DRAKON Visual Language

Tutorial. Part 2:
Interaction with the outside world, parallel algorithms and real time
The extended set of DRAKON icons

Receive by mail

letter from friend

Send by mail

response to friend

area

width \times height

Start

Compilation

\[ T = 0 \]

20 sec.

Input

Output

Shelf

Parallel process

Timer

Pause, Synchronizer
What can a “Shelf” icon do?

- Put a value on the shelf
- Send a command to the doer
- Terminate the algorithm
Put a value on the shelf

- Assign a new value to the property
- Put a new value in the variable
- Set or clear a flag
Put a value on the shelf

Place the product of width and height in the “area” variable

Set the “Power ON” flag

area

width × height

"Power" led is ON

yes
Shelf: send a command to the doer
Shelf: terminate the algorithm

Example:
- “Make a new version” algorithm invokes “Development task” algorithm
- “Development task” algorithm invokes “Analyze dependencies” algorithm
- This “Shelf” icon is executed inside “Analyze dependencies” algorithm

What happens next?
- “Analyze dependencies” terminates
- “Development task” terminates
- The control is returned to “Make a new version”
Interaction with the outside world

- “Input” icon
- “Output” icon
- “Parallel process” icon
Input

Receive by mail
letter from the friend

Output

Send by mail
response to the friend

Parallel process

Start
Compilation
A keyword or a key phrase

The upper floor

Input

Receive by mail
letter from the friend

Output

Send by mail
response to the friend

Parallel process

Start
Compilation
The lower floor

Input
Receive by mail
letter from the friend

Output
Send by mail
response to the friend

Parallel process
Start
Compilation

The details
What does “Input” icon do?

- Receives information from external senders
- Gets messages from a parallel process
- Performs data input into the computer:
  - Receives data from the user via keyboard and mouse
  - Reads data from disk
  - Receives data from the network
What does “Output” icon do?

- Sends information to external recipients
- Sends messages to a parallel process
- Performs data output from the computer:
  - Shows data on the screen
  - Writes data to disk
  - Sends data over the network
An example with “Input” and “Output”

Correspondence

- Receive by mail
- letter from the friend

Read the letter

Is the letter insulting?
  - Yes: Write a response
  - No: End

Send by mail
response to the friend
What is the difference between “Shelf” and “Output”? 

**Shelf**
- Assigns a value to an internal variable
- Gives an order to an internal doer

**Output**
- Sends a message to an external process
- Sends information to an external recipient
The commands that control parallel processes

- **Start**: Starts a parallel process
- **Stop**: Stops the parallel process forever
- **Suspend**: Suspends the parallel process
- **Resume**: Resumes the suspended process
Controlling a parallel process

1. Background compile
2. Show "Building..." window
3. Start
4. Compilation
5. Wait for event
6. (block until any event)
7. Event type
8. Compilation completed
9. "Cancel" pressed
10. Progress
11. Stop
12. Compilation
13. Close "Building..." window
14. End
Commands sent to a parallel process do not suspend the main process.
Internal parallel processes

Make lunch

Cook potatoes
Boil potatoes
Boil meat
End

This double horizontal line starts a parallel skewer
Internal parallel processes

The parallel skewer executes simultaneously with the main skewer

- Make lunch
  - Cook potatoes
  - Boil potatoes
- Boil meat
- End
Internal parallel processes

Make lunch

Cook potatoes
Boil potatoes
Boil meat

End

This horizontal line stops the parallel skewer
Wrong! The main skewer must not be broken

Flowchart:

- Make lunch
  - Cook potatoes
  - Boil potatoes
  - Boil meat
  - End
The following actions can be done in any order after the field of operations is prepared:
- Extend electric power line
- Purchase wires
- Install windows
- Finish roof
- Purchase water pipes

The assembling of the electric panel cannot be started before the installing of windows and finishing the roof are complete.
Gluing plastic

- Clean the surfaces
- Degrease the surfaces
- Put glue on the surfaces
- 20 sec.
- Connect and press together the surfaces
- 24 hours
- Remove the pressing
- End

Pause

Adds a delay between two operators
Timer and synchronizer

Timer “T” starts

1. Wait for event
2. The orientation of crankshaft reaches 0 degrees
3. Receive sensor value
4. Crankshaft rotation speed (period of revolution)
5. T = 0
6. T = 0.3 of period
7. Send command
8. 1st cylinder ignition
9. T = 0.5 of period
10. Send command
11. 2nd cylinder ignition
12. Ignition ON?
   - Yes
   - No
      - End
This operator executes when the specified time has passed since the start of timer “T”. (0.3 of period of revolution of crankshaft)
What is the difference between "Pause" and "Synchronizer"?

**Pause**
- Runs the next operator after some time has passed since the previous operator
- No need for a timer

**Synchronizer**
- Runs the operator after some time has passed since the timer start
- The timer must be started
A "Wait" loop sleeps between iterations.

The dummy web service sends "No connection to database" error for each request.
Stepan Mitkin
stipan.mitkin@gmail.com

The diagrams on the slides were made with DRAKON Editor

http://drakon-editor.sourceforge.net/

June 2013