Executable graph models in Bloc

Louis Müller
Software Composition Seminar, fall 2019
Broader vision

Many software models are based on graphs.

➡ Exploring possibilities in GT to interactively construct graphs.
Project goals

Experiment with ways to quickly build graph models using Bloc:

- interactively construct graphs (syntax)
- assign interpretation to graphs (semantics)
- execute graph models
What is Bloc?

**Glamorous Toolkit (GT)** is a moldable development environment written in Pharo Smalltalk.

**Bloc** is the graphical framework delivered with **GT**.
Bloc hierarchy

BlElement is the root class of all visual elements in Bloc
Implementation background

Applied model-view-controller (MVC) design pattern

GraphController

GraphView

Graph
Implementation results

1. Interactive graph editor

2. Export created graph as code
Implementation results

3. Implementation of finite state automata (FSA)

4. Implementation of petri net
Challenges

- Finding a stable GT version
- Limited sources of knowledge about Bloc
Possible future work

• Implement further graph models, i.e. extend syntax & semantics.

• Add algorithms for solving graph problems (e.g. max flow problem).

• Integration into the interactive actor modeling project.
Summary

Project goals
Experiment with ways to quickly build graph models using Bloc:
- interactively construct graphs
- assign interpretation to graphs (semantics)
- execute graph models

Possible future work
- Implement further graph models, i.e. extend syntax & semantics.
- Add algorithms for solving graph problems (e.g. max flow problem).
- Integration into the interactive requirements modeling project.

Implementation results
1. Interactive graph editor (syntax)
2. Export created graph as code

Implementation results
3. Implementation of finite state automata (FSA)
4. Implementation of petri net