Rigorous Methods for Software Engineering (F21RS-F20RS) Getting started with Spin

Andrew Ireland
Department of Computer Science
School of Mathematical and Computer Sciences
Heriot-Watt University
Edinburgh

Overview

- Context and a little history.
- Accessing **Spin**.
- How to execute a **Promela** program via **iSpin**, *e.g. Hello World!*

A Brief History of Spin

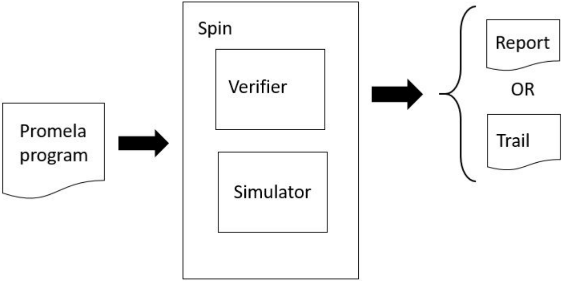
- **Spin** was developed by Gerard J. Holzmann with others in the 1980s at Bell Labs within the Unix group of the Computing Sciences Research Center.
- The tool has been available freely since 1991.
- In 2001 the tool was awarded the ACM System Software Award:

  For SPIN, a highly successful and widely used software model-checking system based on "formal methods" from Computer Science. It has made advanced theoretical verification methods applicable to large and highly complex software systems."
- Holzmann setup NASA/JPL's Laboratory for Reliable Software (LaRS) in 2003, located in Pasadena, CA, USA.

Promela, Spin and iSpin

- **Promela** – PROcess MEta LAngugae – is a language formally modelling distributed communicating systems:
  - Influenced by Dijkstra's guarded command language, Hoare's process algebra CSP and has C-like syntax;
  - Process communication modelled via message channels, both asynchronous and synchronous communication supported.
- **Spin** – Formal analysis tool for Promela programs:
  - Supports simulation, either random or interactive;
  - Supports formal verification, *i.e.* absence of deadlocks; unexecutable code; non-progress execution cycles; model checking for linear time temporal properties.
- **iSpin** – provides a GUI for **Spin**.

The Spin Approach



- If the **Verifier** is successful it generates a **Report** while an unsuccessful verification generates a **Trail** (i.e. a counter-example)
- The **Simulator** allows you to explore counter-examples interactively.

Accessing Spin Remotely and on the Edinburgh Campus

- We will use **Spin** through its GUI which is called **iSpin**.
- **Linux:**
  - Remote access to **iSpin** via the School Linux Lab (EMB 2.50) requires **X2GO**, for details see:

    `https://www.macs.hw.ac.uk/cs/faq.html#Qnx`

  Note: for general instructions on downloading and installing **Spin** and **iSpin** see:
  `http://spinroot.com/spin/Man/README.html`

Accessing Spin Remotely and on the Edinburgh Campus

- We will use **Spin** through its GUI which is called **iSpin**.
- **Windows:**
  - **iSpin** is available on the School Virtual Machine (**MACS VM**) – see: `https://www.macs.hw.ac.uk/VM/`
  - The **MACS VM** can be run on the PCs in the University's GRID Digital Lab (**GRDL**), and these PCs can be accessed via **KeyServer**. For details on **KeyServer** see: `https://www.hw.ac.uk/uk/services/is/it-essentials/keyserver.htm`
  - Note that the following **GRDL** PCs have limited memory so do not have the **MACS VM** installed: PC001; PC015; PC019; PC038 PC068; PC074; PC076; PC095; PC108; PC109; PC115; PC118. However, that still leaves nearly 90 **GRDL** PCs that do have the **MACS VM** installed.
  - Note that there is **NO ACCESS** to the **MACS VM** in the EMB 2.52 Windows Lab.

Accessing Spin on your Laptop via Docker

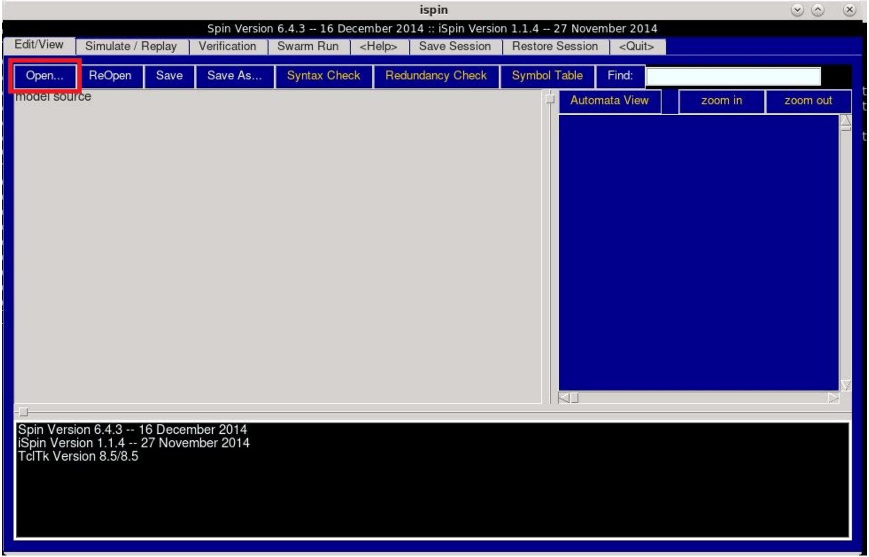
- **Docker** supports software applications via a platform that facilitates OS virtualization.
- For how to install **Docker** and run **iSpin** on your laptop, see Yuhui Lin's video and associated README text file in the **Week 7** module of the course materials on Canvas.

Hello World!

```
active proctype hello(){

/* My first Promela program (this is a comment)!  */

   printf("Hello World!\n")
}
```

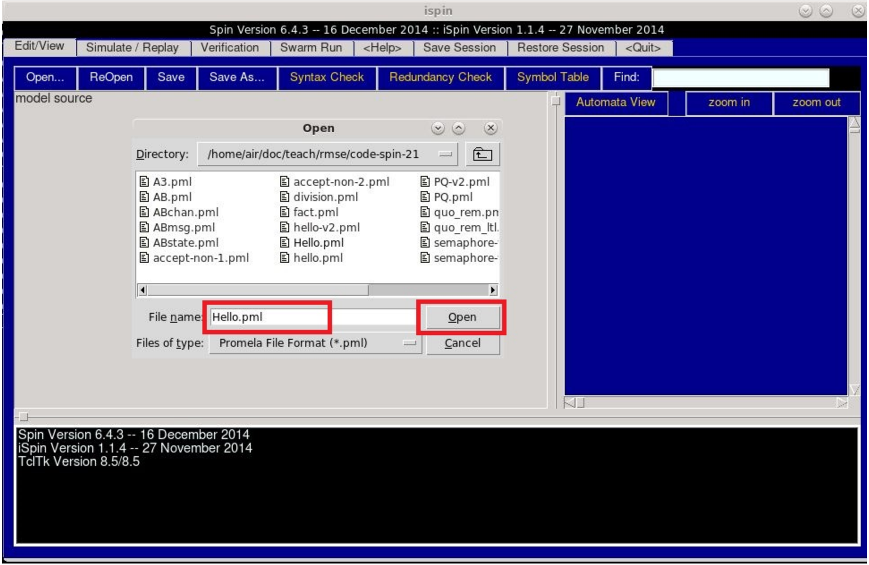
A closer look at the language later, for now let's see how to execute this code.

Hello World



```
Spin Version 6.4.3 -- 16 December 2014
iSpin Version 1.1.4 -- 27 November 2014
TclTk Version 8.5/8.5
```

Hello World



Directory: /home/air/doc/teach/rmse/code-spin-21

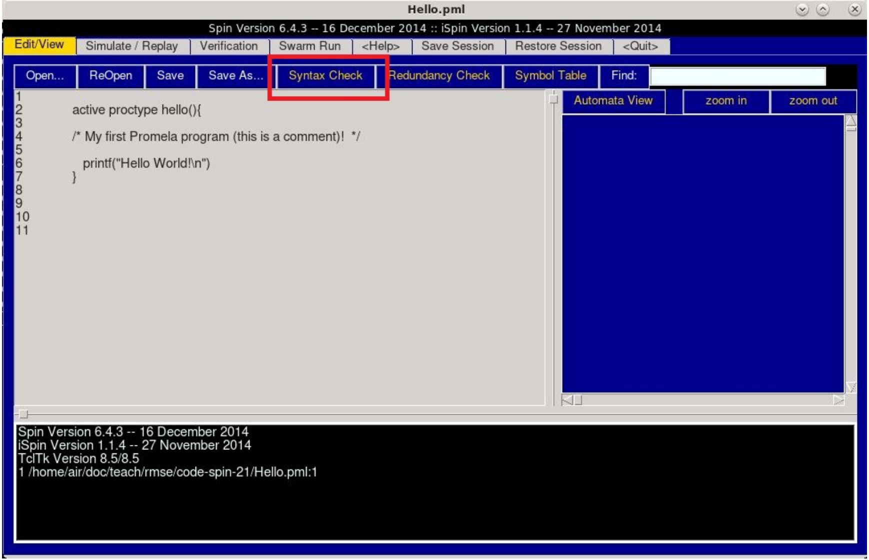
A3.pml
AB.pml
ABchan.pml
ABmsg.pml
ABstate.pml
accept-non-1.pml
accept-non-2.pml
division.pml
fact.pml
hello-v2.pml
Hello.pml
hello.pml
PQ-v2.pml
PQ.pml
quo_rem.pn
quo_rem_ltl.
semaphore-
semaphore-

File name: Hello.pml

Files of type: Promela File Format (*.pml)

```
Spin Version 6.4.3 -- 16 December 2014
iSpin Version 1.1.4 -- 27 November 2014
TclTk Version 8.5/8.5
```

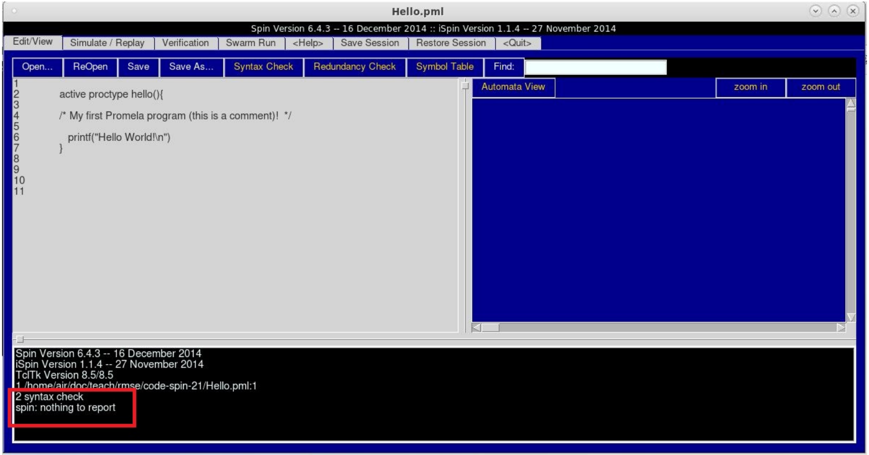
Hello World



```

active proctype hello(){

/* My first Promela program (this is a comment)!  */

   printf("Hello World!\n")
}
```

```
Spin Version 6.4.3 -- 16 December 2014
iSpin Version 1.1.4 -- 27 November 2014
TclTk Version 8.5/8.5
1 /home/air/doc/teach/rmse/code-spin-21/Hello.pml:1
```

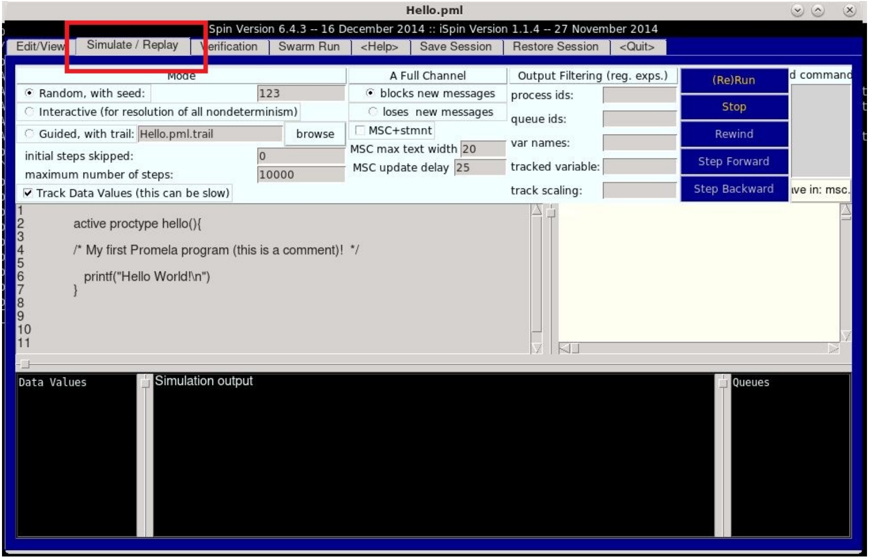
Hello World



```
active proctype hello(){

/* My first Promela program (this is a comment)!  */

  printf("Hello World!\n")
}
```

```
Spin Version 6.4.3 -- 16 December 2014
iSpin Version 1.1.4 -- 27 November 2014
TclTk Version 8.5/8.5
1 /home/air/doc/teach/rmse/code-spin-21/Hello.pml:1
2 syntax check
spin: nothing to report
```

Hello World



Mode

- ● Random, with seed: 123
- ○ Interactive (for resolution of all nondeterminism)
- ○ Guided, with trail: Hello.pml.trail

initial steps skipped: 0

maximum number of steps: 10000

☑ Track Data Values (this can be slow)

A Full Channel

- ● blocks new messages
- ○ loses new messages

□ MSC+stmnt

MSC max text width 20

MSC update delay 25

Output Filtering (reg. exps.)

process ids:

queue ids:

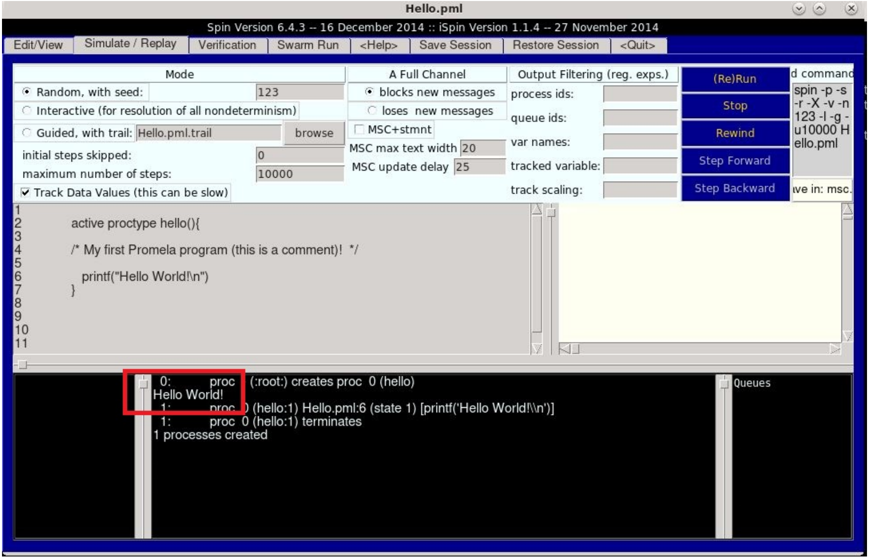
var names:

tracked variable:

track scaling:

```

active proctype hello(){

/* My first Promela program (this is a comment)!  */

   printf("Hello World!\n")
}
```

Hello World



Spin Version 6.4.3 -- 16 December 2014 :: iSpin Version 1.1.4 -- 27 November 2014

Mode

● Random, with seed: 123

○ Interactive (for resolution of all nondeterminism)

○ Guided, with trail: Hello.pml.trail

initial steps skipped: 0

maximum number of steps: 10000

☑ Track Data Values (this can be slow)

A Full Channel

● blocks new messages

○ loses new messages

☐ MSC+stmnt

MSC max text width 20

MSC update delay 25

Output Filtering (reg. exps.)

process ids:

queue ids:

var names:

tracked variable:

track scaling:

d command

spin -p -s -r -X -v -n123 -l -g -u10000 Hello.pml

ve in: msc.

```
active proctype hello(){

/* My first Promela program (this is a comment)!  */

  printf("Hello World!\n")
}
```

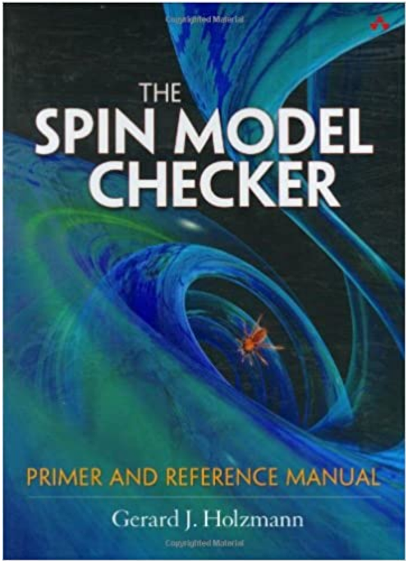
```
0:        proc  (:root:) creates proc  0 (hello)
Hello World!
  1:      proc  0 (hello:1) Hello.pml:6 (state 1) [printf('Hello World!\\n')]
  1:      proc  0 (hello:1) terminates
1 processes created
```

Summary

Learning outcomes:

- A brief history of **Spin**.
- How to access the **Spin**.
- How to execute **Promela** code using **iSpin**.

Summary



Recommended reading:

- “The SPIN MODEL CHECKER Primer and Reference Manual” Holzmann, G.J., Addison-Wesley, 2003.
- Spin homepage: `http://spinroot.com/`