Probabilistic modelling and verification using RoboChart and PRISM

Kangfeng Ye, Ana Cavalcanti, Simon Foster, Alvaro Miyazawa, Jim Woodcock

ROBOSTAR

robostar.cs.york.ac.uk

October 27, 2022

UNIVERSITY of York

Engineering and Physical Sciences Research Council

Outline

Background and motivations

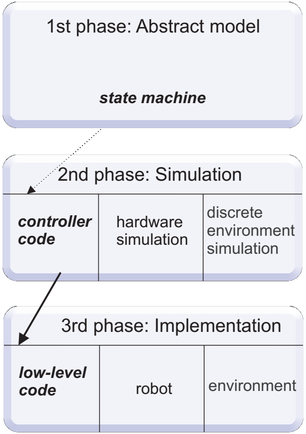
Probabilistic modelling and property language

Automated verification in RoboTool

Probabilistic semantics in PRISM

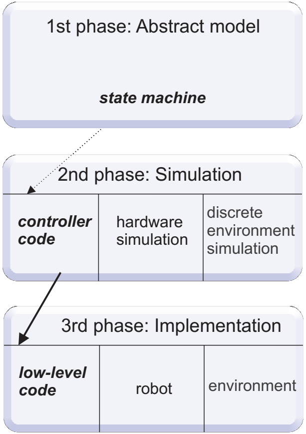
Conclusion

Current practice and problems



ROBOSTAR

Current practice and problems



- No models, or models without precise syntax or formal semantics,
- Time and uncertainty: discussed informally,
- No tool support,
- Loose connections of artefacts,
- Trial-and-error,
- No assurance.

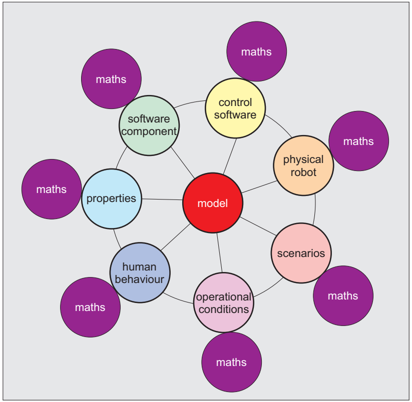
ROBOSTAR

RoboStar

- RoboStar framework: modern modelling and verification technologies, software engineering of robotics

RoboStar

- RoboStar framework: modern modelling and verification technologies, software engineering of robotics
- Vision: model centred, mathematical semantics



ROBOSTAR

RoboChart

- Core notation of RoboStar, DSL for robotics, state machines

RoboChart

- Core notation of RoboStar, DSL for robotics, state machines
- A component model (platform independent + parallel composition of state machines)

RoboChart

- Core notation of RoboStar, DSL for robotics, state machines
- A component model (platform independent + parallel composition of state machines)
- Previous work: modelling and verification of RoboChart with time

RoboChart

- Core notation of RoboStar, DSL for robotics, state machines
- A component model (platform independent + parallel composition of state machines)
- Previous work: modelling and verification of RoboChart with time
- but not uncertainty

Uncertainty in robotics

Unpredictable environment, sensors, actuators, model errors, and control algorithmic approximations (EKF SLAM, swarm robots).

Uncertainty in robotics

Unpredictable environment, sensors, actuators, model errors, and control algorithmic approximations (EKF SLAM, swarm robots).

> *"Managing uncertainty is possibly the most important step towards robust real-world robot systems."*
>
> — *Sebastian Thrun et al.*

ROBOSTAR

Uncertainty in robotics

Unpredictable environment, sensors, actuators, model errors, and control algorithmic approximations (EKF SLAM, swarm robots).

> *"Managing uncertainty is possibly the most important step towards robust real-world robot systems."*
>
> — *Sebastian Thrun et al.*

In RoboChart, we use probabilism to model uncertainty.

ROBOSTAR

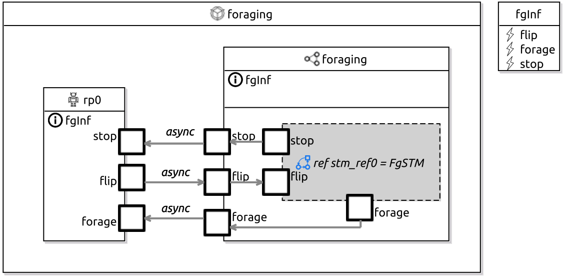
Novel contributions

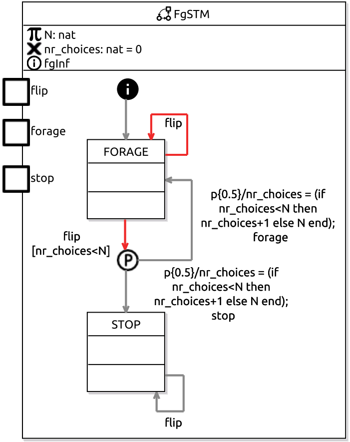
- Extension of RoboChart with probabilistic junctions,
- RoboChart's probabilistic semantics: given in PRISM,
- A metamodel for PRISM,
- A probabilistic property language,
- Implementation in RoboTool for automated verification,

ROBOSTAR

Probabilistic Modelling

A simple foraging robot with a randomising device, every time step (flip), limited number of choices (N)





Probabilistic Property Language

PRISM's property language (PCTL*) enriched with RoboChart elements

- Computation Tree Logic (CTL)

```
prob property P_deadlock_free:
  not Exists [ Finally deadlock]
  with constant N from set {2 to 20 by step 2}
```

Deadlock freedom for various values of N

Probabilistic Property Language

PRISM's property language (PCTL*) enriched with RoboChart elements

- Computation Tree Logic (CTL)
- Linear Temporal Logic (LTL)

```
prob property P_1:
   Forall [Globally ( Finally (fd==2) and (Next (fd==0)))]
```

For all paths, always eventually fd is 2 and fd is 0 immediately afterwards.

ROBOSTAR

Probabilistic Property Language

PRISM's property language (PCTL*) enriched with RoboChart elements

- Computation Tree Logic (CTL)
- Linear Temporal Logic (LTL)
- Probabilistic CTL (quantitative)

```
prob property P_min_terminate:
  Prob min=? of [Finally  FgSTM is in STOP]
  with  constant N from set {2  to  20 by step  2}

prob property P_max_terminate:
  Prob max=? of [Finally FgSTM is in STOP]
  with  constant N from set {2  to  20 by step  2}
```

What's the minimum and maximum probabilities of FgSTM finally in state STOP?

Probabilistic Property Language

PRISM's property language (PCTL*) enriched with RoboChart elements

- Computation Tree Logic (CTL)
- Linear Temporal Logic (LTL)
- Probabilistic CTL (quantitative)
- Rewards/costs

```
rewards nr_of_forages =
  [forage.out] true : 1;
endrewards

prob property R_max_stop:
  Reward {nr_of_forages} max=? of [Reachable FgSTM is in
        STOP]
 with constant N set to 10
```

Each synchronisation on event forage costs 1, and what's the maximum expectation of the cost when FgSTM reaches STOP, considering N is 10.

ROBOSTAR

Probabilistic Property Language

PRISM's property language (PCTL*) enriched with RoboChart elements

- Computation Tree Logic (CTL)
- Linear Temporal Logic (LTL)
- Probabilistic CTL (quantitative)
- Rewards/costs
- Simulations

```
prob property P_max_terminate_sim:
  Prob max=? of [Finally FgSTM is in STOP]
  using sim with CI at alpha=0.01, n=1000, and pathlen=10000
  with constant N from set {2 to 20 by step 2}
```

What's the maximum probabilities of FgSTM finally in state STOP using statistic model checking with method CI (confidence interval)?

ROBOSTAR

Automated verification in RoboTool

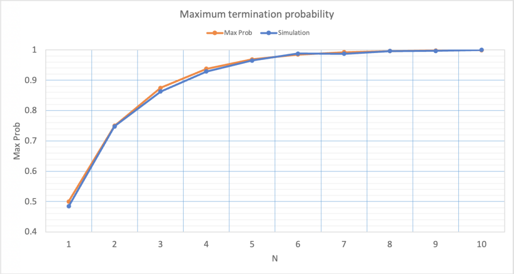
Results of probabilistic analysis of assertions in simulation.assertions using PRISM

Assertion: P_max_terminate

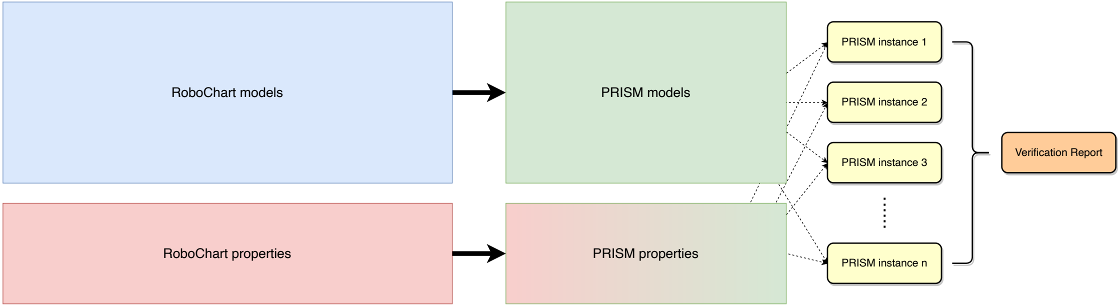
Assertion	Const	states:	transitions:	result:	checkTime:
P_max_terminate	foraging::foraging::stm_ref0::N=2	16	20	0.75	0.004 seconds
P_max_terminate	foraging::foraging::stm_ref0::N=4	30	38	0.9375	0.007 seconds
P_max_terminate	foraging::foraging::stm_ref0::N=6	44	56	0.984375	0.011 seconds
P_max_terminate	foraging::foraging::stm_ref0::N=8	58	74	0.99609375	0.008 seconds
P_max_terminate	foraging::foraging::stm_ref0::N=10	72	92	0.9990234375	0.012 seconds
P_max_terminate	foraging::foraging::stm_ref0::N=12	86	110	0.999755859375	0.013 seconds

Assertion: P_max_terminate_sim

Assertion	Const	states:	transitions:	result:	checkTime:
P_max_terminate_sim	foraging::foraging::stm_ref0::N=2			0.738	
P_max_terminate_sim	foraging::foraging::stm_ref0::N=4			0.947	
P_max_terminate_sim	foraging::foraging::stm_ref0::N=6			0.977	
P_max_terminate_sim	foraging::foraging::stm_ref0::N=8			0.995	
P_max_terminate_sim	foraging::foraging::stm_ref0::N=10			1.0	
P_max_terminate_sim	foraging::foraging::stm_ref0::N=12			1.0	

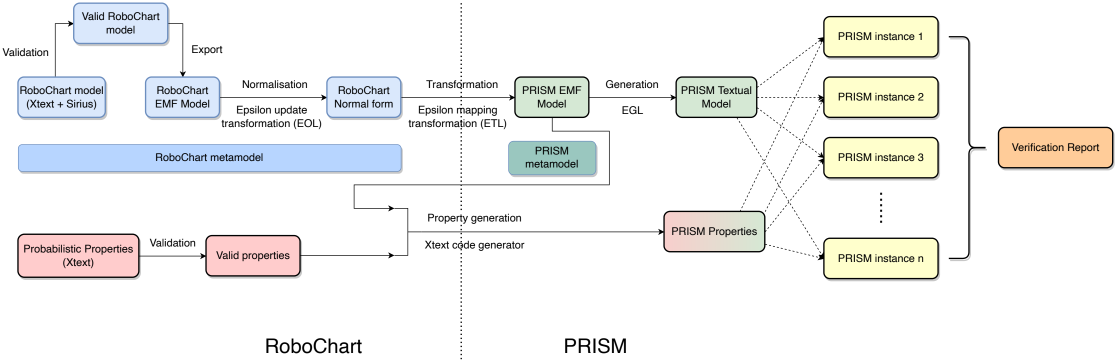


Automated verification in RoboTool - approach



Modelling techniques and mathematical semantics

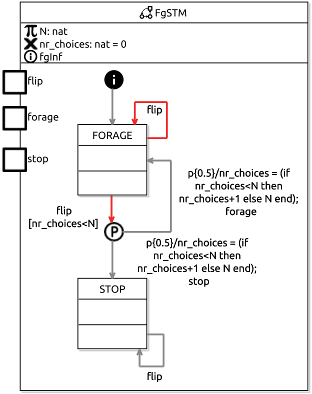
Automated verification in RoboTool - approach



Modelling techniques and mathematical semantics

Probabilistic Semantics

- Nondeterministic choice resolved at states,
- Transitions that exit states implicitly or explicitly enter probabilistic choices,
- Given as DTMC and MDP,
- Defined by a formal translation to PRISM: for verification.



Translation

Challenges

1. Component model,
2. State machines and composite states,
3. Transitions and actions,
4. Communication: input/output triggers and actions,
5. Operations, asynchronous communication, during actions, . . .

Translation

Challenges

1. Component model,
2. State machines and composite states,
3. Transitions and actions,
4. Communication: input/output triggers and actions,
5. Operations, asynchronous communication, during actions, . . .

Solutions

two-stage translation and its Formalisation:

ROBOSTAR

Translation

Challenges

1. Component model,
2. State machines and composite states,
3. Transitions and actions,
4. Communication: input/output triggers and actions,
5. Operations, asynchronous communication, during actions, . . .

Solutions

two-stage translation and its Formalisation:

- normalisation of RoboChart, and

ROBOSTAR

Translation

Challenges

1. Component model,
2. State machines and composite states,
3. Transitions and actions,
4. Communication: input/output triggers and actions,
5. Operations, asynchronous communication, during actions, . . .

Solutions

two-stage translation and its Formalisation:

- normalisation of RoboChart, and
- transformation to PRISM.

Formalisation

Rules: functions and Z notation.

$$\llbracket x : Tx, y : Ty, z : Tz, \cdots \rrbracket_S : T_1 \times T_2 \times \cdots$$

$$(e_1, e_2, \cdots)$$

where

$$e_1 \triangleq \cdots$$

$$e_2 \triangleq \cdots$$

ROBOSTAR

Formalisation

Rules: functions and Z notation.

$$\llbracket x : Tx, y : Ty, z : Tz, \cdots \rrbracket_S : T_1 \times T_2 \times \cdots$$
$$(e_1, e_2, \cdots)$$

where

$$e_1 \triangleq \cdots$$
$$e_2 \triangleq \cdots$$

Rule 32. Probabilistic junction (transitions)

$$\llbracket \begin{array}{l} n : \mathsf{ProbJunc}, cs : \mathsf{NodeContainer}, stm : \mathsf{StateMachineDef}, exit : \mathsf{VarDecl}_{pr}, \\ scpcname : \mathsf{Name}, pcconstrs : \mathbb{P}\,\mathsf{BoolExpr}_{pr}, stnumber : int, trnumber : int \end{array} \rrbracket_{\mathcal{TN}prob}$$
$$: int \times int \times \mathbb{P}\,\mathsf{Constant} \times \mathbb{P}\,\mathsf{Command} =$$
$$(transret.1, transret.2, transret.3, \{cmd\} \cup transret.5)$$

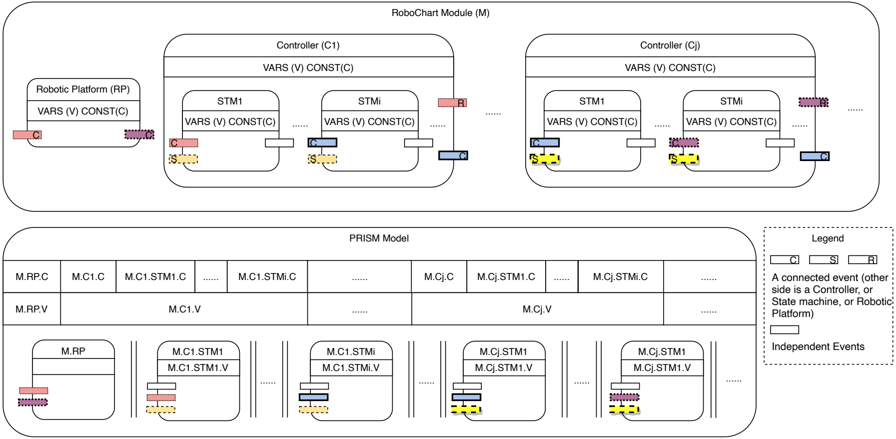
where

$$trans = \{t : cs.transitions \mid t.source = n\}$$
$$transret = \llbracket trans, n, stm, exit, scpcname, pcconstrs, stnumber, trnumber \rrbracket_{\mathcal{TS}^1}$$
$$cmd = [\,] \left(andExprs^6(pcconstrs) \,\&\, \left(scpcname = id(n) \right) \right) \rightarrow transret.4;$$

12 normalisation rules and 84 transformation rules (RoboChart reference manual)

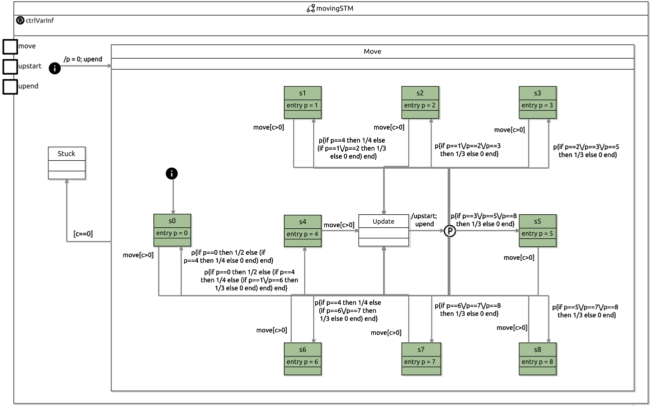
ROBOSTAR

Translation: ① component model



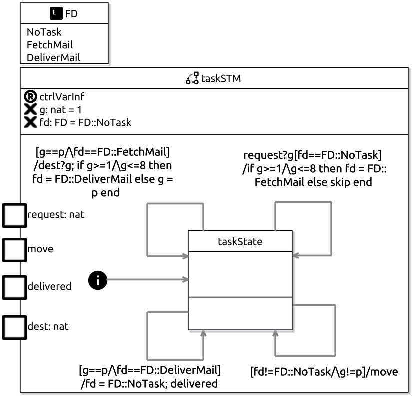
Translation: ② state machines and composite states

- Counter scpc for each machine and composite state,
- Conjunction of counters,
- exit: six stages,
  - None,
  - Requested,
  - Request substate,
  - Waiting,
  - Substate exited,
  - Exited.



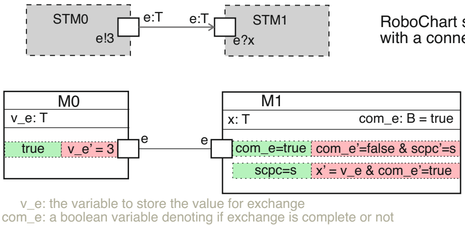
Translation: ③ transitions and actions

- Transition lock for each machine,
- One transition corresponding to multiple commands in PRISM,
- lock avoids interfere from other transitions,
- A transition taken, other transitions in the machine not to be taken till the transition is completed
- Suitable for composite states



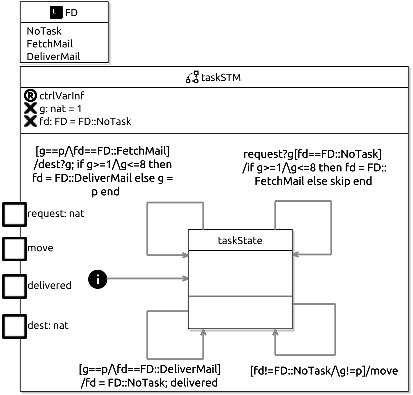
Translation: ④ input and output communication

- A variable (output): value for exchange,
- A boolean variable (input): exchange finished?



RoboChart state machines with a connection for exchange

Communication in PRISM modules



Conclusion

- Extension of RoboChart with probabilistic junction;

Conclusion

- Extension of RoboChart with probabilistic junction;
- Probabilistic semantics of RoboChart in PRISM;

Conclusion

- Extension of RoboChart with probabilistic junction;
- Probabilistic semantics of RoboChart in PRISM;
- Formalisation of semantics by two-stage translation;

Conclusion

- Extension of RoboChart with probabilistic junction;
- Probabilistic semantics of RoboChart in PRISM;
- Formalisation of semantics by two-stage translation;
- A probabilistic property language;

Conclusion

- Extension of RoboChart with probabilistic junction;
- Probabilistic semantics of RoboChart in PRISM;
- Formalisation of semantics by two-stage translation;
- A probabilistic property language;
- Automation and verification support in RoboTool;

ROBOSTAR

Conclusion

- Extension of RoboChart with probabilistic junction;
- Probabilistic semantics of RoboChart in PRISM;
- Formalisation of semantics by two-stage translation;
- A probabilistic property language;
- Automation and verification support in RoboTool;
- Future work: time, during actions, rich types/expressions (data refinement), more case studies

ROBOSTAR

Thank you!

https://robostar.cs.york.ac.uk/

ROBOSTAR