

Methodenbeschreibung BMW Magic Grid

Teil 6: White-Box Behaviour

Inhaltsverzeichnis	Seite
1 Einleitung	3
1.1 Hinweise und Best Practices	3
1.2 Einordnung in die Methodik „Angepasstes MagicGrid“	3
2 Methode: White-Box Behaviour.....	4
2.1 Zielsetzung und Zweck	4
2.2 Modellierungselemente	4
3 Toolspezifische Umsetzung	7
3.1 Activity Diagram für Aktivitäten erstellen	7
3.2 Modellierung von Start und Ende	8
3.3 Modellierung der Actions	8
3.4 Modellierung des Kontrollflusses	9
3.5 Swimlanes erstellen.....	10

1 Einleitung

1.1 Hinweise und Best Practices



Warnung:

Wichtige Hinweise zur Handhabung von Magic Grid und Cameo



Tool-Tip:

Nützliche Tipps zur Bedienung des Tools



Anmerkungen:

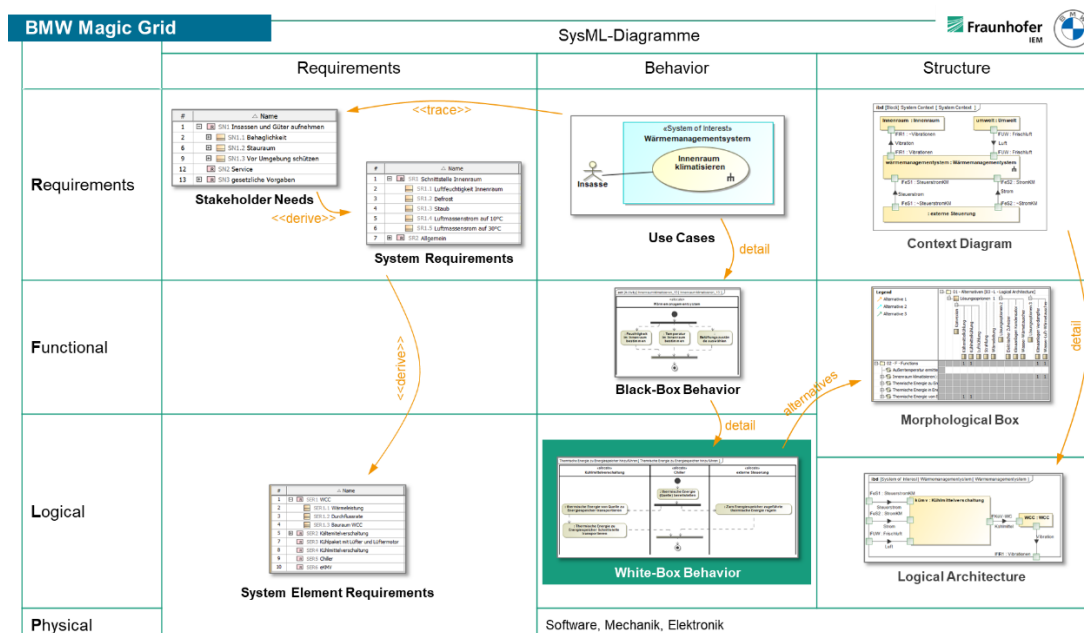
Hinweise zum Umgang mit Bestandteilen der Methodik



Referenzen:

Hinweise zu ergänzender und weiterführender Literatur

1.2 Einordnung in die Methodik „BMW MagicGrid“



2 Methode: White-Box Behaviour

2.1 Zielsetzung und Zweck

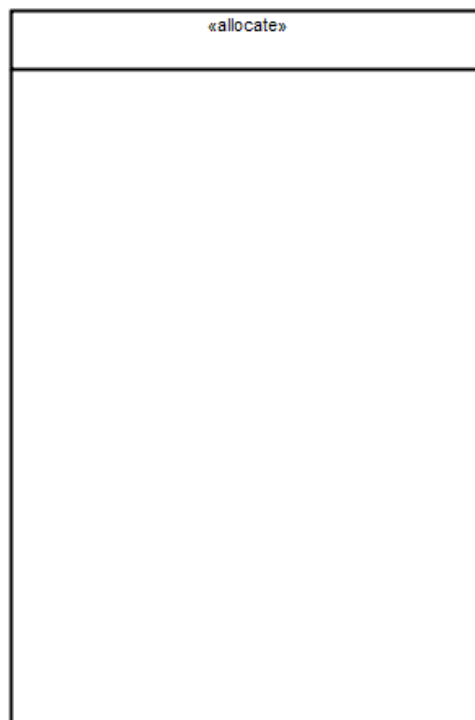
Das Ziel der White-Box-Verhaltensbeschreibung ist es, *Aktivitäten* der Black-Box Verhaltensbeschreibungen zu detaillieren. Der Zweck der Methode ist es ein Verständnis für die Abläufe der *Interaktionen* zwischen den Systemelementen des System of Interest zu bekommen.

2.2 Modellierungselemente

Aktivitätsdiagramm

Swimlanes

Eine Swimlane beschreibt einen Verantwortungsbereich. Alle Aktionen (*Actions*), welche innerhalb einer Swimlane platziert werden, müssen von einer Entität umgesetzt werden. Die Swimlane repräsentiert in der White-Box-Modellierung die Systemelemente.



Start- und Endpunkte

Der *Initial Node* definiert den Start eines Aktivitätsdiagramm.



Der *Activity Final Node* definiert das Ende eines Aktivitätsdiagramm. Es terminiert die gesamte *Aktivität* und somit alle *Flows*.



Der *Flow Final* beendet lediglich den aktuellen *Flow*. Weitere (parallel laufende) *Flows* bleiben erhalten.



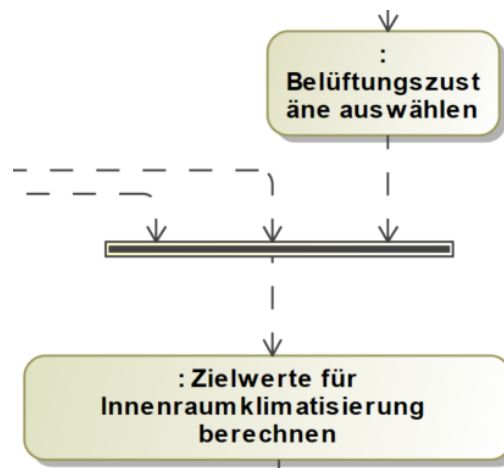
Aktion

Eine *Call Behavior Action* verweist auf ein weiteres Verhalten, das durch ein anderes Aktivitätsdiagramm beschrieben werden kann. Sie kann als Funktion verstanden werden.

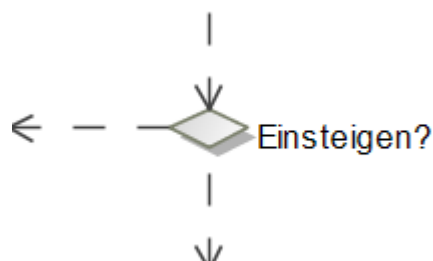


Kontrollfluss

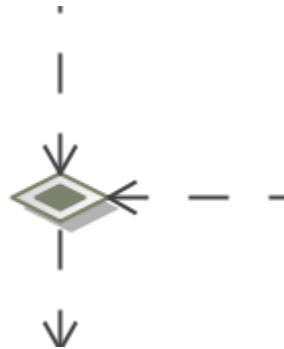
Mithilfe der *ControlFlow*-Pfeile wird die logische Reihenfolge von *Actions* definiert. Es darf nur ein *ControlFlow* von einer *Action* ausgehen.



Decision Nodes erlauben, dass alternative Ausführungspfade durchlaufen werden (vgl. if/else, etc.). Die Bedingung für den jeweiligen Pfad wird an dem jeweils ausgehenden *ControlFlow* angegeben.



Merge Nodes führen mehrere *ControlFlows* wieder zusammen (Logische OR-Gate Funktionalität: es geht weiter, sobald mindestens ein Token anliegt).



Durch einen *Fork Node* können parallele Abläufe gestartet werden.



Join Nodes führen mehrere *ControlFlows* wieder zusammen. (Logische AND-Gate Funktionalität: es geht weiter, sobald alle Tokens anliegen).



Optional: Objektfluss

Mithilfe der *ObjectFlow*-Pfeile wird der *Objektfluss* zwischen *Actions* definiert.

Input Pin bzw. Output Pin: Stelle, an der eine *Action* ein Objekt konsumiert bzw. zur Verfügung stellt.

Fluss: *Fluss* der übertragen wird.

Activity Parameter: Daten, die eine Aktivität als Input entgegennimmt oder als Output produziert. Die Richtung des *Activity Parameters* wird als *Direction* angegeben.

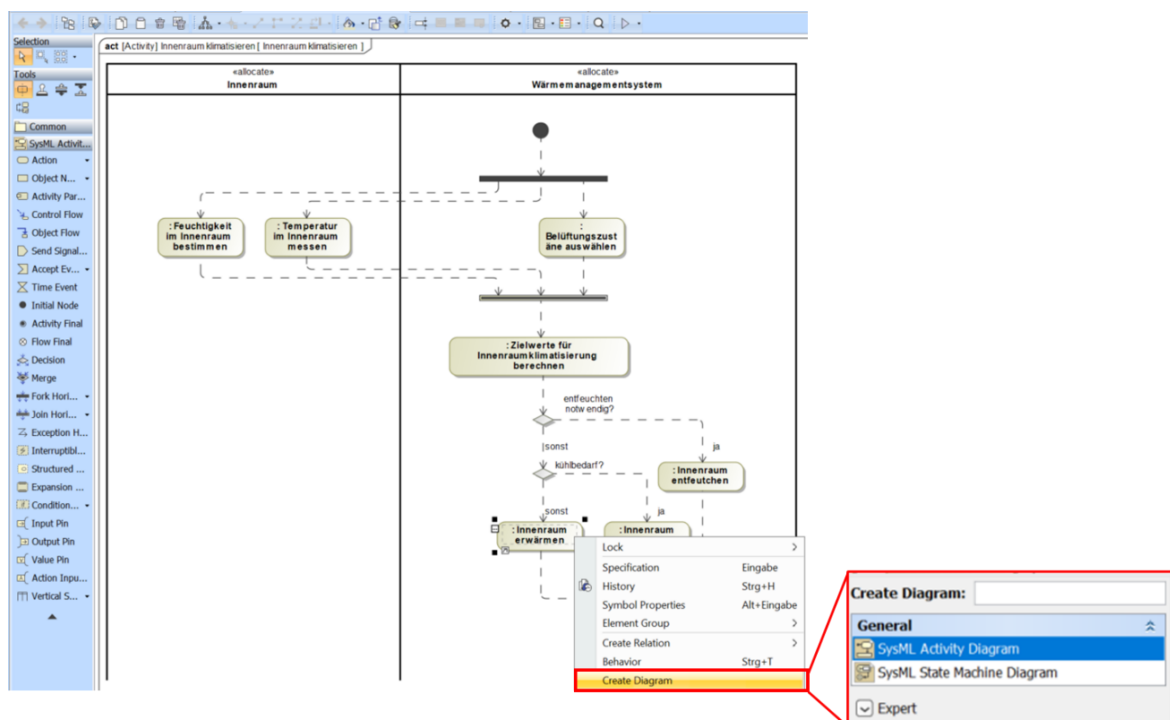
3 Toolspezifische Umsetzung

Die wesentlichen Schritte für die Verfeinerung einer Aktivität durch ein Aktivitätsdiagramm sind:


1. Erstellung eines Activity Diagram (je relevanter *Activity*)
2. Modellierung von Start (*Initial Node*) und Ende (*Activity Final*)
3. Modellierung der *Actions*
4. Modellierung des Kontrollfluss (*Control-Flow*)
5. Modellierung der Swimlanes

3.1 Activity Diagram für Aktivitäten erstellen

Um für eine *Aktivität* ein Aktivitätsdiagramm als dessen Verhaltensmodellierung zu erstellen, wird im Diagramm der *Aktivität* der Eintrag „Create Diagram“ > „Activity Diagram“ ausgewählt.

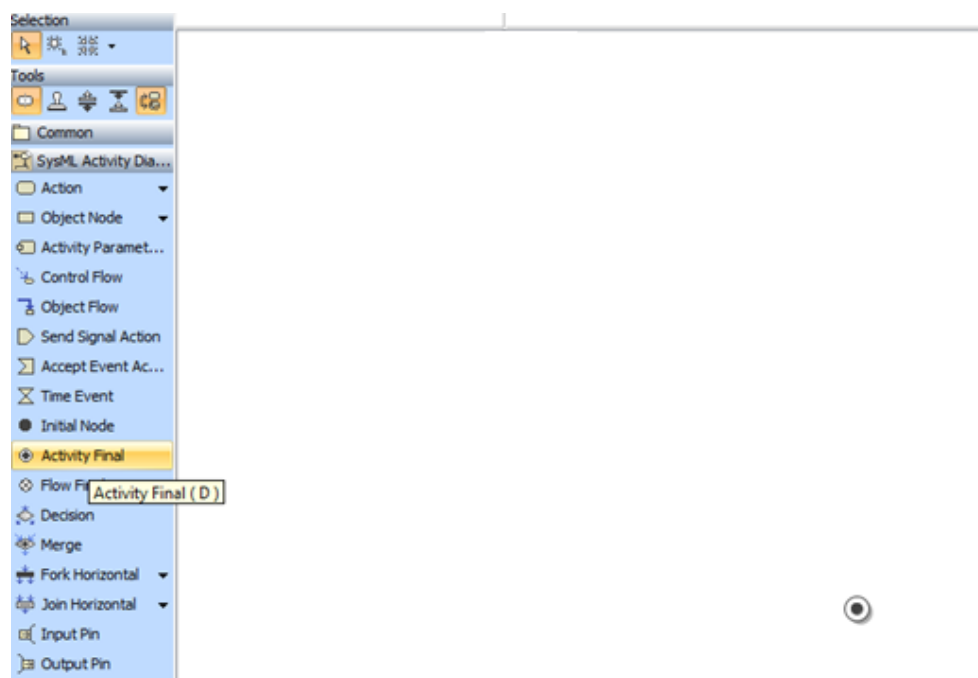


Das zugehörige Aktivitätendiagramm wird automatisch geöffnet.

Im Aktivitätendiagramm erhält die ausgewählte *Aktivität* folgendes Symbol . Dieses Symbol zeigt an, dass es zu dieser *Aktivität* ein verschachteltes Diagramm gibt. Durch Doppelklick auf die *Aktivität* kann diese geöffnet werden.

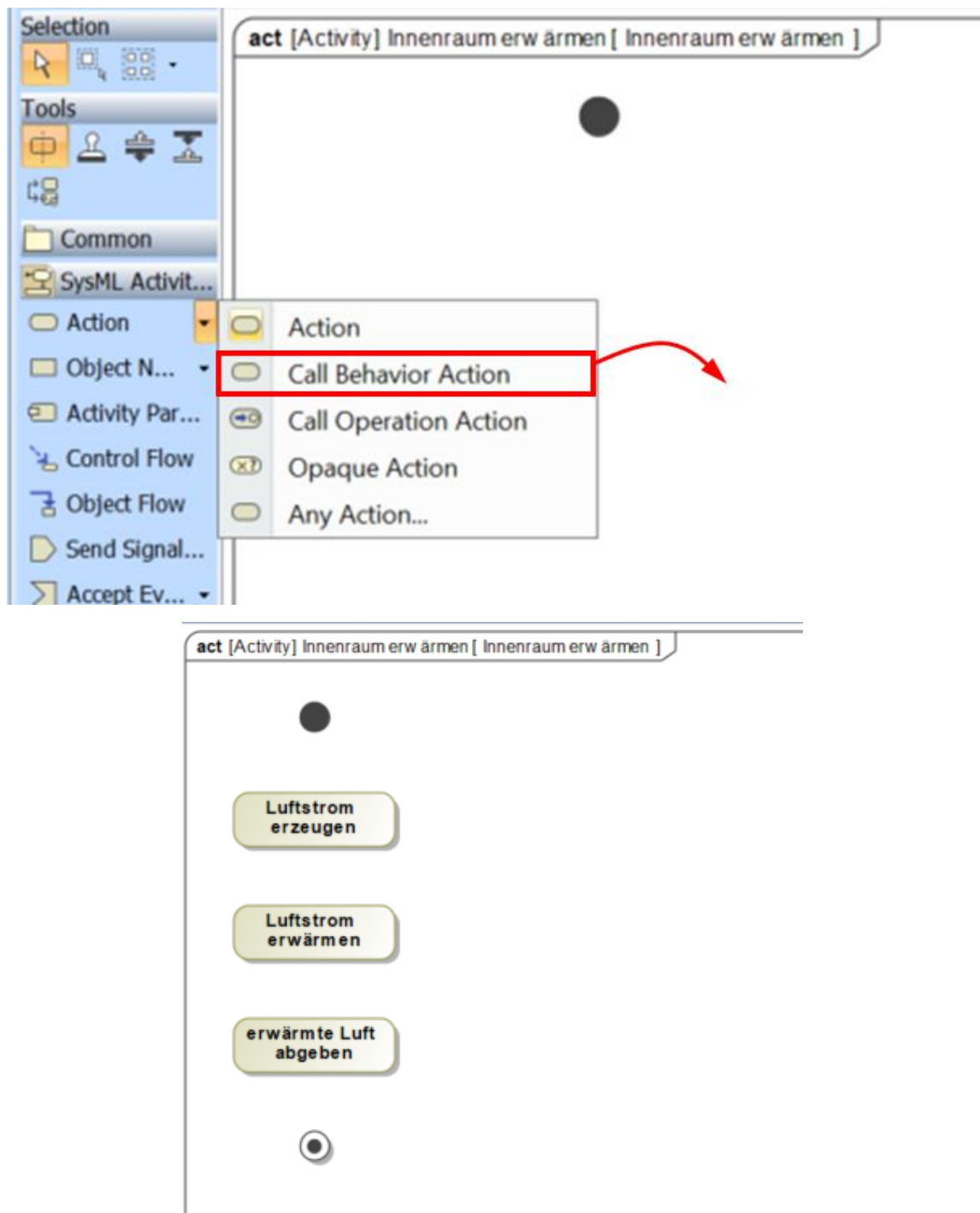
3.2 Modellierung von Start und Ende

Begonnen werden sollte immer mit einem *Initial Node*, der angibt, wo der Ablauf beginnt. Das Ende wird mittels *Activity Final* modelliert. Beide Elemente befinden sich in der Toolbar.



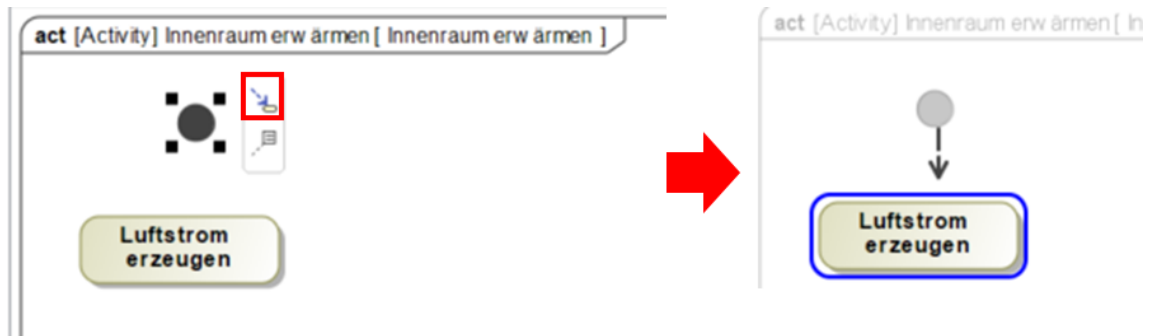
3.3 Modellierung der Actions

Zur Modellierung der *Actions* wird die *Call Behavior Action* in der Toolbar via Klick ausgewählt und anschließend in das Diagramm geklickt, um diese zu erstellen.

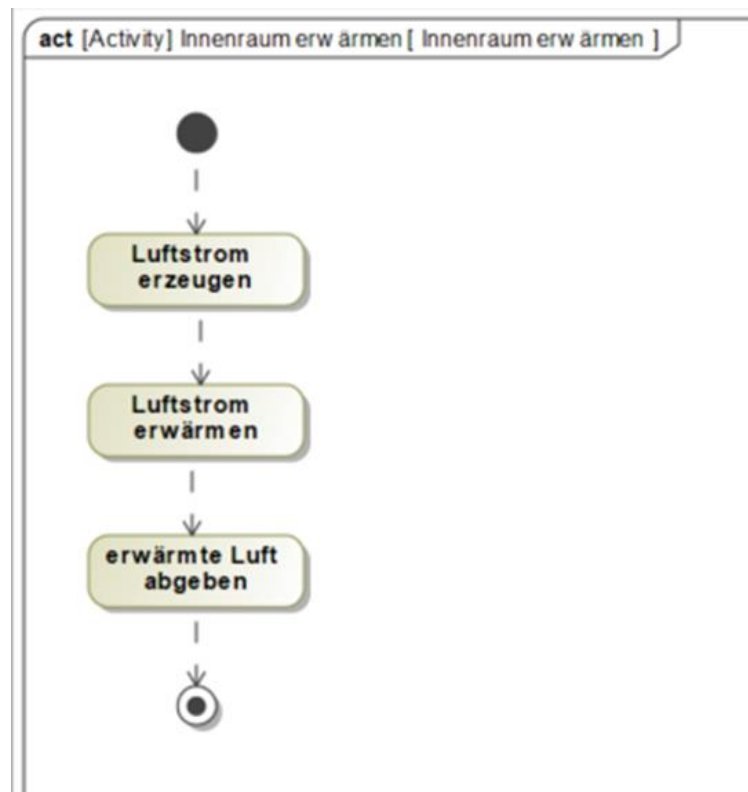


3.4 Modellierung des Kontrollflusses

Ausgehend vom *Initial Node* werden die *Actions* durch einen *Control Flow* verbunden.



Dies wird für alle *Aktivitäten* wiederholt.



Bei der Erstellung können auch *Decision Nodes* und *Fork Nodes* genutzt werden. Weitere Hinweise dazu sind in Methode 05 zu finden.

3.5 Swimlanes erstellen



Anmerkungen:

Dieser Schritt der Modellierung ist erst nach der Definition der logischen Architektur (Methode 08) möglich.

In dem Diagramm werden die an der *Aktivität* beteiligten Systemelemente des System of Interests integriert. Dafür wird eine Swimlane in der Toolbar ausgewählt und via Drag&Drop in das Diagramm gezogen.

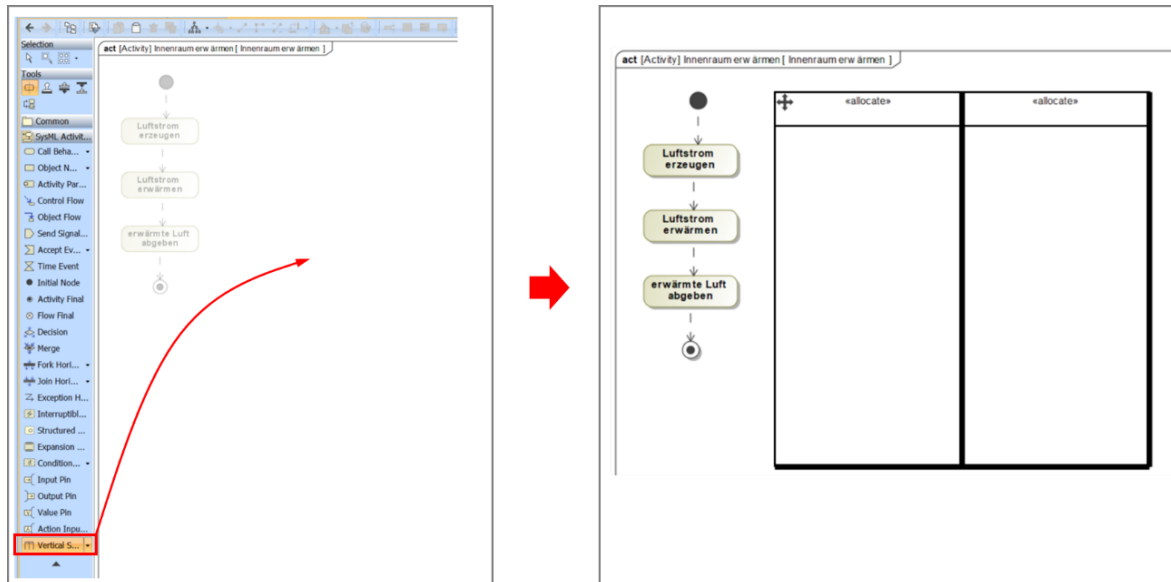
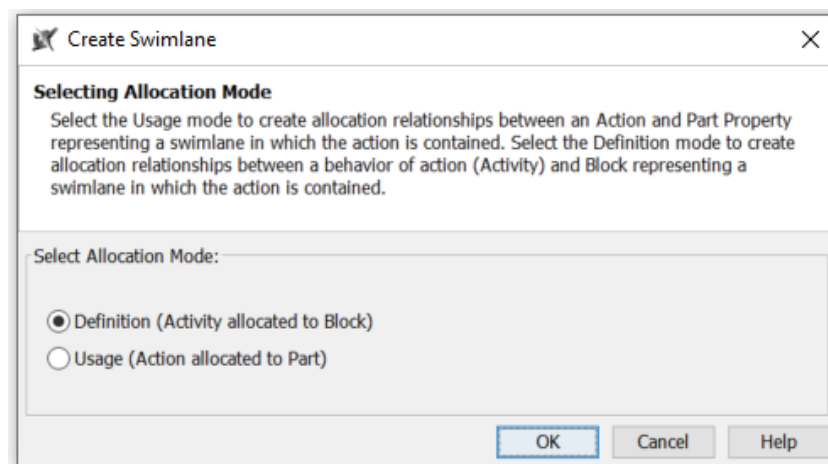
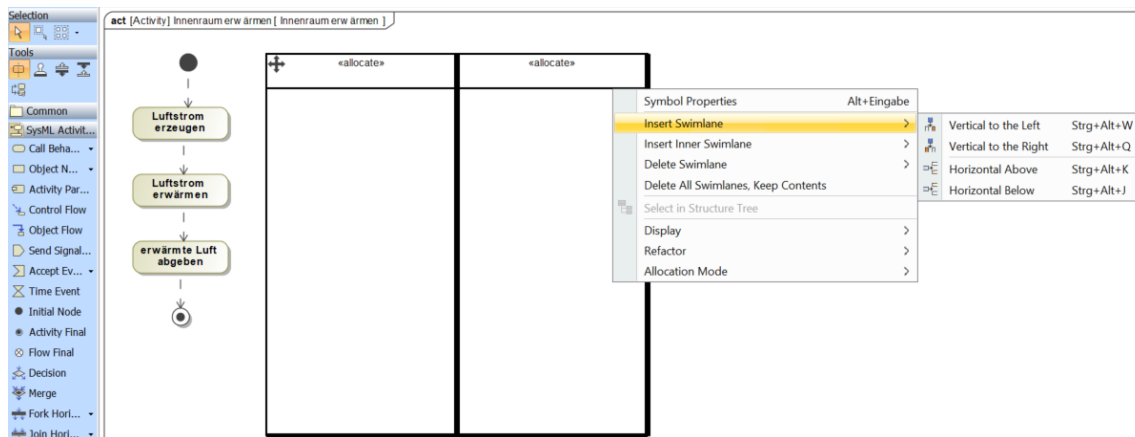


Abbildung 1: Erstellen einer Swimlane

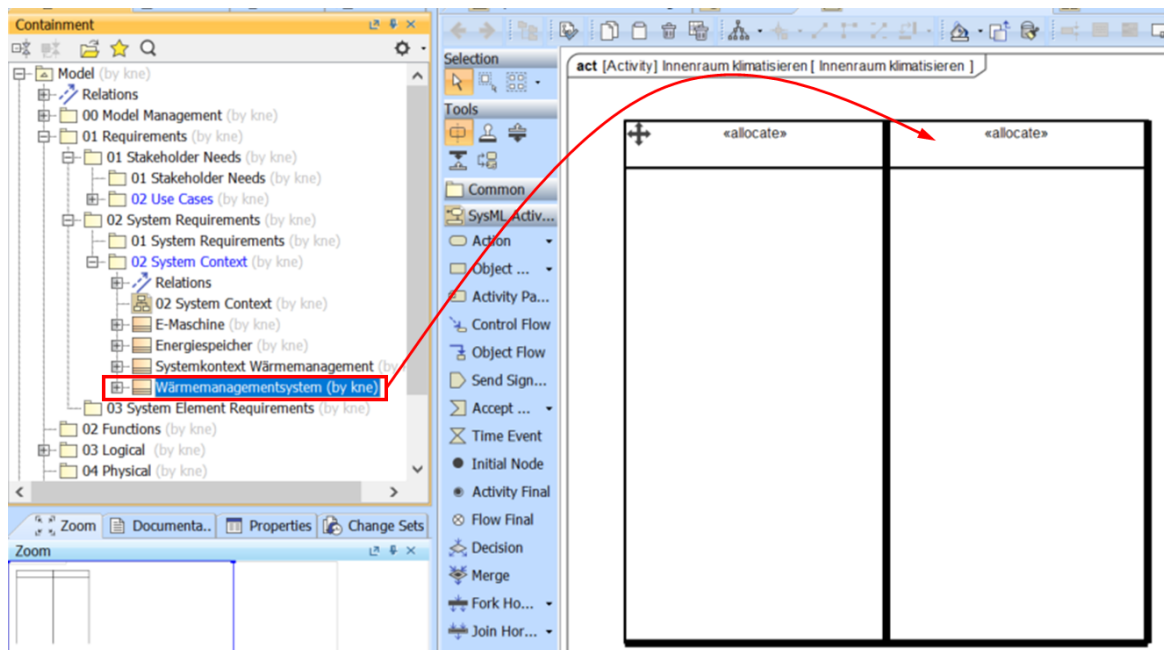
Bei der ersten Erstellung einer Swimlane öffnet sich ein Dialogfenster in dem die Option „Definition“ mit „OK“ bestätigt wird.



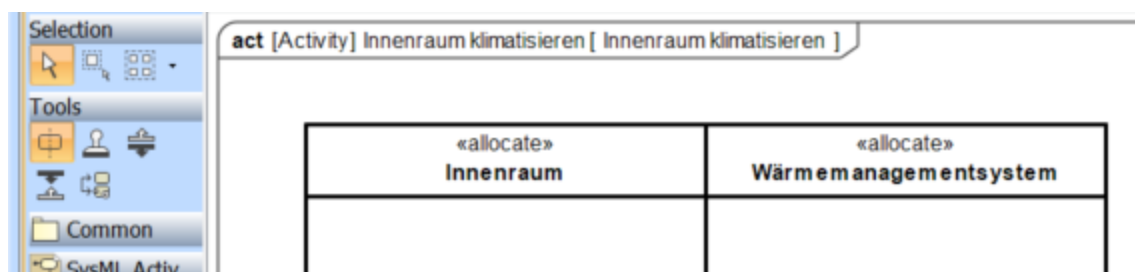
Weitere Siwimlanes können durch rechtsklick auf eine bestehende Swimlane über die Option „Insert Swimlane“ hinzugefügt werden.



Nun müssen die <<System Elements>> des System of Interest den Swimlanes zugewiesen werden. Dazu wird auf die im Rahmen der logischen Architektur (Methode 8) definierten Elemente zurückgegriffen.

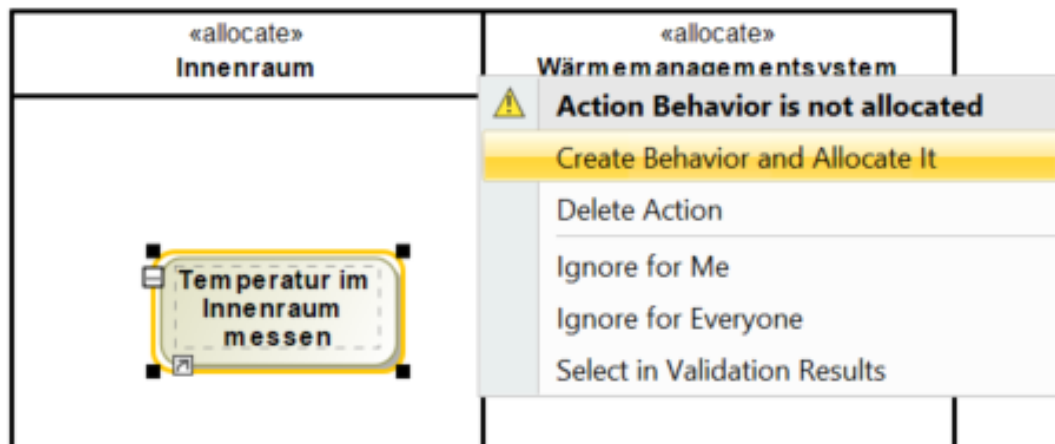


Dies wird für alle Systemelemente, die an dieser *Aktivität* beteiligt sind, wiederholt.



Wird eine *Call Behavior Action* auf eine Swimlane platziert, wird das dazugehörige Verhalten nicht automatisch der Swimlane zugeordnet. Dies wird durch eine gelbe Umrandung der *Action* gekennzeichnet. Um dies zu beheben, muss die *Action* angeklickt und in dem Seitenmenü auf das gelbe Warndreieck ausgewählt werden. Mit der Option

„Create Behavior and Allocate it“ wird das entsprechende Verhalten erzeugt und an den *Block* allokiert, die Warnung verschwindet.



So werden alle weiteren *Aktivitäten* zugeordnet.