

# Verification Support for "Plug-and-Play" Architectural Design

Shangzhu Wang, George S. Avrunin, Lori A. Clarke  
University of Massachusetts Amherst

## Problem

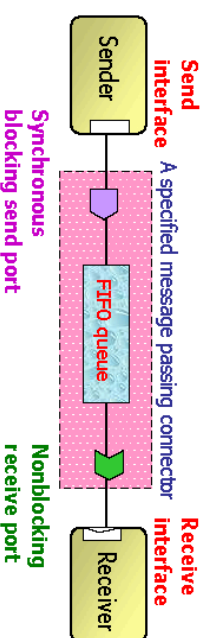
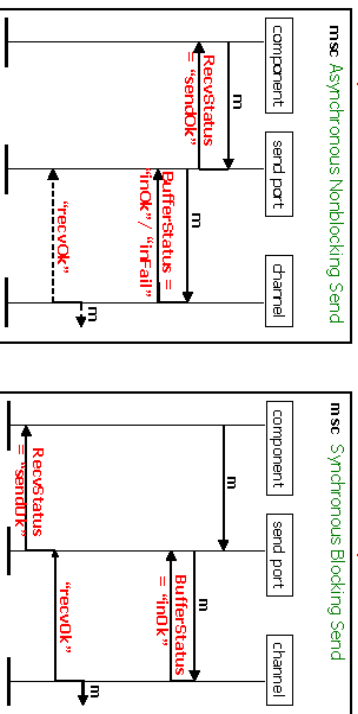
- Component interactions are important yet hard to get right
  - Large design space of alternative semantics
  - Many kinds of interaction mechanisms (RPC, MP, EBN, etc.) with variations
  - Semantics of component interactions intertwined with their computations
  - Changes in the interaction semantics often require changes in the components

## Our Approach

- Supports plug-and-play system design
  - Defines a *small set of standard interfaces* for component interactions
  - The same interface can be used with different interaction semantics
  - Defines a *library of reusable building blocks* for the construction of connectors with different interaction semantics
  - Channels represent communication media of connectors
  - Ports are mediators between components and channels
- Facilitates design-time verification
  - Reuses component models
  - Builds connector models by composing reusable building block models

Ports hide the differences in the

synchronization semantics from the components



Library of message passing building blocks

Send Ports	Asynchronous nonblocking	Asynchronous blocking	Synchronous blocking
Receive Ports	Asynchronous blocking	Synchronous blocking	Synchronous checking
Channels	Blocking receive	Nonblocking receive	Priority queue
	Single-slot buffer	FIFO queue	

## Plug and Play with Message Passing

- ➔ Reduces impact on components when connectors are changed, increasing the reusability of components
- ➔ Allows easy experimentation with alternative design choices supporting a wide range of interaction semantics
- ➔ Saves model-construction time during design-time verification

## Status & Future Work

- Applied to the message passing paradigm
  - Reusable models of building blocks created in Promela and resulting system architectures evaluated using SPIN
- Extended to support other interaction mechanisms
  - RPC, Pub/Sub, etc.
- Tools under construction

## References

Laboratory for Advanced Software Engineering Research (LASER)  
<http://laser.cs.umass.edu>