

## **Methodenbeschreibung BMW Magic Grid**

Teil 1: System Context



Inhaltsverzeichnis	Seite
1 Einleitung .....	3
1.1 Hinweise und Best Practices .....	3
1.2 Einordnung in die Methodik „BMW MagicGrid“.....	3
2 Methode: System Context.....	4
2.1 Zielsetzung und Zweck.....	4
2.2 Modellierungselemente .....	4
3 Toolspezifische Umsetzung .....	7
3.1 Definition der Modellierungselemente mittels Block Definition Diagram .	7
3.1.1 Definition der Elemente des Systemkontexts .....	7
3.1.2 Definition von Flüssen .....	10
3.1.3 Definition von Schnittstellen.....	13
3.2 Erstellung eines Context Diagrams für den Systemkontext.....	14
3.2.1 Modellierung des Kontextdiagramms .....	14
3.2.2 Modellierung der Interaktion zwischen System of Interest und den Umfeldelementen .....	16
3.2.3 Modellierung von Flüssen.....	17
3.2.4 Modellierung von Schnittstellen .....	19

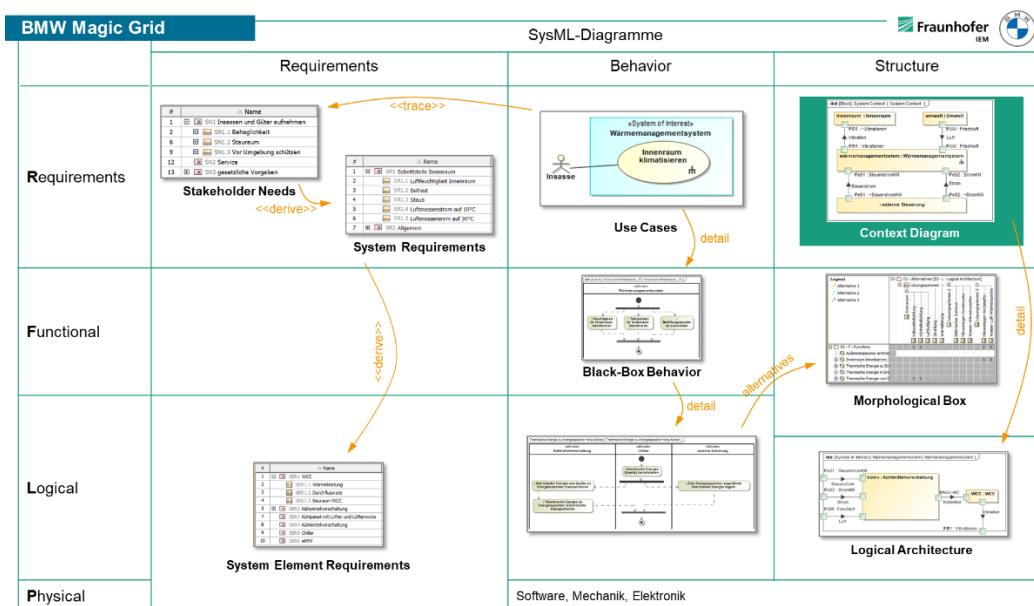


# 1 Einleitung

## 1.1 Hinweise und Best Practices

 <p><b>Warnung:</b></p> <p>Wichtige Hinweise zur Handhabung von Magic Grid und Cameo</p>
 <p><b>Tool-Tip:</b></p> <p>Nützliche Tipps zur Bedienung des Tools</p>
 <p><b>Anmerkungen:</b></p> <p>Hinweise zum Umgang mit Bestandteilen der Methodik</p>
 <p><b>Referenzen:</b></p> <p>Hinweise zu ergänzender und weiterführender Literatur</p>

## 1.2 Einordnung in die Methodik „BMW MagicGrid“



## 2 Methode: System Context

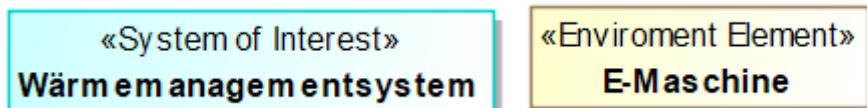
### 2.1 Zielsetzung und Zweck

Das Ziel des System Context ist die Modellierung von relevanten Zusammenhängen zwischen dem System of Interest und seiner Umgebung. Der Zweck ist die Identifikation und Definition von externen Schnittstellen und die Festlegung der Systemgrenze.

### 2.2 Modellierungselemente

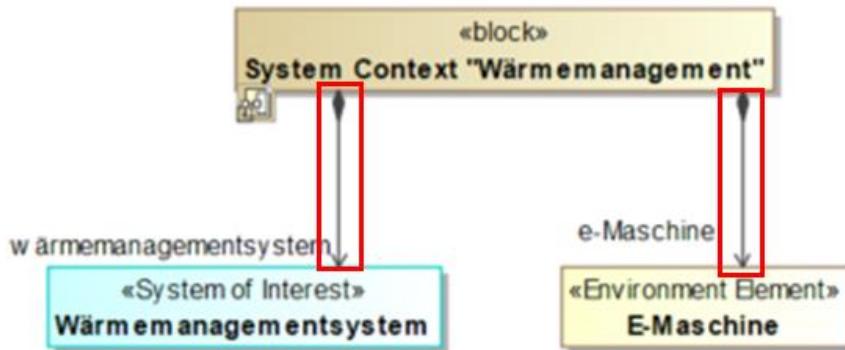
#### Block

Ein *Block* repräsentiert eine abstrakte Einheit, die ein System oder Umgebungselement sein kann. So wird das System of Interest als *Block* mit dem Stereotypen <<System of Interest>> definiert, sowie die Umgebungselemente wie Menschen oder externe Systeme mittels *Blocks* mit dem Stereotypen <<Environment Element>> definiert.



#### Composition

Eine *Composition* beschreibt eine hierarchische Beziehung zwischen zwei *Blöcken*. Dabei wird definiert, dass ein *Block* ein Teil eines anderen *Blocks* ist. Das Wärmemanagementsystem (WMS) und die E-Maschine sind z.B. Teil des Fahrzeugs (bzw. des Systemkontexts des WMS).



#### Port

Ein *Port* ist eine Schnittstelle eines *Blocks*, über die der *Block* mit anderen Systemelementen (Komponenten) oder Umweltelementen interagieren kann. *Ports* repräsentieren den Austausch von Informationen, Materialien oder Energie.



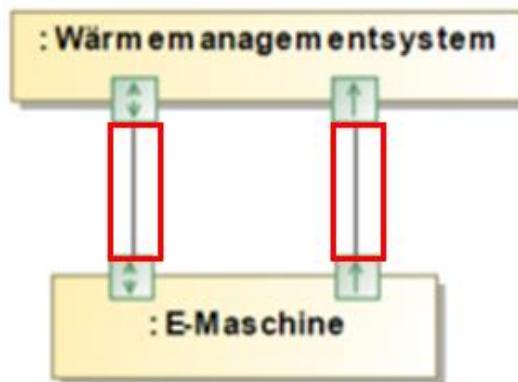
### Interface Block

Ein *Interface Block* ist ein spezieller *Block*, der dazu dient, eine Schnittstelle zu definieren, die einem *Port* zugeordnet wird. Damit wird der *Block* typisiert und bekommt die Eigenschaften des *Interface Blocks*. Dies ermöglicht die Spezifikation von erlaubten *Flüssen* und Flussrichtungen.



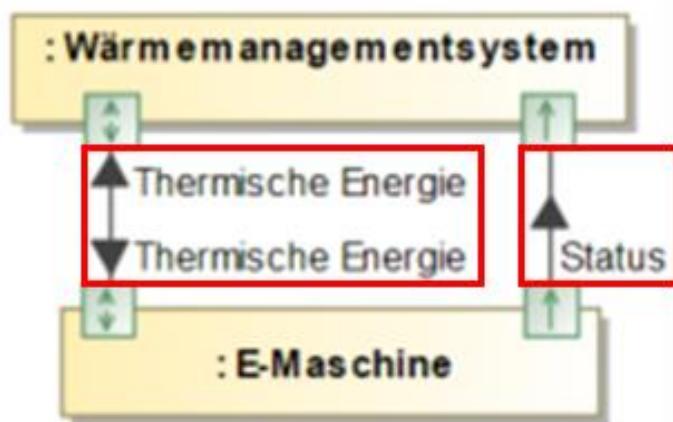
### Interaktion/Connector

Die *Interaktion* zwischen System of Interest und Umfeldelementen wird mittels *Connector* modelliert. Er repräsentiert die Verbindung zwischen zwei *Blöcken*.



### Flüsse und Item Flows

*Flüsse* werden mit Hilfe von Stereotypen in <<Informations-, <<Energie-, <<Materialfluss, sowie <<logischem und <<störendem Fluss>> unterschieden und definiert. Wird ein *Fluss* auf einer *Interaktion* (*Connector*) gezogen, wird dies als *Item Flow* mit einer definierten Richtung dargestellt. Der *Item Flow* repräsentiert also den *Fluss* der über eine *Interaktion* (*Connector*) fließt bzw. ausgetauscht wird. Zwischen dem Wärmemanagementsystem und der E-Maschine fließt beispielsweise Thermische Energie und der Status.



### 3 Toolspezifische Umsetzung

Die wesentlichen Schritte zur Modellierung des Systemkontexts sind:

1. Definition der Modellierungselemente mittels Block Definition Diagram
2. Erstellung eines Context Diagramms mittels Internal Block Diagram
3. Modellierung des System of Interest und der Umgebungselemente
4. Modellierung der Interaktionen zwischen System of Interest und den Umfeldelementen mittels *Connector*
5. Modellierung der Flüsse mittels *Item Flow*
6. Spezifikation der Schnittstellen mittels *Interface Block*

#### 3.1 Definition der Modellierungselemente mittels Block Definition Diagram

Die SysML folgt der Logik: **Zuerst definieren, dann modellieren.** Daher müssen zunächst die einzelnen Modellelemente definiert werden, die im Rahmen des Kontexts benötigt werden. Diese sind die Elemente des Systemkontexts, die *Flüsse*, sowie die Schnittstellen. Dafür werden Block Definition Diagramme verwendet.

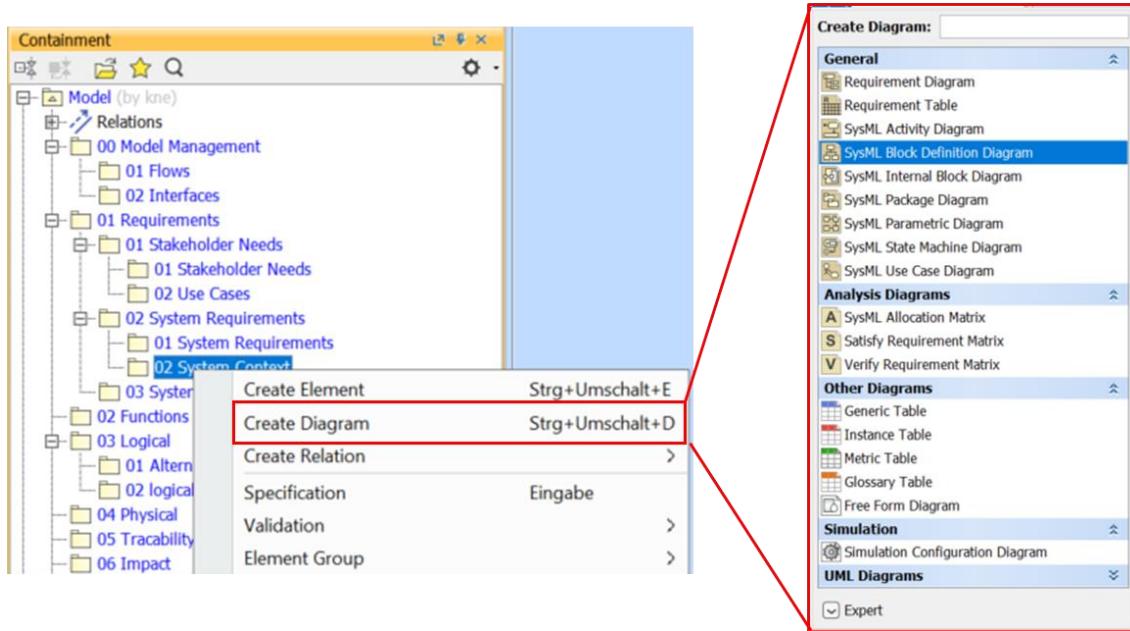


#### Anmerkungen:

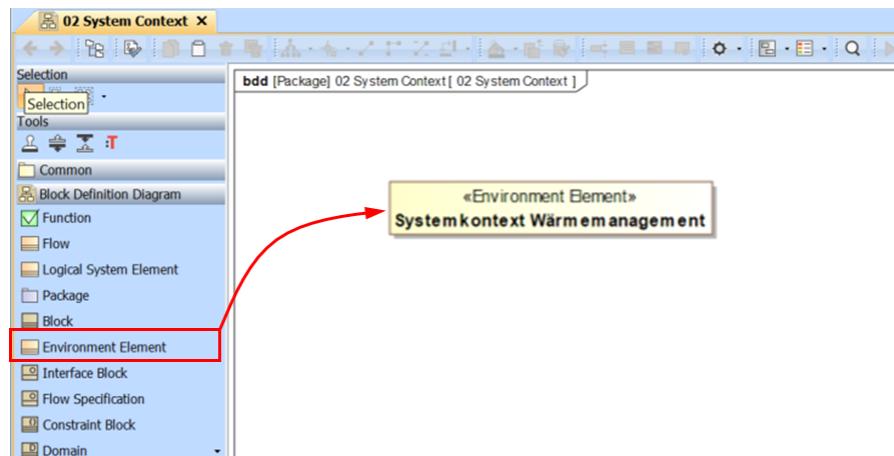
In der Praxis erfolgt die Definition der jeweiligen Modellierungselemente iterativ. Das bedeutet, dass im Rahmen der Modellierungstätigkeiten des Systemkontext (Schritt 3.2) neue Flüsse oder Schnittstellen identifiziert werden, die dann entsprechend definiert werden müssen (Schritt 3.1).

##### 3.1.1 Definition der Elemente des Systemkontexts

Zur Definition der Elemente des Systemkontexts wird ein Block Definition Diagramm (BDD) verwendet. Dazu wird dieses in dem Package „02 System Context“ erstellt.



Das erstellte Block Definition Diagramm öffnet sich automatisch. In diesem Diagramm werden nun die *Blöcke* erstellt und deren Beziehung zueinander definiert. Dafür wird zuerst auf den *Block* mit dem Stereotypen <<Environment Element>> geklickt und anschließend in die Fläche des Diagramms. Somit entsteht das erste Umgebungselement, das jetzt noch benannt werden muss.

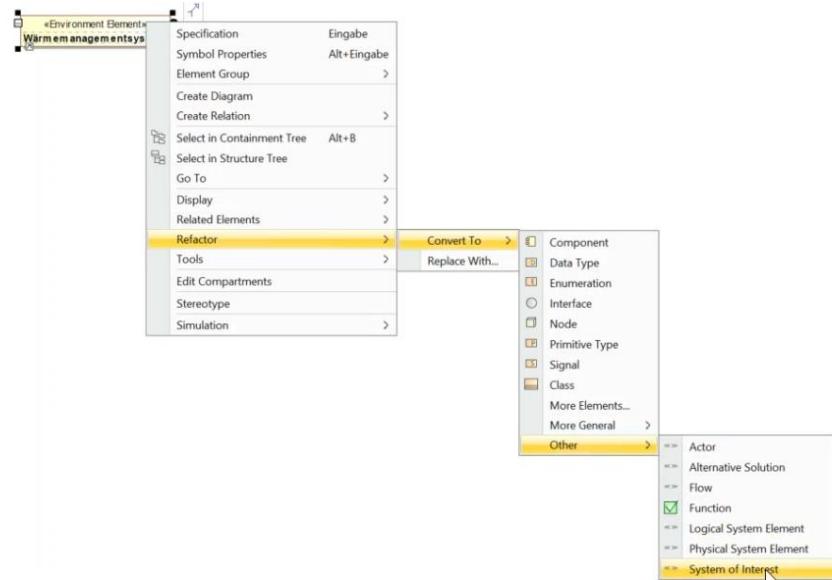


## Anmerkungen:

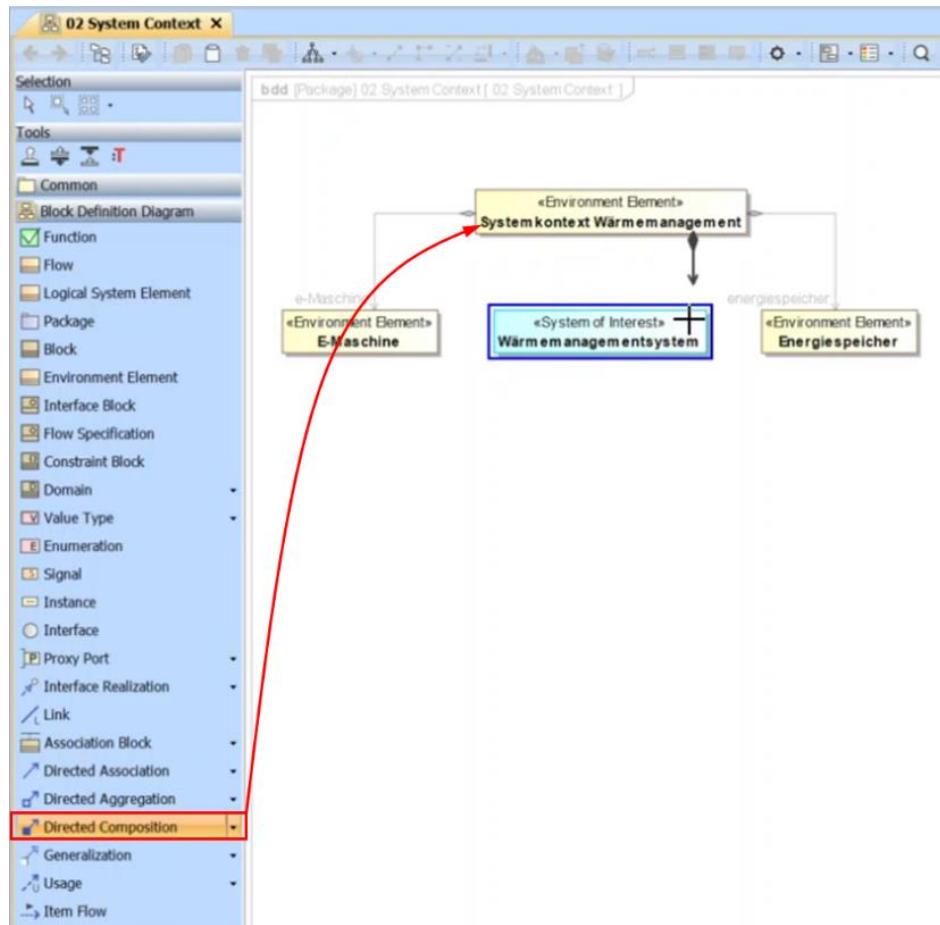


Neben dem System of Interest und den Umfeldelementen muss auch ein Block „Systemkontext“ definiert werden. Dies hat den Hintergrund, dass in der SysML die Interaktionen von Blöcken mit Hilfe von Internen Block Diagrammen (IBD) modelliert werden. Der Block „Systemkontext“ dient in der weiteren Modellierung als Ausgangspunkt für die Modellierung der Zusammenhänge.

Für alle weiteren Elemente kann auf gleiche Weise fortgefahren werden. Für das System of Interest ist noch ein weiterer Schritt notwendig, um es von den Umfeldelementen abzuheben. Dafür wird mit der rechten Maustaste auf das System of Interest geklickt und die Option „Refactor“ ausgewählt, wie im Bild beschrieben.

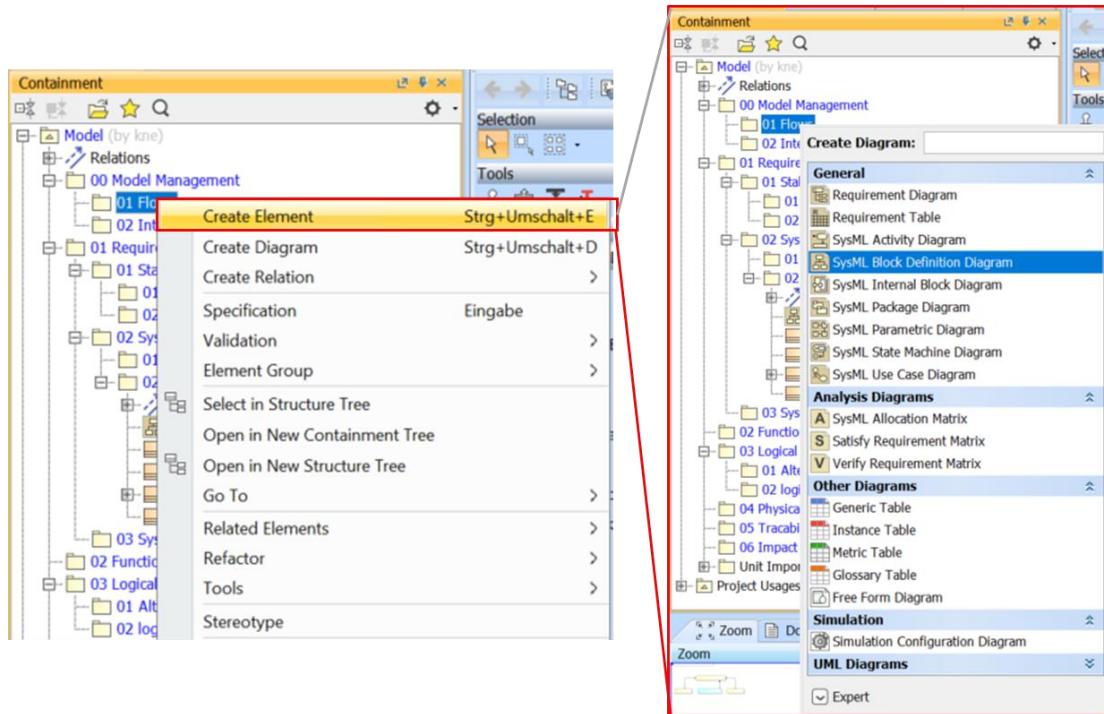


Das Ergebnis sind alle Elemente des Kontexts. Diese müssen nun noch mithilfe von einer *Composition* zueinander in Beziehung gesetzt werden. Dafür wird zunächst in der Toolbar auf „Directed Composition“ geklickt. Anschließend auf das Elternelement, in diesem Fall der Block „Systemkontext“ und anschließend auf das Kinderelement, z.B. das <<System of Interest>>.



### 3.1.2 Definition von Flüssen

Analog zu der Definition der Elemente des Kontextes müssen auch die *Flüsse*, die zwischen den Elementen ausgetauscht werden sollen, definiert werden. Dazu wird ein Block Definition Diagram in dem Package „01 Flows“ erstellt.



Das erstellte Block Definition Diagramm öffnet sich automatisch. In diesem Diagramm werden nun die *Flüsse* definiert. Dafür wird zuerst auf den *Block* mit dem Stereotypen <<Flow>> geklickt und anschließend in die Fläche des Diagramms. Der entstandene *Fluss* muss jetzt noch benannt werden.

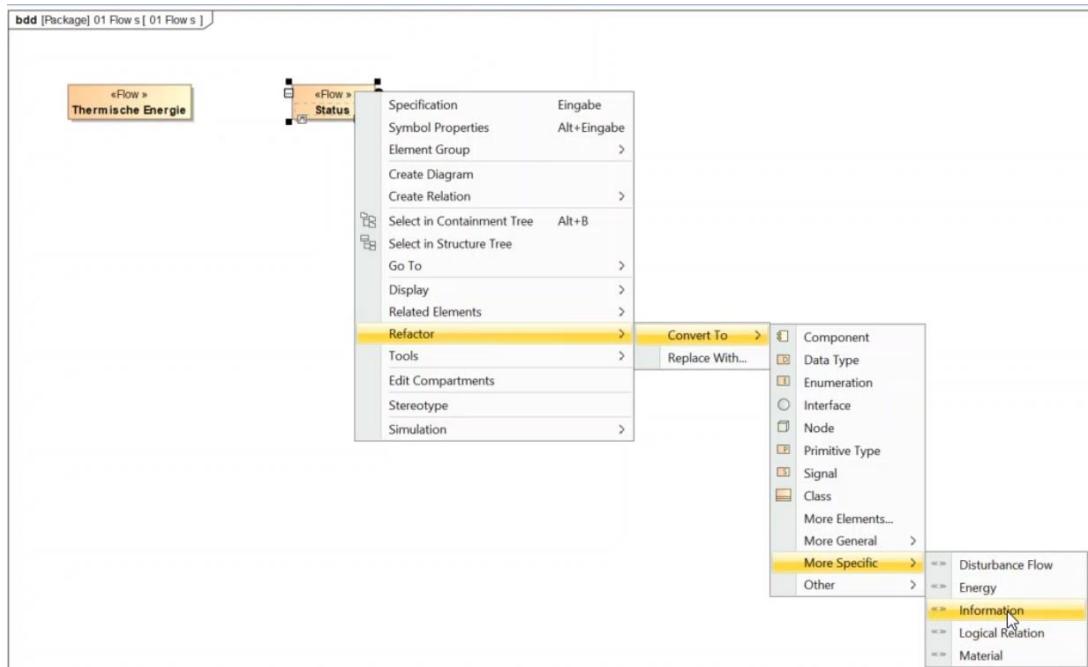


### Tool-Tip:

Die Flüsse und Interfaces sind in dem Package „Model Management“ abgelegt, da diese hier zentral für das gesamte Modell definiert werden. In jeder weiteren Sicht können diese (Requirements, Function, Logical) wiederverwendet werden.

Die Entkopplung ermöglicht später auch den Aufbau einer übergreifenden Modellbibliothek.

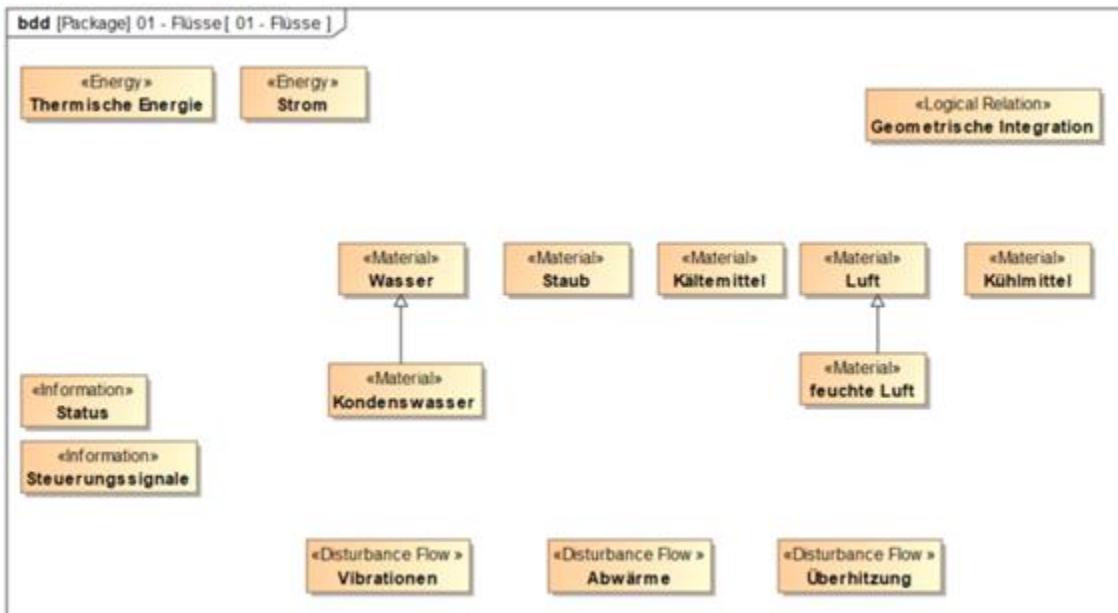
Die *Flüsse* können nun noch typisiert werden, was später eine Unterscheidung ermöglicht. Dafür Rechtsklick auf den *Fluss* und über die Option „Refactor“ den *Fluss* einen anderen Stereotypen (z.B. <<Information>>) zuweisen.



### Tool-Tip:

Für eine einfachere Handhabbarkeit können die definierten Stereotypen auch in die Toolbar integriert werden. Dann kann der „Refactor“-Schritt übersprungen werden

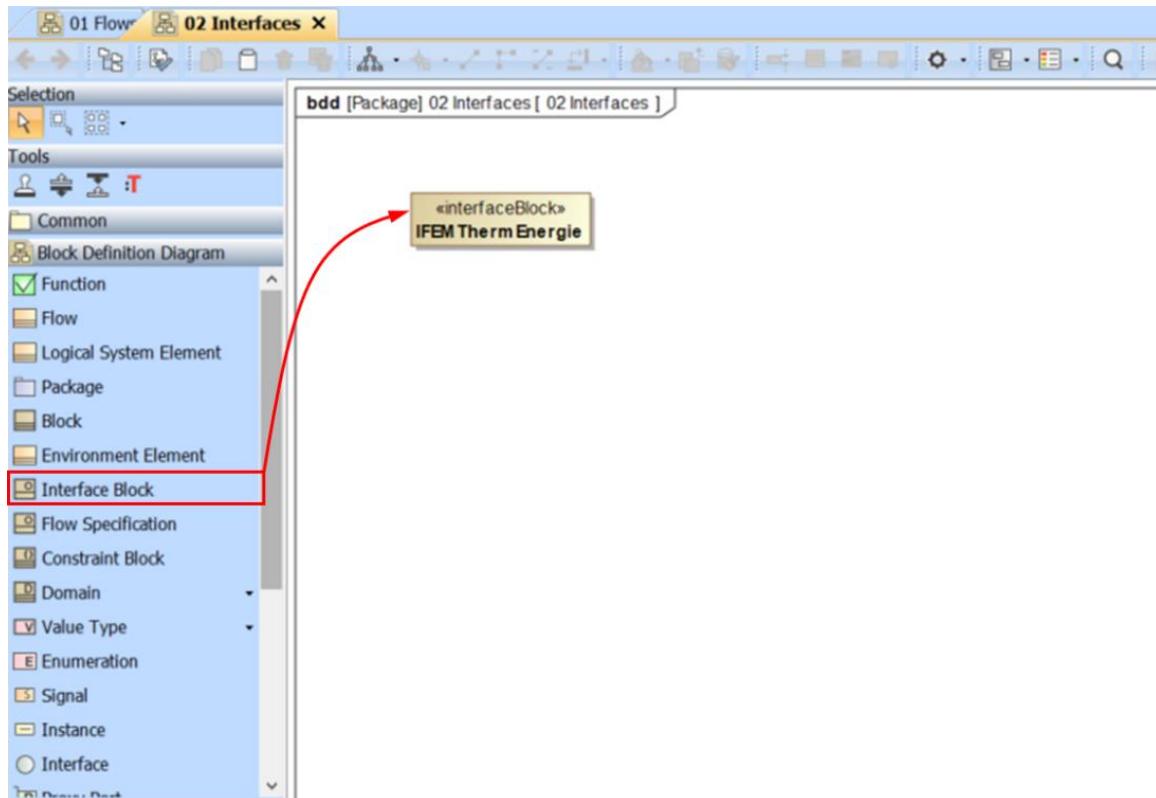
Dies wird so lange wiederholt, bis alle *Flüsse* definiert sind. Dabei können optional unterschiedliche *Flüsse* auch z.B. generalisiert werden. Damit ist die Definition der *Flüsse* abgeschlossen.



### 3.1.3 Definition von Schnittstellen

Der letzte Definitionsschritt umfasst die Definition von Schnittstellen. Dafür wird analog, wie bei der Definition der *Flüsse*, ein Block Definition Diagram erstellt; dieses Mal in dem Package „01 Interfaces“.

Das erstelle Block Definition Diagramm öffnet sich automatisch. Wie zuvor wird dieses mal der Stereotyp *Interface Block* gewählt. Der so angelegte *Block* muss noch benannt werden.

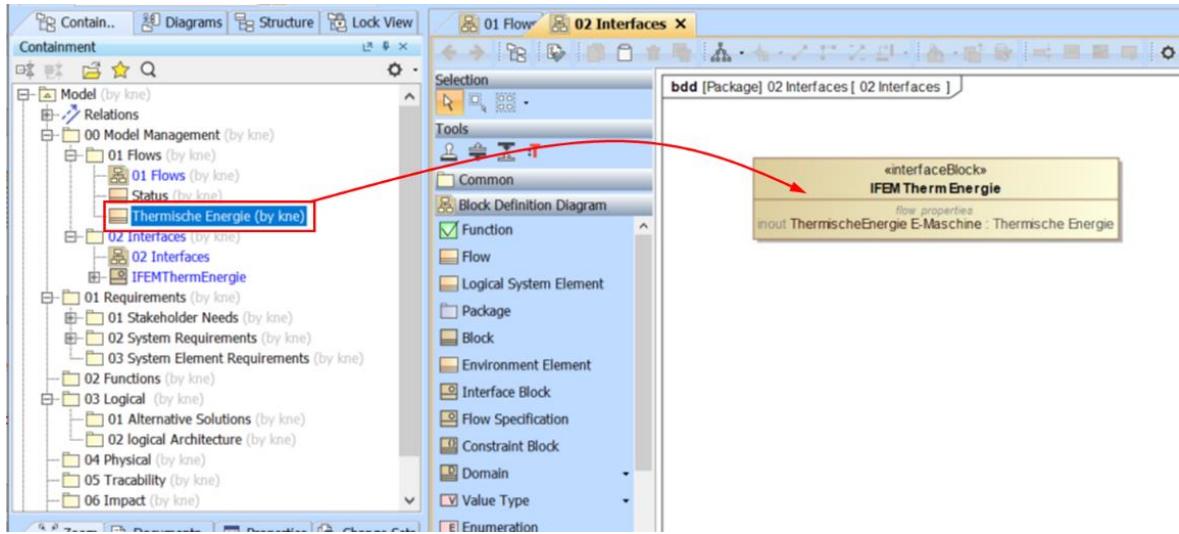


#### Anmerkungen:

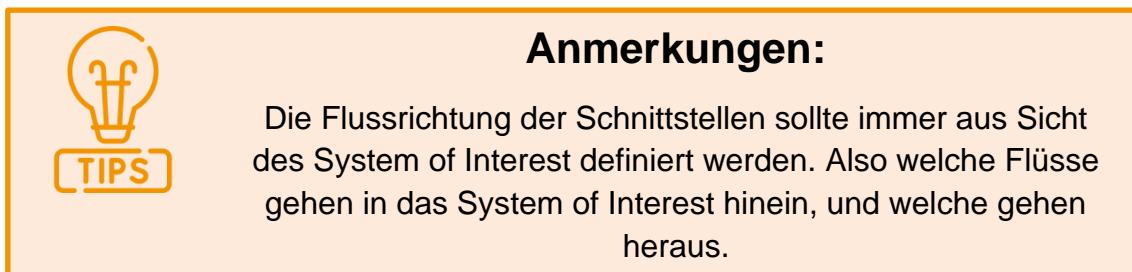


Bei der Benennung der Interface Blöcke, empfiehlt es sich, diese an die zu definierende Schnittstelle anzupassen. Bspw. IFEM (Interface E-Maschine) für die Schnittstelle zwischen WMS und E-Motor.

Anschließend müssen für die Schnittstelle noch die Flusseigenschaften (*Flow Properties*) festgelegt werden. Das bedeutet, es wird definiert, welche *Flüsse* über die Schnittstelle gehen und welche Richtung diese haben. Dazu muss im Containment-Tree einer der zuvor definierten *Flüsse* via Drag&Drop auf den *Interface Block* gezogen werden. Dadurch entstehen automatisch die *flow properties*. Diesen *Fluss* kann man jetzt spezifizieren und die Richtung definieren. Der Standard ist die doppelte Flussrichtung „in-out“, dieser wird durch das Ändern der Benennung in „in“ oder „out“ verändert. Dazu einfach zweimal auf die *flow property* klicken.



Dies ist für alle weiteren Schnittstellen zu wiederholen. Damit ist die Definition der Schnittstellen abgeschlossen.

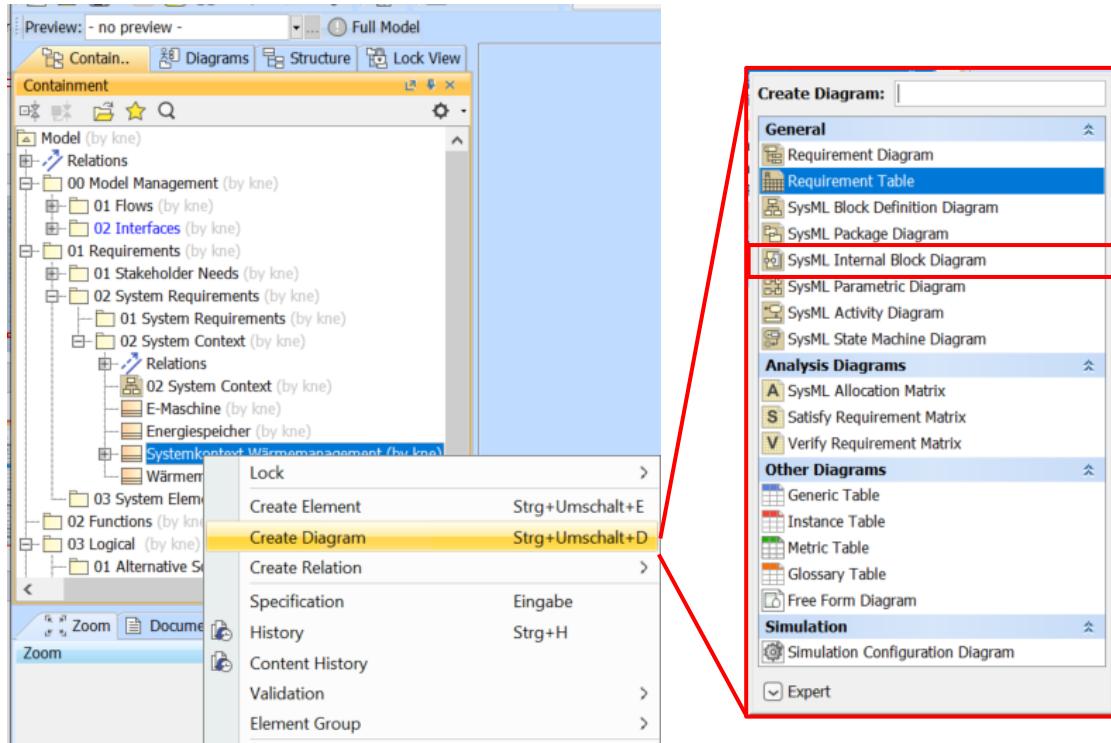


## 3.2 Erstellung eines Context Diagramms für den Systemkontext

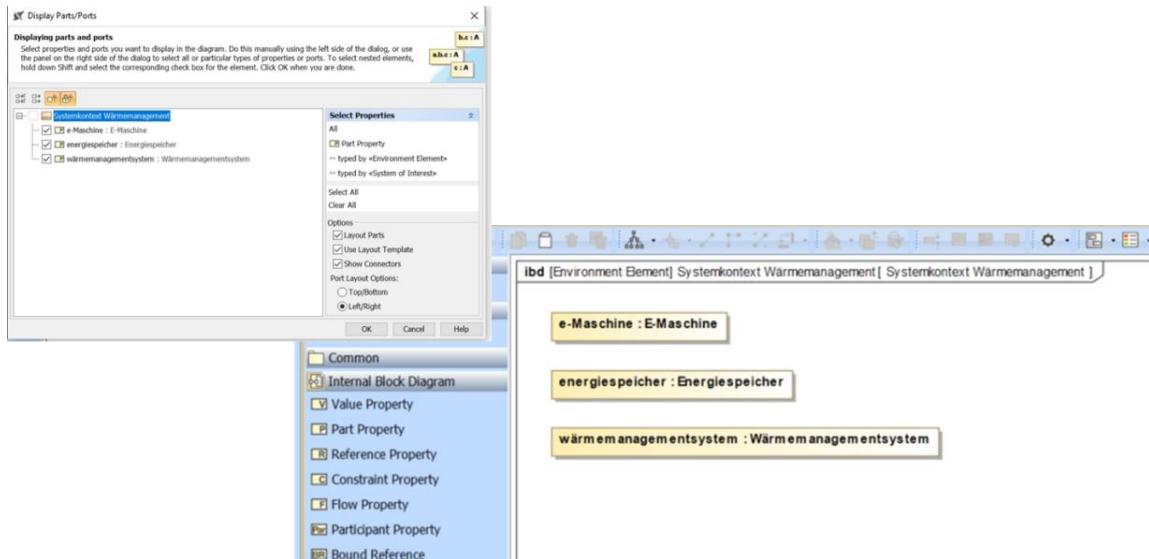
Nachdem alle notwendigen *Blöcke* definiert sind, kann mit der Modellierung der Zusammenhänge begonnen werden.

### 3.2.1 Modellierung des Kontextdiagramms

Zur Modellierung des Systemkontextes wird ein Internal Block Diagram (IBD) verwendet. Dazu wird der definierte *Block* „Systemkontext“ ausgewählt und nach einem Rechtsklick „Create Diagramm“ → Internal Block Diagram ausgewählt.



Bei dem sich daraufhin öffnenden Dialogfenster mit „OK“ bestätigen. Dadurch werden alle zuvor definierten Parts dieses *Blocks* automatisch in dem IBD angezeigt.

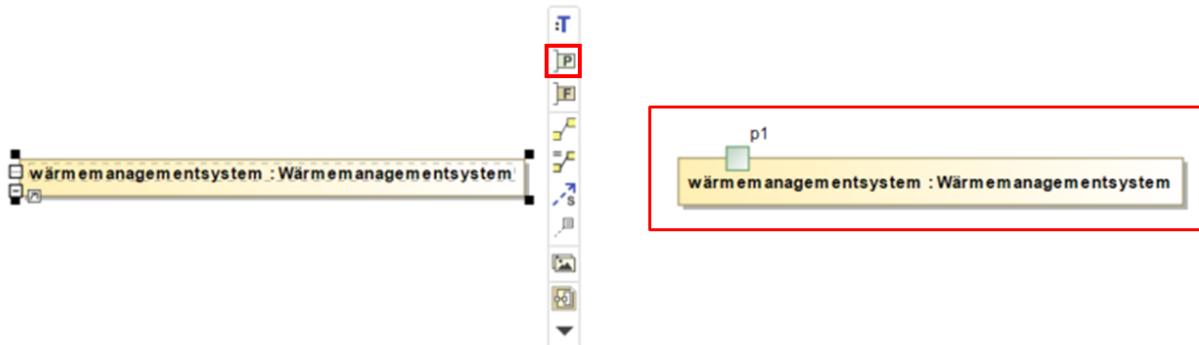


Sollten Umfeldelemente fehlen müssen diese zunächst in dem Block Definition Diagram (BDD) des Systemkontexts definiert werden (Schritt 3.1.1) und können anschließend entweder via Drag&Drop aus dem Containment-Tree, oder via Rechtsklick in das Diagram → Display → Parts/Ports eingeblendet werden.

Das System of Interest wird in die Mitte verschoben und die Umfeldelemente werden drum herum angeordnet.

### 3.2.2 Modellierung der Interaktion zwischen System of Interest und den Umfeldelementen

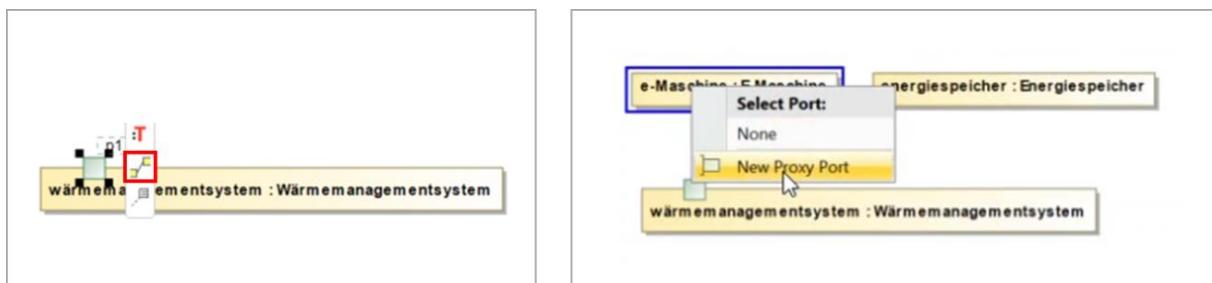
Anschließend werden die Interaktionen zwischen dem System of Interest und den Umfeldelementen modelliert. Dazu wird zunächst auf das „System of Interest“ geklickt und in der Seitenauswahl die Schaltfläche „Proxy Port“ ausgewählt. *Ports* sollten immer zuerst am System of Interest angelegt werden.



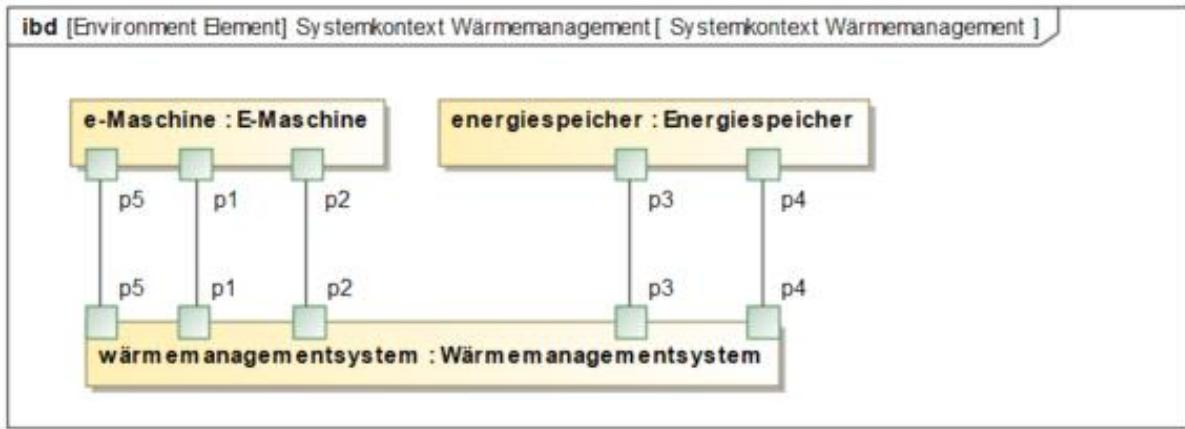
#### Tool-Tip:

Durch das Klicken auf die Schaltfläche „Proxy Port“ wird ein Port angelegt. Dieser erhält automatisch einen Namen p1,p2,...pn. Diese müssen nicht umbenannt werden, da Sie in der Semantik von SysML keine weitere Rolle spielen.

Um nun eine Interaktion zwischen *Ports* zu modellieren, wird der erstelle *Port* angeklickt und auf der erscheinenden Schaltfläche „Connector“ ausgewählt. Anschließend wird der *Part* (bspw. ein Umgebungselement) angeklickt, zu dem die Interaktion definiert werden soll. Dabei erscheint eine Schaltfläche in der „New Proxy Port“ ausgewählt wird.

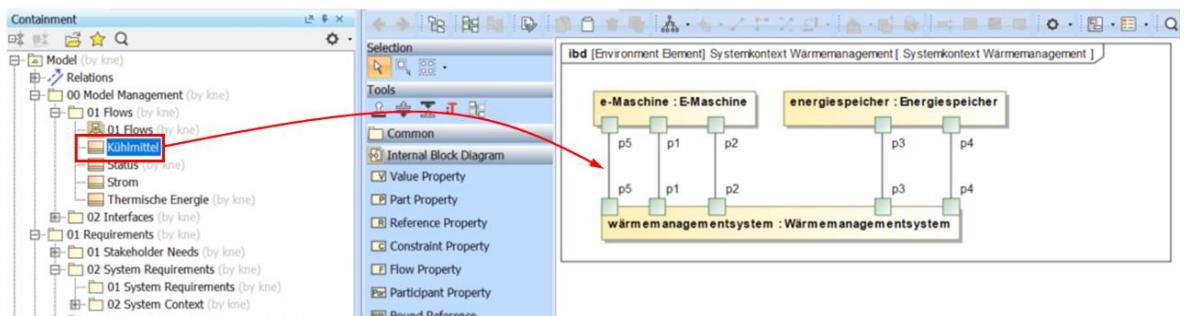


Damit ist die Erste Interaktion modelliert. Mit allen weiteren wird analog fortgefahrene, bis alle Interaktionen definiert sind.

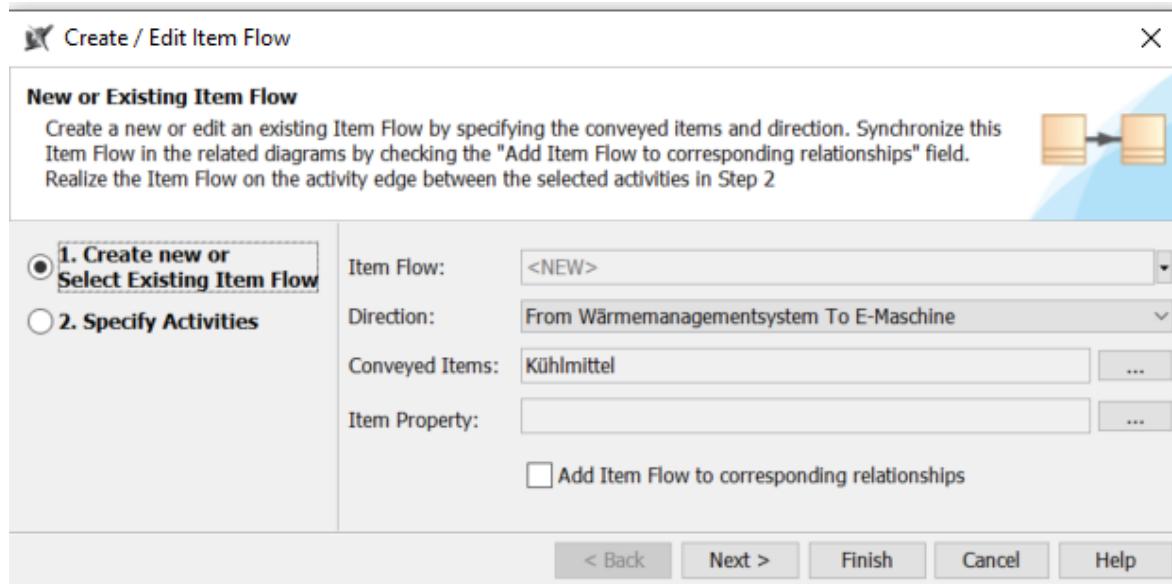


### 3.2.3 Modellierung von Flüssen

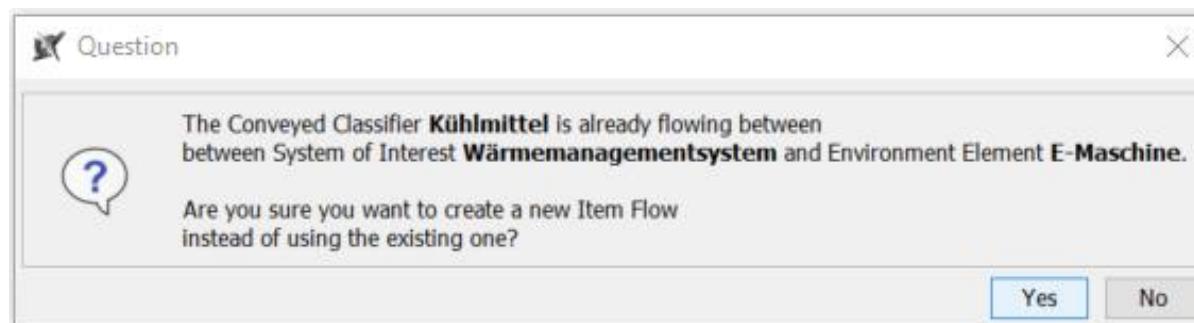
Nach dem die Interaktionen definiert sind, werden im nächsten Schritt die *Flüsse* die über diese Interaktionen zwischen dem System of Interest und den Umfeldelementen ausgetauscht werden modelliert. Dafür werden die zuvor definierten *Flüsse* wiederverwendet. Der *Fluss* wird in Containment-Tree ausgewählt und mittels Drag&Drop auf einen *Connector* gezogen.



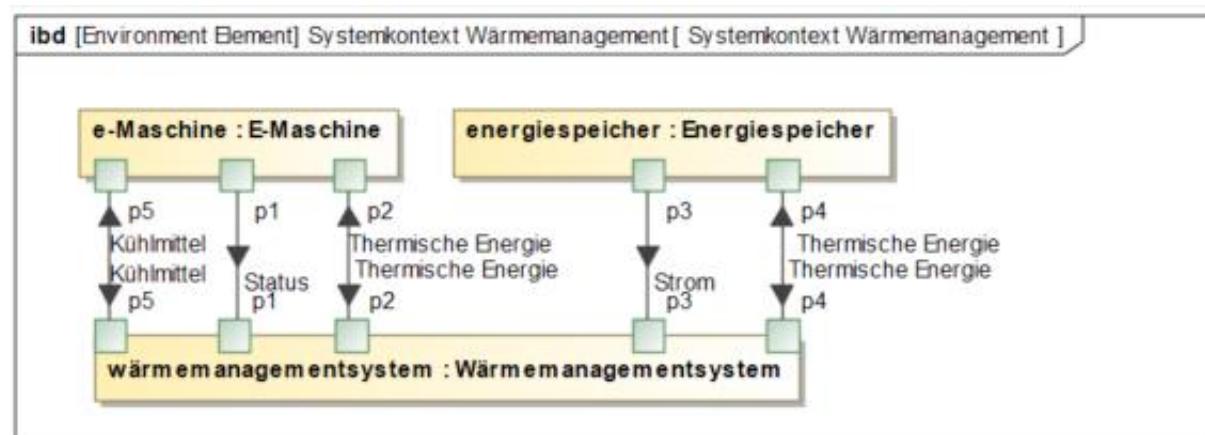
Dabei erscheint ein Dialogfeld. Hier kann über die Option „Direction“ angegeben werden in welche Richtung der *Fluss* fließen soll. Über „Finish“ wird der *Fluss* dann angelegt.



Soll ein *Fluss* bidirektional sein, so ist der Vorgang zu wiederholen und in diesem Dialogfeld entsprechend der Richtung zu ändern. Dabei erscheint eine Schaltfläche, die mit „Yes“ bestätigt wird.



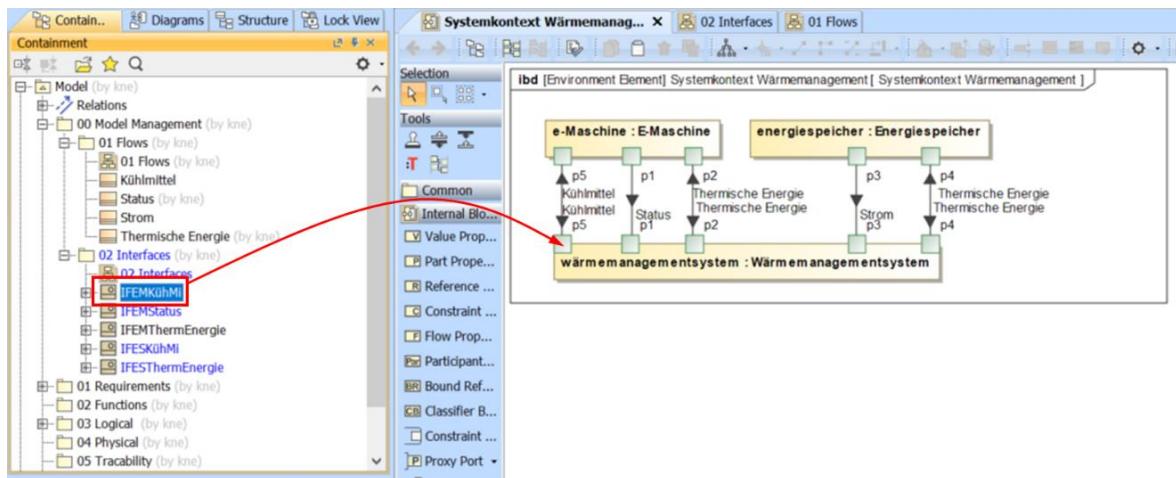
Dies wird nun für alle Interaktionen wiederholt, bis sämtliche *Flüsse* definiert sind. Dabei kann es passieren, dass *Flüsse* identifiziert werden, die noch nicht definiert sind. Diese nicht vorhandenen *Flüsse* müssen dann wie in 3.1.2 beschrieben definiert werden.



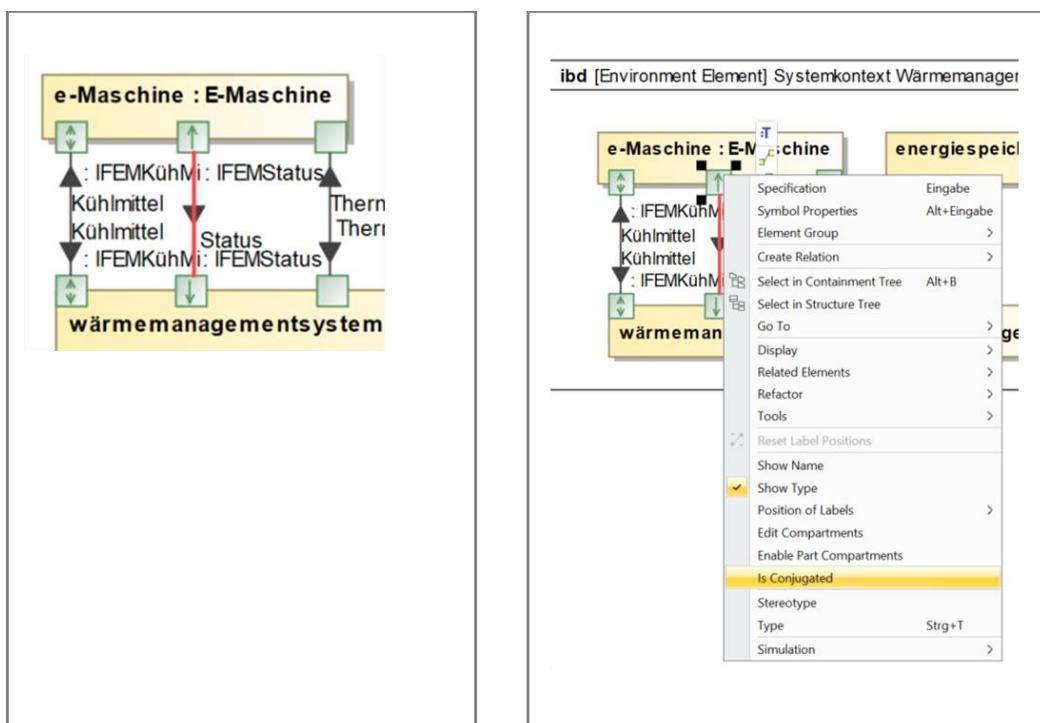
### 3.2.4 Modellierung von Schnittstellen

Der letzte Schritt der Systemkontextmodellierung ist die Modellierung der Schnittstellen. Dafür werden die in Abschnitt 3.1.3 definierten *Interface Blocks* wiederverwendet.

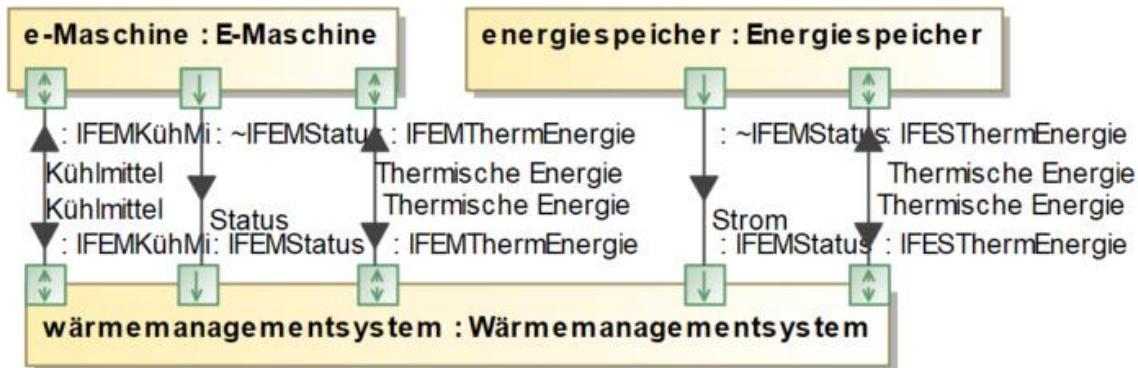
Die zuvor definierten *Interface Blöcke* werden im Containment-Tree ausgewählt und mittels Drag&Drop auf einen *Port* gezogen. Dabei ist zu beachten, dass der vorher definierte *Item Flow* des *Interface Blocks* zu dem *Fluss* auf dem *Connector* passt.



Analog wird dies für den gegenüberliegenden *Port* des Umgebungselement wiederholt. Ist in dem *Interface Block* nur eine Flussrichtung definiert führt dies zu einem Fehler, den das Werkzeug automatisch erkennt, da nun beide *Blöcke* einen In-Flow haben. Um diesen Fehler zu beheben, wird mit Rechtsklick der *Port* des Umgebungselements ausgewählt und die Schaltfläche „is conjuncted“ ausgewählt. Damit wird der *Interface Block* konjugiert, also umgekehrt.



Dies wird für alle Schnittstellen wiederholt, bis der Systemkontext fertig modelliert ist.



### Tool-Tip:

Catia Magic ermöglicht es Blöcke automatisch mittels „Layout“ zueinander auszurichten.



Um das Modell übersichtlicher zu machen, gibt es die Möglichkeit *Ports* und deren Typen (*Interface Blöcke*) ein- und auszublenden. Dazu Rechtsklick auf einen (oder mehrere ausgewählte) *Ports* und „Show Name“ und „Show Type“ an oder abwählen.

