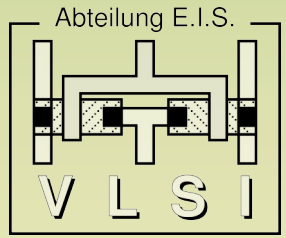
GreenSocs and GreenBus Overview

Wolfgang Klingauf

12th ESCUG Meeting

FDL'05 Lausanne



GreenSocs

Project

- Open source project
- Community based
- Wiki
- SourceForge repository

Company

- UK based Limited Liability(Ltd) Services company
- Built on a partnership model using open source collaboration
- Services the GreenSocs Project

Bring SystemC to a wider community
Help develop de-facto standards

GreenSocs SystemC Sandpit

Contributions

GreenSocs Engineering

Input to Standards Process

In partnership with (e.g.) OSCI

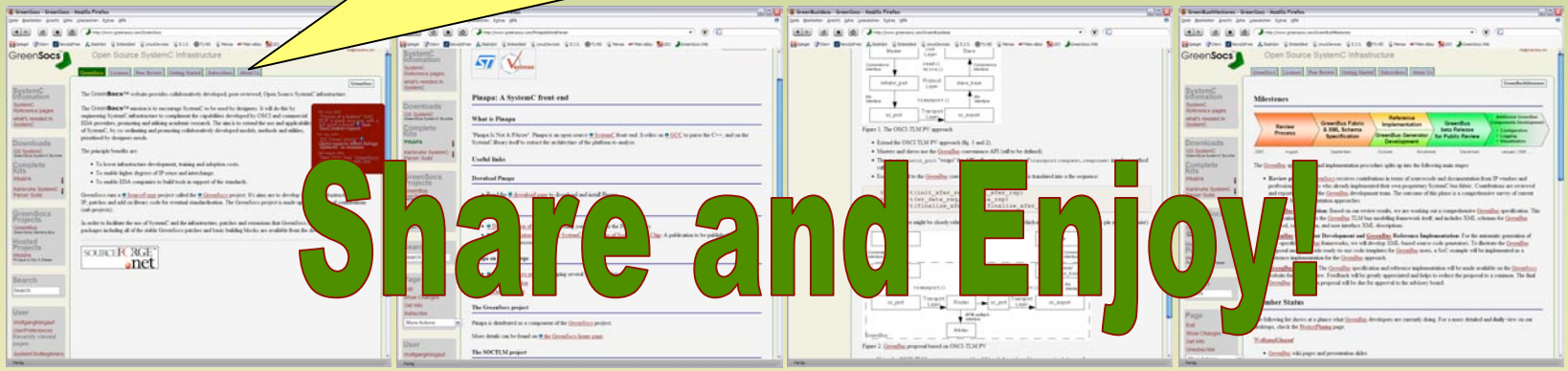
Needs Guidance:
Subscribers and GreenSocs Engineering

Open Playground

- Op
- Ins
  -
- Pe
- Onl

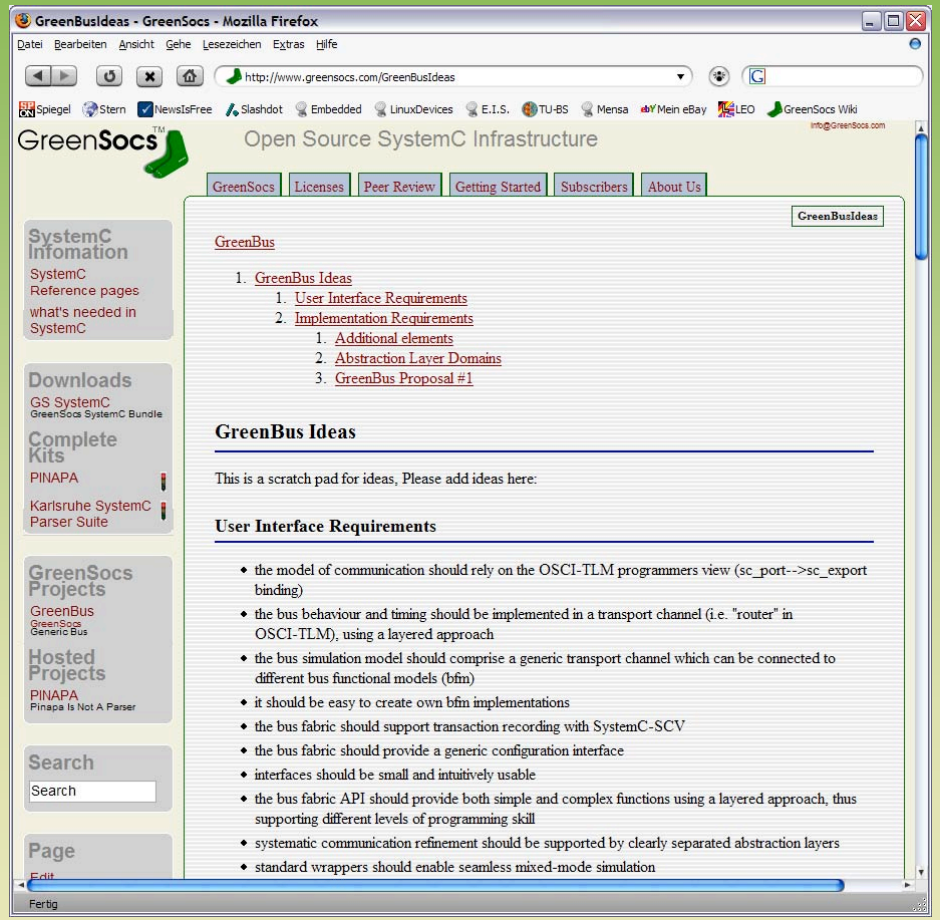
Please

Contribute!



Share and Enjoy!

Open Database

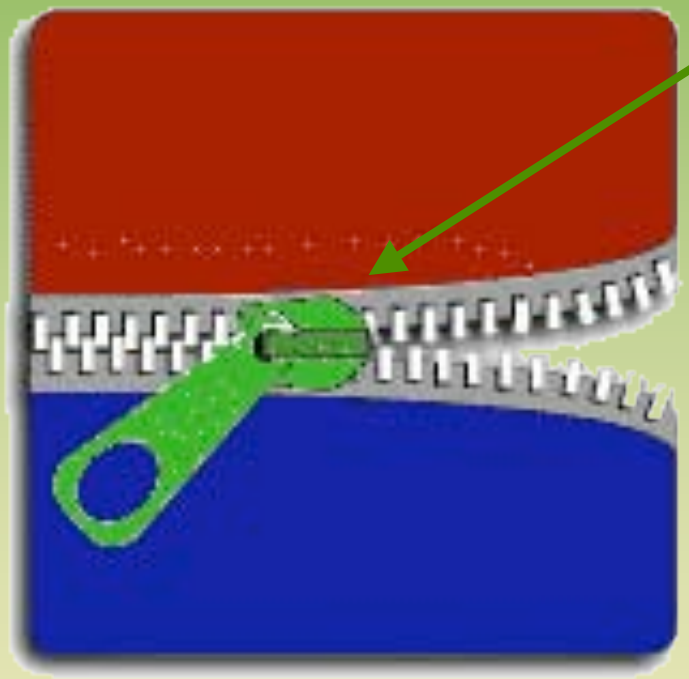


- **What's there today?**
  - Public Wiki
  - SystemC parser kits
  - GreenBus project pages
  - Place to put your stuff, and ideas!
- **More to come…**
  - Eclipse plugins
  - SystemC tutorials
  - EDA tool database

GreenSocs Zipper

Always more to do!

New frameworks and standard proposals being developed bringing IP and tools together



Standards

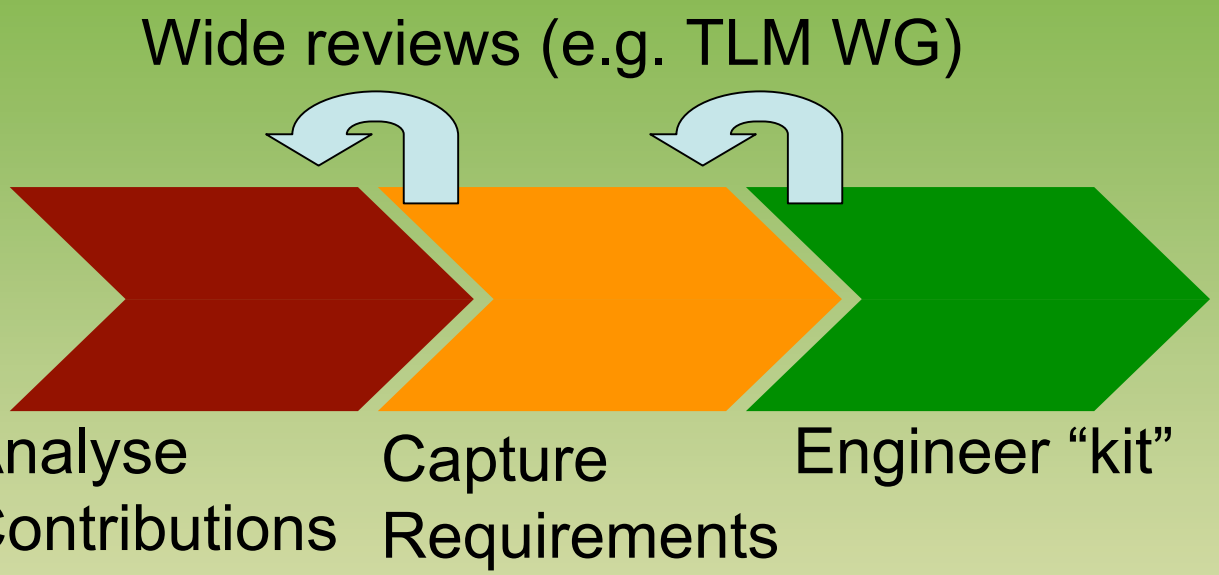
User developments

Time…


GreenBus

Process to follow:

Standards Process (e.g. TLM WG)



GreenSocs Input to process

- So far 2 contributions have already been analysed
- Some requirements captured
- Initial “engineering” experiments underway…

Review Criteria

- **Review criteria** can be found on **www.greensocs.com/GreenBusReviewCriteria**
- Currently divided into
  - **Developers view** (implementation detail)
  - **User view** (API detail)
- **Please contribute ideas** about what should be reviewed!
- **Reviews will be made public when code is public** (ST's review within hours!)

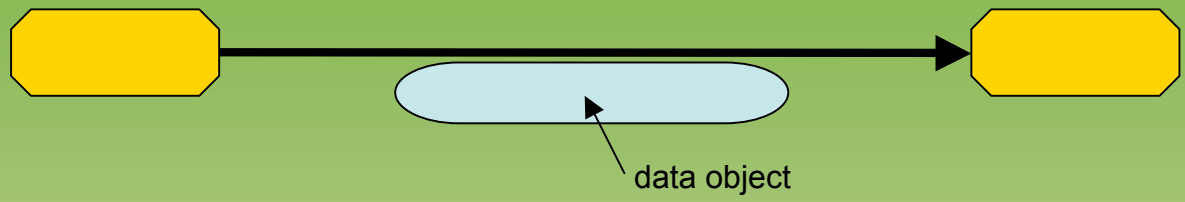
Initial Requirements…

- **What we need** is **not** a bus, it is a **framework** for building TLM models of buses
- **It must be**
  - based on the thorough **analysis of industrial TLM SoC models** and **SoC design requirements**
  - aimed at strongly contributing to **IP-exchange** between companies
  - illustrated by an example **reference implementation**
- **Goal:** It will be easy to use and highly runtime-configurable, (even with respect to simulation performance)

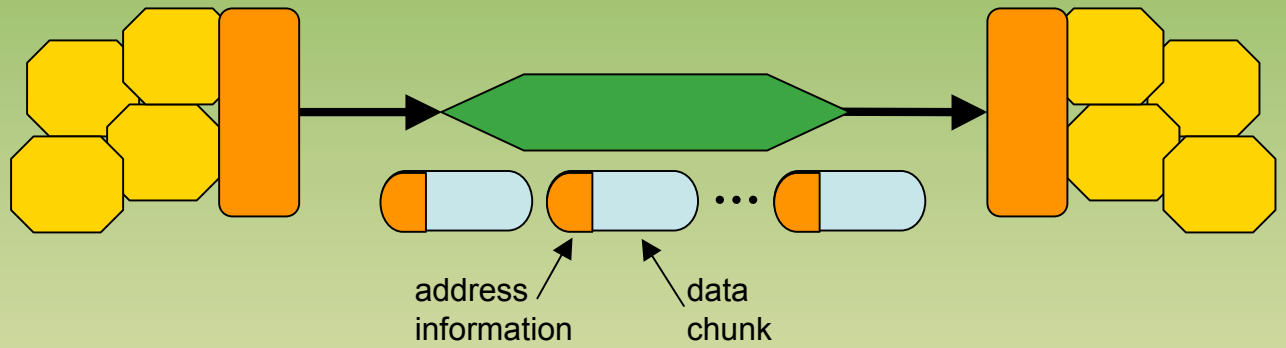
(Details, as ever, on www.greensocs.com)

About Abstraction Levels

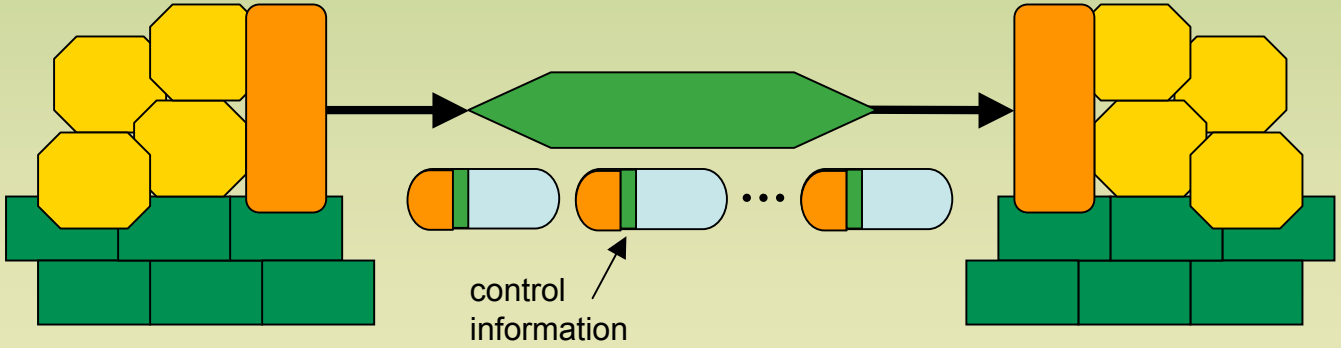
Communicating processes execute algorithm and transmit complete data blocks (to specific interface)



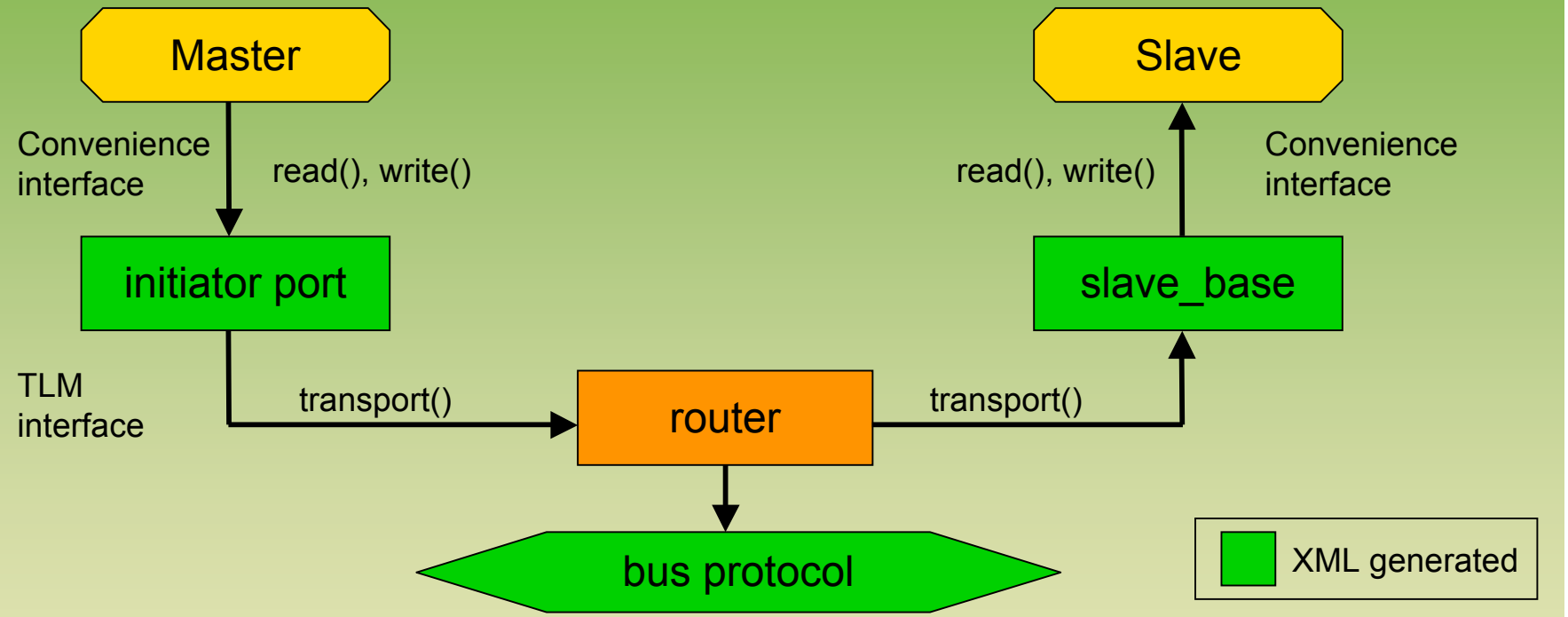
PV/PVT serialization:
To transmit over bus, PV/PVT modules must serialize data



BA/CC serialization:
Refinement adds control information and serializes still further



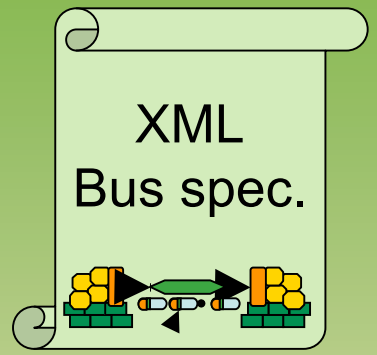
Proposed Use Case



- **Mixes** of abstraction level and protocol **are possible**, but require wrapper functions which affect simulation performance

From XML generate Bus!

- **XML descriptions** of:
  - User TLM interface
  - Bus protocol and parameters
  - Bus configuration
- Generate SystemC code **automatically**
  - Application-specific convenience API
  - Serialization of transaction objects
  - Bus arbitration and timing
- Generated bus fabric supports:
  - different **user TLM APIs**
  - application-specific **highest simulation performance**

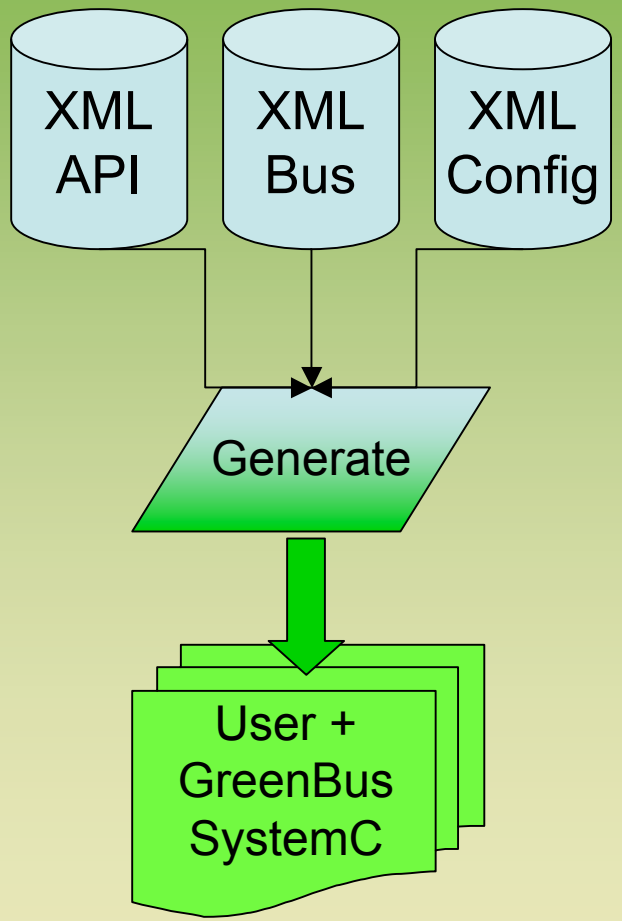


XML Generator

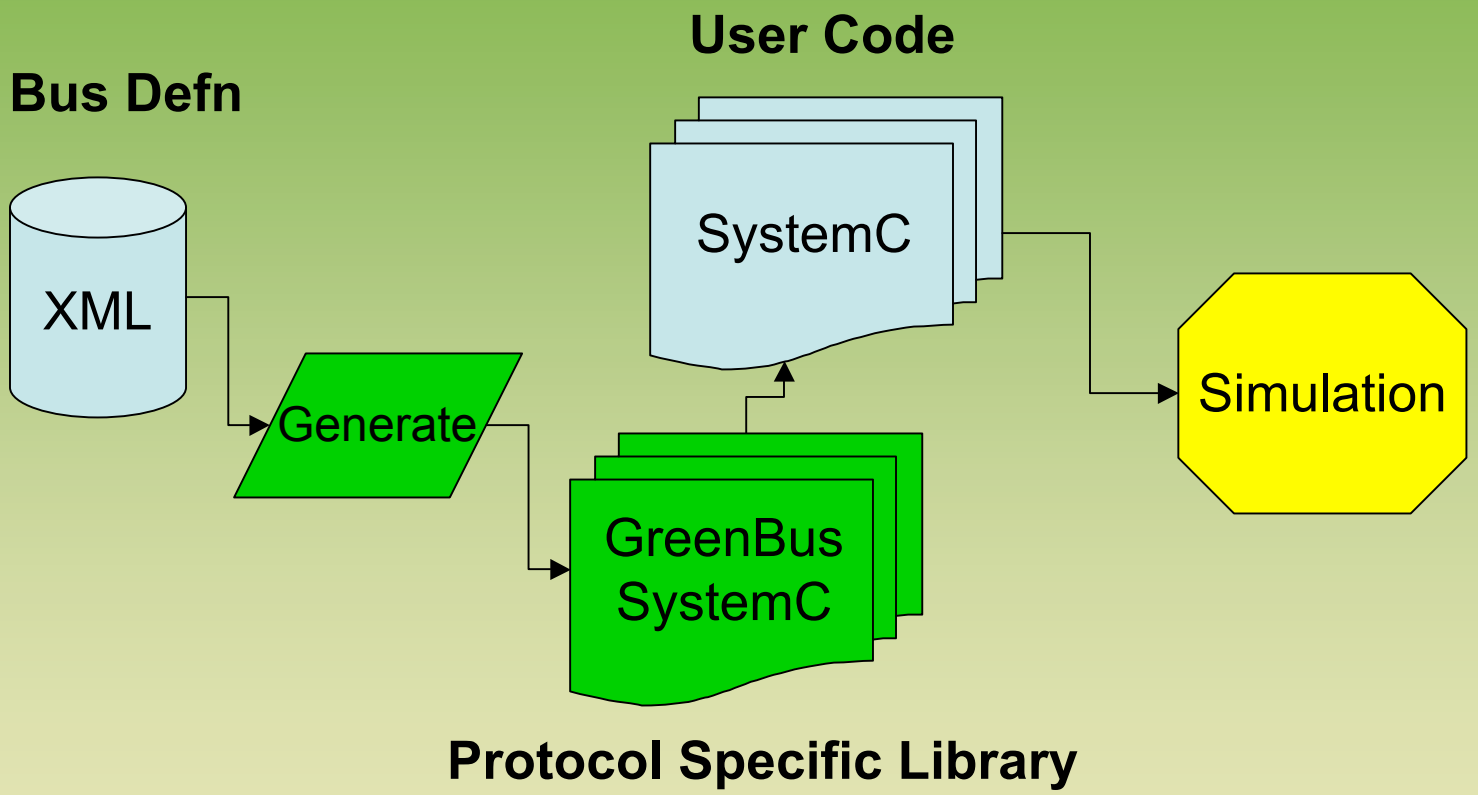
- **Serializers**
  - User data → PV/PVT
  - PV/PVT → BA/CC
- **Data structures**

```
class PVS : PVBase {
  unsigned char[1024] data;
  sc_uint address;
  enum {Read,Write} IFName;
};

class BAS: BABase{
   struct data{
    unsigned char[8] data;
    enum {Error,Complete} Status;
  } data;
  struct timing {
    bool dataValid;
    bool statusValid;
  }
};
```



Flow Overview

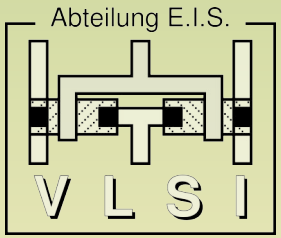


Conclusion

- GreenSocs helps to develop kits for SystemC
- Open source community, supported by contributing companies and institutions
- GreenBus is our project to help generate a generic framework for bus modelling with SystemC
- All GreenSocs “products” will be designed to meet industrial requirements

Thank You!

mark@greensocs.com
www.greensocs.com



w.klingauf@tu-bs.de
www.eis.cs.tu-bs.de