
INTELLIGENT LOGISTICS SYSTEMS

ACTIVITY #1

Process Flow – basic concepts

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 master the basic concepts related to process modeling using *Process Flow*, or diagram flow. It allows for a significant expansion of the basic modeling capabilities in 3D mode and the combination of both methods in one model.

New skills
<i>Token Role in creating a flowchart (Process Flow)</i>
Basic types of <i>Process elements Flow</i>
<i>Shared Resources Assets) on the example of using the operator</i>
Basic analysis of simulation results using the capabilities of <i>Process elements Flow</i>

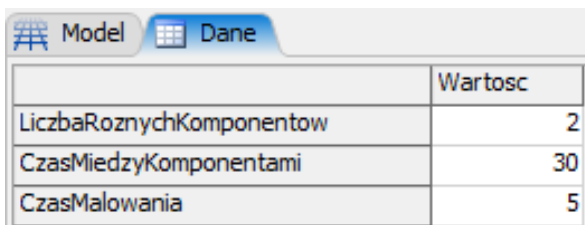
2. Model assumptions and input data

The model being created will combine 3D modeling with the use of diagram flow control functions. The behavior of objects on the model plane will therefore be controlled by a sequence of events associated with them in *the Process Flow*.

The model being created will represent the process of painting two components according to the following assumptions:

1. Components to be painted appear in the system according to an exponential distribution with an expected value of 30 seconds.
2. The number of component types is 2.
3. The painting time is fixed at 5 seconds for a single component.
4. Components are moved between stations by an operator.
5. After painting, the products are placed in a common buffer.

The input data are located in the global *Data* table (Fig. 1), while the target model layout is shown in Fig. 2.



	Wartosc
LiczbaRoznychKomponentow	2
CzasMiedzyKomponentami	30
CzasMalowania	5

Fig. 1. Model parameters in the *Data table*

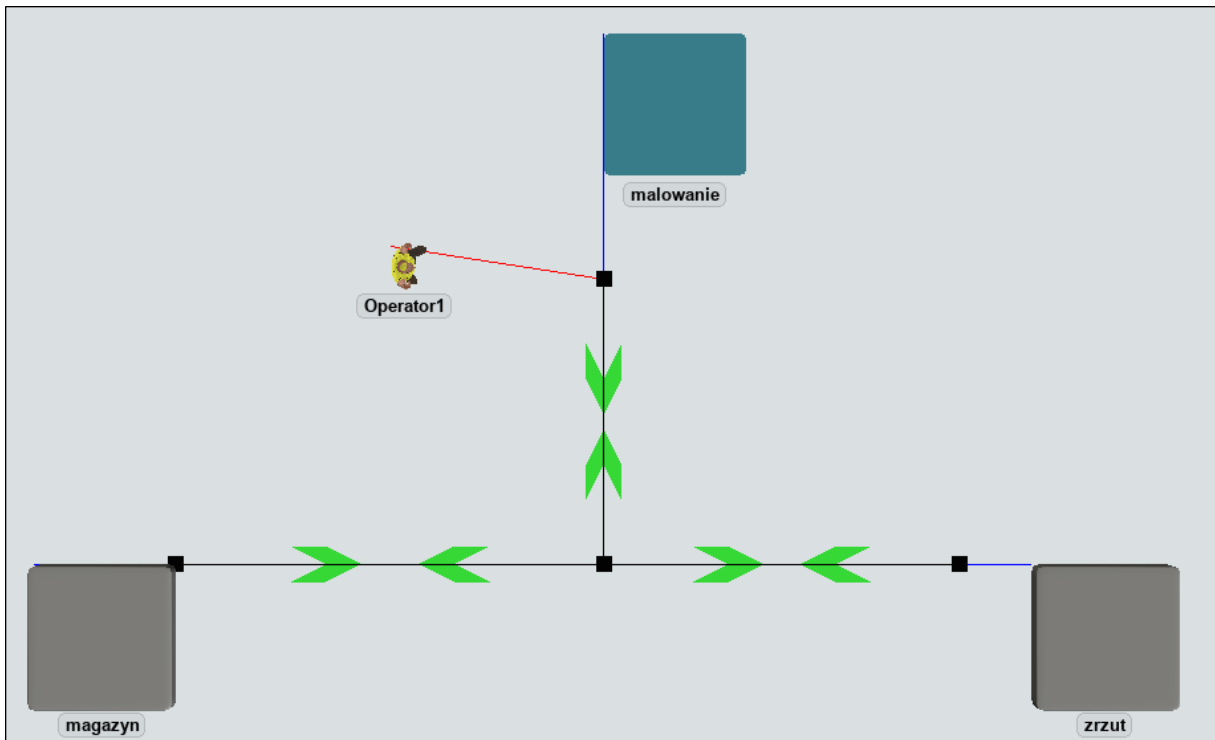


Fig. 2. 3D model layout

3. Model control logic

3.1. Model structure

3.1.1. Logical diagram of the model

The model control logic that will be mapped to *Process Flow*, is shown in Fig. 3.

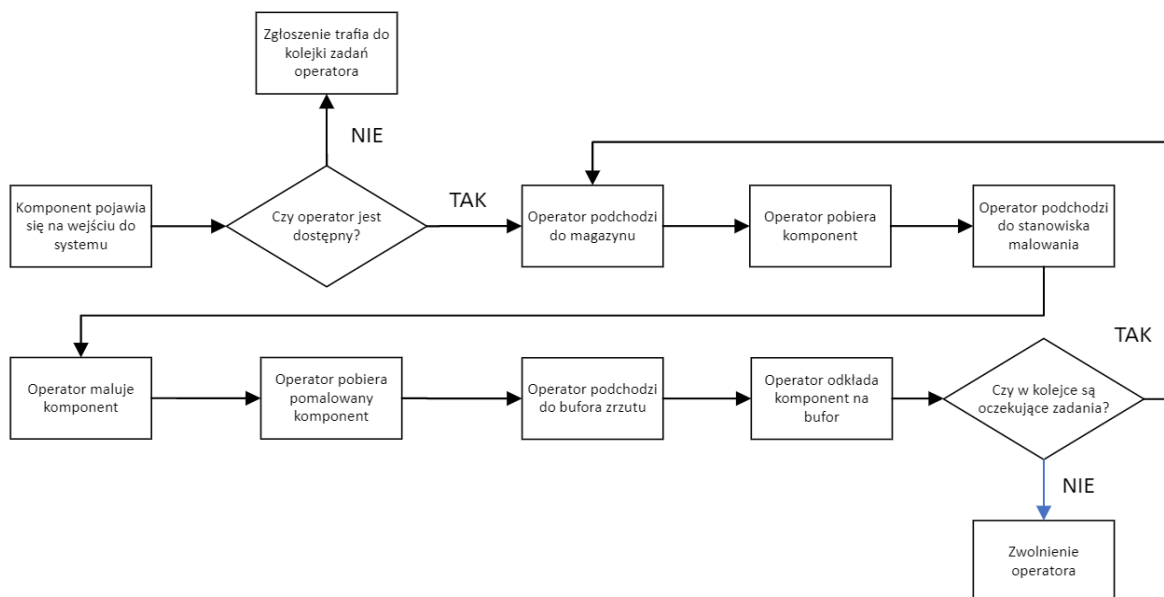


Fig. 3 Model control logic

3.1.2. Process Structure Flow

The structure of the block diagram corresponding to the logical diagram of the model is shown in Fig. 4.



Fig. 4 Block diagram

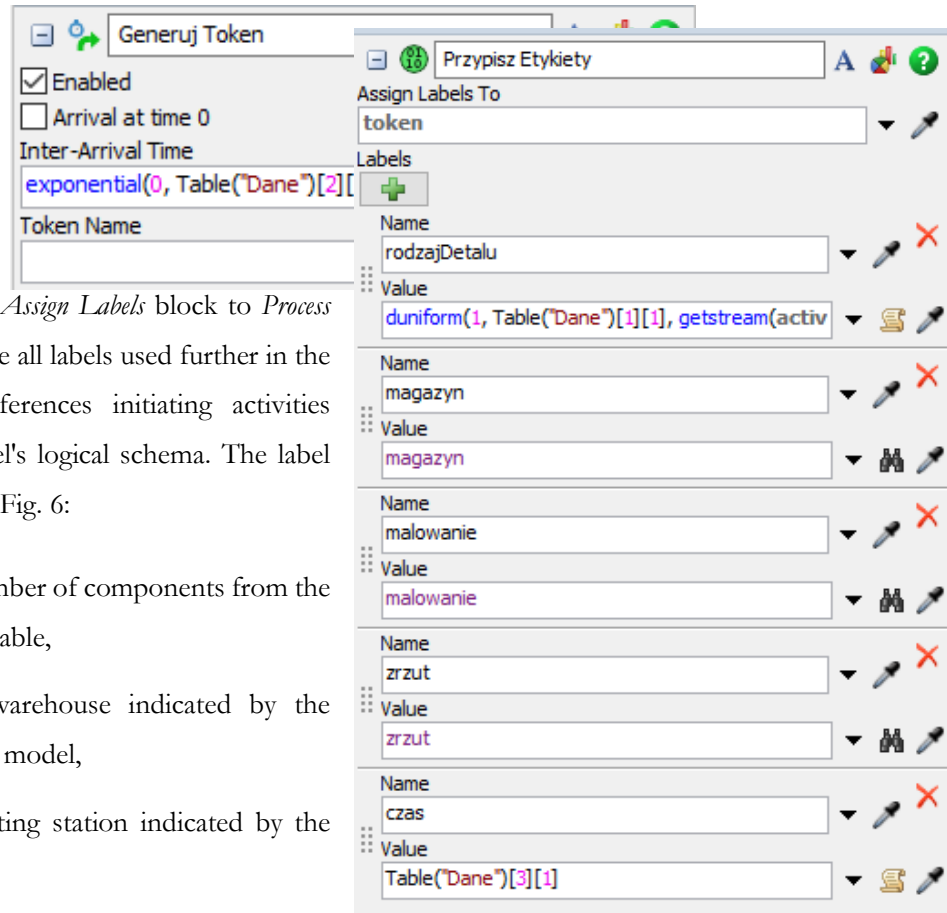
3.2. Token Generation - *Iner-Arrival Source*

There is no source generating flow units in the 3D model, so the source in *Process Flow* will take over this role. This source will generate tokens that will then be associated with the action generating flow elements in the 3D model on the *Warehouse* buffer. On the *Process Flow* tab you should place the *Inter-Arrival Source*

block and configure it according to Figure 5. Flow elements will be generated on average every 30 seconds according to the data in row [2] of the *Data table*.

Fig. 5 *Inter- Arrival Source*

3.3. Labels - Assign Labels



You need to add an *Assign Labels* block to *Process Flow*, which will define all labels used further in the model to create references initiating activities included in the model's logical schema. The label settings are shown in Fig. 6:

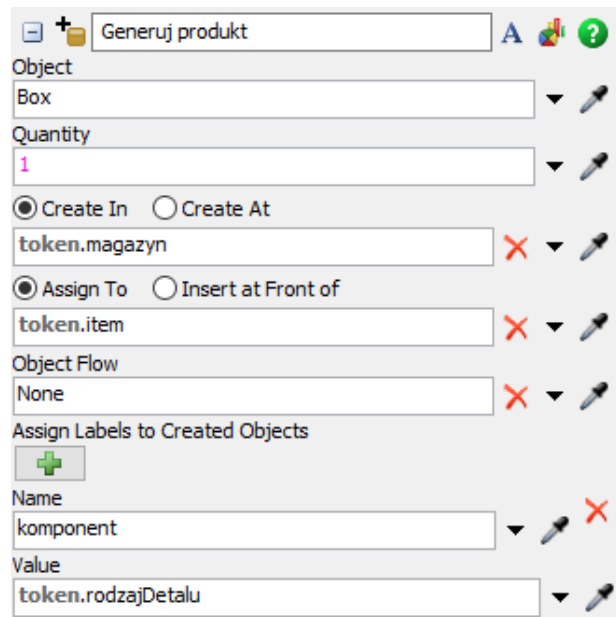
- rodzajDetalu – number of components from the field [1][1] of the table,
- magazyn – the warehouse indicated by the sampler in the 3D model,
- malowanie – painting station indicated by the sampler,
- zrzut – dump buffer indicated by the sampler,
- czas – painting time according to field [3][1] of the table.

Assign Settings Labels

3.4. Generating Components – Create Object

The next task is to generate flow elements in the indicated place, i.e. in the *warehouse* buffer. The *Create Object* block is used for this purpose, which will generate a given number (*Quantity*) of flow elements (*Object*) each time a token is entered. The reference to the place where the elements are generated is saved in the

warehouse label. Therefore, in the *Create in n* field but choose *Token Label* and **token.magazyn** (Fig. 7). The reference to the component type will be saved in the label named **komponent**, corresponding to the value of the label **rodzajDetalu** on the token .



Create Object Settings

The effect of the introduced changes is shown in Fig. 8. The flow element has one label, while the token that generated it has seven labels, including the operator indication.

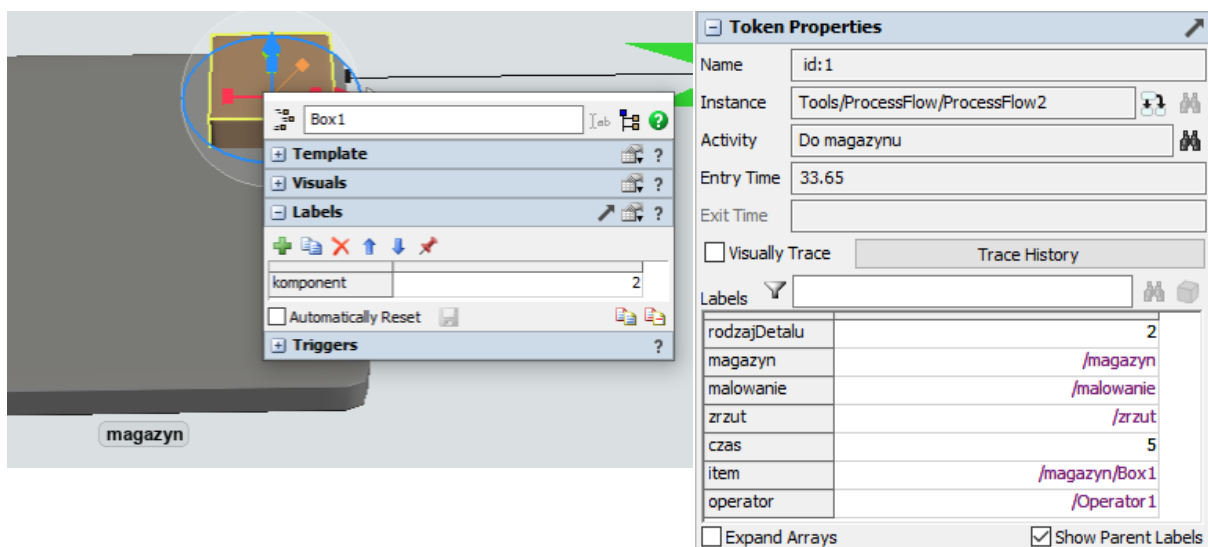

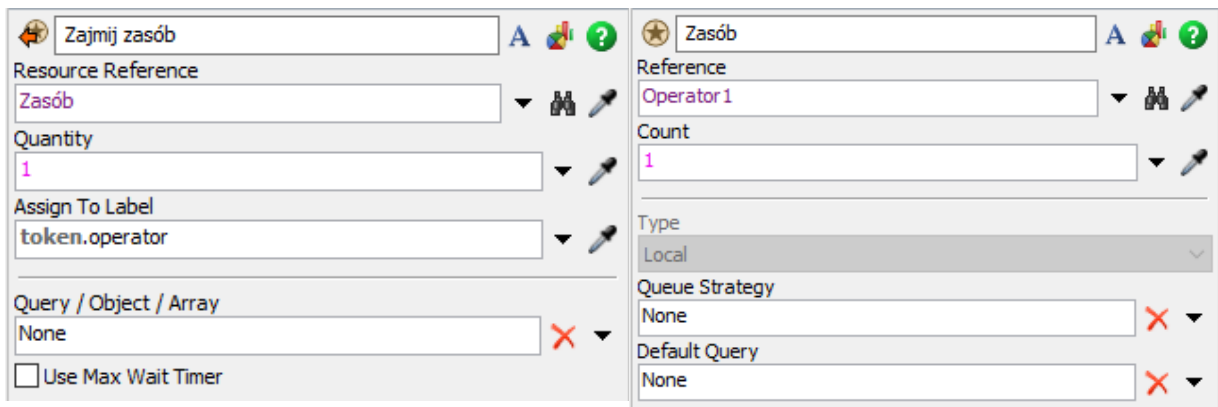


Fig. 8. *Box element* generated using the *Create Object* block

3.5. Operator – Resource

Subsequent actions in the model require the participation of an operator, who will be treated as a resource (*Resource*). The pair of blocks *Acquire* and *Resource* are responsible for assigning tasks. The token entering the *Acquire* block reports a request to occupy the resource indicated in *Resource*. Both blocks must be connected to each other - the lack of  connection is indicated by a red exclamation mark. The *Acquire* and *Resource* blocks are shown in Figure 9.



Acquire and Resource block settings

In the *Resource* block, the picker is used to indicate the operator in the 3D model, setting the available number of units to 1. In the *Claim Resource* block, there is a reference to the related *Resource block* and the number of occupied resources (1). The *Assign to Label* field is used to define a label whose value will be a reference to the occupied resource.

An example of the operation of both blocks and the creation of a list of waiting tasks is shown in Fig. 10. It shows one token symbolizing an active action performed by the resource, in this case moving the operator to the warehouse to pick up a component. The red token with the number 2 indicates two tasks (components) waiting in the *Magazyn* buffer to be picked up by the operator.

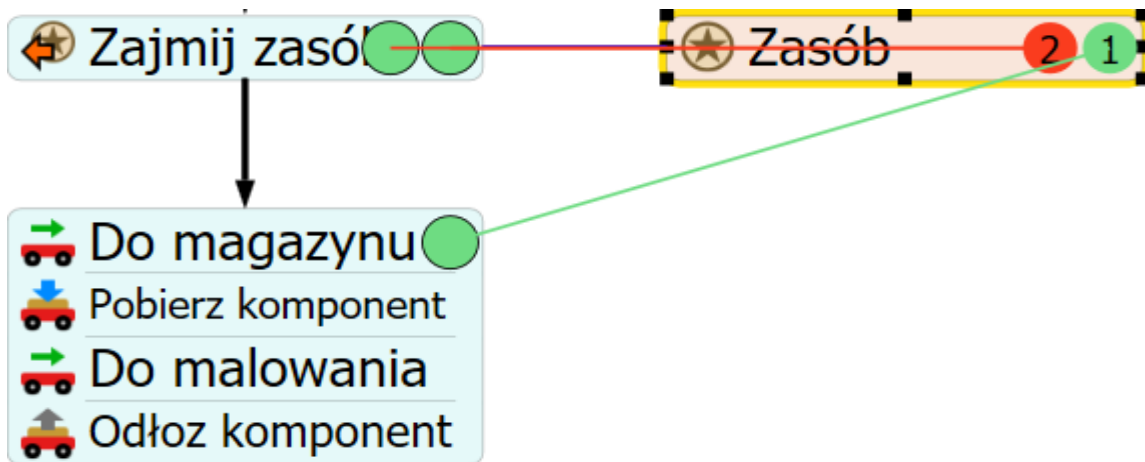
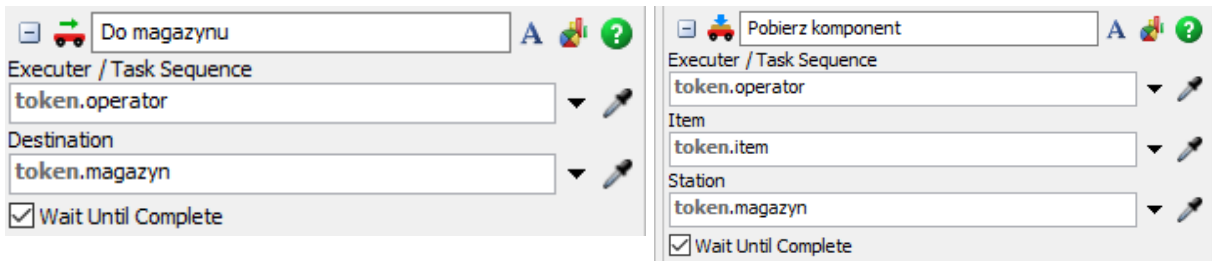


Fig. 10. Acquire and Resource block actions

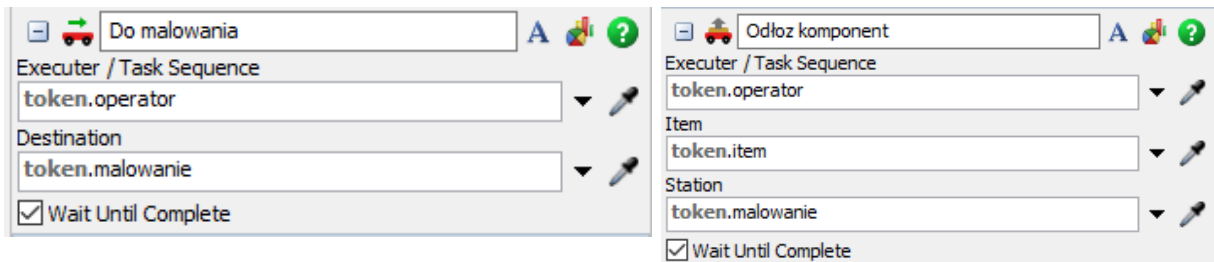
3.6. Task sequence - taking up the component

The component is picked up by the operator using the *Travel*, *Load* and *Unload* task sequence. In the model being developed, references to objects are implemented using previously entered labels. In the case of the blocks under consideration, this can also be done by indicating individual elements (e.g. *Executer* and *Destination* with the picker in the model area). Configuration of the *Travel* and *Load* tasks shows Fig. 11.



Travel and Load block settings (To warehouse/Load component)

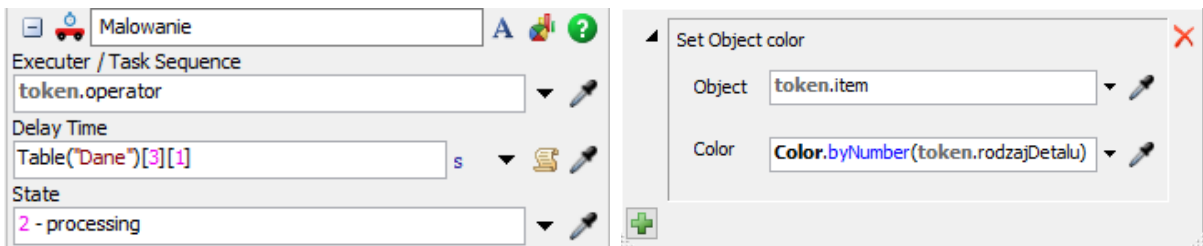
The next step is to move the operator with the collected component to the painting station and put the component away (Fig.12). The destination, i.e. the *Malowanie* buffer, is indicated using a label.



Paint and Travel/Unload block settings

3.7. Painting – Delay, Change Visual

After the component is put on the station, the painting process is mapped. *The Delay* (painting time) and *Change Visual* (color change depending on the component type) blocks will be used for this purpose. The painting time was specified in the data table and transferred to the time label. In the *State* field, *2-processing* should be selected from the list of possible states. The color change is triggered by the *Set Object Color* option, where the color type (its number in the palette) depends on the part number (1 or 2). The settings of both blocks are shown in Fig. 13.



Malowanie (Delay) block settings and Set Object Color color changes

3.8. Task sequence - component put away

The *Load*, *Travel* and *Unload* actions are repeated and the destination is the *Zrzut* buffer (Fig. 14).

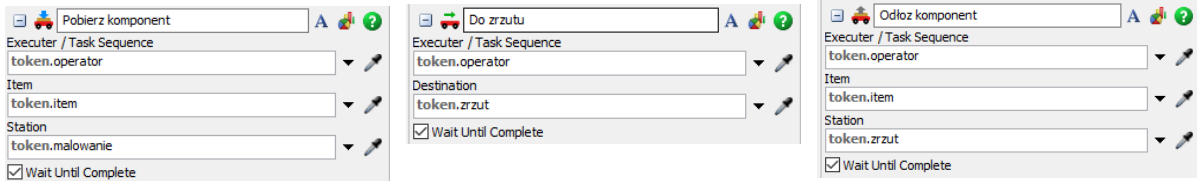


Fig. 14. Sequence of actions - putting the component away

3.9. Operator Release

The last block in the *Flow Process* is to release the operator after completing the given activities (*Release*). After releasing, the resource can be used by subsequent tokens to perform tasks waiting in the queue (Fig. 15).

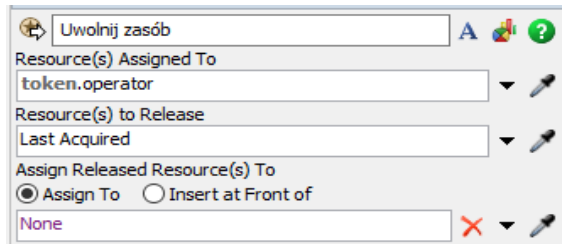


Fig. 15 *Release* – releasing the resource

4. Simulation results

Each block collects a certain set of statistics about its activity. They are available in the upper right corner of each block (Fig. 16). After selecting it, a list of generated tokens will appear



Fig. 16 Statistics option in the *Inter- Arrival Time* block

Statistics can also be presented graphically (chart or table). This is done using the red pin next to the data field. Select *Pin to New Dashboard* and, for example, *Output vs. Time*. The result is shown in Fig. 17.

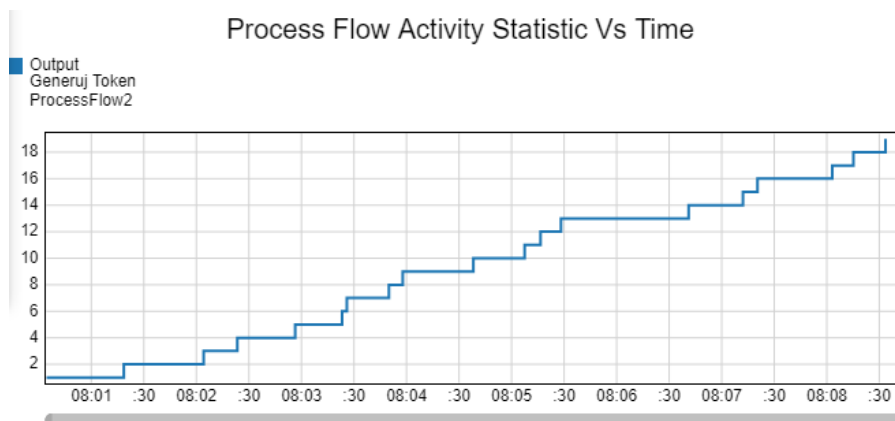


Fig. 17 Number of generated tokens

In a similar way, the total number of processed components can be assessed. It can be read from the *Sink block statistics* (Fig. 18).

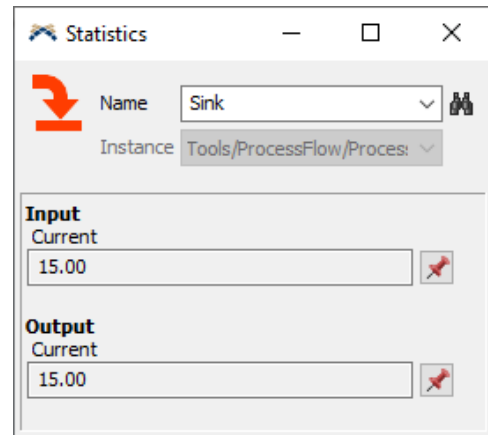


Fig. 18 Number of processed elements

Resource block statistics (Fig. 19) provide a lot of useful information. They concern the area between the *Acquire Resource* and *Release Resource* blocks. For example, the average time of manipulating a single element (*Staytime*) was 29.47 seconds. The average waiting time in the queue (*Request Staytime*) was 36.34 sec and the maximum number of items waiting in the queue was 4. The average resource load (*Content*) was 88%.

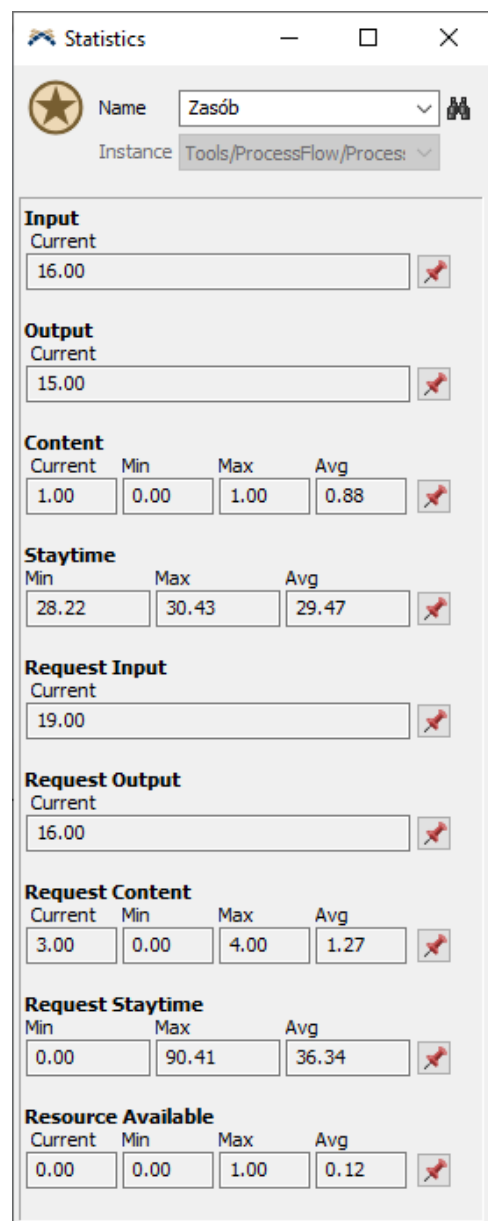


Fig. 19 *Resource block* statistics