

Workflows editor – SRS

This document specifies software requirements on the workflow editor module of the FlowOpt project.

Used terms and notes

For the sake of this document, clicking refers to left clicking; dragging refers to dragging the mouse while holding the left mouse button.

In the text, whenever I mention some term that will be referred to later in the document, it is written in *italics*. Fragments of declarative format code will be written in `courier`. In some cases, I'm not yet certain on the solution of the specific problem – these cases are marked by (TBD) mark and will be resolved later (advice on such problems is especially welcome).

Both the Nested TNA model and the (simple) TNA model are thoroughly defined in <http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.69.5684&rep=rep1&type=pdf>.

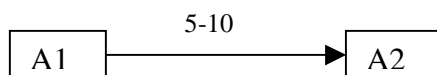
By declarative format, I mean the data format used to pass a created workflow to the scheduling engine. Its complete description will be in project specification. In this document, I use n-ary Activity predicate instead of unary simply to save space, therefore instead of:

```
Activity(A1, Duration of A1, Cost of A1, ...)  
Activity(A2, Duration of A2, Cost of A2, ...)  
Activity(A3, Duration of A3, Cost of A3, ...)
```

I only write

```
Activity(A1, A2, A3)
```

Also, for every arc there will be a predicate with its temporal constraint like in following case:



Produces a predicate like

```
Temporal(A1, A2, 5, 10, ES)
```

I don't explicitly write these temporal predicates, again to save space. All temporal predicates that are explicitly written are independent of individual arcs (they are a part of specific pattern transformation).

Main modes

The workflows editor will operate in two basic modes: the *Nested TNA mode* and the *freelance* mode for free user editing. The former will focus on more constrained editing in the sense that the user will not be able to directly control neither the workflow structure, nor the workflow's visualization. User will define the structure of the workflow only by using one of several predefined editing actions, which are guaranteed to keep the workflow in Nested TNA mode. The user's ability to influence visualization in this mode will be limited to setting padding between activities on the screen and zooming, the user won't be able to directly move

activities or arcs between them in Nested TNA mode. The idea of this mode is to constrain the user to a specified model (Nested TNA), which can be handled well by the application – it can be visualized nicely without the user's input and it can be used for scheduling by other FlowOpt modules easily and effectively.

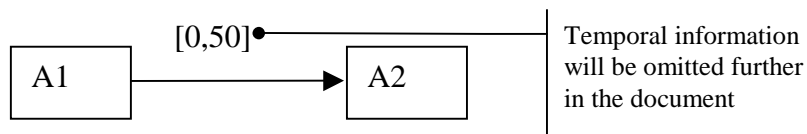
The freelance mode is the counterpart of Nested TNA mode – the user can freely change both structure and visualization of the workflow. In this case, the network will be in the TNA model (**TBD**). The user will be able to arbitrarily create and connect activities with arcs, as well as freely move and delete activities. The application's part in visualization will be limited to (user activated) aligning of activities and automatic arc resizing when the user moves or resizes an activity (so that the arcs move with the activities). Workflows created in this mode can still be used for scheduling, but it may not be as effective as in case of those created in Nested TNA mode.

Basics

The aspects described in this section apply for both the Nested TNA mode and the freelance mode.

The workflow consists of *activities* connected by *arcs*, i.e. our workflow is a (connected) DAG (directed acyclic graph), where vertices are activities and edges represent precedence (and temporal) relations.

The workflow is visualized by drawing activities as rectangles with their IDs and arcs connecting them as arrows with temporal information, as in the following figure. In the rest of the figures in this document, temporal information will be omitted for the sake of simplicity.



In both modes the user will be able to perform an undo operation.

Every activity has a number of properties – i.e. ID, duration, cost, due date, late cost, resources needed etc. Definition of properties is in project specification. Every arc has a single temporal constraint – minimal and maximal time span between the activities this arc connects.

Below is the basic overview of the main form.

| | |
|---|---|
| Main menu – File Edit ... Help | |
| Toolbar – icon shortcuts for aligning, orientation switch, calendar etc. | |
| Drawing area – where the workflow is displayed and edited | Activity properties – ID, duration, cost, due date etc, When an activity is selected, it's data can be seen and edited here |
| Status bar, for ex. Current mode (Nested TNA/Freelance), workflow name etc. | |

Zooming

The user will be able to zoom in and out in both modes. The application will handle automatic detail hiding when the user zooms out far enough. There won't be a magnifying glass functionality (i.e. The possibility to enlarge just a selected part of the screen, without the rest changing size).

Selection

The user will be able to only select a single activity by left clicking on it. By doing so, its properties will be displayed in the activity property form, where they can be viewed and edited. As for the arcs, the user may input temporal information for any arc by double clicking on it, which will cause an input field to appear, allowing user to enter the desired time interval (minimal – maximal time distance between the two connected activities).

Nested TNA mode

In this mode, the user will only be able to perform such actions, which will leave the workflow in a state corresponding to the Nested TNA model. This makes the user's ability to change both the workflow and the way it is visualized much more limited than in the freelance mode.

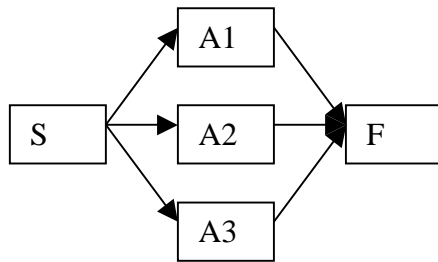
Visualization

In the Nested TNA mode, the application controls most of the visualization logic – due to the Nested TNA model; it is possible to draw the workflow without the user's help. The user will be able to perform the following actions in the Nested TNA mode:

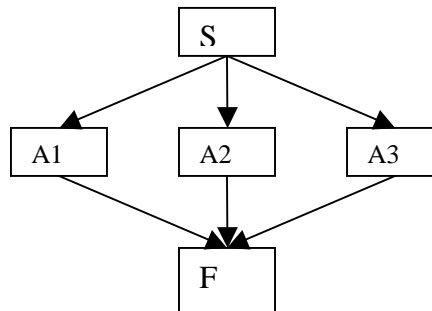
Choice of orientation

In any moment, the user will be able to switch between two orientations – horizontal (left to right) and vertical (top to bottom). The following figure illustrates both cases:

Horizontal orientation:



Vertical orientation:



Editing

In this mode, the user can only perform three editing actions: serial, parallel and alternative decomposition – all of them preserve the workflow in Nested TNA form. Each of them is further described below, together with a figure that illustrates its use. Also, for each of these editing actions, there is an explanation of how to encode the resulting workflow (after the action is performed) into the declarative format we use for scheduling.

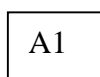
In the beginning, the user starts with a workflow consisting of a single activity (vertex)¹. For sake of this specification, this activity will be called the *initial activity*. On this elementary workflow, the user applies the following three editing actions:

- Serial decomposition – the user can split a single activity into multiple activities, which will be executed one after another in place of the original activity.
- Parallel decomposition – the user can split a single activity (vertex) into multiple activities, which will be executed in parallel in place of the original activity (AND split / join).
- Alternative decomposition – the user can split a single activity into multiple activities, out of which a single activity will be executed in place of the original activity (XOR split / join).

Description of editing process

Here is a step-by-step creation of a simple workflow to illustrate the process. It assumes the horizontal orientation (for vertical orientation, the left-right dragging is replaced by up-down dragging and vice versa).

- 1) The user starts with a single activity (the initial activity), which represents the entire workflow.

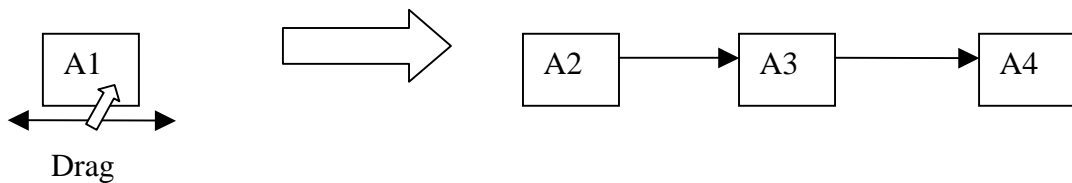


Declarative format transcription:

Activity(A1)

¹ In the formal Nested TNA model, the user starts with two activities connected by an arc, however unlike the formal model, we decompose the vertices rather than arcs, hence the slight change of model.

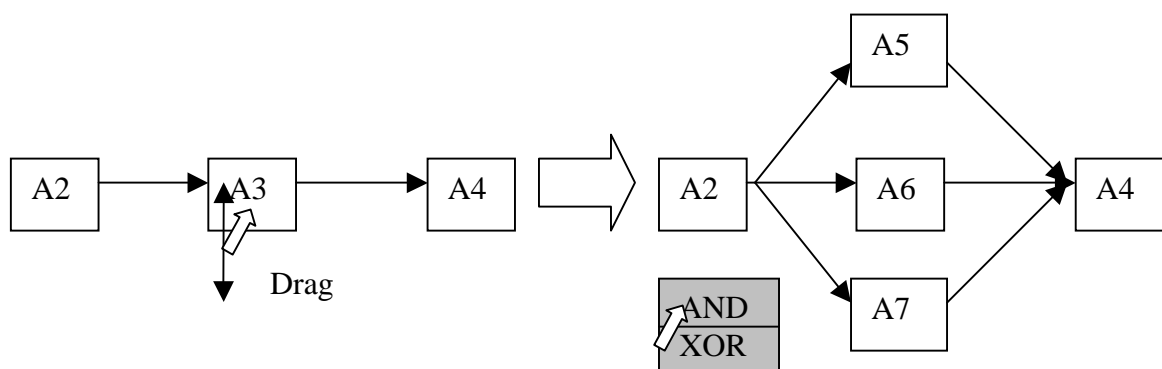
- 2) Serial decomposition application – the user clicks on the activity, which he wants to decompose (in this case A1) and drags the mouse while holding the left mouse button. Dragging to the right adds more activities to the decomposition, dragging to the left removes activities from the decomposition (in this case, the user splits the A1 activity into three serially executed activities A2,A3 and A4):



Declarative format transcription:

```
Activity(A1,A2,A3,A4)2  
Decomposition(A1,[A2,A3,A4],AND)  
Temporal(A2,A3,0,inf,ES)  
Temporal(A3,A4,0,inf,ES)
```

- 3) Parallel decomposition application – The user clicks on the activity, which should be decomposed (in this case A3) and by holding the left mouse button and dragging the mouse he specifies the number of activities in the decomposition similarly as in the serial decomposition case – dragging up adds more activities into the decomposition, dragging down removes activities from the decomposition. Here the user decomposed the A3 activity into three activities A5, A6 and A7, which should be executed in parallel in place of the original A3 activity. When the user releases the mouse button, a pop-up menu automatically appears with the choice of either AND split (parallel decomposition) or a XOR split (alternative decomposition) – when the user wishes to perform an alternative decomposition, the process is the same as with parallel, except for selecting XOR split in the pop-up menu.

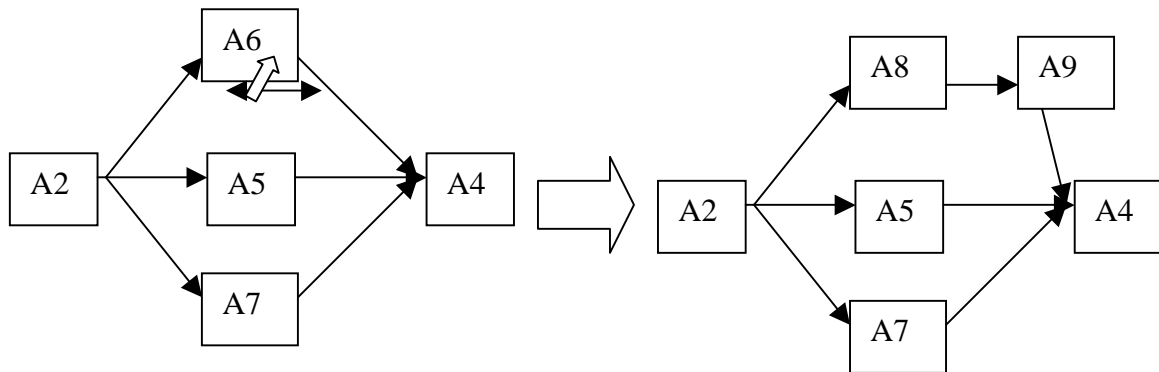


² To save space, I use n-ary predicate instead of unary for specifying activities.

Declarative format transcription:

```
Activity(A1,A2,A3,A4,A5,A6,A7)  
Decomposition(A1,[A2,A3,A4],AND)  
Temporal(A2,A3,0,inf,ES)  
Temporal(A3,A4,0,inf,ES)  
Decomposition(A3,[A5,A6,A7],AND)
```

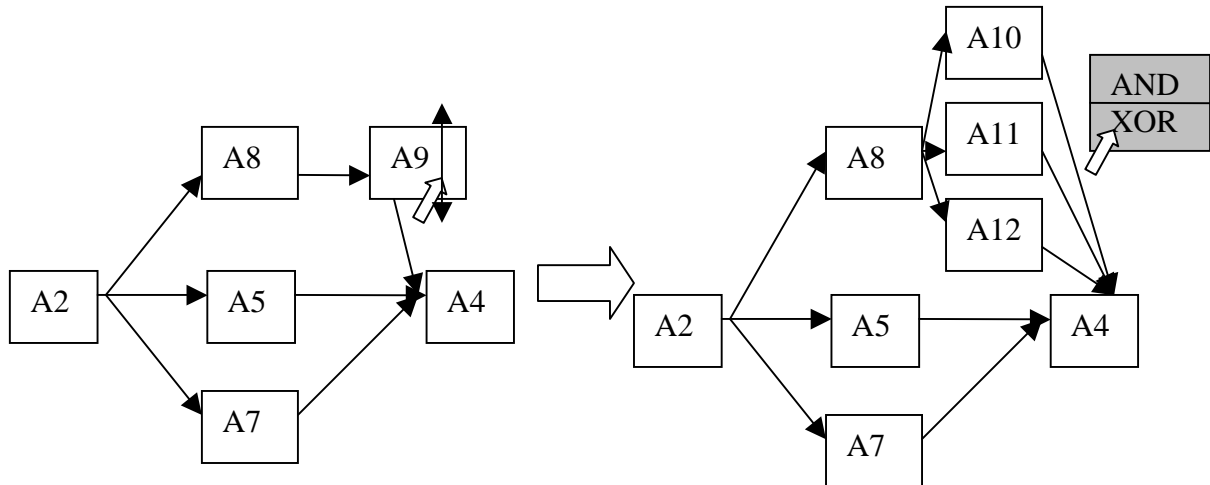
- 4) Further serial decomposition. This time the user chooses to decompose A6 into two activities A8 and A9:



Declarative format transcription:

```
Activity(A1,A2,A3,A4,A5,A6,A7,A8,A9)  
Decomposition(A1,[A2,A3,A4],AND)  
Temporal(A2,A3,0,inf,ES)  
Temporal(A3,A4,0,inf,ES)  
Decomposition(A3,[A5,A6,A7],AND)  
Decomposition(A6,[A8,A9],AND)  
Temporal(A8,A9,0,inf,ES)
```

- 5) Alternative decomposition – As stated earlier, the process is similar to that of parallel decomposition, except in the pop-up menu XOR is selected indicating that the user wishes for an alternative decomposition, not parallel. Here the user chose to decompose the A9 activity into three activities A10, A11 and A12. When the A9 activity should be performed in the original workflow, exactly one of the activities A10, A11, A12 will be executed in the new workflow:



Declarative format transcription:

```

Activity(A1,A2,A3,A4,A5,A6,A7,A8,A9,A10,A11,A12)
Decomposition(A1,[A2,A3,A4],AND)
Temporal(A2,A3,0,inf,ES)
Temporal(A3,A4,0,inf,ES)
Decomposition(A3,[A5,A6,A7],AND)
Decomposition(A6,[A8,A9],AND)
Temporal(A8,A9,0,inf,ES)
Decomposition(A9,[A10,A11,A12],XOR)
  
```

Hierarchy

The user will be able to work with the natural hierarchy induced by the Nested TNA model. That is, every use of an editing action converts a single activity into several activities, which are executed in certain way (depending on the type of decomposition) in place of the original activity. Such a set of activities is referred to as a *nest*. For example, in the figure above, the activities A10, A11 and A12 form a nest created from the activity A9. This forms a hierarchy of nests within the workflow – each decomposition creates one nest in place of the original activity. The user will be able to show or hide any subset of the nests he created.

In this context there are two kinds of activities – the *composite* activities, and the *elementary* activities. The former have at some point been decomposed (they contain a nest of activities), the latter haven't.

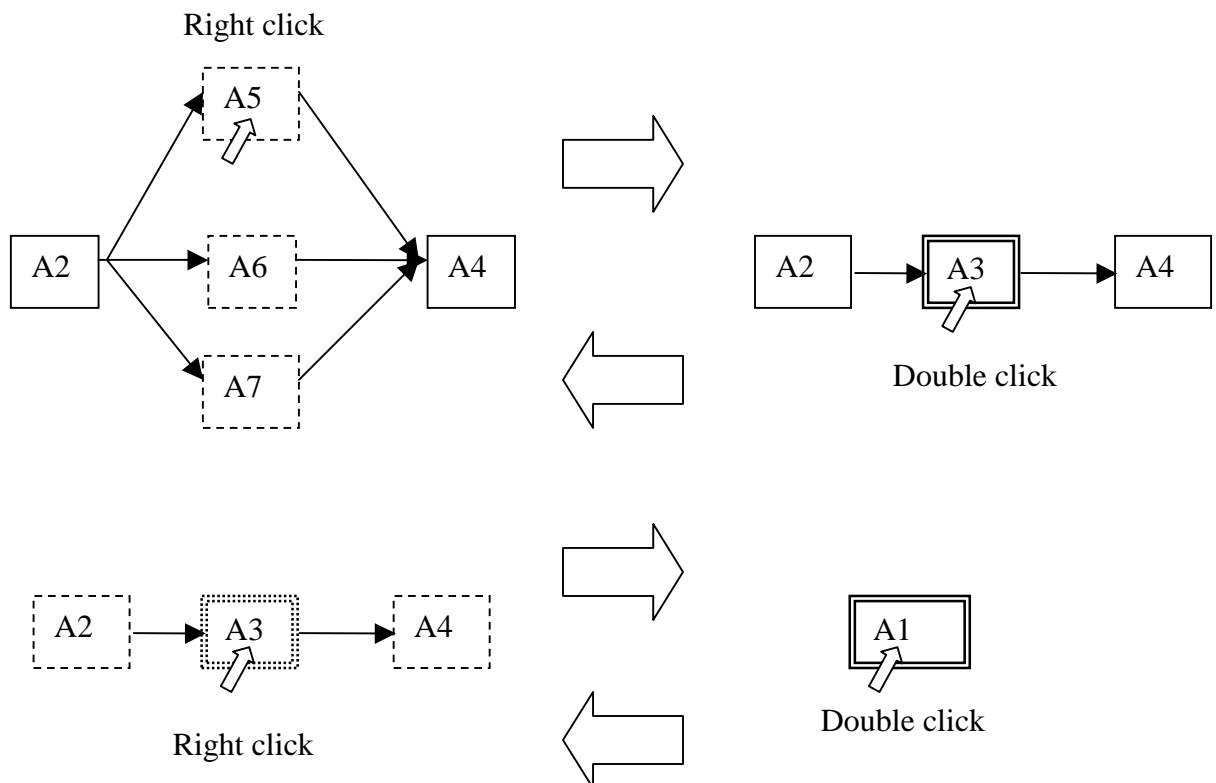
Process description

When the user moves mouse pointer over any activity, all other activities belonging into the same nest are highlighted automatically. Upon right clicking on the activity, the nest is collapsed into a single activity (the original activity), marked in a special way indicating that it is not an elementary activity, but a composite one containing a nest of activities. Upon double clicking on a composite activity, it is once again replaced by the nest of activities within it. This process can be applied arbitrary number of times, giving two extremes: fully expanded workflow, where every elementary activity is visible and fully collapsed workflow, where only one composite activity corresponding to the initial activity (the first one in the workflow, representing the entire process) is displayed.

Example:

In this case, the user created a network consisting of 5 elementary activities A2, A4, A5, A6 and A7. Activities A5, A6 and A7 form a nest decomposed from activity A3, so upon right clicking on any of them, this nest is collapsed as seen in the first figure (arrow to the right). On the other hand, when A3 is collapsed, it can be expanded again by double clicking on it.

The second figure illustrates the same thing one hierarchy level higher – since A2, A3 and A4 are also a nest created from A1 (the initial activity), the user can collapse this nest too. Upon right clicking on A2, A3 or A4, the nest is collapsed and only the initial (now also composite) activity A1 is displayed. Double clicking on it can now expand this activity.



Manipulating activities

It will also be possible to do the following:

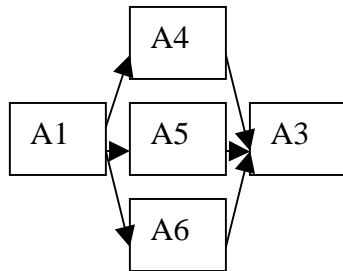
- Add more activities to a nest – right click + “Add more activities to this nest”
- Remove activities from a nest – select activity and press the delete key or right click + “Remove selected activity”.
- Change the order of activities in a nest (“swap” activities) – changes the execution order in case of sequence, for parallel / alternative decomposition it is only for user’s convenience. It will probably be done by using a special “cursor mode” for swapping activities within the same nest.

Padding

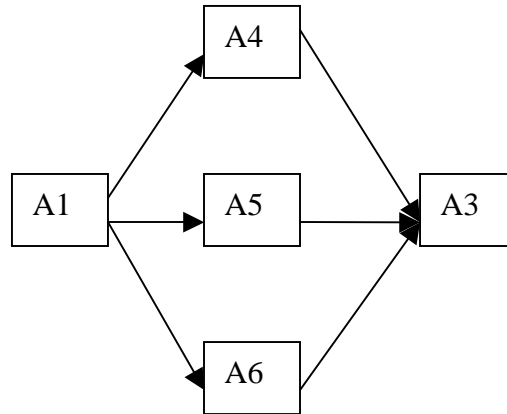
The user will be able to set the padding between activities, i.e. how much space is there between activities.

Example:

Low padding



Higher padding



In the Nested TNA mode, the user won't be able to change the workflow beyond the possibilities described above. For example, the user won't be able to arbitrarily move activities like in the freelance mode.

Freelance mode

In this mode, the focus is on the user – the user can freely edit both the workflow structure by arbitrarily adding/removing activities and arcs between them and the workflow visualization by arbitrarily moving the activities anywhere on the drawing area. As a result of free structure editing, the created workflow may not be scheduled as effectively by other FlowOpt modules as Nested TNA workflows. Similarly as a result of free visualization editing, the application does not „help“ the user with visualization in any way beyond (user triggered) activity aligning and similar actions. Therefore, the user is responsible for making the resulting workflow look nice and manageable.

In this mode, the workflows conform to the (simple) TNA model (TBD).

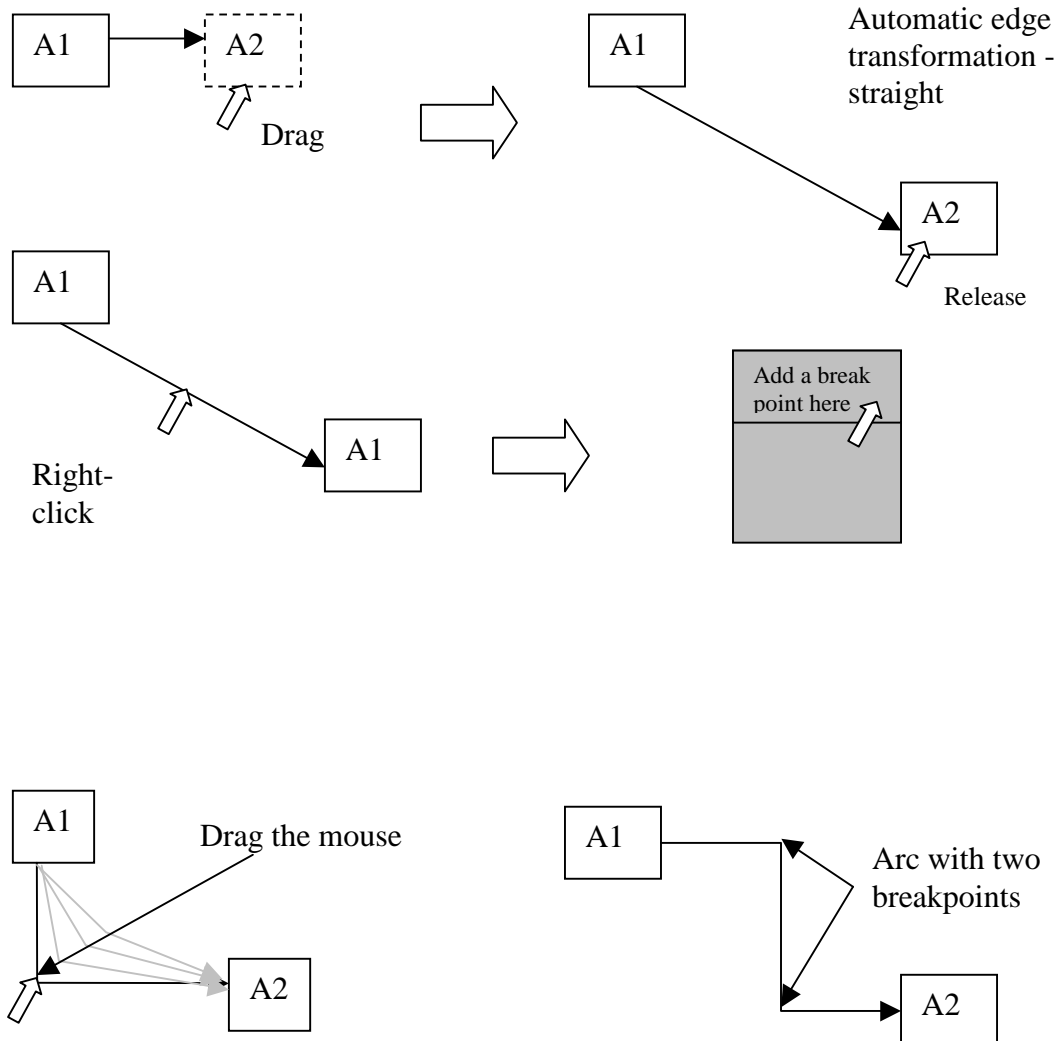
Visualization

In this mode, the user can affect workflow visualization in following ways.

Moving

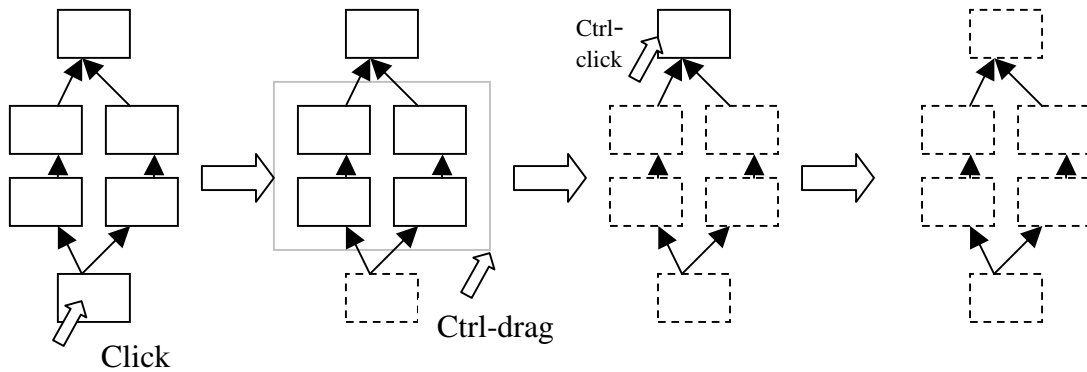
The user can move any activity on the screen by dragging it with the mouse. The activity is moved together with all incident arcs, which are transformed accordingly – they are resized so that they follow the moved activity. Also, the user can edit the arc's shape by adding breakpoints on it. Whenever a breakpoint is placed on an arc, it can be dragged to change the arc's shape as illustrated by following figure:

Example



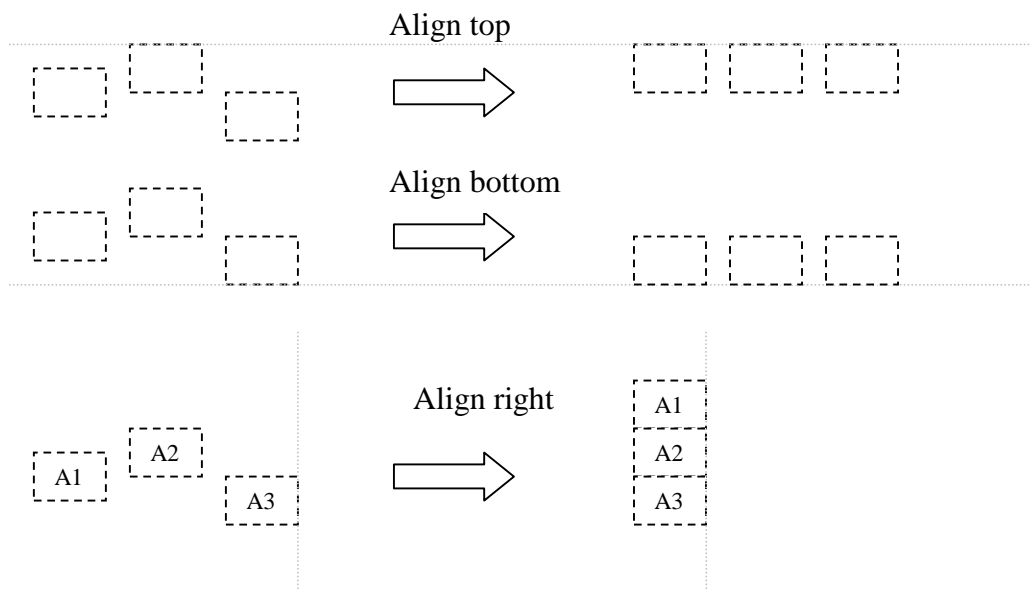
Selection

In this mode, the user will be able to select arbitrary number of activities and arcs in a standard fashion. By clicking on an activity it is selected (only this activity) – its properties are displayed in the activity property form to be viewed and edited. An arc is selected in the same way. By ctrl-clicking on activities, they are added/removed to/from current selection. By dragging the mouse when nothing is selected, a rectangle appears indicating selection – upon releasing the mouse button, all activities within the rectangle will be selected. By ctrl-dragging the mouse, multiple activities can be added to selection. When multiple activities are selected, their properties cannot be edited; the activity property form is only active when a single activity is selected. Selected activities are highlighted (dashed line).



Aligning

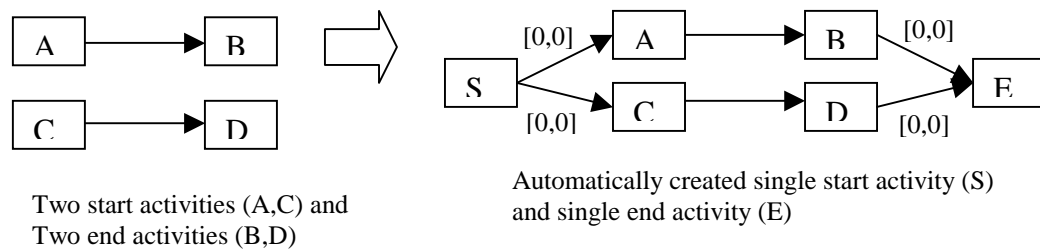
The user can select a number of activities (as described above) and align them so that their top/right/bottom/left edges are on the same level, illustrated by following figure:



Editing

In this mode, the user will be able to edit the structure of the workflow freely. At the beginning, the user starts again with the initial workflow, which in this mode (unlike in the Nested TNA mode) is an empty workflow.

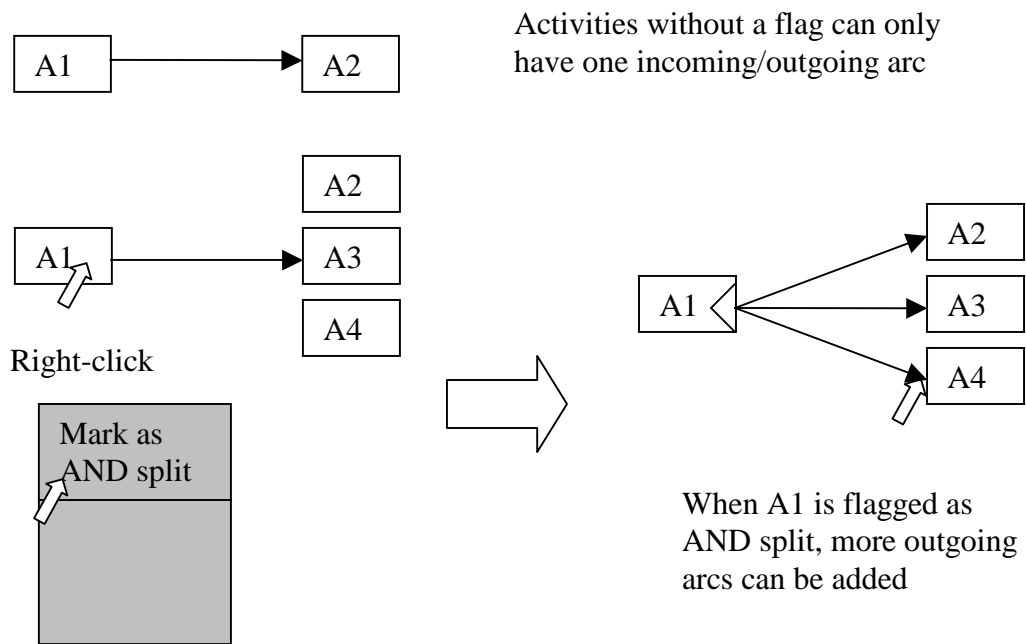
There are no fixed activities like in some workflow editors (i.e. a starting activity and an end activity) – the semantics of TNA allows for multiple “starting” and “end” activities. For the sake of scheduling, all activities with no incoming arcs are considered starting and all activities with no outgoing activities are considered end activities. The application will either require that all the starting activities are scheduled, or it will automatically create a couple of utility activities, that will become a single start activity and single end activity (illustrated by figure), whatever will be deemed more convenient (**TBD**).



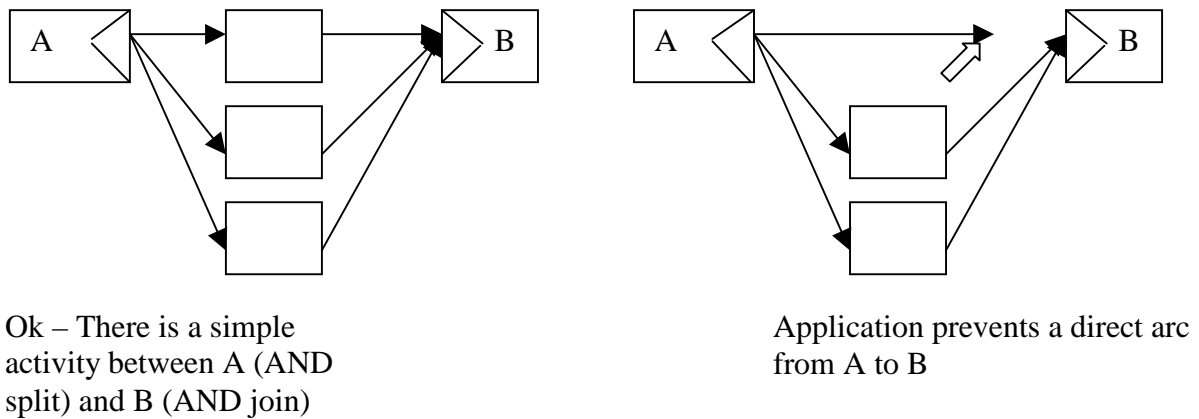
The user can add arbitrary number of activities and connect them with arcs. In this context, two kinds of activities are recognized: *simple* activities and *flagged* activities. All activities are created as simple activities, meaning they can only be a part of a sequence, which means they can have at most one incoming and at most one outgoing arc. If an activity is to have multiple incoming/outgoing arcs, it has to be marked with a flag, becoming a flagged activity. The flag on an activity indicates the role it plays. Here are all the possible flags:

- AND split – the activity splits into several activities, which can be executed in parallel. This flag allows multiple outgoing arcs from the flagged activity. It is mutually exclusive with the XOR split flag.
- XOR split – This activity splits into several activities, out of which exactly one is to be executed. This flag allows multiple outgoing arcs from the flagged activity. This flag is mutually exclusive with the AND split flag.
- AND join – the activity is a synchronization point – it can only start after all the activities with arcs leading into this activity have finished. This flag allows multiple incoming arcs to the flagged activity. This flag is mutually exclusive with the XOR join flag.
- XOR join – This activity can only be executed when one of the activities with arcs leading into this activity finishes. This flag allows multiple incoming arcs into the flagged activity. This flag is mutually exclusive with the AND join flag.

An activity can have no flag, a single flag or two flags, in which case the two flags cannot be both join or both split, as stated above.



Also, activity A cannot be connected by an arc directly with activity B in case A is marked by any split flag and B is marked with any join flag (in TNA terminology – all fan in/fan out sub graphs have to be pair wise edge-disjoint), as illustrated by the following figure:



Actions on selection

Once the user created a selection as described above, following actions are available on the selected activities:

- Removal – the user can remove the selected activities / arcs. For every deleted activity, all incident arcs are automatically deleted too (can be disabled in settings).
- Moving – the user can move all selected activities / arcs simultaneously
- Resizing of activities – only available when a single activity is selected, the user can resize the activity, all arcs incident with this activity transform accordingly, so that they are drawn correctly in relation to resized activity.

All the above actions will be done in a standard way, i.e. removal through pressing the delete button or via a choice in right-click pop-up menu, moving via dragging the mouse and resizing via dragging a corner of selected area.

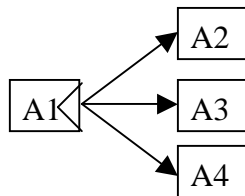
Declarative format

Since the workflows created in freelance mode are potentially more complex than those created in Nested TNA mode, transforming them into the declarative format may prove more difficult. Here is a suggestion on how to transform individual workflow patterns (**TBD**):

Sequence

Same as in the Nested TNA mode

Parallel split



Option 1 – just temporal relations:

```
Activity(A1,A2,A3,A4)
Temporal(A1,A2,0,inf,ES)
Temporal(A1,A3,0,inf,ES)
Temporal(A1,A4,0,inf,ES)
```

Option 2 – via decomposition and logical predicate:

```
Activity(A1,A2,A3,A4)
Decomposition(A5,[A2,A3,A4],AND)
Temporal(A1,A5,0,inf,ES)
Logical(1,5,=>)
```

Alternative split (same figure, A1 is alternative split)

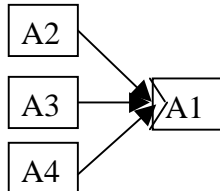
Option 1:

```
Activity(A1,A2,A3,A4)
Temporal(A1,A2,0,inf,ES)
Temporal(A1,A3,0,inf,ES)
Temporal(A1,A4,0,inf,ES)
Logical(A2,A3,A4,XOR)
```

Option 2:

```
Activity(A1,A2,A3,A4)
Decomposition(A5,[A2,A3,A4],XOR)
Temporal(A1,A5,0,inf,ES)
Logical(1,5,=>)
```

Parallel join



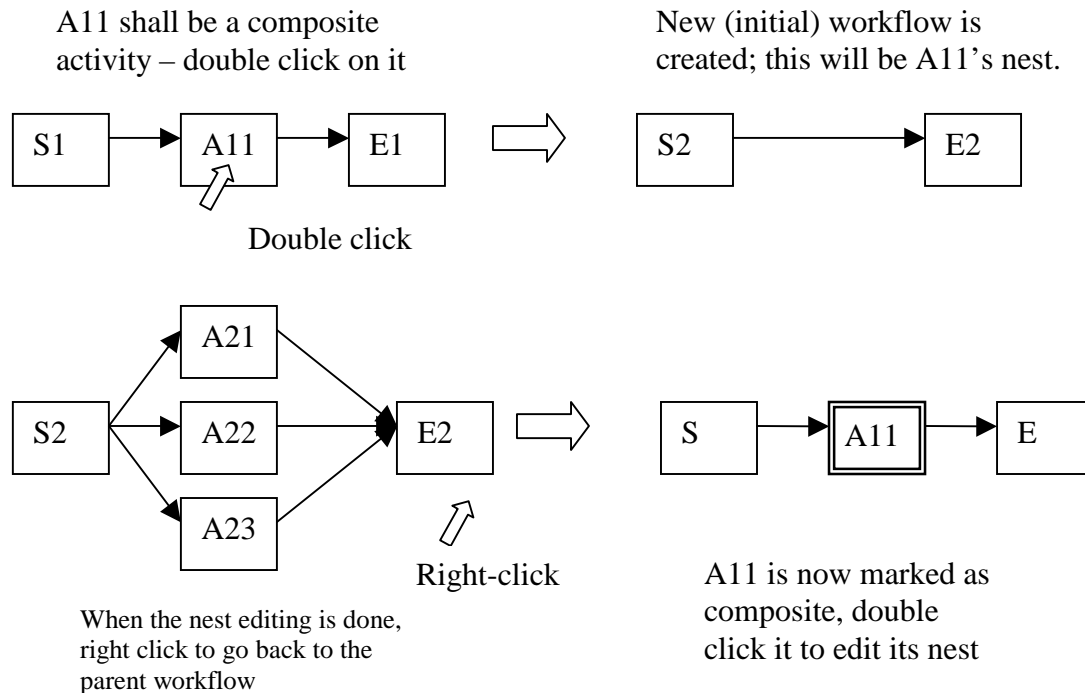
```
Activity(A1,A2,A3,A4)
Decomposition(A5,[A2,A3,A4],AND)
Temporal(A5,A1,0,inf,ES)
Logical(A1,A5,<=>)
```

Alternative join (same figure, A1 is alternative join)

```
Activity(A1,A2,A3,A4)
Decomposition(A5,[A2,A3,A4],XOR)
Temporal(A5,A1,0,inf,ES)
Logical(A1,A5,<=>)
```

Hierarchy

Hierarchy in freelance mode will be different from the Nested TNA mode – since there is no set hierarchy as in Nested TNA model, the user will specify nests manually. That is, any activity may be marked as a composite activity by double clicking on it – thereafter a new initial workflow is created as a nest for this activity and may be freely edited, just like any other workflow in the freelance mode. Subsequently, this newly created nested workflow is executed in place of the composite activity.



Workflow analysis

If the user chooses to use workflow created in freelance mode for scheduling by other modules in FlowOpt, the application will automatically check for basic correctness of the workflow. The user can also invoke this check explicitly.

Basic correctness means, that the workflow is a non-empty (i.e. with at least one activity) connected graph with no cycles. If this check fails, the workflow cannot be used for scheduling.

Calendar

The application will also provide a simple calendar for the user to better visualize the workflow's time span.

Import/Export of the workflow

The user will be able to import/export a workflow from/to the YAWL format (<http://www.yawlfoundation.org/>). Since the YAWL format is significantly more complex than our workflow format, there will be some modifications on the workflow begin imported from YAWL – the user will be notified of these modifications and, in case it is needed, the users feedback will be requested on how should the YAWL workflow be transformed to fit into our workflow format. The main goal here is to create a workflow visually resembling the one being imported from YAWL, not to create an identical workflow in our format (that is impossible due to said differences between our format and YAWL) (TBD).

Table of Contents

| | |
|--|----|
| Workflows editor – SRS | 1 |
| Used terms and notes..... | 1 |
| Main modes | 1 |
| Basics | 2 |
| Zooming | 3 |
| Selection | 3 |
| Nested TNA mode..... | 3 |
| Visualization..... | 3 |
| Choice of orientation | 3 |
| Editing | 4 |
| Description of editing process | 4 |
| Hierarchy | 7 |
| Process description | 7 |
| Manipulating activities | 8 |
| Padding | 9 |
| Freelance mode | 9 |
| Visualization..... | 9 |
| Moving | 9 |
| Selection | 10 |
| Aligning..... | 11 |
| Editing | 11 |
| Actions on selection | 14 |
| Declarative format..... | 14 |
| Sequence..... | 14 |
| Parallel split..... | 14 |
| Alternative split (same figure, A1 is alternative split) | 14 |
| Parallel join | 15 |
| Alternative join (same figure, A1 is alternative join)..... | 15 |
| Hierarchy | 15 |
| Workflow analysis..... | 16 |
| Calendar | 16 |
| Import/Export of the workflow | 16 |