
MODELING OF LOGISTICS PROCESSES

ACTIVITY #5

Triggers – introduction

dr hab. Daniel Kaszubowski, prof. PG

Department of Transportation Engineering



Co-funded by
the European Union

Co-funded by the European Union. Views and opinions expressed are however those of the author or authors only and do not necessarily reflect those of the European Union or the Foundation for the Development of the Education System. Neither the European Union nor the entity providing the grant can be held responsible for them.



1. Objective and new skills

The aim of the task is to organize the basic concepts used in the program, concerning simulation events and the logic of using object states. Introducing the concept of an object state allows for a detailed discussion of the assumptions of using triggers (*Triggers*), which allow for invoking specific actions or processes.

New skills
The concept of a simulation event and state
<i>Principle of using triggers</i>
The ability to indicate where in the process selected actions can be initiated
Knowledge of basic actions that can be initiated in selected object states

2. Basic concepts

2.1. Simulation events

An example of real events would be:

- placing the pallet on the rack,
- completion of processing,
- issue of goods from the warehouse.

FlexSim is a program for discrete event simulation (*Discrete Event Simulation*). Each event starts at a specific moment and lasts for a specific time. The modeled system is therefore represented as a sequence of consecutive events. However, the start moments of these events are the only situations in which the state of the object can change.

2.2. States

Each object in the modeled system is characterized by the current value **(state) assigned to it**. A machine processing products will have the *Processing state*. Each object can be in various states characteristic of itself, but only in one at a time.

3. Triggers

3.1. The idea of using

A trigger is a logic of action (action, operation) that is started when a specific state of an object (event) occurs. Such an action can be, for example, changing the color of a flow element, changing the value of a label or adding a row to a table. It can also be sending a message to another object after a certain action is completed. Each object has its own unique set of triggers. For example, in the *Processor object* in the *Process phase Item* there will be two triggers available, activated in different places (Fig. 1). The triggers available for the Source are shown in Fig. 2. The most important of them will be discussed in the following points.

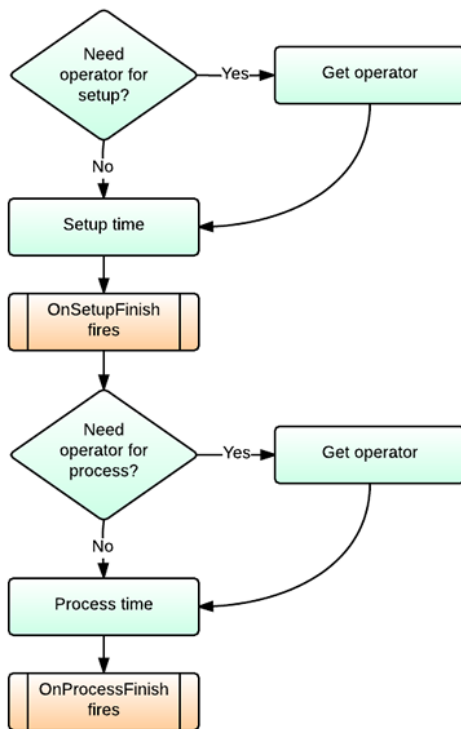


Fig. 1. Examples of triggers

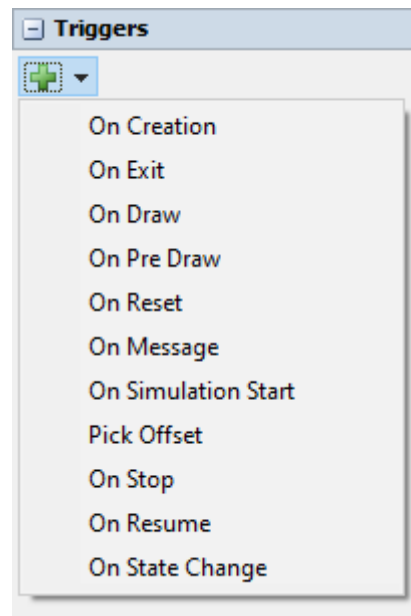


Fig. 2 Triggers for Source

3.2. Generating Flow Elements – On Creation

On Creation activates the specified operation at the moment when the source generates a flow element. The categories of actions shown in Fig. 3 will be discussed in detail later in the material.

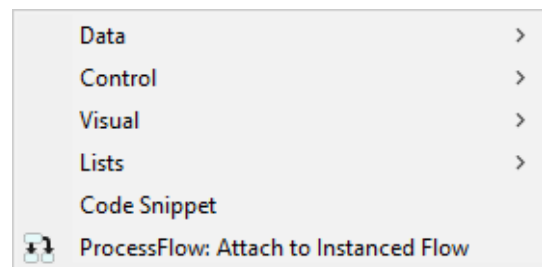


Fig. 3 On Creation Section

3.3. Leaving the facility – *On Exit*

The specified operation will be triggered when the flow element leaves the object. The same options as in *On Creation* are available .

3.4. Communication between objects – *On Message*

On Message runs the selected action whenever the object receives a message . The available actions are shown in Figure 4.

Each sent message has a certain set of parameters (group "*Control*") . The object receiving the message has access to them and can use them.

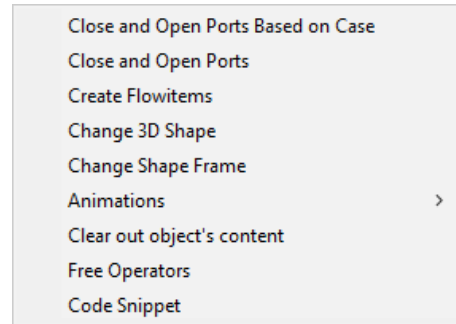


Fig. 4. Options for *On Message*

Sample settings:

1. **Close and Open Ports Based on Case** – based on the defined value of the given expression (*Case Value* field), the appropriate action is performed. The settings in Fig. 5 mean that the output port of the current object `current.output.close()` will be closed if the object receives a message in which the value of the first parameter `msgparam(1)` will be equal to 1.

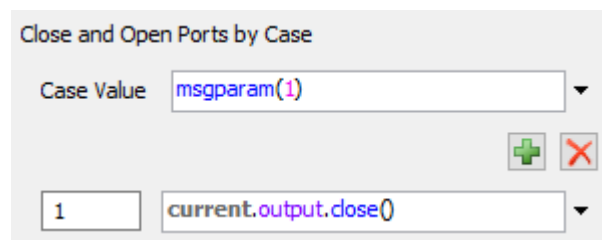


Fig. 5 *Close and Open Ports* Settings

2. **Create Flowitems** – allows you to generate a flow item. The number of items to create is given in the *Quantity to Create* field . The type of the flow item (*Flowitem Bin Rank*) is given as the next numeric value in the list of available items in *the Flowitem Bin* section. The place where the objects appear is determined by *Destination*. The values of these parameters must be related to the parameters of the received message (Fig. 6)

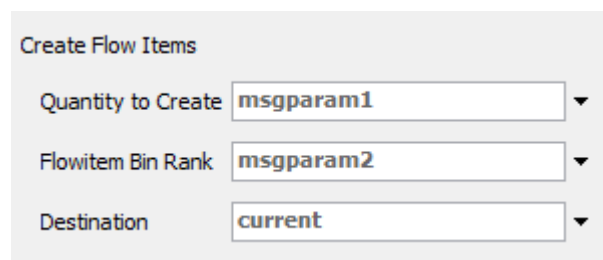


Fig. 6. *Create Flowitems* settings

3. **Clear out objects content** – removes all elements that are on the object at the time the message is received (Fig. 7).

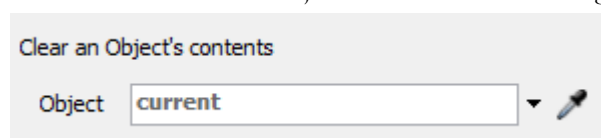


Fig. 7. *Clear out objects content* option

3.5. Common triggers

Many objects have very similar triggers, especially if they belong to the same category. Therefore, in addition to the triggers already mentioned, it is common to use:

1. **On Entry** – allows to execute the given logic when the element enters the given object. The command is not available for the source, because it is the beginning of the system.
2. **On Setup Finish** – triggers a specific action after the object is re-armed. Occurs on *Processor*, *Combiner* and *Separator*.
3. **On Process Finish** - calls action when the processing time of an element has expired. Occurs on *Processor*, *Combiner* and *Separator*.

4. Action categories available for invocation

4.1. Group Date

It is possible to parameterize flow elements and/or model objects. The most commonly used options are:

1. **Set Label** – allows you to change the value of an existing label or add a new label with a given value. In *the Object* field, you specify the object to which the label is assigned, in the *Label* field you define the name of the label and in the *Value* field its value (Fig. 8).

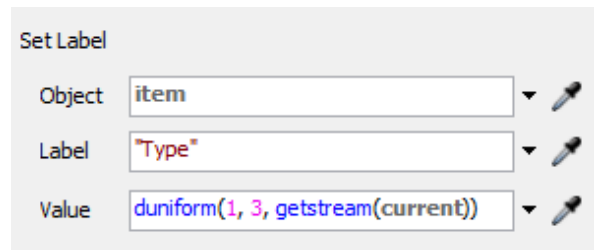


Fig. 8 *Set Label* option

2. **Set Label and Color** – works identically to *Set Label*, but a color is automatically added to the selected flow element. It is set based on the numeric value of the label, to which the color number from the RGB palette is selected.
3. **Set Label by Percentage** – allows you to change the label value based on the percentage distribution. Use the green button to add more cases (Fig. 9).

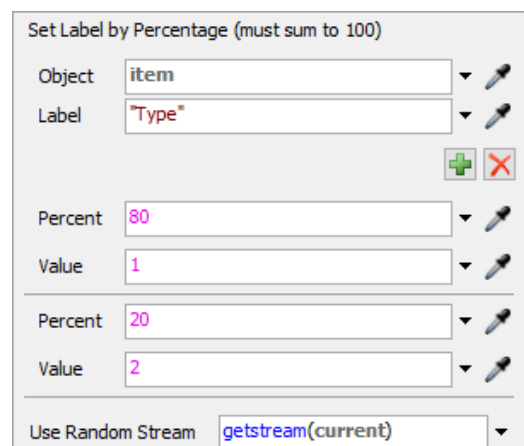


Fig. 9 Option *Set Label by Percentage*

4. **Write to Global Table** – writes the specified value (*Value field*) to the selected global table (*Table field*). In the *Row fields* and *Column* – specify the row and column number in which the saved value will be located (Fig. 10).

Fig. 10 Option *Write to Global Table*

5. **Add Row and Data to Global Table** – allows you to add another row to the given global table (*Table field*) and fill the next cells with appropriate values according to the established column layout. The green cross adds additional fields referring to columns in the target table, in which the method of entering values is defined (Fig. 11).

Fig. 11 Add Row and Data to Global Table option

6. **Increment Value** – allows you to change the value of the expression given in the *Increment field* by the value given in the *by field* (Fig. 12).

Fig. 12. Increment Value option

4.2. Control Group

The commands in this group allow you to control objects. The most commonly used are:

1. **Send Message** – allows you to send a message with specific parameters from a given object. You define the object that sends the message (*From*) and the object that receives it (*To*). In the *Condition field*, you can specify the condition that must be met for the message to be sent, while in *Delay Time* the delay time for sending the message. The message parameter is given in the *Parameter 1-3 fields* (Fig. 13).

2. **Close and Open Ports** – allows you to close or open the input or output ports of a given object.

The action to be performed is selected from the *Action field*. The target object is indicated in the *Object field*, while the condition to be met is indicated in the *Condition field* (fig. 14).

Fig . 14 *Close and Open Ports*

4.3. Visual Group


Visual collects actions related to graphical settings of both flow elements and model objects. The most commonly used options are:

1. **Set Location** – allows you to change the position of the object indicated in the *Object field*. The position is specified in the XYZ reference system (Fig. 15). The XY coordinates define the position of the center of the lower base of the indicated object.

Fig. 15 *Set Location*

2. **Set Rotation , Size and Location** – allows you to rotate, change the size and location of the object (Fig. 16). The appropriate option is selected in the *Set* field.

Fig. 16 *Set Roatation , Size and Location*

3. **Set Color by Case** – adds a color to the selected object (Fig. 17). The color is assigned based on the value of the expression in the *Value field* . Color numbers can also be assigned manually by adding subsequent items using  (Fig. 17).

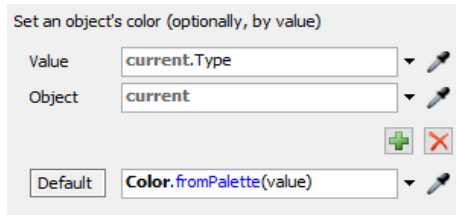


Fig. 17 Set Roatation , Size and Location

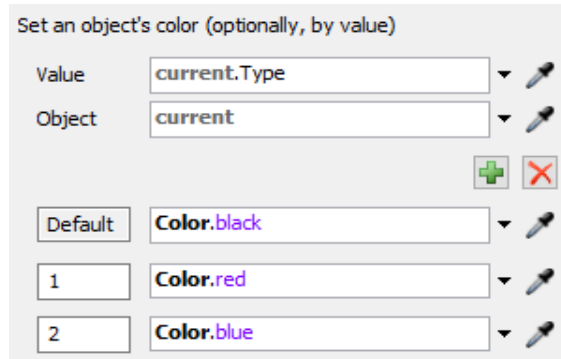


Fig. 17 Options adding colors

4. **Set Object Color** – allows you to set the color of the object selected in the *Object* field (Fig. 18).

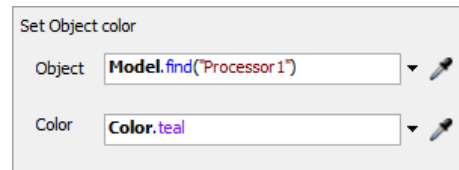


Fig. 18 Set Object Color

5. **Change 3D Shape** – allows you to change the shape of the object indicated in the *Object* field to the shape available on the list in the *Shape* field (Fig. 19).

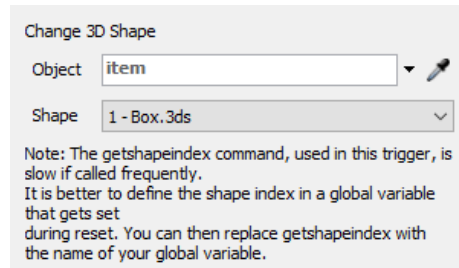


Fig. 19 Change 3D Shape

6. **Start Animation /Stop Animation** – allows you to start or stop the indicated animation (Fig. 20). You can indicate the animation by entering its name or its assigned number, e.g.: the animation of opening doors in a truck is called *OpenDoorsStopped* and has the number 1.

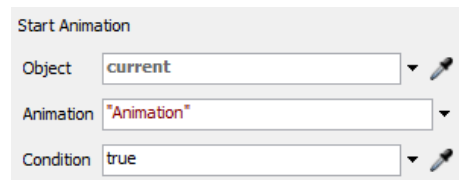


Fig. 20 Start Animation /Stop Animation

4.4. Lists Group

Actions allow you to add or remove items from the list. Adding is triggered by *Push to List* , removing by *Pull From List* (Fig. 21).

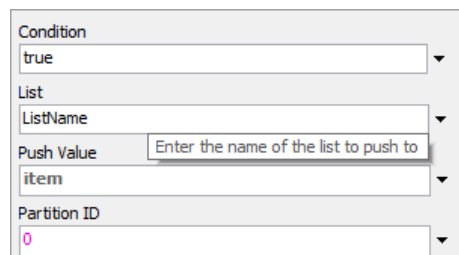



Fig. 21 Push to List option

4.5. Code Group Snippet

The action enables the saved your own logic using the built-in *Flex:Script language* (Fig. 22). The same effect can be achieved by selecting the icon  next to the selected trigger, which allows direct editing of its script as shown in Fig. 23.

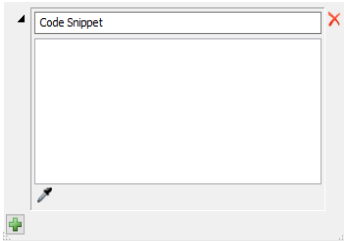


Fig. 21 Code Snippet Window

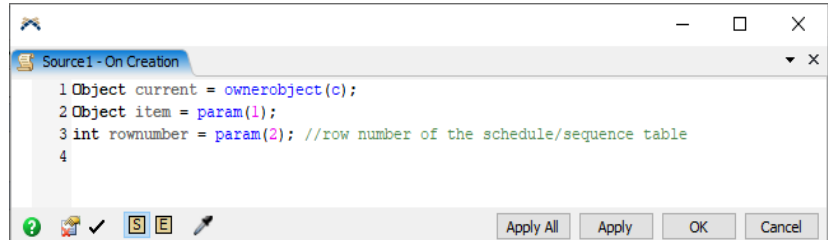


Fig. 22. Direct script editing