SPEET
SDL Performance Evaluation Tool

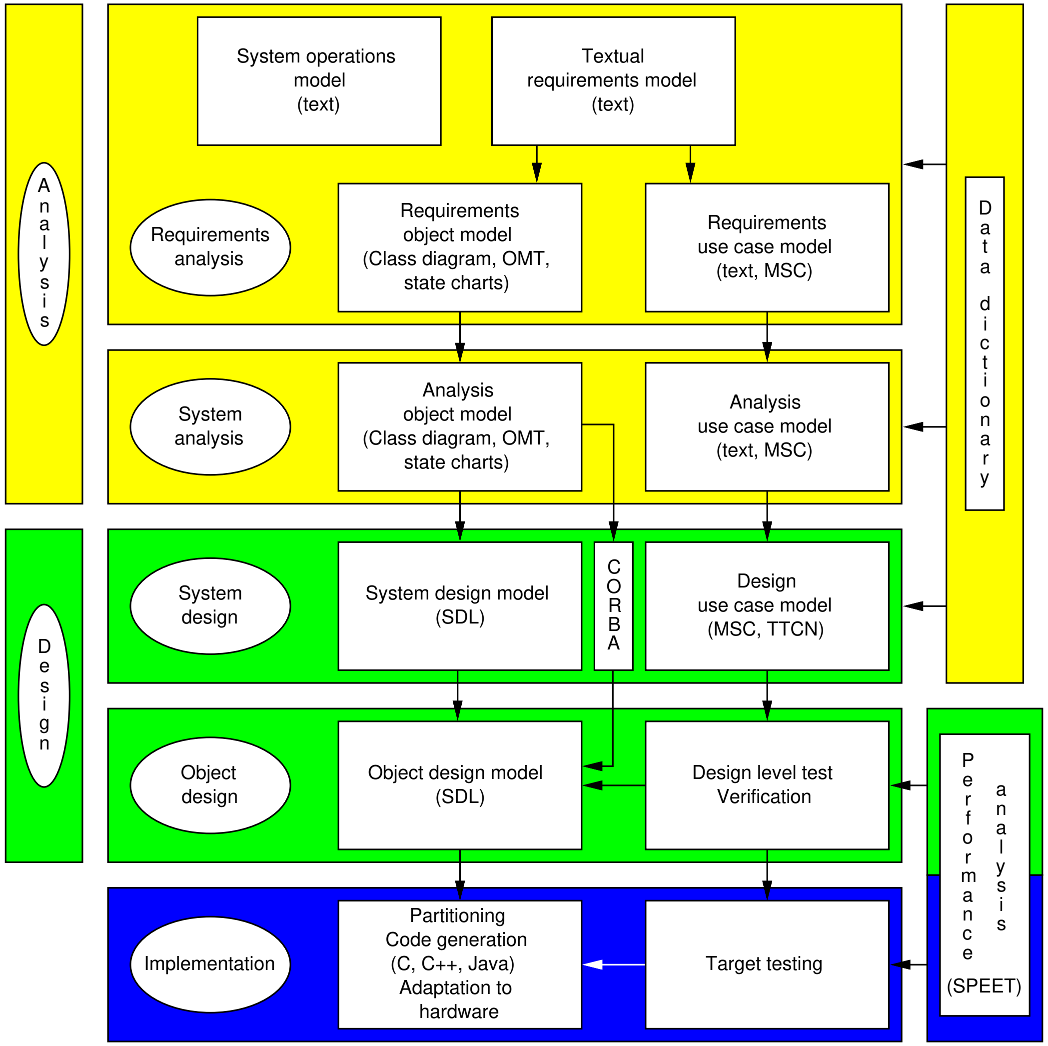
Martin Steppler

1. Introduction
2. Product development as a twofold recursive cycle
3. SPEET tool components
4. Traffic load generators
5. Transmission models
6. Hardware emulator
7. Statistical evaluation
8. Conclusions

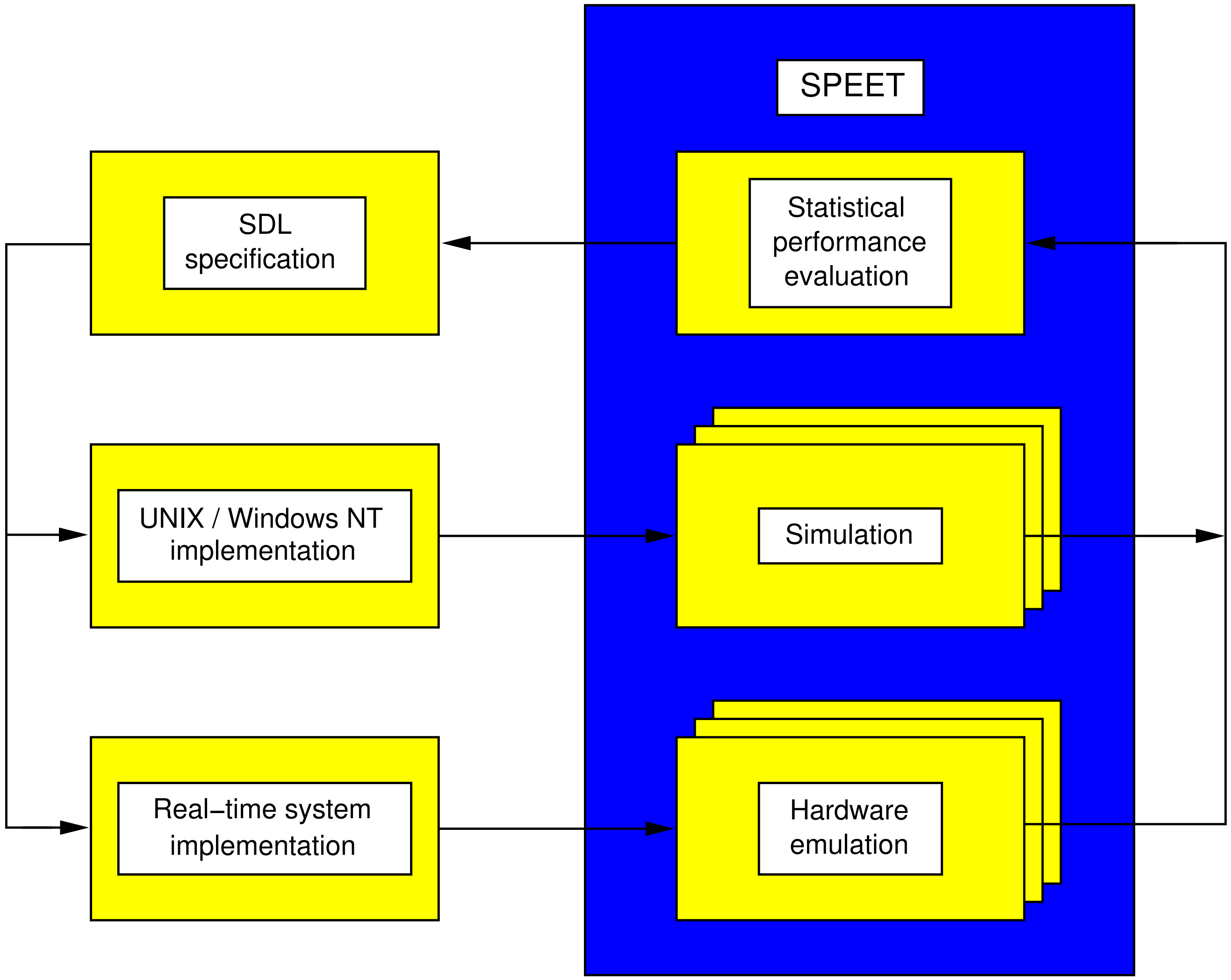
Introduction

- Formal specification languages
  - LOTOS (ISO 8807)
  - Estelle (ISO 9074)
  - SDL (ITU-T Z.100)
- Performance evaluation of formally specified protocols by means of simulation and emulation
- Prediction of hardware requirements, e. g. used amount of memory, and whether real–time conditions are met
- Emulation of SDL implementations for target processor systems

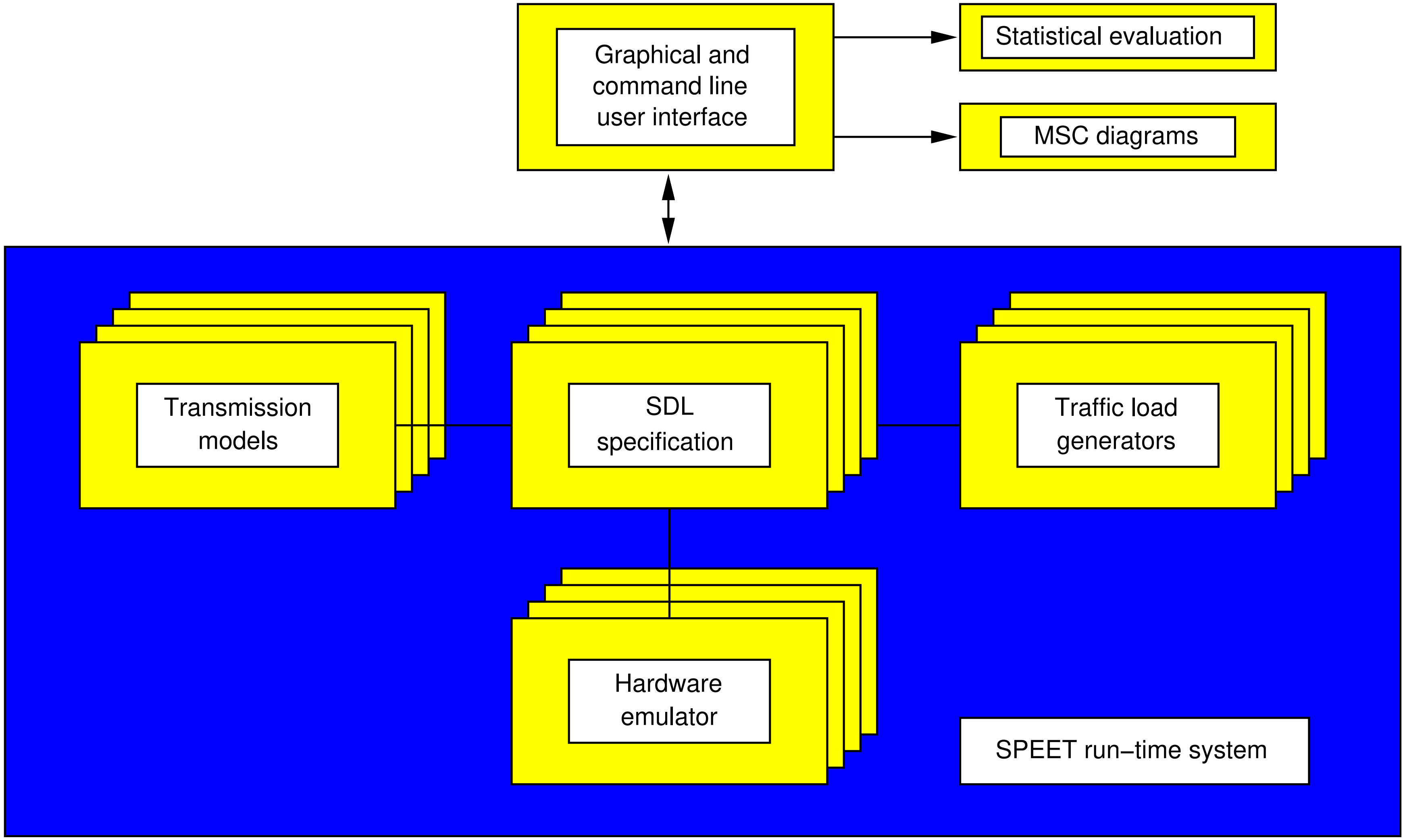
Object–oriented software design



Product development as as twofold recursive cycle

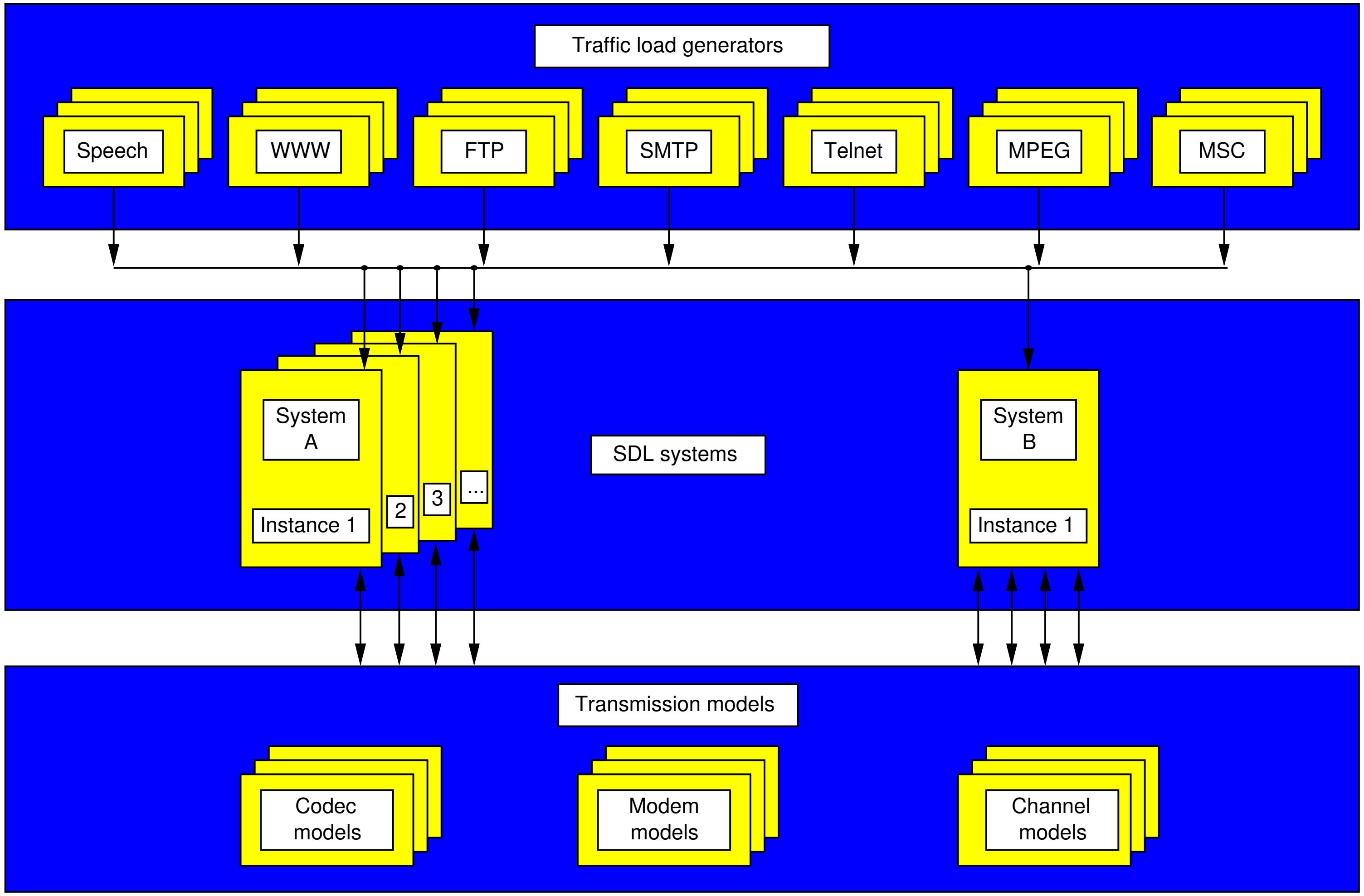


SPEET tool components

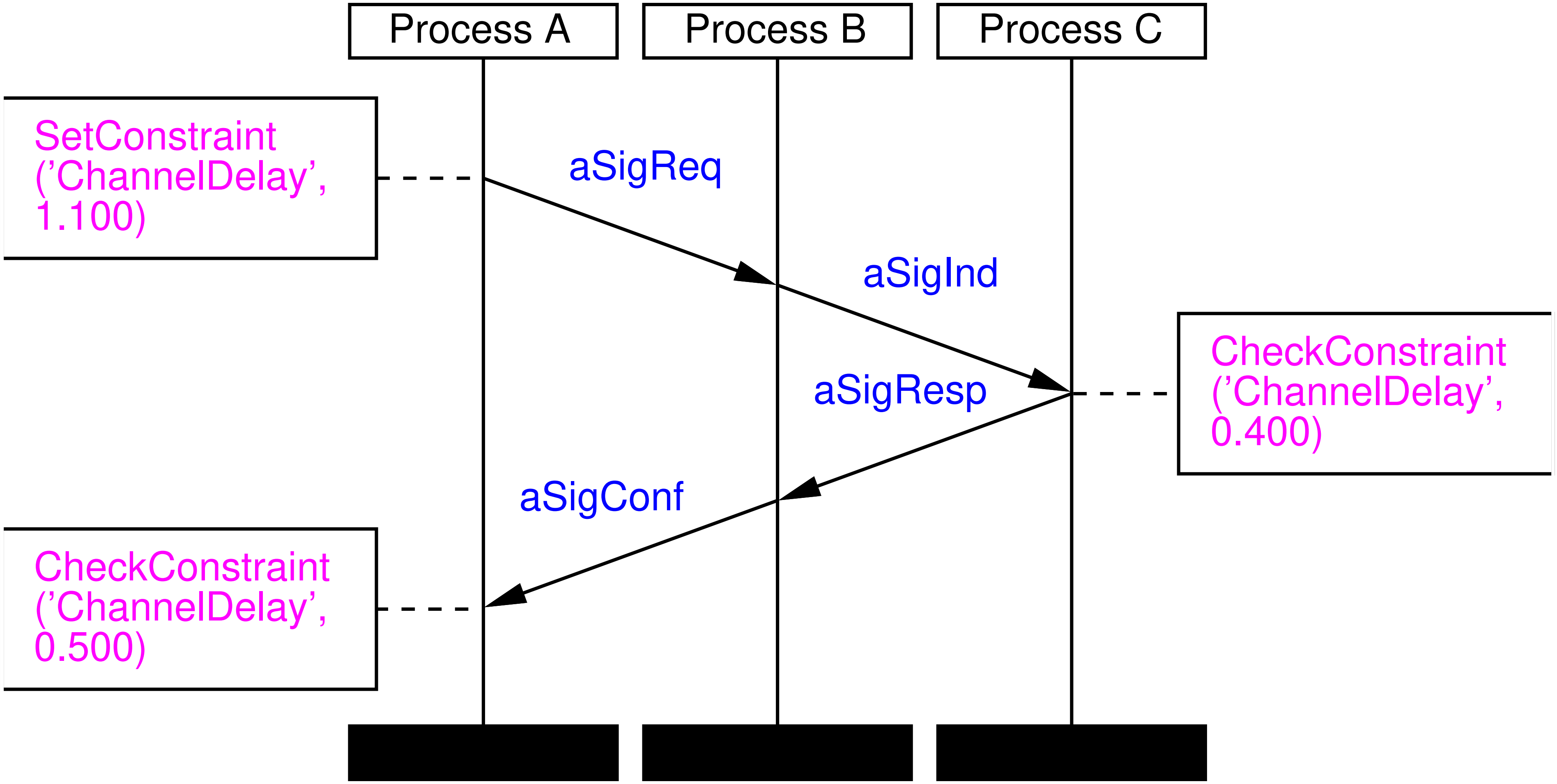


Traffic load generators

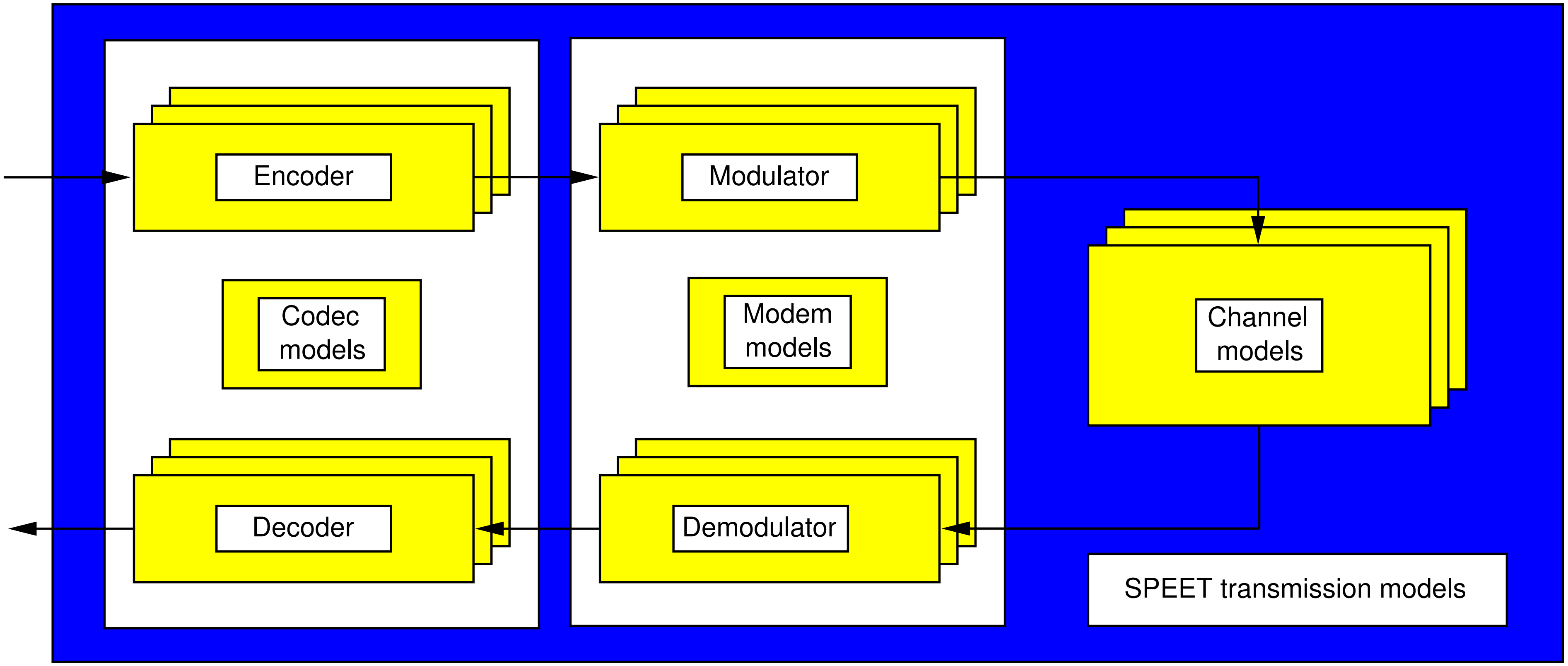
Typical SPEET scenario



MSC with time constraints



Transmission models

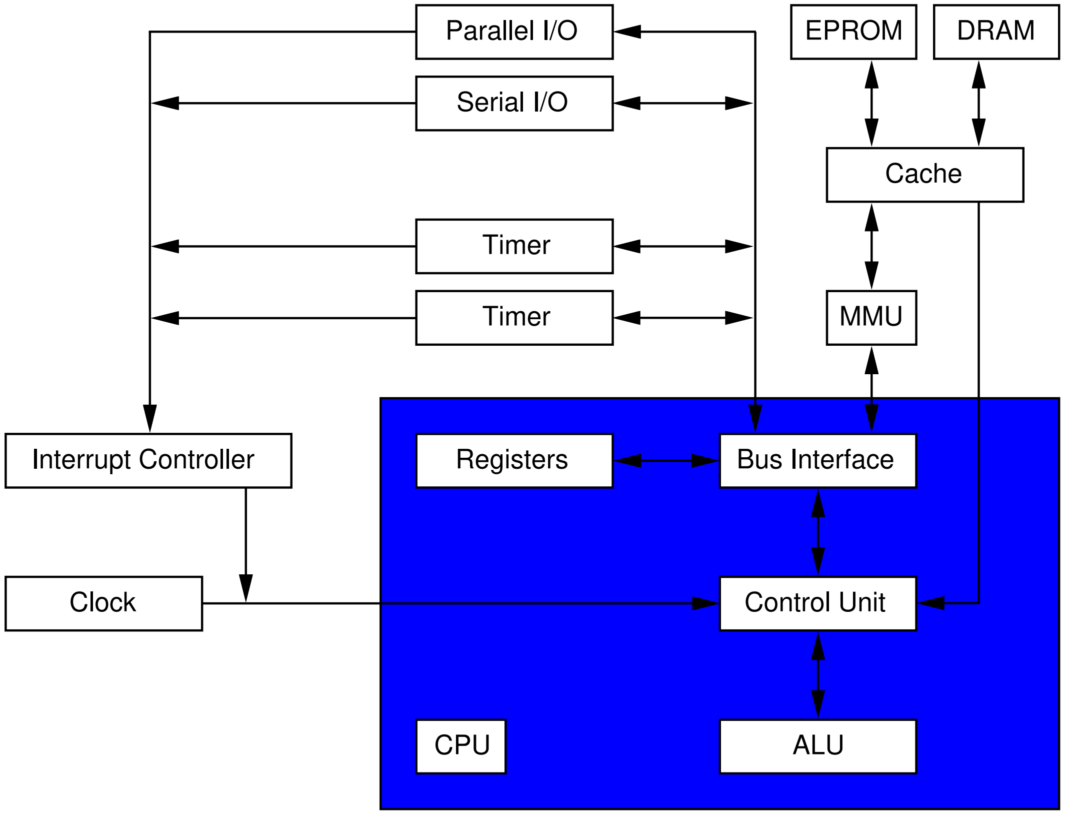


Hardware emulator

Processors

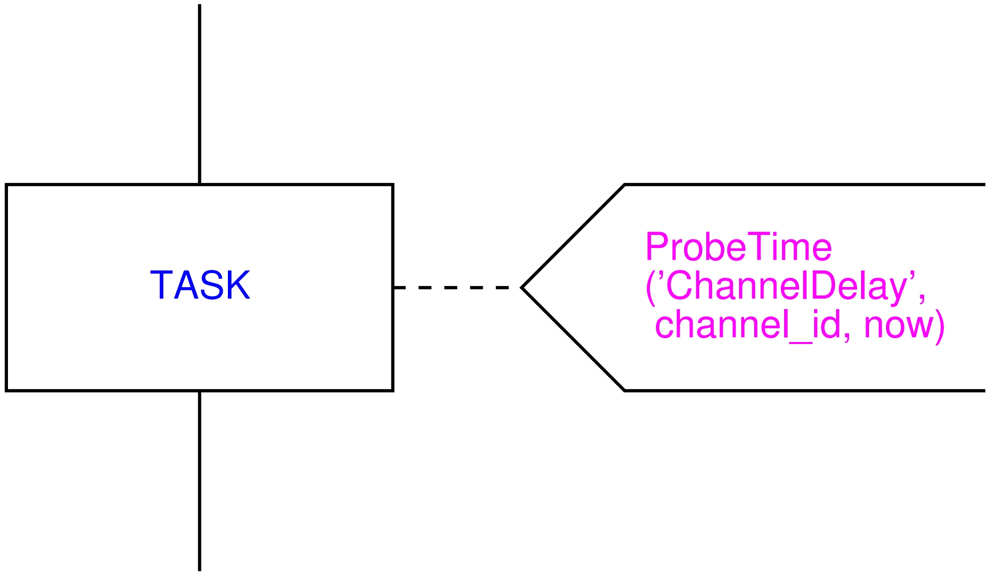
- Intel 8051
- Siemens 80C166/7
- Motorola 680x0
- Hyperstone

Components of an example processor system



Statistical evaluation

- Evaluation method „Limited Relative Error“
- Simulation run–length dependent on measured correlation
- Insertion of probes via a new SDL symbol



Conclusions

- Formal CASE methods extended by performance evaluation
- SPEET features
  - Parallel simulation of multiple SDL systems
  - Parallel emulation of multiple SDL systems implemented for real–time target processor systems
  - Traffic load generators: Video–, audio–, speech– and packet data applications
  - Models for codecs, modems and channels
  - Hardware emulators for Intel 8051, Siemens 80C167, Motorola MC680x0
  - Statistical evaluation via the LRE method