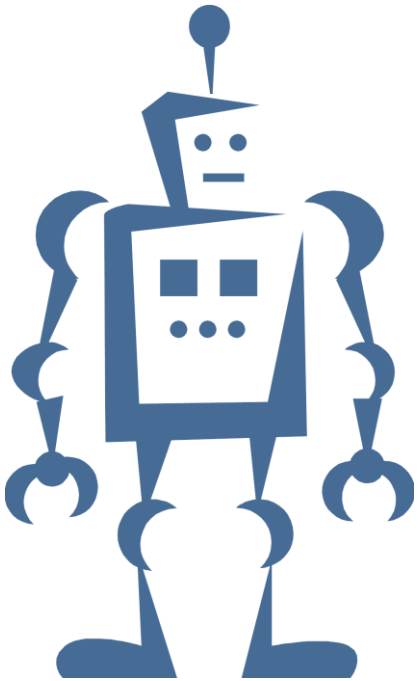
SE SOFTWARE ENGINEERING

RWTH AACHEN UNIVERSITY

Black-box Integration of Heterogeneous Modeling Languages for Cyber-Physical Systems

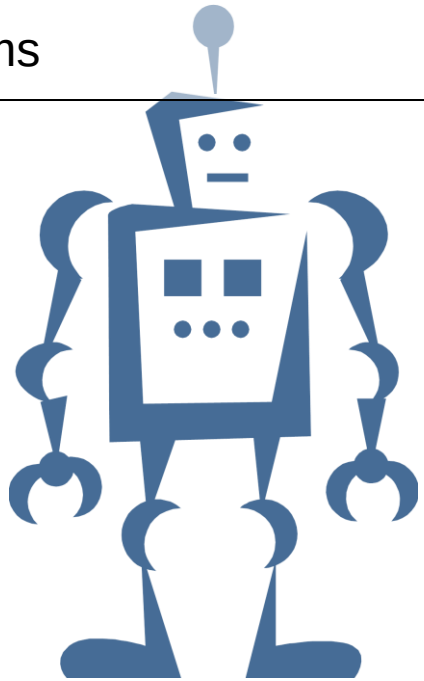


Markus Look, Antonio Navarro Perez,
Jan Oliver Ringert, Bernhard Rumpe,
and Andreas Wortmann
Software Engineering
RWTH Aachen

http://www.se-rwth.de/

Content

1. Motivation, MontiArcAutomaton, MontiCore
2. Language Integration Mechanisms

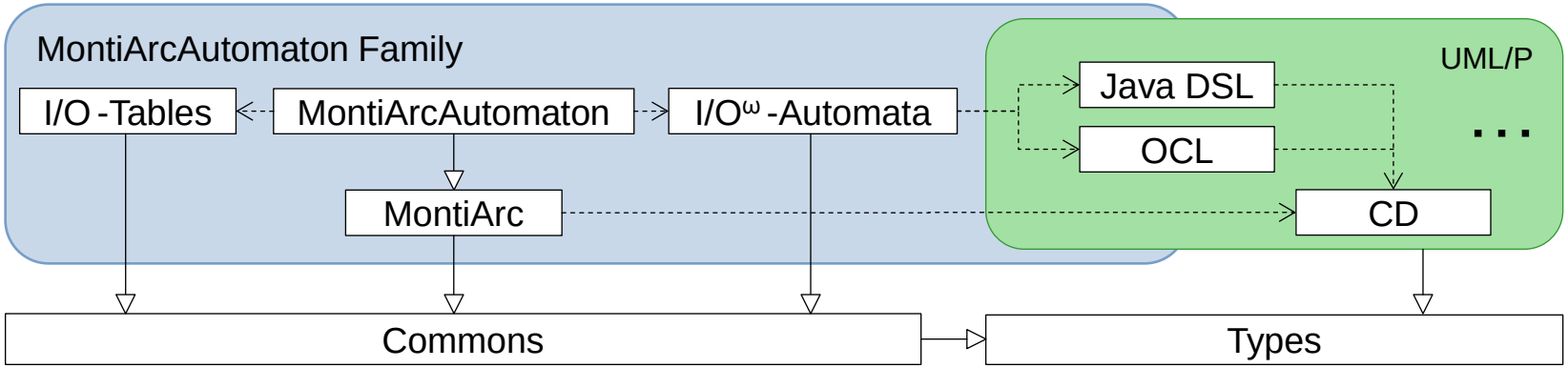


Motivation

MontiArc Automaton

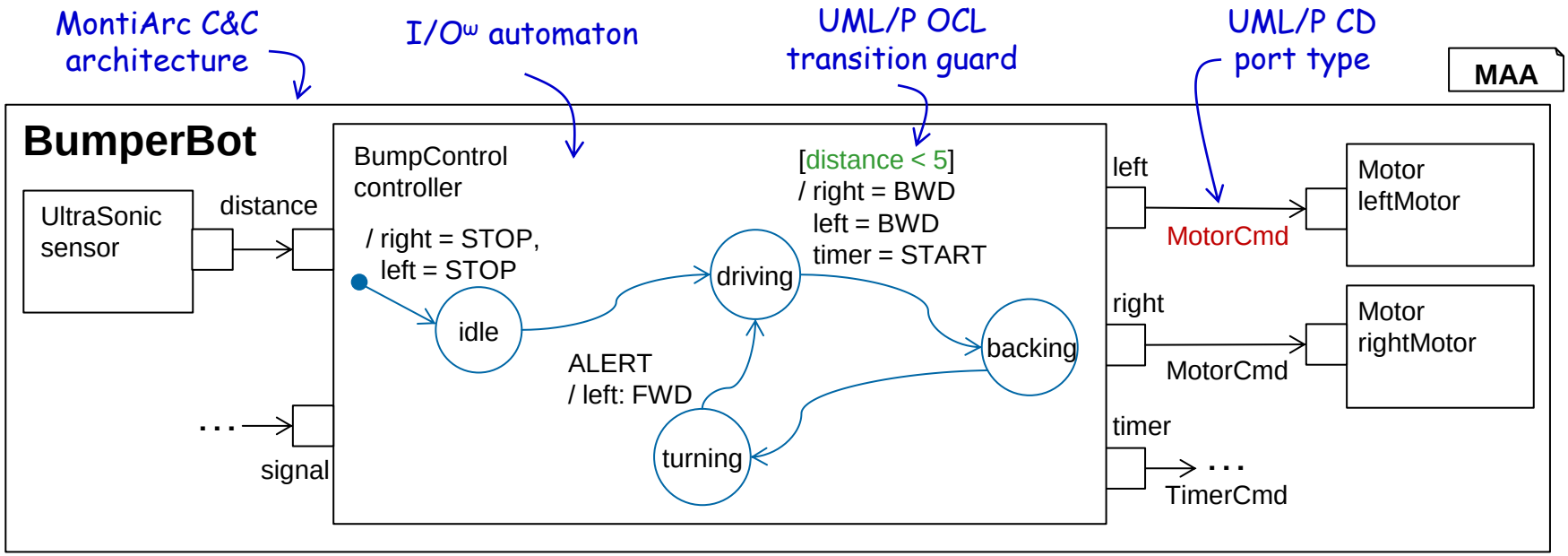
- Highly heterogeneous domain
  - different domain challenges
  - little re-use and portability
  - ➢ Modeling

- MontiArcAutomaton modeling language family and framework
  - MontiArc C&C ADL + problem specific behavior languages



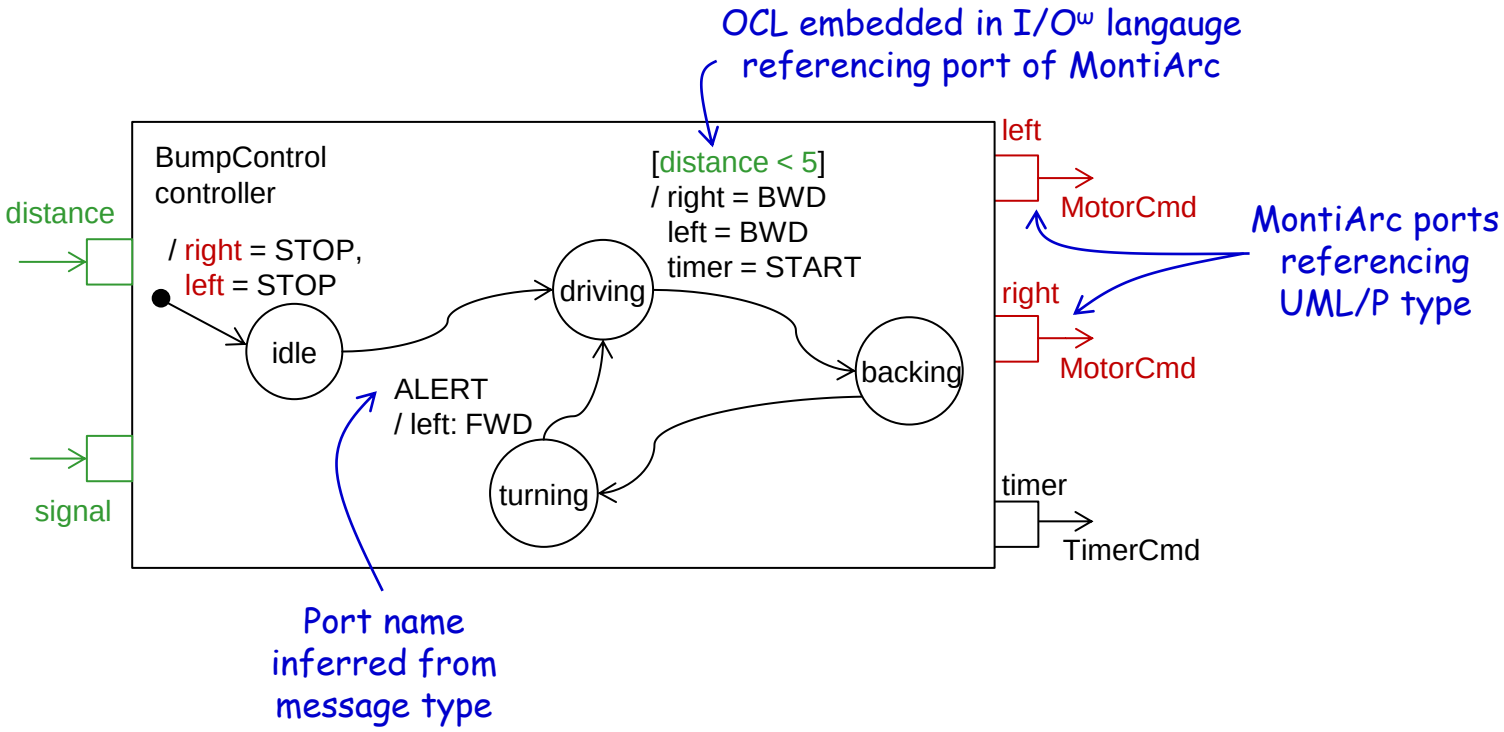
MontiArcAutomaton Example

- Combines MontiArc
  - with I/O$^{\omega}$ automata
  - UML/P class diagrams for port types, and
  - UML/P OCL for transition guards



Challenges

- How does the embedded automaton know about the types of messages on outgoing ports?
- How to type check embedded OCL guard?

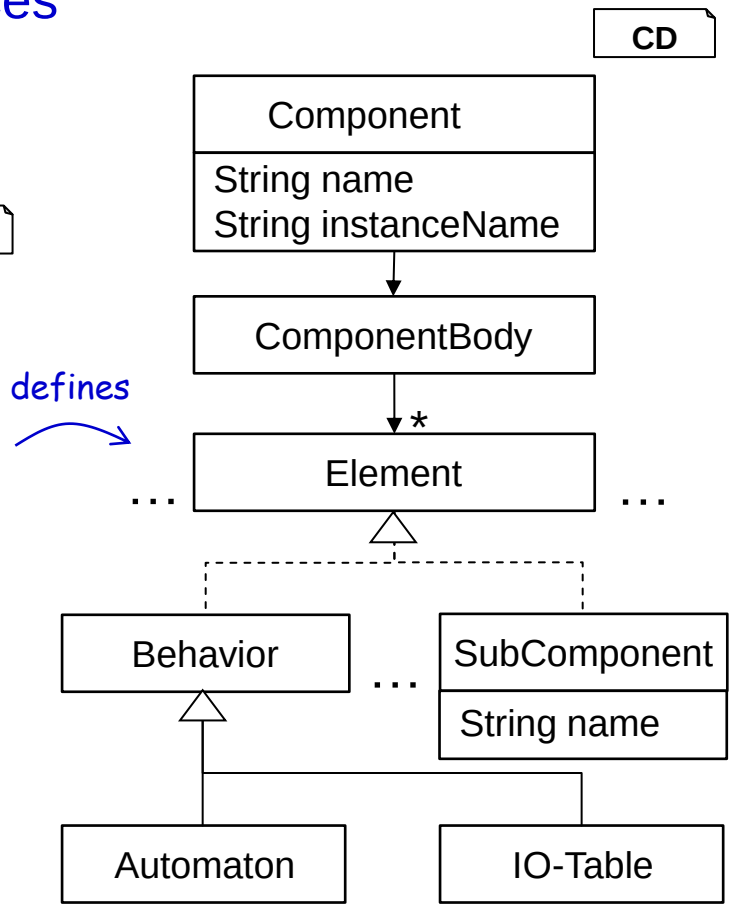


MontiCore

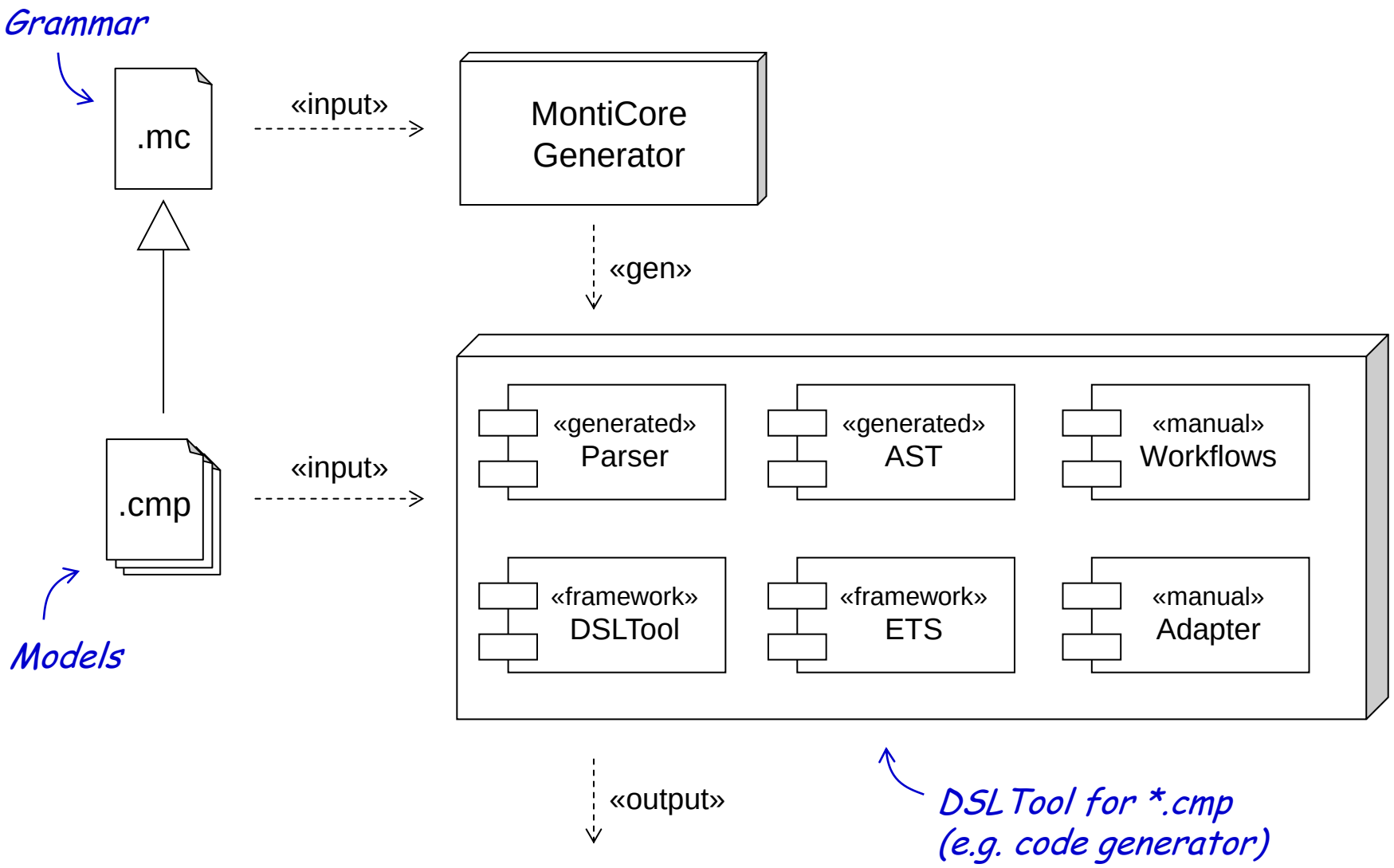
- Language and framework for modeling languages
- Generates parsers for CFGs
- Textual models represent AST instances

MCG

```
...
  Component implements Element =
    "component" Name (instanceName:Name)?
    ComponentHead ComponentBody;

  ComponentBody = "{" Element* "}";

  SubComponent implements Element =
    "component" ReferenceType Name;

  external Behavior implements Element;
...
```

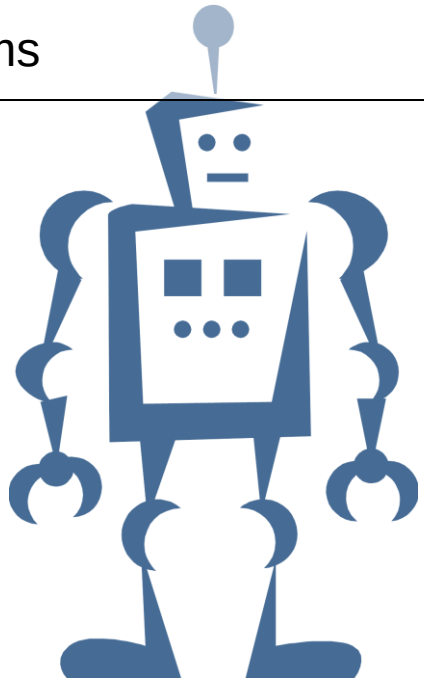


MontiCore Framework Components



Content

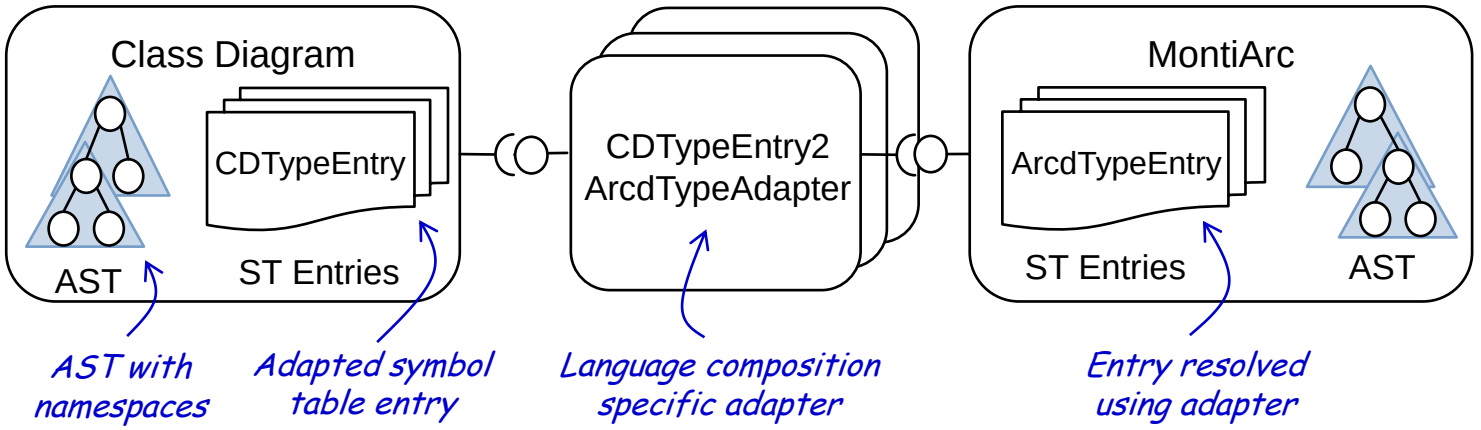
1. Motivation, MontiArcAutomaton, MontiCore
2. Language Integration Mechanisms



Extensible Type System (ETS)

- Compositional black-box language integration
- Inter-language adaptation of concepts

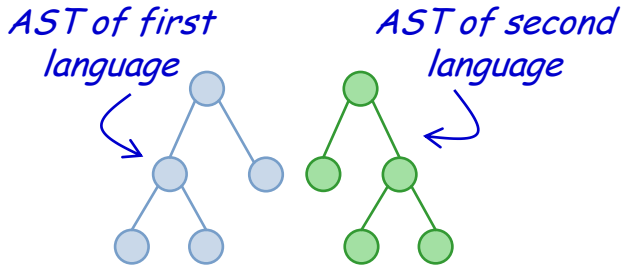
- Core components
  - entries: model elements
  - symbol tables: provide entries
  - resolvers: look up entries



- Manual implementation of core components

Language Aggregration

- Model orthogonal aspects
- Composite language families
- Also known as language unification



MontiArc

```
import nxt.Types.MotorCmd;

component Motor {
    port
        in MotorCmd cmd;
}
```

*.arc

References

CD

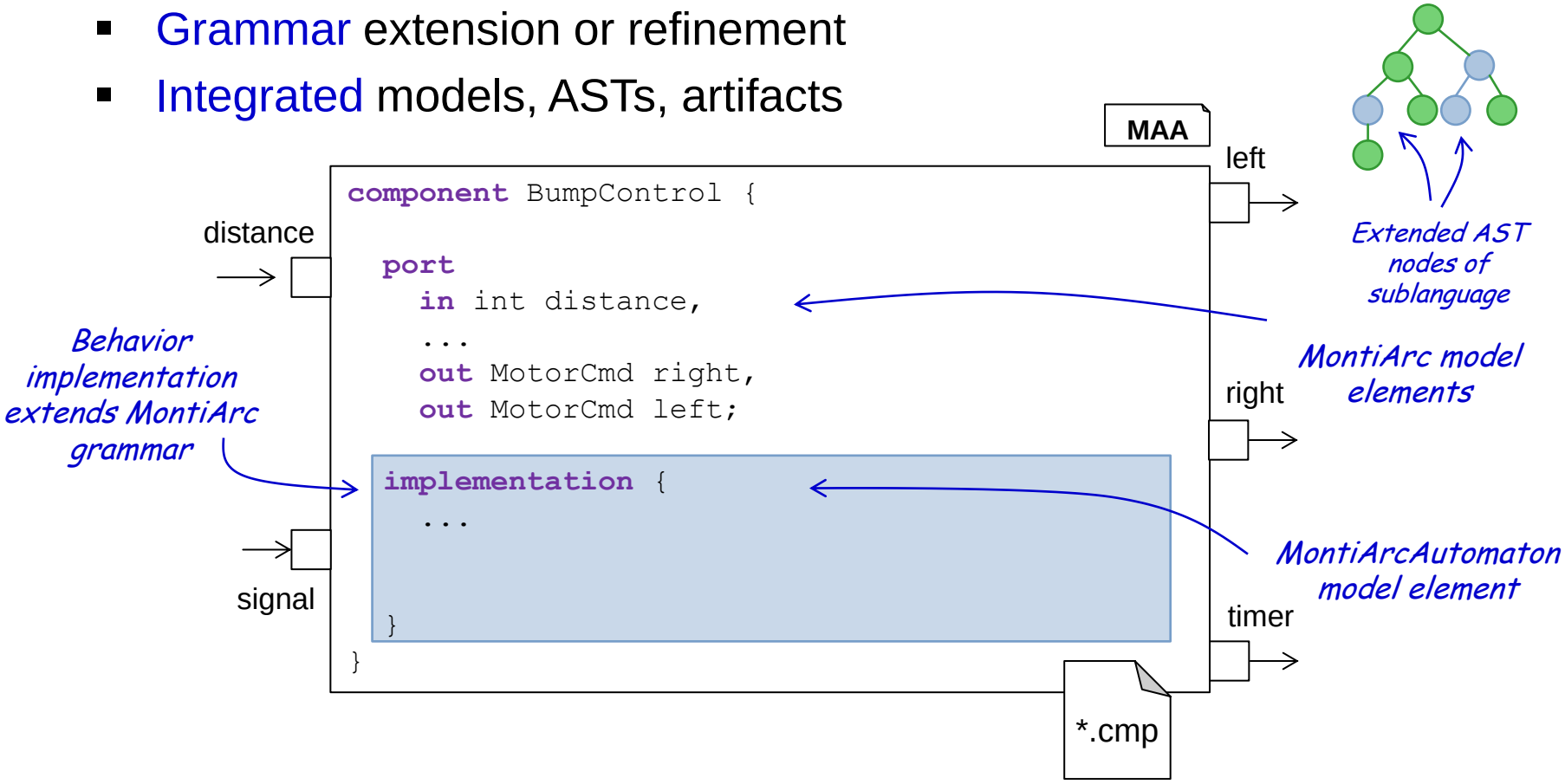
```
classdiagram Types {

    enum MotorCmd {
        FWD, BWD, STOP;
    }
}
```

*.cd

- Write ETS adapters
- Reuse grammars, entries, STs, resolvers, CoCos, …
- Minimal coupling

Language Inheritance

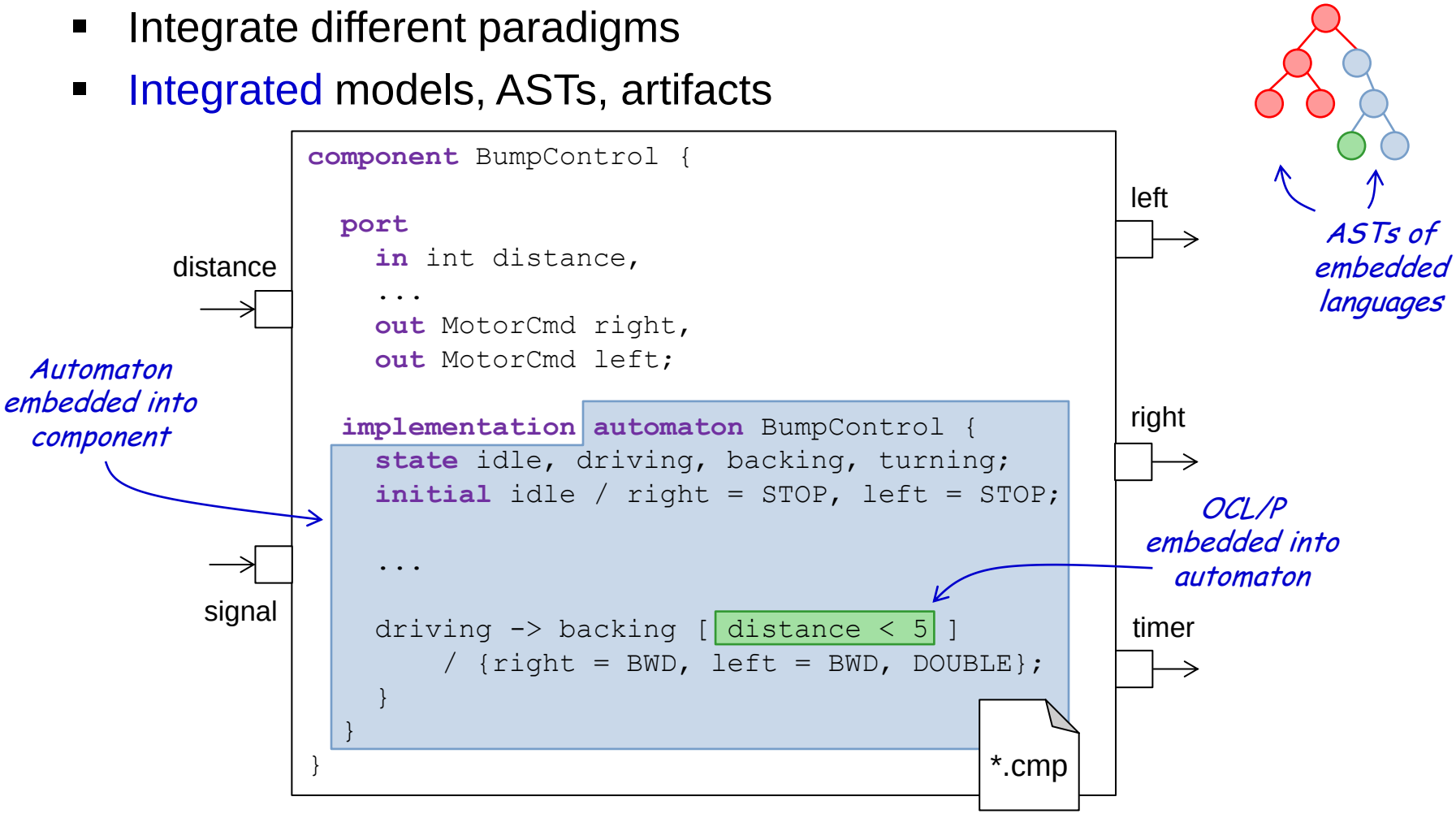
- Grammar extension or refinement
- Integrated models, ASTs, artifacts



- Reuse ETS of parent language
- Extend parent entries and/or add new ones

Language Embedding

- Integrate different paradigms
- Integrated models, ASTs, artifacts



- Write ETS adapters
- Reuse grammars, entries, STs, resolvers, CoCos, …

Conclusion

- Engineering of complex heterogeneous systems requires integration of multiple heterogeneous modeling languages
- Integration of heterogeneous modeling languages works
- MontiCore language integration mechanisms helps
- ETS artifacts support a posteriori language integration via
  - language aggegration, inheritance, embedding
- Future work:
  - generation of ETS elements
  - identification of integration sub-patterns

Thank you for your attention!