

Model-Based Testing of Robotic Applications

Juhan Ernits

Institute of Cybernetics / Dept. of
Computer Science, Tallinn University of
Technology

Arvutiteaduse teooriapäevad Vanaõuel, 29. 09.2007

Status of the Work

- This is a report of work in progress, thus there are lots of loose ends and open questions.

What to Model?

- The requirements, example from RoboticsTutorial5 of Microsoft Robotics Studio:
 - Whenever there is a sufficiently large open space ahead, move into the space. Of course, this is not sufficient to be called *explorer* because the robot might easily get stuck in a cul-de-sac. Hence, you also need:
 - If there is no space ahead, scan the entire environment and select the direction with the largest open space to move into. These rules are seemingly sufficient for the robot to roam around office corridors, but our fellow colleagues from across the corridor also insist on:
 - Avoid running into obstacles, and...
 - If an obstacle is accidentally hit, stop and retreat.

What to Model?

- Environment
 - to see how the robot reacts to weird combinations of the parameters of the environment
- Partners
 - To see if the robot can cope with communication and whatever else it is supposed to do
- Data from certain sensors
 - To test a particular functionality
- ...

Modelling with Model Programs using NModel

```
enum RobotState {MovingForward, Stopped}

public static class Contract
{
    static RobotState state = RobotState.Stopped;

    [Action]
    public static void MoveForward()
    {
        state = RobotState.MovingForward;
    }
    public static bool MoveForwardEnabled()
    { return state == RobotState.Stopped; }

    [Action]
    public static void Stop()
    {
        state = RobotState.Stopped;
    }
}
```

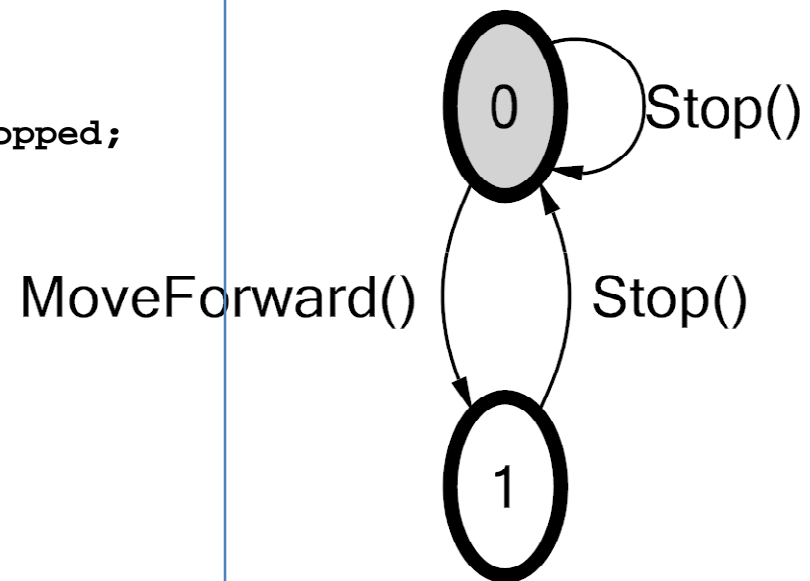
Modelling with Model Programs

```
enum RobotState {MovingForward, Stopped}

public static class Contract
{
    static RobotState state = RobotState.Stopped;

    [Action]
    public static void MoveForward()
    {
        state = RobotState.MovingForward;
    }
    public static bool MoveForwardEnabled()
    { return state == RobotState.Stopped; }

    [Action]
    public static void Stop()
    {
        state = RobotState.Stopped;
    }
}
```



Finite state machine
view of the model
program

MS Robotics Studio

- Microsoft Robotics Studio is an environment for building robotics applications with interfaces to a variety of available platforms like, [Pioneer 3](#), [Lego NXT](#), [iRobot Roomba](#), etc.
- Supports coding in several languages, most notably C# and a visual dataflow language.
- In MSRS everything is a service.

MSRS Architecture

- All communication takes place using a Decentralised Software Services Protocol that runs on top of SOAP.
- Services live in DSSHosts, that provide some infrastructure (see example).

Demo

- We had a look at how a service implemented the IStepper of NModel.
- Due to the DifferentialDrive abstraction, the testing demo runs on a robot (Lego NXT) and in the simulation environment of MSRS.

Conclusion

- We had a look at a trivial example of testing robotic services using NModel and Microsoft Robotics Studio
- We did not have a look at feedback and test verdicts which is an essential part of testing.
- There is a lot to do in this area and by the way, there is some math behind the scenes too.
- If anybody is interested in joining the experiments, drop an e-mail to juhan@cc.ioc.ee!

Links

- <http://www.microsoft.com/robotics>
- <http://www.codeplex.com/NModel>

Acknowledgements

- The ideas presented here have emerged from discussions with [Margus Veanes](#) and [Johannes Helander](#)
- A BSc student, [Konstantin Saveljev](#), helped to explore the topic in his BSc work in the Spring of 2007.