYRURE HLTransition

wendet eine Netztransmutationsregel einmal auf ein Objektnetz an

Vorbereich: mindestens eine NET-Stelle, genau eine RULE-Stelle

Nachbereich: beliebig viele NET-Stellen, beliebig viele RULE-Stellen

Aktivierung: mindestens ein ON-Token liegt auf jeder Vorbereichs-NET-Stelle und mindestens ein Regeltoken liegt auf der Vorbereichs-RULE-Stelle, (und es existiert ein Match von mindestens einer Regel in die Union aller ON-Token)

Schaltverhalten:

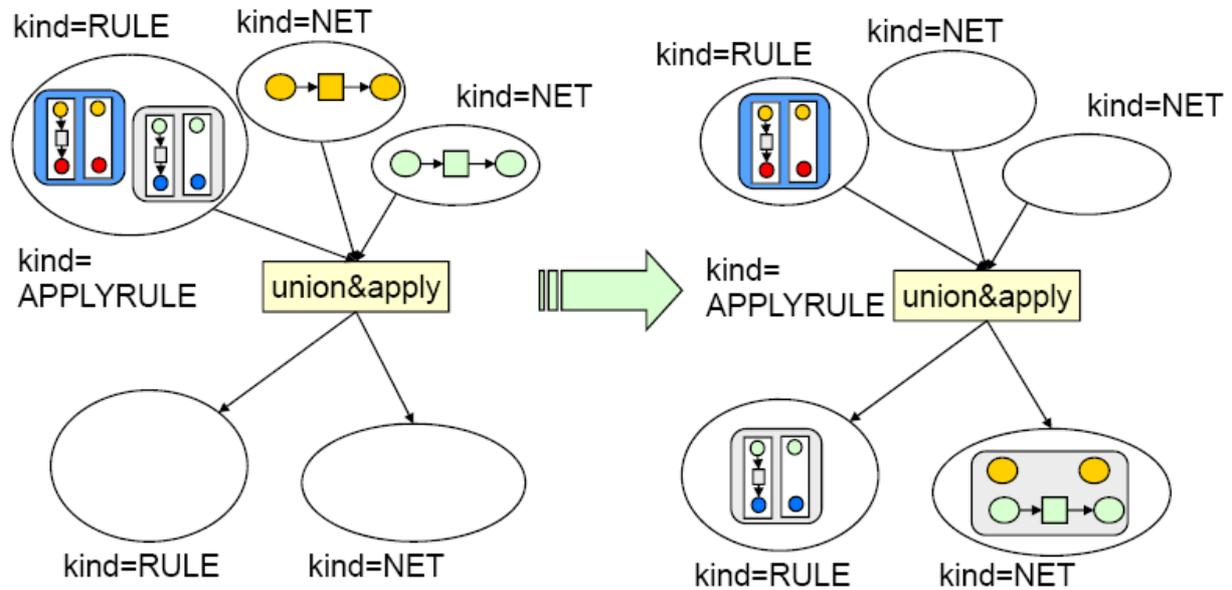
Von jeder NET-Stelle im Vorbereich wird ein ON-Token entfernt.

Von der RULE-Stelle im Vorbereich wird ein Regel-Token entfernt.

Die Regel wird auf die Union aller entfernten ON-Token angewendet.

Auf jede Nachbereichs-NET-Stelle wird das Resultat der Regelanwendung gelegt. Auf jede Nachbereichs-RULE-Stelle wird die angewendete Regel gelegt.

Beispiel: Apply one Rule to the union of two Nets



SPLIT HLTransition

Spaltet ein Objektnetz, das aus n Zusammenhangskomponenten besteht, auf in n entsprechende zusammenhängende Objektnetze

Vorbereich: genau eine NET-Stelle, keine RULE-Stelle

Nachbereich: genau eine NET-Stelle, keine RULE-Stellen

Aktivierung: mindestens ein ON-Token liegt auf der Vorbereichs-NET-Stelle

Schaltverhalten: Ein ON-Token der Vorbereichsstelle wird ausgewählt. Besteht das ON aus mehreren Zusammenhangskomponenten, so wird auf der Nachbereichsstelle für jede einzelne Zusammenhangskomponente ein ON-Token erzeugt, das der Komponente entspricht.

Beispiel: Split a Net into three components

