#### The Java MPI Simulator 2001

**Rok Preskar** 

29th August 2001

Overview

- MPI Simulation
- How the program works
- The progress of the project
  - Repair some of the existing problems
  - Further extend the functionality

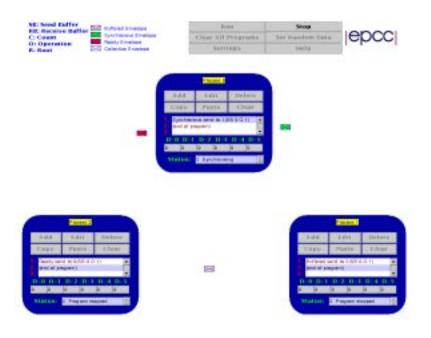
- Message Passing Interface
- Standard for parallel processing
- A message passing library available for C and Fortran

#### Simulator in Java

- Program written as an applet
- First developed for simulating blocking point-to-point communications in 1997 by SELLIC
- Collective communications were added later by an SSP project in 1998
- Some problems were removed and code was better documented by another SSP project in 2000



#### The appearance of the applet



- Three processes exchanging data
- Envelopes represent transfer of data



- ▶ The control buttons
- Program and data presentation

- Point-to-point communications
  First implemented only as envelopes
- Collective communications
  First implemented only as data transfer
- Changes to both representations to make them consistent

- Good points of Java Swing package
  - True portability
  - Easier to change the appearance of components
- Encountered problems with Swing
  - Conversion should be simple enough but sometimes is not
  - Mixing light-weight and heavy-weight components

#### Java Beans

#### epcc

#### General description

- Just a convention for writing the code
- Reusable components
- ... but not if the conventions are ignored

#### Beans in the applet

- Several different beans in the applet work independently
- Compiler forced independency

#### Saving programs

- Usability of saving
  - Better way of constructing examples
  - Making an illustrative tutorial available on the Internet
- File saving through applets
  - Different security model since Java 2 SDK 1.2
  - The problem is each user has to define it on his/her machine

- Why change the number of processes?
  - Useful for individual examples
  - No point in over-extending functionality, though
- What needed to be changed
  - Mostly just changing a constant to a normal variable
  - There were some hidden suppositions

- Comprehensibility of the applet
  - Not apparent at first what needs to be done
  - Explanation of MPI is not given
- These windows should help a user start a program

- Several errors have to be detected at input time
  - The size of the sent and receive message
  - Detecting whether user entered sensible data
- Other errors are detected at run-time

#### Conclusion

# epcc

► Live demonstration of the applet

http://www.epcc.ed.ac.uk/~rok/

Questions