Re-using *models* and creating new ones

A model is the mathematical function, or set of functions, of which your simulation gets its values. In the other chapters you have used the variables of a model to present a graph, or design an assignment. This chapter teaches you how to re-use an existing model and how to create new ones.

Re-using models

When you re-use an existing model, you do this in two general steps:

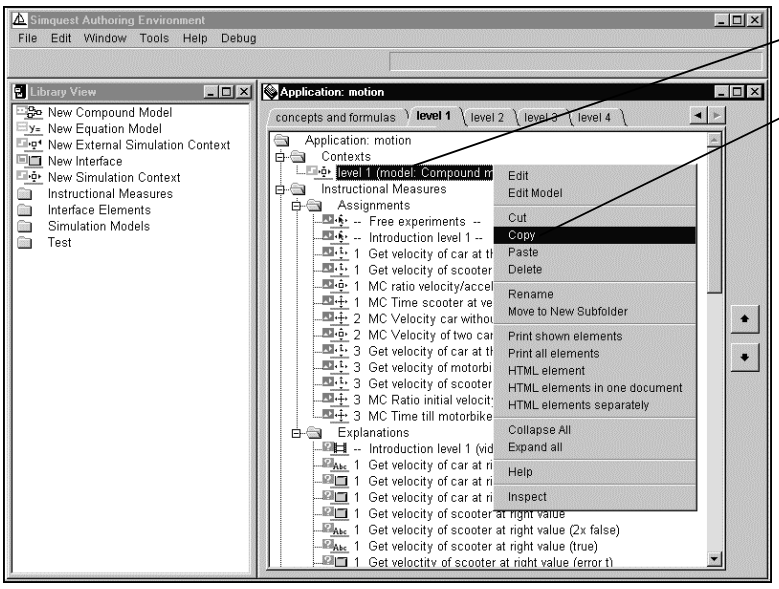
- copying and renaming an existing model
- saving and checking your work

Copying and renaming a model

Before you can re-use an existing model, you have to select the model you want to re-use, copy it, and rename the model.

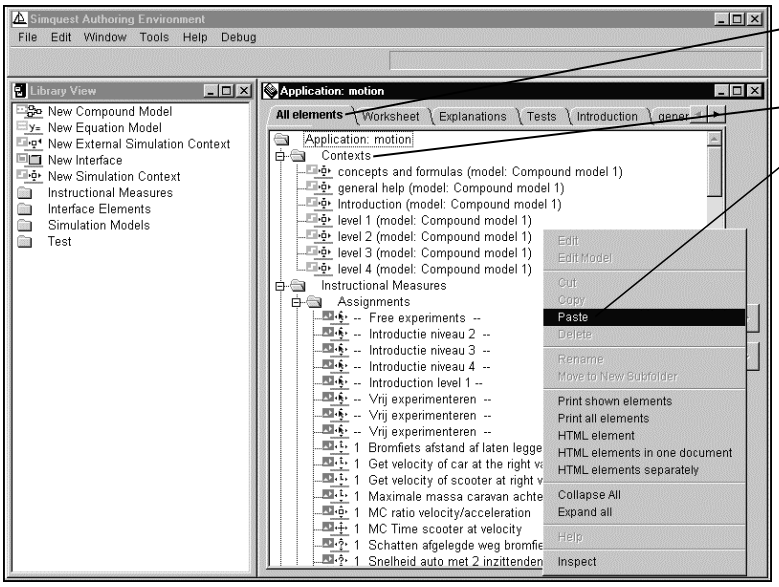
Selecting and copying a model

The motion application contains several models. Each model represents one level in your application. You have to keep the original model, because it is already used in the motion application. If you do remove the model, all the elements in that tabsheet also will be removed. Therefore, you now make a copy of an existing model and use the copied one to re-use.



1 In *level 1* in the *Application window*, select **level 1**

2 Click your right mouse button and choose **Copy**



3 Select the **All elements** tabsheet

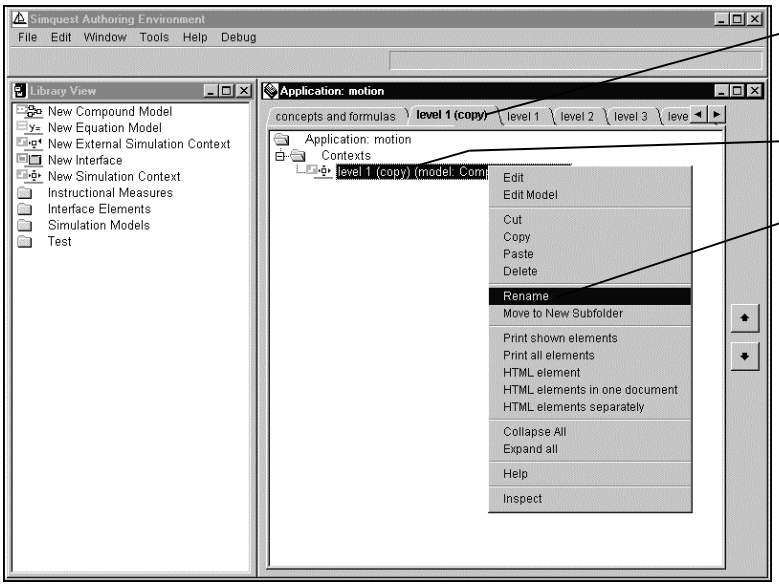
4 Select the folder **Contexts**

5 Click your right mouse button and choose **Paste**

Check if the model level 1 (copy) is added to the All elements tabsheet. Also check if a new tabsheet called level 1 (copy) is added to your application.

Renaming a model

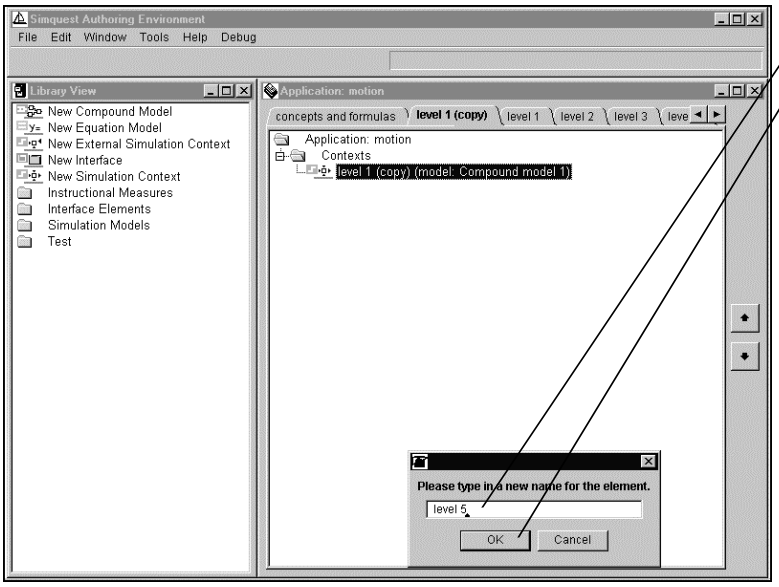
You can rename the model. When you change the name of the model, you automatically change the name of the tabsheet.



1 In the *Application window*, select the **level 1 (copy)** tabsheet

2 Select the model **level 1 (copy)**

3 Click your right mouse button and choose **Rename**

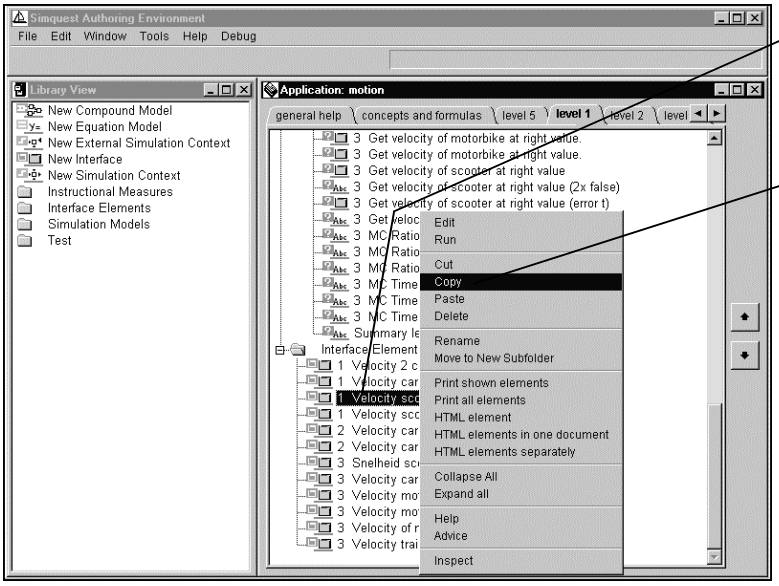


4 Type: `level 5`
5 Click **OK**

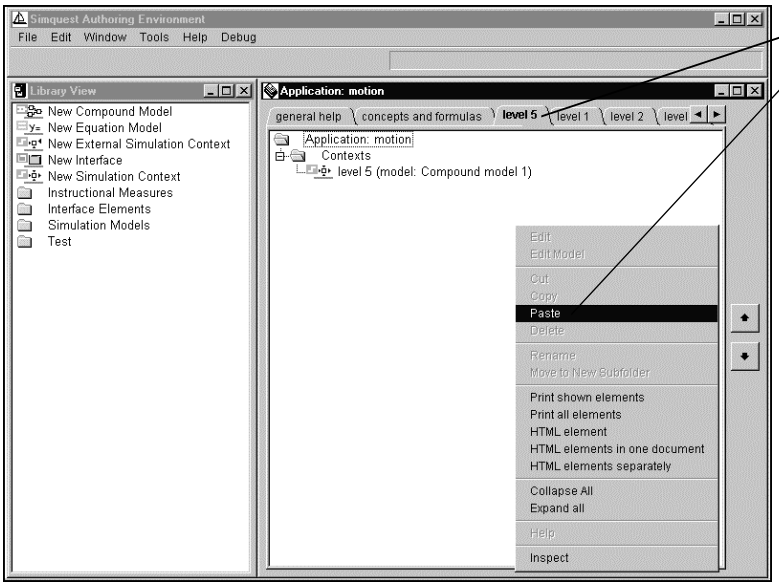
Check if the name of the tabsheet has also changed from level 1 (copy) into level 5.

Checking and saving your work

You have added a new level to the application using an existing model. To check if the model still works, you can copy an existing interface from level 1, paste it in level 5, and see if you can still use it properly.

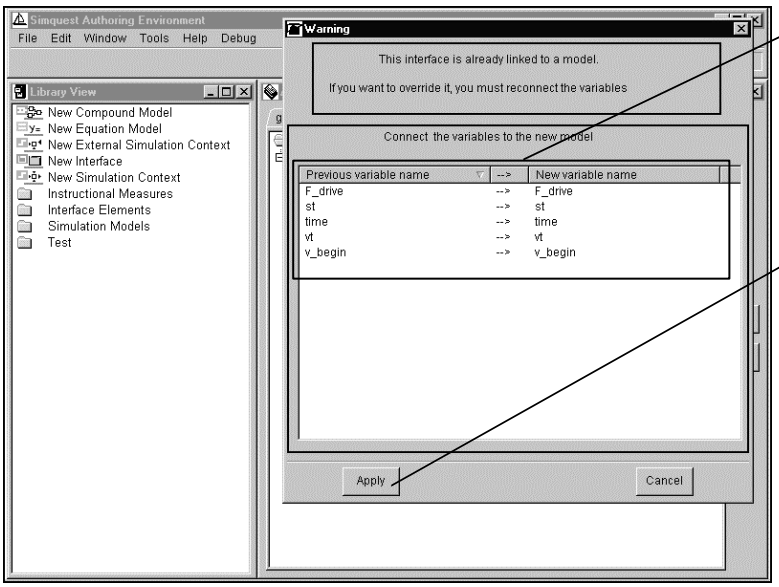


1 In the folder *Interface elements* of *level 1* of the *Application window*, select **1 Velocity scooter**
2 Click your right mouse button and choose **Copy**



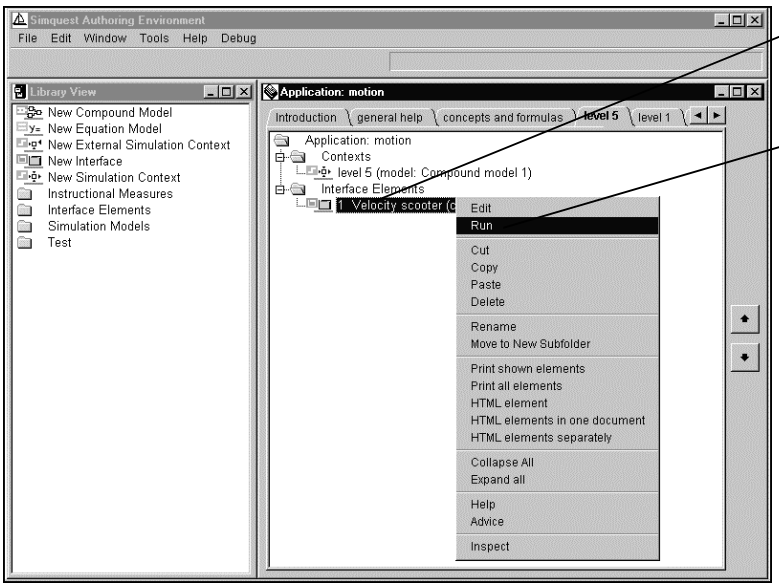
4 Select the **level 5** tabsheet

5 Click your right mouse button and choose **Paste**



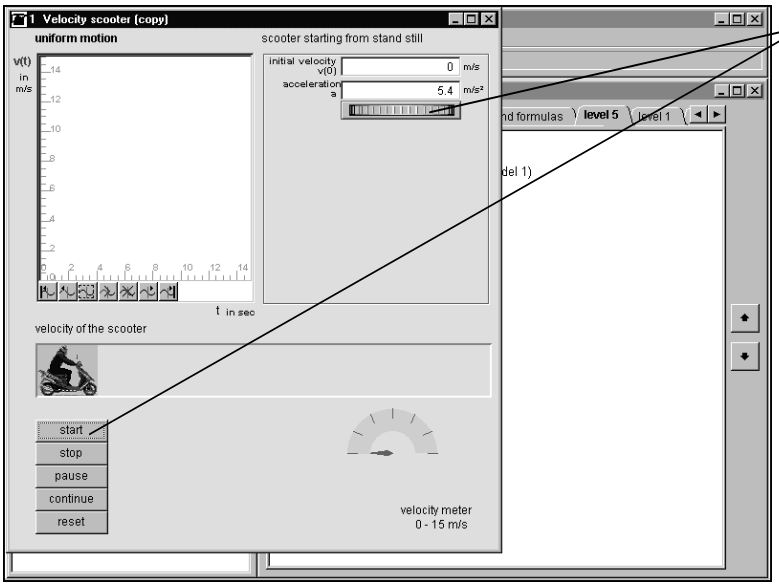
6 In the *Warning window*, check if the Previous and New variable names are the same. If this is not the case, select the appropriate New variable name(s)

7 Click **Apply**



8 In *level 5* of the *Application window*, select **1 Velocity scooter (copy)**

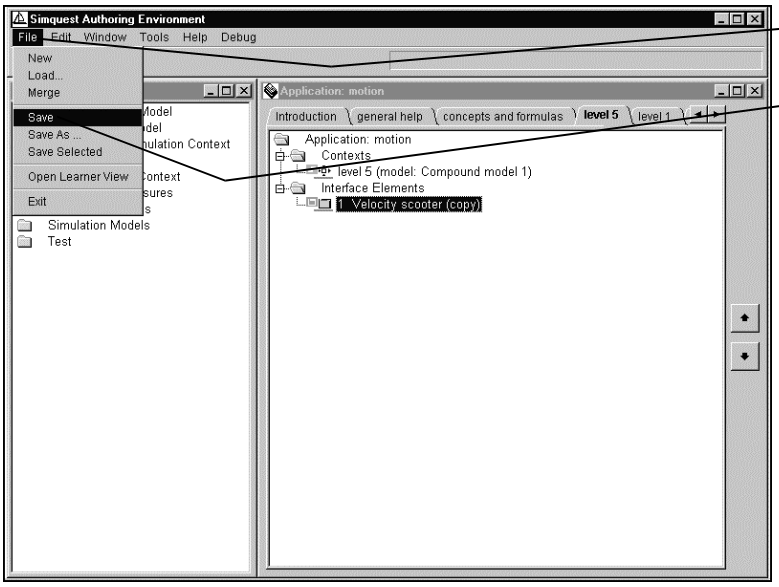
9 Click your right mouse button and choose **Run**



10 Check if the interface works properly

11 Close the interface

Saving the application Finally, you can save the application.



1 In the *SimQuest menu*, choose **File**

2 Choose **Save**

In the next paragraph, you will learn how to create a model from scratch.

Creating a new model

Creating a new model means putting the mathematical functions you want to use for your simulation into SIMQUEST. There are two ways to create a new model. This chapter teaches you both ways. In general, to create a new model, you have to carry out four general steps:

- adding a new model to your application and naming it
- deciding what method you will use to create the model
- editing the model
- saving and checking your work

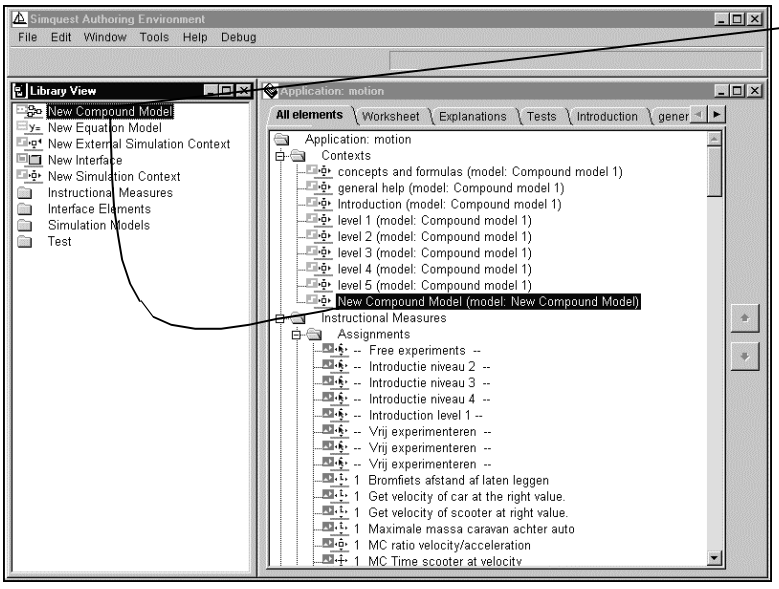
Adding a new model and naming it

Before you can start creating your model, you must:

- add an empty model to your application, and
- give it a name.

Dropping a new model in your application

The Library window contains several elements that start with the word: **New**. These elements are the starting point for creating your application. They serve as empty "shells" which you fill with other elements later.
You want to create a new model. For this you need the library element: New Compound Model.

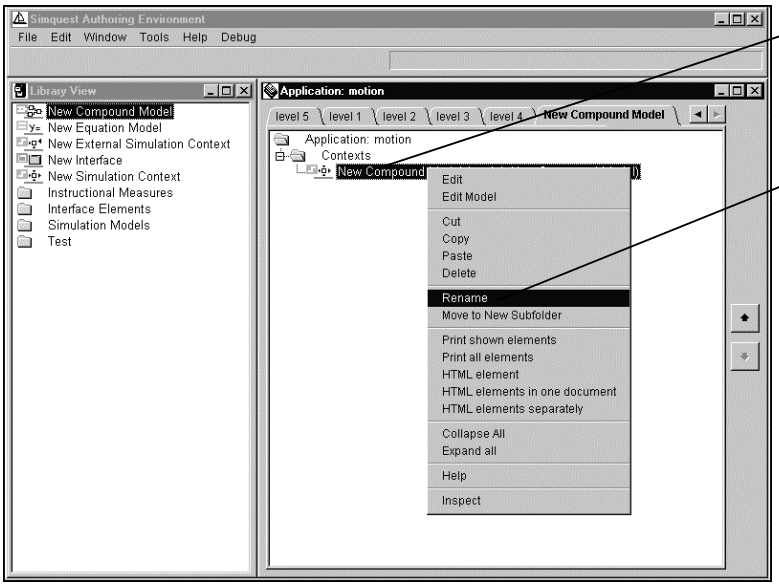


1. In the *Library window*, select **New Compound Model**
2. Drag **New Compound Model** from the *Library window* and drop it into the folder *Contexts* of the tabsheet *All elements* of the *Application window*

Check if the New Compound model is added to the All elements tabsheet. Also check if a tabsheet called New Compound model is added to your application.

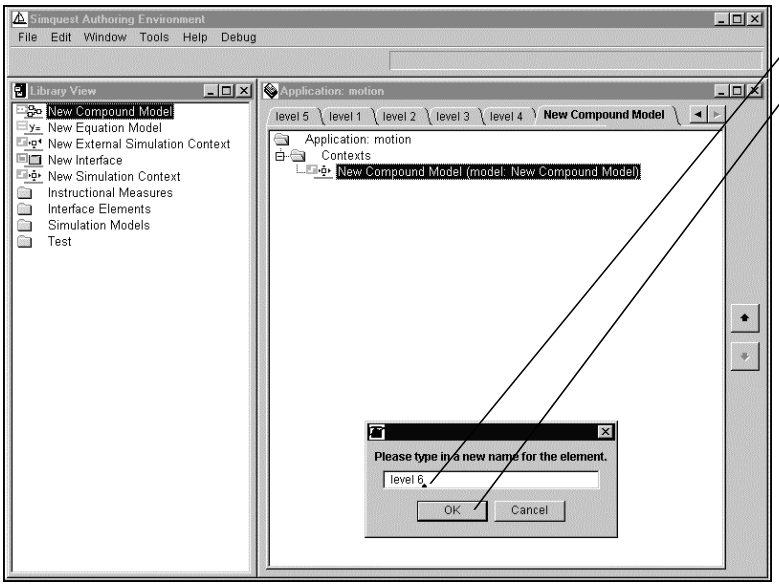
Naming your new model

A new tabsheet is added to your application called: New Compound model. To make sure that you keep a clear overview of the organisation of your application, you can (re)name the elements in your application. When you rename the Compound Model element, you also rename the tabsheet.



1 In the *New Compound Model* tabsheet of the *Application window*, select **New Compound Model**

2 Click your right mouse button and choose **Rename**



3 Type: `level 6`

4 Click **OK**

2 ways of creating a simulation model

A simulation model is the mathematical model which is used to calculate the values for your simulation. There are two different ways in which you can create this mathematical model:

- by dragging, dropping, and linking library model elements in the model editor
- by typing equations in the equation editor

You will first create the model using the model editor and after that doing the same using the equation editor.
In both cases you are going to create the model:

$$v(t) = v(0) + a*time$$

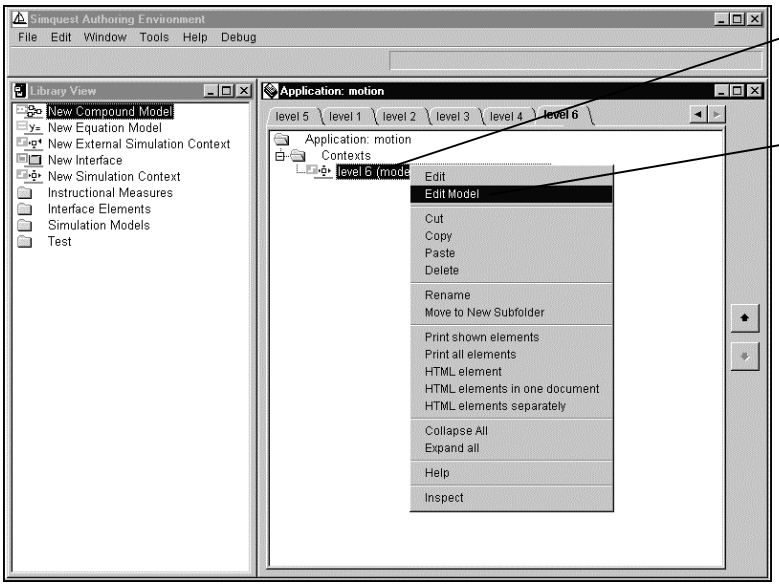
where:	*is meant:*
v(t)	speed at point in time t
v(0)	speed at starting time (point in time 0)
a	acceleration
time	time

Creating the model using the model editor

You use the model editor to combine basic model elements from the library into a larger, more complex model. Before you can add basic model elements to the model editor, you must open the editor.

Opening an empty model

You can open the model editor using your right mouse button.



1 In *level 6* of the *Application window*, select **level 6**
2 Click right and choose **Edit Model**

The model editor appears on the screen.

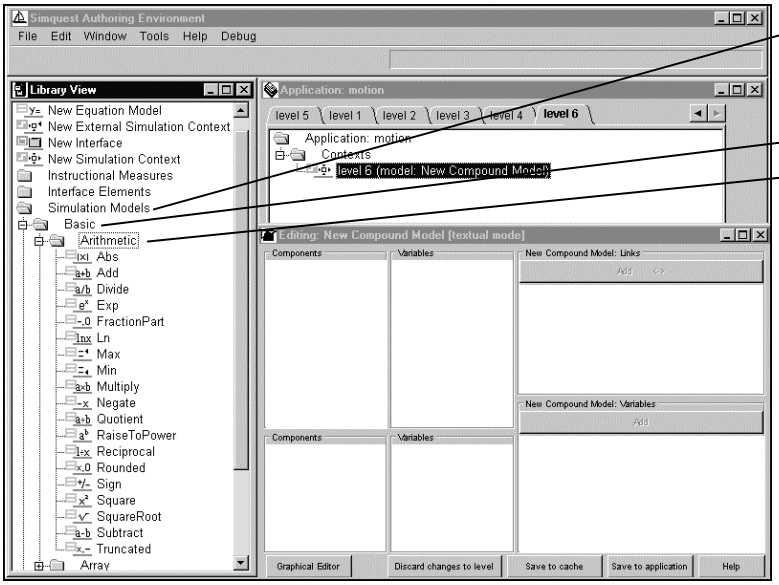
Adding model elements

In the model you are going to create, you need two basic models: Add and Multiply.

$$v(t) = v(0) + a * t$$

Add Multiply

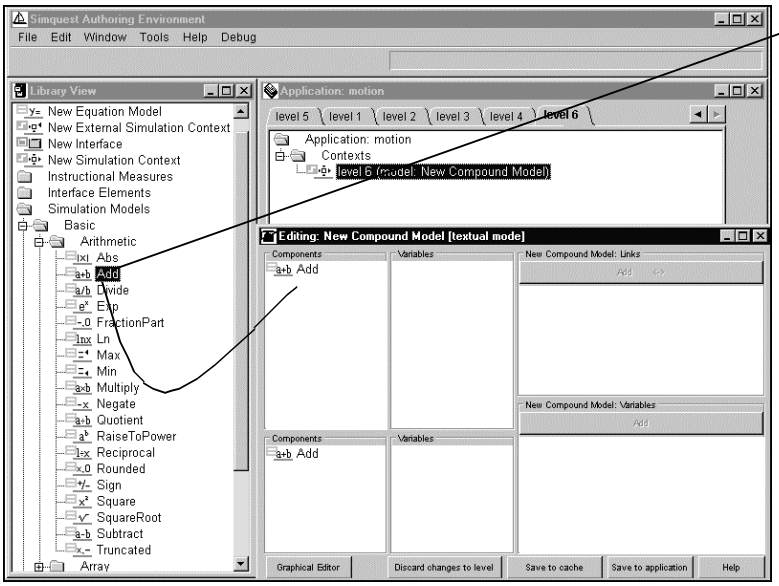
You are now going to select these two basic model elements and add them to the model editor.



1 In the *Library window*, select the folder **Simulation models**

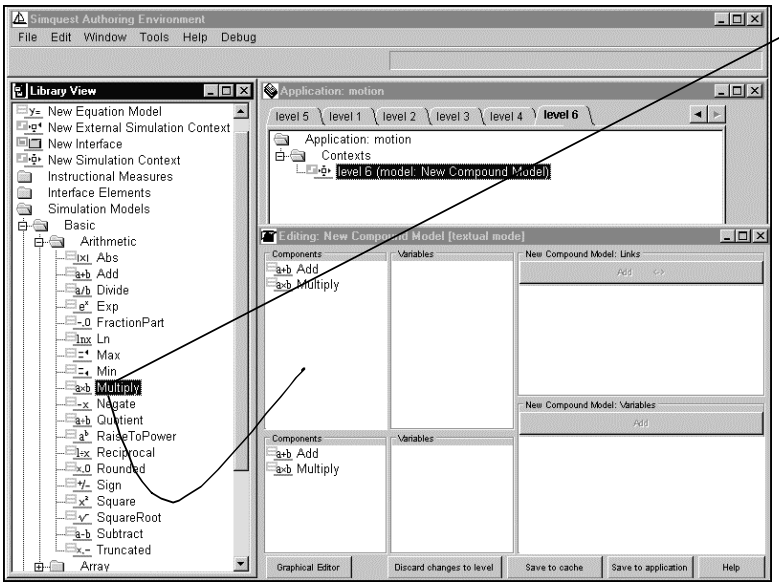
2 Select the subfolder **Basic**

3 Select the subfolder **Arithmetic**



5 Select the element **Add**

6 Drag **Add** from the *Library window* and drop it into the *Components area* of the *Model editor*



7 In the subfolder *Arithmetic* of the *Library window*, select the element **Multiply**

8 Drag **Multiply** from the *Library window* and drop it into the *Components area* of the *Model editor*

Renaming variables

When you click on one of the models in the model editor, you can see the variables the basic models contain. You can now rename these variable names into the names of your model. Look below and examine how to rename the variables.

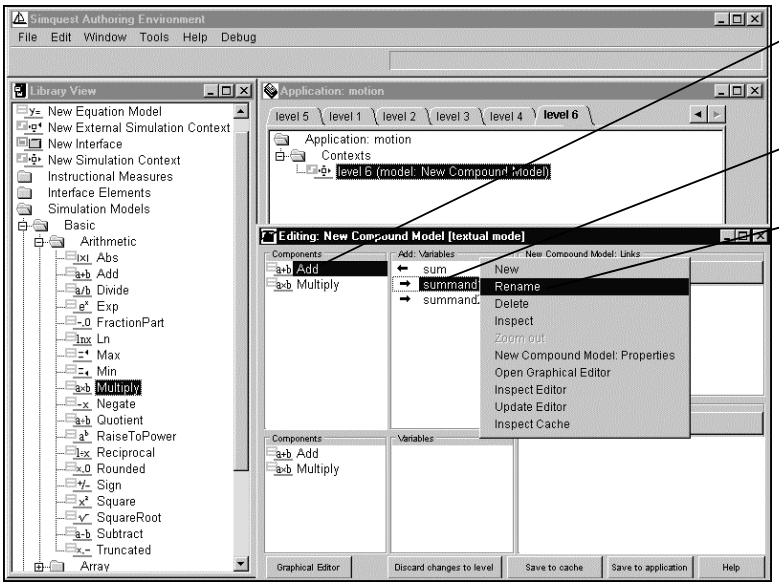
v(t) = v(0) + a * t

basic model: Multiply → multiplicant1 * multiplicant2 → product

sum = summand1 + summand2

As you can see, you have to rename 4 variables.

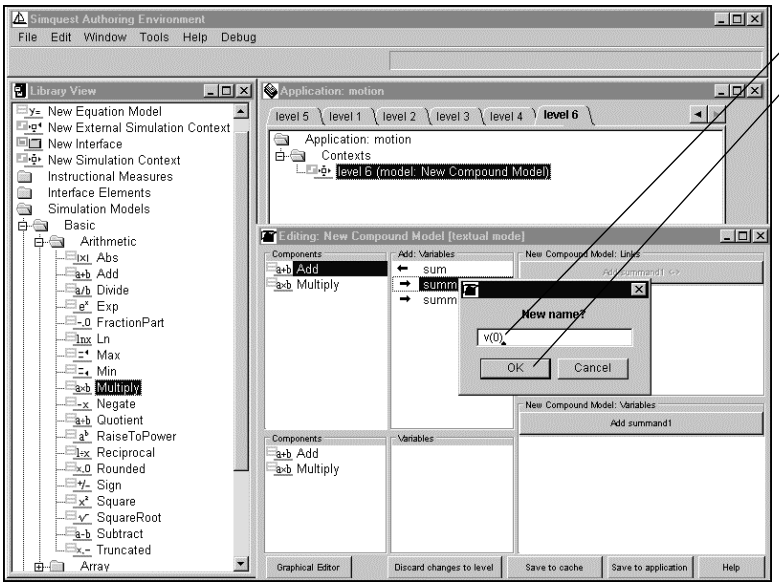
Rename:	**into:**
summand1	v(0)
sum	v(t)
multiplicant1	a
multiplicant2	t



1 In the *Components area* of the *Model editor*, select **Add**

2 In the *Variables area*, select **summand1**

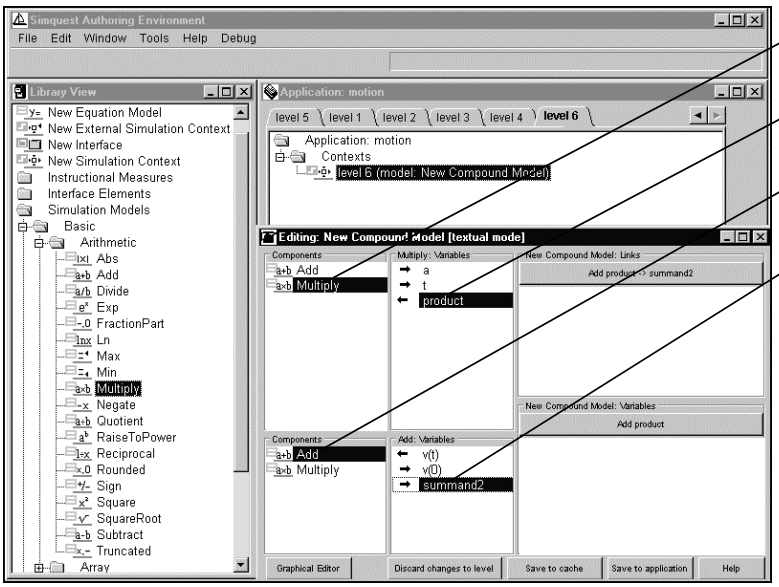
3 Click your right mouse button and choose **Rename**



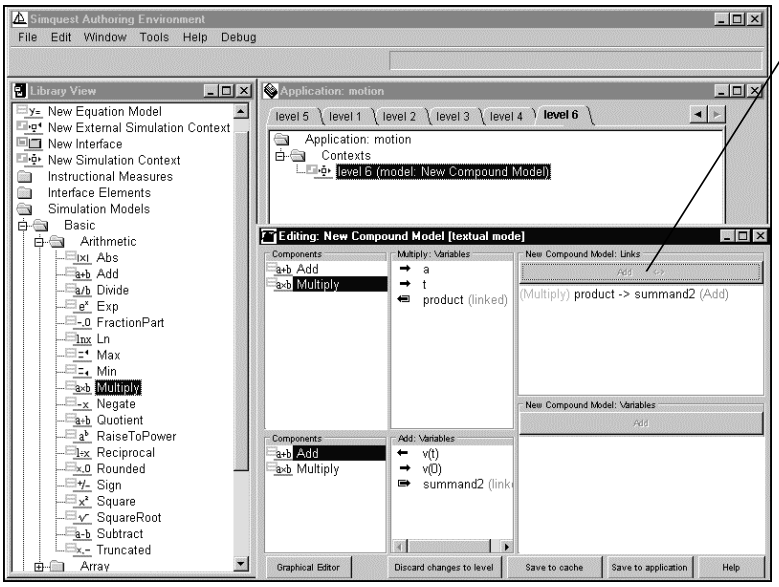
4 Type: `v(0)`
5 Click **Ok**
6 Repeat the above steps to rename: **sum** into **v(t)**, and of the basic model Multiply **multiplicant1** into **a**, and **multiplicant2** into **t**

Linking variables You also have to link the two basic models.

Link
product → summand2



1 In the *top Components area*, select **Multiply**
2 In the *top Variables area*, select **product**
3 In the *bottom Components area*, select **Add**
4 In the *bottom Variables area*, select **summand2**

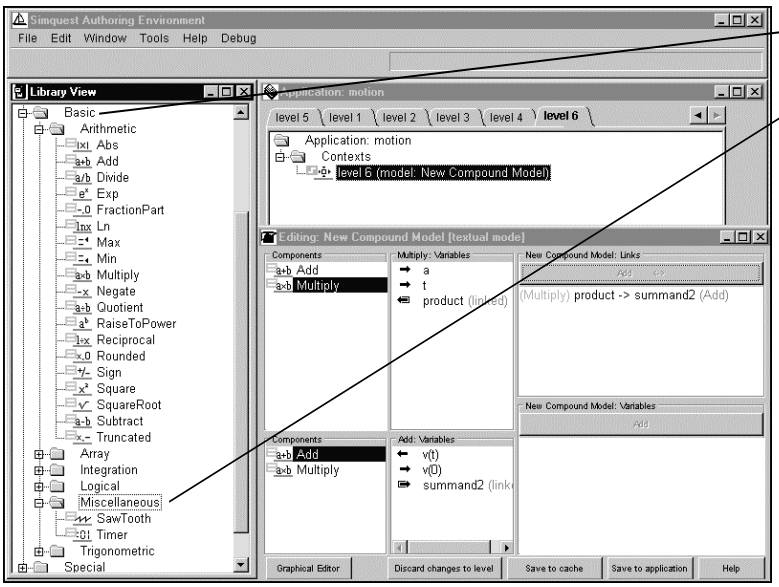


5 In the *Links area*, click the **Add product → summand2** button

Making the model dynamic

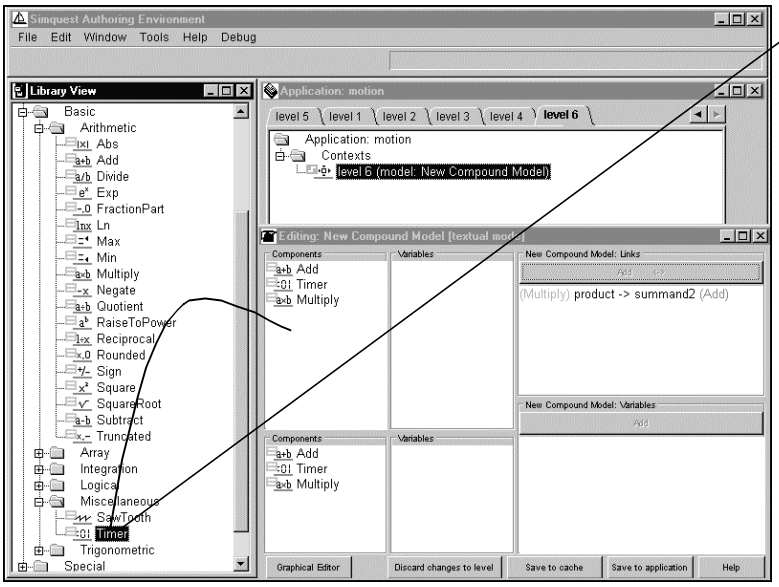
You have now created the model, but you are not finished yet. The model is now a static model. This means that it calculates only one value. What you want is to make the model dynamic. It should calculate values in the course of time. For this, you have to add the basic model Timer and link this one to the time-variable of the model. In this case:

Link
timer → t



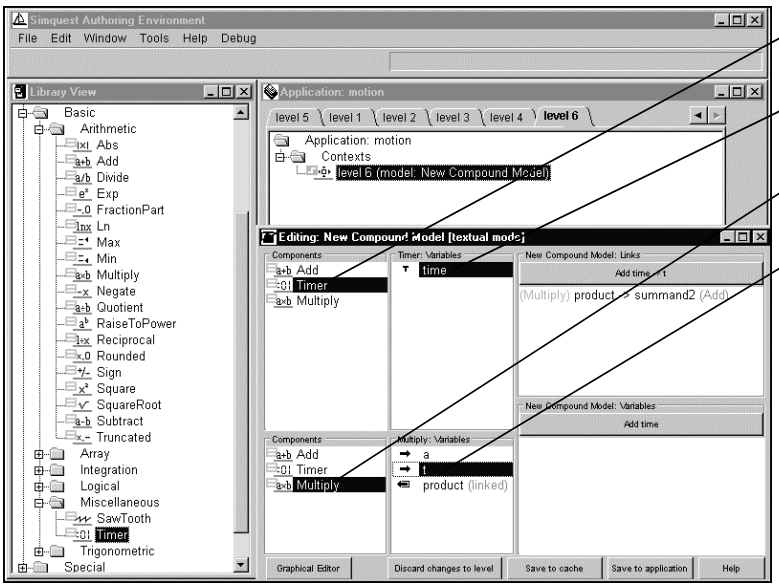
1 In the *Library window*, folder **Basic**

2 Select the subfolder **Miscellaneous**



4 Select the model element **Timer**

5 Drag **Timer** from the *Library window* and drop it into the *Components area* of the *Model editor*

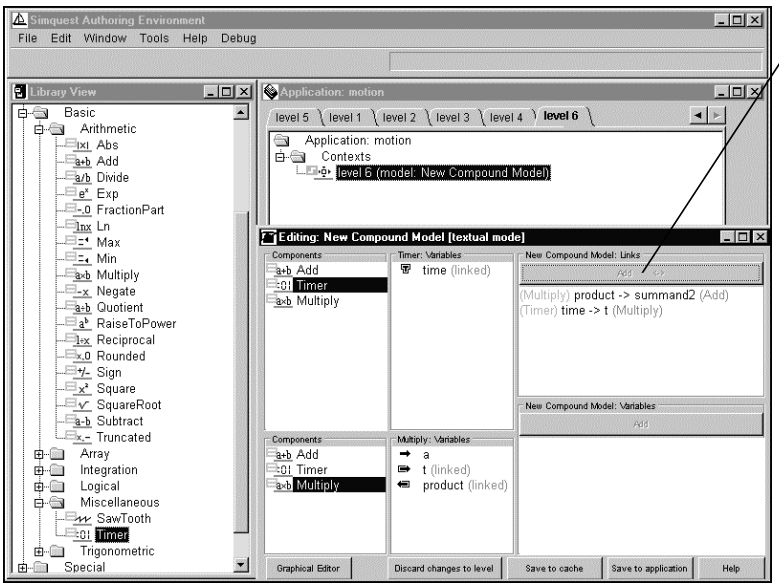


6 In the *top Components area*, select **Timer**

7 In the *top Variables area*, select **time**

8 In the *bottom Components area*, select **Multiply**

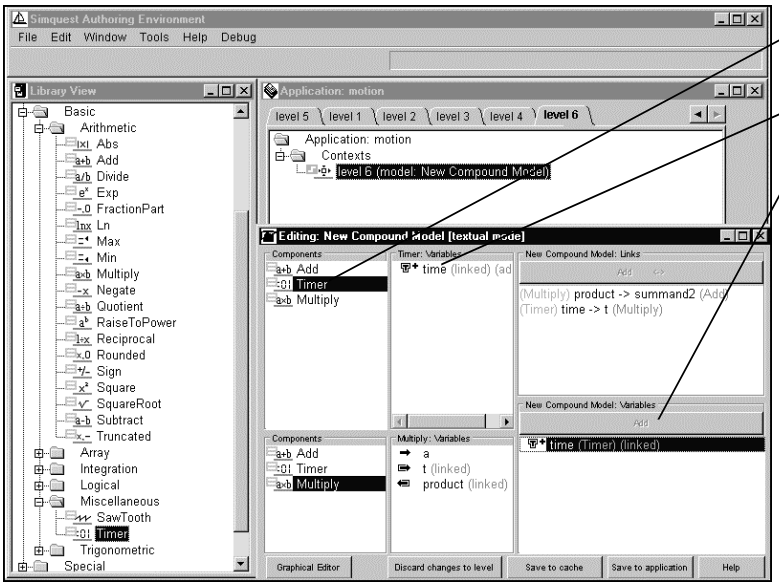
9 In the *bottom Variables area*, select **t**



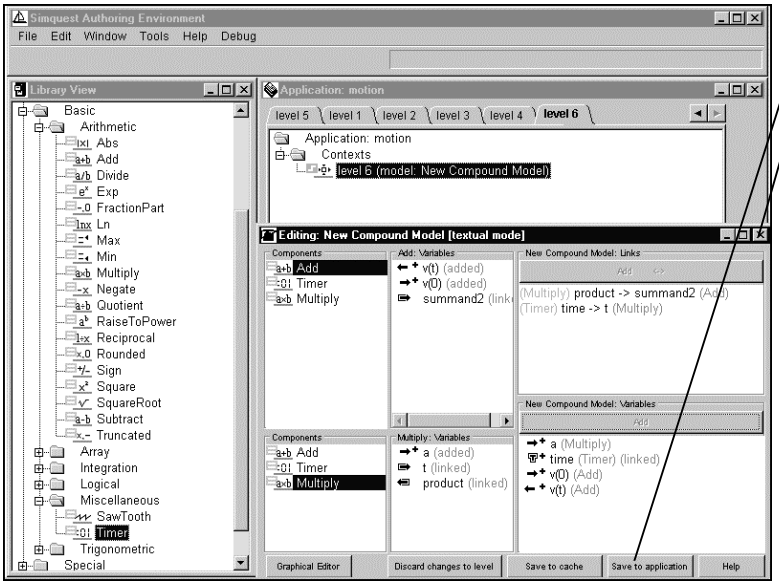
10 In the *Links area*, click the **Add time → t** button

Adding free variables

To be able to use the variables in your simulation, you have to make them available. That is, define them as free variables. In this case, the variables: time, a, v(0), and v(t), are used in the simulation and must therefore be made free variables.



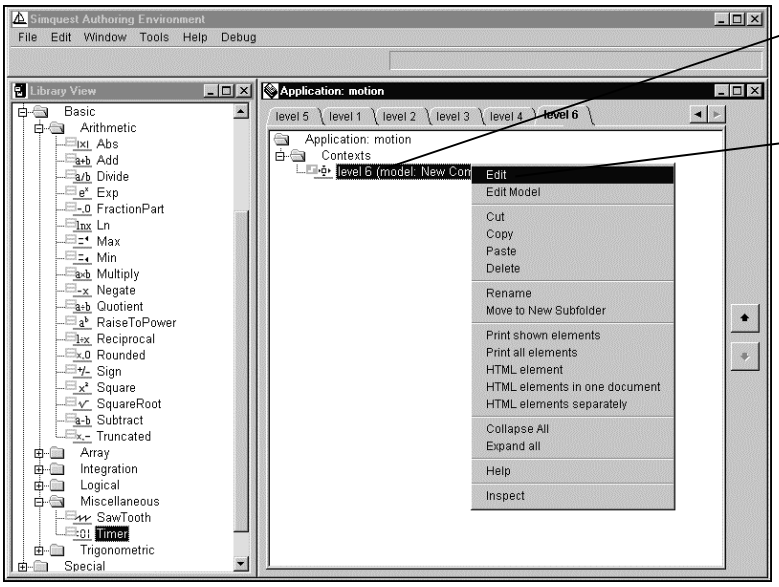
1. In the *Components area*, select **Timer**
2. Select **time**
3. Click the **Add time** button
4. Repeat from step 1 for: **a**, **v(0)**, and **v(t)**



5. Click **Save to Application**
6. Close the **Model editor**

Specifying initial values of variables

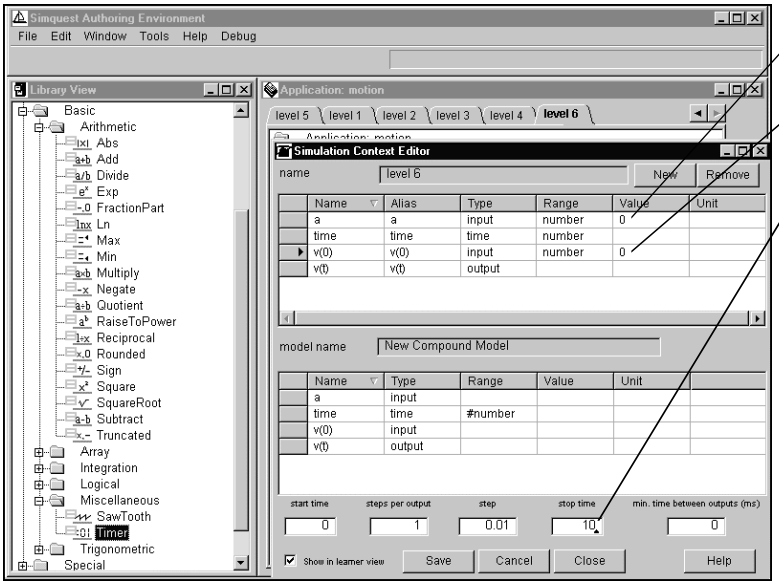
So far, you have created the model and defined the variables that you want to use in the simulation. Next, you have to specify the initial states for each of your simulation variables. You do this in the context editor.



1 In *level 6* of the *Application window*, select **level 6**

2 Click your right mouse button and choose **Edit**

Because you use the basic model elements, the variables are automatically of the right kind. In the context editor, you have to set the initial values of the variables



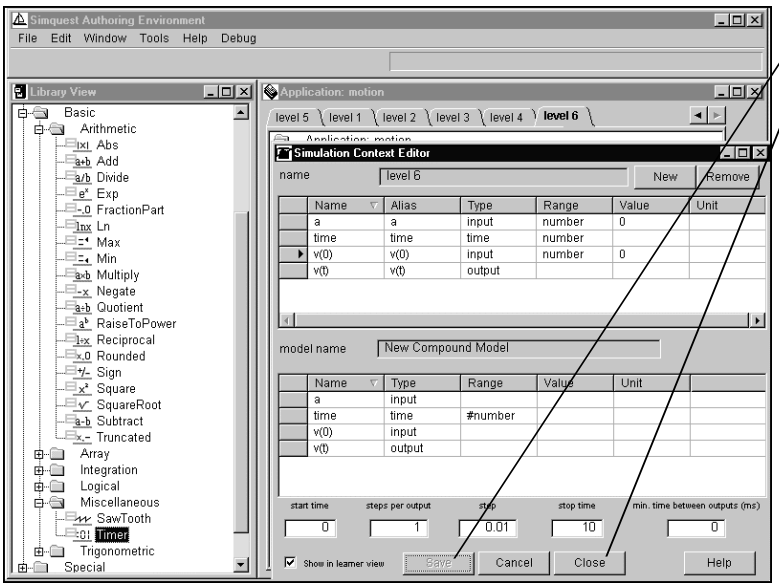
4 In the *value area* of **a**, type: `0`

5 In the *value area* of **v(0)**, type: `0`

6 In the *stop time area*, type: `10`

Saving and closing the context editor

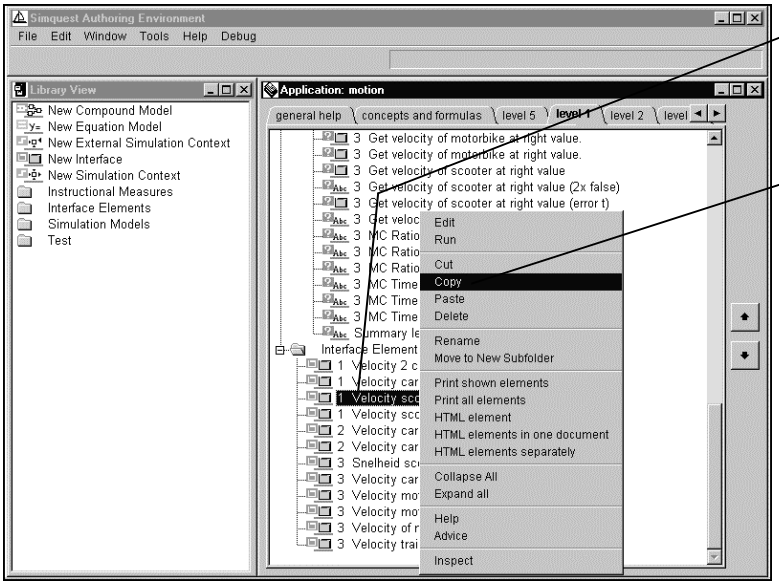
You have now created a model and specified its initial values in the context editor. Before you can check if your model works, you must save it.



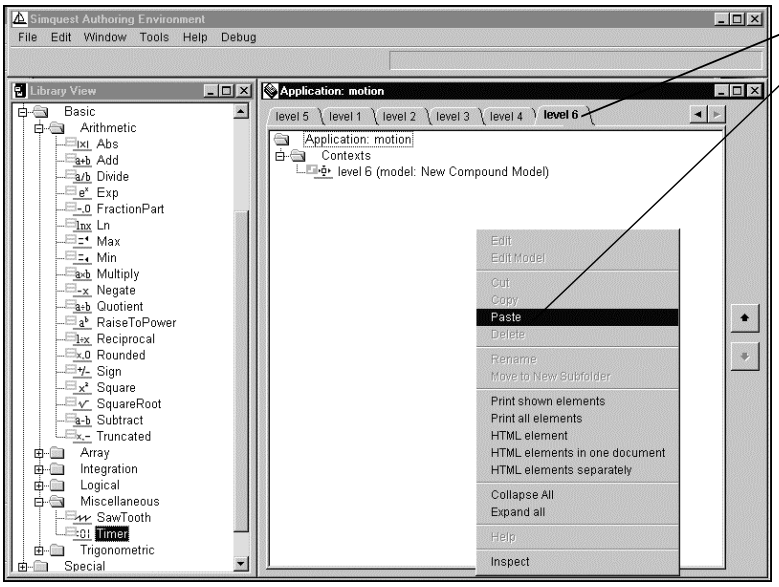
1 Click **Save**
2 Click **Close**

Checking and saving your work

To check if the model works, you can copy an existing interface from level 1, paste it in level 6, and see if you can use it properly.

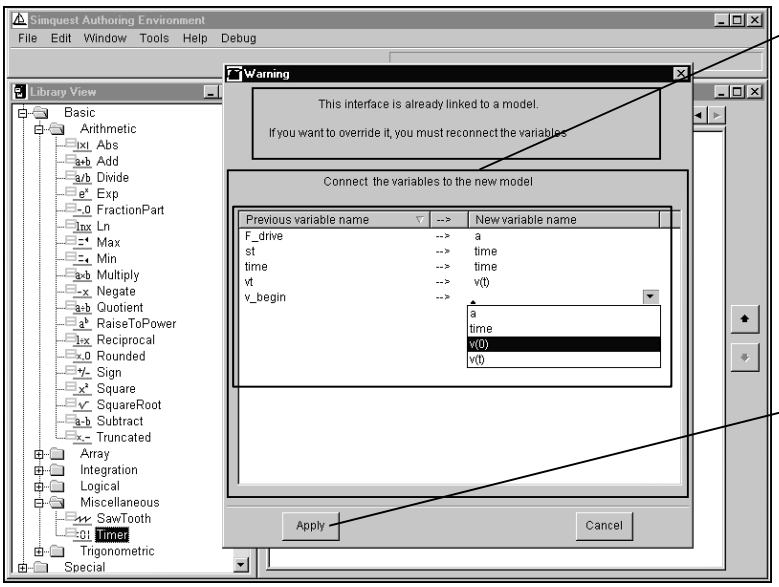


1 In the folder *Interface elements* of *level 1* of the *Application window*, select **1 Velocity scooter**
2 Click your right mouse button and choose **Copy**



3 Select the **level 6** tabsheet

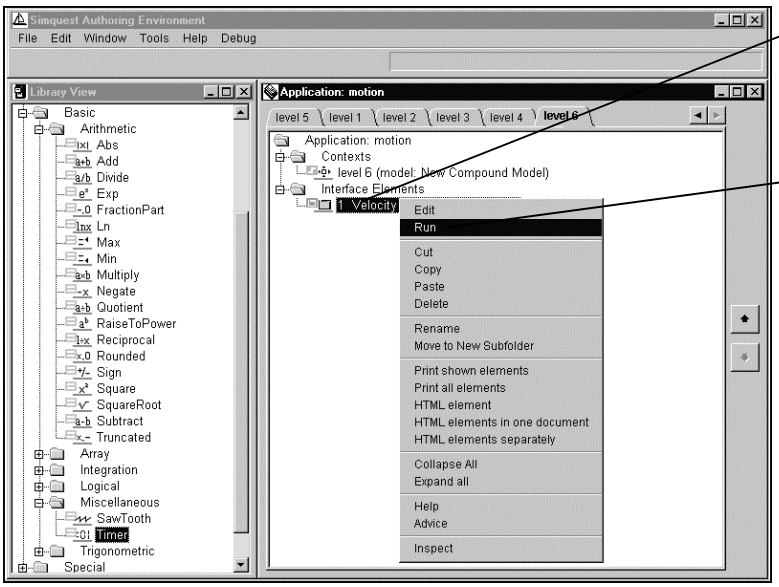
4 Click your right mouse button and choose **Paste**



5 In the *Warning window*, check if the Previous and New variable names are the same. If this is not the case, select the appropriate New variable name(s):

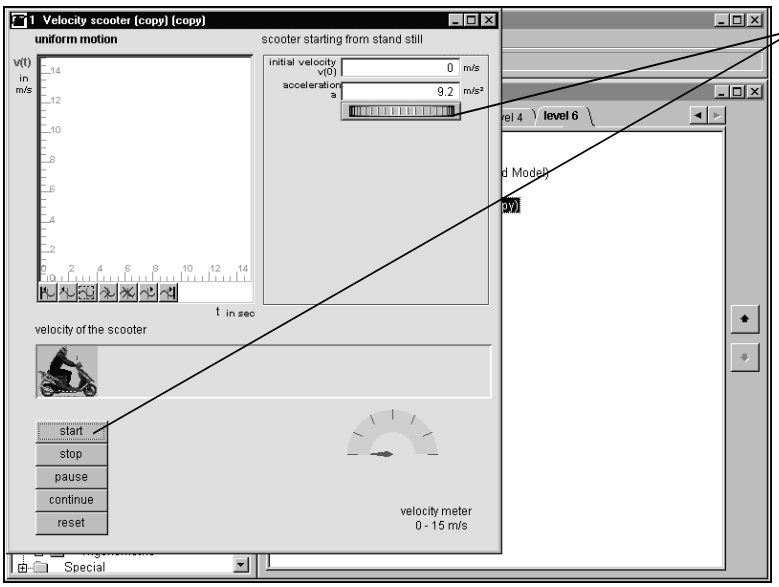
F_drive → a
st → time
time → time
vt → v(t)
v_begin → v(0)

6 Click **Apply**



7 In *level 6* of the *Application window*, select **1 velocity scooter (copy) (copy)**

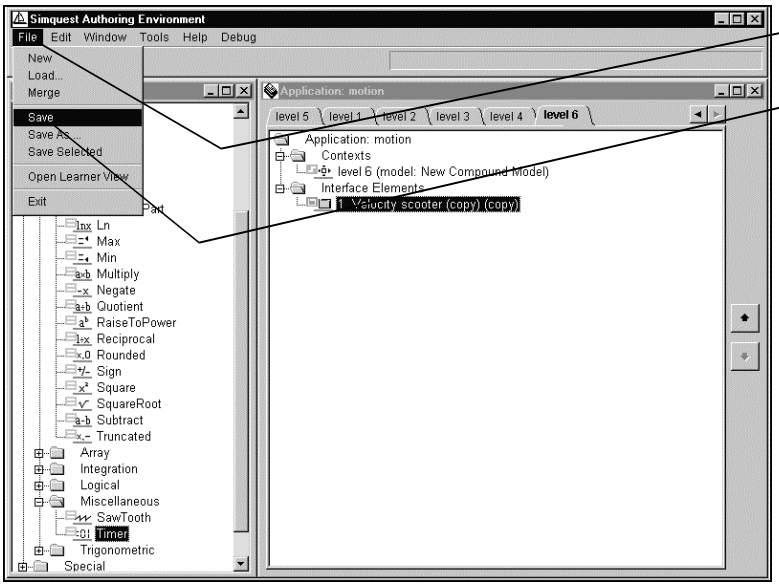
8 Click your right mouse button and choose **Run**



9 Check if the interface works properly

10 Close the interface

Saving the application Finally, you can save the application.

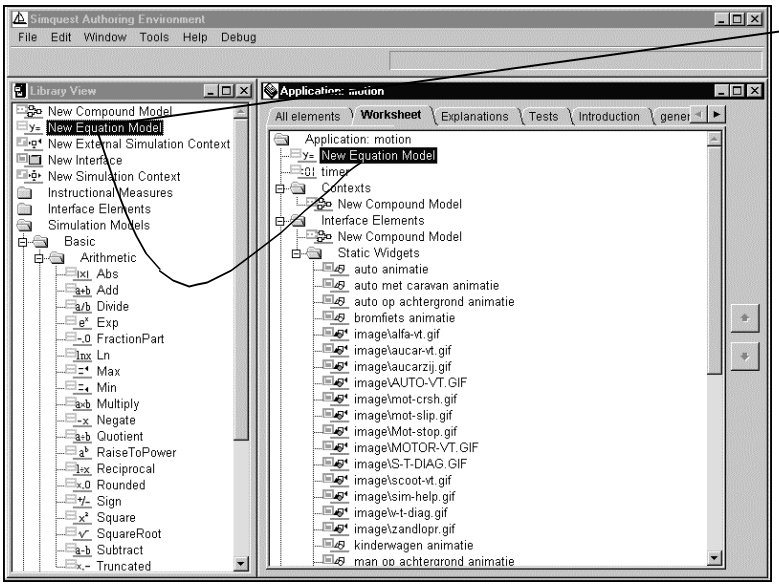


1 In the *SimQuest menu*, choose **File**

2 Choose **Save**

Creating the model with the equation editor

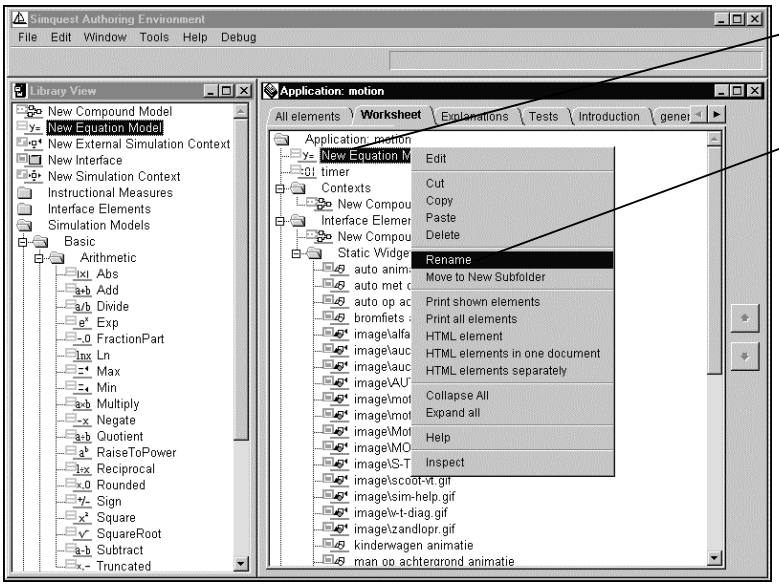
Creating a model with the equation editor works slightly different than creating it with the compound model editor. The equation model is a separate library element you first have to drag into your application and later drag into a compound model.



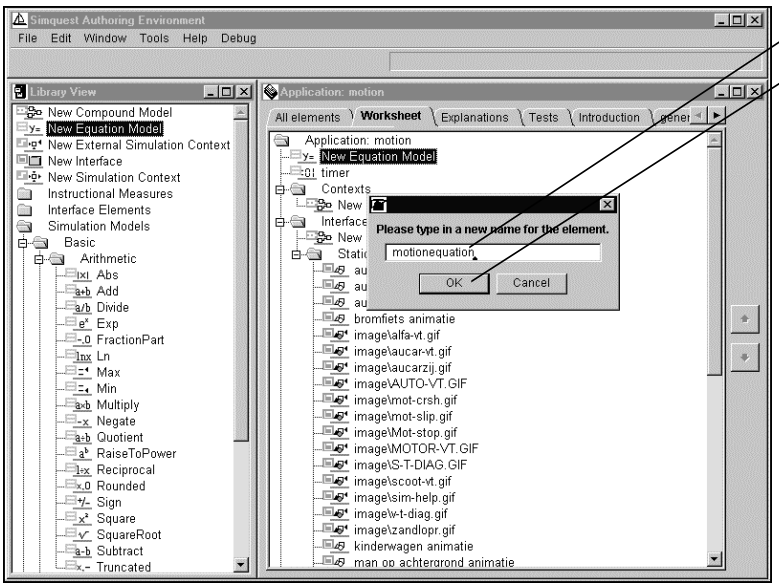
1. In the *Library window*, select **New Equation Model**
2. Drag and drop **New Equation Model** into the folder *Application: motion* of the *Worksheet* of the *Application window*

Naming your equation model

To make sure that you keep a clear overview of the organisation of your application, you can (re)name the elements in your application. In this case, you rename the New Equation Model element.

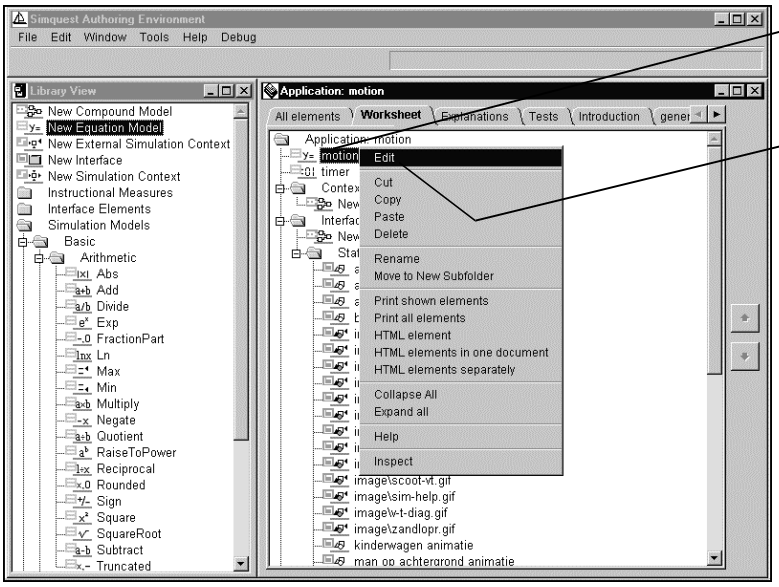


1. In the *Worksheet* of the *Application window*, select **New Equation Model**
2. Click your right mouse button and choose **Rename**



3 Type: `motionequation`
4 Click **OK**

Typing equations Next you open the equation editor. After that you can type the equations.



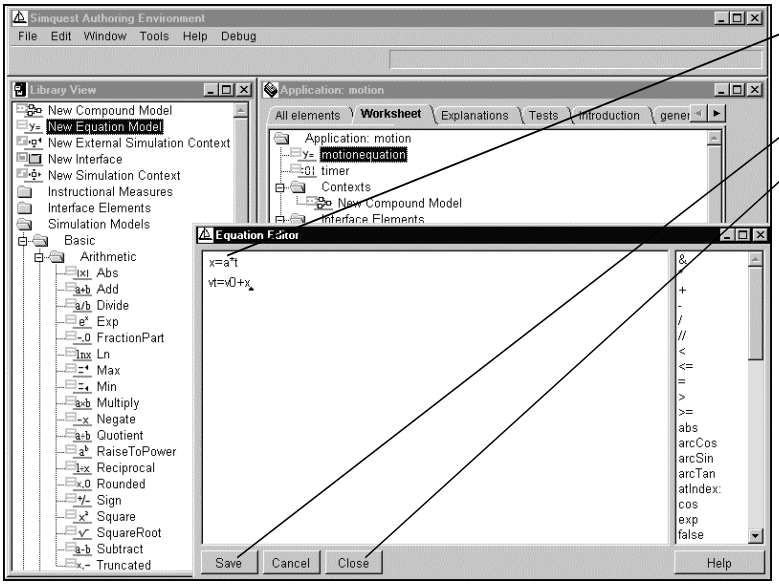
1 In the *Worksheet* of the *Application window*, select **motionequation**
2 Click your right mouse button and click **edit**

The empty equation model editor appears on the screen.
In this case you are going to create the motion model which contains the following equations:

x=a*t
vt=v0+x

Note that:

- the order in which you place the equations is important. If you put the second equation first, this model will not work.
- the sequence of a single equation must always be y=… and not reversed
- equations are calculated from right to the left, with no priorities. This means that multiplying has the same priority as adding.



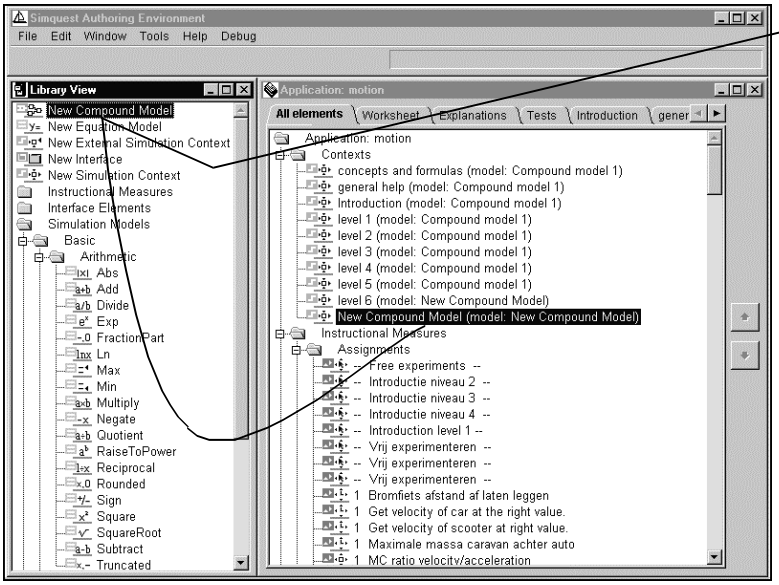
3 Type the equations:
`x=a*t`
`vt=v0+x`

4 Click **Save**

5 Click **Close**

Dropping a new model in your application

The equation you just made in the equation editor now serves as a basic model element you can use in a regular compound model editor. The remainder of this paragraph is therefore the same as creating a model with the model editor.



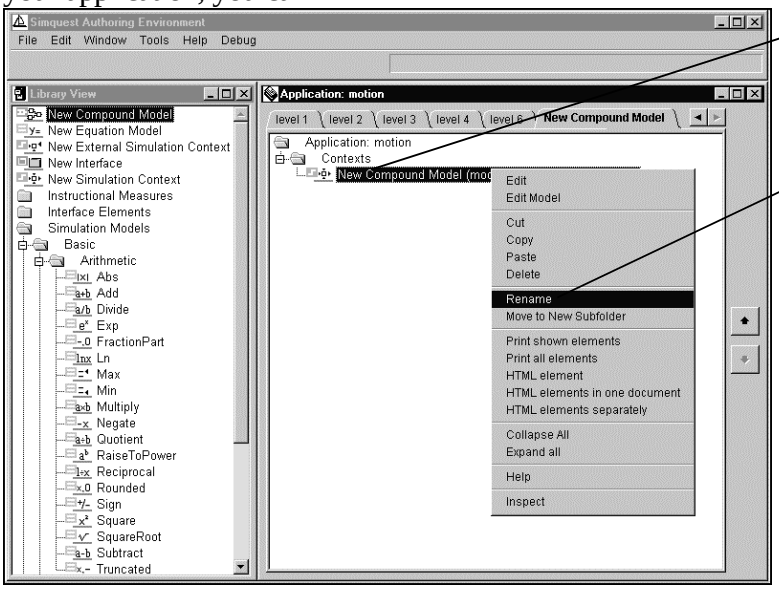
1 In the *Library window*, select **New Compound Model**

2 Drag **New Compound Model** from the *Library window* and drop it into the folder *Contexts* of the tabsheet *All elements* of the *Application window*

Naming your new model

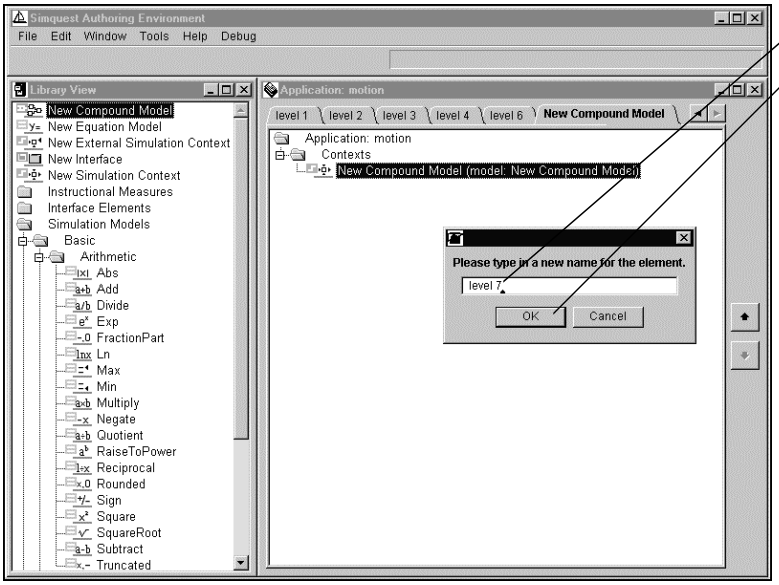
A new tabsheet is added to your application called: New

Compound model. To make sure that you keep a clear overview of the organisation of your application, you can (re)name the elements in your application. In this case, when you rename the Compound Model element, you also rename the tabsheet.



1 In the *New Compound Model* tabsheet of the *Application window*, select **New Compound Model**

2 Click your right mouse button and choose **Rename**

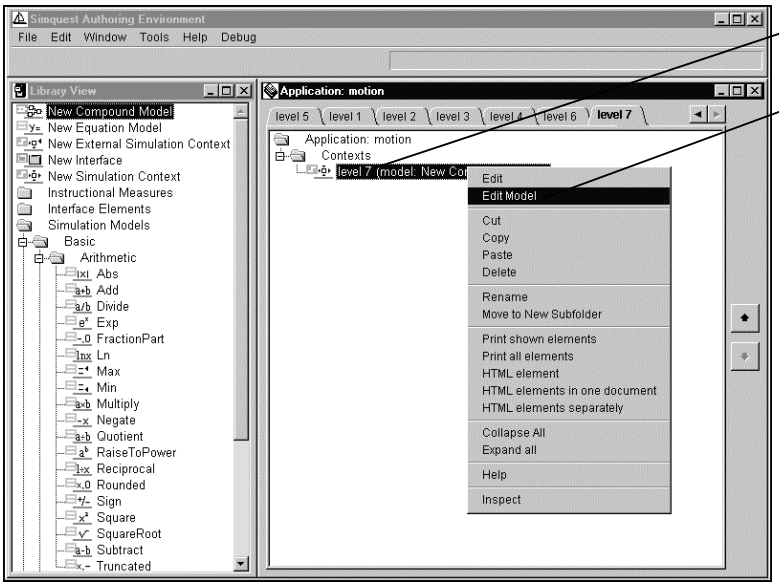


3 Type: `level 7`

4 Click **OK**

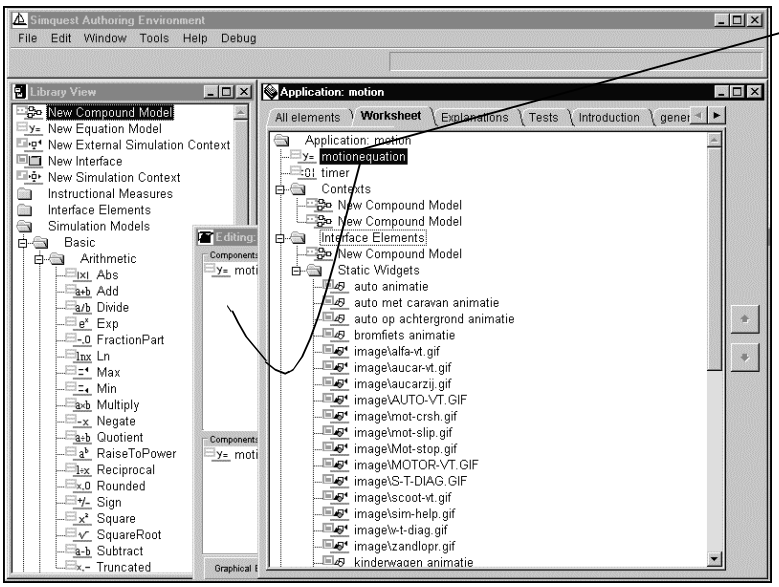
Adding the equation to a model-shell

To be able to use the equations, you have to add them to a model shell just like a regular basic block. The difference is that you do not drag and drop the Equation model from the library but from the worksheet of your Application.



1 Select **level 7** in the *Application window*

2 Click right and choose **Edit Model**



3 In the *Worksheet* of the *Application window*, select **motionequation**

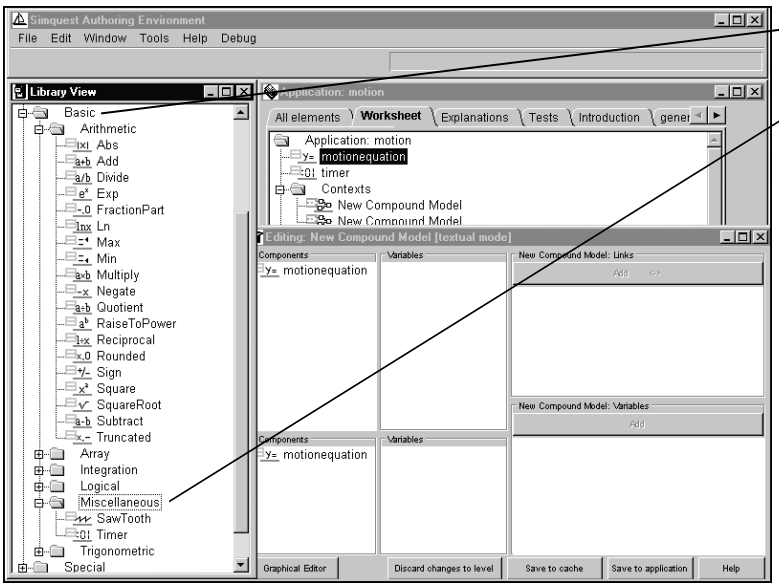
4 Drag **motionequation** from the *Worksheet* and drop it into the *components area* of the *model-editor*

The rest of the procedure is the same as you did using the basic model blocks. The difference now is that you only have to make a link to make the model dynamic. The other links are already established by your set of equations.

Making the model dynamic

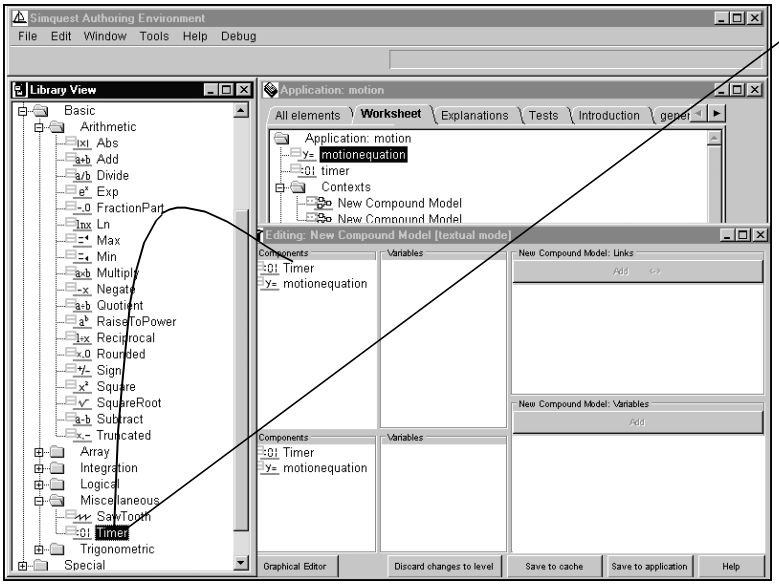
You have now created the model in a static way. This means that it calculates only one value. What you want is to make the model dynamic. It should calculate values in the course of time. For this, you have to add the basic model timer and link this one to the time-variable of the model. In this case:

Link
timer → t



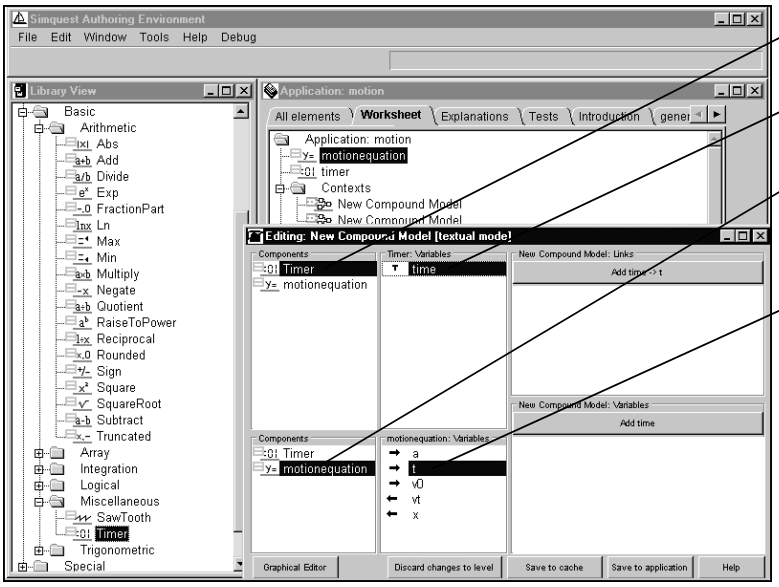
1 In the *Library window*, select the subfolder **Basic**

2 Select the subfolder **Miscellaneous**



3 Select the model element **Timer**

4 Drag **Timer** from the *Library window* and drop it into the *Components area* of the *Model editor*

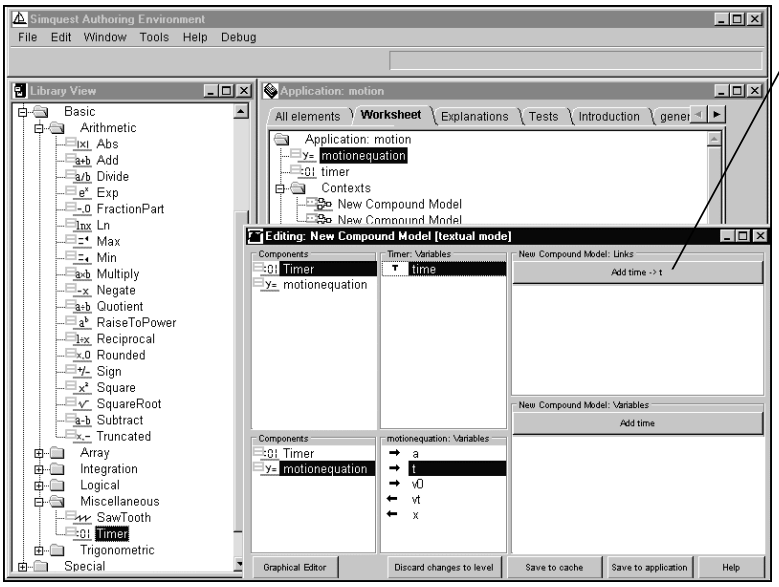


7 In the *top Components area*, select **Timer**

8 In the *top Variables area*, select **time**

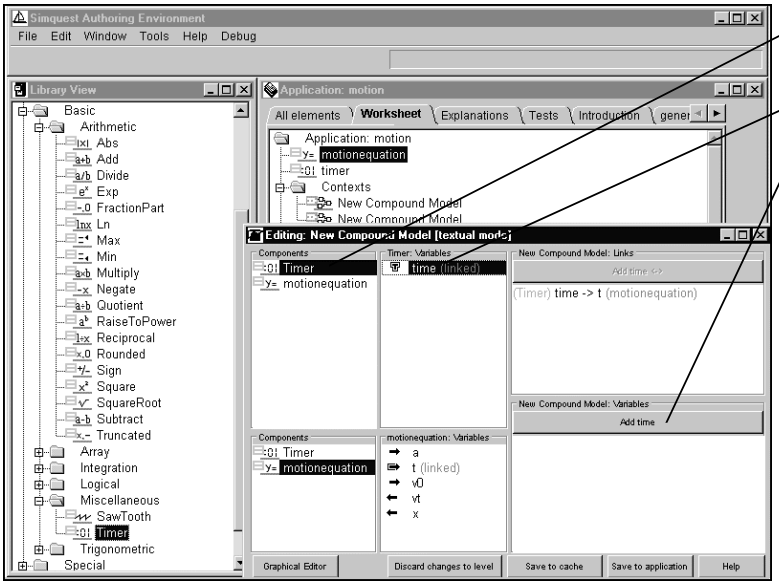
9 In the *bottom Components area*, select **motionequation**

10 In the *bottom Variables area*, select **t**

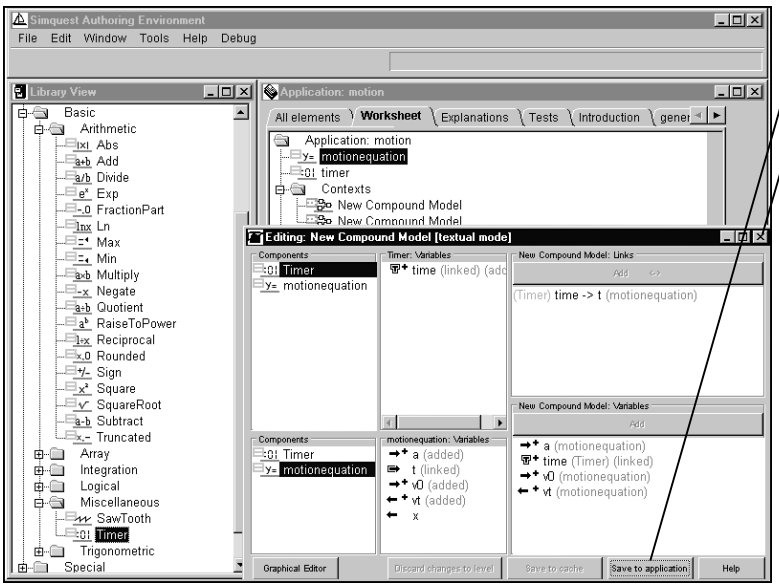


11 In the *Links area*, click the **Add time → t** button

Adding free variables To be able to use the variables in your simulation, you have to 'set them free'. In this case, the variables: time, a, v0, and vt, are used in the simulation.



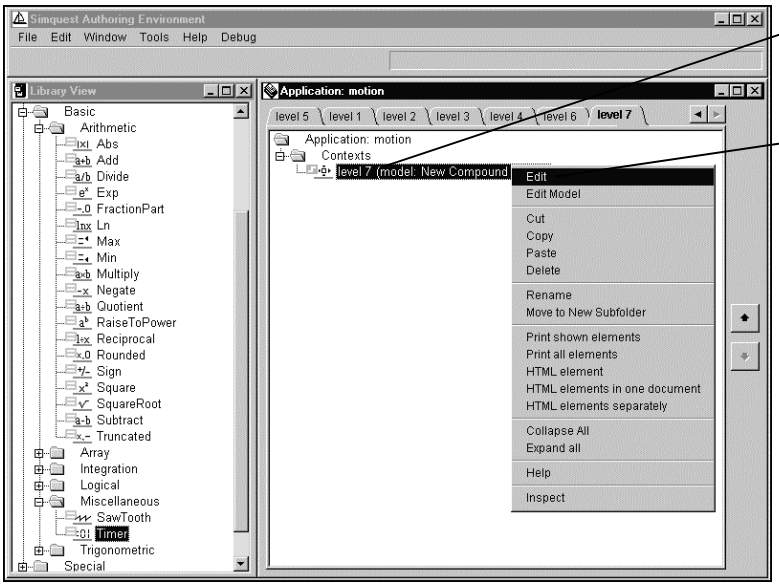
1 In the *Components area*, select **Timer**
2 Select **time**
3 Click the **Add time** button
4 Repeat for the model motionequation to add as free variables: **a**, **v0**, and **vt**



5 Click **Save to Application**
6 Close the **Model editor**

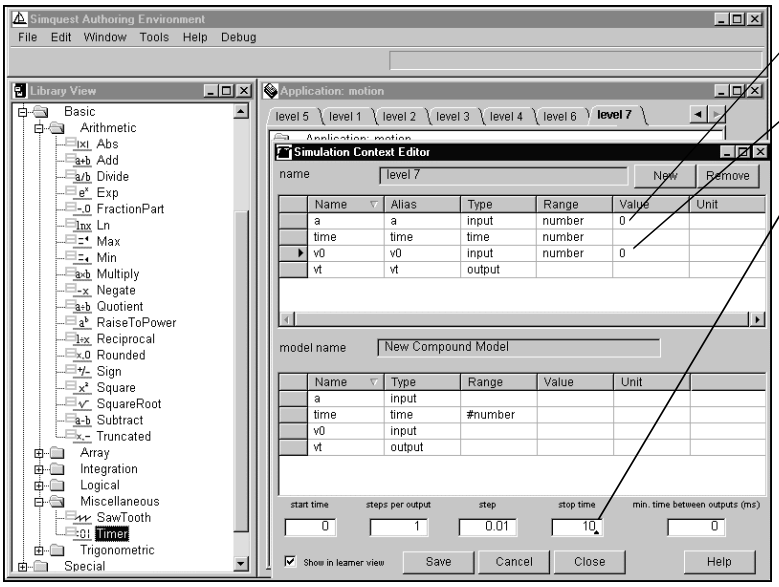
Specifying initial values of variables

Next, you have to specify the initial states for each of your simulation variables. You do this in the context editor.



1 In *level 7* of the *Application window*, select **level 7**
2 Click your right mouse button and choose **Edit**

In the context editor, you have to set the initial values of the variables.



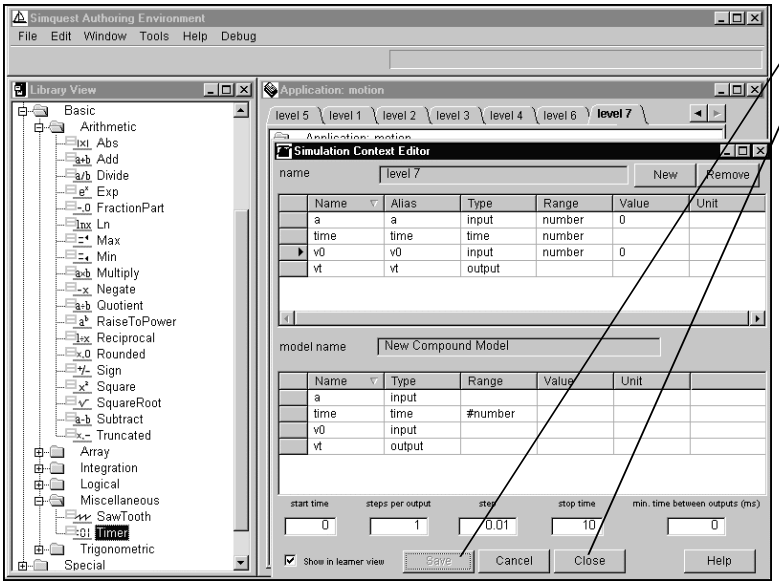
4 In the *value area* of **a**, type: `0`

5 In the *value area* of v0, type: `0`

6 In the *stop time area*, type: `10`

Saving and closing the context editor

You have now created a model and specified its initial values in the context editor. Before you can check if your model works, you must save it.

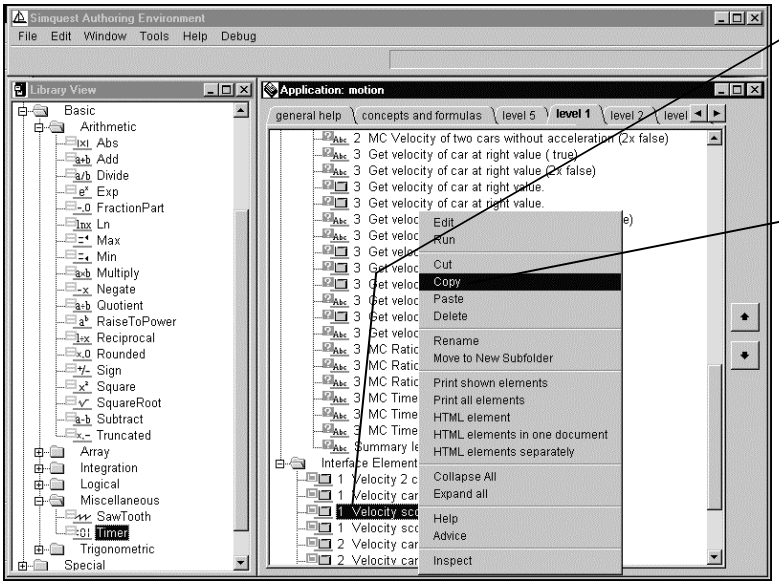


1 Click **Save**

2 Click **Close**

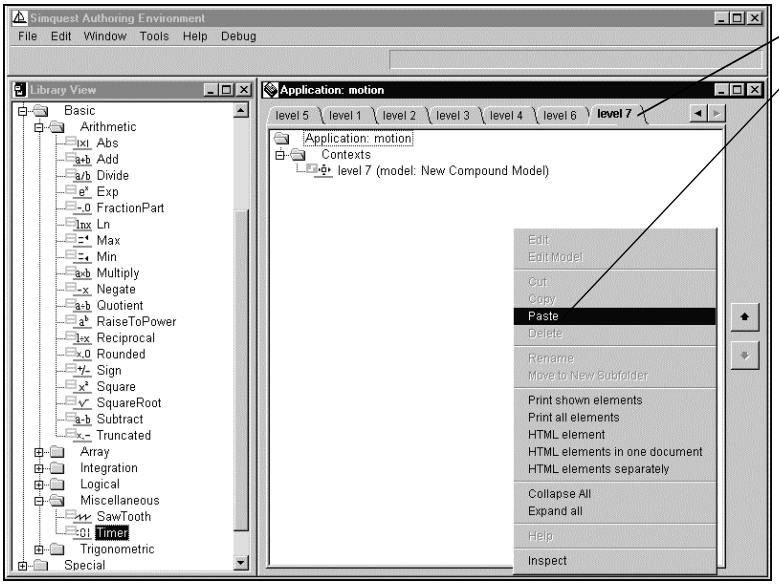
Checking and saving your work

To check if the model works, you can copy an existing interface from level 1, paste it in level 7, and see if you can still use it properly.



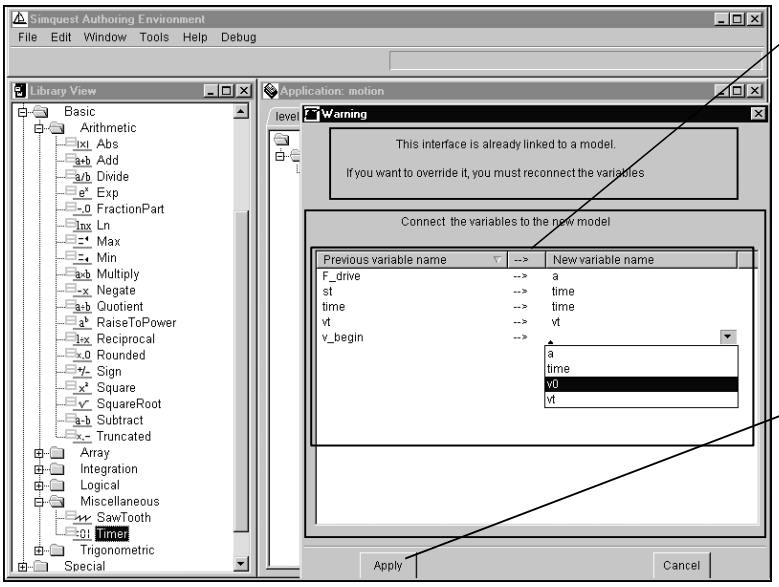
1 In the folder *Interface elements* of the tabsheet *level 1* of the *Application window*, select **1 Velocity scooter**

2 Click your right mouse button and choose **Copy**



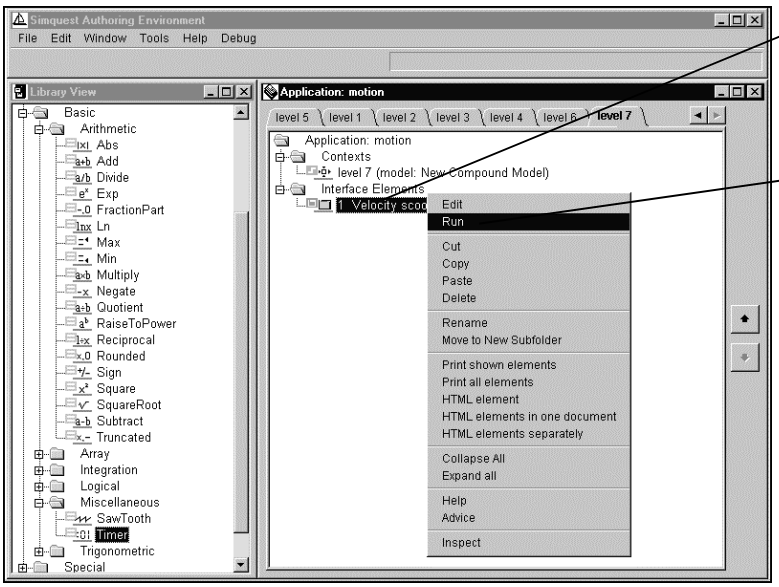
3 Select the **level 7** tabsheet

4 Click your right mouse button and choose **Paste**



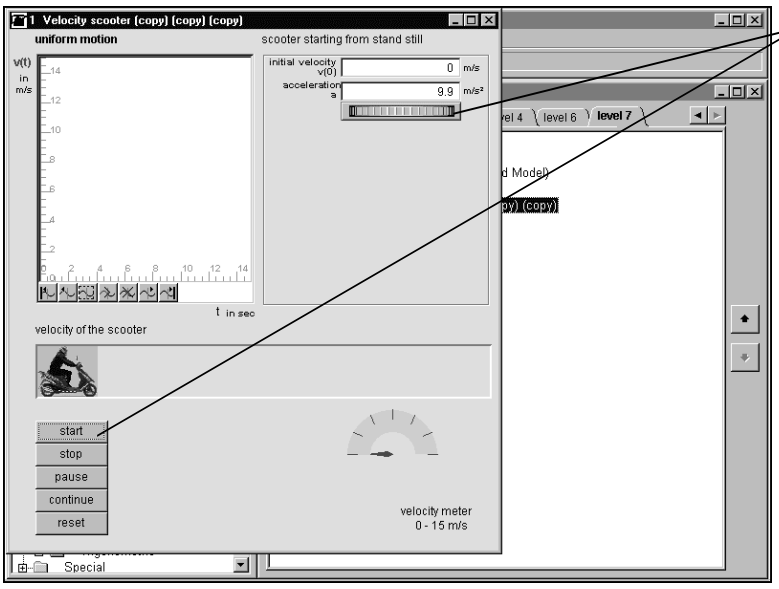
5 In the *Warning window*, check if the Previous and New variable names are the same. If this is not the case, select the appropriate New variable name(s):
F_drive → a
st → time
time → time
vt → vt
v_begin → v0

6 Click **Apply**



7 In *level 7* of the *Application window*, select **1 Velocity scooter (copy) (copy) (copy)**

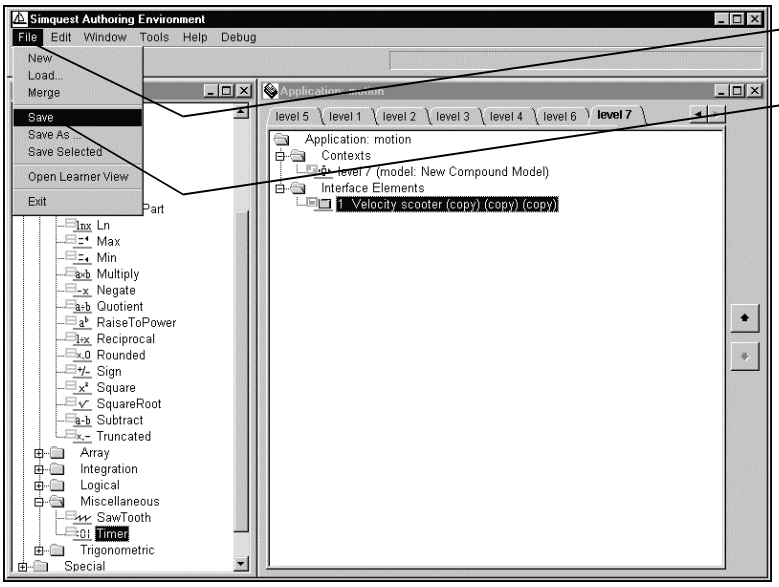
8 Click your right mouse button and choose **Run**



9 Check if the interface works properly

10 Close the interface

Saving your application Finally, you should save your application.



1 In the *SimQuest menu* choose **File**

2 Choose **Save**

Summary

You have now modified and created models.

You *modified* a model by copying and renaming an existing model.

You *created* a model by dragging an empty model into your application, dropping model elements into it, and specifying the properties of the model elements.

You also used equations to create a new model.

6 Models