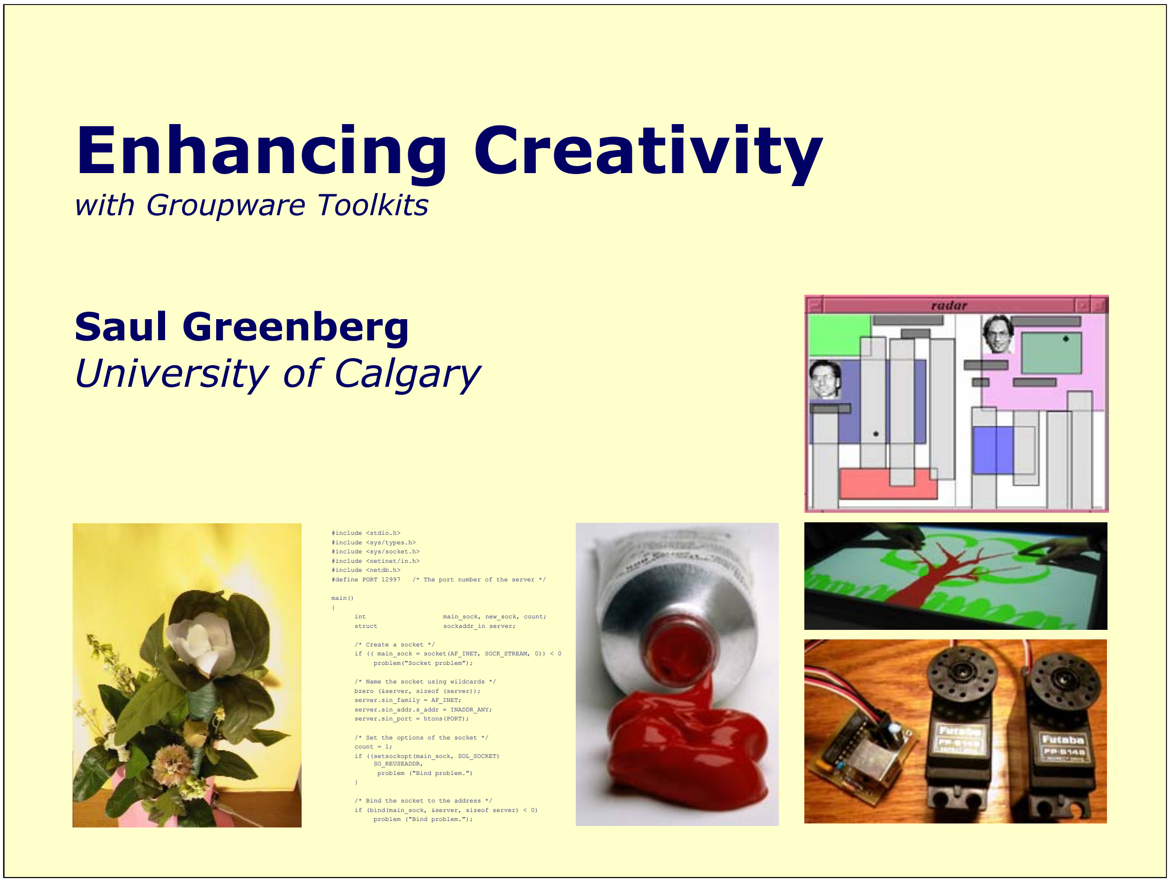


Enhancing Creativity

with Groupware Toolkits

Saul Greenberg

University of Calgary

```
#include <stdio.h>
#include <sys/types.h>
#include <sys/socket.h>
#include <netinet/in.h>
#include <netdb.h>
#define PORT 12997   /* The port number of the server */

main()
{
      int                    main_sock, new_sock, count;
      struct                 sockaddr_in server;

      /* Create a socket */
      if (( main_sock = socket(AF_INET, SOCK_STREAM, 0)) < 0
           problem("Socket problem");

      /* Name the socket using wildcards */
      bzero (&server, sizeof (server));
      server.sin_family = AF_INET;
      server.sin_addr.s_addr = INADDR_ANY;
      server.sin_port = htons(PORT);

      /* Set the options of the socket */
      count = 1;
      if ((setsockopt(main_sock, SOL_SOCKET)
           SO_REUSEADDR,
            problem ("Bind problem.")
      }

      /* Bind the socket to the address */
      if (bind(main_sock, &server, sizeof server) < 0)
           problem ("Bind problem.");
```

Good morning

•I am delighted to be here at <INSERT HERE>.

•I specialize in Real Time Groupware, where I look at both the technical and social aspects of groupware design and use

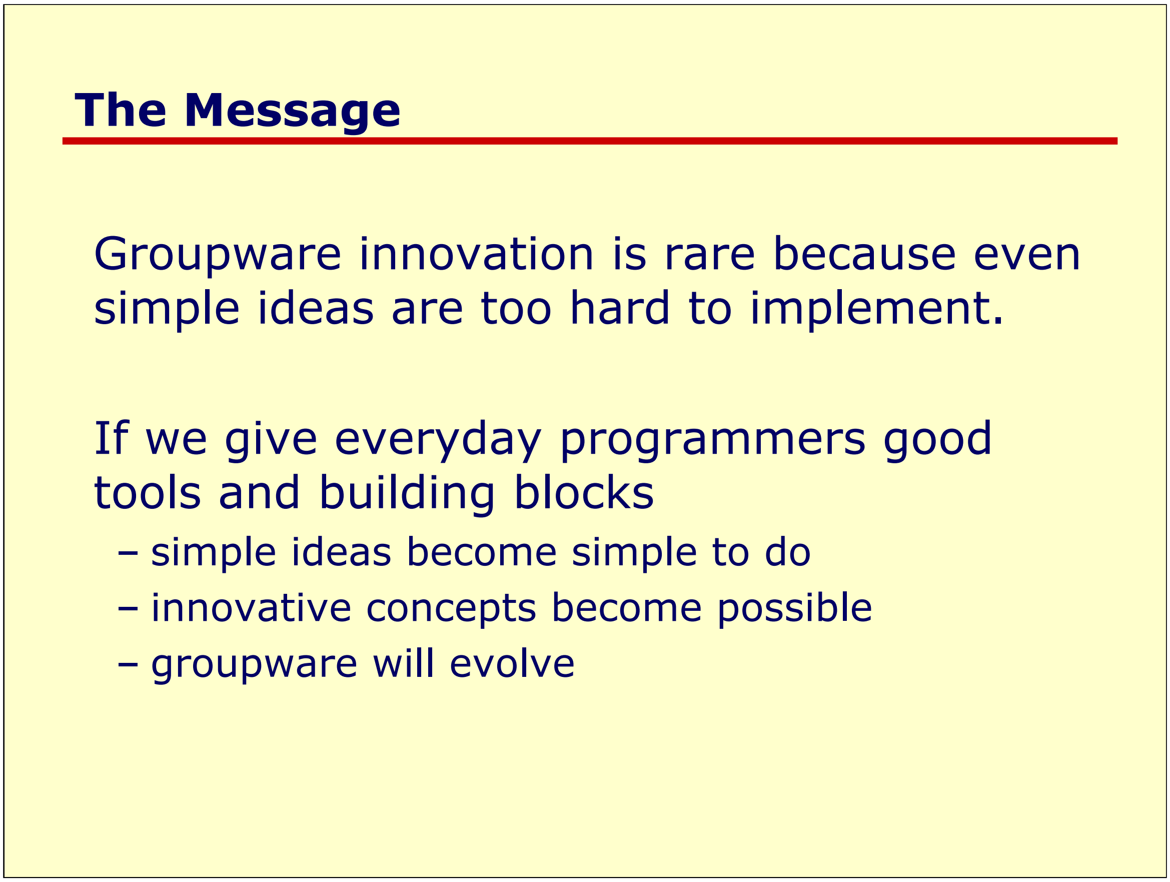
•Today, I will concentrate on the technical side of groupware design, where I will speak about “Enhancing Creativity with Groupware Toolkits”.

•However, you will also see that the point goes beyond groupware, and it should apply to any innovate area within interaction design.

This presentation handout approximates the lectures given at several places on Enhancing Creativity with Groupware Toolkits. A paper summarizing this paper is available as:

Greenberg, S. (2003) Enhancing Creativity with Groupware Toolkits. Invited keynote talk. Proceedings of the CRIWG ' 2003 9th International Workshop on Groupware (Sept 28 - Oct 2, Autrans, France), LNCS vol. 2806, 1-9, Springer-Verlag. Copy available at http://www.cpsc.ucalgary.ca/grouplab/papers/

While videos and demonstrations are not included here, many of the videos shown are available at www.cpsc.ucalgary.ca/grouplab/papers/videos/ and www.cpsc.ucalgary.ca/grouplab/phidgets/gallery/



The Message

Groupware innovation is rare because even simple ideas are too hard to implement.

If we give everyday programmers good tools and building blocks

- simple ideas become simple to do
- innovative concepts become possible
- groupware will evolve

The message that I will deliver over the next hour is that

<Read>

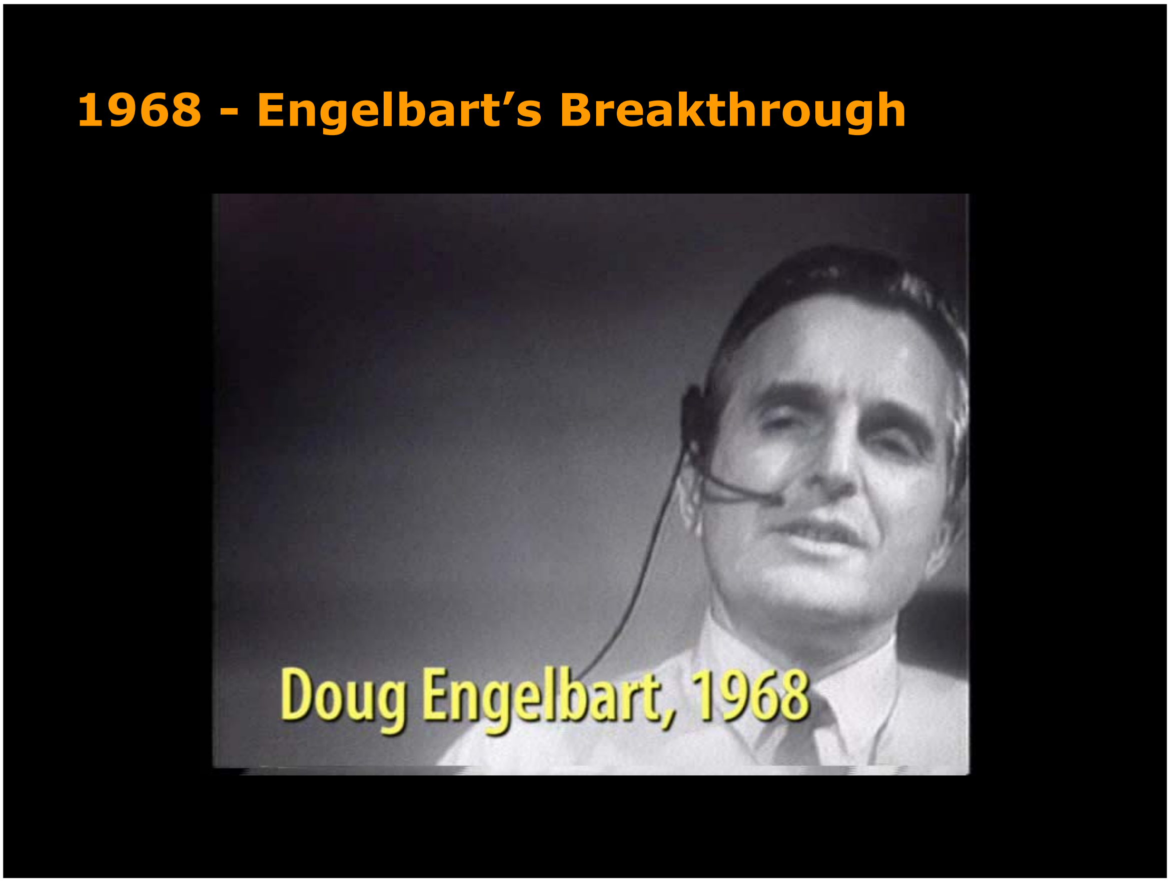
…as programmers generate new ideas and refine creative ideas of others

And that this evolution is instrumental to the ultimate success of groupware in the everyday world

I will deliver this message by

-first giving some historical motivation of why this is important.

-then going through several case studies of how particular toolkits, each reflecting research waves within particular groupware genres, have significantly enhanced the creativity of students within my laboratory.



1968 - Engelbart’s Breakthrough

For motivation, I’ll begin with Doug Engelbart’s 1968 historic demonstration of his NLS system.

Engelbart’s vision, implemented 35 years ago, showed computers as a means to augment human intellect by giving people

-desktop productivity tools for information creation,

-hypertext for knowledge organization and access,

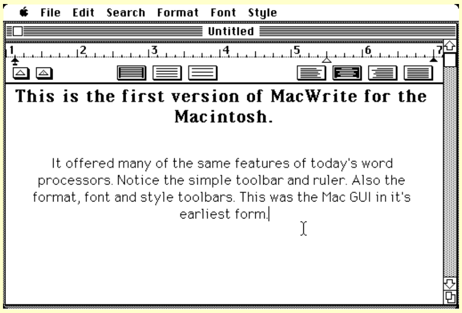
-and groupware for collaborative work.

<START VIDEO>

This clip extracts the groupware features.

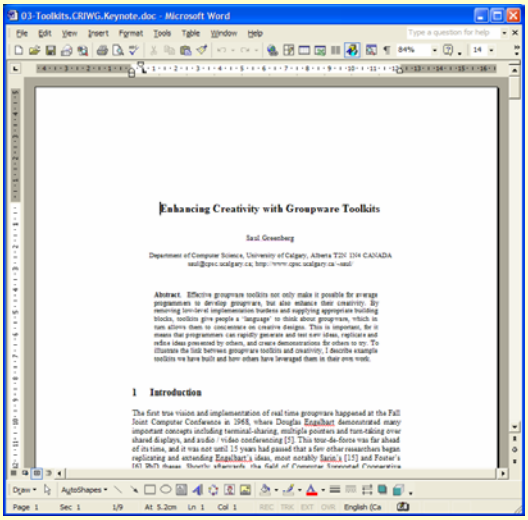
What happened since 1968?

Desktop productivity



This is the first version of MacWrite for the Macintosh.

It offered many of the same features of today's word processors. Notice the simple toolbar and ruler. Also the format, font and style toolbars. This was the Mac GUI in it's earliest form.



Enhancing Creativity with Groupware Toolkits

Saul Greenberg

Department of Computer Science, University of Calgary, Alberta T2N 1N4 CANADA
saul@cpsc.ucalgary.ca; http://www.cpsc.ucalgary.ca/~saul/

Abstract. Effective groupware toolkits not only make it possible for average programmers to develop groupware, but also enhance their creativity. By removing low-level implementation burdens and supplying appropriate building blocks, toolkits give people a 'language' to think about groupware, which in turn allows them to concentrate on creative designs. This is important, for it means that programmers can rapidly generate and test new ideas, replicate and refine ideas presented by others, and create demonstrations for others to try. To illustrate the link between groupware toolkits and creativity, I describe example toolkits we have built and how others have leveraged them in their own work.

1 Introduction

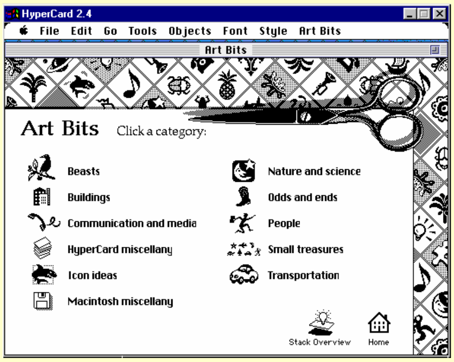
The first true vision and implementation of real time groupware happened at the Fall Joint Computer Conference in 1968, where Douglas Engelbart demonstrated many important concepts including terminal-sharing, multiple pointers and turn-taking over shared displays, and audio / video conferencing [5]. This tour-de-force was far ahead of its time, and it was not until 15 years had passed that a few other researchers began replicating and extending Engelbart's ideas, most notably Sarin's [15] and Foster's [6] PhD theses. Shortly afterwards, the field of Computer Supported Cooperative

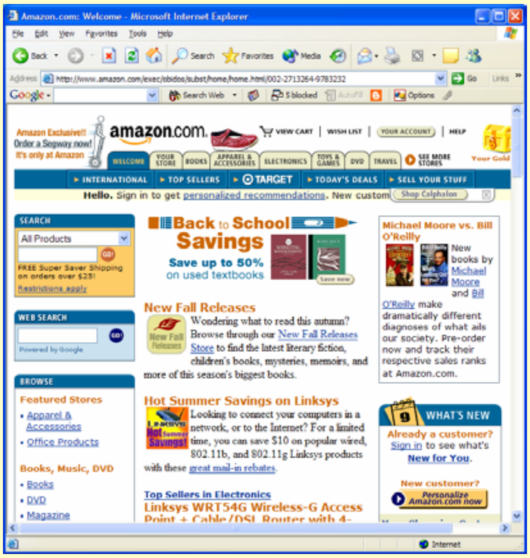
What happened with Engalbart's vision since 1968?

We have seen desktop productivity tools take off, with early word processors such as the Xerox Star and Apple's Macwrite creating the new genre of desktop publishing for the masses.

What happened since 1968?

Hypertext




Amazon.com: Welcome - Microsoft Internet Explorer
File Edit View Favorites Tools Help
Back Search Favorites Media
Address http://www.amazon.com/exec/obidos/subst/home/home.html/002-2713264-9783232
Go Links
Google Search Web 5 blocked AutoFill Options
Amazon Exclusive!! Order a Segway now! It's only at Amazon
amazon.com
VIEW CART | WISH LIST | YOUR ACCOUNT | HELP
WELCOME YOUR STORE BOOKS APPAREL & ACCESSORIES ELECTRONICS TOYS & GAMES DVD TRAVEL SEE MORE STORES
Your Gold
INTERNATIONAL TOP SELLERS TARGET TODAY'S DEALS SELL YOUR STUFF
Shop Calphalon
SEARCH
All Products
GO!
WEB SEARCH
GO!
Powered by Google
BROWSE
Featured Stores
Apparel & Accessories
Office Products
Books, Music, DVD
Books
DVD
Magazine
Save now
WHAT'S NEW
Personalize Amazon.com now
Internet

Hello. Sign in to get personalized recommendations. New custom

FREE Super Saver Shipping on orders over $25! Restrictions apply

Back to School Savings

Save up to 50% on used textbooks

New Fall Releases

Wondering what to read this autumn? Browse through our New Fall Releases Store to find the latest literary fiction, children's books, mysteries, memoirs, and more of this season's biggest books.

Hot Summer Savings on Linksys

Looking to connect your computers in a network, or to the Internet? For a limited time, you can save $10 on popular wired, 802.11b, and 802.11g Linksys products with these great mail-in rebates.

Top Sellers in Electronics

Linksys WRT54G Wireless-G Access Point + Cable/DSL Router with 4-

Michael Moore vs. Bill O'Reilly

New books by Michael Moore and Bill O'Reilly make dramatically different diagnoses of what ails our society. Pre-order now and track their respective sales ranks at Amazon.com.

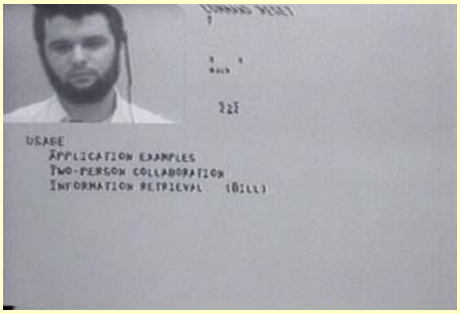
Already a customer? Sign in to see what's New for You.

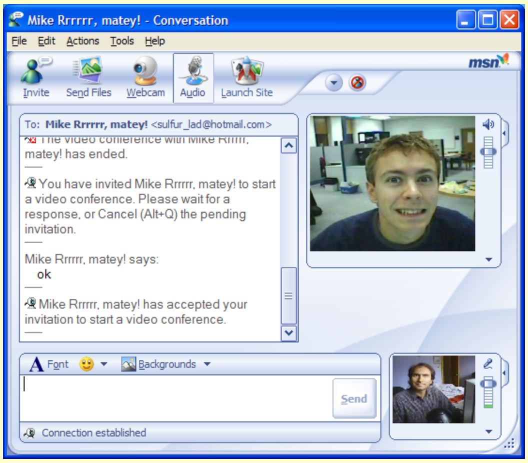
New customer?

Hypertext, originally popularized by the Apple Hypercard system, has exploded into the mainstream with the introduction of the WWW.

What happened since 1968?

Real time groupware





Mike Rrrrrr, matey! - Conversation

To: Mike Rrrrrr, matey! <sulfur_lad@hotmail.com>

The video conference with Mike Rrrrrr, matey! has ended.

You have invited Mike Rrrrrr, matey! to start a video conference. Please wait for a response, or Cancel (Alt+Q) the pending invitation.

Mike Rrrrrr, matey! says:
ok

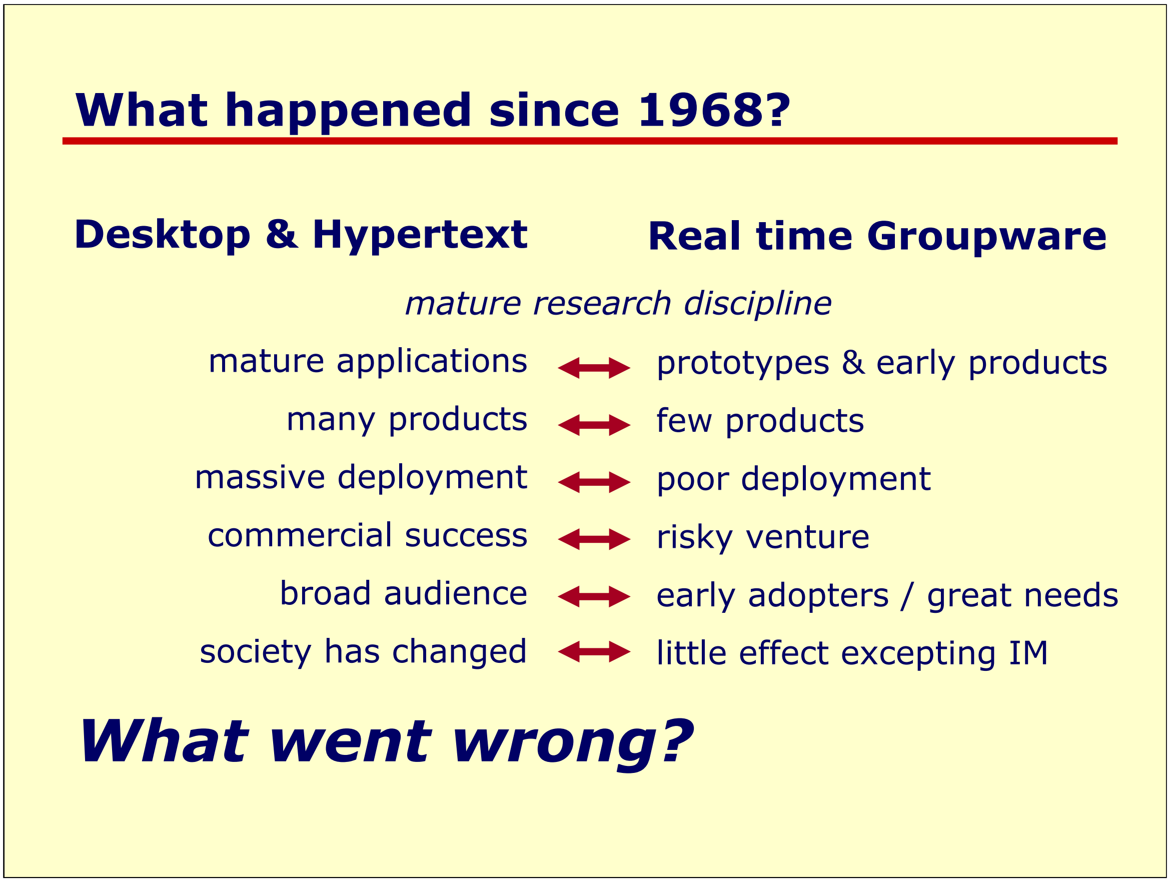
Mike Rrrrrr, matey! has accepted your invitation to start a video conference.

Connection established

Yet when we look at real time groupware, little has happened since Engelbart.

Our most popular real time conferencing tools are based on text chatting.

While we are now seeing them include video and other facilities, they tend to be unreliable, unimaginative, and awkward.



What happened since 1968?

Desktop & Hypertext		Real time Groupware
mature research discipline		
mature applications	↔	prototypes & early products
many products	↔	few products
massive deployment	↔	poor deployment
commercial success	↔	risky venture
broad audience	↔	early adopters / great needs
society has changed	↔	little effect excepting IM

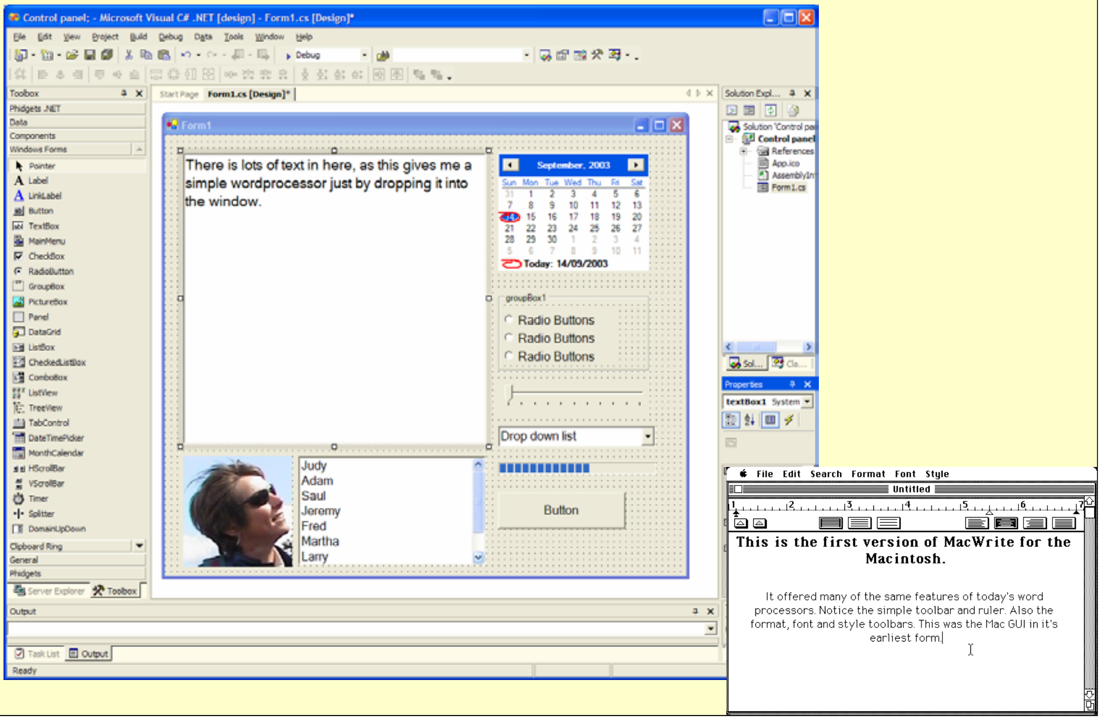
What went wrong?

This is a problem. When we compare desktop productivity and hypertext with Groupware, its clear that groupware has not succeeded.

<READ>

<ANIMATE>

Desktop Productivity



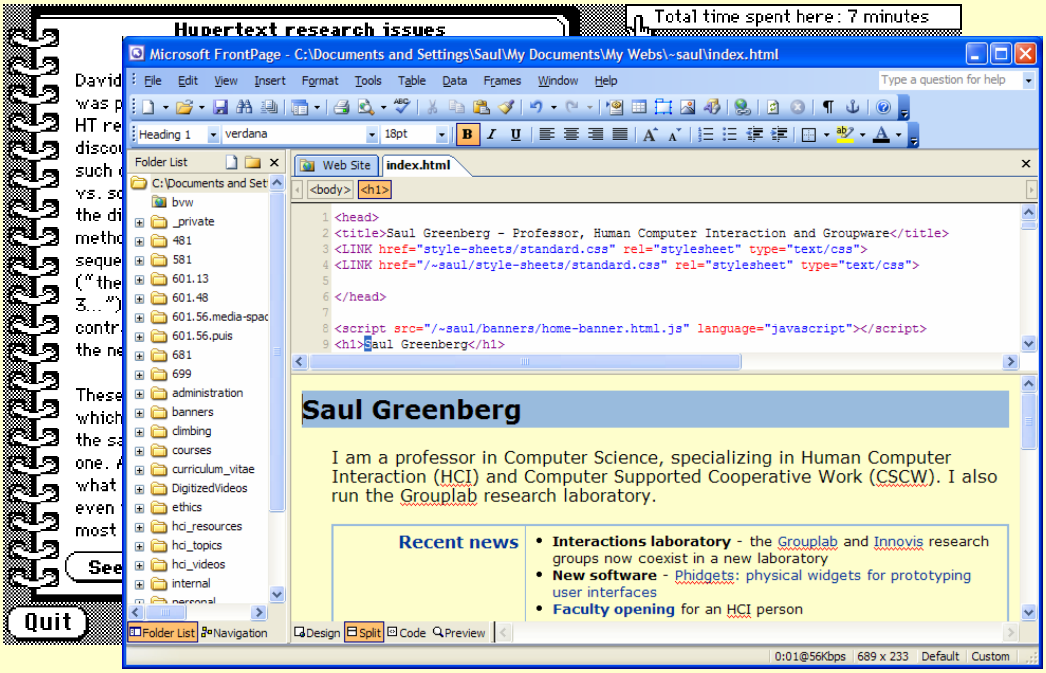
There is lots of text in here, as this gives me a simple wordprocessor just by dropping it into the window.

This is the first version of MacWrite for the Macintosh.

It offered many of the same features of today's word processors. Notice the simple toolbar and ruler. Also the format, font and style toolbars. This was the Mac GUI in it's earliest form|

- To understand the problem, lets look at these successes and failures from the programmer's point of view.
- Graphical User Interface toolkits, which have been around since the 1970s significantly eased a programmer's task of creating desktop applications.
- As a consequence, there are thousands, if not millions of developed applications – some commercial, some built for fun, some as student learning exercises.

Hypercard / HTML



```
<head>
<title>Saul Greenberg - Professor, Human Computer Interaction and Groupware</title>
<LINK href="style-sheets/standard.css" rel="stylesheet" type="text/css">
<LINK href="/~saul/style-sheets/standard.css" rel="stylesheet" type="text/css">

</head>

<script src="/~saul/banners/home-banner.html.js" language="javascript"></script>
<h1>Saul Greenberg</h1>
```

Saul Greenberg

I am a professor in Computer Science, specializing in Human Computer Interaction (HCI) and Computer Supported Cooperative Work (CSCW). I also run the Grouplab research laboratory.

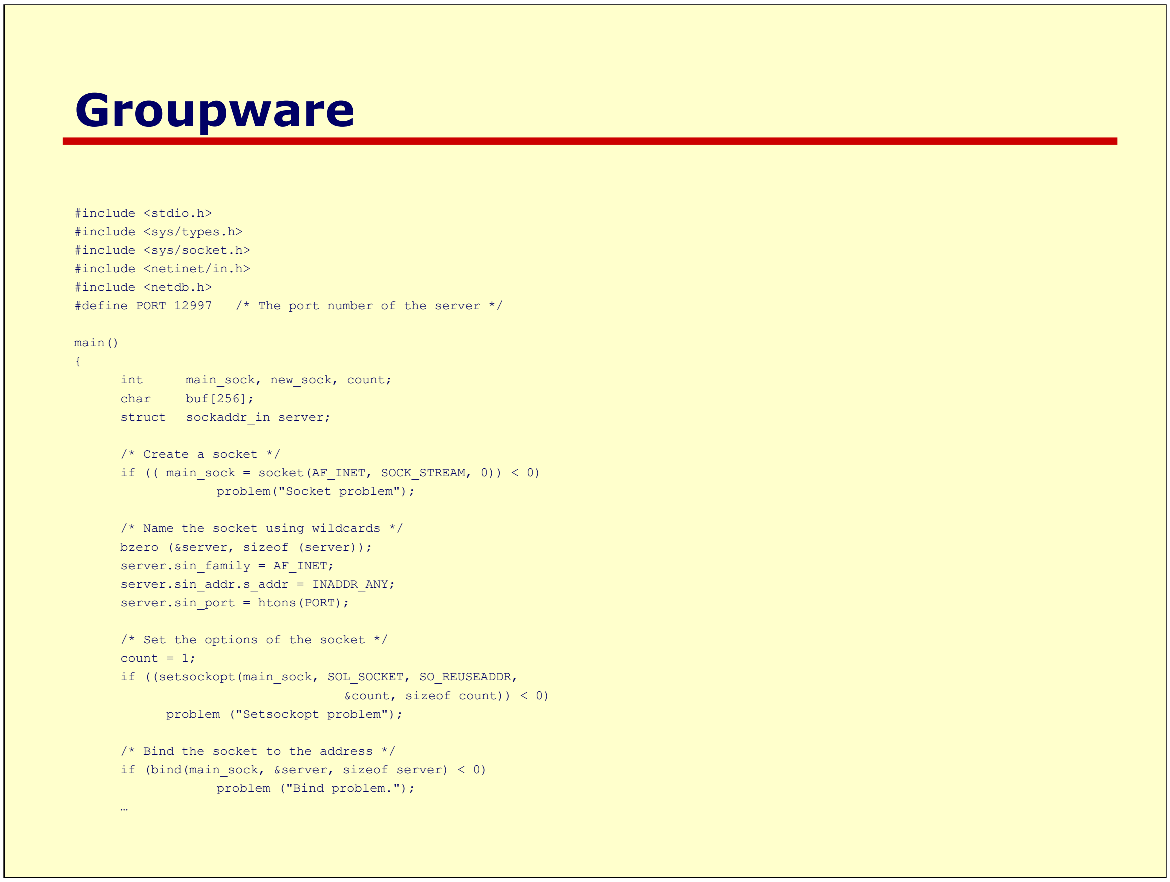
Recent news

- **Interactions laboratory** - the Grouplab and Innovis research groups now coexist in a new laboratory
- **New software** - Phidgets: physical widgets for prototyping user interfaces
- **Faculty opening** for an HCI person

Similarly, in the 1980s Apple’s Hypercard let amateur programmers rapidly create interesting hypertext documents.

<ANIMATE>

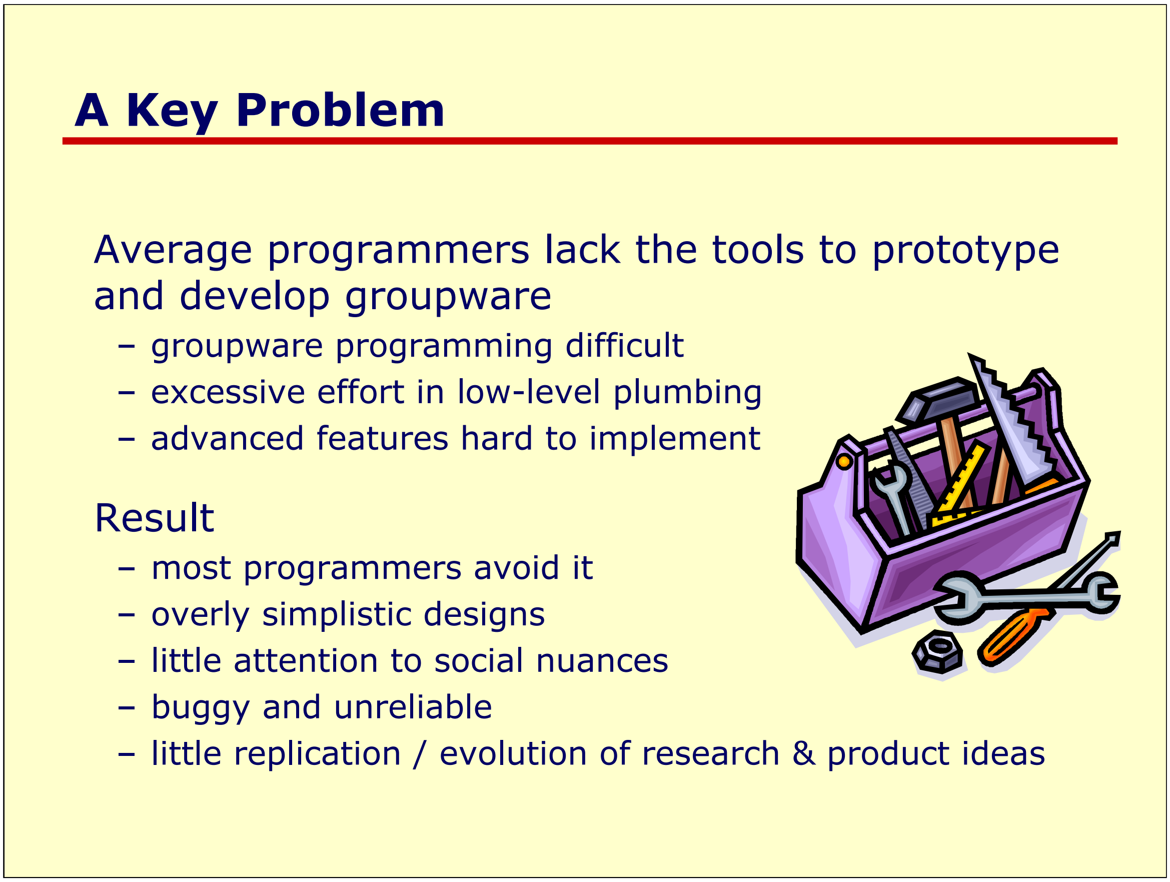
With the introduction of HTML and the many web page editors, everyday people with minimal computer experience now rapidly create and modify hypertext web documents to the point that this is now being taught to elementary school students as part of basic computer literacy.



Groupware

```
#include <stdio.h>
#include <sys/types.h>
#include <sys/socket.h>
#include <netinet/in.h>
#include <netdb.h>
#define PORT 12997   /* The port number of the server */

main()
{
      int      main_sock, new_sock, count;
      char     buf[256];
      struct   sockaddr_in server;

      /* Create a socket */
      if (( main_sock = socket(AF_INET, SOCK_STREAM, 0)) < 0)
                  problem("Socket problem");

      /* Name the socket using wildcards */
      bzero (&server, sizeof (server));
      server.sin_family = AF_INET;
      server.sin_addr.s_addr = INADDR_ANY;
      server.sin_port = htons(PORT);

      /* Set the options of the socket */
      count = 1;
      if ((setsockopt(main_sock, SOL_SOCKET, SO_REUSEADDR,
                                   &count, sizeof count)) < 0)
            problem ("Setsockopt problem");

      /* Bind the socket to the address */
      if (bind(main_sock, &server, sizeof server) < 0)
                  problem ("Bind problem.");
      …
```

Yet when we look at groupware, tools are back in the dark ages, where groupware development means writing low-level programs.



A Key Problem

Average programmers lack the tools to prototype and develop groupware

- groupware programming difficult
- excessive effort in low-level plumbing
- advanced features hard to implement

Result

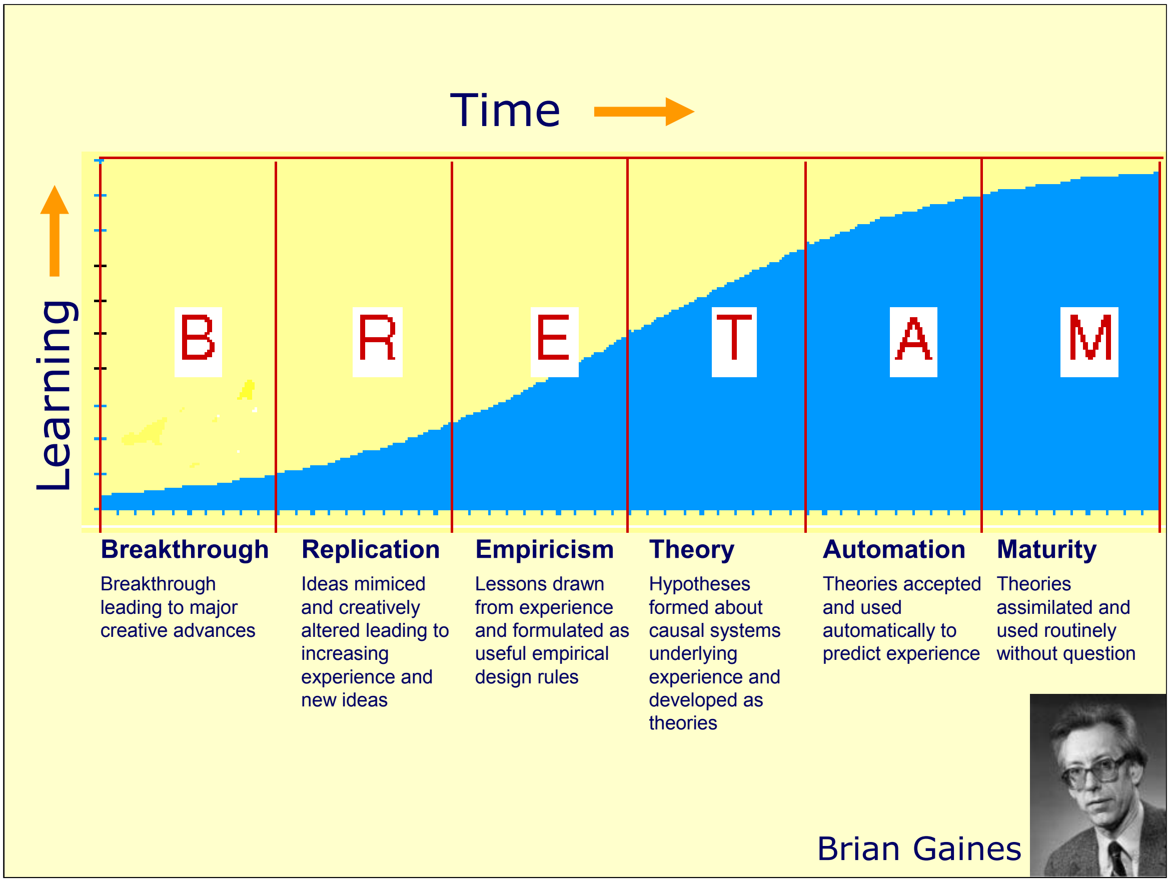
- most programmers avoid it
- overly simplistic designs
- little attention to social nuances
- buggy and unreliable
- little replication / evolution of research & product ideas

This introduces what I believe to be the key technical problem behind groupware failure.

<READ>

<ANIMATE>

Most programmers avoid groupware -> advanced topic in graduate school



Breakthrough

Breakthrough leading to major creative advances

Replication

Ideas mimiced and creatively altered leading to increasing experience and new ideas

Empiricism

Lessons drawn from experience and formulated as useful empirical design rules

Theory

Hypotheses formed about causal systems underlying experience and developed as theories

Automation

Theories accepted and used automatically to predict experience

Maturity

Theories assimilated and used routinely without question

Brian Gaines

But is this lack of tools really a bottleneck?

Well, in the 1980s, Computer Scientist Brian Gaines introduced a model of how science technology develops over time, and how people learn from this development

It unfolds from <READ>

Breakthrough -> Invention

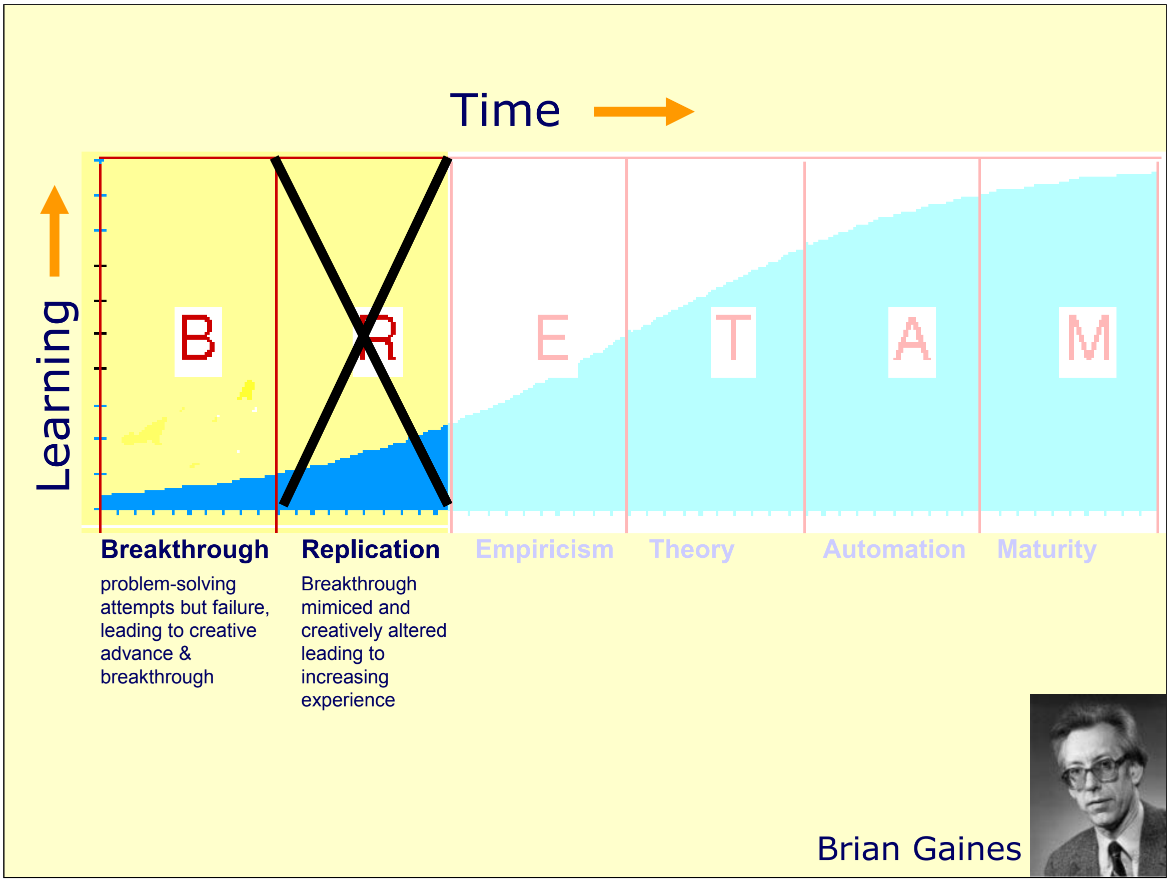
Replication -> Research

Empiricism -> Product Innovation

Theory -> Product Lines

Automation -> Low cost products

Maturity -> Throw away products



Breakthrough

problem-solving attempts but failure, leading to creative advance & breakthrough

Replication

Breakthrough mimiced and creatively altered leading to increasing experience

Empiricism

Theory

Automation

Maturity

Brian Gaines

But because groupware is hard to build, we have essentially throttled its replication.

This has minimized product invention and innovation as well as hands-on experience to all but the CSCW research community and a few well-resourced developers.

Thus the necessary creativity leading to product evolution was stifled.

Tools as media and language

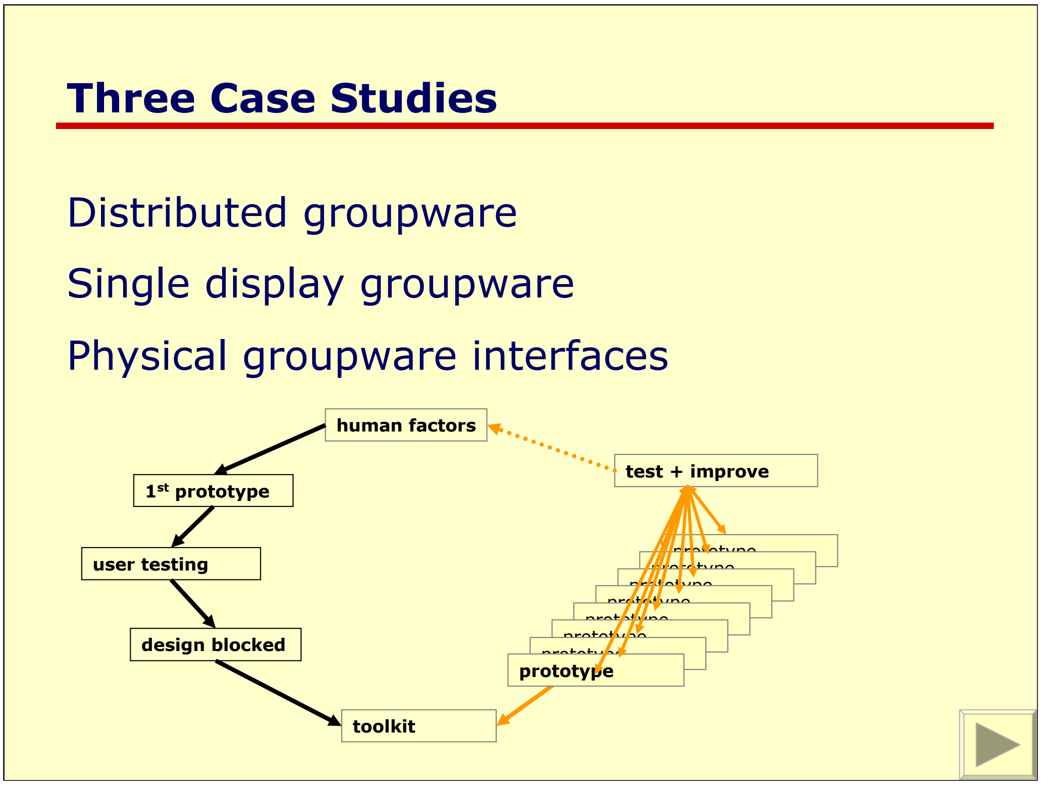
Language influences our thoughts and behaviors

-Sapir-Whorf Hypothesis (~1920s)

Another way of saying this is that all design disciplines recognize the importance of creative media and media tools in how the ‘average’ designer thinks.

Echoing the Sapir-Whorf Hypothesis in linguistics, which states that language influences how we think and behave, the media becomes a language that influences creative thoughts, that indicates design directions, and that lets them concentrate on their design.

Groupware has, in general, failed to give people this creative media.



Three Case Studies

Distributed groupware

Single display groupware

Physical groupware interfaces

human factors

1st prototype

user testing

design blocked

toolkit

prototype

test + improve

What I would like to do now is show how toolkits promoted creativity within our laboratory.

I will use three case studies reflecting three genres of groupware:

-Real time distributed groupware,

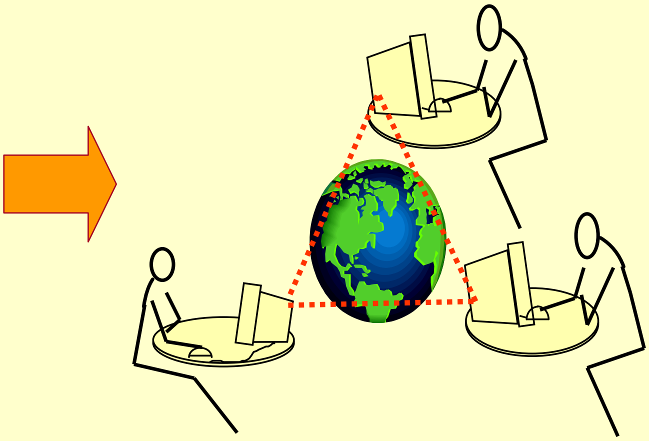
-Single display groupware, and

-Tangible groupware.

Each case study will follow a similar pattern, where we will see how students interested in the human factors of groupware faced serious problems developing and testing early prototypes, and how the introduction of a toolkit profoundly changed the ability of students to rapidly prototype and evolve groupware design after they observed how its used.

Case study

Real Time Distributed Groupware



Distributed shared workspaces

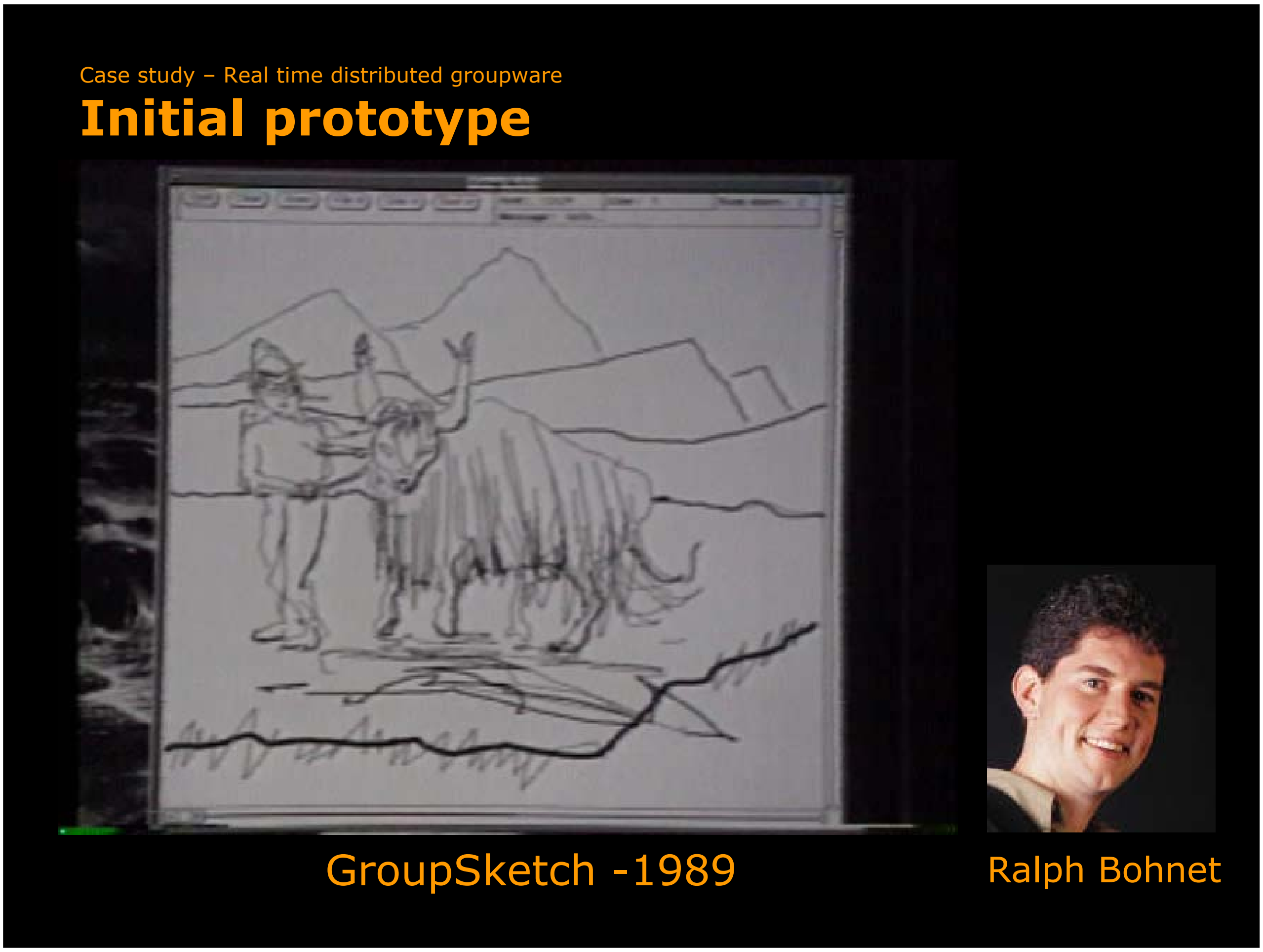
- understand group interaction
- develop effective interaction techniques
- generalize as design requirements

I will begin with a case study of real time distributed groupware, and will continue later on to other groupware topics.

In 1989, when I first became interested in CSCW, I decided to focus on how technology can support the how distributed groups could use a shared visual workspace.

I wanted to

<READ>



Case study – Real time distributed groupware

Initial prototype

GroupSketch -1989

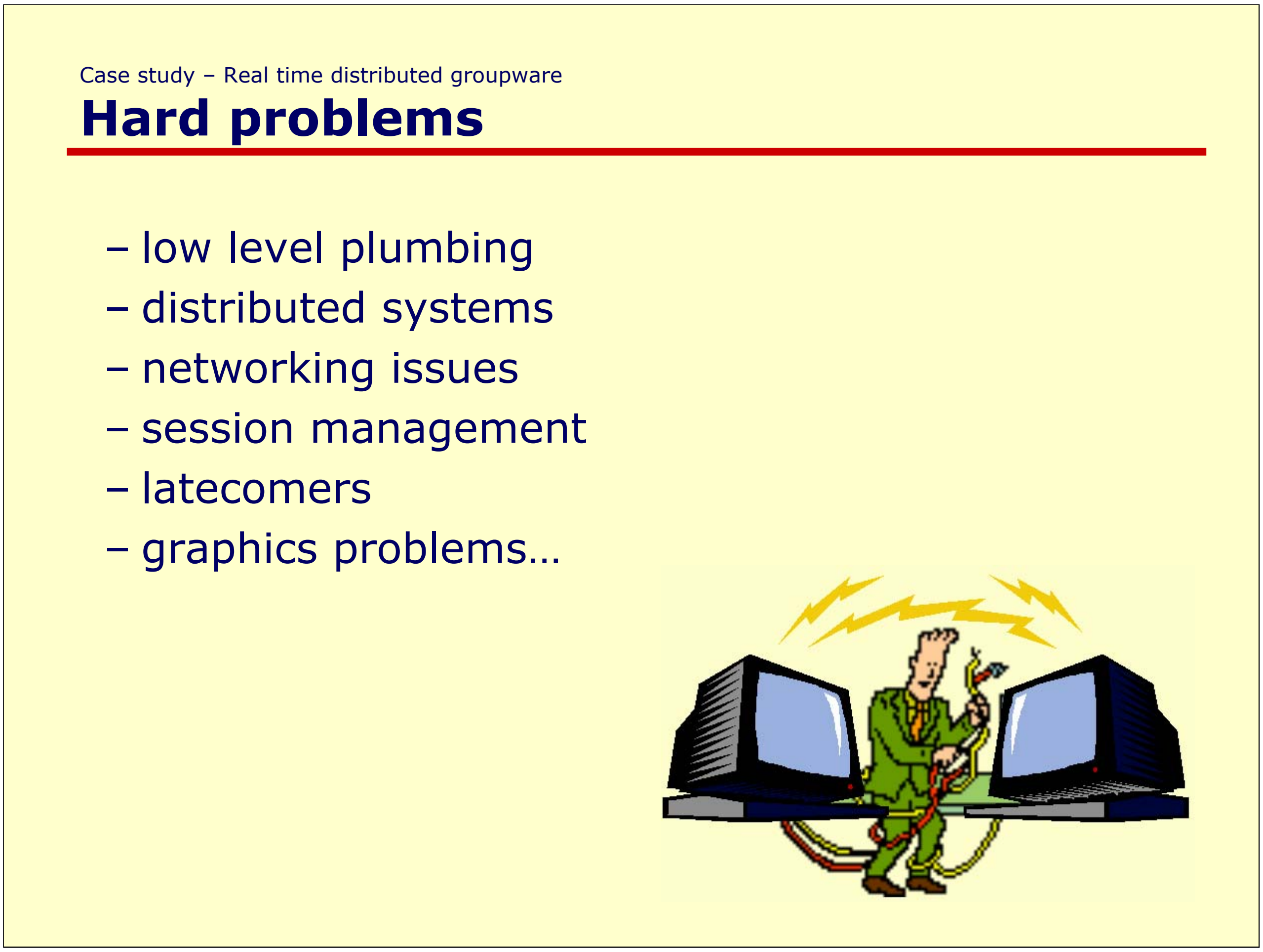
Ralph Bohnet

To help us understand these distributed workspaces, we decided to develop a groupware sketchpad.

Student Ralph Bohnet took on this task, and 4 months plus many thousands of lines of code later, he produced GroupSketch.

This video will give you a flavor of its use by a distributed group, although I am sure many of you have seen similar systems today.

<SHOW VIDEO>



Case study – Real time distributed groupware

Hard problems

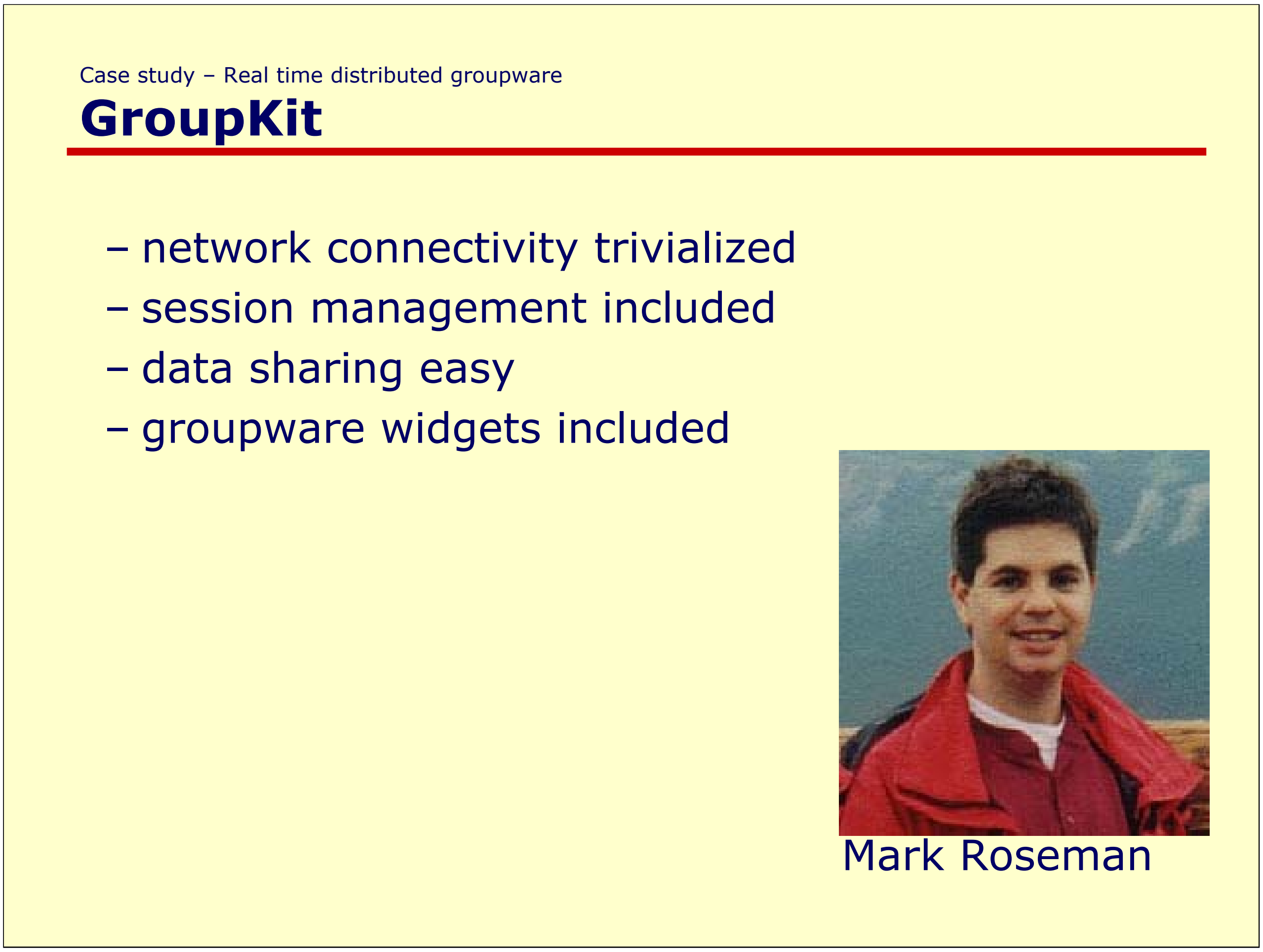
– low level plumbing
– distributed systems
– networking issues
– session management
– latecomers
– graphics problems…

From our evaluations of these systems, we had ideas for new design directions.

But we were discouraged from doing so.

This is because we ran into hard technical problems, which severely limited our ability to efficiently replicate and evolve the system designs.

<READ>



Case study – Real time distributed groupware

GroupKit

- network connectivity trivialized
- session management included
- data sharing easy
- groupware widgets included

Mark Roseman

Instead, my student Mark Roseman decided to build a toolkit that would ease the job of creating real time distributed groupware applications.

The result was GroupKit, built around 1992.

Groupkit ‘s design tried to make simple groupware designs very simple to do in practice.

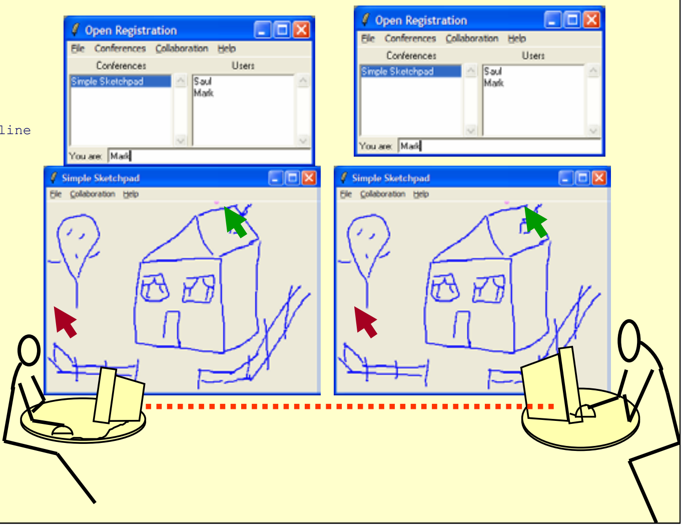
For example,

<Read>

Case study – Real time distributed groupware

GroupSketch in GroupKit

```
gk::menus addstandard
.menubar.file insert 0 command -label "Clear Canvas" -command "gk::to all clearCanvas"
.menubar.file insert 1 separator
grid [canvas .c] -column 0 -row 0 -sticky nwes
grid columnconfigure . 0 -weight 1
grid rowconfigure . 0 -weight 1
bind .c <1> "global lastX lastY; set lastX %x; set lastY %y"
bind .c <B1-Motion> {draw %x %y}
proc draw {thisX thisY} {
    global lastX lastY
    set x1 $lastX
    set y1 $lastY
    set lastX $thisX
    set lastY $thisY
    gk::to all doDraw $x1 $y1 $thisX $thisY
}
proc doDraw {lastX lastY thisX thisY} {
    .c create line $lastX $lastY $thisX $thisY -width 2 -tag line
    gk::schedule update "update idletasks"
}
proc clearCanvas {} {
   .c delete line
}
gk::event bind updateEntrant "sendDrawing .c %U"
proc sendDrawing {canvas usernum} {
     foreach line [$canvas find all] {
          set x1 [lindex [$canvas coords $line] 0]
          set y1 [lindex [$canvas coords $line] 1]
          set x2 [lindex [$canvas coords $line] 2]
          set y2 [lindex [$canvas coords $line] 3]
          gk::to $usernum doDraw $x1 $y1 $x2 $y2
}}
gk::telepointers attach .c
```

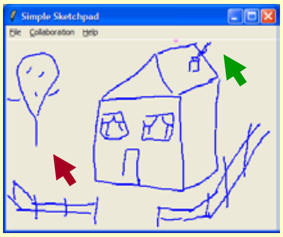


To illustrate, here is the complete Groupkit code that replicates GroupSketch.

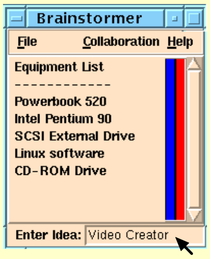
It is 33 lines long, and took about 20 minutes to write. This contrasts to the 4 months and > 5000 lines of the original Groupsketch system!

Case study – Real time distributed groupware

Rapid prototyping – Simple groupware



GroupSketch-33

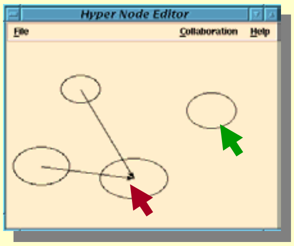


Equipment List

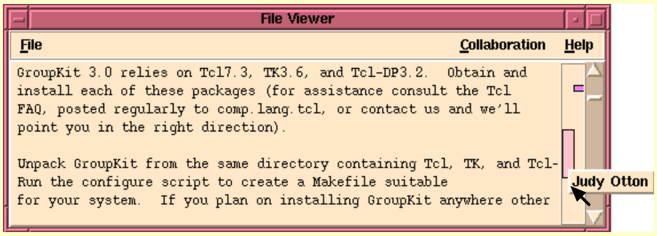
Powerbook 520
Intel Pentium 90
SCSI External Drive
Linux software
CD-ROM Drive

Enter Idea: Video Creator

Brainstormer-74

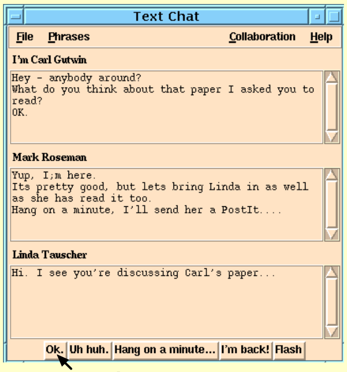


Concept map-213



```
GroupKit 3.0 relies on Tcl7.3, TK3.6, and Tcl-DP3.2.  Obtain and
install each of these packages (for assistance consult the Tcl
FAQ, posted regularly to comp.lang.tcl, or contact us and we'll
point you in the right direction).

Unpack GroupKit from the same directory containing Tcl, TK, and Tcl-
Run the configure script to create a Makefile suitable
for your system.  If you plan on installing GroupKit anywhere other
```

File sharing-51


Text Chat
File Phrases Collaboration Help
Ok. Uh huh. Hang on a minute... I'm back! Flash

I'm Carl Gutwin

Hey - anybody around?
What do you think about that paper I asked you to read?
OK.

Mark Roseman

Yup, I;m here.
Its pretty good, but lets bring Linda in as well as she has read it too.
Hang on a minute, I'll send her a PostIt....

Linda Tauscher

Hi. I see you're discussing Carl's paper...

Text chat-80

Our lab members then replicated many simple systems, where they found it only slightly harder and longer to build groupware than single-user counterparts.

For example,….

Case study – Real time distributed groupware

Rapid prototyping – Workspace overviews

-basic overview
-telepointers
-radar view
-portrait radar
-active radar

Carl Gutwin

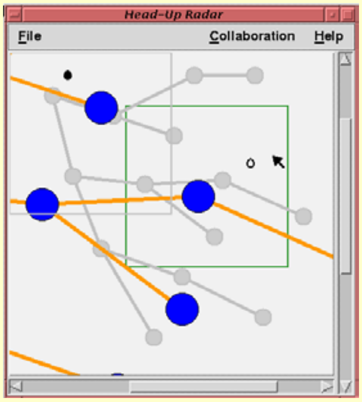
More importantly, we now how the means to create and evolve new ideas.

As one example, student Carl Gutwin wanted to create and refine the notion of workspace overviews, a (now well accepted) mechanism that lets people stay aware of other’s activities as they work in different parts of a shared visual space.

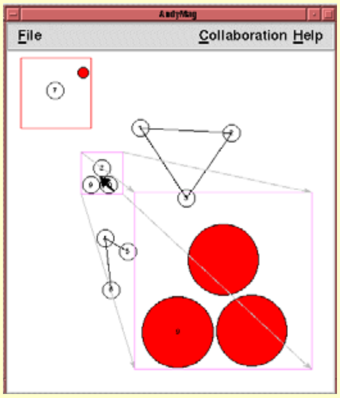
What I would like to do now is show you an early videos (mid-90s) that shows various evolutions of this concept, beginning with a very basic visual overview and ending with a fully featured radar view that lets people not only view the entire workspace, but interact through it.

Case study – Real time distributed groupware

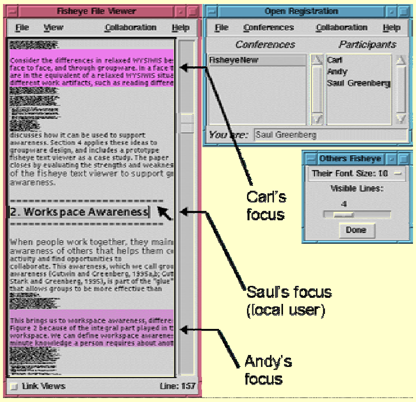
Rapid prototyping – other creative ideas



Transparent multi-views



Magnification



Consider the differences in relaxed WYSIWIS be
face to face, and through groupware. In a face t
are in the equivalent of a relaxed WYSIWIS situa
different work artifacts, such as reading differe

discusses how it can be used to support
awareness. Section 4 applies these ideas to
groupware design, and includes a prototype
fisheye text viewer as a case study. The paper
closes by evaluating the strengths and weaknes
of the fisheye text viewer to support gr
awareness.

2. Workspace Awareness

When people work together, they main
awareness of others that helps them c
activity and find opportunities to
collaborate. This awareness, which we call grou
awareness (Gutwin and Greenberg, 1995a,b; Gut
Stark and Greenberg, 1995), is part of the "glue"
that allows groups to be more effective than

This brings us to workspace awareness, differe
Figure 2 because of the integral part played in t
workspace. We can define workspace awarene
minute knowledge a person requires about anot

Conferences: FisheyeNew

Participants: Carl, Andy, Saul Greenberg

You are: Saul Greenberg

Their Font Size: 10

Visible Lines: 4

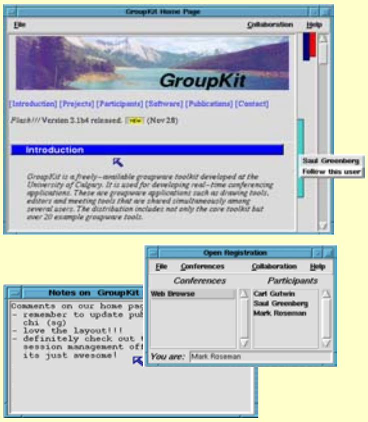
Fisheye lens

What the video did not show was the explosion of other ideas that Carl and other lab members had of how to support workspace awareness.

I don't have the time to go through these, but one set of projects experimented with a variety of information visualization techniques as a possible solution to the problem. Left to right, these ranged from transparent multi-level views of the workspace, to magnification lenses, to fisheye lenses, one for each participant.

Case study – Real time distributed groupware

Rapid prototyping – major applications



GroupKit Home Page

File Collaboration Help

GroupKit
[Introduction] [Projects] [Participants] [Software] [Publications] [Contact]

Flash!!! Version 5.1b4 released. new (Nov 28)

Introduction

GroupKit is a freely-available groupware toolkit developed at the University of Calgary. It is used for developing real-time conferencing applications. These are groupware applications such as drawing tools, editors and meeting tools that are shared simultaneously among several users. The distribution includes not only the core toolkit but over 20 example groupware tools.

Notes on GroupKit

```
Comments on our home pag
- remember to update pub
  chi (sg)
- love the layout!!!
- definitely check out t
  session management off
  its just awesome!
```

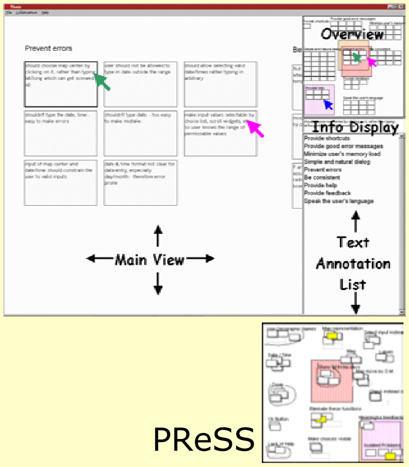
Open Registration

File Conferences Collaboration Help

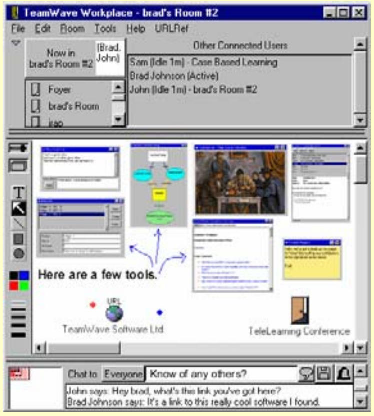
Conferences	Participants
Web Browse	Carl Gutwin Saul Greenberg Mark Roseman

You are: Mark Roseman

Groupweb



PReSS



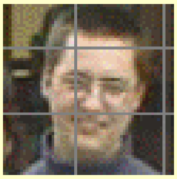
TeamWave Workplace - brad's Room #2

Sam (Idle 1m) - Case Based Learning
Brad Johnson (Active)
John (Idle 1m) - brad's Room #2

Know of any others?

John says: Hey brad, what's this link you've got here?
Brad Johnson says: It's a link to this really cool software I found.

Teamrooms



As well, many major groupware applications were built. These include:

Groupweb, likely the first groupware browser ever made.

The PreSS tool that allows distributed usability engineers to transform heuristic evaluation results into problem reports

and TeamRooms, a very sophisticated system that let people create and enter rooms full of persistent groupware artifacts. As one person entered a room, they were immediately connected to all others in the same room. Teamrooms was eventually commercialized…

Case study – Real time distributed groupware

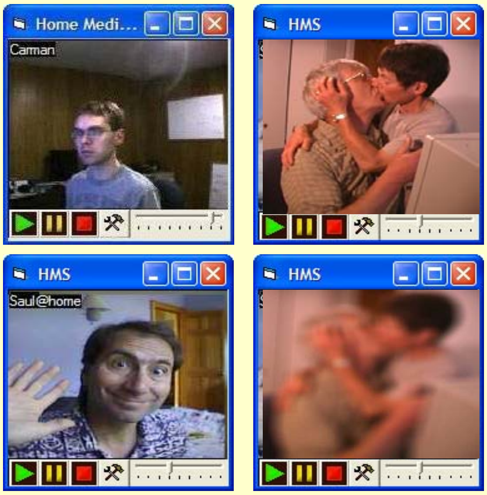
Grouplab Collabrary

Modern collaborations include multimedia

–capture and display

–analysis and manipulation

–multimedia data sharing



Michael Boyle

While all the above helped us understand the human factors of groupware, our toolkits evolved as well.

For example, we recognized that modern groupware collaboration require multimedia such as video, yet Groupkit did not support this.

Consequently, student Michael Boyle implemented a Groupkit replacement called the Collabrary, a toolkit that

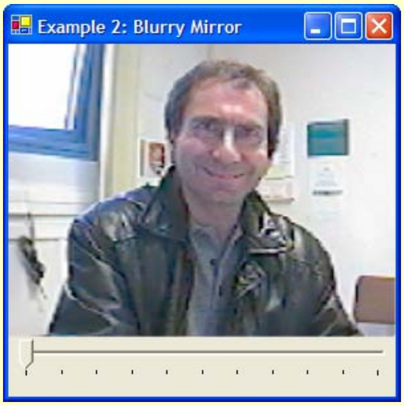
-makes it easy to capture and display multimedia.

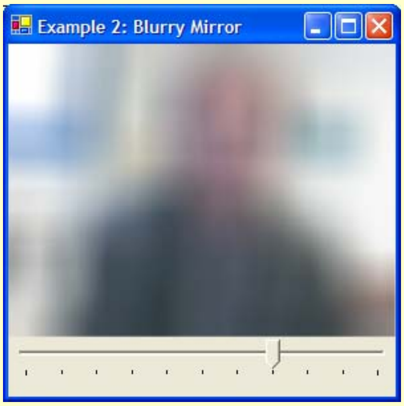
-provides rich functionality for analysing and manipulating video frames,

-and improves on Groupkit's ability to share data by including a means to marshall and distribute multimedia.

Case study – Real time distributed groupware

Video Mirror





```
…
    this.camera = new CameraClass();
    this.camera.Captured += new CameraEvents CapturedEventHandler(Captured);
    this.camera.FrameRate = 30;

private void Captured(IPhoto Frame) {
    Frame.Distort(1 -  trackBar1.Value, Collabrary.PhotoDistortStyle.Blur);
    this.pictureBox1.Image = Image.FromHbitmap(Frame.Hbitmap);
}
```

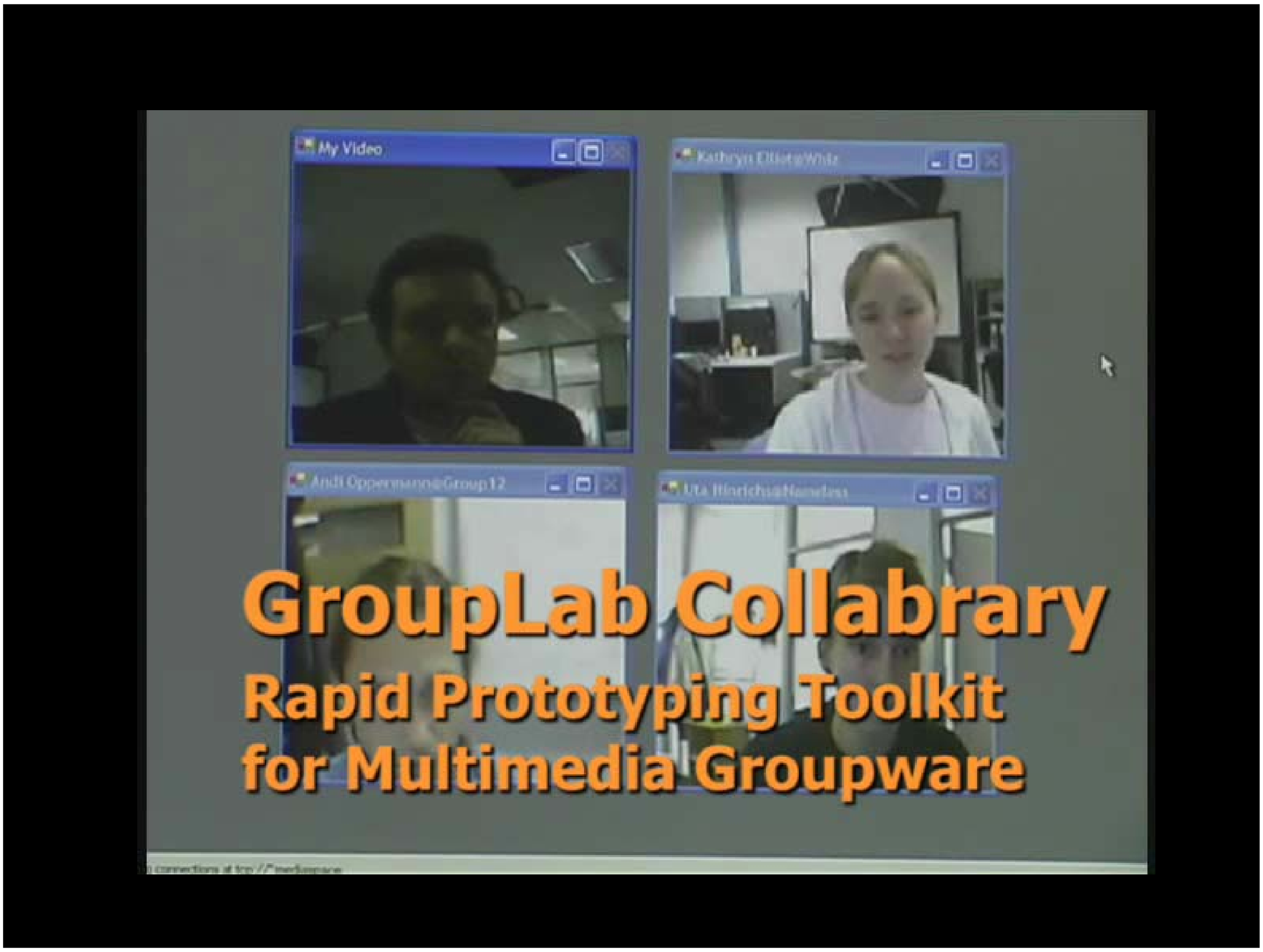
To show the simplicity of capturing and displaying multimedia, here is an example of a video mirror.

<SHOW DEMO>

where a video stream is captured from any camera attached to a PC and displayed in real time.

A programmer only has to write 5 conceptually simple lines of code to completely implement this example.

The top 3 lines creates a camera and sets its capture frame rate, while the bottom 2 lines blurs and displays each frame of the image.



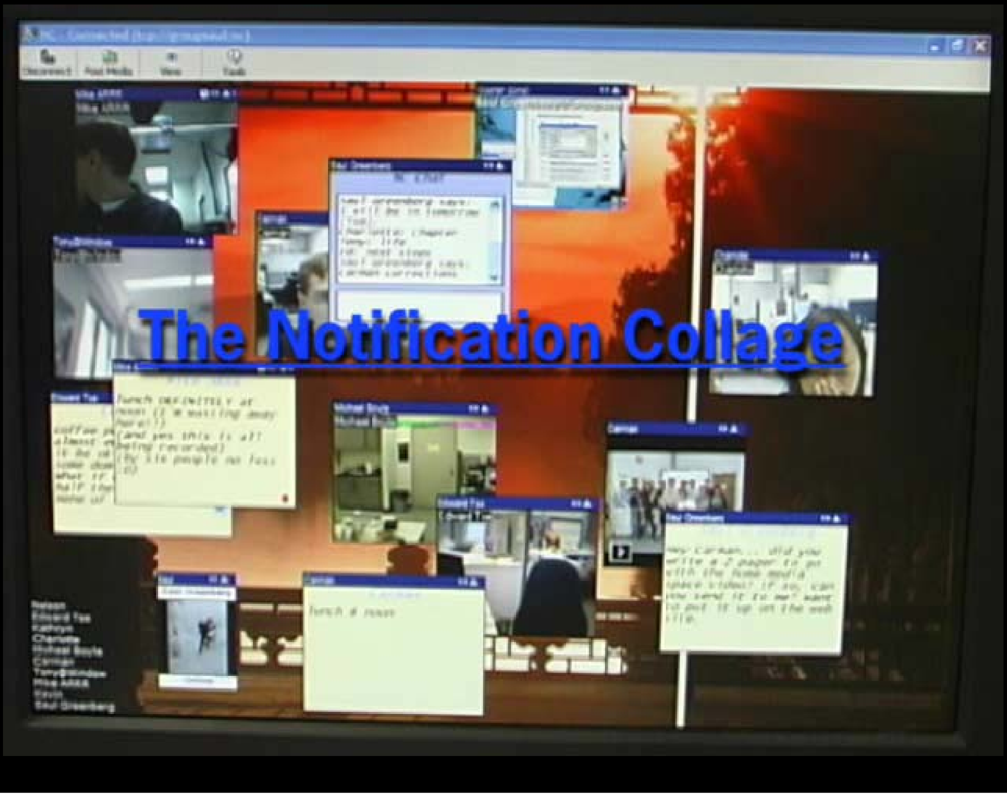
GroupLab Collabrary

Rapid Prototyping Toolkit for Multimedia Groupware

Lets take a look at several simple media space applications, where each was developed in a very short amount of time.

Case study – Real time distributed groupware

Notification Collage



Mike Rounding

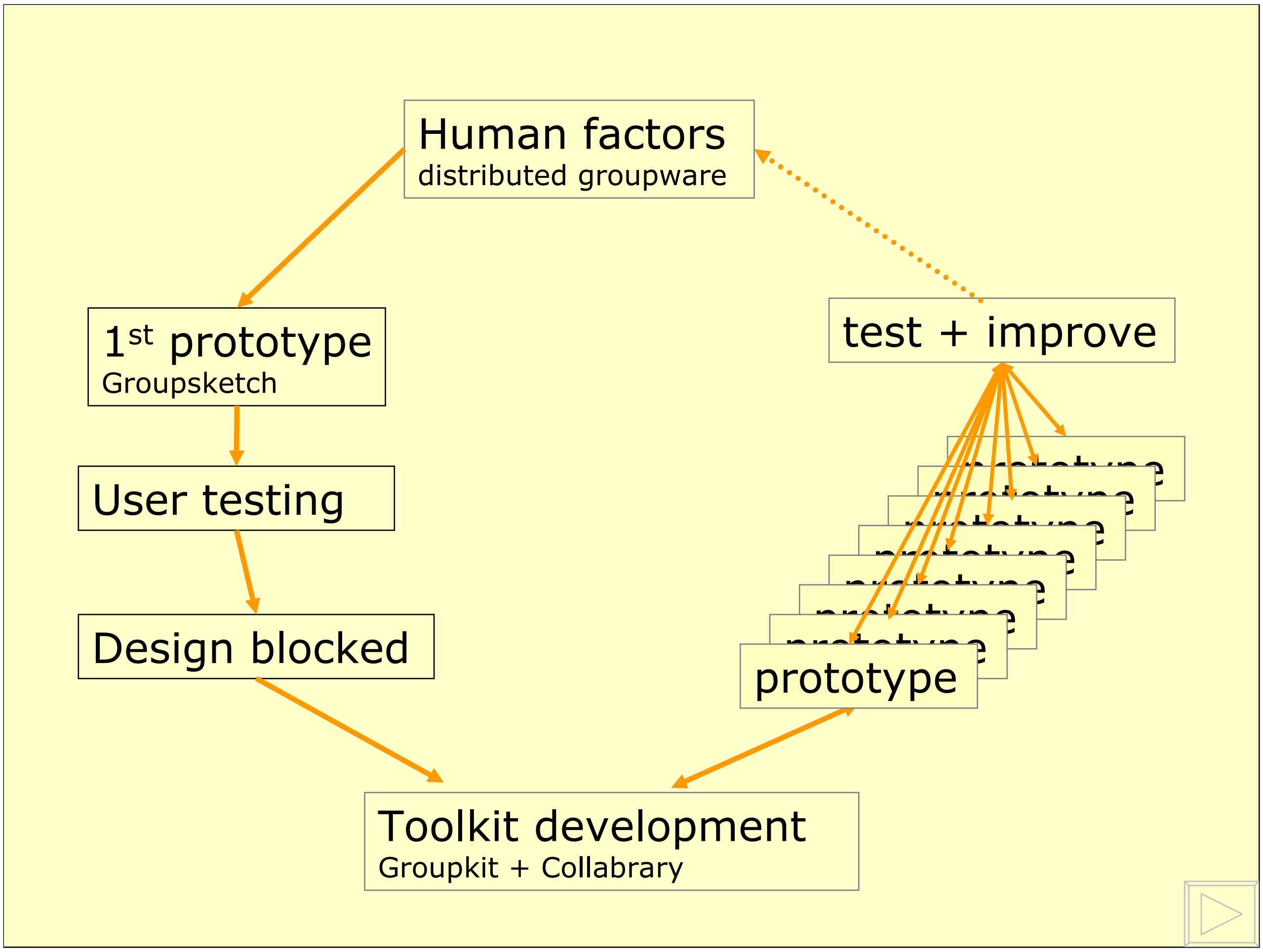
The collabrary became a language for students to think about multimedia within their distributed groupware designs.

One example, built by student Mike Rounding, is the Notification Collage, which used a bulletin board metaphor to support casual interaction between distant collaborators.

Its main idea was to provide a group with awareness of one another through media items displaying a variety of information.

Lets look at a video for an example interaction.

<STOP AFTER VIDEO LINK GOES LIVE>



Human factors
distributed groupware

1st prototype
Groupsketch

User testing

Design blocked

Toolkit development
Groupkit + Collabrary

prototype

test + improve

This case study just illustrated an important development cycle that we will see repeated over and over again.

First, we were motivated by the human factors of design within a groupware genre, in this case distributed groupware.

To help us understand the critical factors, we developed a 1st prototype, in this case GroupSketch, and then observed how people use it through user testing.

While this led to new design ideas, design was blocked because it was just too hard or too time-consuming to modify the prototype.

Instead, we stepped back and developed a toolkit, in this case Groupkit, that would let us rapidly build distributed groupware.

The toolkit then leads to an explosion of prototype development.

As each prototype was tested and improved, it feed both into our understanding of the critical human factors of groupware and to our understanding of what the toolkit should offer.

In the end, these various toolkits proved critical to our group’s creativity, for it let us concentrate on ideas and to rapidly design and test prototypes.

Case study

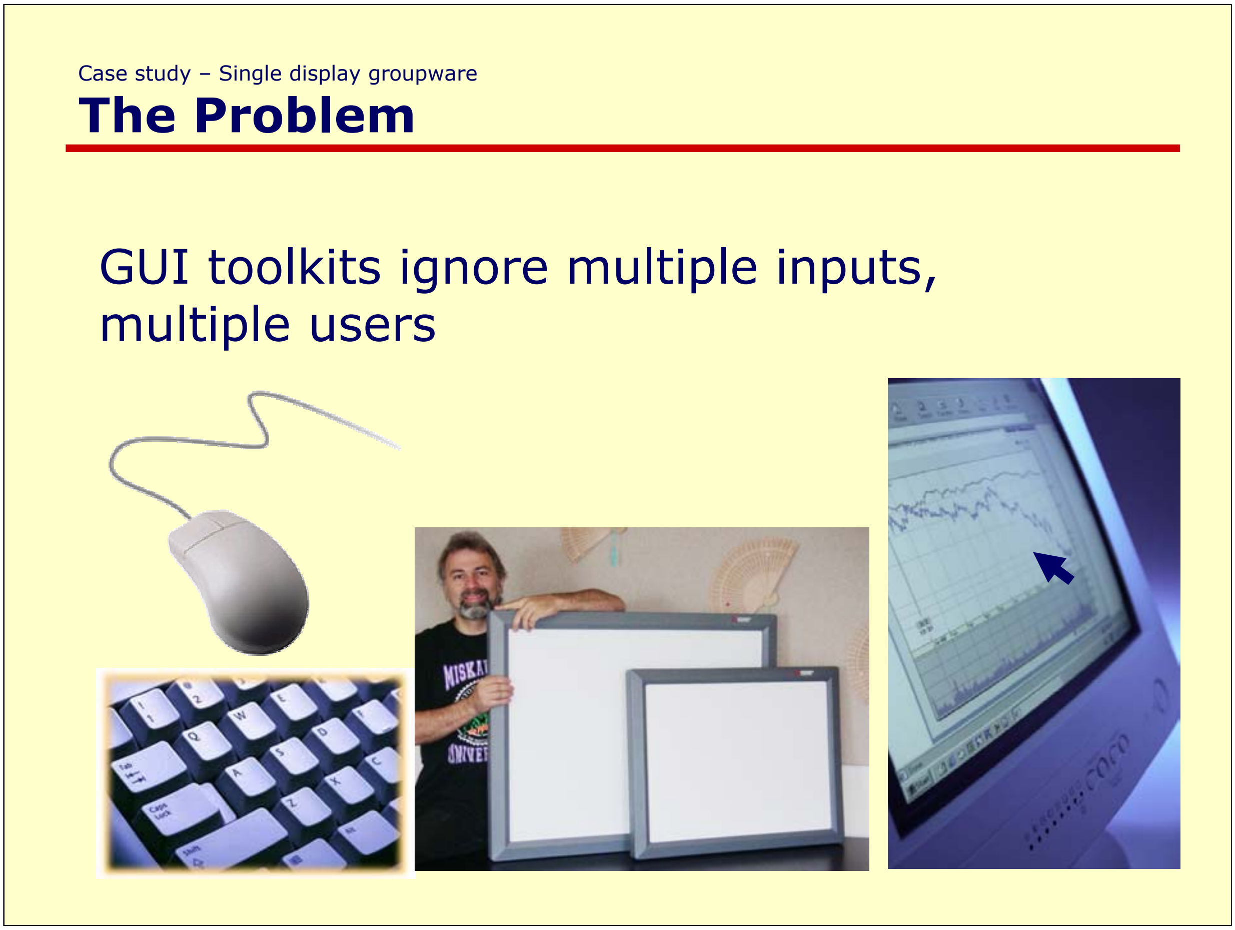
Single Display Groupware

Co-located people work together over

- a table, a wall display, or a monitor through
- multiple mice, keyboards, multi-touch surfaces

I would now like to move to another case study – Single Display Groupware

In this groupware genre, <read>



Case study – Single display groupware

The Problem

GUI toolkits ignore multiple inputs, multiple users

Within the gaming world, SDG is in standard use, albeit in limited ways.

Within productivity applications, the problem is that our

<READ>

In practice, this meant that only a few research laboratories have developed SDG, and that the wheel is reinvented over and over again.

Case study - Single display groupware

SDGToolkit + Grouplab DiamondTouch

multiple mice and keyboards

multitouch surfaces

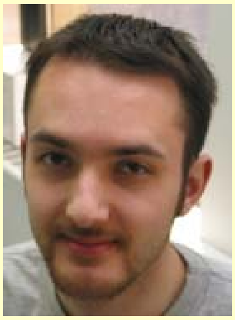
multiple cursors

tabletop orientations

SDG widget layer

SDG widgets

Edward Tse



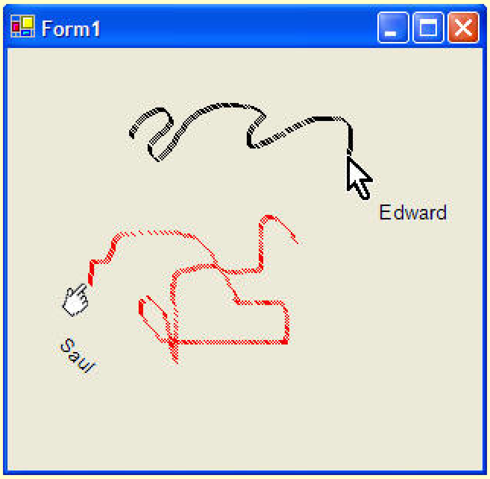
Rob Diaz-Marino

To make SDG development easier, students Edward Tse and Rob Diaz developed the SDG Toolkit, and the Grouplab DiamondTouch Toolkit. Both are serious efforts at providing programmers with basic and advanced SDG building blocks, including

<READ>

Case study - Single display groupware

SimpleSketch for a tabletop



```
//1. Add SdgManager

//2. Set relativeTo property to Form 1

//3. Add sdgMouseMove event

//4. Add after InitializeComponent - Cursors and Orientation
Cursor[] sdgCursors = {Cursors.Arrow, Cursors.Hand, Cursors.Cross};
String[] sdgText = {"Edward", "Saul", "Mike"};
int[]    sdgDegreeRotations = {0, 180, 145};
for (int i=0; i < sdgManager1.Mice.Count && i < 3; ++i) {
  sdgManager1.Mice[i].Cursor = sdgCursors[i];
  sdgManager1.Mice[i].Text = sdgText[i];
  sdgManager1.Mice[i].DegreeRotation = sdgDegreeRotations[i];
}

//5. Add to SDGMouseMove - Drawing lines
if (e.ID > 2 ) return;
Graphics g = this.CreateGraphics();
Brush[] colour = {Brushes.Black, Brushes.Red, Brushes.Gold};
if ((e.Button & MouseButtons.Left) > 0)
    g.FillEllipse(colour[e.ID], e.X, e.Y, 5, 5);
```

Lets take a look at a table-aware program equivalent to GroupSketch.

<RUN PROGRAM>

What you see here on the right is a complete tutorial that illustrates how one can completely implement this program using SDGToolkit.

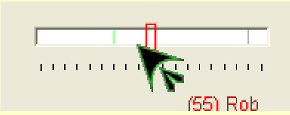
What makes this unusual from a normal program is that multiple input streams and cursors are recognized as first class programming concepts.

In fact, the top bit of code specifies each cursors' appearance and tabletop orientation, while the bottom reacts to each person's mouse actions by drawing appropriately.

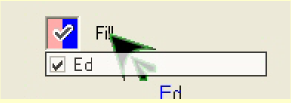
Case study - Single display groupware

Multi-user widgets

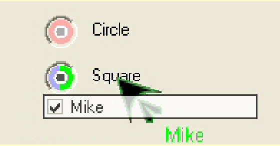
slider

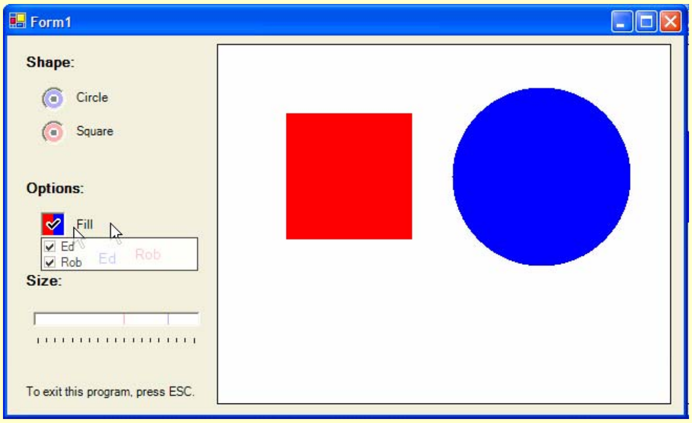


checkbox



radio buttons





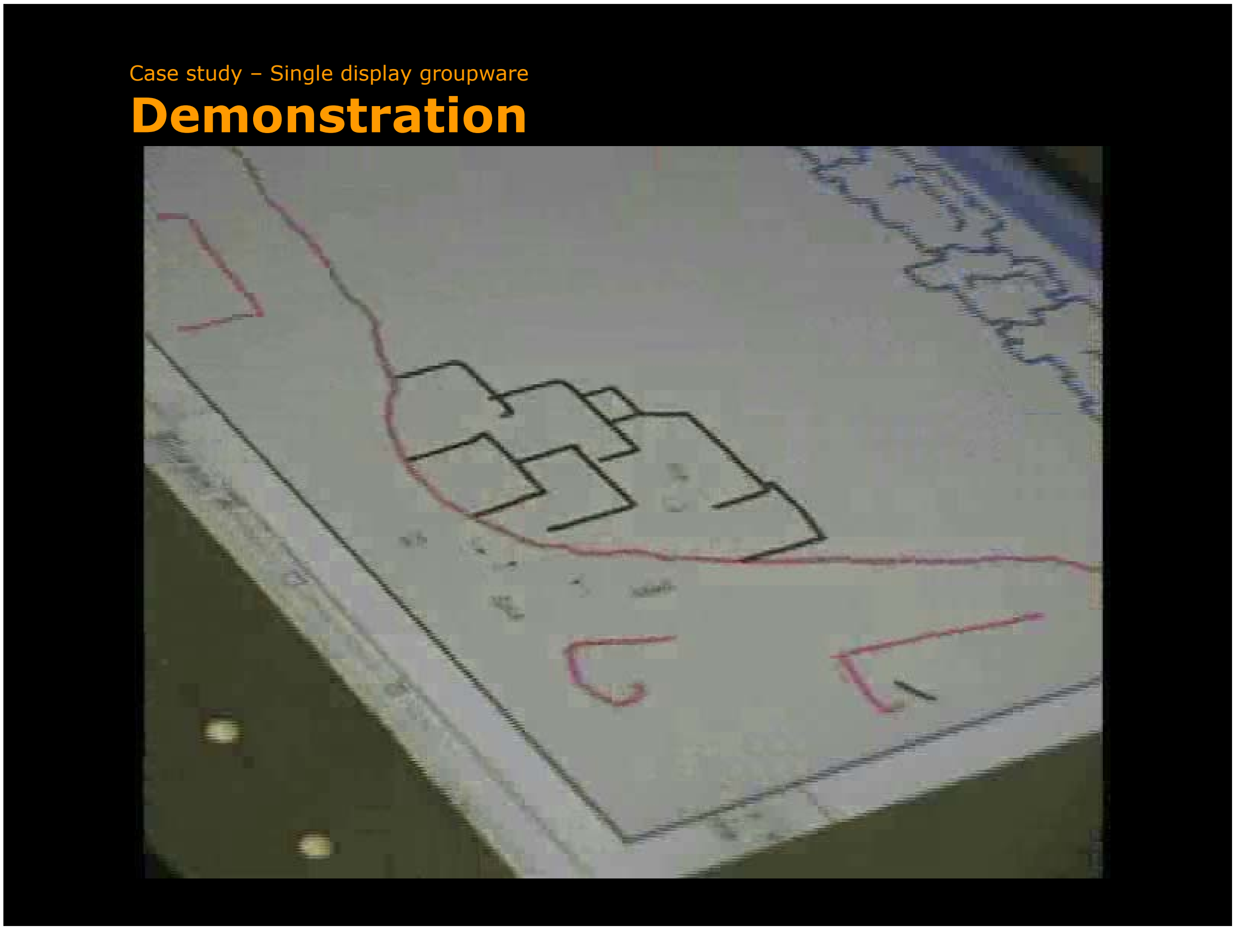
To exit this program, press ESC.

Rob Diaz

This summer, undergraduate student Rob Diaz came on board to develop some sample SDG widgets atop Edward’s widget layer.

After a few weeks, he designed and experimented with several ideas, and I’ll show you just a few of them here.

<Run DEMO>



Case study – Single display groupware

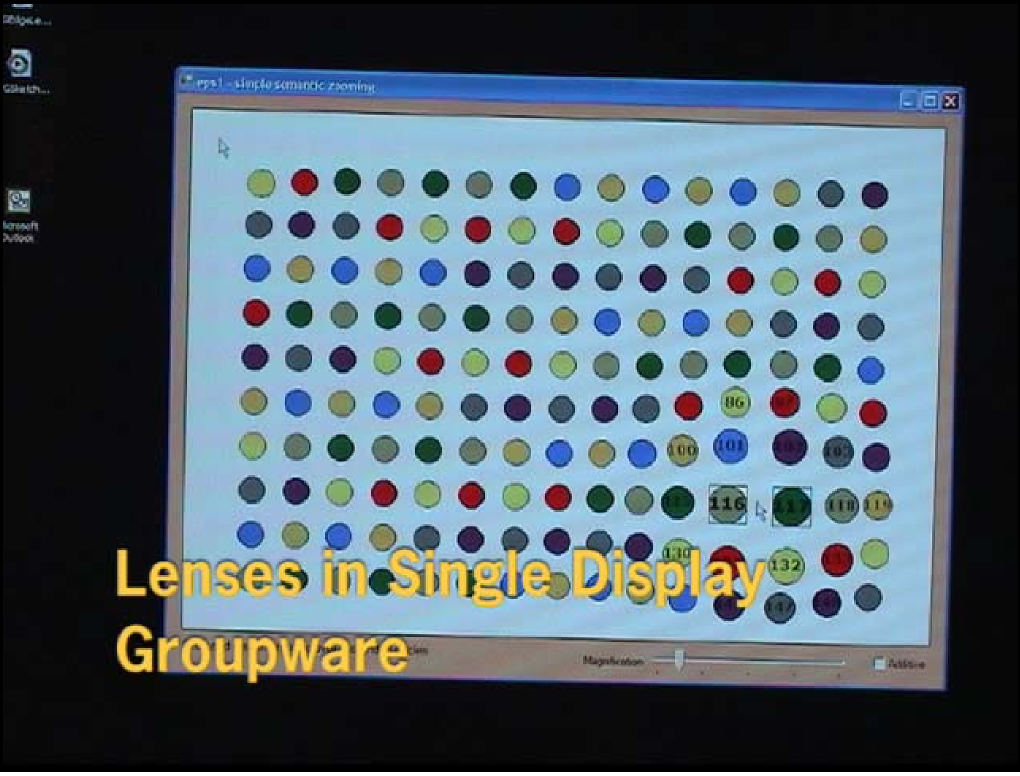
Demonstration

Lets watch a video of how the toolkit works, several of these widgets, and several other examples of SDG.

As before, what I want to stress is that there was an explosion of creativity by many students.

Case study – Single display groupware

Demonstration


eps1 - single semantic zooming
86
100
101
116
132
Magnification
Additive
Lenses in Single Display Groupware

In this next example, built by myself, we see how the power of SDG toolkit is leveraged by combining it with other toolkits.

I was teaching myself Sheelagh Carpendale’s EPS toolkit that makes it easy to add fisheye and other lenses into an information space.

I finished a toy application, and then went to bed. …

<TELL STORY>

Case study – Single display groupware

MultiTouch Surfaces

The next video shows examples of how both simple and complex multitouch SDG applications can be created with the Grouplab DiamondTouch Toolkit.

Case study – Single display groupware

MultiTouch Surfaces

SoundBoard

Using the GroupLab DiamondTouch™ Toolkit

http://grouplab.ucalgary.ca/diamondtouch

This next example, also by Rob Diaz, is hot off the press. Because I knew he was intererested in music compostition, my only instructions to him was “Create an SDG system that exploits sound”. This is what he produced after a few weeks, where most of the programming complexity was in the sound rather than the multitouch SDG capabilities..

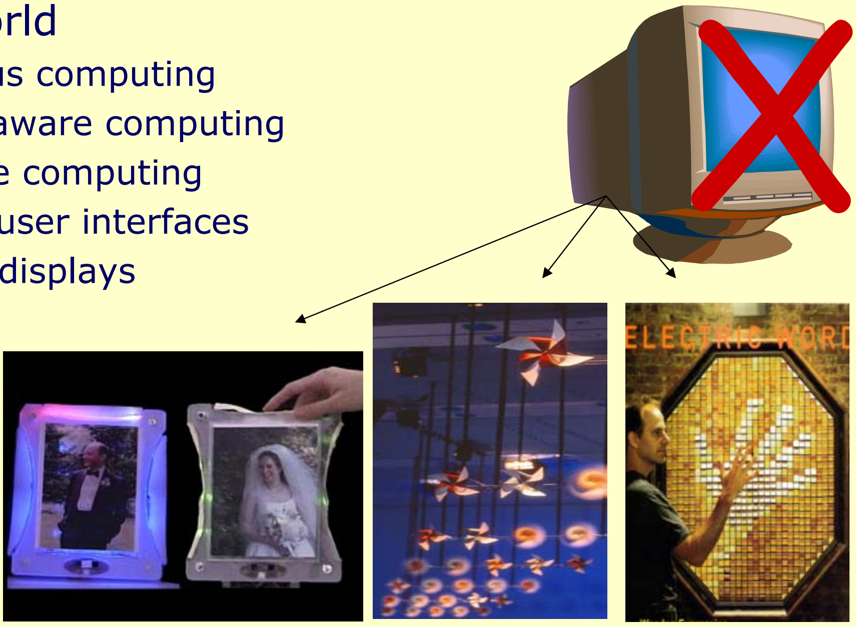
Listen carefully to the sound, how the volume changes, and how it pans between speakers.

Case study

Physical User Interfaces

People interact with physical devices located in their real world

- – ubiquitous computing
- – context-aware computing
- – pervasive computing
- – tangible user interfaces
- – ambient displays
- – …



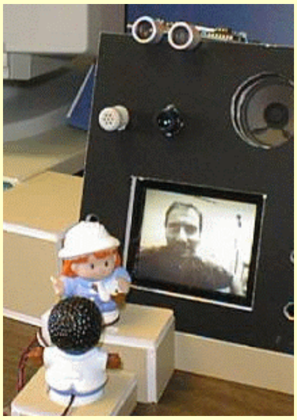
Our third and final case study moves to quite a different area, that of physical user interfaces.

In these systems,

<READ>

Case study – Physical user interfaces

Physical but Digital Surrogates



Greenberg & Kuzuoka

While one would think that this has little to do with CSCW, in actual fact we can create many devices that enhance presence and communication between people.

For example, Our own first efforts in Physical interfaces began in 1999, when Japanese visitor Hideaki Kuzuoka and I built an always-on video-based media space augmented by physical devices.

The system was built from scratch, and took several months to do.

I'll show you an excerpt.

The first part shows how physical devices can provide awareness of others,

The second part shows how physical sensors can mediate privacy concerns.

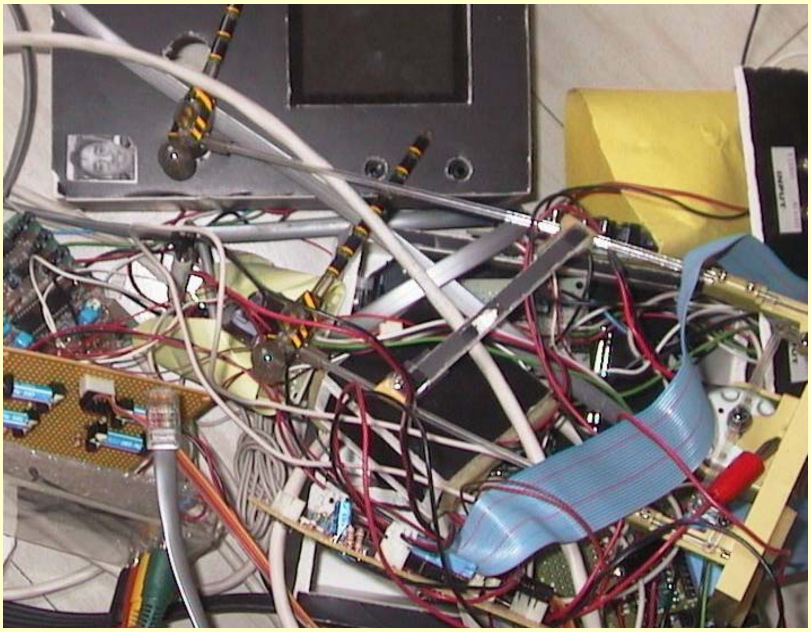
<Show Video>

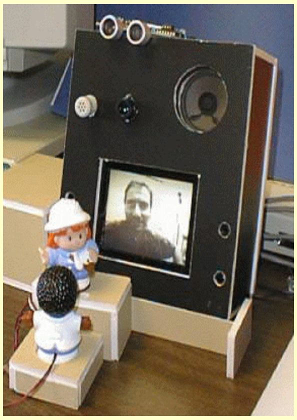
The problem is that when Hideaki left, we lost his engineering expertise and the system became impossible to maintain, let alone extend.

The result was…

Case study – Physical user interfaces

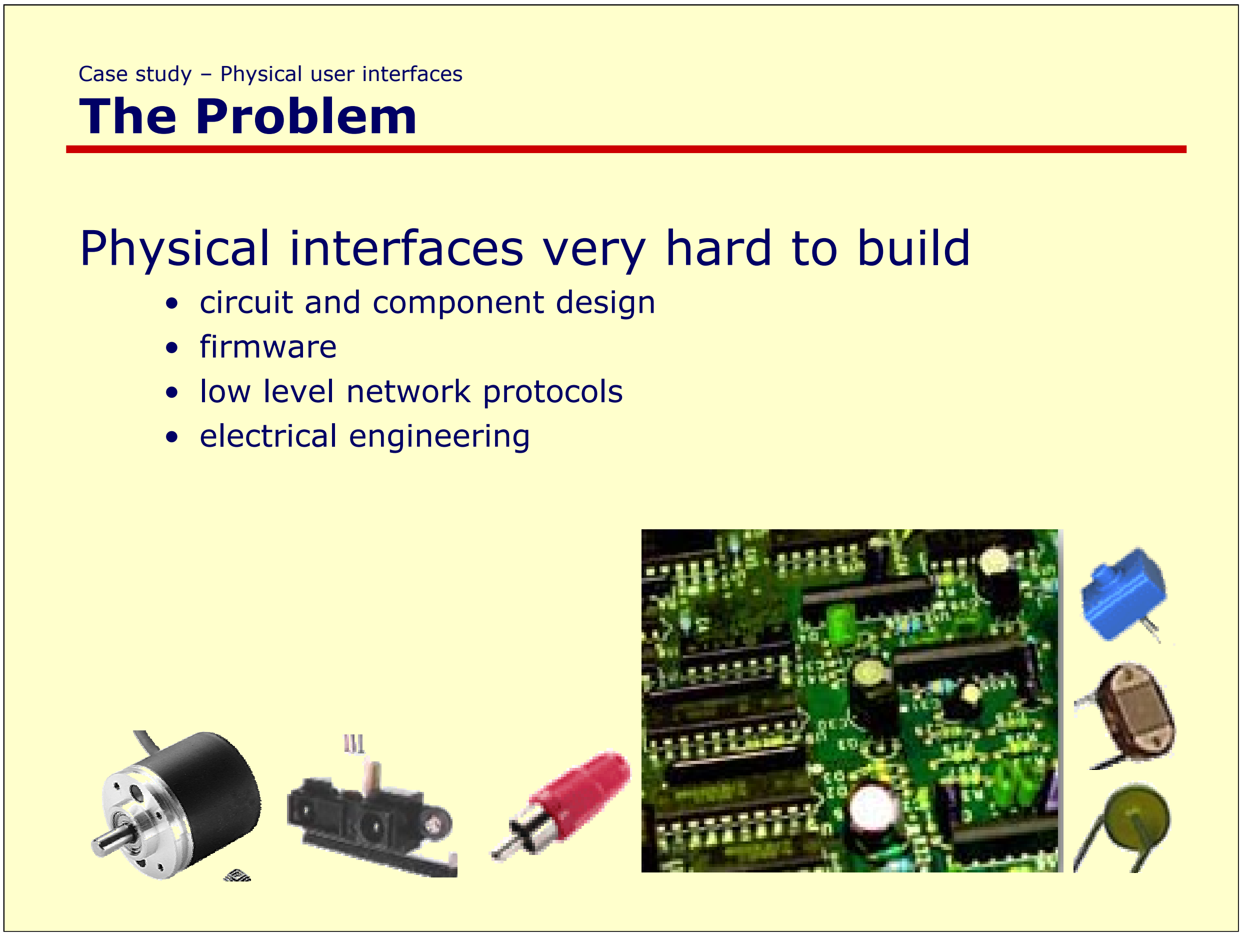
Physical but Digital Surrogates





Greenberg & Kuzuoka

… a big mess.



Case study – Physical user interfaces

The Problem

Physical interfaces very hard to build

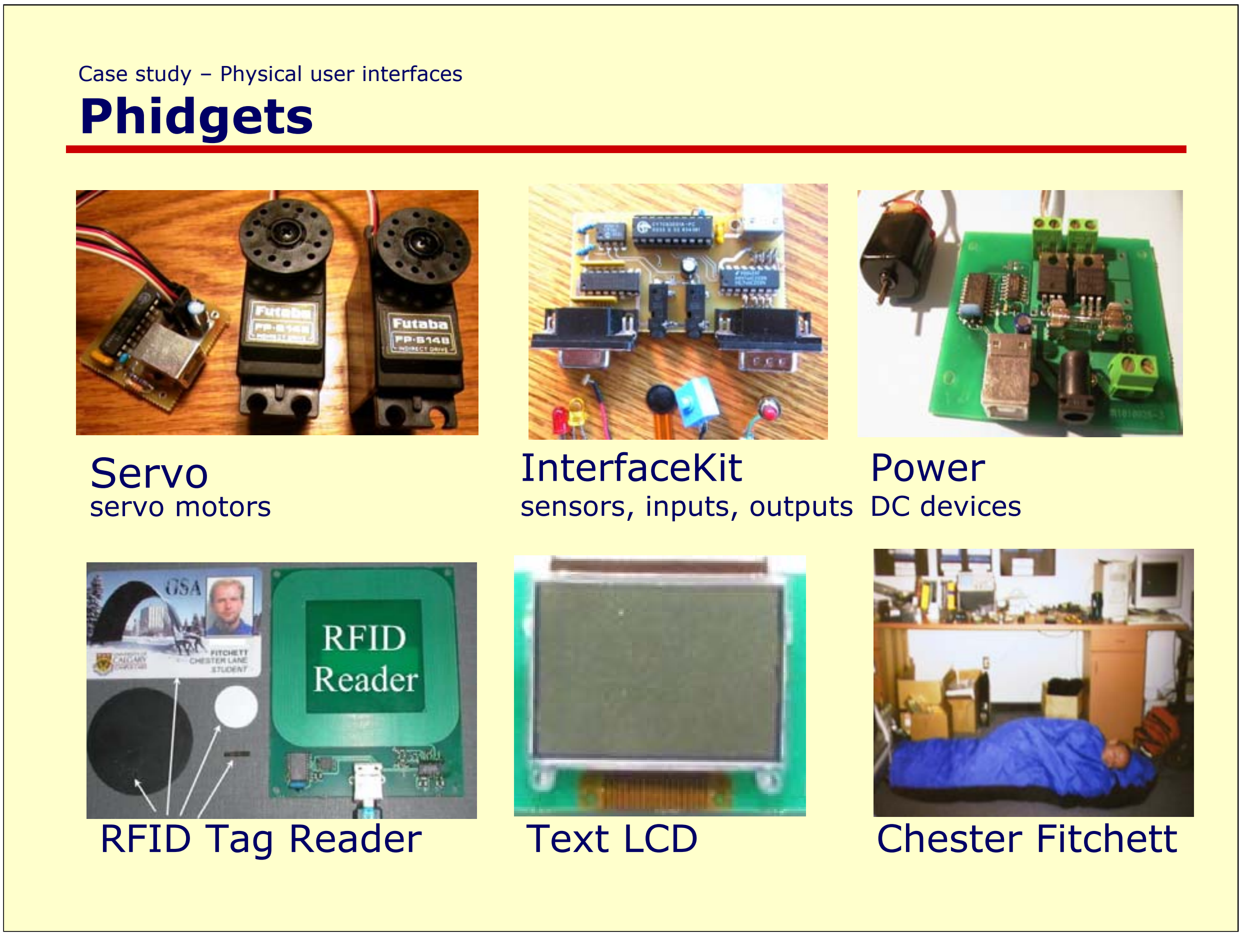
- circuit and component design
- firmware
- low level network protocols
- electrical engineering

As interface designers, we found physical interfaces hard to build.

We had neither the expertise or the interest to concern ourselves with

<READ>

Rather, we wanted to rapidly prototype systems.



Case study – Physical user interfaces

Phidgets

Servo
servo motors

InterfaceKit
sensors, inputs, outputs

Power
DC devices

RFID Tag Reader

Text LCD

Chester Fitchett

Again, we retrenched and decided to build a toolkit to make building physical user interfaces easier.

I hired Chester Fitchett, one of the few undergraduate computer science students keenly interested in hardware.

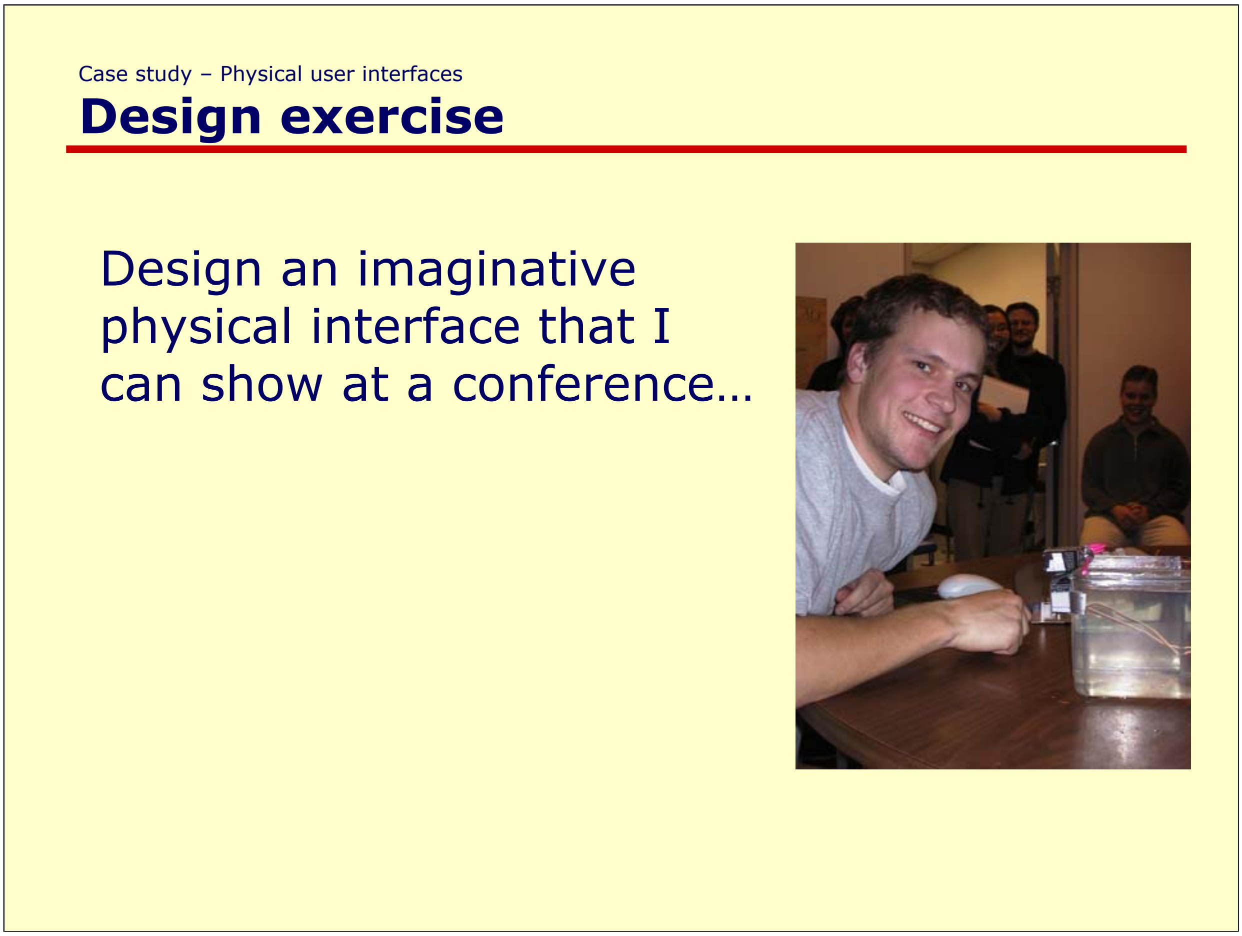
His job was to build a phidget infrastructure, as well as several phidget components.

So far, we have built several phidgets. These include <READ>…

Let me show you what its like to program these.

As with our other systems, the idea is that simple things should be simple and hard things possible.

<SHOW VB EXAMPLE OF SERVO, IF TIME OF INTERFACE KIT>



Case study – Physical user interfaces

Design exercise

Design an imaginative physical interface that I can show at a conference…

I then gave these phidgets to a group of undergradaute students who were taking an HCI course.

They had 3 weeks in total to learn how phidgets worked, to design an interface, to build it, to demonstrate it, and to produce a video.

project demonstrations

I have over 45 videos of student work, all of them inspiring examples of creativity.

I selected a few that could be used in a groupware application.

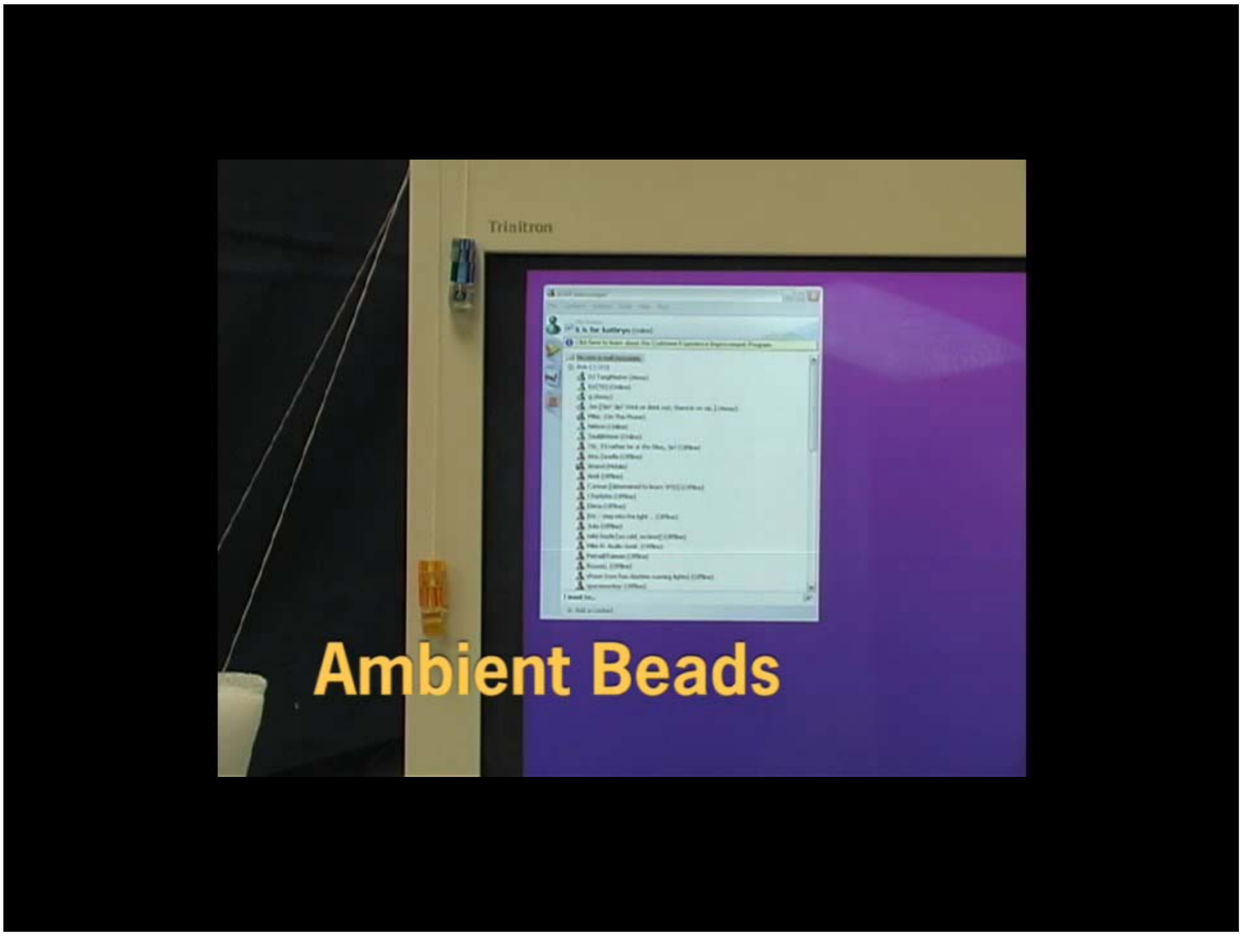
For example, the flower in bloom can be used to signal the presence of a remote person.

project demonstrations

Our next example is Phidget Eyes, which can also be used to signal presence and interests of others, albeit in a far different way

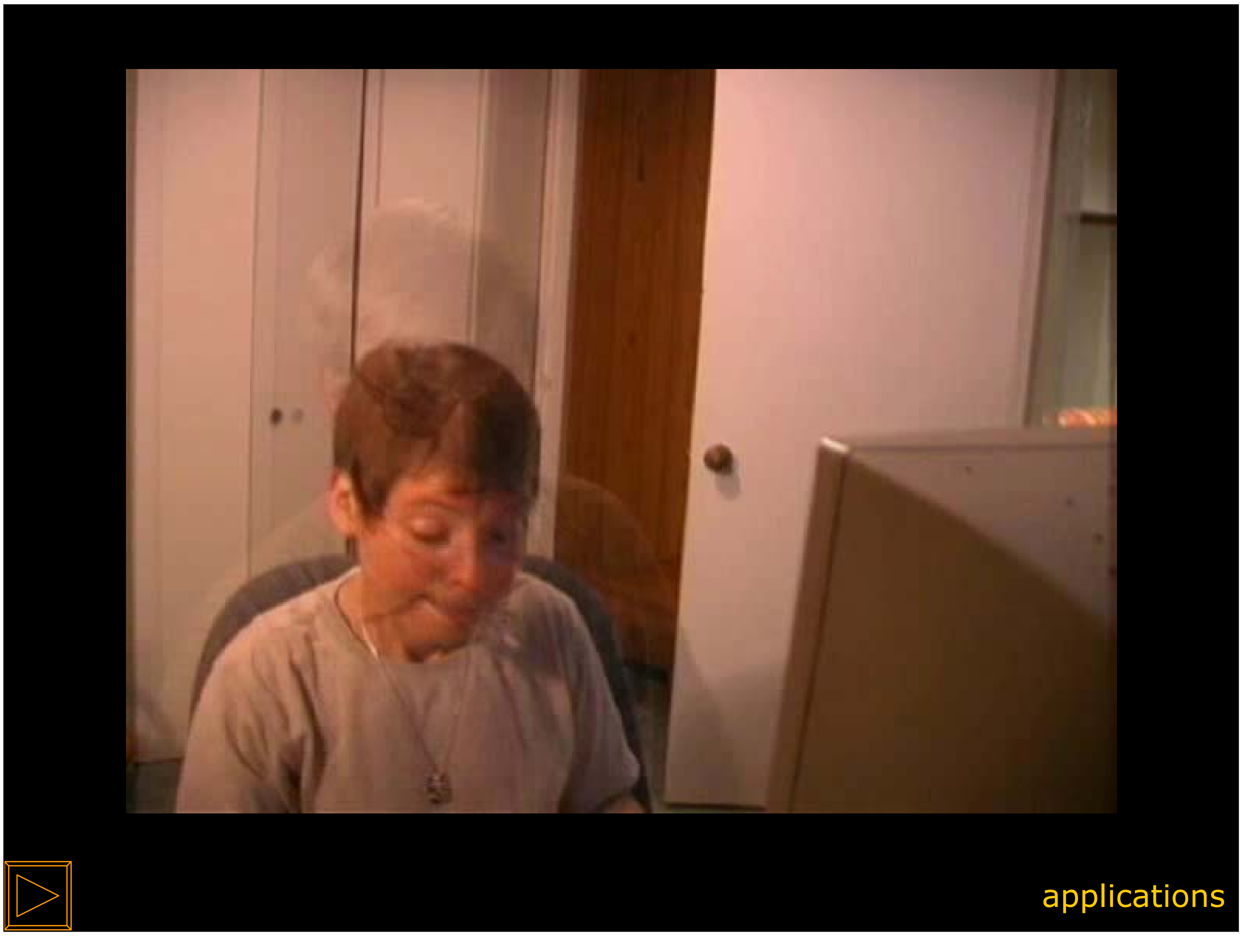
project demonstrations

Kari’s magnetic dessert could signal the presence of others in a public room, such as a coffee room.



Many students looked at physical interfaces to Instant Messenger.

Here is one that not only shows presence but lets people move into contextually relevant interaction with others.



applications

Our final example, just presented at UBICOM 2003, illustrates how Graduate student Carmen Neustaedter experimented with a context-aware privacy preserving media space that connects telecommuters and home workers. You will see how he combined the Collabrary and Phidgets toolkit into a sophisticated environment, which he has up and running in his home.

<SHOW VIDEO>

While the appearance of his space is crude, what is important is that the various toolkits let him evolve his design as he detected problems within it under everyday use.

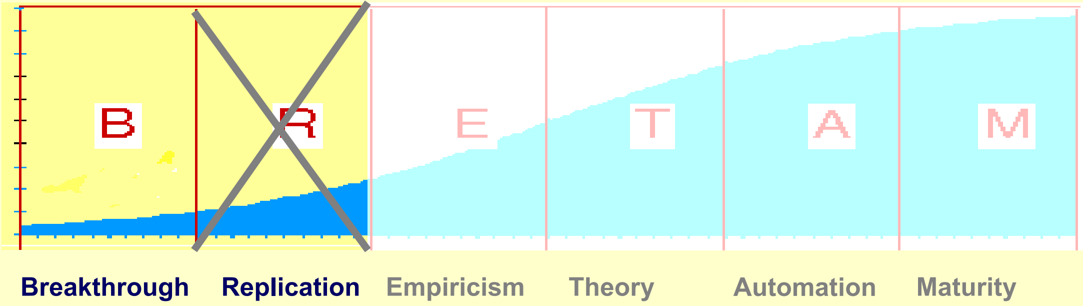
Again, toolkits led to creativity and idea evolution.

Summary – the Problem

Groupware innovation is rare because even simple ideas are too hard to implement.

- programmers lack tools to prototype / develop groupware

Poor evolution of research & creative product ideas



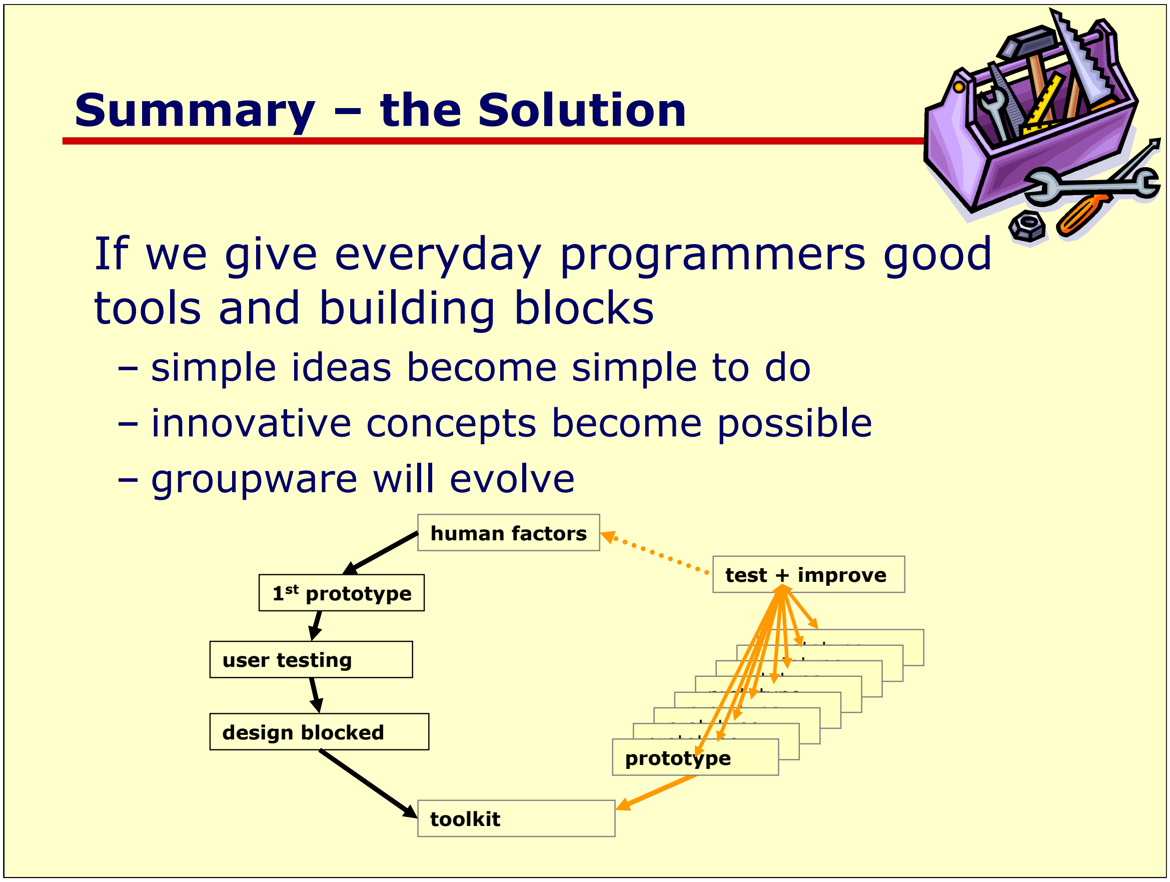
To summarize,

I started this presentation by stating what I believe is the key technical problem

<READ>

<Animate>

This has resulted in …

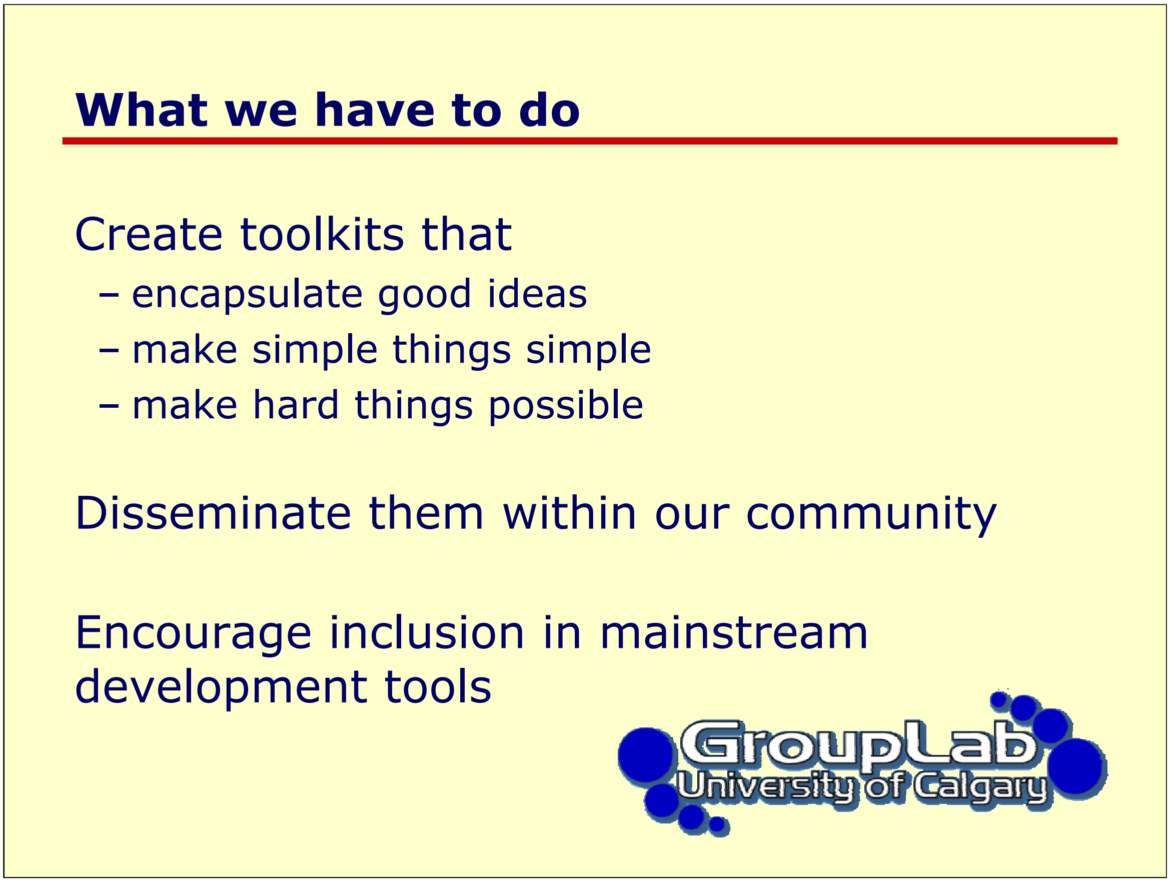


Summary – the Solution

If we give everyday programmers good tools and building blocks

- simple ideas become simple to do
- innovative concepts become possible
- groupware will evolve

The solution is, I believe, straightforward.



What we have to do

Create toolkits that

- encapsulate good ideas
- make simple things simple
- make hard things possible

Disseminate them within our community

Encourage inclusion in mainstream development tools

GroupLab
University of Calgary

To realize this solution, there are several things we have to do.

<READ>

Thank you very much for your attention, and I would be happy to answer any questions.

I would also be happy to demonstrate any of our tools/applications or show more example student projects after the seminar.