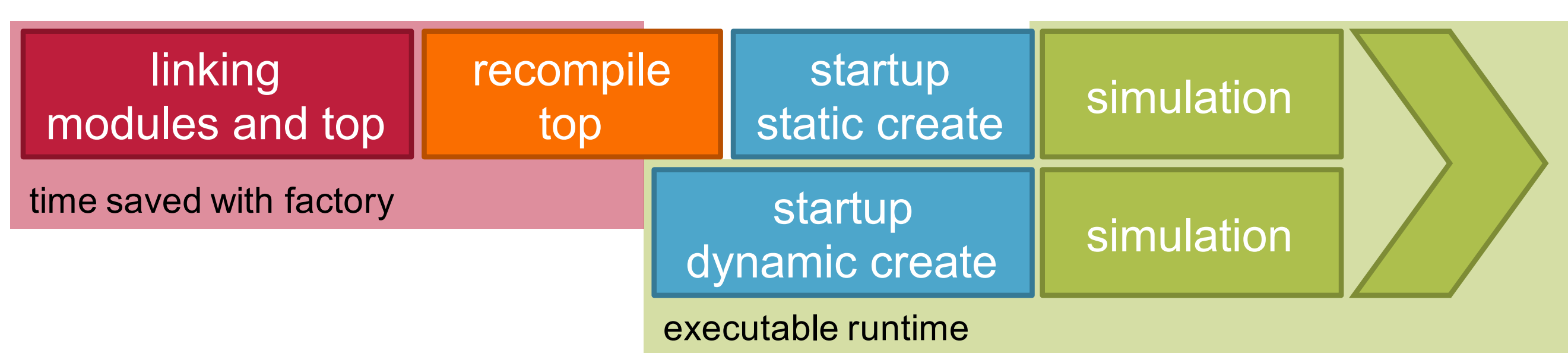


Transparent SystemC Model Factory for Scripting Languages

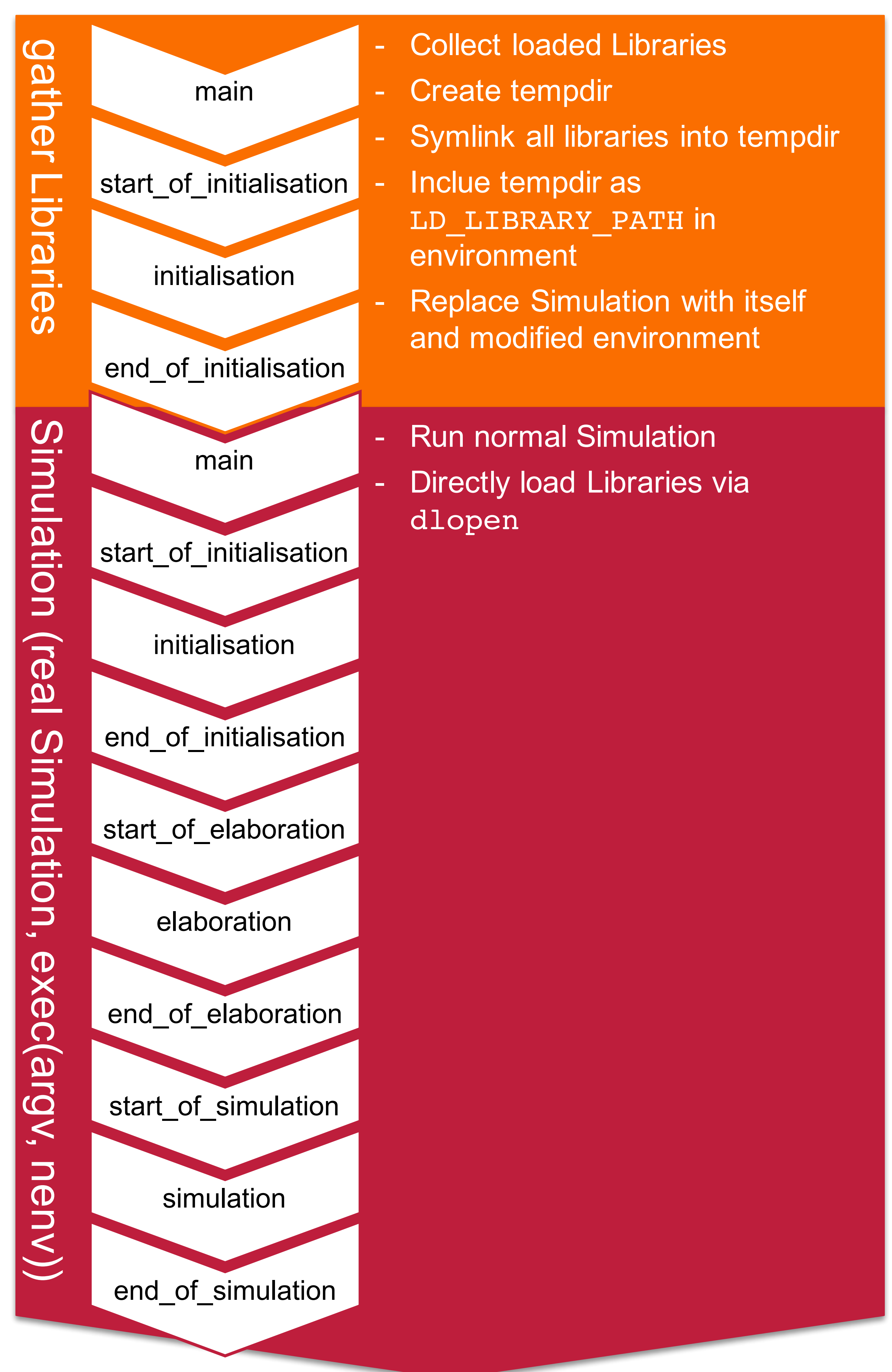
Rolf Meyer, Bastian Farkas, Syed Abbas Ali Shah, Mladen Berekovic – Technische Universität Braunschweig

Save compilation and link time



- Introducing dynamic creation overhead
- No need to recompile and link the design multiple times
- Introduces small instantiation overhead
- Allows scripting flexibility into the simulation

Dynamic library loading



- Support for advanced scenarios
- Binary distribution of models
- Runtime extendability of simulations

Top-Level Python script

```

1 import usi
2 class Top(usi.Module):
3     def __init__(modulename):
4         super(Top, self).__init__(modulename)
5         self.ahbctrl = usi.registry.AHBCtrl(
6             "ahbctrl", rrobin=True)
7         self.ahbctrl.ahbOUT.bind(self.apbctrl.ahb)
8         # ...
9 usi.registry.load('./build/models/libahbctrl.so')
10 top = Top("top")
11 usi.start()

```

1. Defines a structural sc_module Top
2. which instantiates a AHBCtrl model,
3. parametrizes it over CCI
4. and connects it to other models.
5. Loads a Library containing the ahbctrl model.
6. Instantiating the Top model.

Registration of a simple sc_model

```

1 #include <systemc>
2 #include <sr_registry>
3 SR_HAS_MODULE(AHBCtrl);
4
5 class AHBCtrl : public sc_module {
6     // ...

```

- Registers a sc_module class
- If more constructor parameters than the class name are needed a constructor function needs to be provided.

Sources

The implementation is online available to everyone:

- **Source code:** https://github.com/socrocket/sr_registry
- **Licence:** Apache=2.0