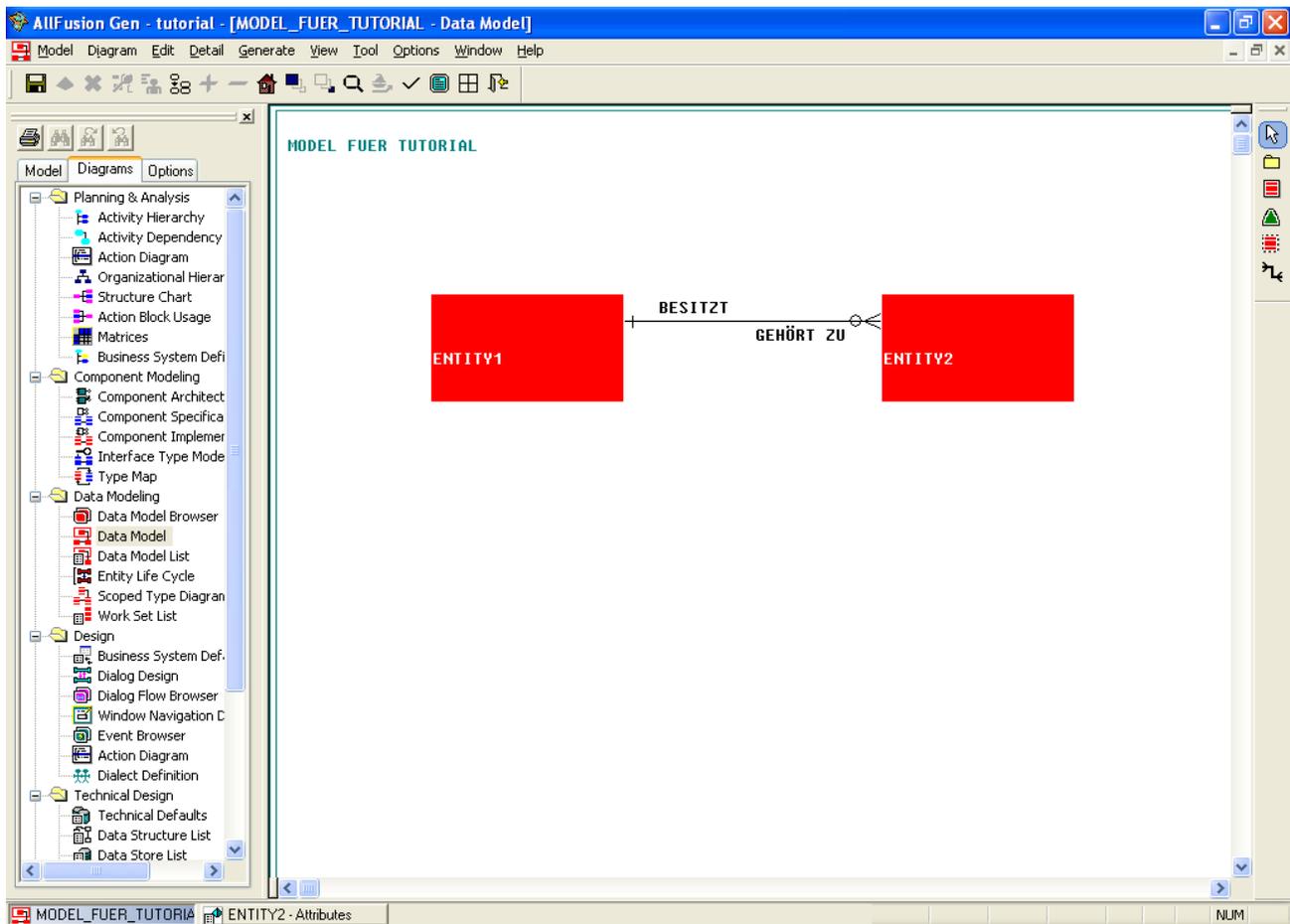


10. Beziehungen im Dialog Design

In Kapitel 3 haben wir uns schon kurz damit auseinandergesetzt, wie man im Data Model eine Beziehung zwischen zwei Entitätstypen herstellt. In diesem Kapitel wollen wir nun diese Beziehung auch bei der Verwaltung von Objekten implementieren.

Hier nochmal die Ausgangssituation. Wir haben zwei Entitäten und bereits die Beziehung zwischen diesen definiert.



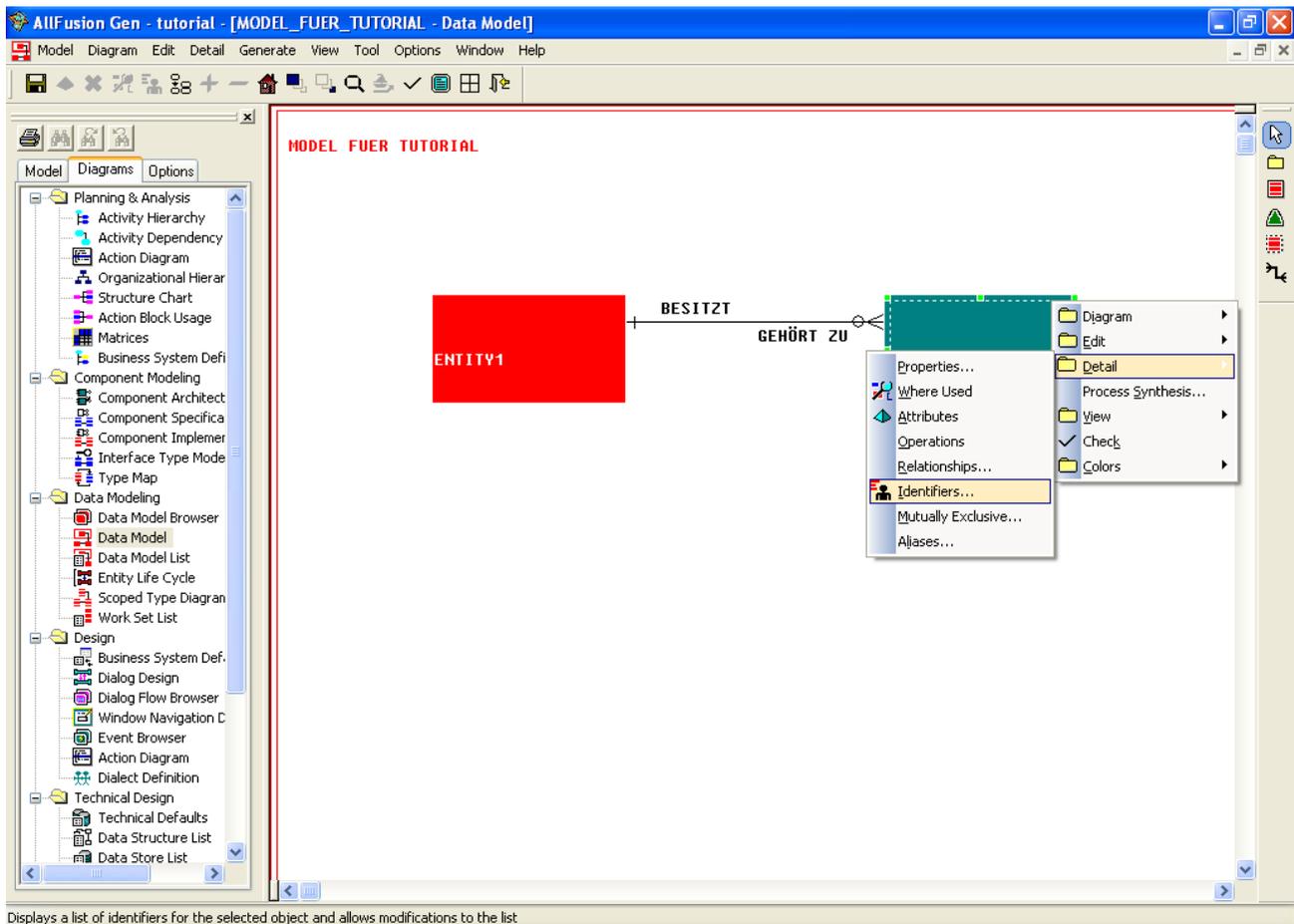
Jedoch besitzt die Entität "Entity2" noch keinen Primärschlüssel.

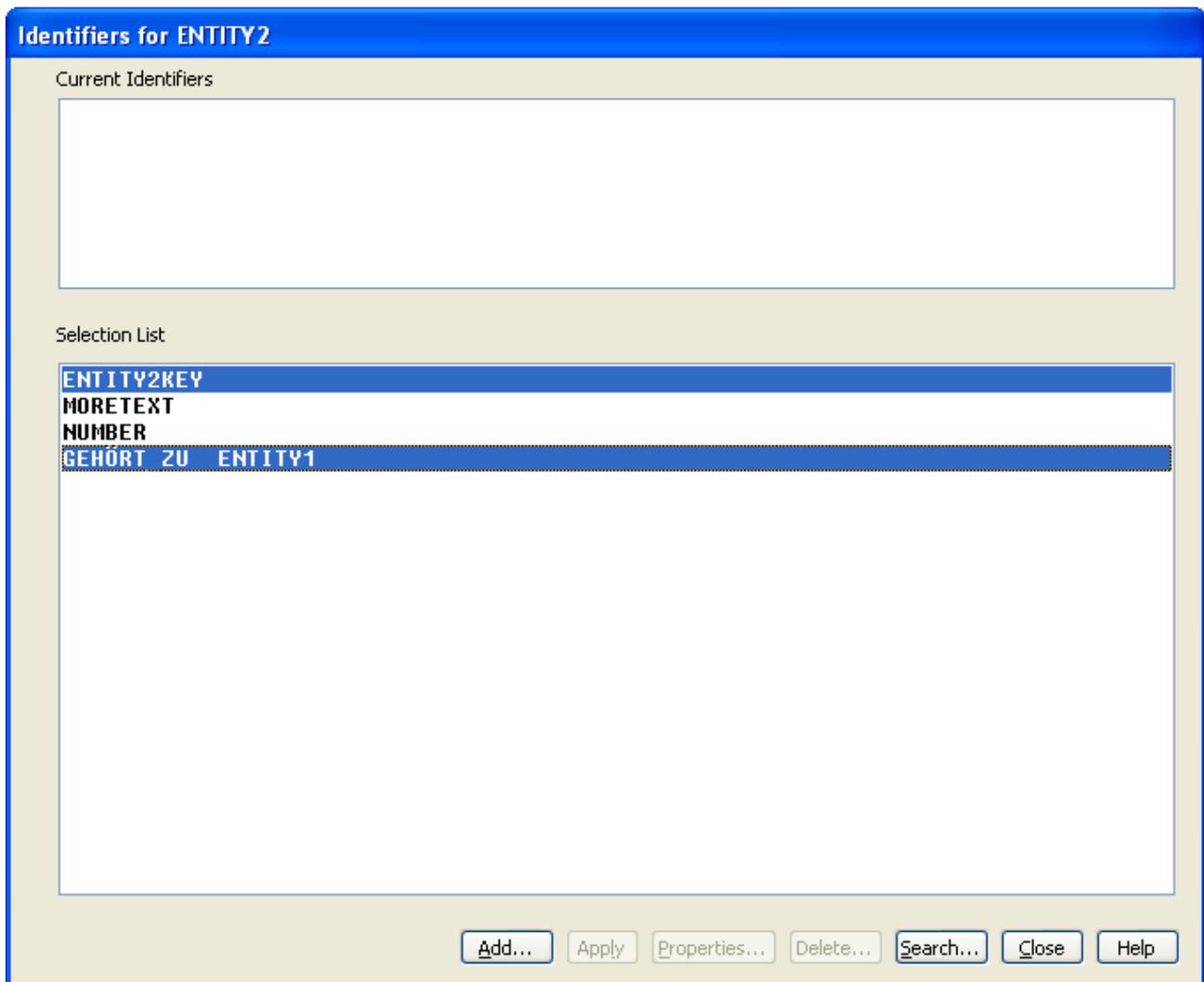
Nun gibt es zwei Möglichkeiten:

Man definiert einen Primärschlüssel wie auch bei Entity1 einfach aus einem einzelnen Attribut. Dies hat jedoch einen Nebeneffekt. Angenommen Entity1 ist "Stadt" und Entity2 ist "Haus". Der Primärschlüssel von "Haus" ist die Hausnummer. Nun können in der gesamten Datenbank **nicht** zwei Häuser mit der selben Hausnummer existieren, weil die Hausnummer eindeutig sein muss.

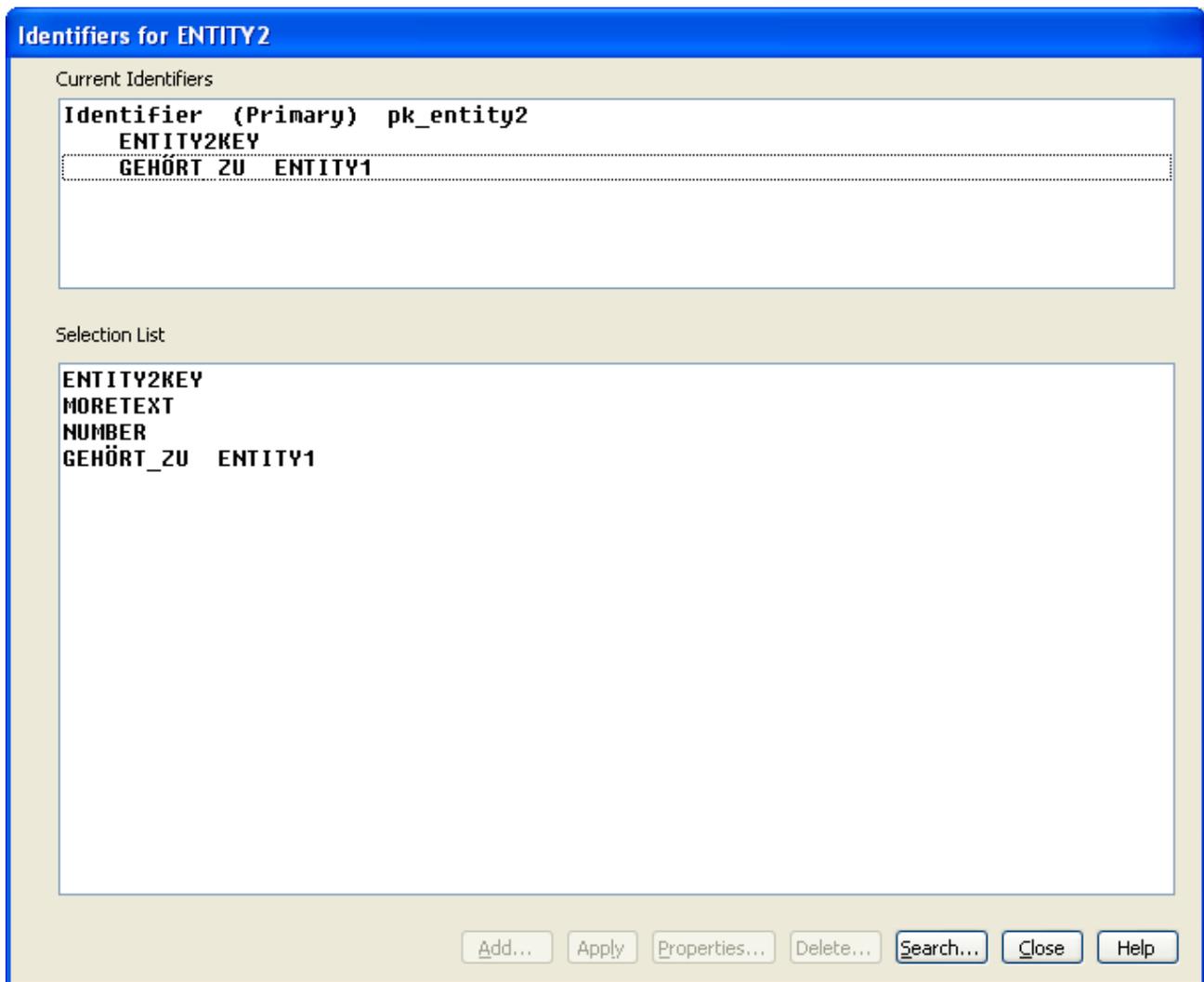
Deshalb die zweite Möglichkeit: Man konstruiert einen Primärschlüssel aus einem Attribut der zweiten Entität und dem Primärschlüssel der ersten Entität. Somit muss die Hausnummer nur eindeutig sein bei Häusern die zur selben Stadt gehören.

Öffnen Sie das Fenster zum Erstellen der Primärschlüssel.



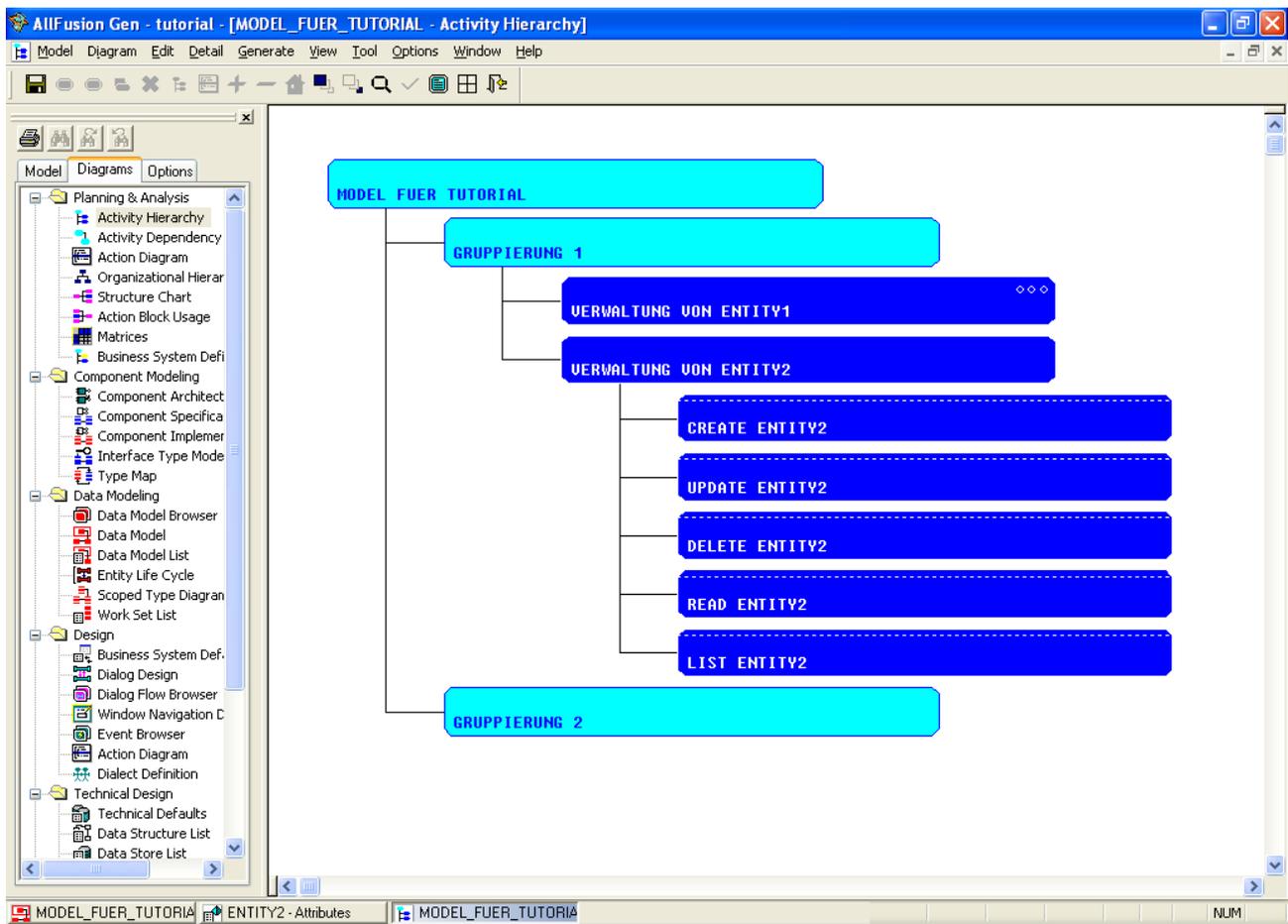


Hier sehen Sie jetzt neben den von Ihnen erstellten Attributen eine weitere Auswahlmöglichkeit: Die Beziehung zur ersten Entität. Wählen Sie sowohl die Beziehung als auch ihr Schlüsselattribut aus und klicken Sie auf "Add". Geben Sie dem Primärschlüssel einen Namen und klicken Sie auf OK.

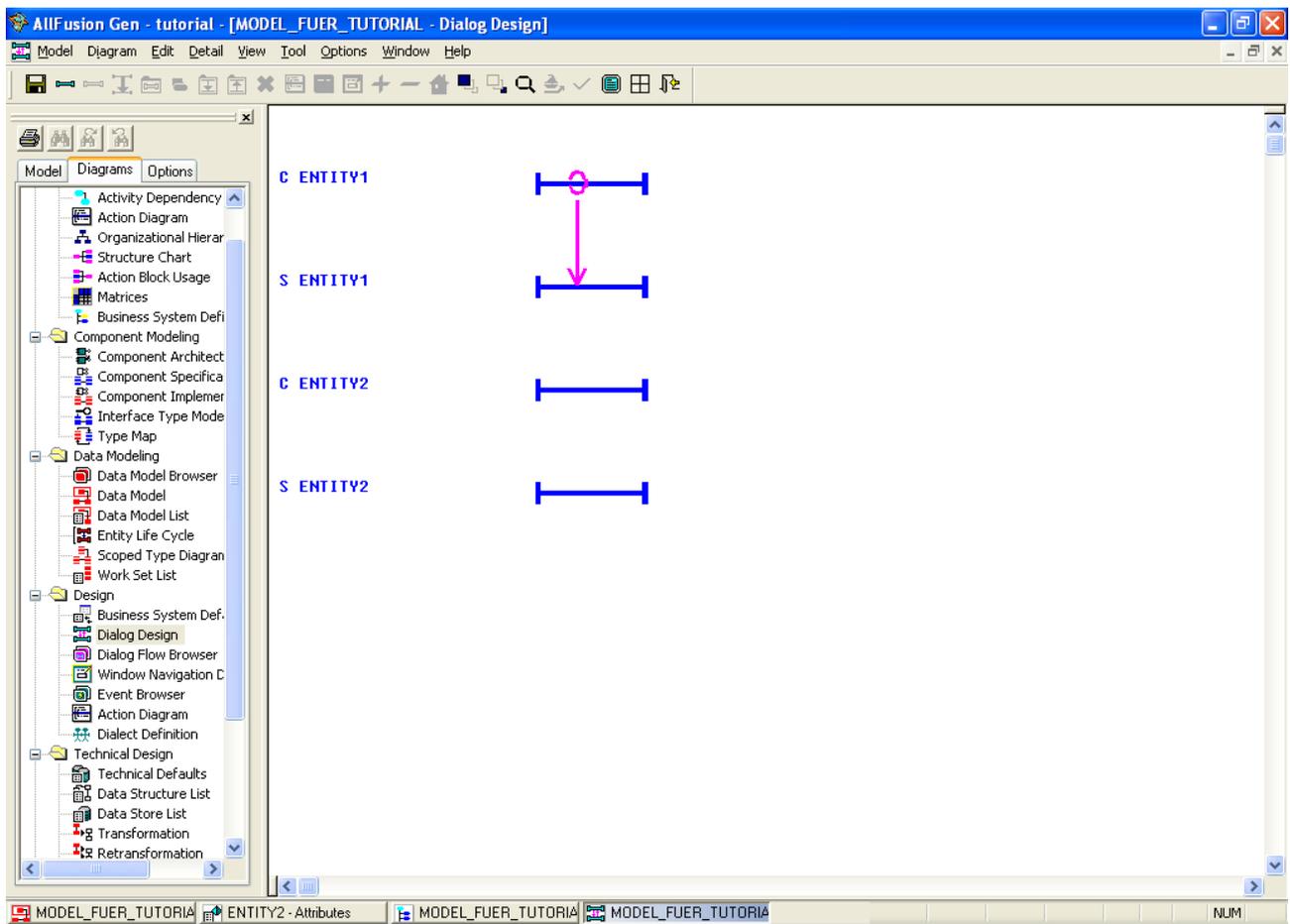


Nun haben Sie einen zusammengesetzten Primärschlüssel.

Die folgenden Schritte haben Sie bereits in den anderen Kapiteln behandelt und werden nun nur verkürzt dargestellt. Wenn Sie sich nicht sicher sind, wie sie diese Schritte durchführen, schauen Sie nochmal in die entsprechenden Kapitel.

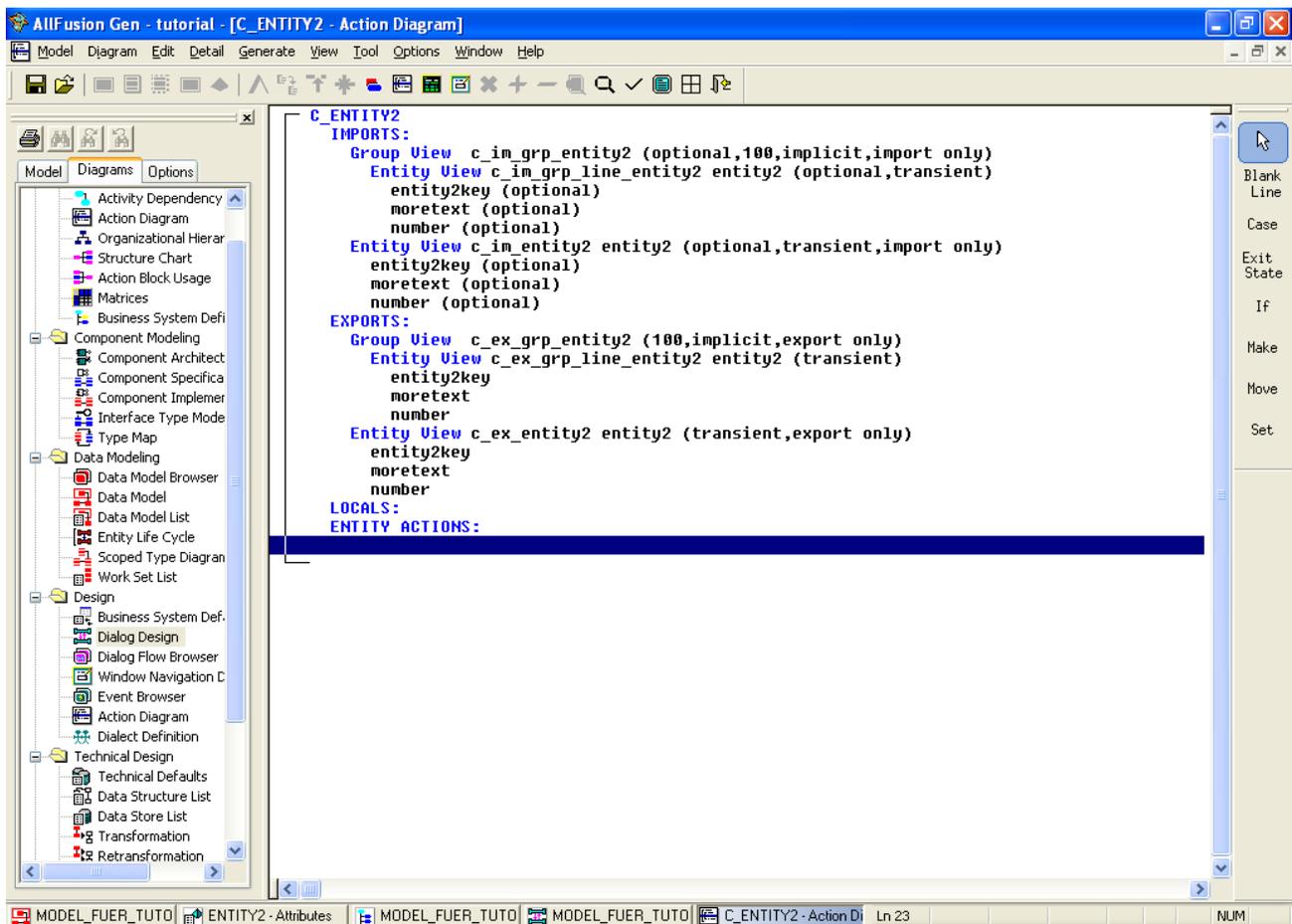


Zunächst generieren Sie sich für Ihre zweite Entität die Elementarprozesse. Überlegen Sie sich bitte vorher, ob diese Entität zum selben Business System gehören soll, sprich in welchen der Untergruppen Sie den Prozess erstellen.

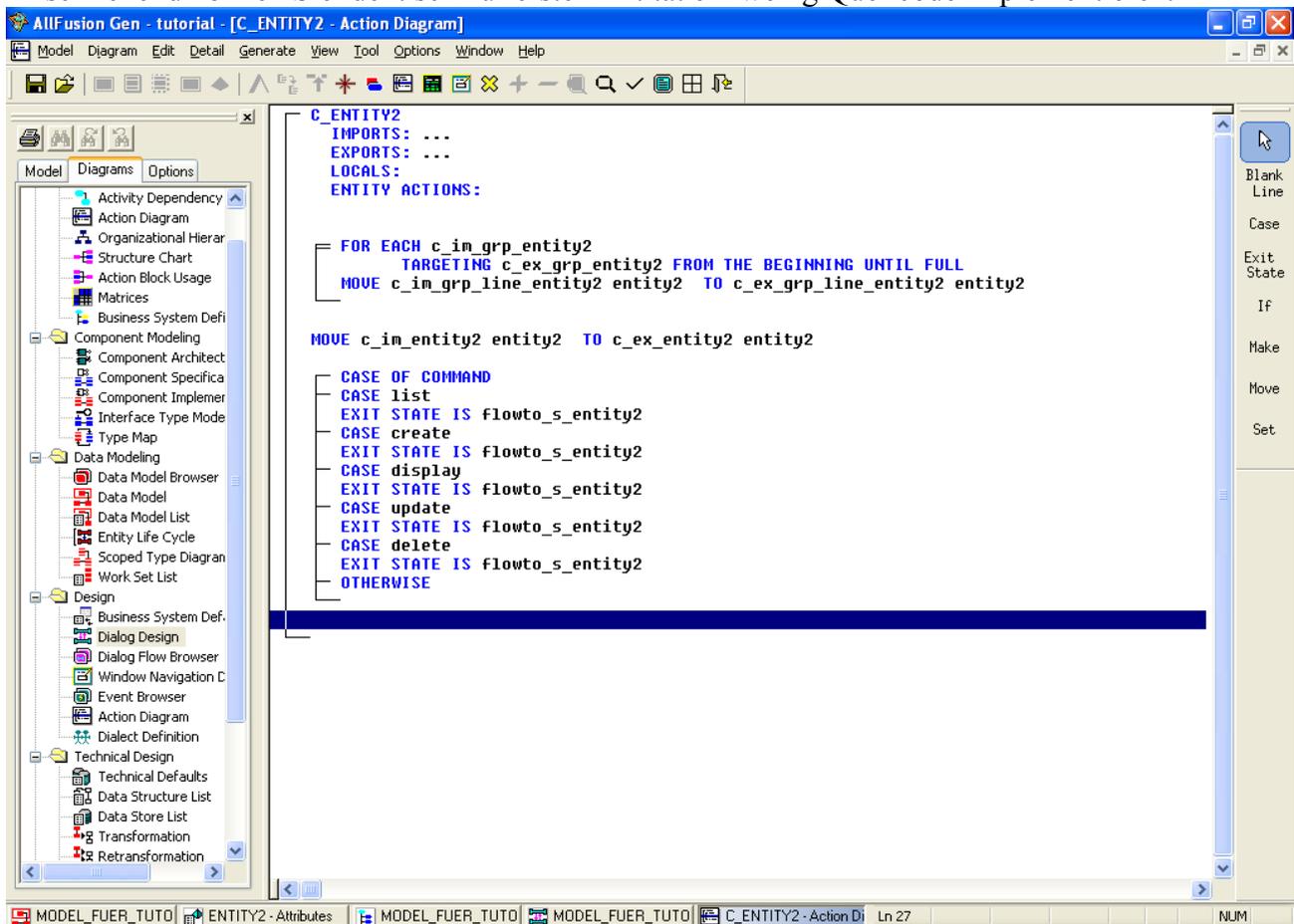


Anschließend erstellen Sie für ihre neue Entität wie schon zuvor für die erste Entität zwei Dialoge: Client und Server. Denken Sie den Server so umzustellen, dass er kein Display besitzt (Kapitel 6).

Öffnen Sie anschließend das Action Diagram für den Client.

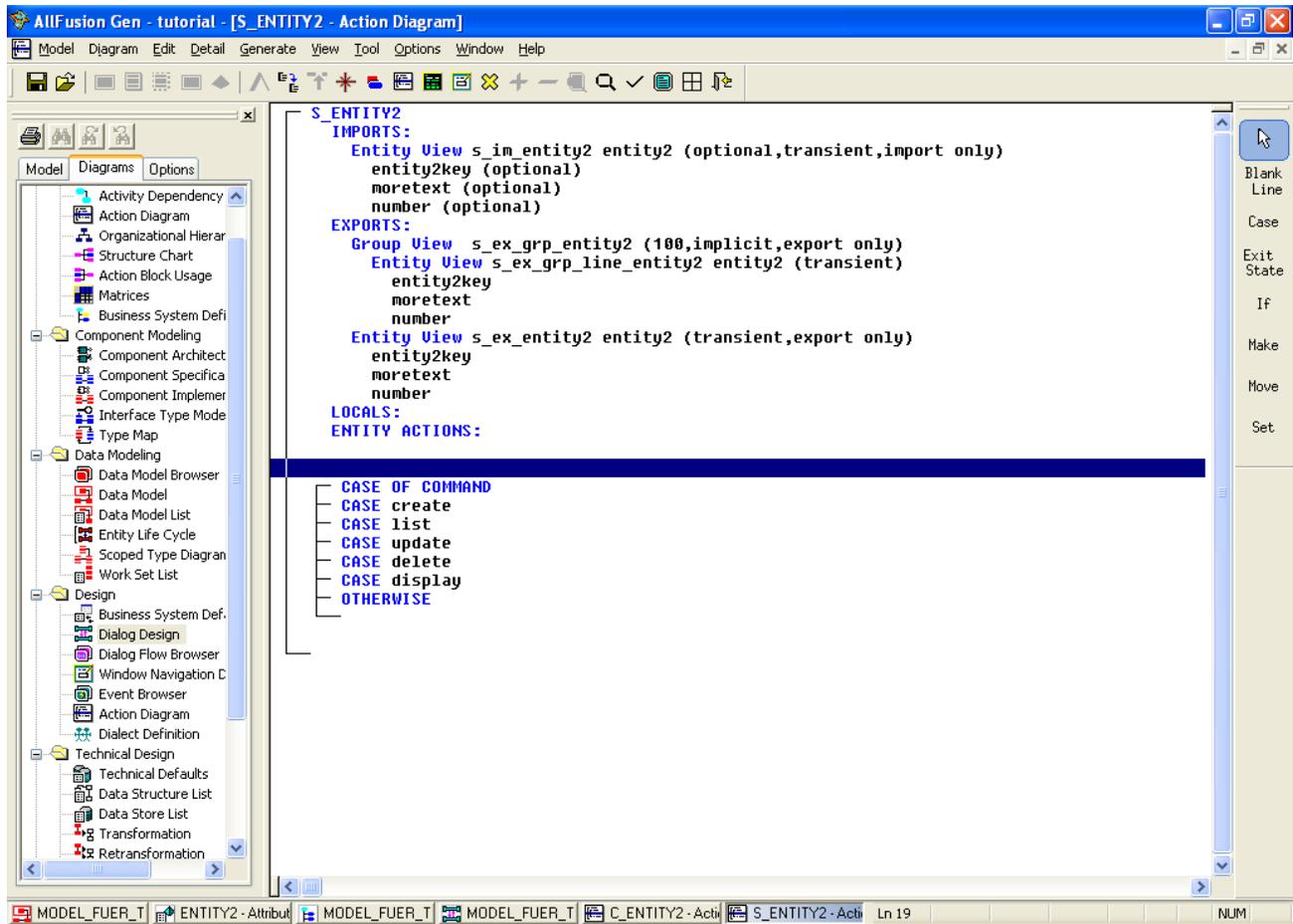


Erstellen Sie je eine Group View und eine Entity View wie im Bild oben für ihre zweite Entität. Anschließend können Sie identisch zur ersten Entität ein wenig Quellcode implementieren.

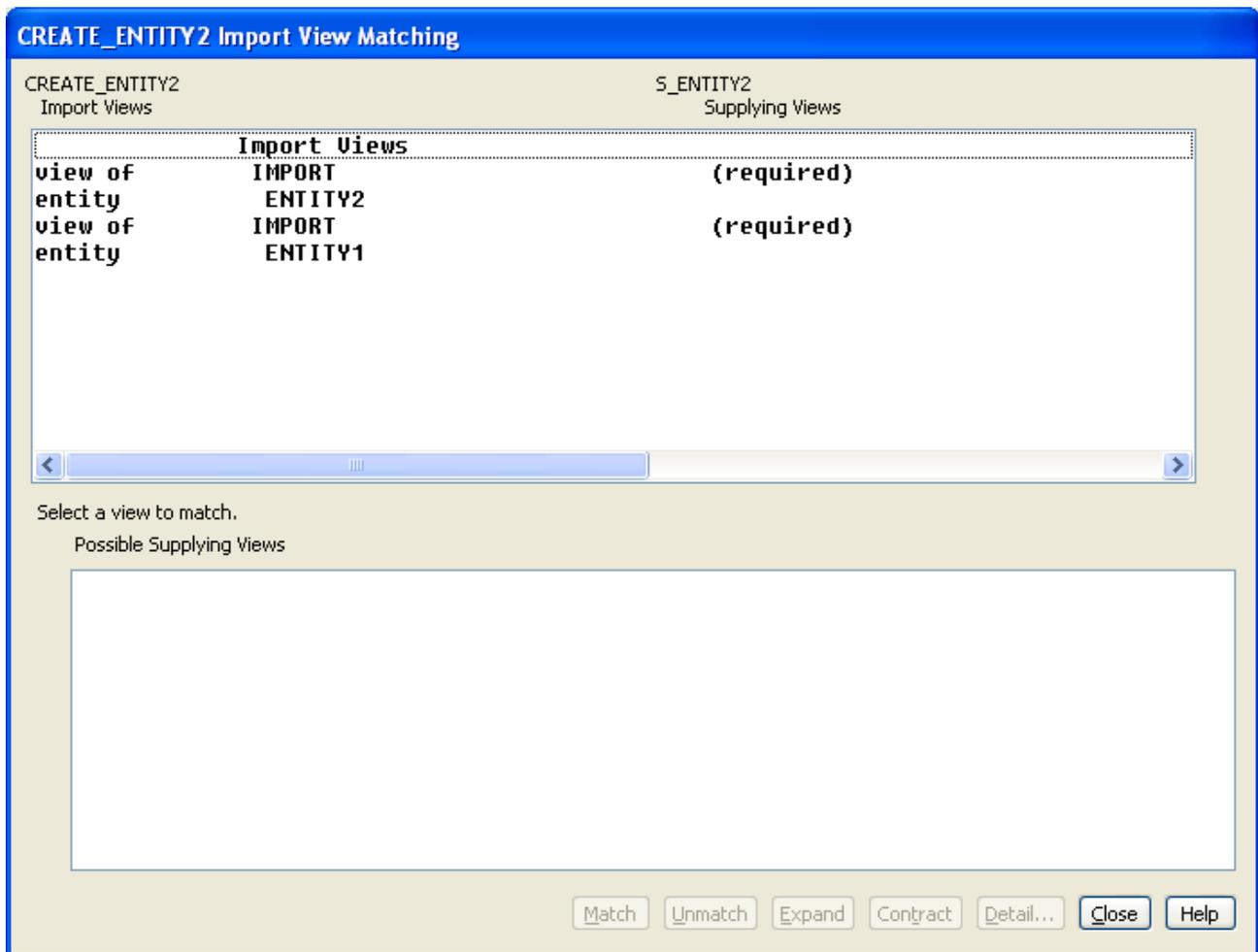


Hinweis: Denken Sie daran einen neuen Exit State zu erzeugen für ihre neue Entität.

Als nächstes öffnen Sie das Action Diagram des Servers. Implementieren Sie analog zum Server der ersten Entität die Views und ein "Case of Command".

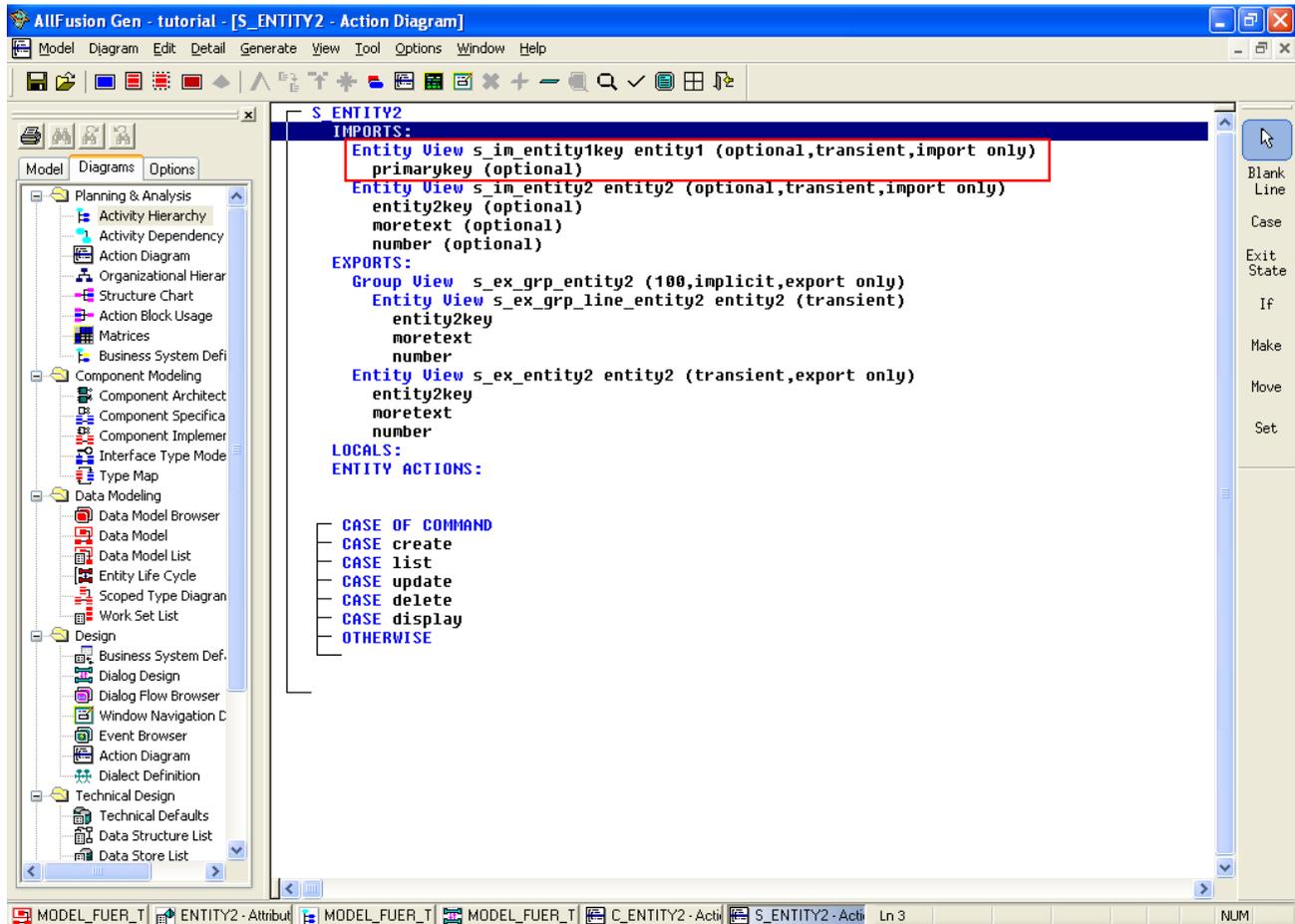


Sie werden jetzt aber auf ein Problem stoßen, wenn Sie versuchen mithilfe des "Use" Statements einen der Elementarprozesse zu verwenden.

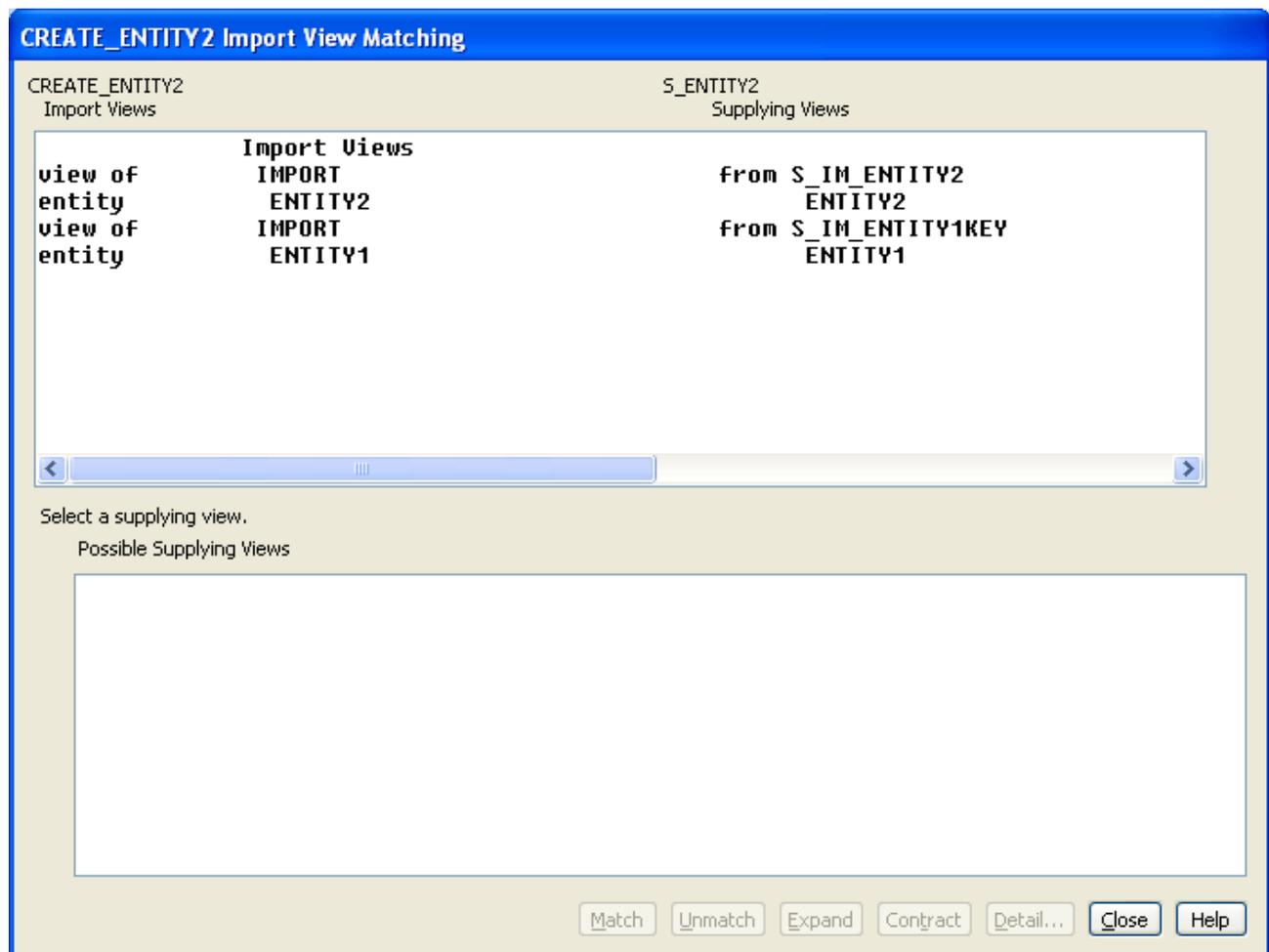


Der von Ihnen generierte Elementarprozess "Create_Entity2" benötigt **zwei** statt nur einer Entity View als Input. Das hängt damit zusammen, dass dies letztlich die Beziehung definiert. In der Datenbank ist dies der Fremdschlüssel, den der Eintrag in "Entity" erhält.

Fügen Sie dem Server also eine Entity View vom Typ der ersten Entität hinzu. Als Attribute brauchen Sie diesmal **nur** den Primärschlüssel.



Wenn Sie dies getan haben, besitzen Sie nun ein View, dass für die Use Statements verwendet werden kann. Also können Sie nun den Server fertig implementieren.



AIIFusion Gen - tutorial - [S_ENTITY2 - Action Diagram]

Model Diagram Edit Detail Generate View Tool Options Window Help

Entity View s_ex_grp_line_entity2 entity2 (transient)
 entity2key
 moretext
 number

Entity View s_ex_entity2 entity2 (transient,export only)
 entity2key
 moretext
 number

LOCALS:
 ENTITY ACTIONS:

CASE OF COMMAND
 CASE create
 USE create_entity2
 WHICH IMPORTS: Entity View s_im_entity2 entity2 TO Entity View import entity2
 Entity View s_im_entity1key entity1 TO Entity View import entity1
 WHICH EXPORTS: <none> FROM Entity View export entity2
 <none> FROM Entity View export entity1

CASE list
 USE list_entity2
 WHICH EXPORTS: Group View s_ex_grp_entity2 FROM Group View group_export

CASE update
 USE update_entity2
 WHICH IMPORTS: Entity View s_im_entity1key entity1 TO Entity View import entity1
 Entity View s_im_entity2 entity2 TO Entity View import entity2
 WHICH EXPORTS: <none> FROM Entity View export entity2

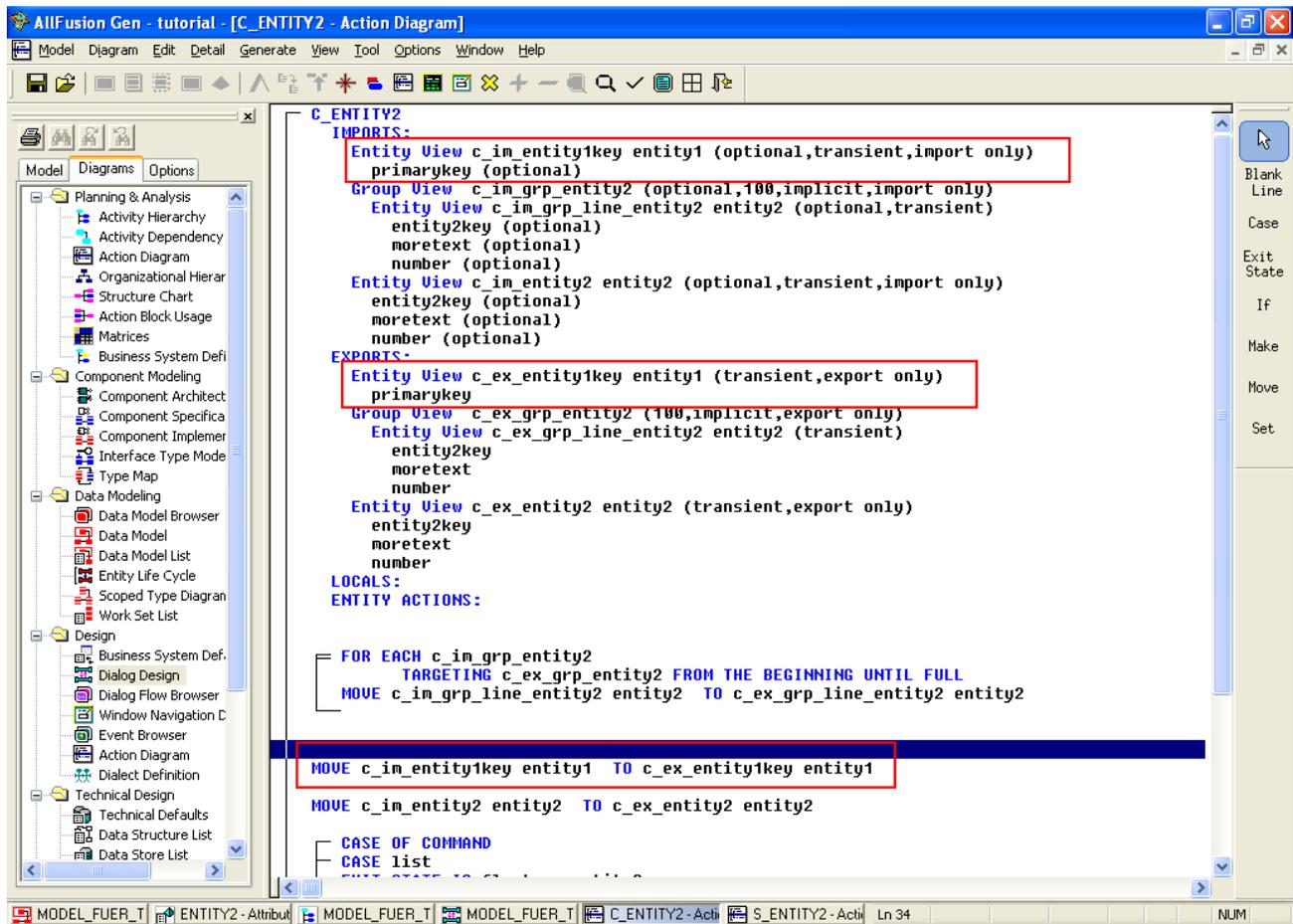
CASE delete
 USE delete_entity2
 WHICH IMPORTS: Entity View s_im_entity1key entity1 TO Entity View import entity1
 Entity View s_im_entity2 entity2 TO Entity View import entity2
 WHICH EXPORTS: <none> FROM Entity View export entity2

CASE display
 USE read_entity2
 WHICH IMPORTS: Entity View s_im_entity1key entity1 TO Entity View import entity1
 Entity View s_im_entity2 entity2 TO Entity View import entity2
 WHICH EXPORTS: Entity View s_ex_entity2 entity2 FROM Entity View export entity2

OTHERWISE

MODEL_FUER_T ENTITY2 - Attribut MODEL_FUER_T MODEL_FUER_T C_ENTITY2 - Acti S_ENTITY2 - Acti Ln 21 NUM

Nun müssen Sie den Client noch erweitern. Fügen Sie wie beim Server eine weitere Entity View ein, diesmal aber beim Im- und Export. Außerdem brauchen Sie wie bei jeder Entity View im Client ein Move Statement. Die Änderungen sind im Bild rot markiert.



Wenn Sie hiermit fertig sind, können Sie den Join zwischen den beiden Dialogen aufbauen. Hier läuft es genauso ab wie schon bei der ersten Entität, Sie müssen aber zusätzlich bei "Data Sent" die Entity View mit dem Primärschlüssel zwischen Server Import und Client Export matchen.

Danach können Sie die Oberfläche bauen. Gehen Sie hier genauso vor wie bei der ersten Entität. Fügen Sie für jedes Attribut aus der zweiten Entität ein Textfeld ein, für jeden Befehl einen Commandbutton und eine Listbox für die Anzeige aller Datenbankeinträge von der zweiten Entität.

Entry Field Properties

Attribute

Import View: C_IM_ENTITY1KEY

Export View: C_EX_ENTITY1KEY

Entity: ENTITY1

Name: PRIMARYKEY

Domain: Number

Length: 3 Case Sensitive

Decimals: 0 Varying Length

Field

Name: txtEntity1Key

Prompt: Entity 1 Key:|

Edit Pattern: <Default Edit Pattern>

Edit Align: Default

Autoscroll Margin Box

Read Only Upper Case

Auto Tab Password

OK Cancel Help

Desweiteren fügen Sie ein Textfeld ein, welches als Import/Export View den View mit dem Primärschlüssel der ersten Entität benutzt, und als Attribut dieses Schlüsselattribut besitzt.

Entity2 Verwaltung

Entity2 Key: ZZ9

Another Text Field: XXXXXXXXXXXX

Number: ZZZZ9

Entity 1 Key: ZZ9

Entity2 Key: Another Text Field: Number:

ZZ9	XXXXXXXXXXXX	ZZZZ9

Erstellen Ändern Anzeigen Löschen Liste

Fertig! Sie können das Programm nun testen (Siehe Kapitel 9).