Center for
LifeLong
Learning
& Design
University of Colorado at Boulder

> Wisdom is not the product of schooling
> but the lifelong attempt to acquire it.
> - Albert Einstein

User Modeling: The Long and Winding Road

Gerhard Fischer
Center for LifeLong Learning & Design (L^{3}D)
http://www.cs.colorado.edu/~l3d/
Department of Computer Science and Institute of Cognitive Science
University of Colorado, Boulder

UM’99, Banff, June 1999

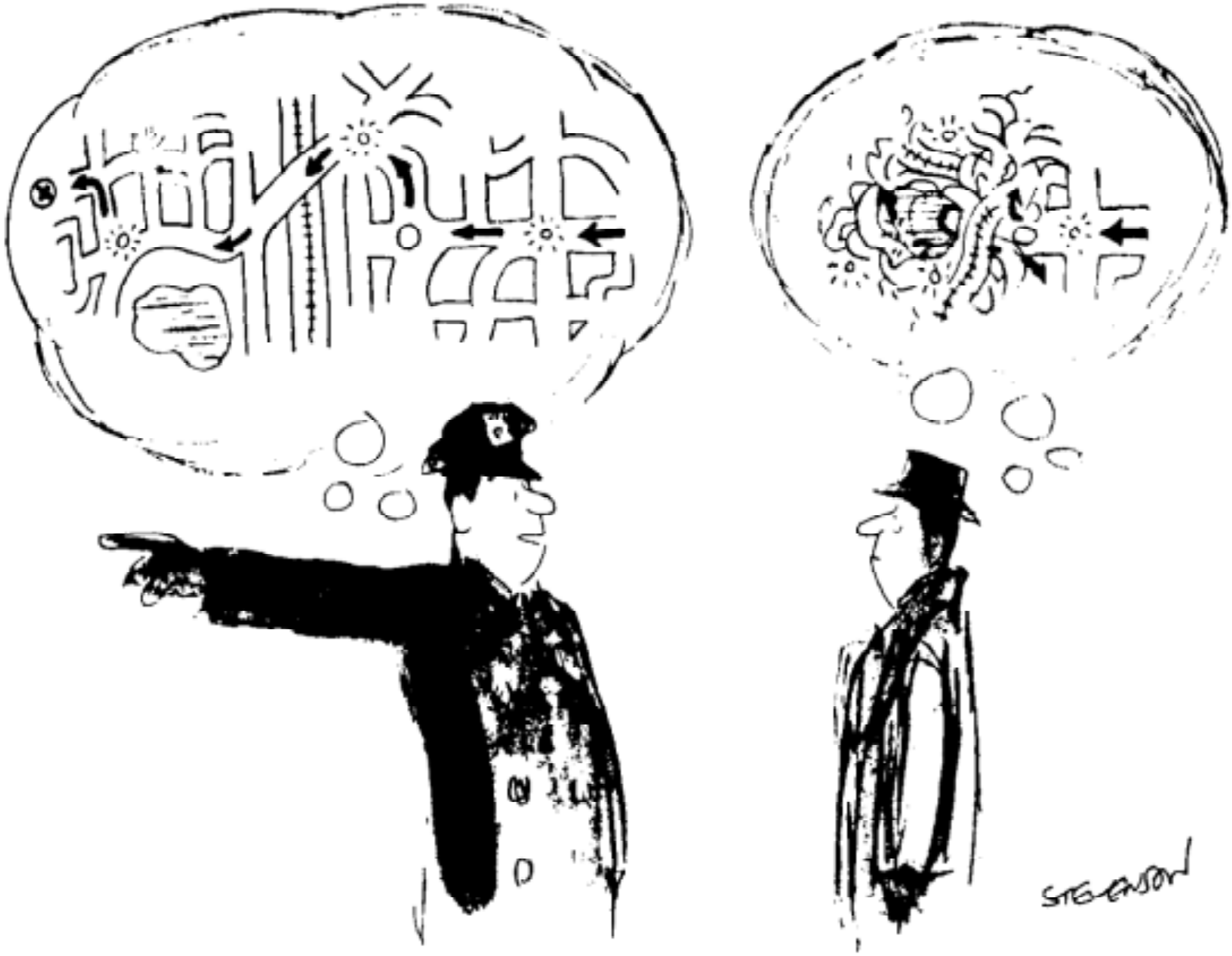
slides available at: http://www.cs.colorado.edu/~l3d/presentations/gf-um99.pdf

Overview

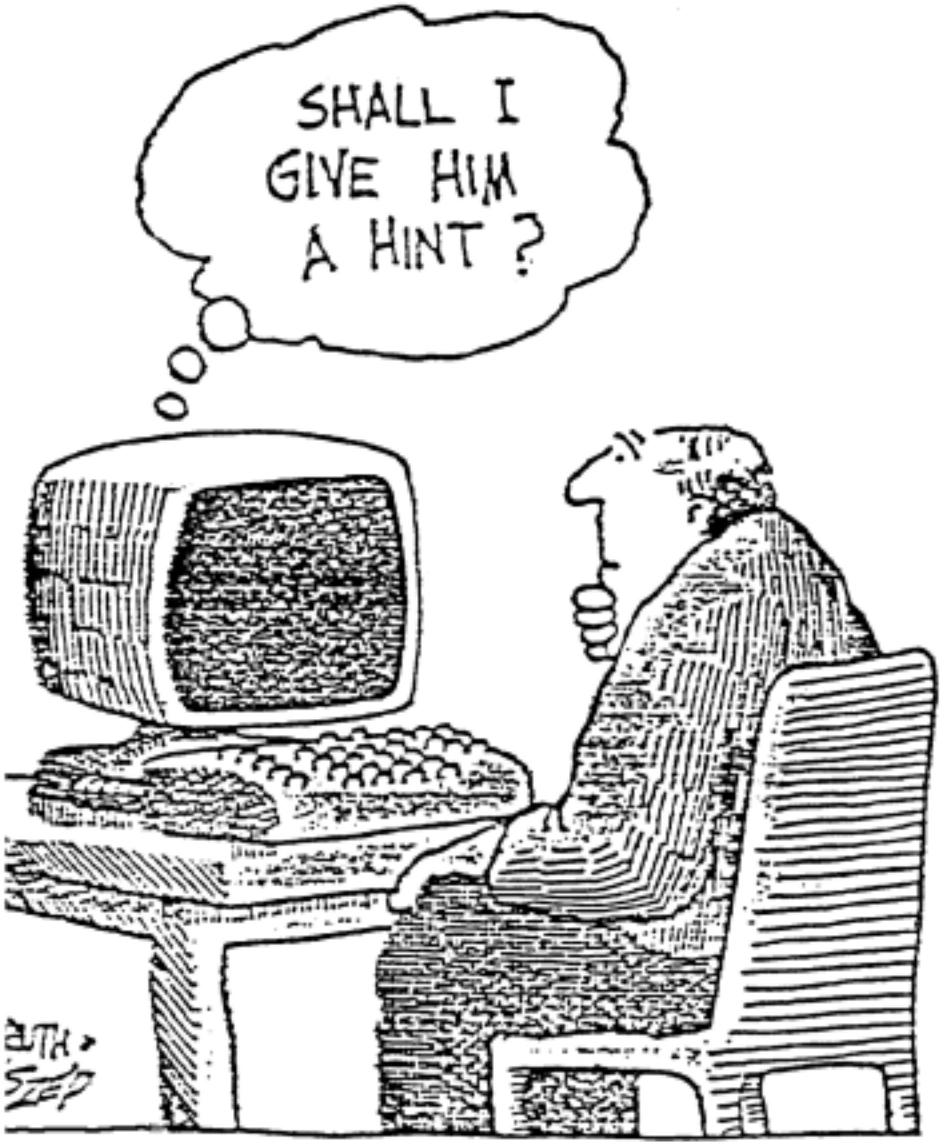
- some motivating examples
- history (global and personal)
- basic assumptions
- high-functionality applications (HFA)
- assessment
- conclusion

many thanks to: Ph.D. and Master students, members of L3D, Bernd Gutkauf, Tom Mastaglio, Christoph Thomas, Stefanie Thies

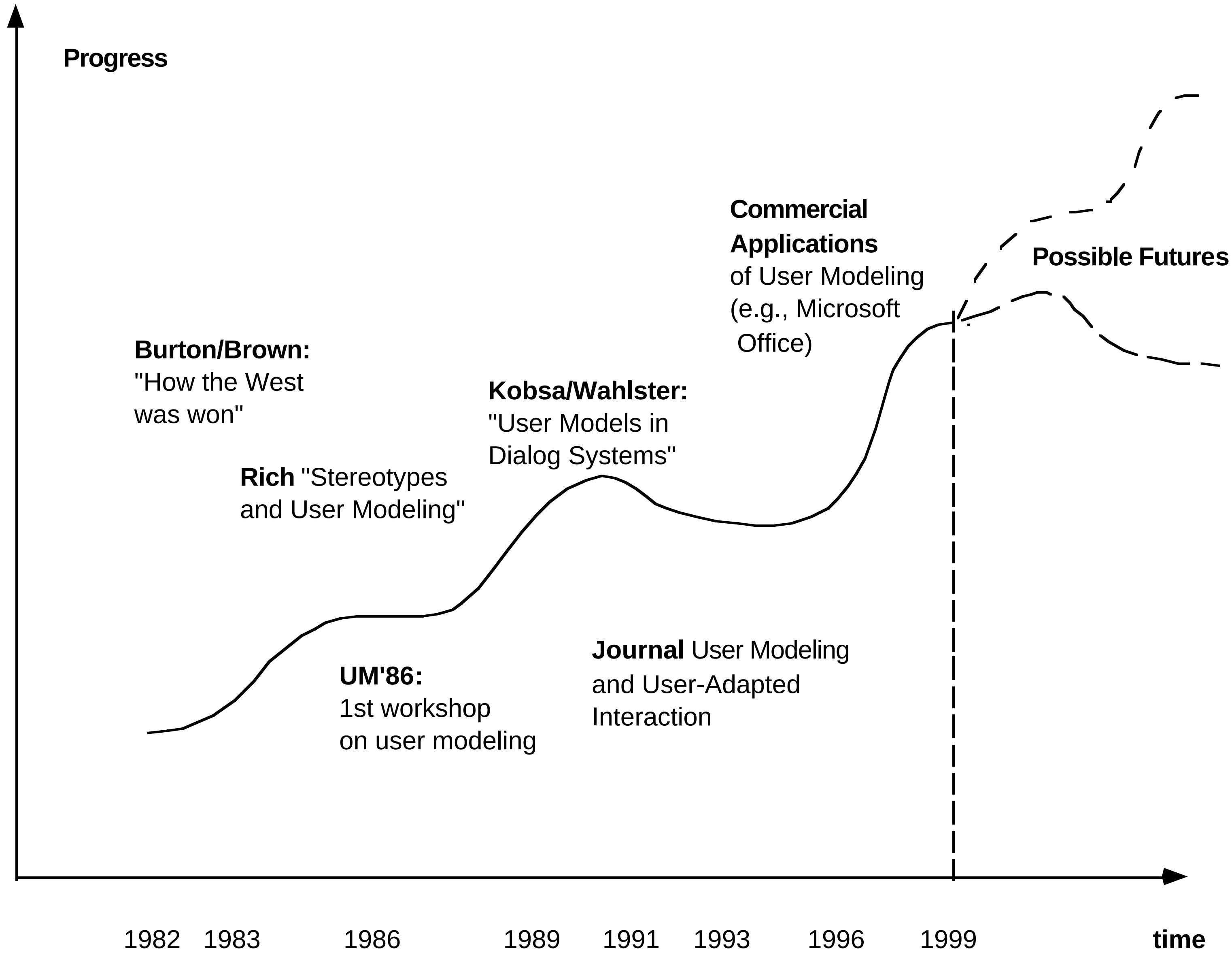
Shared Understanding and Background Knowledge



Information Delivery, Contextualization and Intrusiveness



User Modeling: Global Developments



Burton/Brown: "How the West was won"

Rich "Stereotypes and User Modeling"

UM'86: 1st workshop on user modeling

Kobsa/Wahlster: "User Models in Dialog Systems"

Journal User Modeling and User-Adapted Interaction

Commercial Applications of User Modeling (e.g., Microsoft Office)

Possible Futures

Two Major Approaches in Human-Computer Collaboration

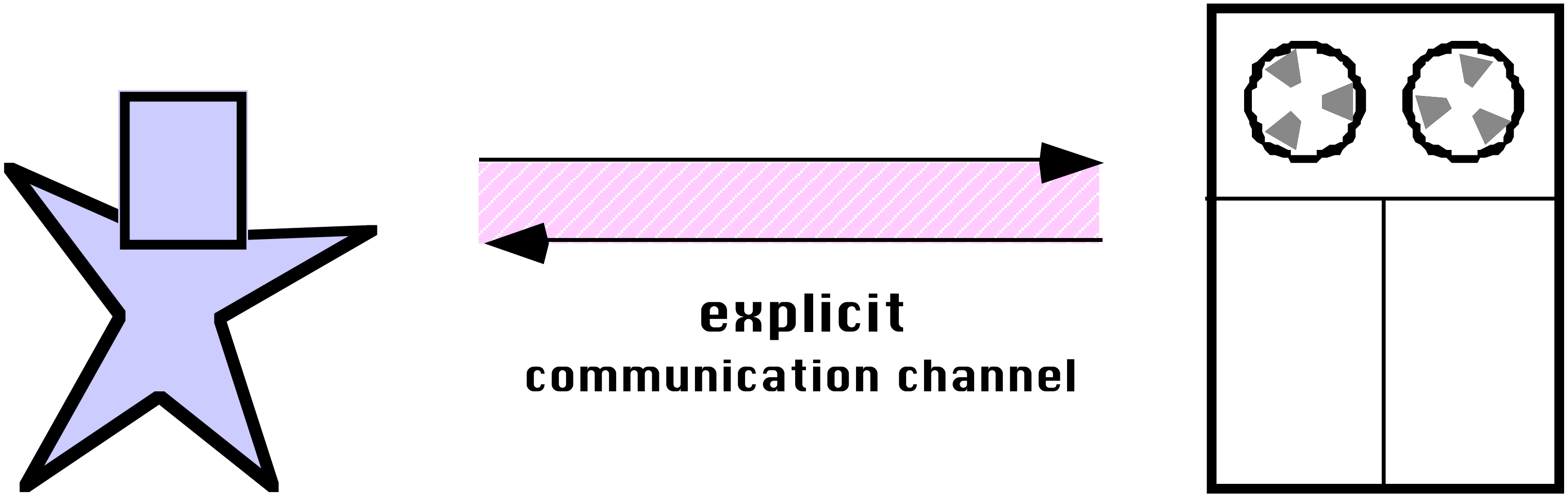
- **emulation or replacement approach (narrow AI approach)**
  - one major motivation for user modeling research
  - claim: in many cases, it is not the preferred mode of communication

- **claims** (Lucy Suchman, 1987)
  - interaction between people and computers requires essentially the same interpretive work that characterizes interaction between people, but with fundamentally different resources available to the participants
  - people make use of linguistic, nonverbal, and inferential resources in finding the intelligibility of actions and events, which are in most cases not available and not understandable by computers

- **complementary approach (IA = Intelligence Augmentation approach)**
  - based on the asymmetry between human and computer
  - the design of the collaboration
    - ➢ is not only a problem of simulating human to human collaboration
    - ➢ but of inventing engineering alternatives to interaction related properties

The Beginning of Human-Computer Interaction (HCI)

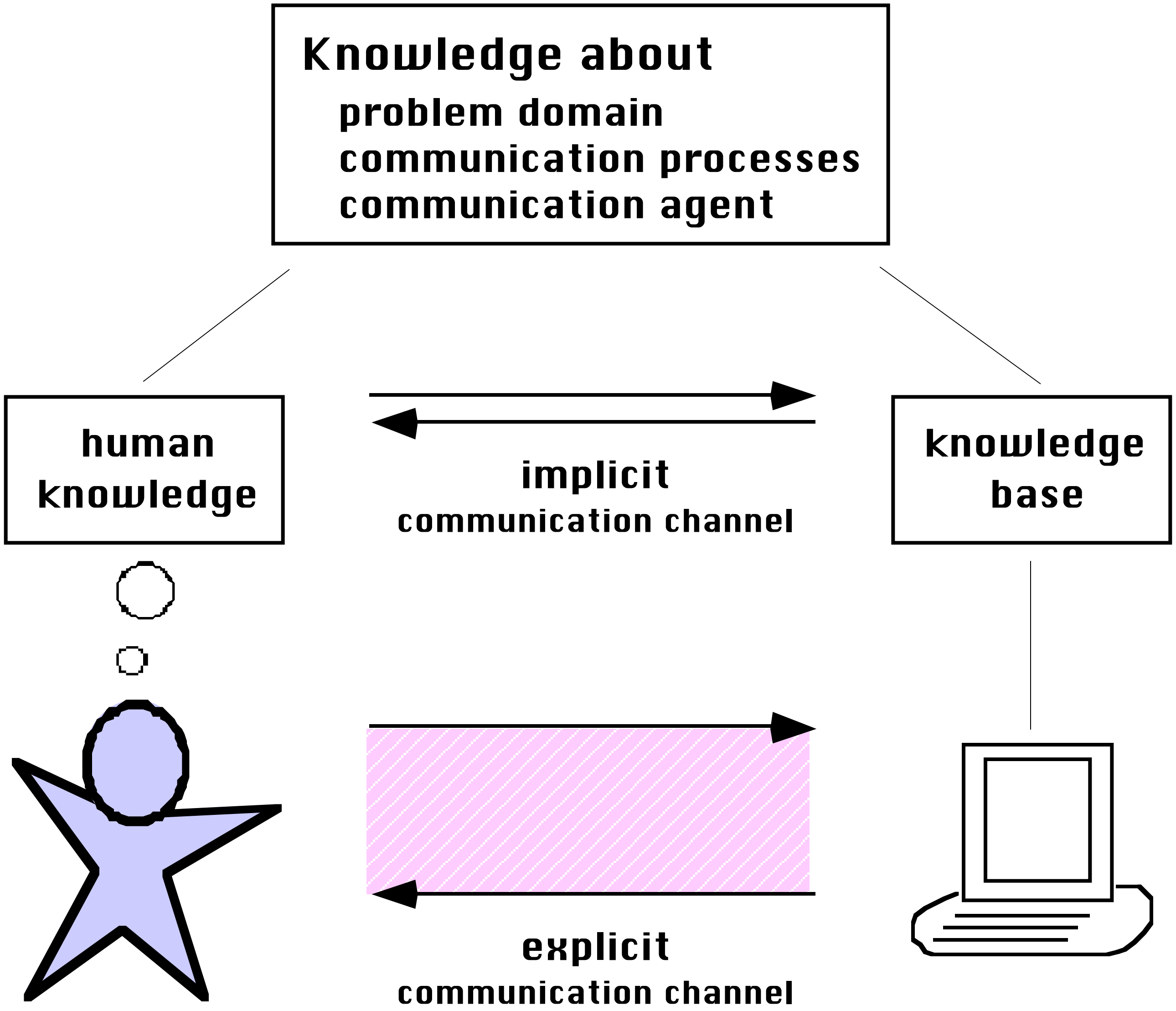
Human and Computer connected by a narrow explicit communication channel



explicit

communication channel

Knowledge-Based Human Computer Collaboration



Knowledge about

- problem domain
- communication processes
- communication agent

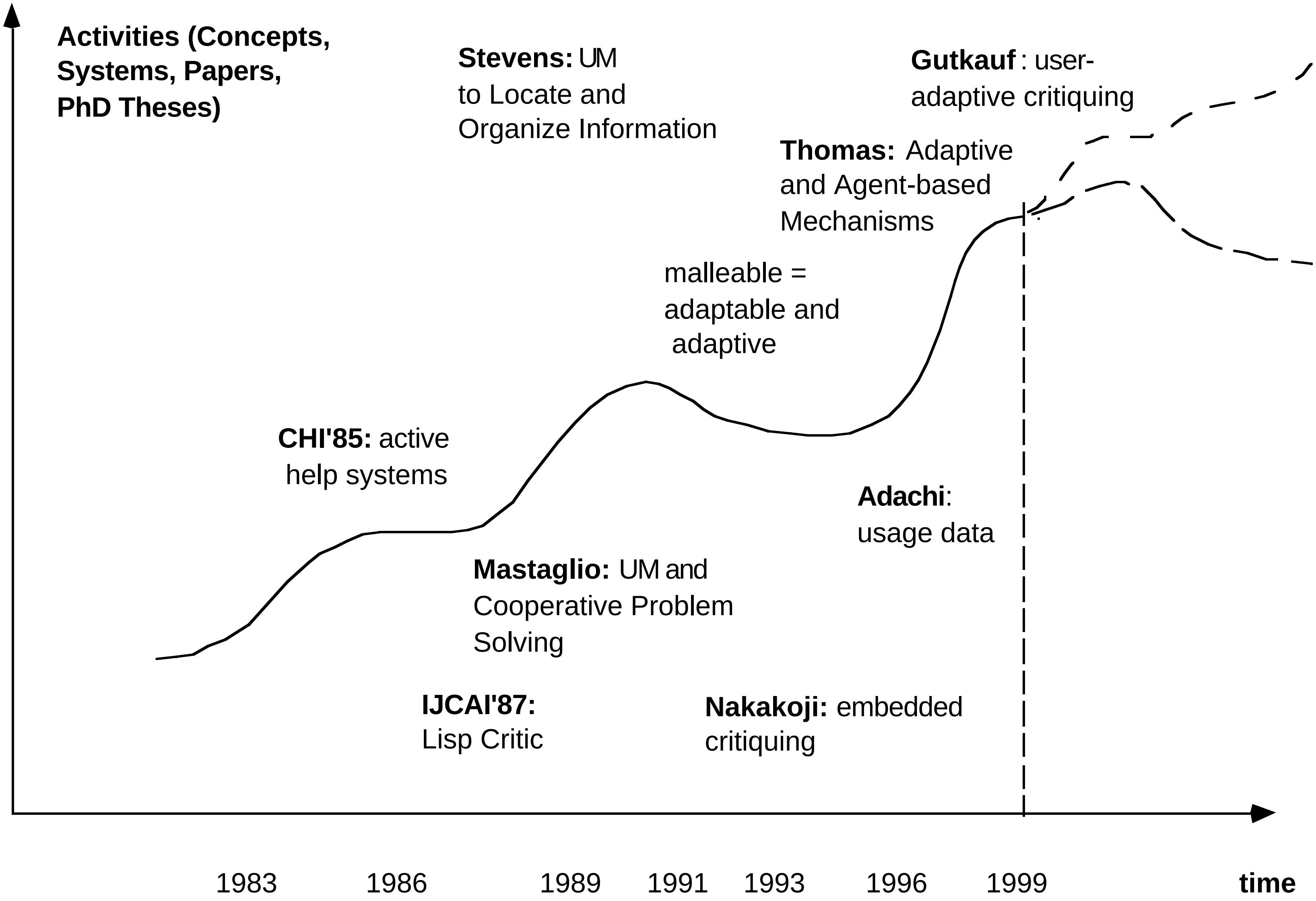
human knowledge

implicit communication channel

knowledge base

explicit communication channel

User Modeling: L^{3}D contributions



CHI'85: active help systems

IJCAI'87: Lisp Critic

Mastaglio: UM and Cooperative Problem Solving

Stevens: UM to Locate and Organize Information

malleable = adaptable and adaptive

Nakakoji: embedded critiquing

Adachi: usage data

Thomas: Adaptive and Agent-based Mechanisms

Gutkauf : user-adaptive critiquing

Ph.D. Theses in the Area of UMUAI

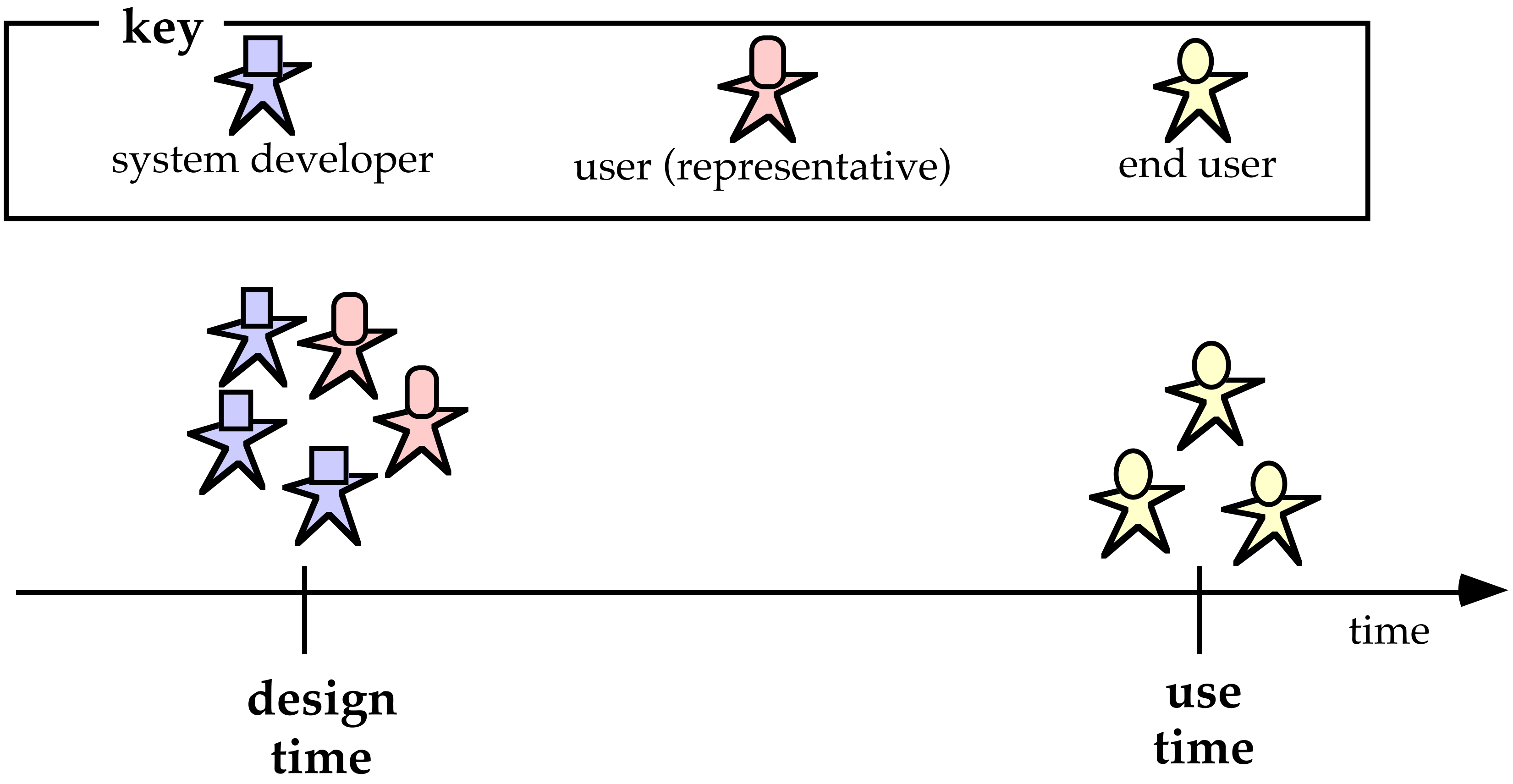
- **Mastaglio, T. (1991)** — A User-Modeling Approach to Cooperative Problem Solving
- **Stevens, C. (1993)** — Helping Users Locate and Organize Information
- **Nakakoji, K. (1993)** — Increasing Shared Understanding of a Design Task Between Designers and Design Environments: The Role of a Specification Component
- **Thomas, C. (1996)** — To Assist the User: On the Embedding of Adaptive and Agent-based Mechanisms (University of Koblenz, Germany)
- **Gutkauf, B. (1998) —** Improving Design & Communication of Business Graphs Through User Adaptive Critiquing (University of Paderborn, Germany)
- **Adachi, T. (1998)** — Utilization of Usage Data to Improve Organizational Memory (Master's Thesis)

User Modeling in Different Domains and Important Distinctions

- **different domains:**
  - natural language dialog
  - human computer interaction
  - intelligent tutoring systems
    - ➢ tasks known at design time
    - ➢ relatively uniform background knowledge among users
  - domain-oriented design environments including critiquing
    - ➢ tasks need to be inferred at use time
    - ➢ diverse background knowledge among users
    - ➢ lifelong learning
  - high-functionality applications (HFA)

- **important distinctions:**
  - adaptive versus adaptable components
  - explicit versus implicit modeling techniques
  - user models versus task models
  - canonical versus individual models
  - long-term versus short-term models

Design Time and Use Time



the fundamental challenge: how do you write software for millions of users (at design time), while making it work as if it was designed for each individal user (only known at use time)?

Some Challenging Research Problems

- **identify user goals from low-level interactions**
  - active help systems, data detectors
  - *“every wrong answer is the right answer to some other question”*

- **integrate different modeling techniques**
  - domain-orientation
  - explicit and implicit
  - give a user specific problems to solve

- **capture the larger (often unarticulated) context and what users are doing** (especially beyond the direct interaction with the computer system)
  - embedded communication
  - ubiquitous computing

- **reduce information overload by making information relevant**
  - to the task at hand
  - to the assumed background knowledge of the users

- **support differential descriptions** (relate new information to information and concepts assumed to be known by the user)

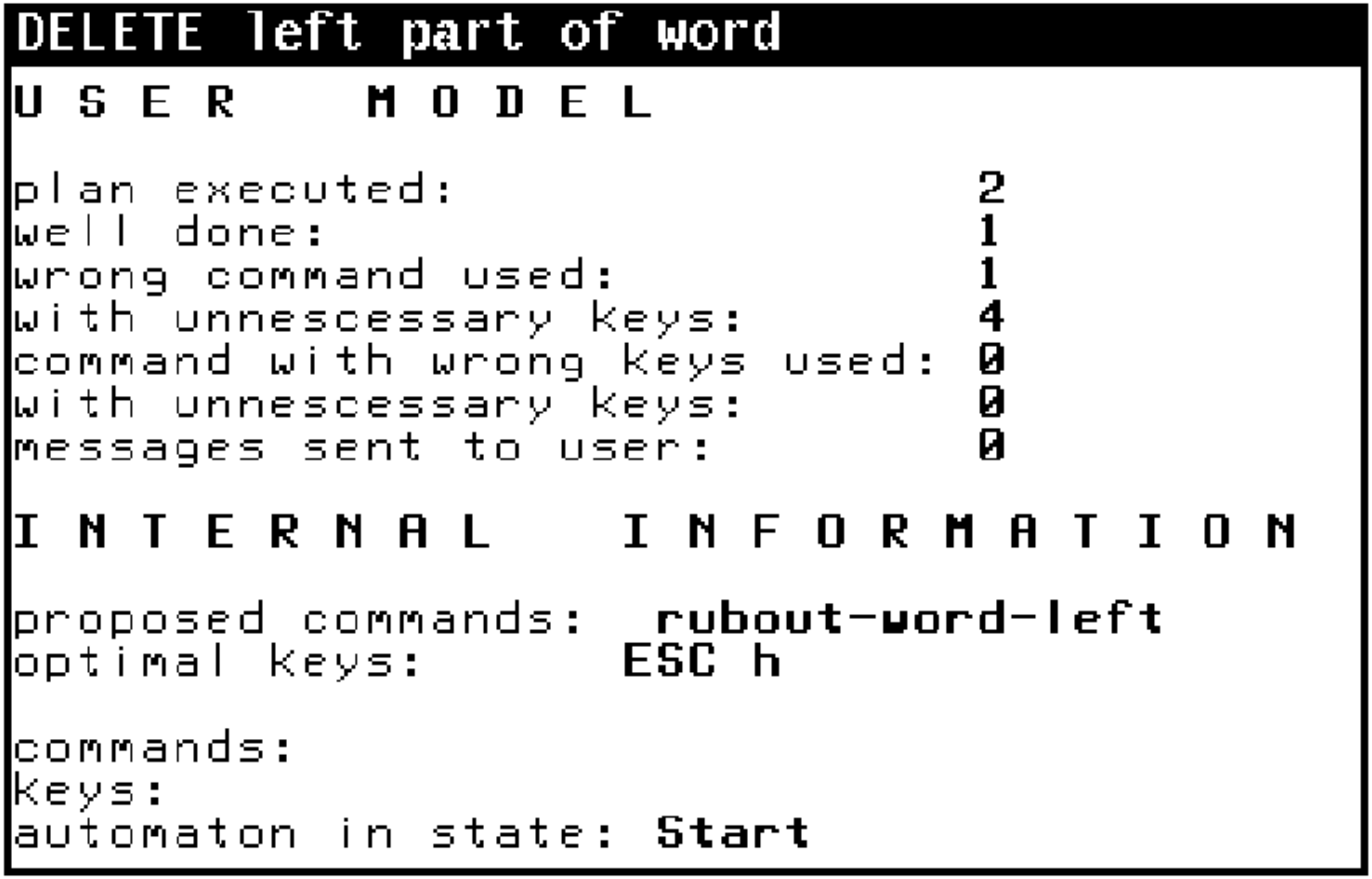
Early Example: Knowledge-Based Help Systems (CHI'85) — Activist (and Passivist)

- **Activist — an active help system** for an EMACS-like editor, deals with two different kinds of suboptimal behavior:
  - the user does not know a complex command and uses "suboptimal" commands to reach a goal ("suboptimal": main streets and side streets?)
  - the user knows the complex command but does not use the minimal key sequence to issue the command

- similar to a human observer, **Activist handles the following tasks**:
  - recognizes what the user is doing or wants to do
  - evaluates how the user tries to achieve his/her goal
  - constructs a model of the user based on the results of the evaluation task
  - decides (dependent on the information in the model) *when* and *how* to interrupt (tutorial intervention)

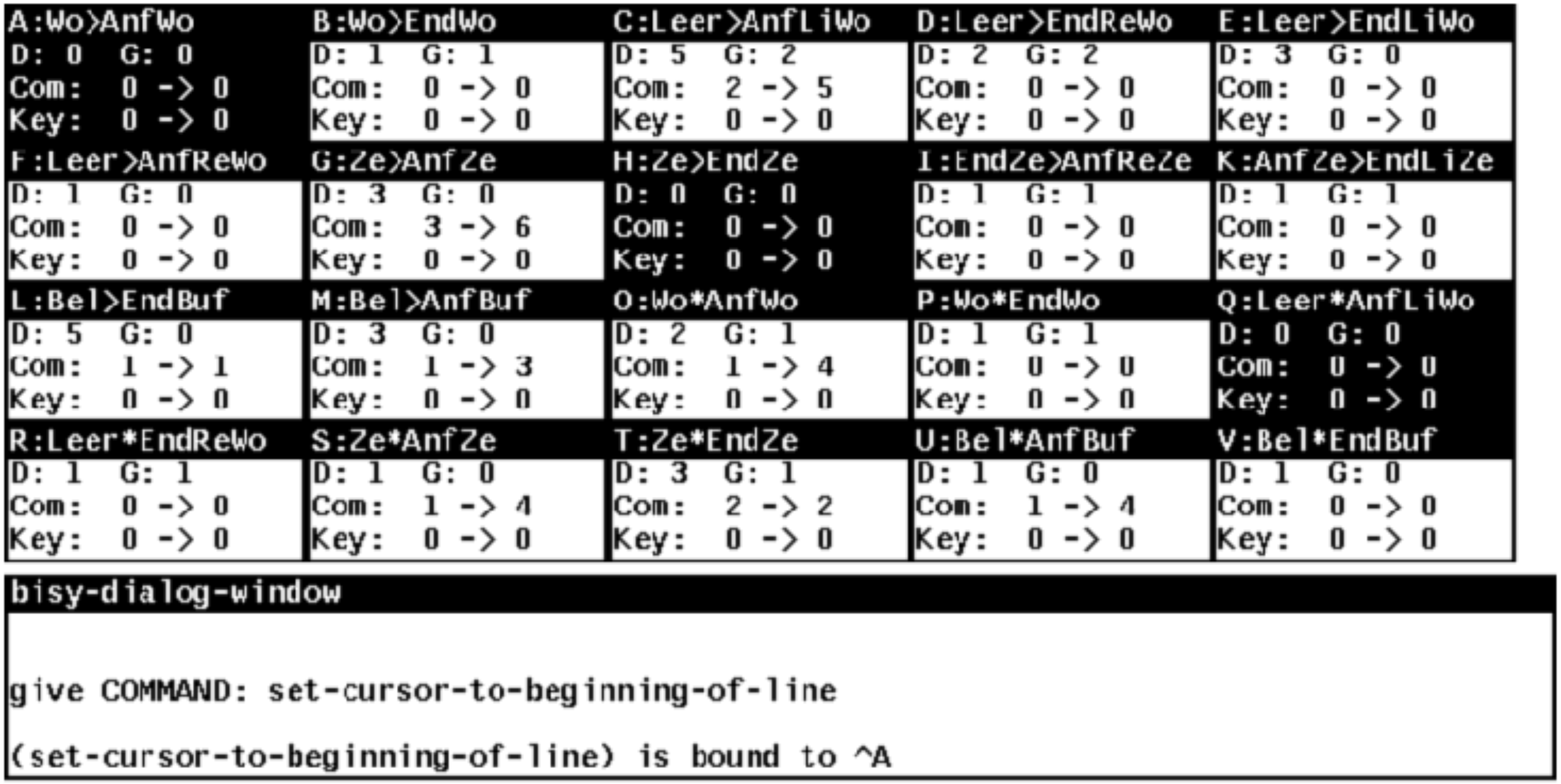
- the recognition and evaluation task is delegated to **20 different plan specialists**

A Plan Specialist in Activist



```
DELETE left part of word
U S E R   M O D E L

plan executed:                    2
well done:                        1
wrong command used:               1
with unnescessary keys:           4
command with wrong keys used:     0
with unnescessary keys:           0
messages sent to user:            0

I N T E R N A L   I N F O R M A T I O N

proposed commands:  rubout-word-left
optimal keys:       ESC h

commands:
keys:
automaton in state: Start
```

The User Model in Activist



```
A:Wo>AnfWo        B:Wo>EndWo        C:Leer>AnfLiWo    D:Leer>EndReWo    E:Leer>EndLiWo
D: 0   G: 0       D: 1   G: 1       D: 5   G: 2       D: 2   G: 2       D: 3   G: 0
Com:   0 -> 0     Com:   0 -> 0     Com:   2 -> 5     Com:   0 -> 0     Com:   0 -> 0
Key:   0 -> 0     Key:   0 -> 0     Key:   0 -> 0     Key:   0 -> 0     Key:   0 -> 0
F:Leer>AnfReWo    G:Ze>AnfZe        H:Ze>EndZe        I:EndZe>AnfReZe   K:AnfZe>EndLiZe
D: 1   G: 0       D: 3   G: 0       D: 0   G: 0       D: 1   G: 1       D: 1   G: 1
Com:   0 -> 0     Com:   3 -> 6     Com:   0 -> 0     Com:   0 -> 0     Com:   0 -> 0
Key:   0 -> 0     Key:   0 -> 0     Key:   0 -> 0     Key:   0 -> 0     Key:   0 -> 0
L:Bel>EndBuf      M:Bel>AnfBuf      O:Wo*AnfWo        P:Wo*EndWo        Q:Leer*AnfLiWo
D: 5   G: 0       D: 3   G: 0       D: 2   G: 1       D: 1   G: 1       D: 0   G: 0
Com:   1 -> 1     Com:   1 -> 3     Com:   1 -> 4     Com:   0 -> 0     Com:   0 -> 0
Key:   0 -> 0     Key:   0 -> 0     Key:   0 -> 0     Key:   0 -> 0     Key:   0 -> 0
R:Leer*EndReWo    S:Ze*AnfZe        T:Ze*EndZe        U:Bel*AnfBuf      V:Bel*EndBuf
D: 1   G: 1       D: 1   G: 0       D: 3   G: 1       D: 1   G: 0       D: 1   G: 0
Com:   0 -> 0     Com:   1 -> 4     Com:   2 -> 2     Com:   1 -> 4     Com:   0 -> 0
Key:   0 -> 0     Key:   0 -> 0     Key:   0 -> 0     Key:   0 -> 0     Key:   0 -> 0
```

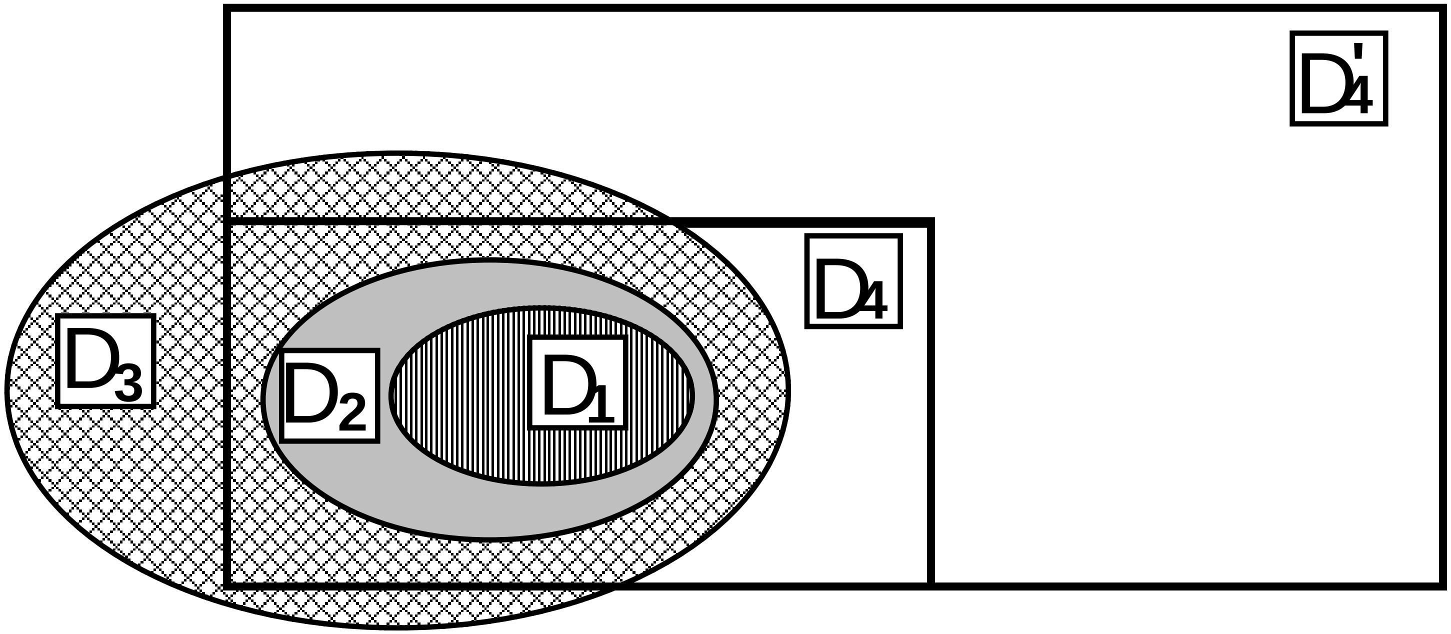
```
bisy-dialog-window


give COMMAND: set-cursor-to-beginning-of-line

(set-cursor-to-beginning-of-line) is bound to ^A
```

- **black boxes:** plan specialist currently switched off
- **white boxes:** plan specialist currently switched on

High Functionality Applications (HFA)

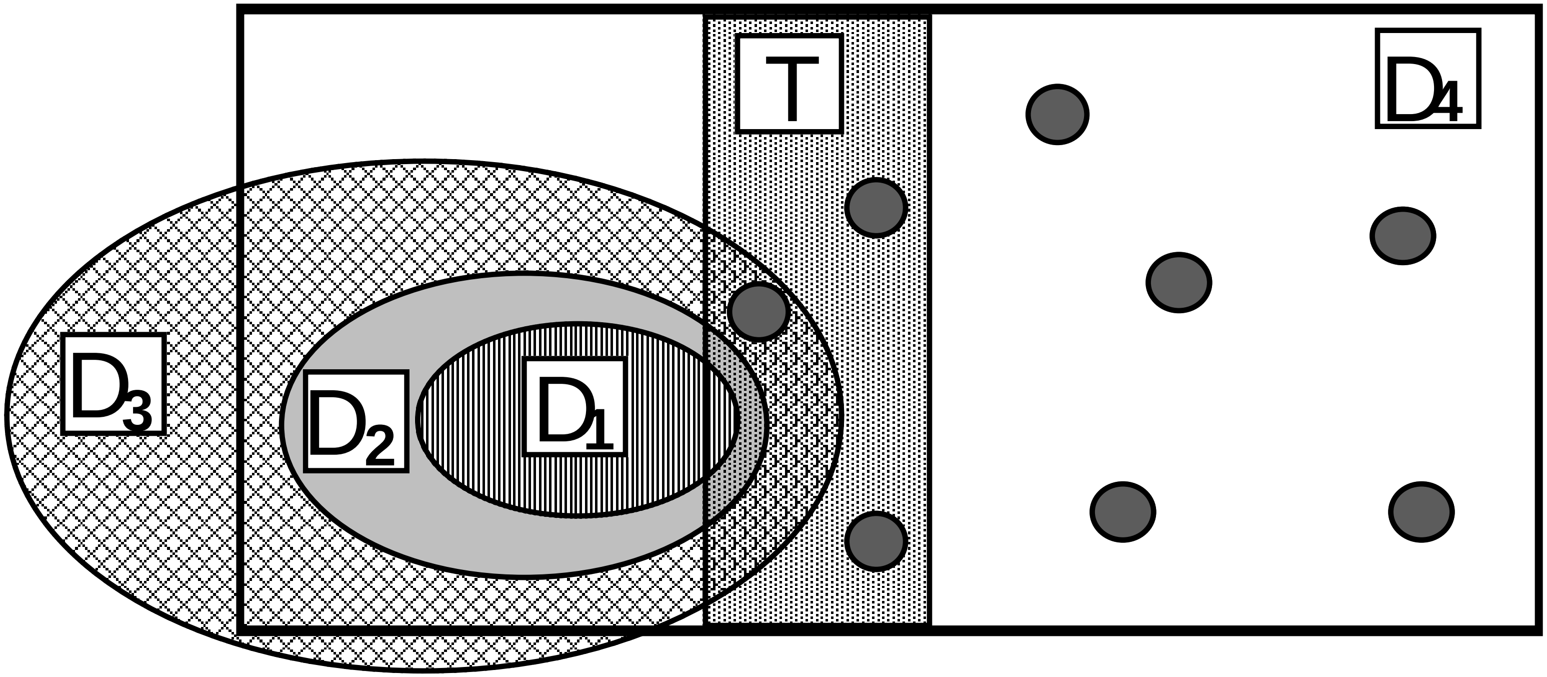
Levels of Users' Knowledge About a System's Information Spaces

(based on numerous empirical investigations)

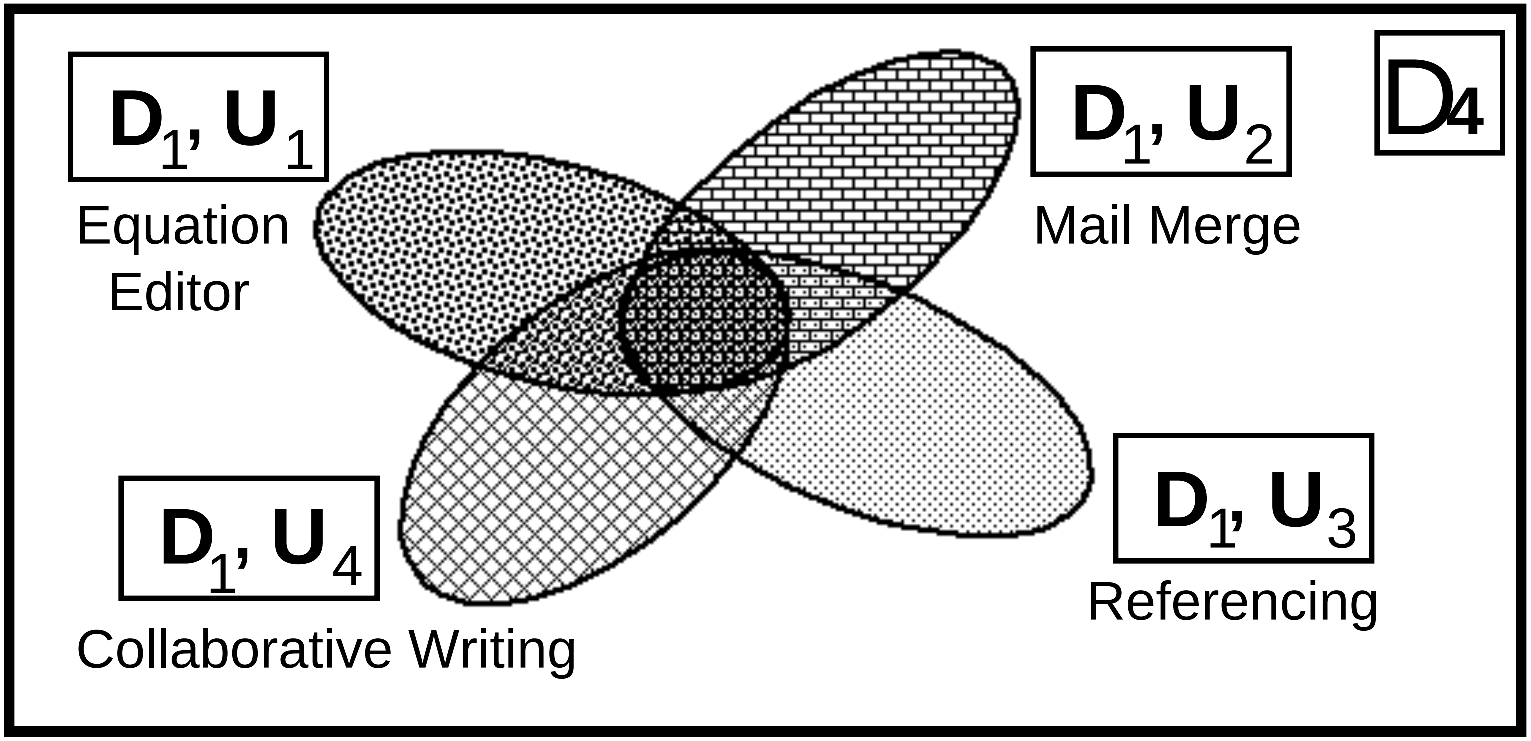


Functionality and its Relevancy to the Task at Hand in HFAs

Why “Did You Know (DYK)” and “MS Tip of the Day” is of limited success



Expertise in HFAs is an Attribute of a Context, not of a Person



Problems with HFA: L3D

- users do not know about the existence of tools $(D_4 \neg \wedge D_3)$

- users do not know how to access tools

- users do not know when to use tools (lack applicability conditions)

- users cannot combine, adapt, and modify tools according to their specific needs

- additional complicating factor: in the real world problems are not given but emerge, implying that no precise goals and specifications can be articulated -> intertwining of problem formulation and problem solving

Problems with HFA: Microsoft’s View and Objectives

- some "routine" tasks could be and needed to be automated
- some tasks were used too infrequently by users to make it worthwhile for them to learn how to complete them and complex enough that users would need to relearn how to perform them each time they tried to accomplish the task
- complex tasks may include options that could benefit the users - options that the user might never take advantage of
- users have different levels of expertise and backgrounds and therefore require different levels of support
- tasks supported by software are broad
- users don't want to become technical experts, they just want to get their tasks done
- users don't know about all software features that could help them
- help is insufficient, spread out over the user interface, hard to use, and requires prior knowledge of computer software lingo
- users want tailored help delivered in a friendly and easy to understand manner

How Our Research Addresses the Problems Created by HFAs

- *active help systems* — analyze the behavior of users and infer higher-level goals from low-level operations

- *specification components* — allow users to enrich the description of their tasks

- *critiquing components* — analyze and infer the task at hand; detect and identify the potential for a design information need; present contextualized knowledge for designers

- *increase user and task relevance* by integrating specification component and critiquing components; *generic critics* (defined at design time) → *specific critics* (information only known at use time)

- *create malleable systems* by integrating adaptive and adaptable components

Embedding Critics

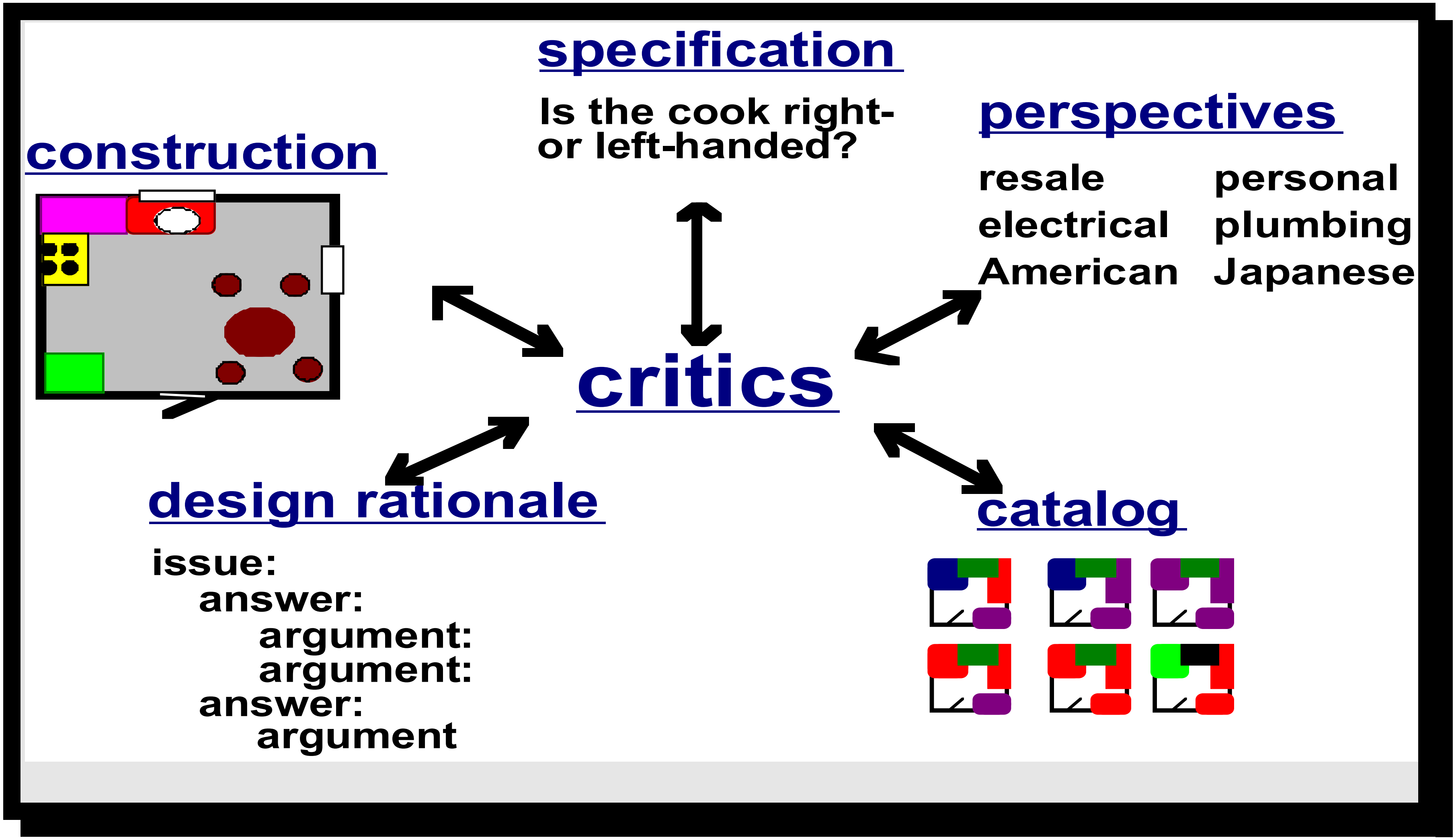
Saying the “right” thing

at the “right” time,

in the “right” way

- **benefits of embedding critics**
  - integrate design environment components
  - allow system to infer the task at hand and user characteristics
  - enable only the most relevant critic rules
  - modify critic rules to reflect task at hand and user characteristics
  - deliver more relevant information

Domain-Oriented Design Environments



specification

Is the cook right- or left-handed?

construction

perspectives

resale	personal
electrical	plumbing
American	Japanese

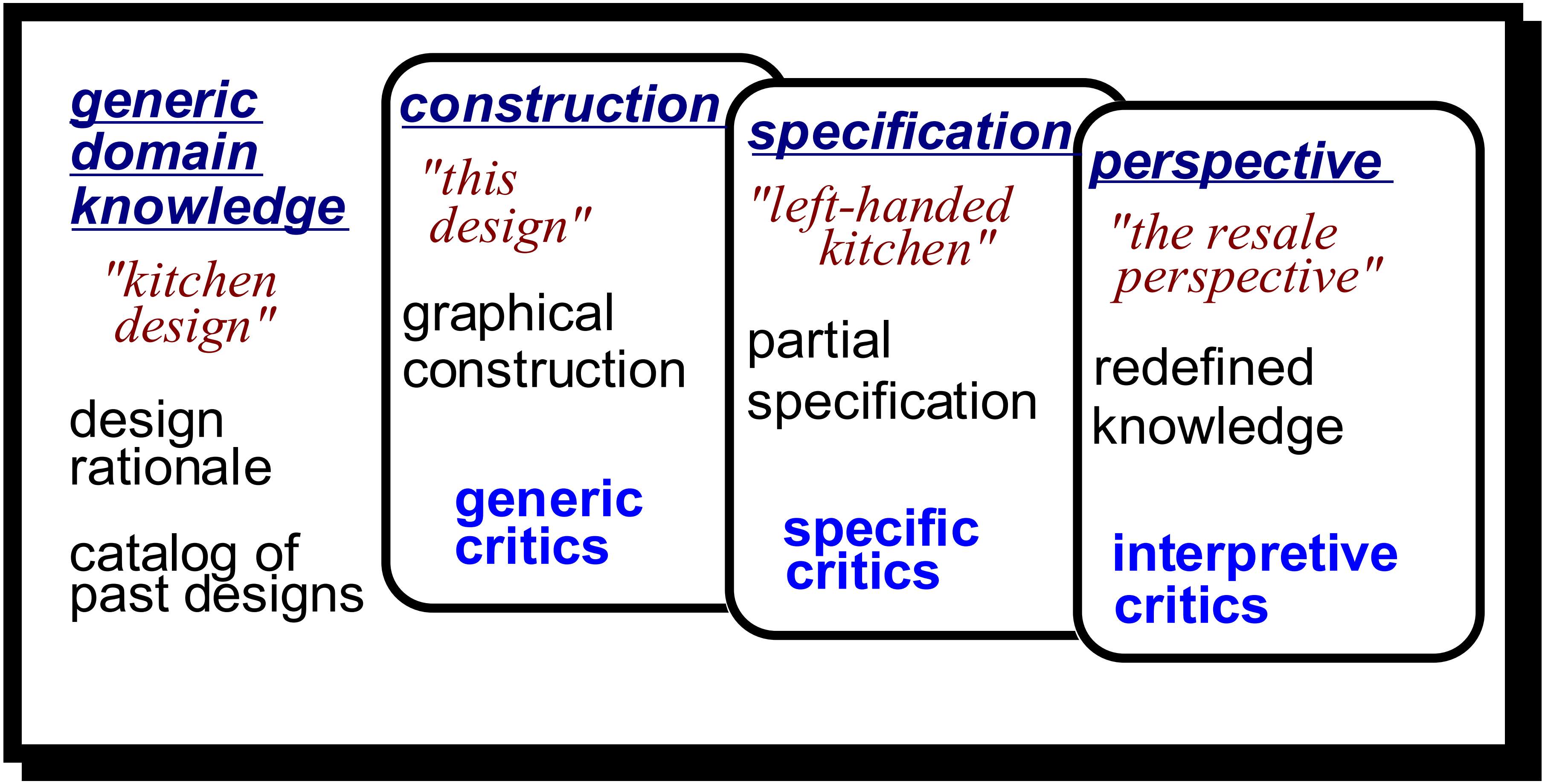
critics

design rationale

issue:
- answer:
  - argument:
  - argument:
- answer:
  - argument

catalog

Embedding Critics in the Contexts of Design (Context defined by appropriate User Characteristics)



generic domain knowledge

"kitchen design"

design rationale

catalog of past designs

construction

"this design"

graphical construction

generic critics

specification

"left-handed kitchen"

partial specification

specific critics

perspective

"the resale perspective"

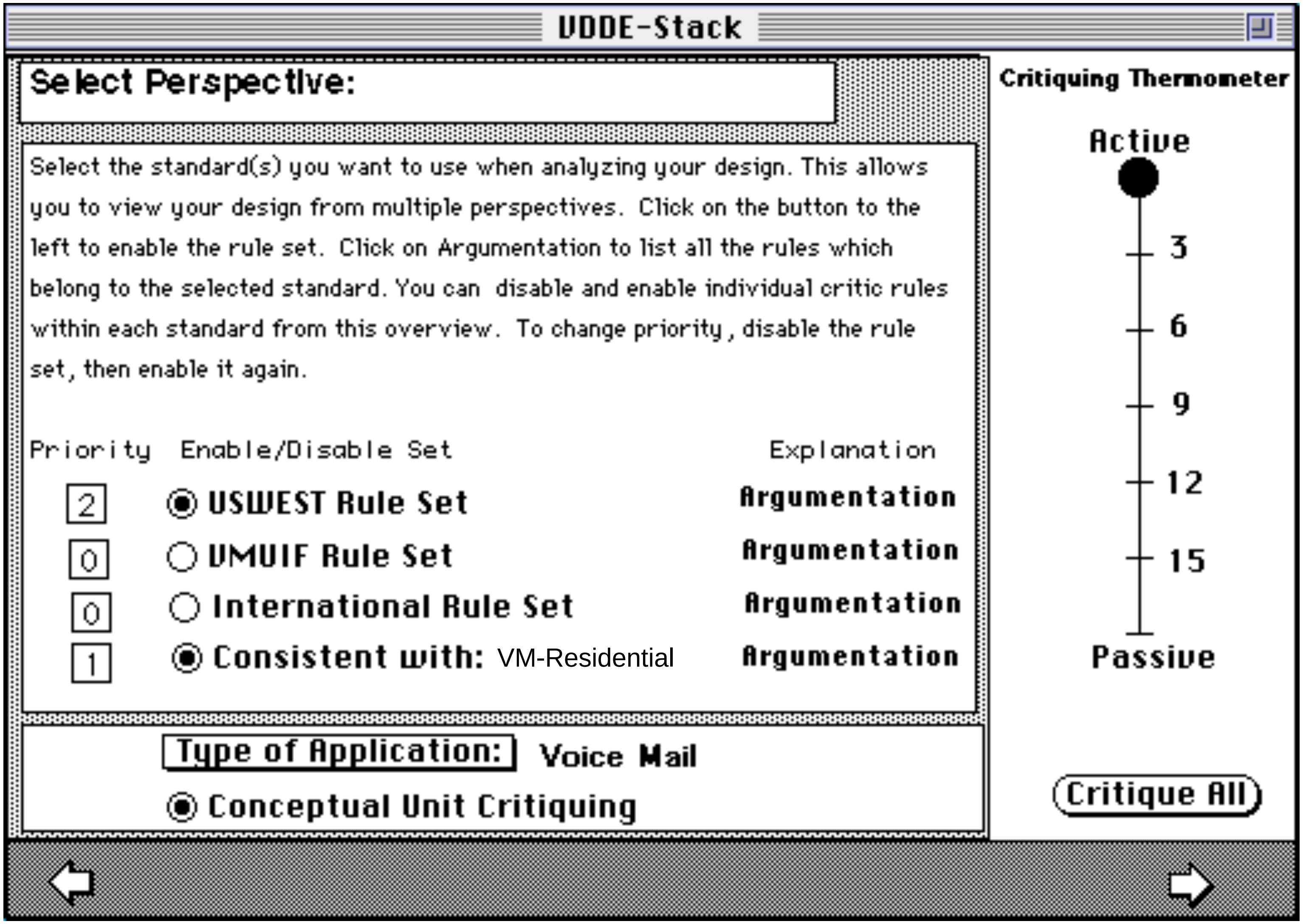
redefined knowledge

interpretive critics

A Comparison Between Adaptive and Adaptable Systems

	Adaptive	Adaptable
Definition	dynamic adaptation by the system itself to current task and current user	user changes (with substantial system support) the functionality of the system
Knowledge	contained in the system; projected in different ways	knowledge is extended
Strengths	little (or no) effort by the user; no special knowledge of the user is required	user is in control; system knowledge will fit better; success model exists
Weaknesses	user has difficulty developing a coherent model of the system; loss of control; few (if any) success models exist (except humans)	systems become incompatible; user must do substantial work; complexity is increased (user needs to learn the adaptation component)
Mechanisms Required	models of users, tasks and dialogs; knowledge base of goals and plans; powerful matching capabilities; incremental update of models	layered architecture; human problem-domain communication; “back-talk" from the system; design rationale
Application Domains	active help systems; critiquing systems; differential descriptions; user interface customization	end-user modifiability, tailorability, filtering, design in use

Adaptation Mechanism to Control Different Critiquing Rule Sets and Different Intervention Strategies



UDDE-Stack

Select Perspective:

Select the standard(s) you want to use when analyzing your design. This allows you to view your design from multiple perspectives. Click on the button to the left to enable the rule set. Click on Argumentation to list all the rules which belong to the selected standard. You can disable and enable individual critic rules within each standard from this overview. To change priority, disable the rule set, then enable it again.

Priority	Enable/Disable Set	Explanation
2	● USWEST Rule Set	Argumentation
0	○ UMUIF Rule Set	Argumentation
0	○ International Rule Set	Argumentation
1	● Consistent with: VM-Residential	Argumentation

Type of Application: Voice Mail

● Conceptual Unit Critiquing

Commercial Applications: Microsoft’s IntelliSense

- **technology started to appear in Office 97**

- **claims: the software “understands”**
  - the context of an end-user's actions
  - recognizes the user's intent
  - automatically produces the correct result

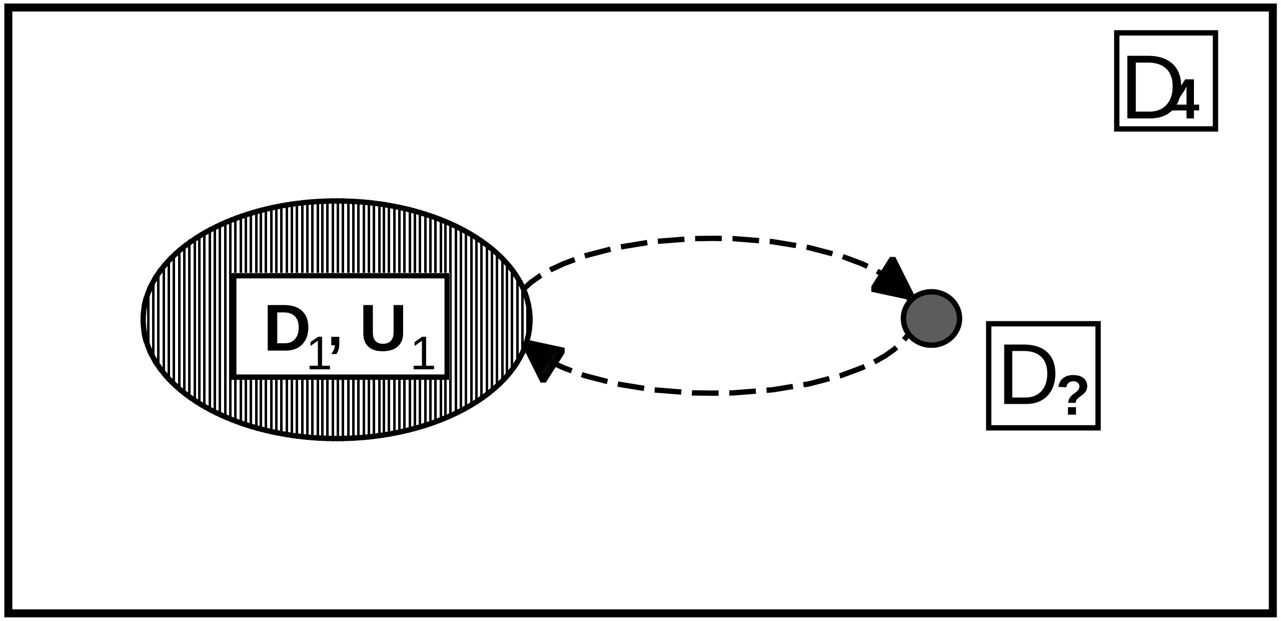
- **features**
  - routine task automation
    - ➢ background spelling and grammar checks
    - ➢ automatic formatting of one paragraph based on format of the previous paragraph
  - tasks are simplified through the offering of wizards (e.g., wizards for creating faxes or letters)
  - personalization of the software
    - ➢ allowing users to control how the office assistant behaves
    - ➢ allowing developers to program additional features
    - ➢ allowing users to create additional features (e.g., macros)

Entering Unknown Parts of D4 — Opportunity or Problem

- **issues:** a user hits the wrong keys (but the keystrokes get interpreted in D4); the system infers the "wrong" intentions from the users actions **— *"every wrong answer is the right answer to some other question"***

- **problem**: "smart" systems which guess wrong (e.g., in MS-Word: AutoCorrect, Tables, Bullets and Numbering, ……)

- **opportunity**: serendipity



The Road Ahead

- **the ultimate objective:** UM is done for the benefit of the users

- **create basic concepts** and terminology, theory, applications, substrates (UM shells; Kobsa et al), practice, assessment

- **integration / embedding** with help systems, tutoring systems, critiquing systems, high-functionality applications, ……

- what is **not** user modeling (e.g., analysis and exploitation of credit cards, shopping cards, …..)

- desk-top environment → web-based environment

Conclusions

- **contradiction:** some research domains, such as user modeling, reuse, exploiting the social (organizational memory, organizational learning, social creativity) are
  - in principle: very important and appealing themes
  - in practice: there is a lack of evidence for success

- **the best is often the enemy of the good**
  - UM support does not need to be “perfect”; it should be useful and usable by serving the users
  - maybe some of the “simpler” mechanisms are more promising (such as user profiles, user preferences, specification components, …..)?

- **UM as one important technique for**
  - human-centered design (user experience, comfort, control, privacy)
  - contextualization of information
  - integration of working and learning

The Long and Winding Road

The long and winding road
that leads to your door
will never disappear
I've seen that road before
It always leads me here
Lead me to you door

The wild and windy night
that the rain washed away
Has left a pool of tears
crying for the day
Why leave me standing here
let me know the way

Many times I've been alone
and many times I've cried
Any way you'll never know
the many ways I've tried

But still they lead me back
to the long winding road
You left me standing here
a long long time ago
Don't leave me waiting here
lead me to your door

But still they lead me back
to the long winding road
You left me standing here
a long long time ago
Don't keep me waiting here
lead me to your door
Yeah, yeah, yeah, yeah