A Distributed Education Environment Based on Mathematica

Wolfgang Schreiner
Research Institute for Symbolic Computation (RISC-Linz)
Johannes Kepler University, A-4040 Linz, Austria

Wolfgang.Schreiner@risc.uni-linz.ac.at
http://www.risc.uni-linz.ac.at/people/schreine

Introduction

- Computer-supported education.
  - School and university education.
  - Vocational training.
- Pedagocical benefits.
  - Online problem solving facilities.
  - Multimedia presentation features.
- Remote access.
  - Distributed participation of remote audience.
  - “Virtual classrooms”.
- Today’s focus: *video-conferencing*.
  - Traditional extension of classroom model.
- Tomorrow: *distributed collaboration*.
  - Shared software tools.
  - Integratation of domain-specific applications.
- Our focus: mathematical education.
  - Scientific and engineering disciplines.
  - Application of *Mathematica*©.

Mathematica

© Wolfram Research, Inc.

- "A system for doing mathematics by computer".
  - Execution kernel and algorithmic library.
  - Multimedia hypertext documents (*notebooks*).
- Application in science and engineering.
  - Computational workbench.
  - Presentation and visualization facilities.
- Commercial system.
  - No access to source code.
  - No modification of internal operation.

How to integrate Mma into a distributed environment?

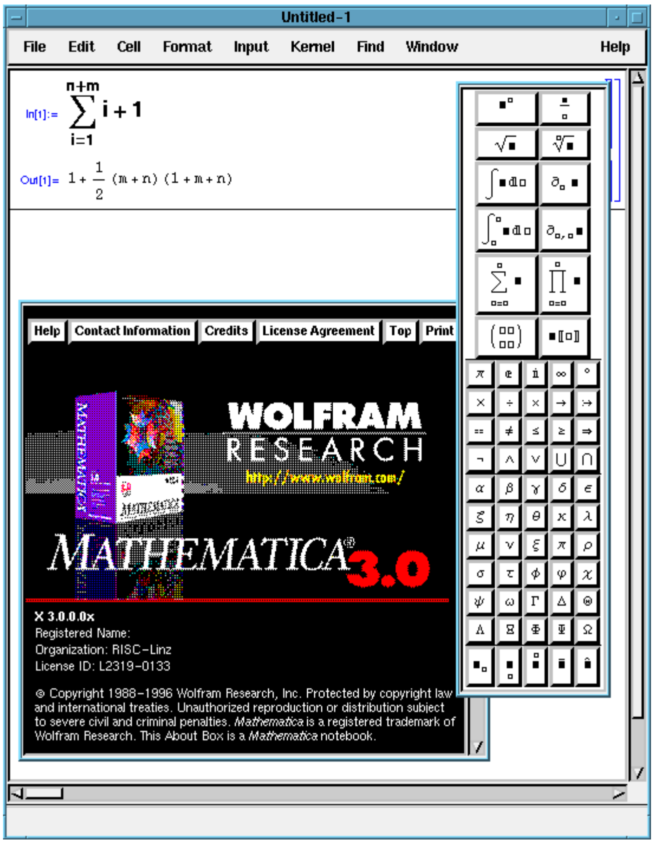
Mathematica Building Blocks

- Kernel and library

```
In[1]:= Sum[i+1, i, 0, n+m]

                       (m + n) (1 + m + n)
Out[1]= 1 + m + n + -------------------
                               2
```

- Notebook frontend



Education Scenario

Classroom lectures based on Mma.

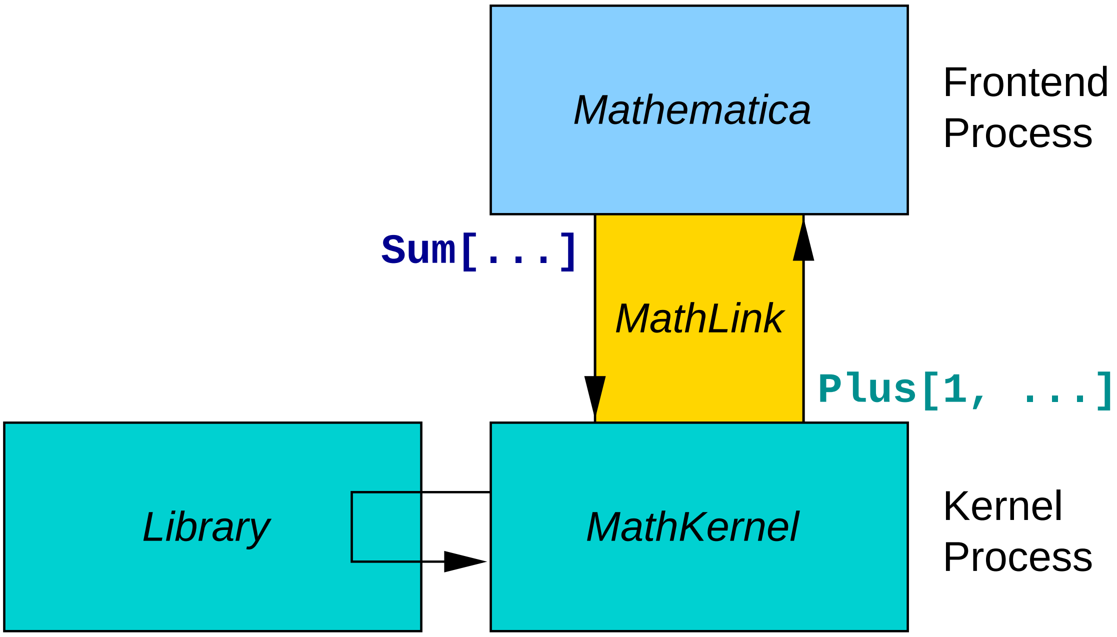
- Lecturer presents Mma notebooks on her computer.
  - Scrolling of document.
  - Opening and closing of cells.
  - Triggering execution of active contents.
- Overhead panel projects computer screen to audience.
  - Students listen to teacher.
  - Students watch projection.

How to work in distributed environment?

- Students exercise on local copies of notebooks.
  - Files have to be transferred.
  - Local systems have to be started up and initialized.
  - Results have to be returned.

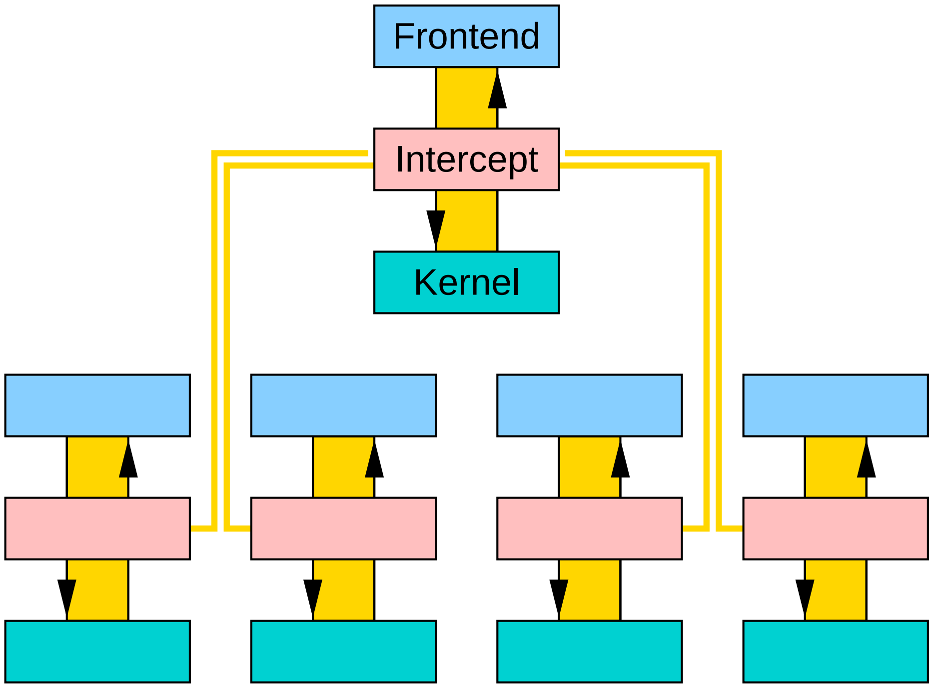
How to facilitate organization?

MathLink



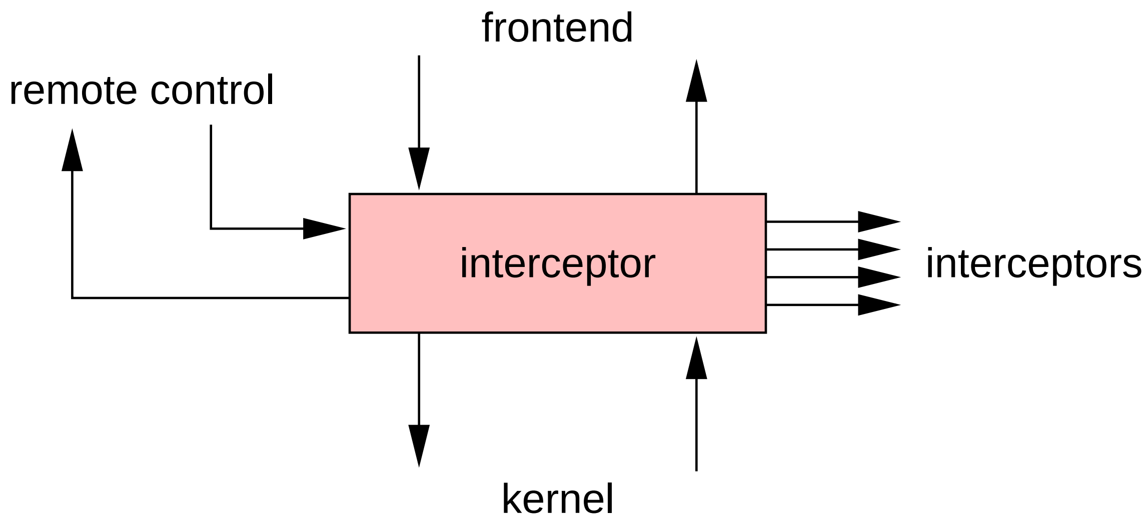
- Kernel and frontend are separate processes.
- *MathLink* for communication.
  - MMa communication protocol.
  - C library for program development.
  - Portable exchange of MMa data between processes.
- Basis for networked applications.

A Distributed Mathematica Environment



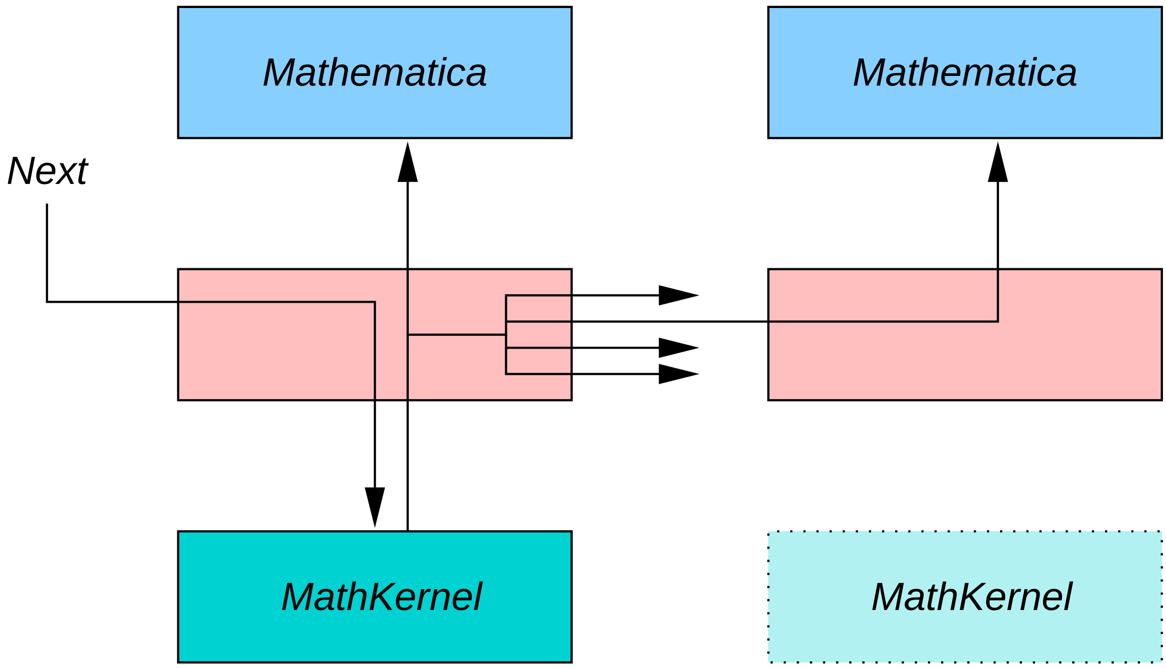
- Each session participant has a local Mma instance.
- Insert *interceptor* process between frontend and kernel.
- Multicast teacher's activities to student systems.

Interceptor



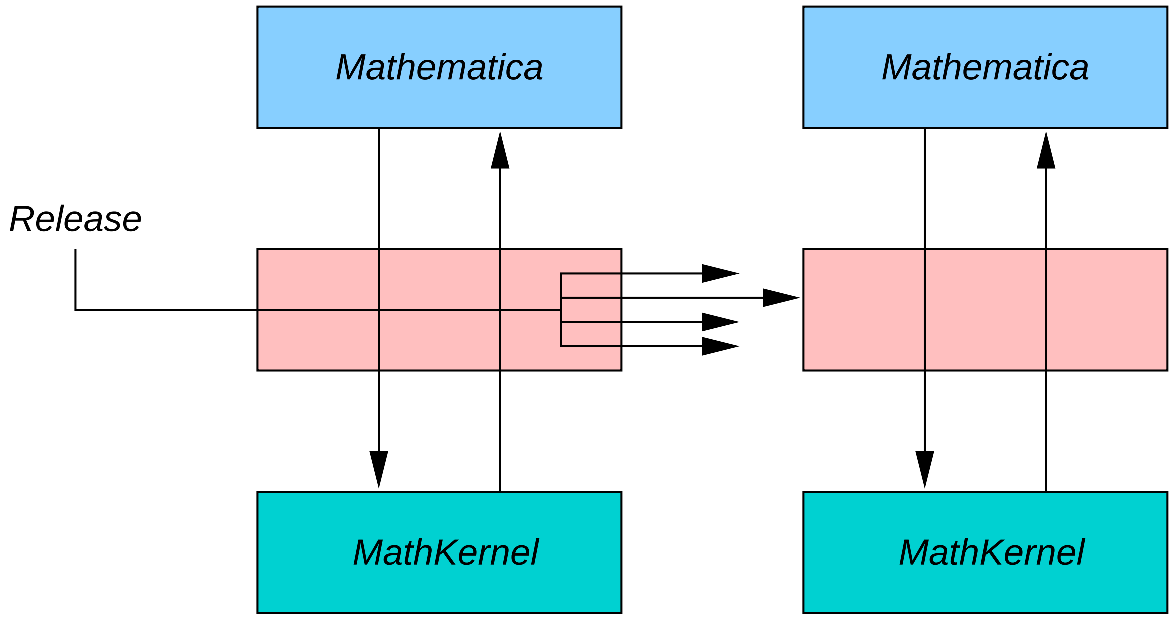
- Intercept frontend/kernel communication.
- Transparent operation:
  - Presents itself to frontend as a *kernel*.
  - Presents itself to kernel as a *frontend*.
- Collaboration with other interceptors.
  - *Multicasting* facilties.
- Controlled by "remote control" interface.
  - User input augments frontend actions.

Operation — Controlled Mode



- Notebook loaded on teacher's computer.
- Notebook multicast to audience.
- Navigation events multicast to audience.
- Audience frontends controlled by lecturer.

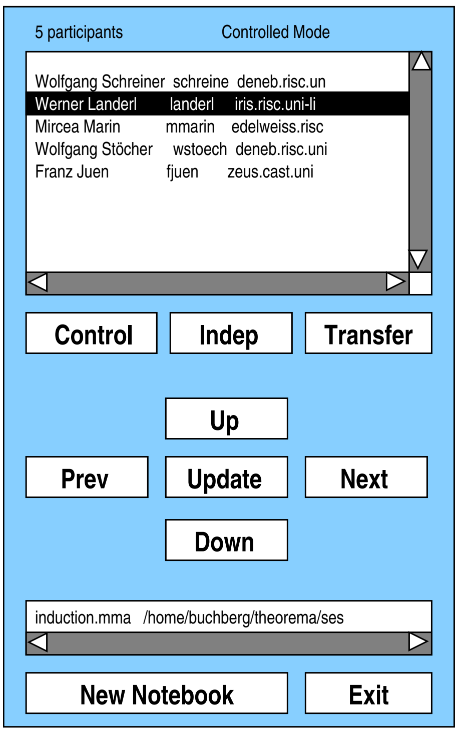
Operation — Independent Mode

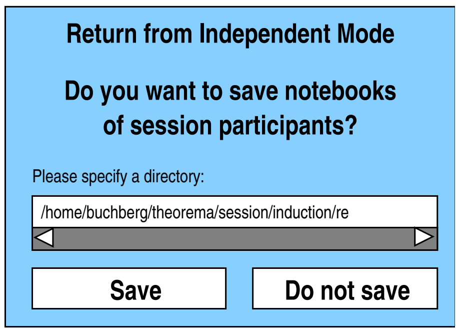


- Audience frontends are released.
- Audience kernels are started.
- Independent execution of systems.
- Results are transferred back to lecturer.

Notebook Control

- Basic requirements:
  1. Frontend interaction must trigger messages to kernel.
  2. Kernel messages must be able to trigger frontend actions.
- First demand is not fulfilled by frontend!
  - Basic notebook interaction handled locally.
- Possible solutions
  - Embed control into notebook.
  - Extend frontend by control palette.
  - Introduce separate control interface.





Conclusions

- Design study.
  - Not competitor to special distance learning systems.
  - How to build distributed education environment based on a domain-specific software system.
- Various constraints.
  - Not imaginable to start development from scratch.
  - No access to internals of system.
  - Cope with limited control interfaces.
- May still cover interesting scenarios.
  - Audio/video support is important but an orthogonal issue.
- Being implemented in a diploma thesis.