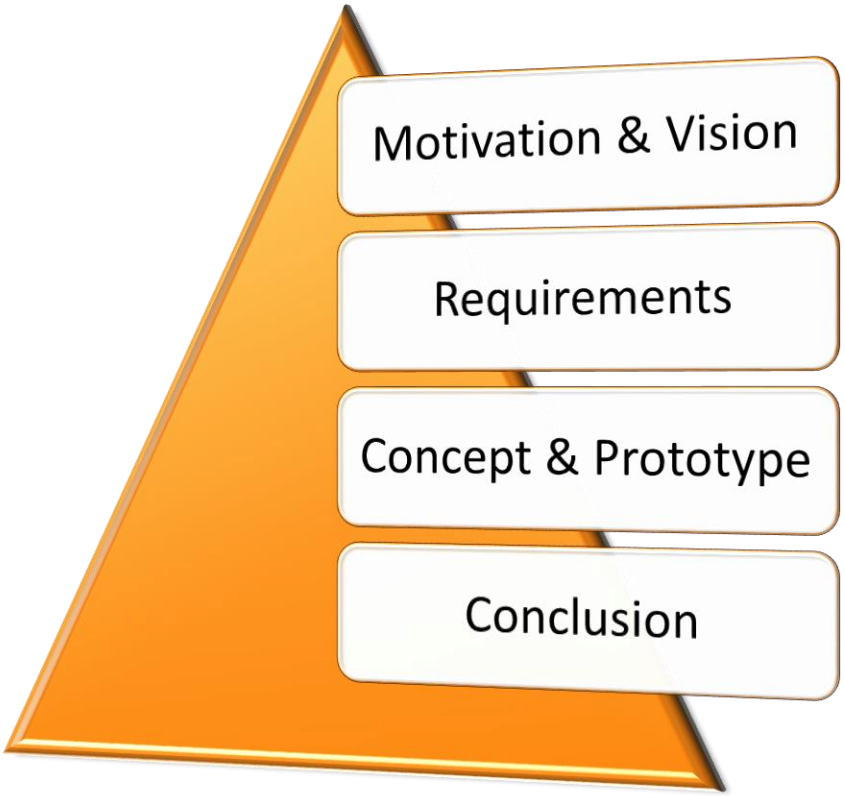
Towards a graphical language for quadrotor missions

Benjamin Schwartz, **Ludwig Nägele**, Andreas Angerer, Bruce A. MacDonald

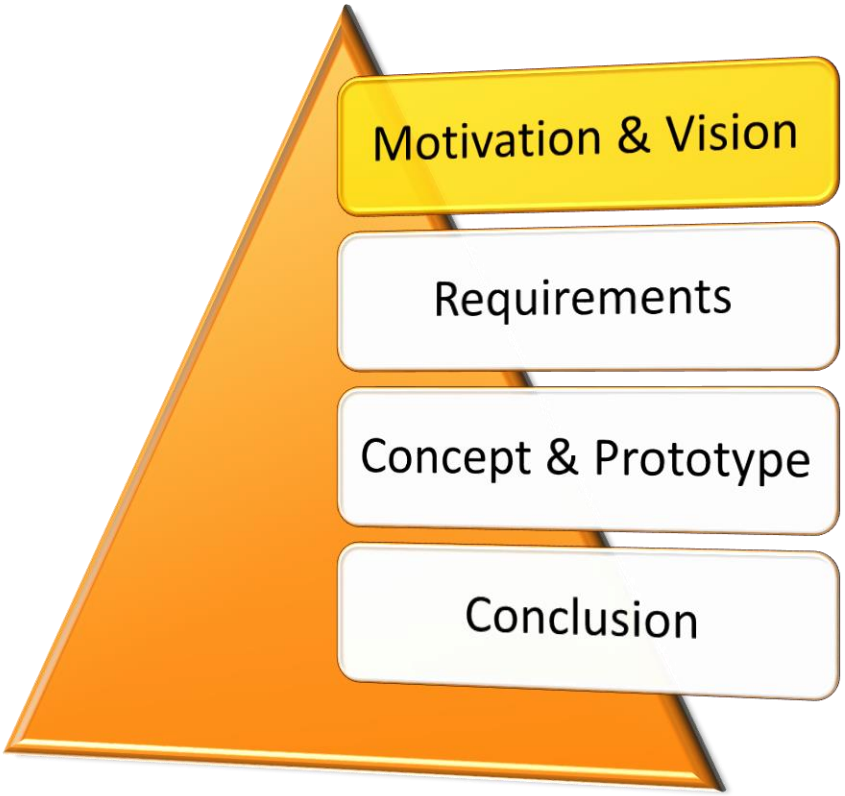
Agenda

Institute for Software & Systems Engineering



Agenda

Institute for Software & Systems Engineering



Intuitive specification of quadrotor missions

Institute for Software & Systems Engineering

Current situation

- Hard-coded UAV missions in general-purpose programming languages
- Increasing number of application domains (agriculture, archaeology, …)
- Graphical end-user solutions with limited functionality

APM Planner

Institute for Software & Systems Engineering

- ✓ Route defined by waypoints
- ✓ Commands specificable for each waypoint
- ⃠ No parallel activities
- ⃠ No branching
- ⃠ No complex movements (e.g. obstacle avoidance)

Intuitive specification of quadrotor missions

Institute for Software & Systems Engineering

Current situation

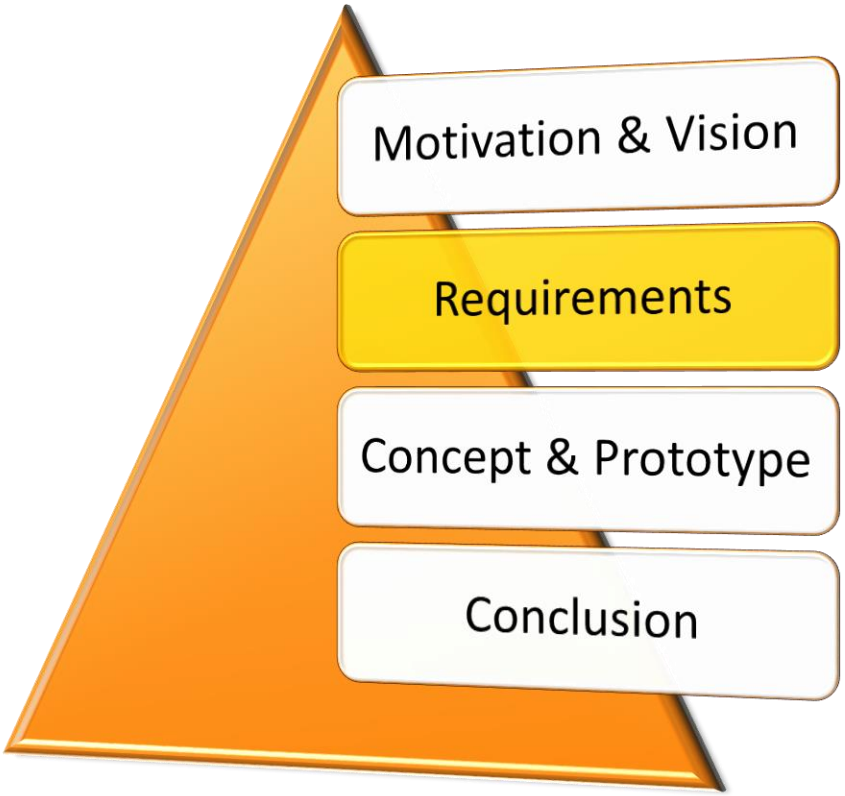
- Hard-coded UAV missions in general-purpose programming languages
- Increasing number of application domains (agriculture, archaeology, …)
- Graphical end-user solutions with limited functionality

Vision

- Intuitive mission definitions by non-programmers
- Powerful interface for programmers
- Hardware-independent specification of missions (or even simulator-independent)

Agenda

Institute for Software & Systems Engineering



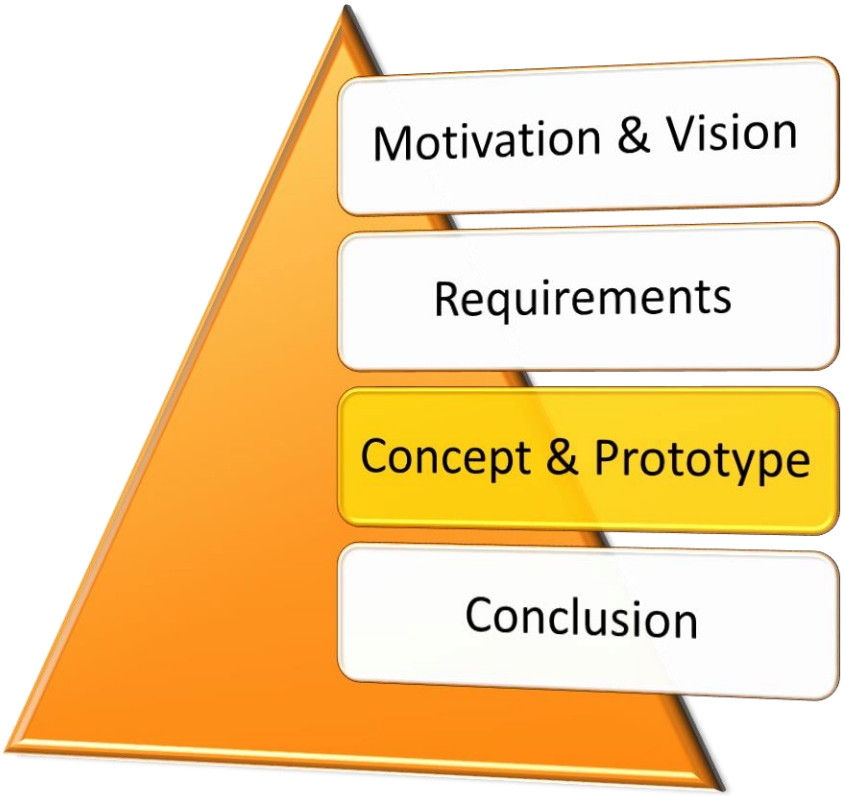
Requirements

Institute for Software & Systems Engineering

- Graphical language for intuitive mission definition by non -programmers
- Clear representation of the main workflow
- Sufficient expressivenes needed for common use cases: Branches, loops, parallel actions
- Concept of extensibility of the language for programming experts
- Extensions in turn reusable by non-programmers
- Hardware-independent: Applicable to different target platforms or simulators (programming languages, frameworks, etc.)

Agenda

Institute for Software & Systems Engineering



First steps

Institute for Software & Systems Engineering

- Specification of a graphical language for quadrotor missions
- Working editor in eclipse for the language (GMF)
- A code generator for MORSE simulator code
  Only partial support yet (evaluation purpose)

Concept

Institute for Software & Systems Engineering

- Separation between:
  => Workflow: Description of waypoints, branching (stable)
  => Actions: Special behaviour (modifyable, extensible)

- Workflow elements are defined by meta-model

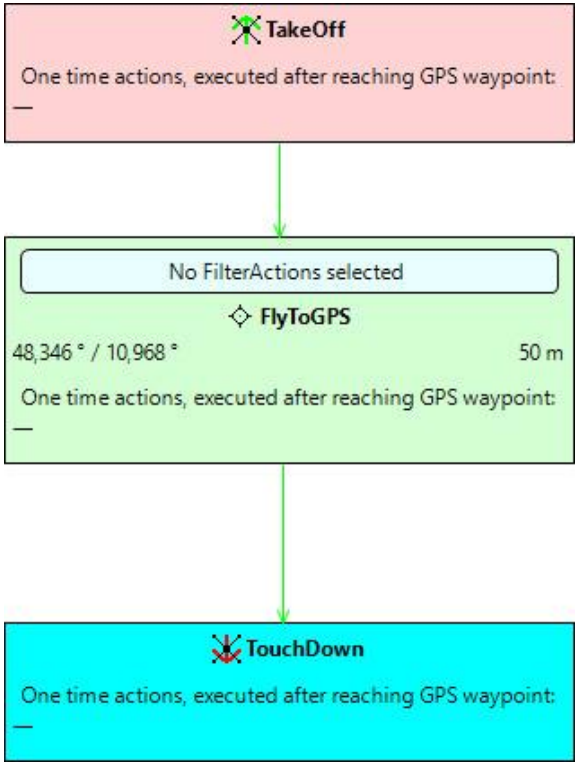
- Action interfaces are defined by meta-model;
  Action implementations are part of model

Concept – language elements

Institute for Software & Systems Engineering

Routing elements

- Basic quadrotor functionality
- Describe the main workflow
- States and arrows



Examples:

TakeOff, FlyTo(GPS), HoldPosition, TouchDown

Concept – language elements

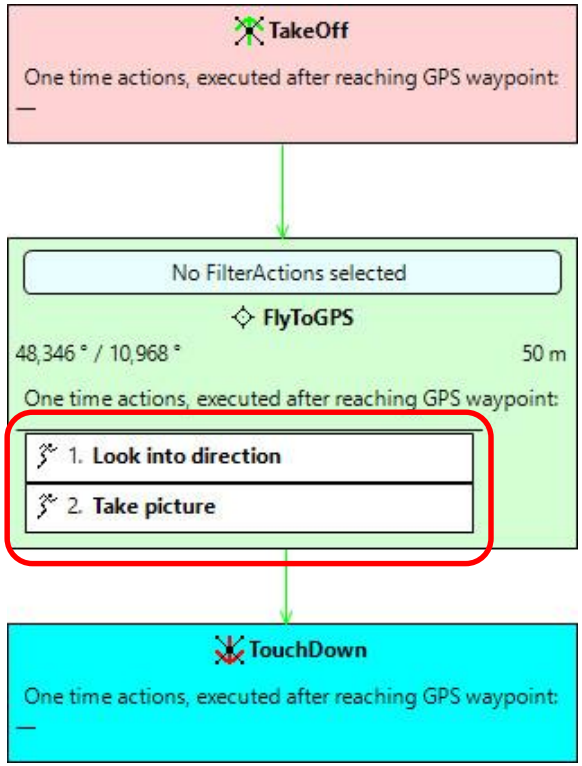
Institute for Software & Systems Engineering

OneTimeActions

- Extended functionality
- Annotated to routing elements
- Executed once

Examples:

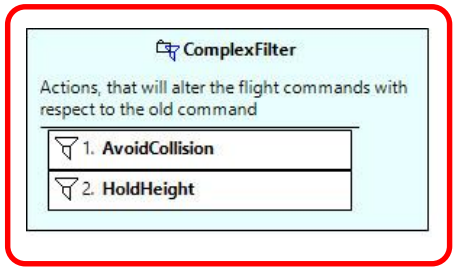
Look into direction, Take picture, …

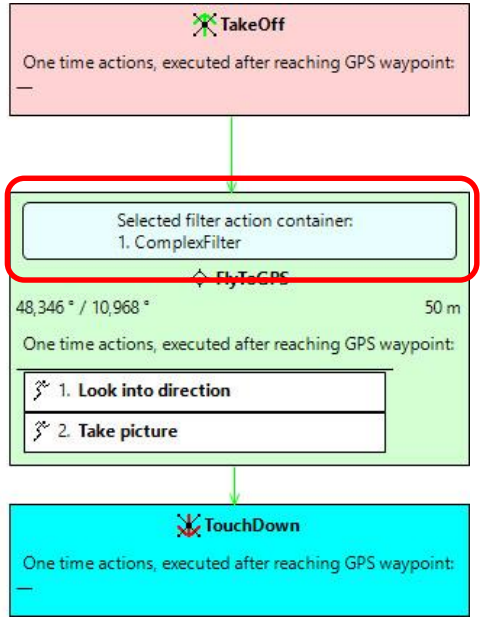


Concept – language elements

Institute for Software & Systems Engineering

Filter elements and FilterActions





- Influence quadrotor flight commands
- Grouping of multiple FilterActions in Filter elements
- Filter elements are annotated to routing elements
- Running synchronously, internal priorisation

Examples: AvoidCollision, HoldHeight, …

Concept – language elements

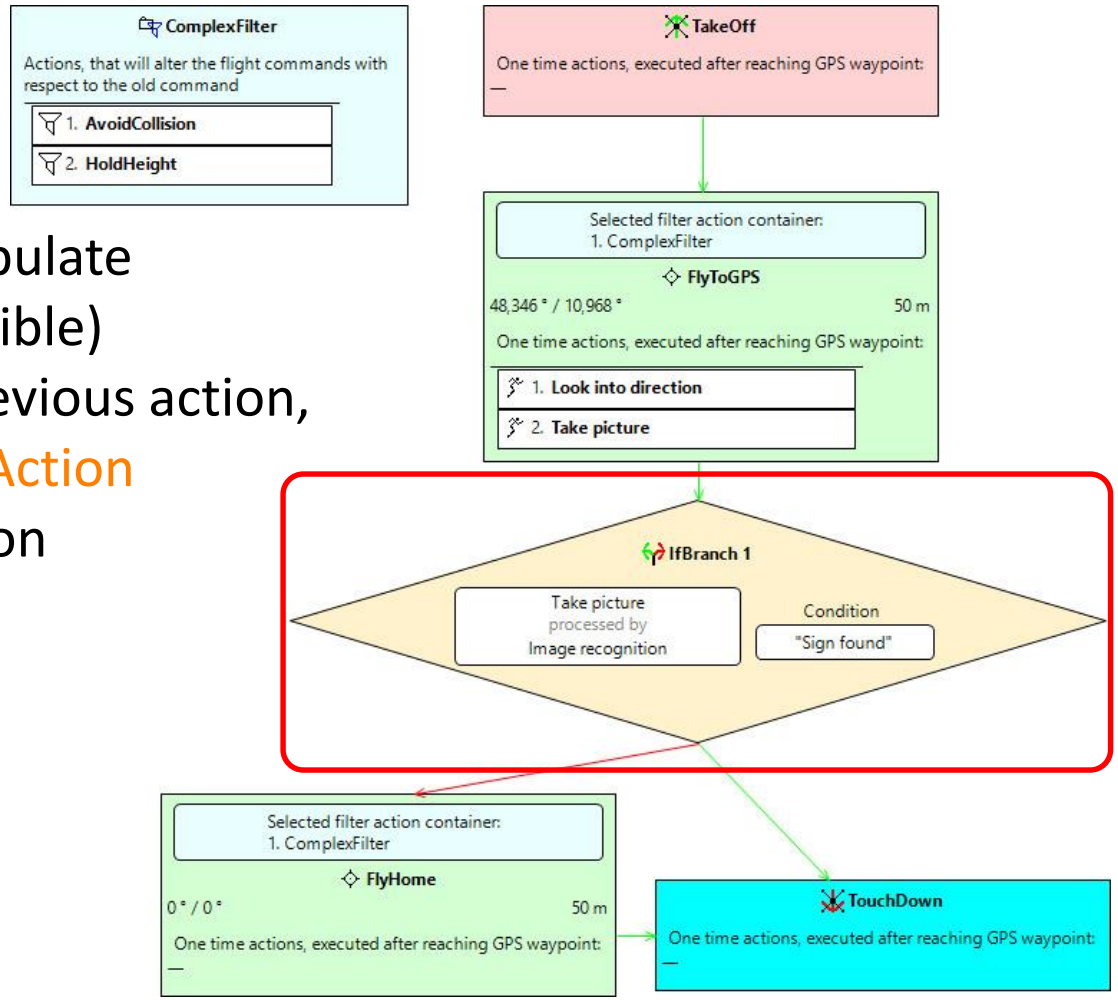
Institute for Software & Systems Engineering

IfBranch elements and ProcessingActions

- IfBranch elements manipulate workflow (loops are possible)
- Reference to result of previous action, processed by ProcessingAction
- Comparison with condition

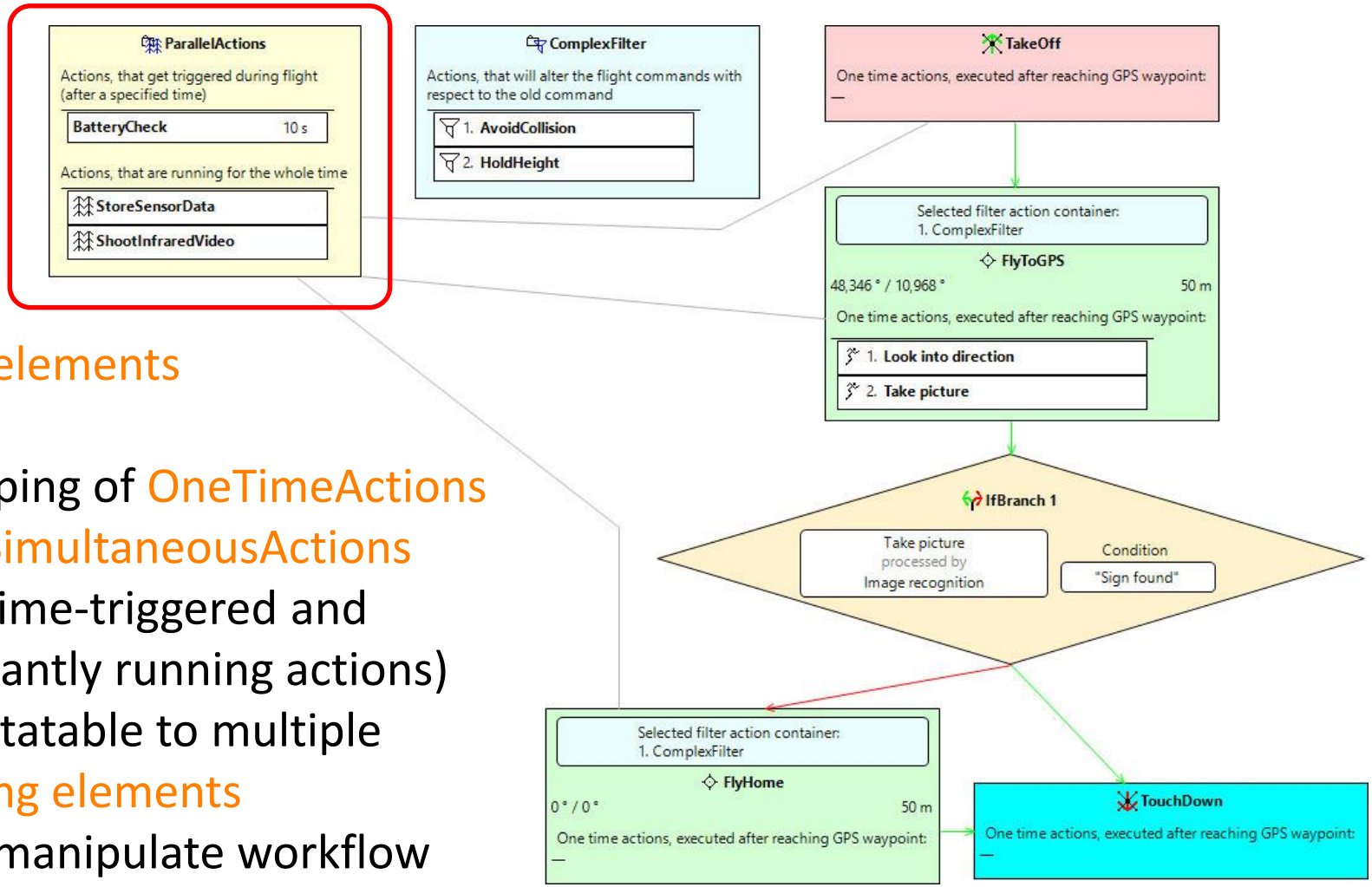
Examples:

Image recognition, …



Concept – language elements

Institute for Software & Systems Engineering

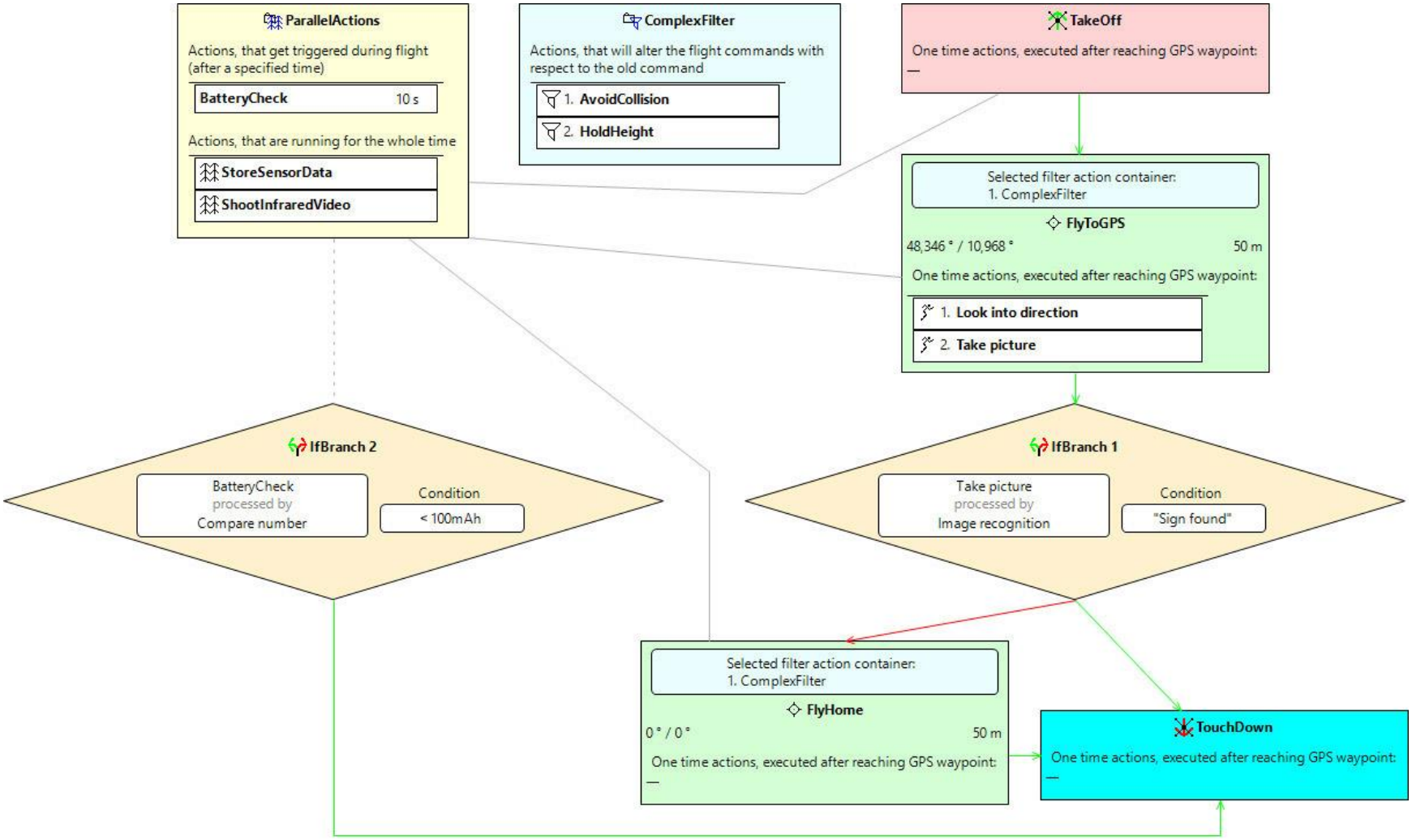


Parallel elements

- Grouping of OneTimeActions and SimultaneousActions (i.e. time-triggered and constantly running actions)
- Annotatable to multiple routing elements
- May manipulate workflow

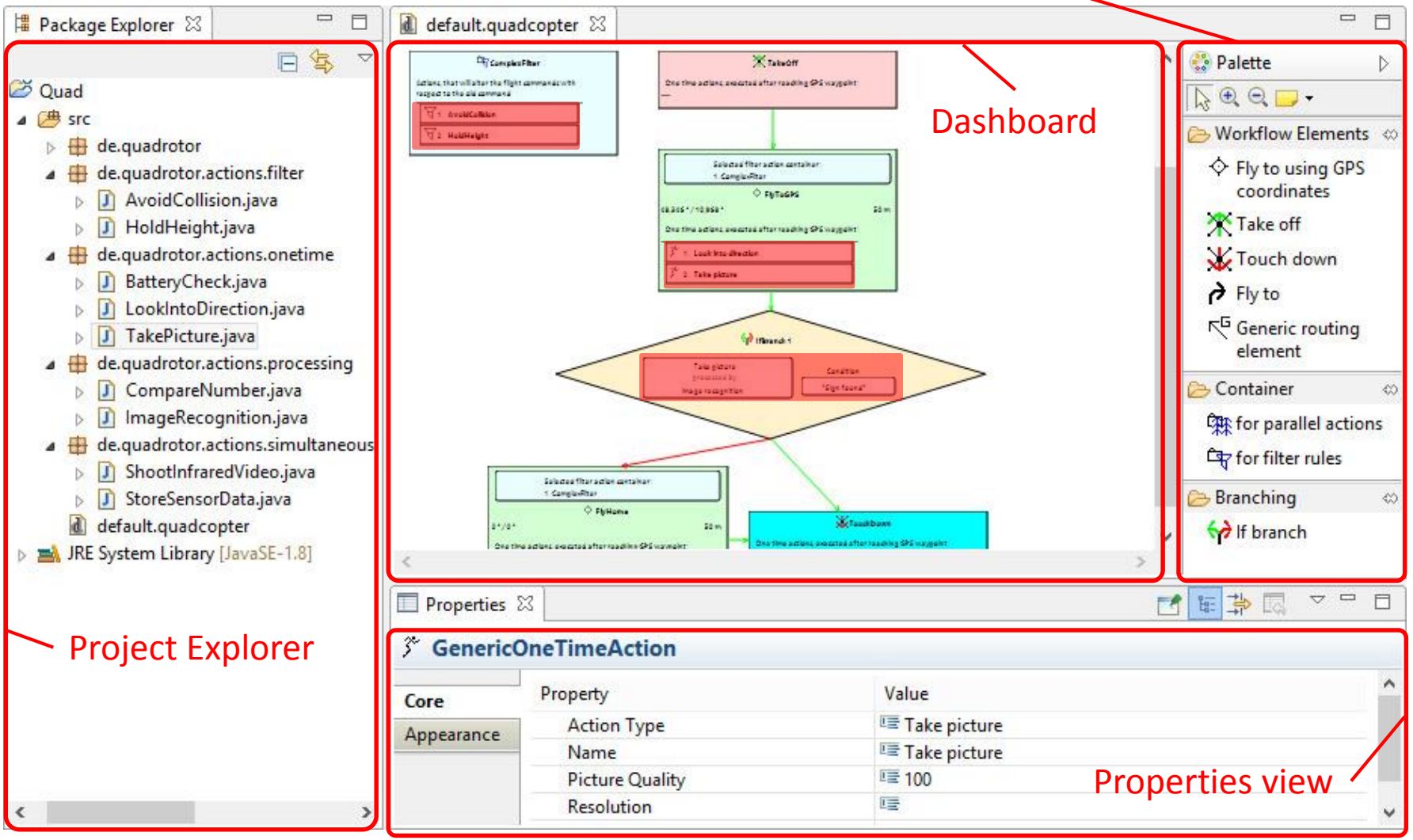
Concept – language elements

Institute for Software & Systems Engineering



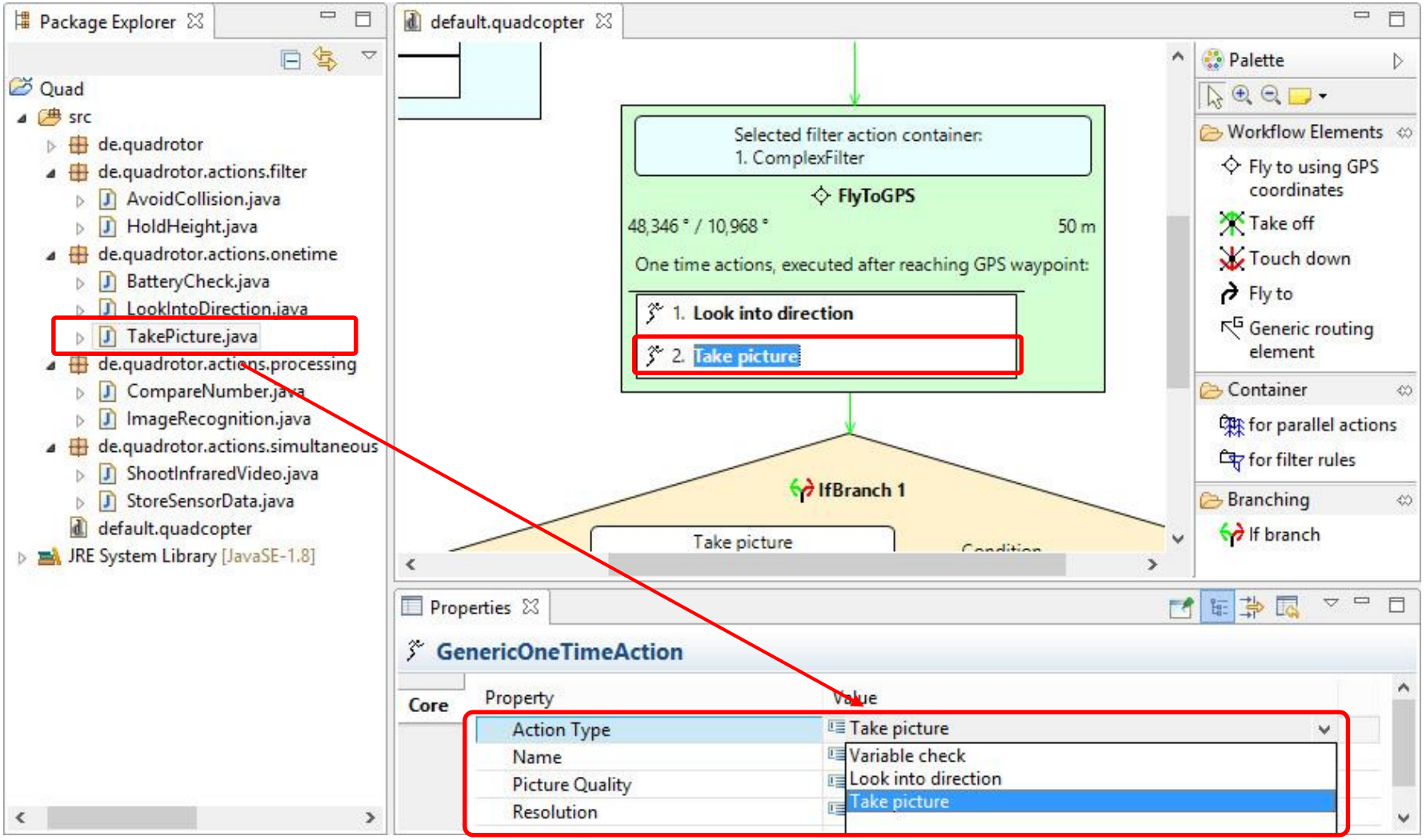
Prototype

Institute for Software & Systems Engineering



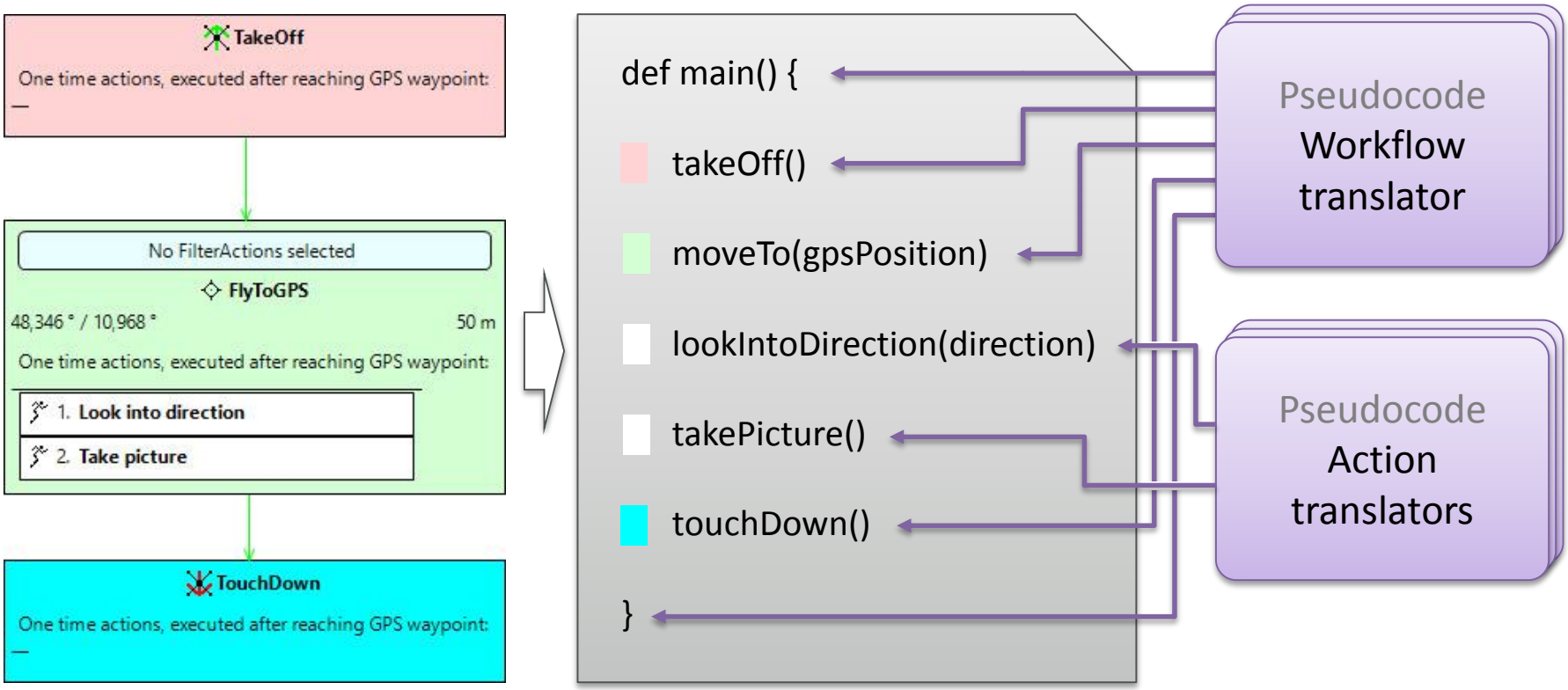
Extensibility by new actions

Institute for Software & Systems Engineering



Code generation

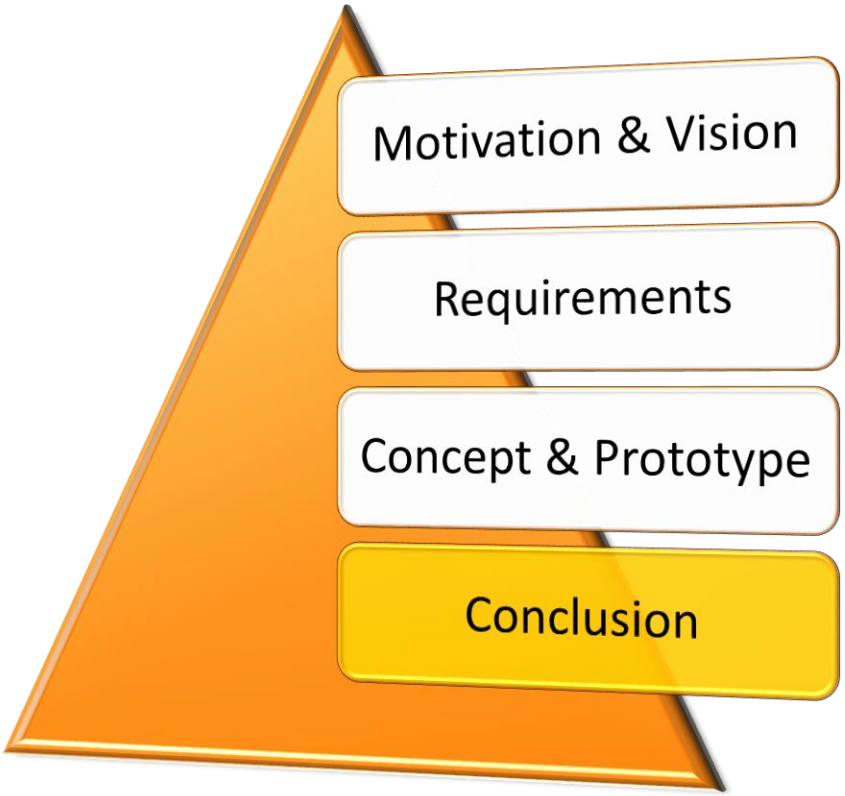
Institute for Software & Systems Engineering



In our scope: Python-codegenerator for MORSE simulator (quadrotor simulation)

Agenda

Institute for Software & Systems Engineering



Conclusion

Institute for Software & Systems Engineering

- Language and editor for graphically specifying quadrotor missions
- Extensible by new (reusable) actions
- Thus, appropriate for skilled programmers and non-programmers
- First code generator to MORSE simulator code showed proof of concept

Future work:

- Distinction between target-platform independend and dependend semantics (limitations on target platforms: no multi-threading etc.)
- Complete evaluation on (different) real hardware


Thank
you