Rapid Prototyping and Deployment of User-to-User Networked Applications

Wan-Teh Chang

Department of Electrical Engineering
and Computer Sciences
University of California at Berkeley

Acknowledgments

Faculty advisors: Prof. David G. Messerschmitt and Prof. Edward A. Lee

Collaborators:

- Joe Buck, Stephen Edwards, Alain Girault, Bilung Lee, and Kennard White
- Weiyi Li and Houman Hashemi

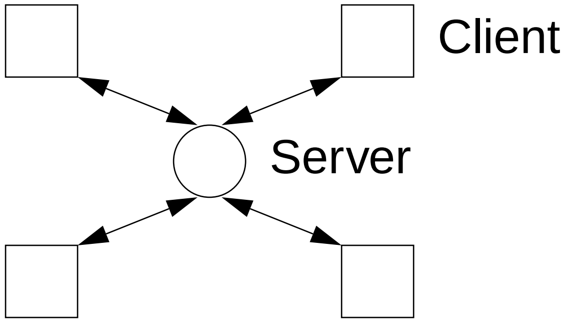
Taxonomy of networked applications

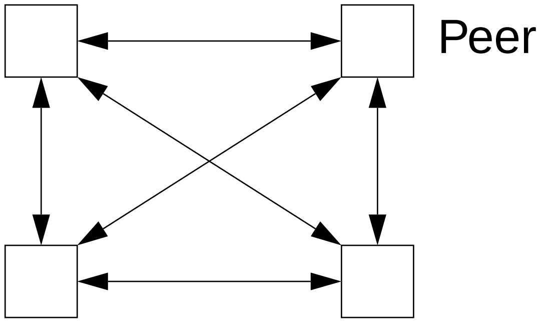
Networked applications:

- **User-to-information-server:** file transfer, news, gopher, Web browsing, and video on demand
- **User-to-user:** telephony, video conferencing, voice mail, e-mail, shared whiteboard, and shared editor

Implementation architectures:

- **Client-server:** chat
- **Peer-to-peer:** talk





Motivation

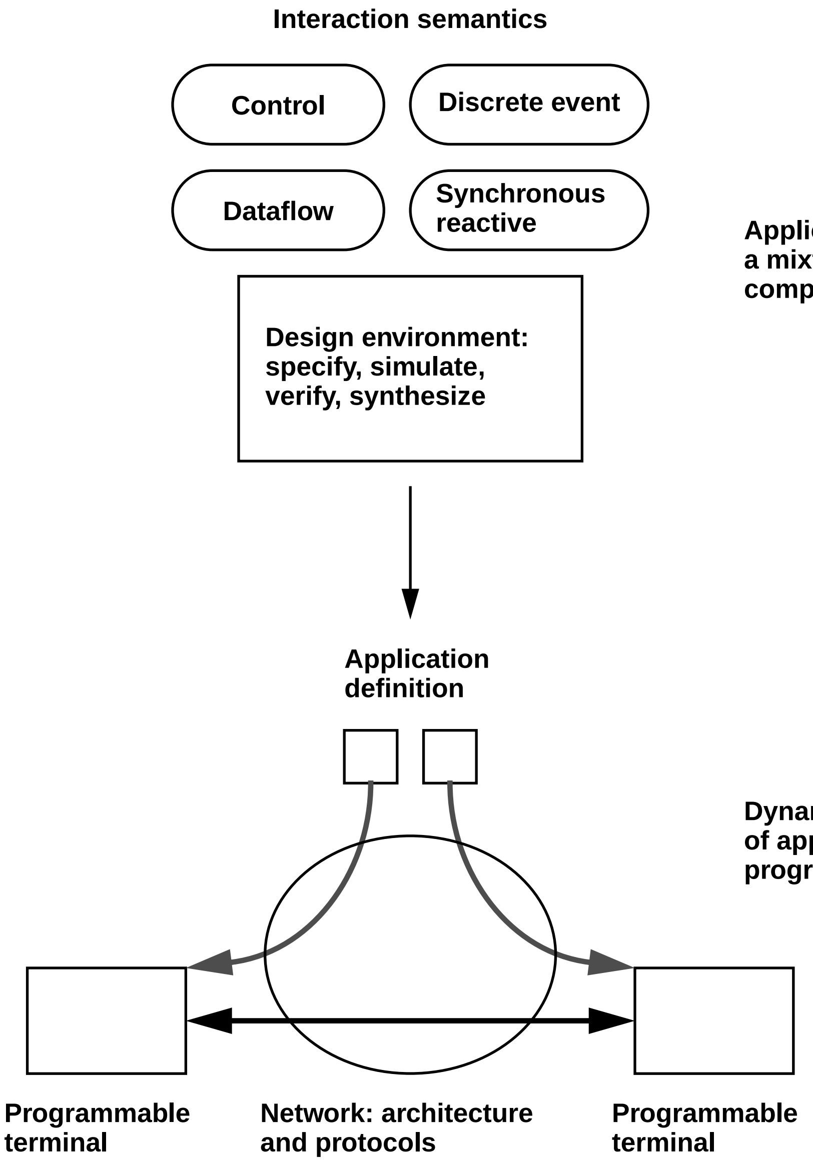
Problem: relatively few user-to-user applications

Objective: proliferation of user-to-user applications

Design: rapid prototyping methodology

Deployment: dynamic network deployment

Design and deployment



Interaction semantics

Control

Discrete event

Dataflow

Synchronous reactive

Design environment: specify, simulate, verify, synthesize

Application definition

Programmable terminal

Network: architecture and protocols

Programmable terminal

Application design using a mixture of models of computation

Dynamic network deployment of application definition to programmable terminals

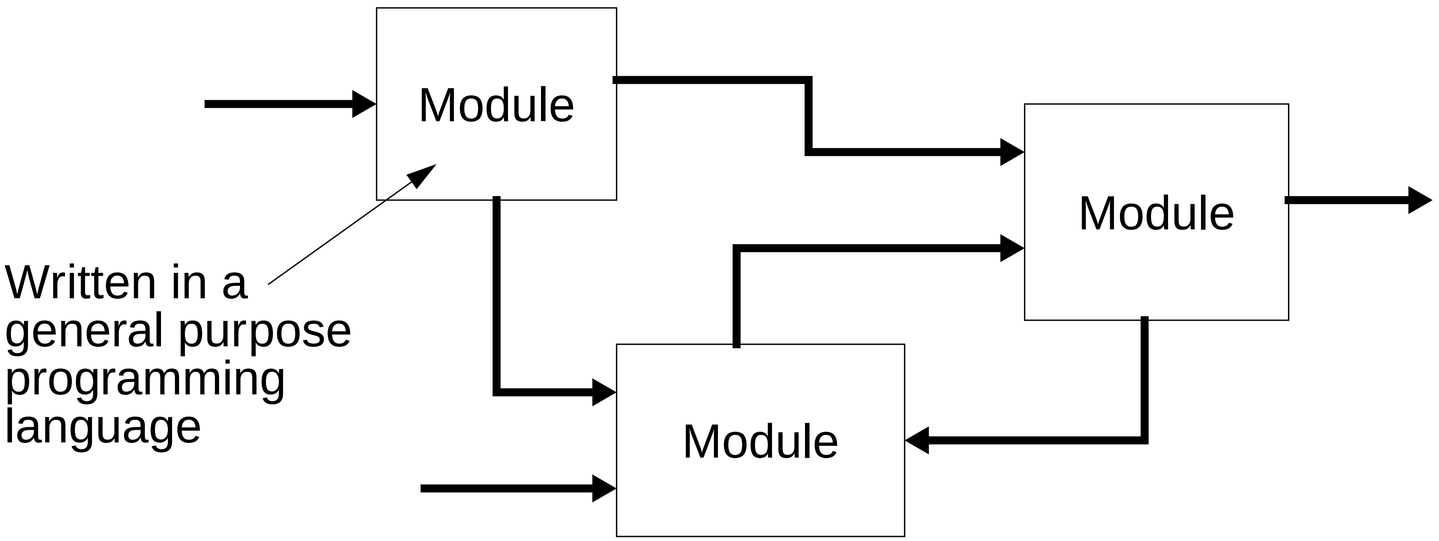
Rapid prototyping methodology

Design environment: specification, modeling, synthesis, and verification

Models of computations:

- **Dataflow, discrete event, synchronous/reactive, finite state machine**

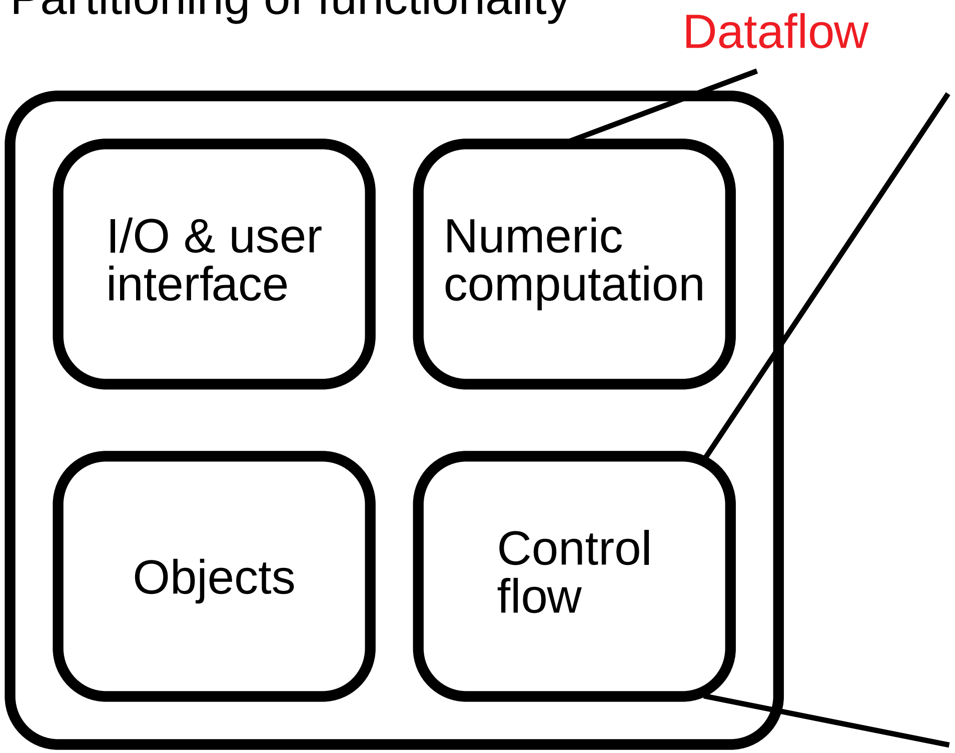
Interacting modules



Framework

Ptolemy’s heterogeneous approach: mixing domain-specific models of computation

Partitioning of functionality



Sequencing of operations

- Handling GUI events
- Networking protocols

Focus on:

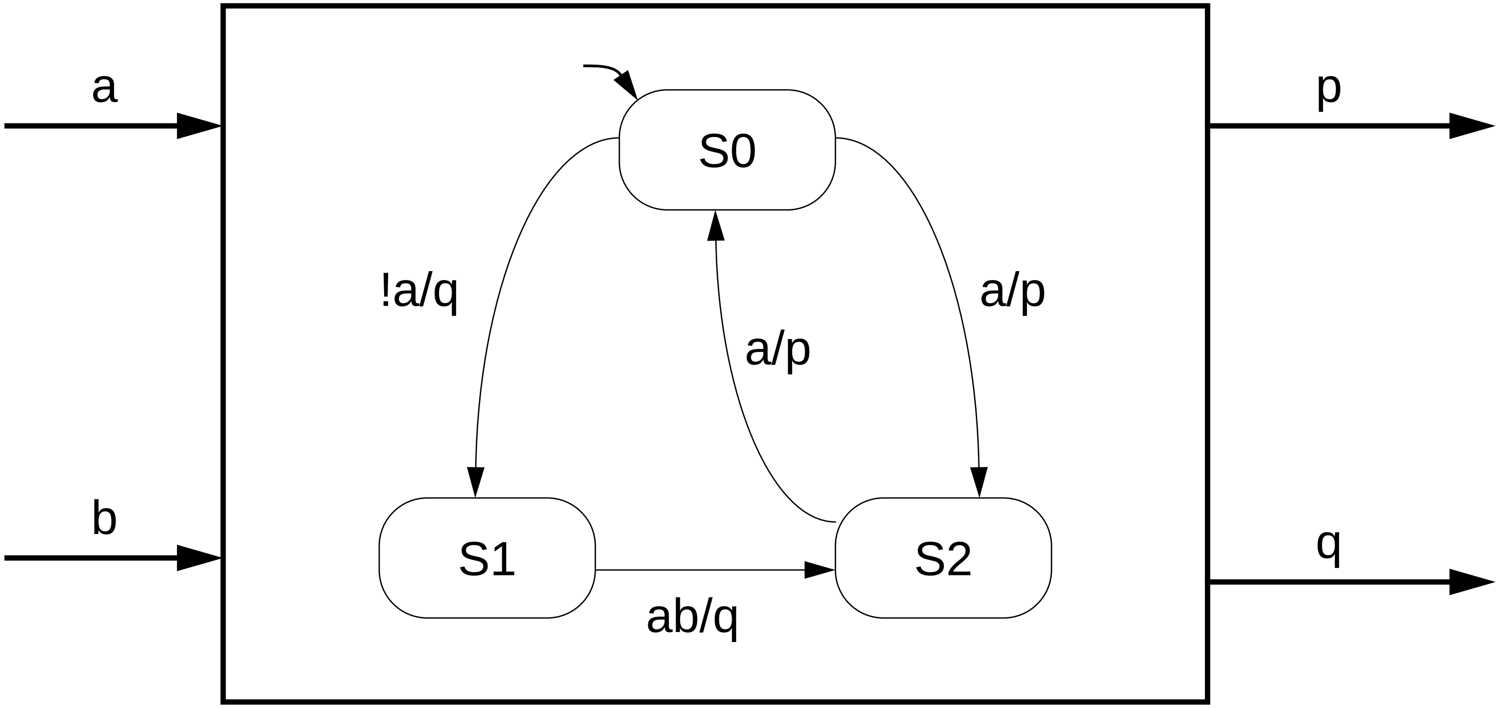
1. Good formal model for specifying control flow
   - state/event based

2. Combine control with rest of system

Finite state machine (FSM)

Advantages: **intuitive, formal mathematical theory**

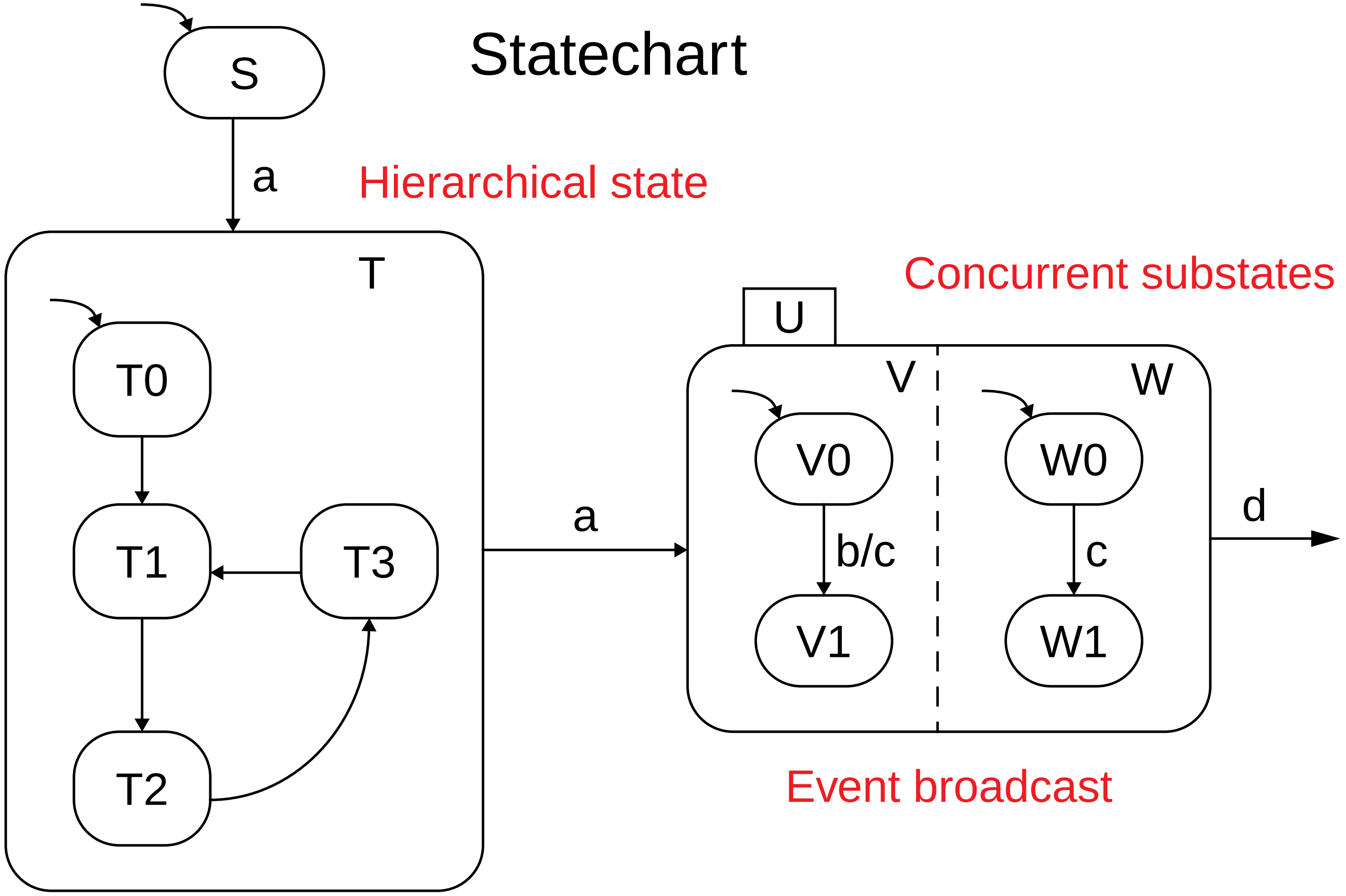
Weaknesses: **flat, sequential**



Organizing complex state space

Solution: add hierarchy and concurrency to basic FSM

- **Statechart, Argos, Esterel**



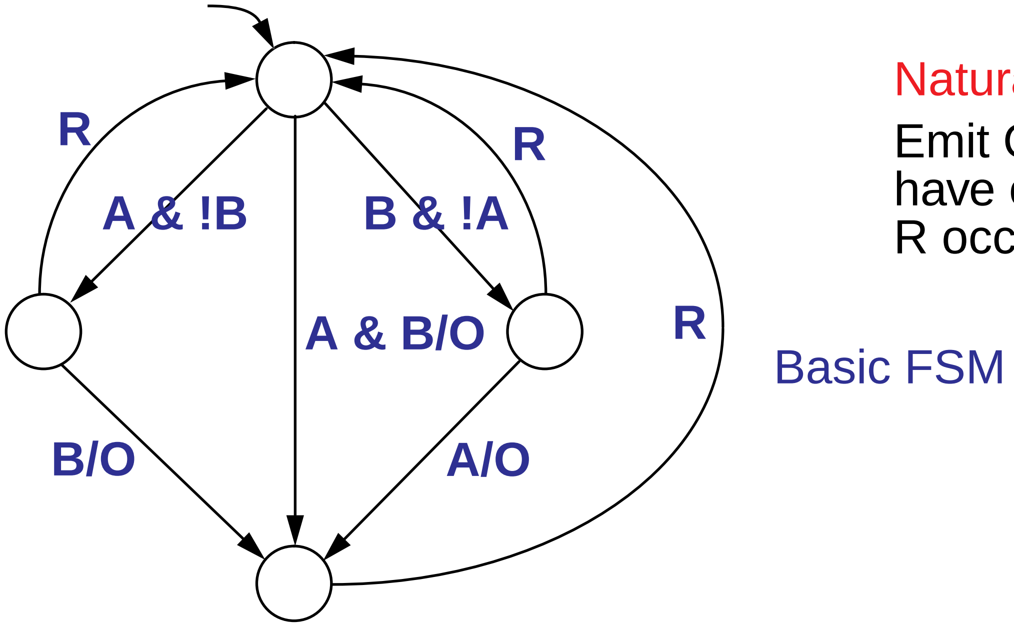
Statechart

Hierarchical state

Concurrent substates

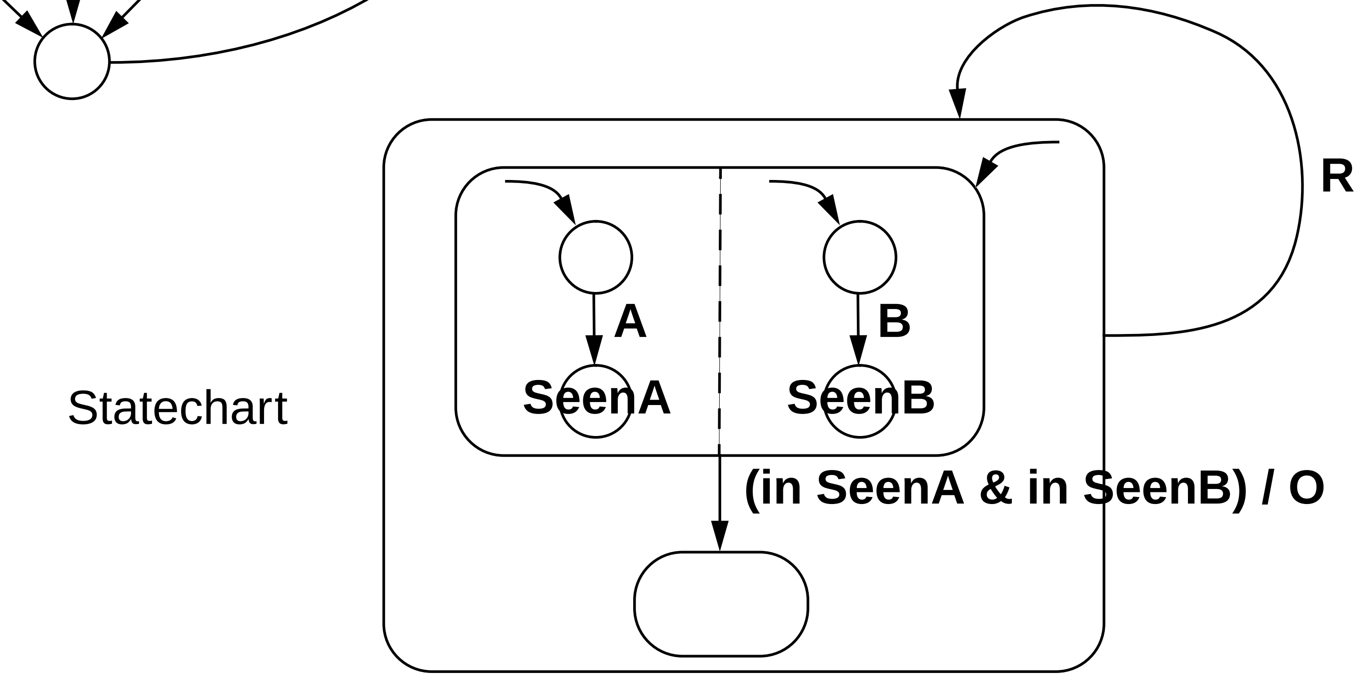
Event broadcast

Statechart example



Natural language specification:

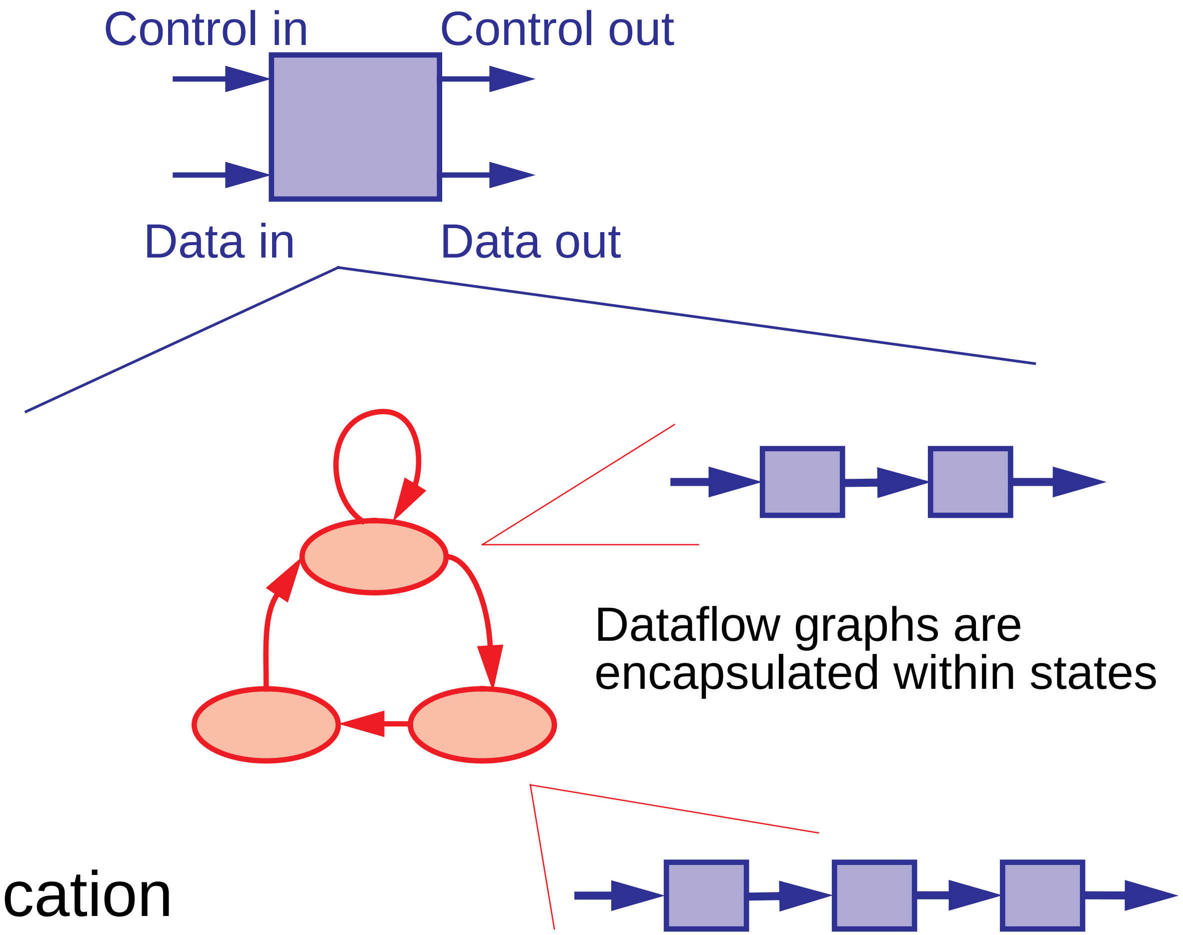
Emit O whenever both A and B have occurred. Reset whenever R occurs.



Mixing control with dataflow

Embed dataflow graphs in states:

- **Switching between modes of operation**

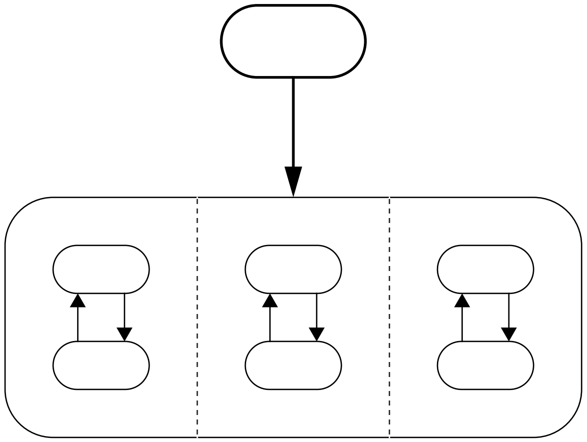


Dataflow graphs are encapsulated within states

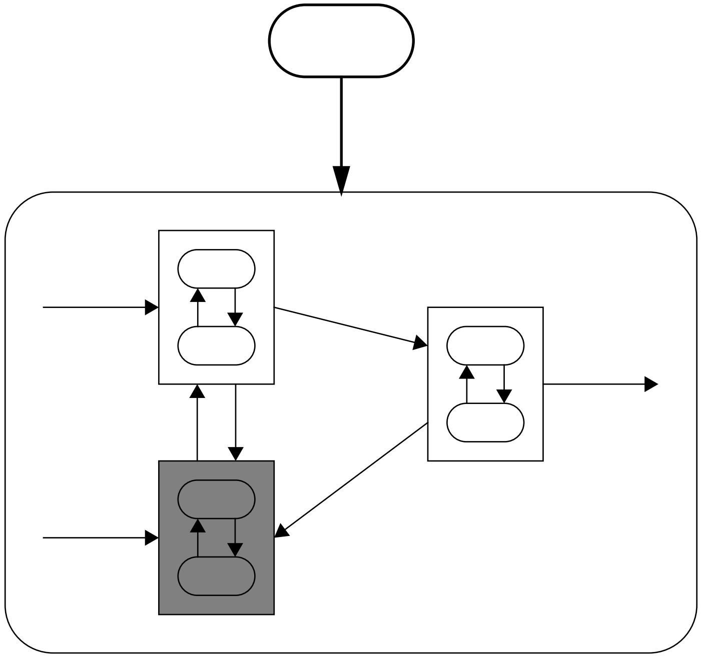
Specification

Nesting FSM and concurrency models

- **Concurrency and hierarchical FSM are orthogonal semantic properties**
- **Hierarchical nesting of FSM and concurrency models (dataflow, synchronous/reactive) subsumes variants of Statecharts: *charts**



Event
broadcast

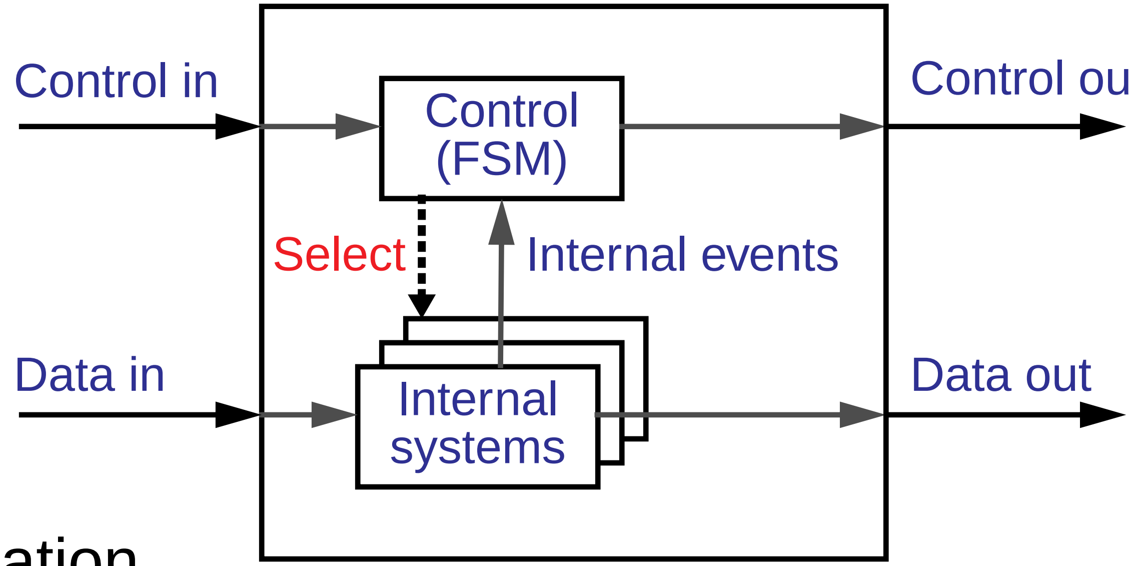


Point-to-point

Implementation of hierarchical FSM

Select is the key primitive

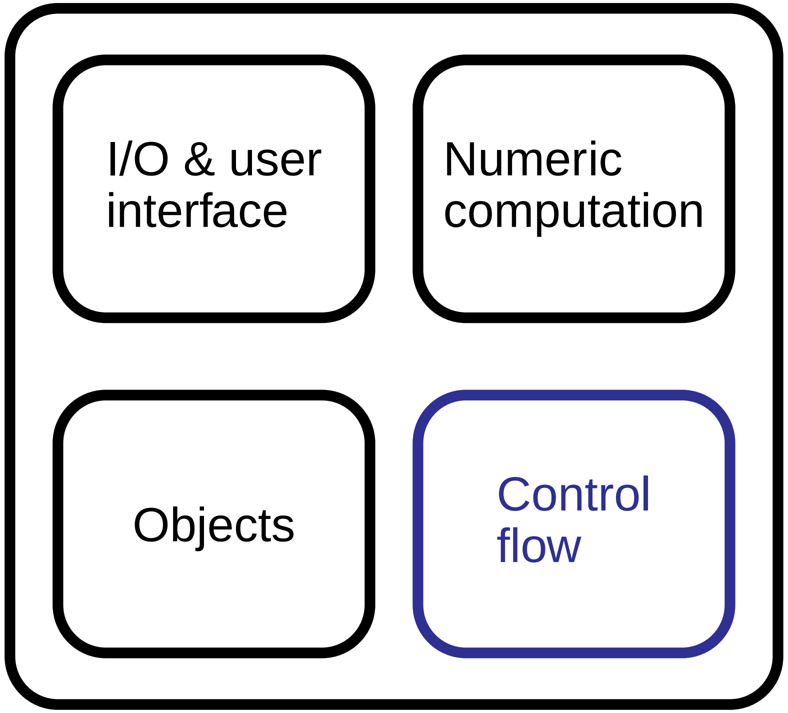
- **A block is replaced by one of a set of internal systems.**
- **The choice of internal system is controlled dynamically by a FSM.**



Implementation

Summary of rapid prototyping methodology

- *charts good for specifying complex control and combining control and dataflow
- Semantics defined
- Implementation in Ptolemy: simulation; code generation



Rapid deployment: motivation

Two obstacles to rapid deployment of new networked applications:

- **Architectural constraints: application functionality implemented by the network**
- **Standardization at application level**

Major economic barrier to deployment of user-to-user applications:

- **Network externality problem: early users derive little benefit from the applications**

Solution

Applications defined in user terminals, and increasingly in software.

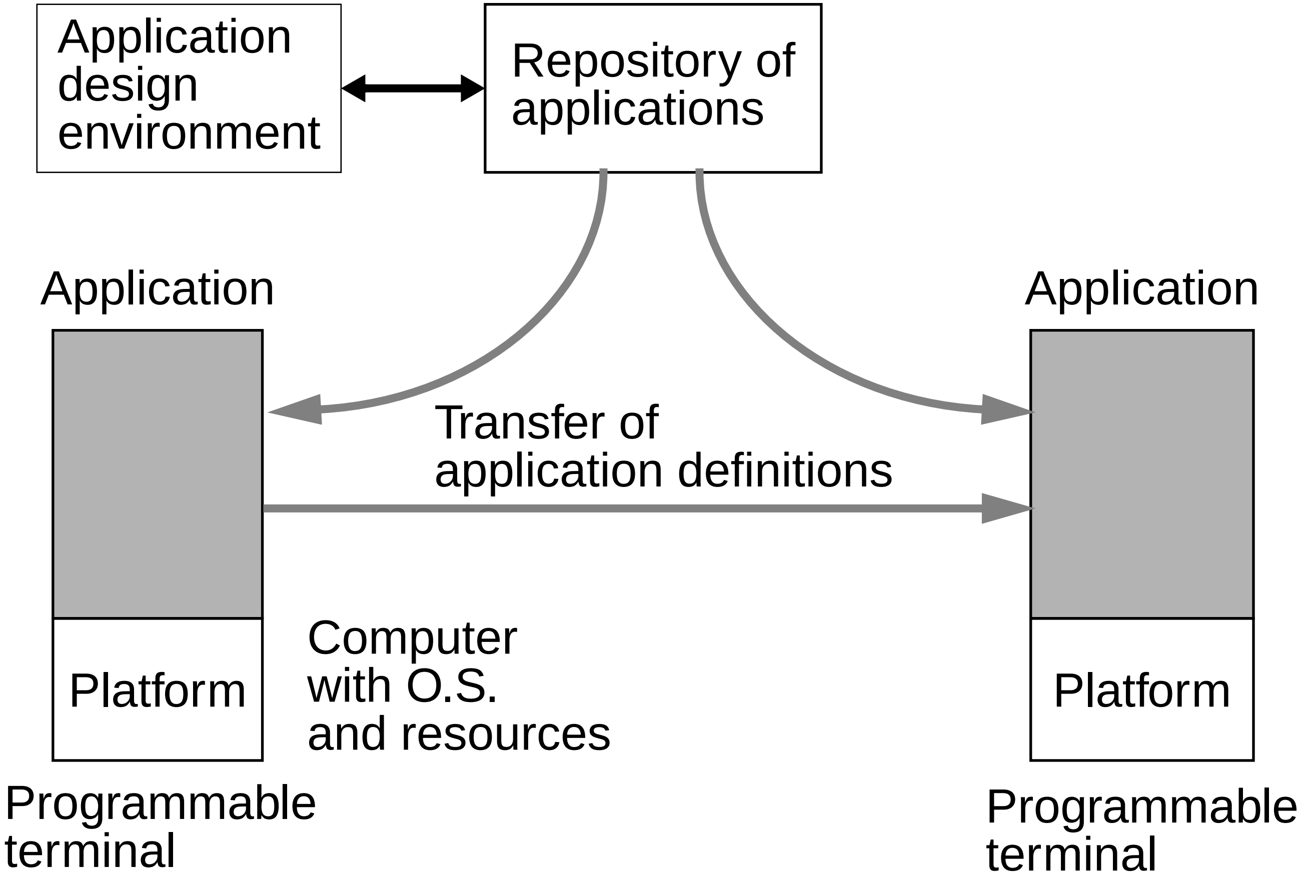
Network deployment: software-defined applications can be distributed via the network.

- Web browsers, document viewers, audio players
- Manual file transfer and installation
- Have to anticipate the need

Dynamic network deployment: transfer application definition at session establishment (and during the session).

The dynamic network deployment approach

- Platform
- Application definition language
- Protocol for transfer of application definitions


Application design environment
Repository of applications
Application
Application
Transfer of application definitions
Platform
Computer with O.S. and resources
Platform
Programmable terminal
Programmable terminal

Discussion

Limit standardization to infrastructure elements

Downloadable software definition of remainder of application functionality

Bypass network externality problem: a community of interest consisting of all networked platforms

Similar ideas:

- Postscript, Telescript agents, Java applets, MSDL
- File servers on LANs

Issues

Security:

- Application definition language must be a high-level language with restricted functionality
- Authentication of trusted sources

Hardware/O.S. independence ==> high-level language

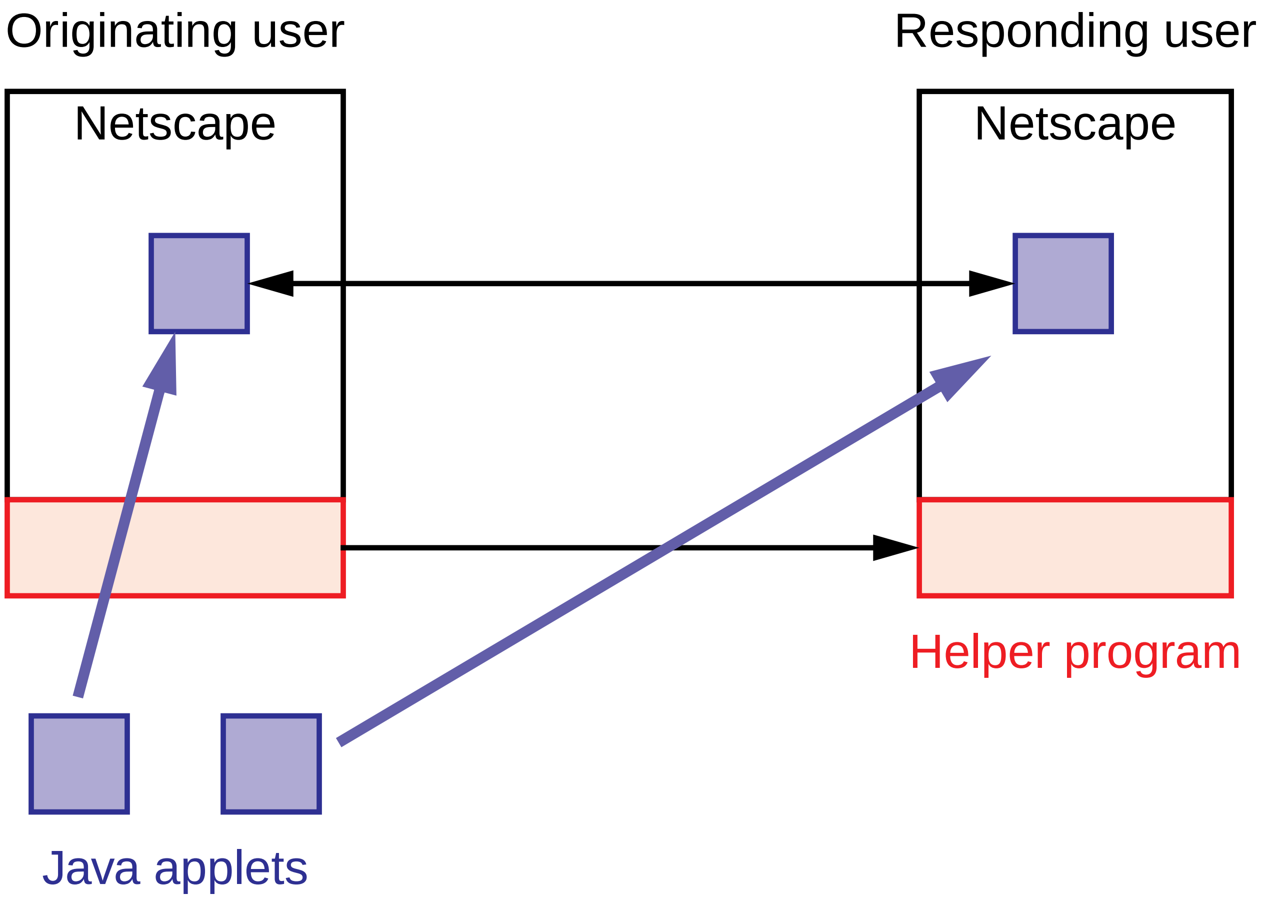
Performance:

- Session establishment time: download time, interpretation/compilation
- Run-time: interpretation overhead

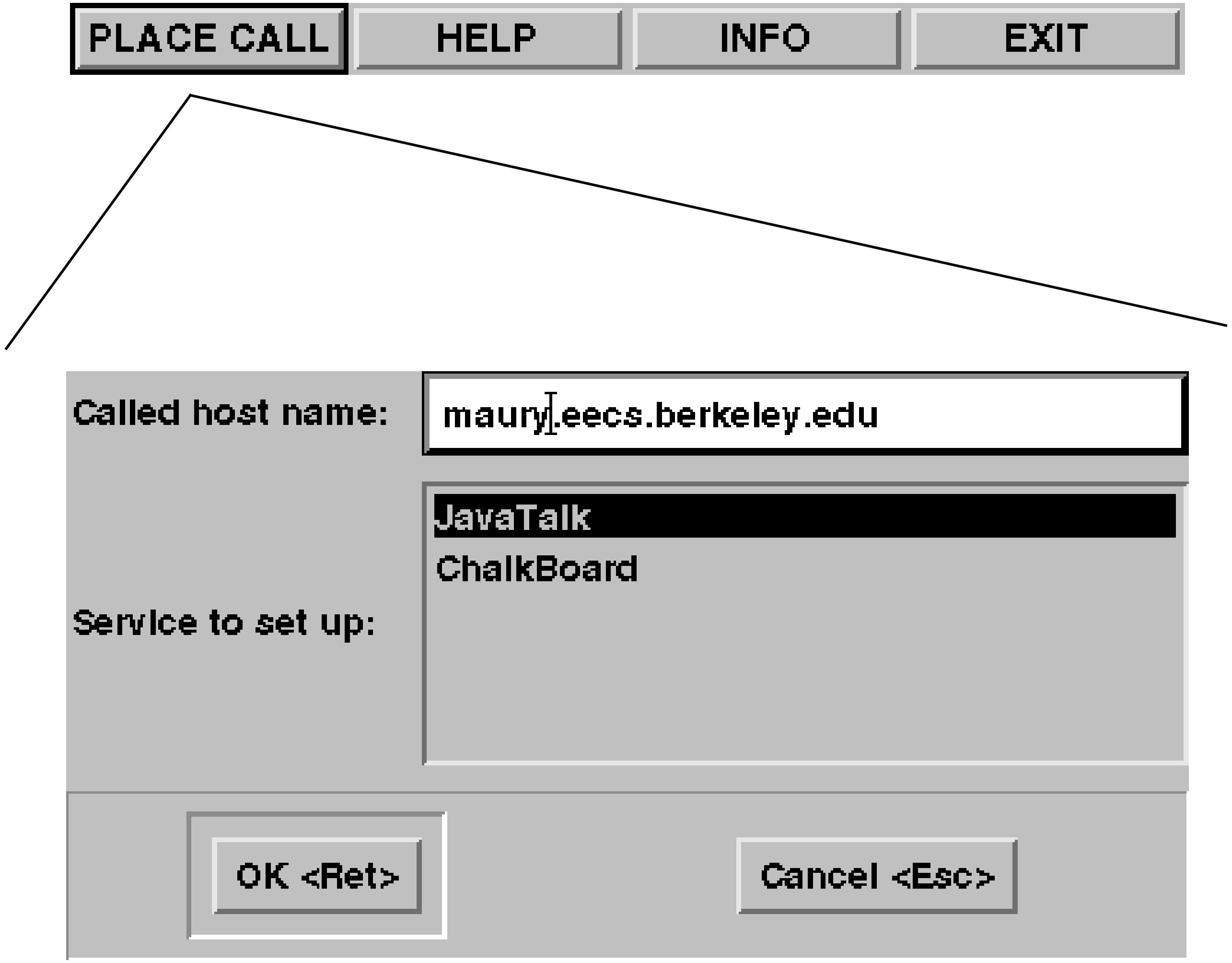
Pricing and charging; licensing; learning to use

Prototype based on Java and WWW

- Application definition language: Java
- Platform: Netscape + helper program
- Session establishment procedure



Originating user places a call

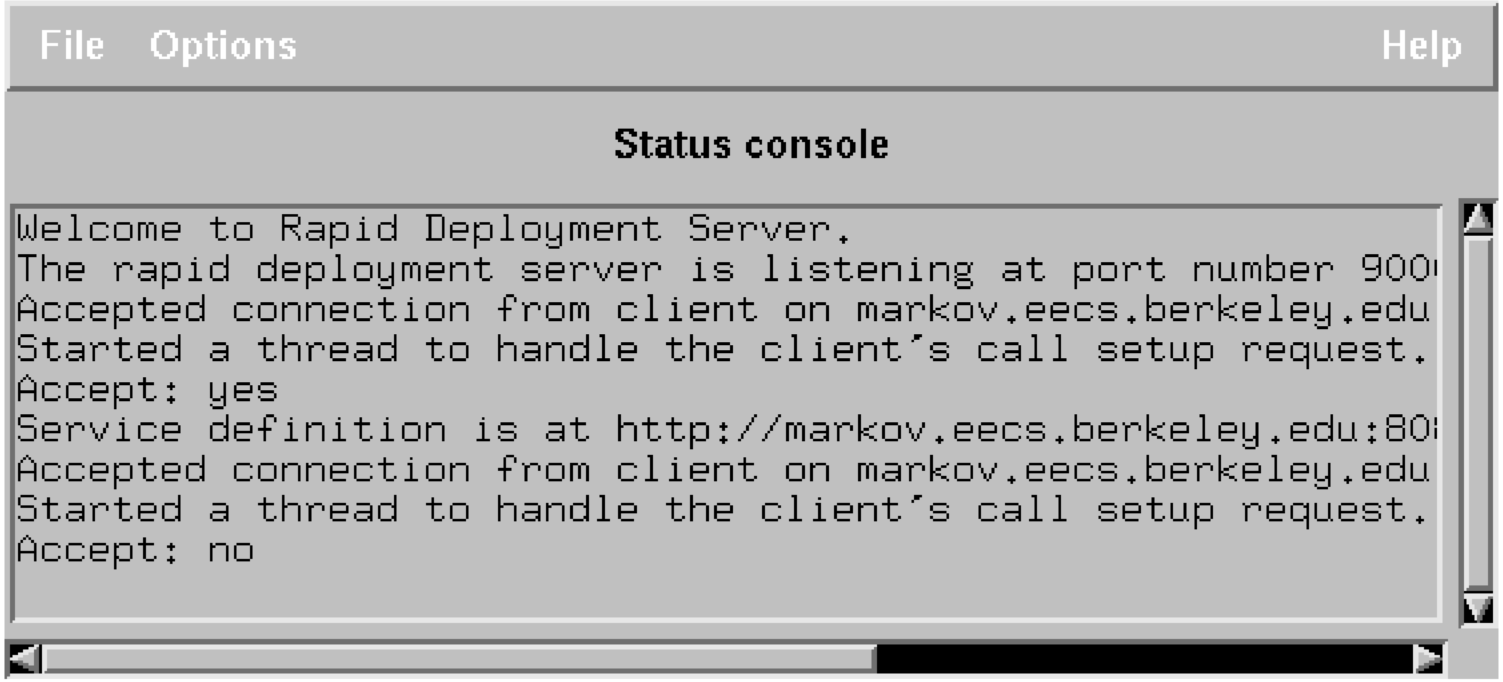


Called host name: maury.eecs.berkeley.edu

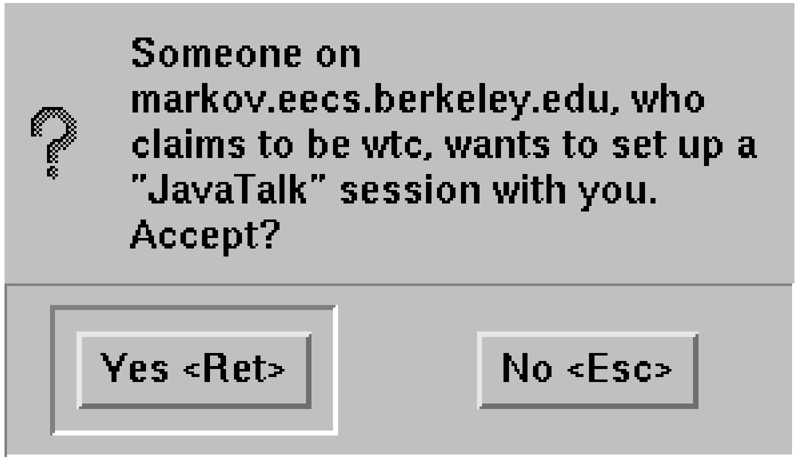
Service to set up:

- JavaTalk
- ChalkBoard

Alerting the responding user

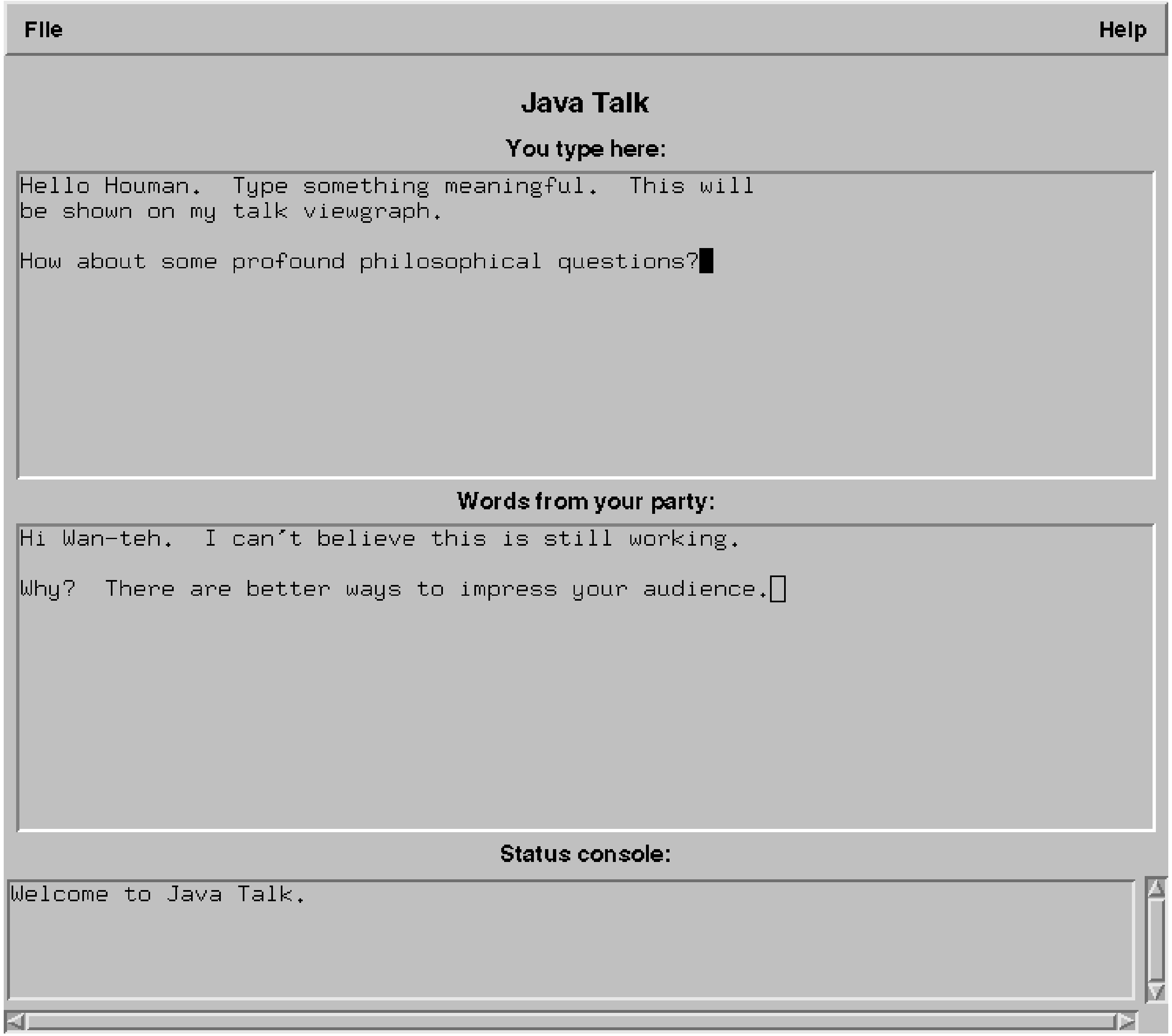


```
Welcome to Rapid Deployment Server.
The rapid deployment server is listening at port number 900
Accepted connection from client on markov.eecs.berkeley.edu
Started a thread to handle the client's call setup request.
Accept: yes
Service definition is at http://markov.eecs.berkeley.edu:80
Accepted connection from client on markov.eecs.berkeley.edu
Started a thread to handle the client's call setup request.
Accept: no
```



Someone on markov.eecs.berkeley.edu, who claims to be wtc, wants to set up a "JavaTalk" session with you. Accept?

JavaTalk



Java Talk

You type here:

```
Hello Houman.  Type something meaningful.  This will
be shown on my talk viewgraph.

How about some profound philosophical questions?
```

Words from your party:

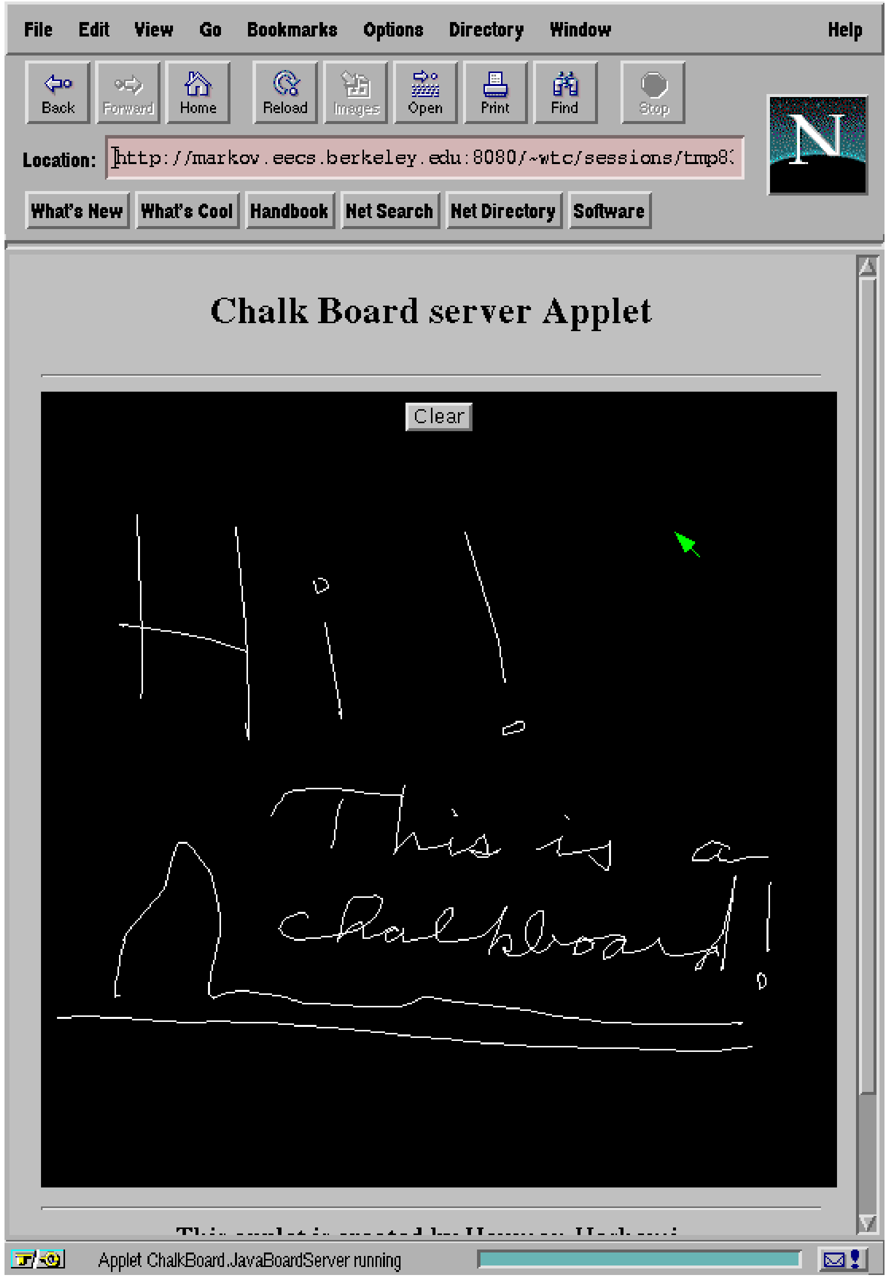
```
Hi Wan-teh.  I can't believe this is still working.

Why?  There are better ways to impress your audience.
```

Status console:

```
Welcome to Java Talk.
```

14.6 KB, uses a widget library of 18.2KB

Chalkboard



Location: http://markov.eecs.berkeley.edu:8080/~wtc/sessions/tmp8

Chalk Board server Applet

Hi!

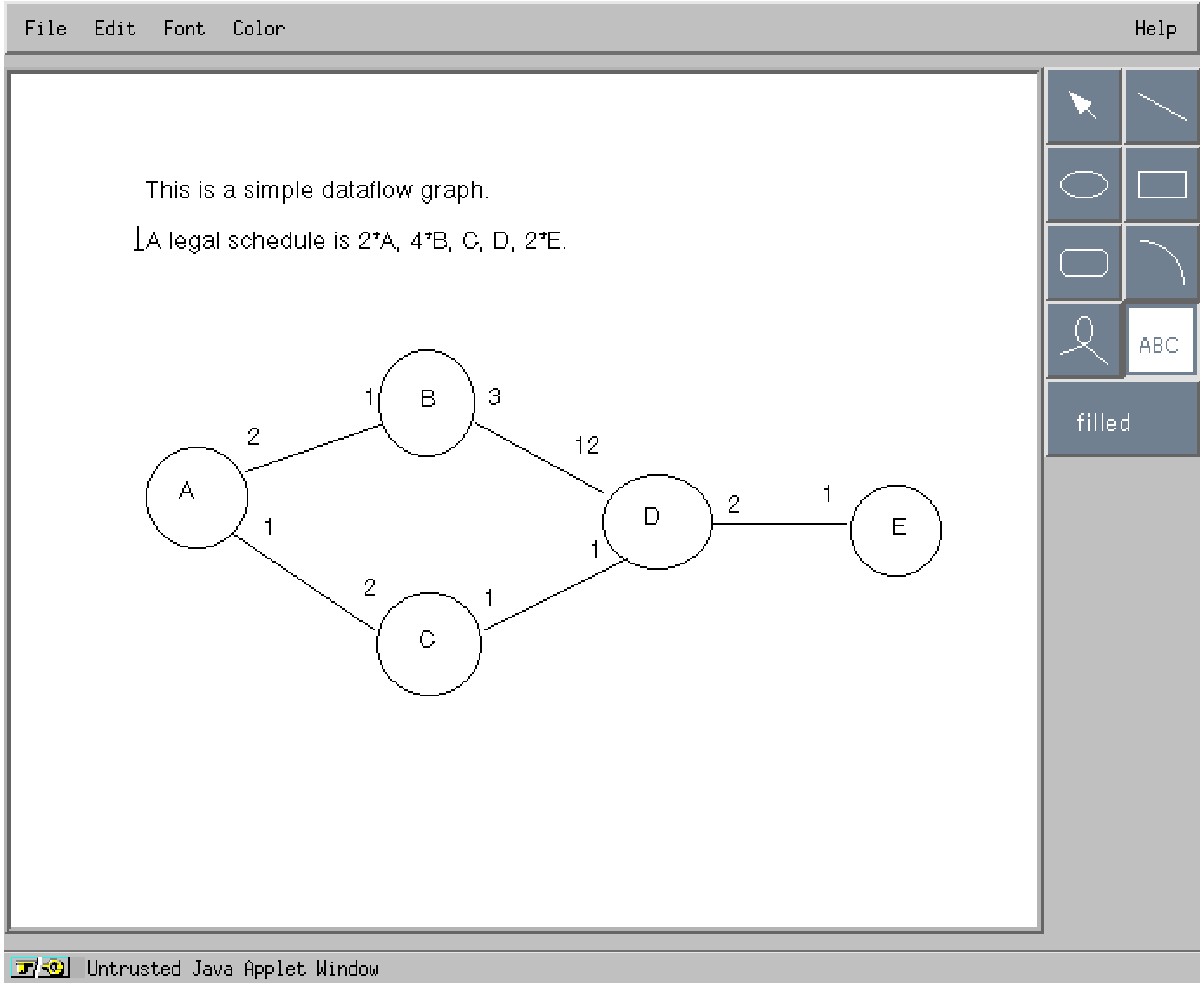
This is a chalkboard!

This applet is created by Hermann Hashemi

Applet ChalkBoard.JavaBoardServer running

10.8KB

Whiteboard (being developed)



This is a simple dataflow graph.

A legal schedule is 2*A, 4*B, C, D, 2*E.

Untrusted Java Applet Window

>35KB

Conclusions

Rapid prototyping: heterogeneous approach

- Systems consisting of DSP, control, GUI, etc.
- Combining domain-specific design styles

Dynamic network deployment

- Avoid standardization of actual application
- Limit network externality problems
- Security and high-speed networking are key
- Java-enabled Web browser as integrated environment for user-to-server and user-to-user applications
- Encourage a proliferation of innovative user-to-user applications

Future directions

Service configuration: terminal-network signaling

- **Heterogeneity** in networks, terminals, and applications
- **Negotiation** of processing and quality of service
- **Adapting** to changing conditions at run time
- Application of **transportable computation** (mobile programs)

Design environment:

- Design of **flexible**, **adaptive** applications