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

Learning on Demand and High-Functionality Applications
—
User Modeling and HCI

Gerhard Fischer and Hal Eden
Spring Semester 2004
March 8, 2004

paper: Fischer, G. (2001) "User Modeling in Human-Computer Interaction," User Modeling and User-Adapted Interaction (UMUAI), Dordrecht, The Netherlands: Kluwer Academic Publishers, 11(2), pp. 65-86.
http://www.cs.colorado.edu/~gerhard/papers/umuai2000.pdf

Why Learning on Demand

- **information overload, high functionality systems, and the rapid change** of our world have created new problems and new challenges for learning and education

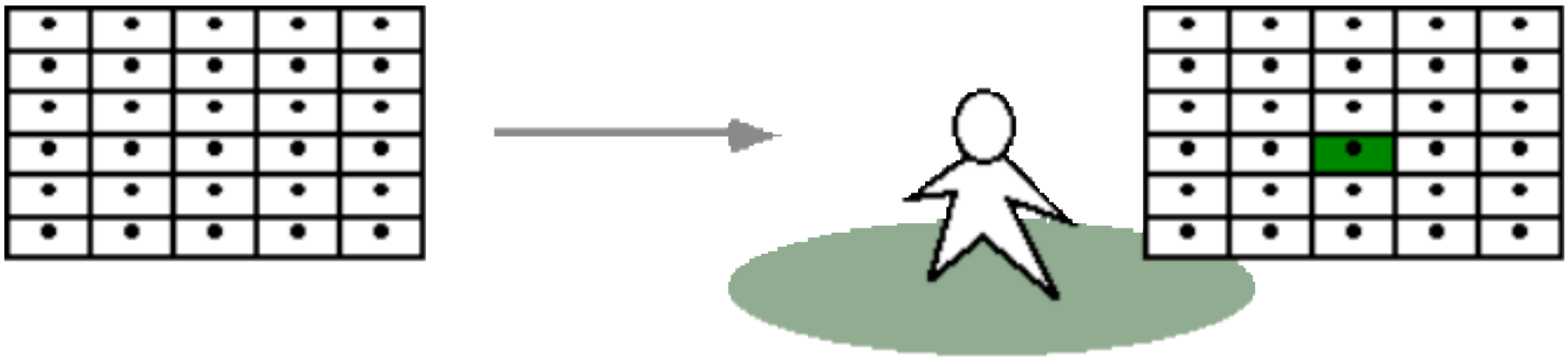
- humans settle on plateaus of **sub-optimal behavior**

- new instructional approaches are needed to circumvent the unsolvable problems of **coverage** and **obsolescence**

- education needs to be a distributed lifelong process, where **one learns the material as one needs it**

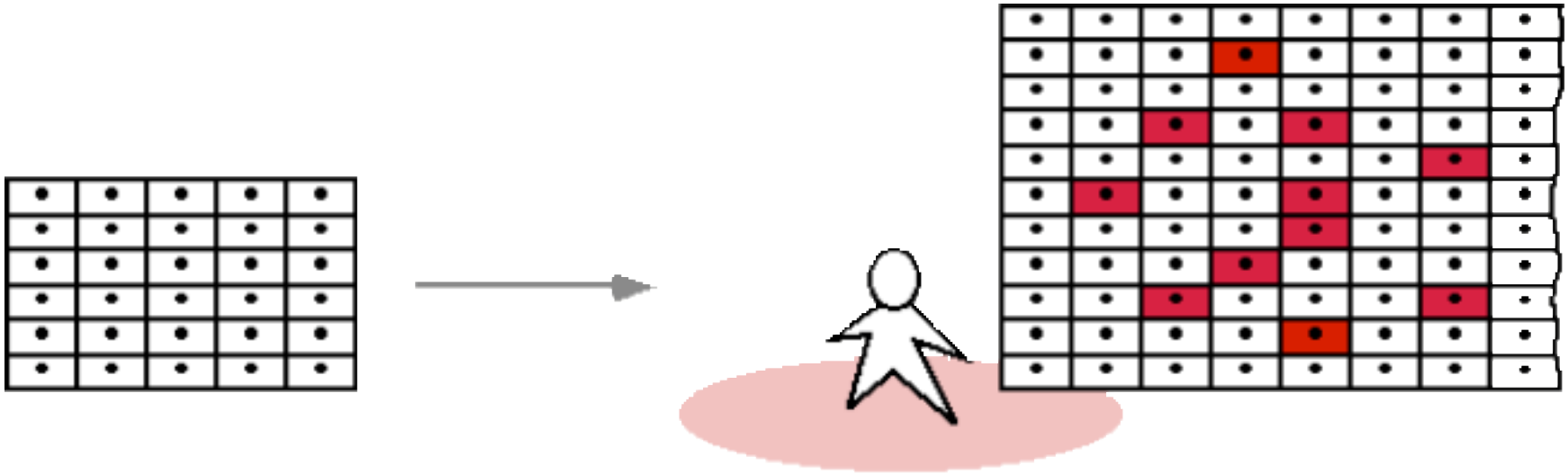
Old Model: Learn in School What is Needed in Life

school **life**



Problem with the Old Model in Today's World
—
Coverage and Obsolescence

school **life**



If Information is Plentiful — What is Scarce?

- we are rooted in a culture where time was plentiful, but information was scarce → we sometimes think we must **pay attention to information just because it is there**

- **mismatch between**
  - information **generation and duplication** capabilities → increased dramatically
  - human capacities for **absorbing** information → increased very little

- as long as information is in large supply and **human attention is in short supply**: we will "miss something" (a challenge for curricula development)

- create / use **knowledgeable filters** (humans or computational agents) to attend to the *important* information (personal and task relevant)

A Characterization of Learning on Demand

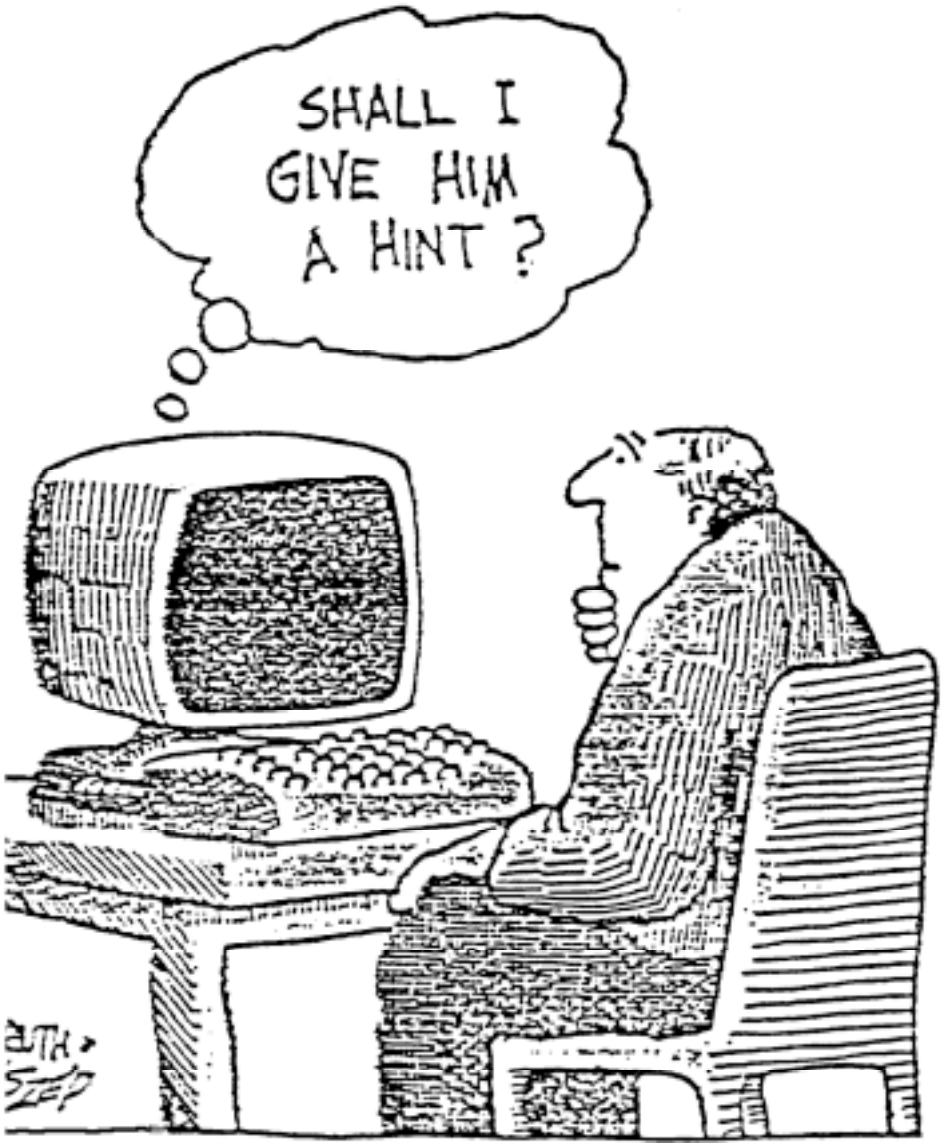
- contextualizes learning by allowing it to be **integrated into work** rather than relegating it to a separate phase
- makes new information **relevant to the task at hand**
- one **applicability condition** for new knowledge is known
- addresses the **discrepancy** between the amount of potentially relevant knowledge and the amount users can know and remember
- **"on demand"** sets computers **apart** from other media (such as books, TV, ...)
- **claim**: learning on demand can be supported by critiquing systems and design environments

Learning on Demand: Example Domains

- **suited:**
  - tools: VCR (OTR → Programming)
  - tools: High Functionality Applications
  - domains: kitchen design, network design, transportation system design, ...

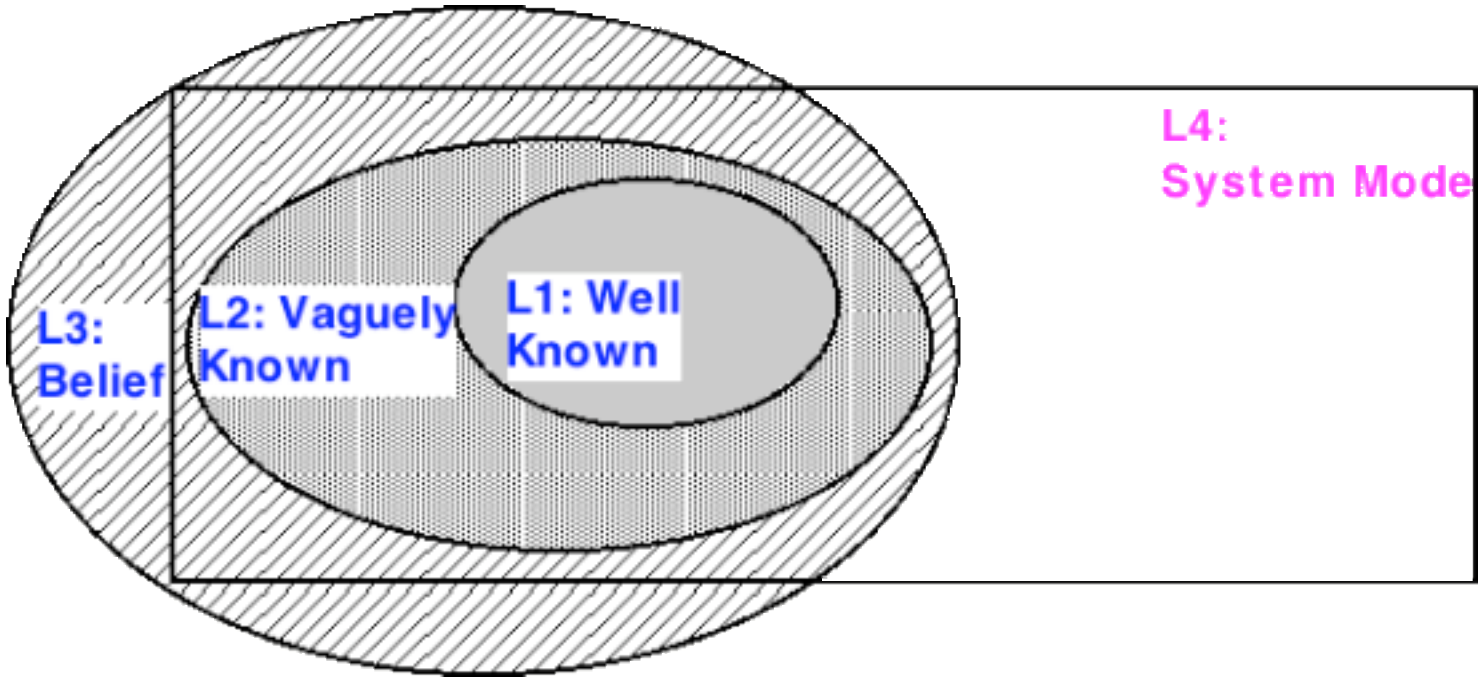
- **not suited:**
  - flying an airplane (but: do people learn on demand in a flight simulator???)
  - being dropped by a helicopter with skis on top of a high mountain

Information Delivery, Contextualization and Intrusiveness



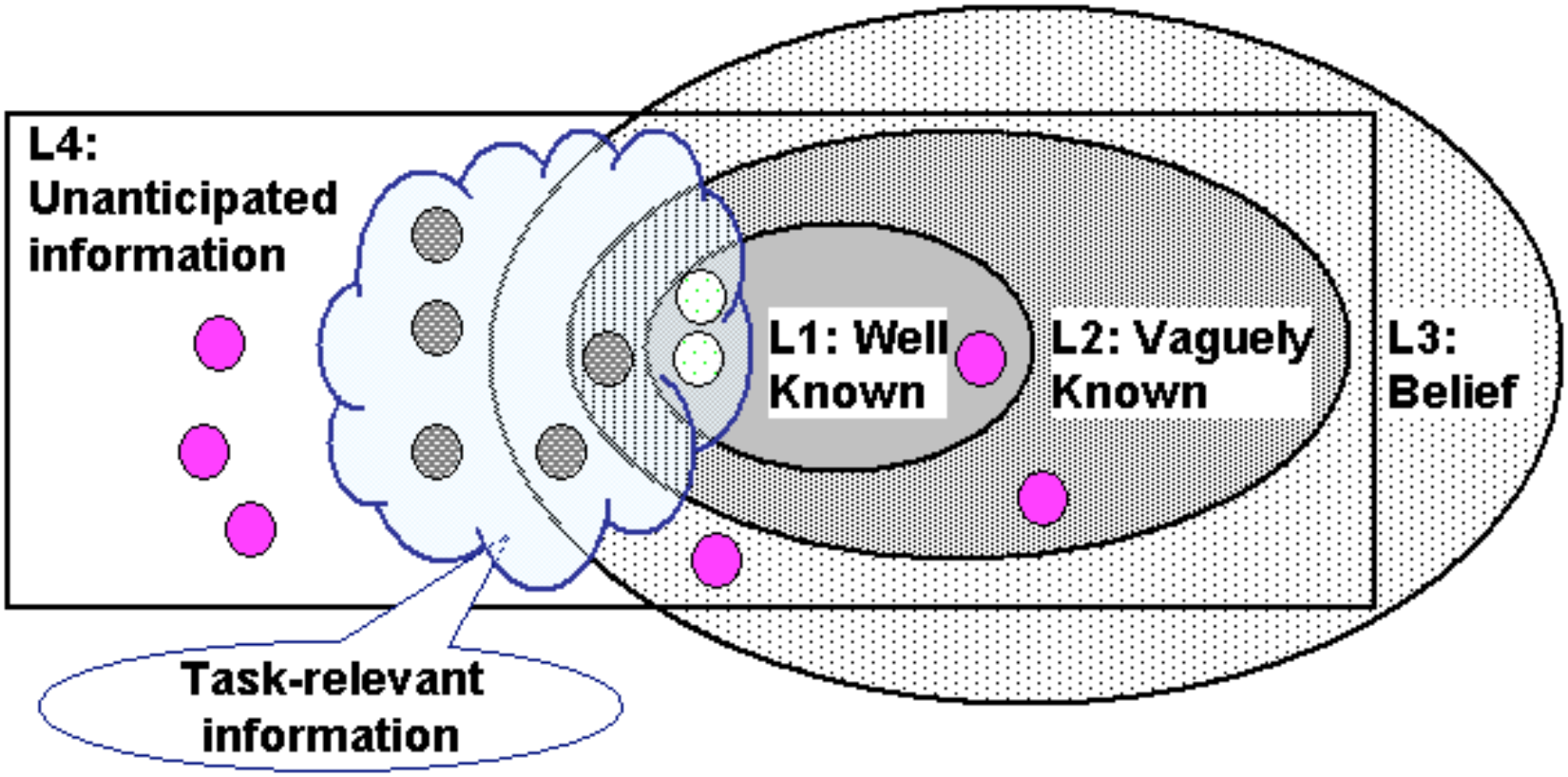
High-Functionality Applications (HFAs)

Mental Models Held byUser (L1,L2,L3) and System Model (L4)

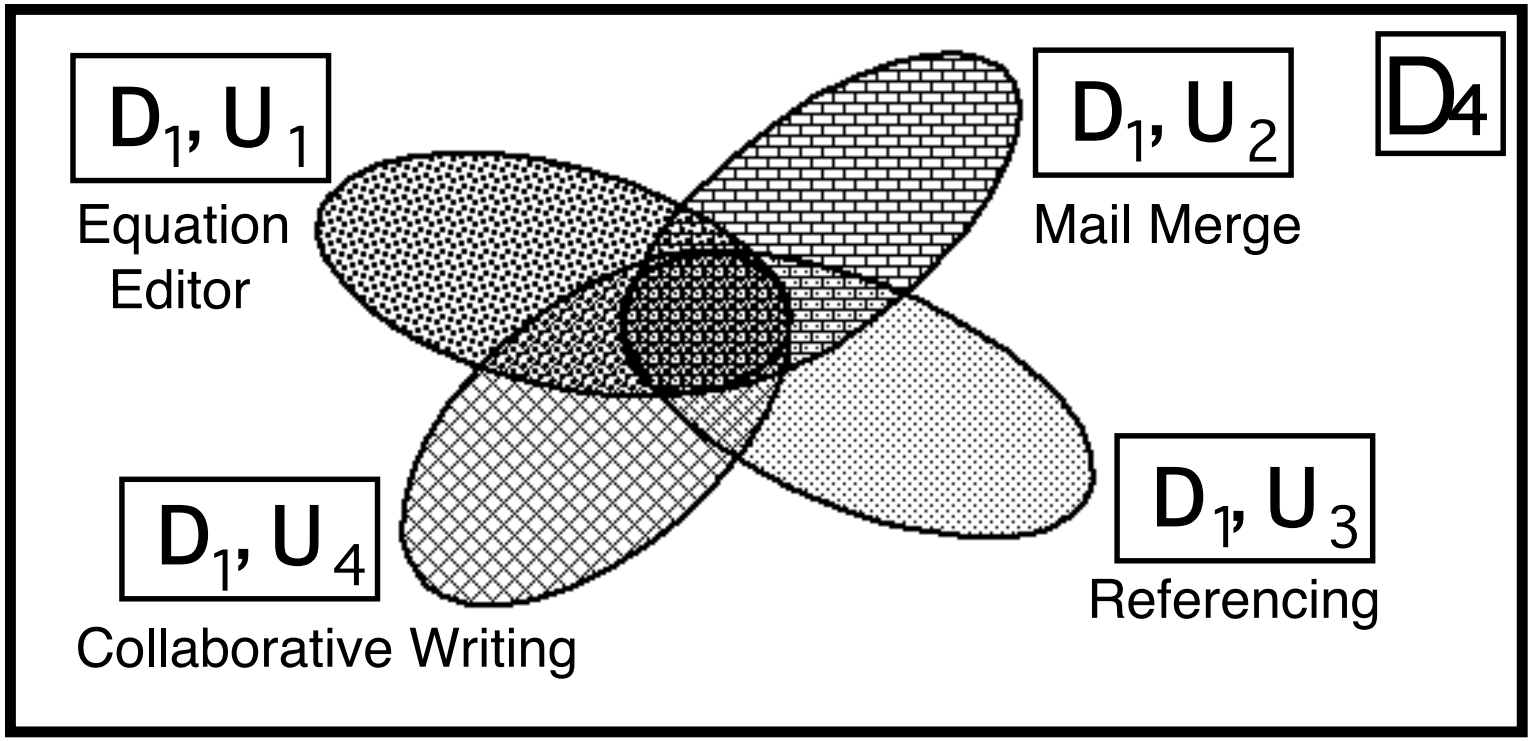


User Modeling and Identification of the Task at Hand in HFAs

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

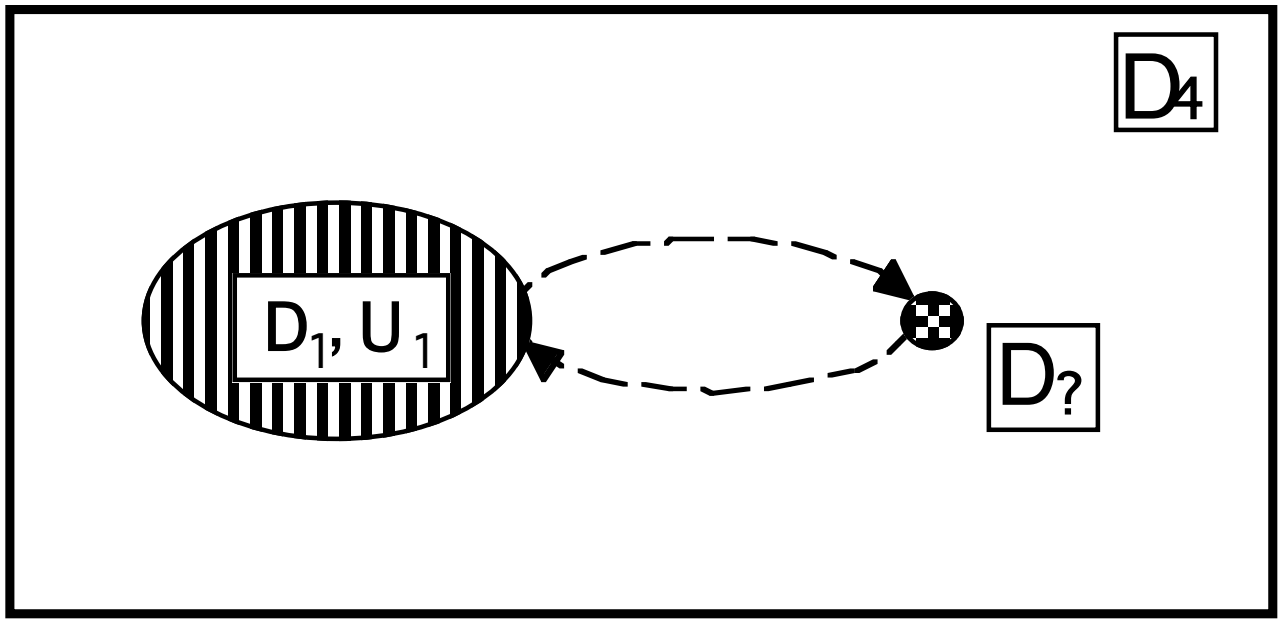


Expertise in HFAs is an Attribute of a Context, not of a Person



Entering Unknown Parts of D4 — Opportunity or Problem

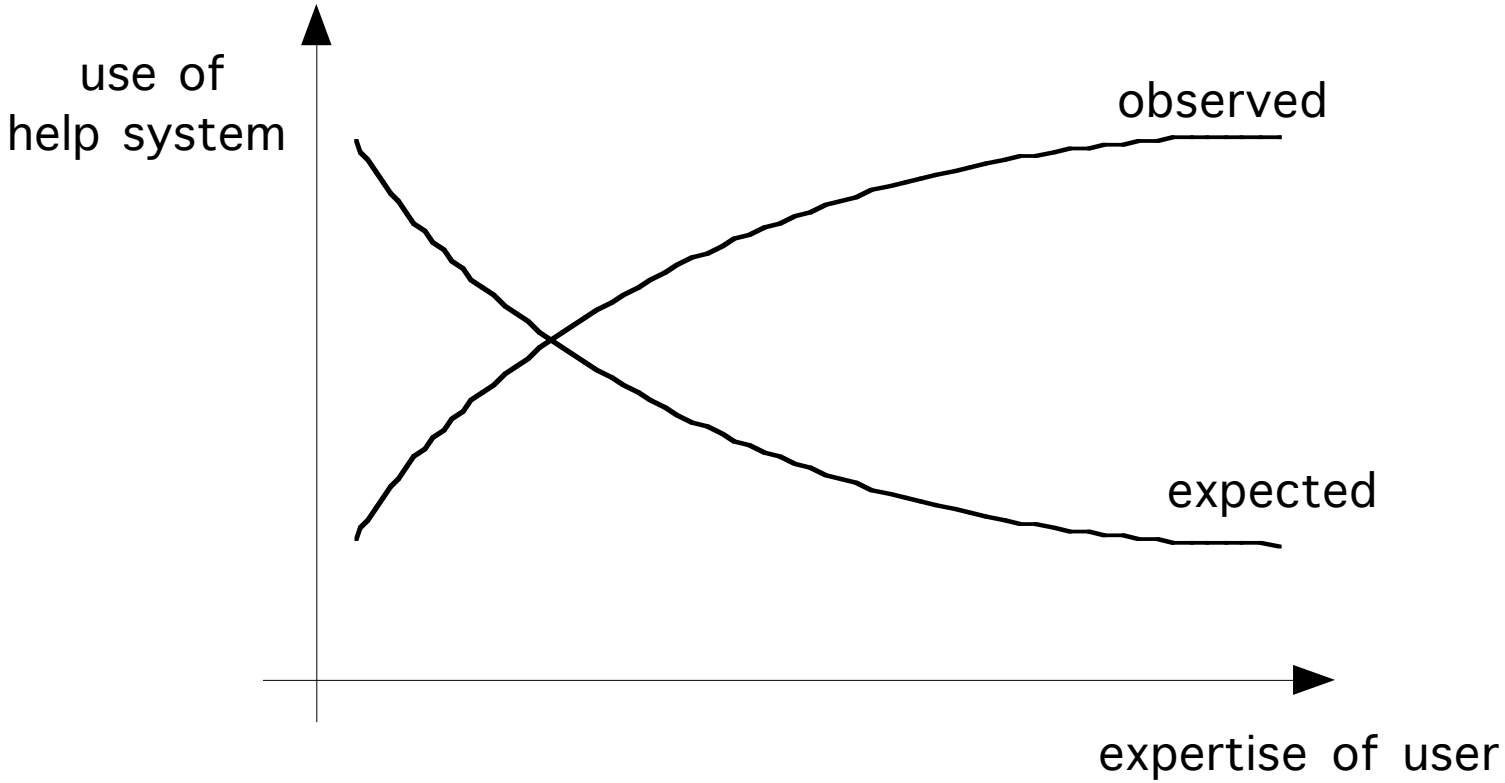
- **issues:** a user hits the wrong keys (but the keystrokes get interpreted in D4); the system infers the "wrong" intentions from the users actions — ***"every wrong answer is the right answer to some other question"***
- **problem**: "smart" systems which guess wrong (e.g., in MS-Word: AutoCorrect, Tables, Bullets and Numbering, ……)
- **opportunity**: serendipity



Problems with HFA

- users do not know about the **existence** of tools (D_4 □ □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

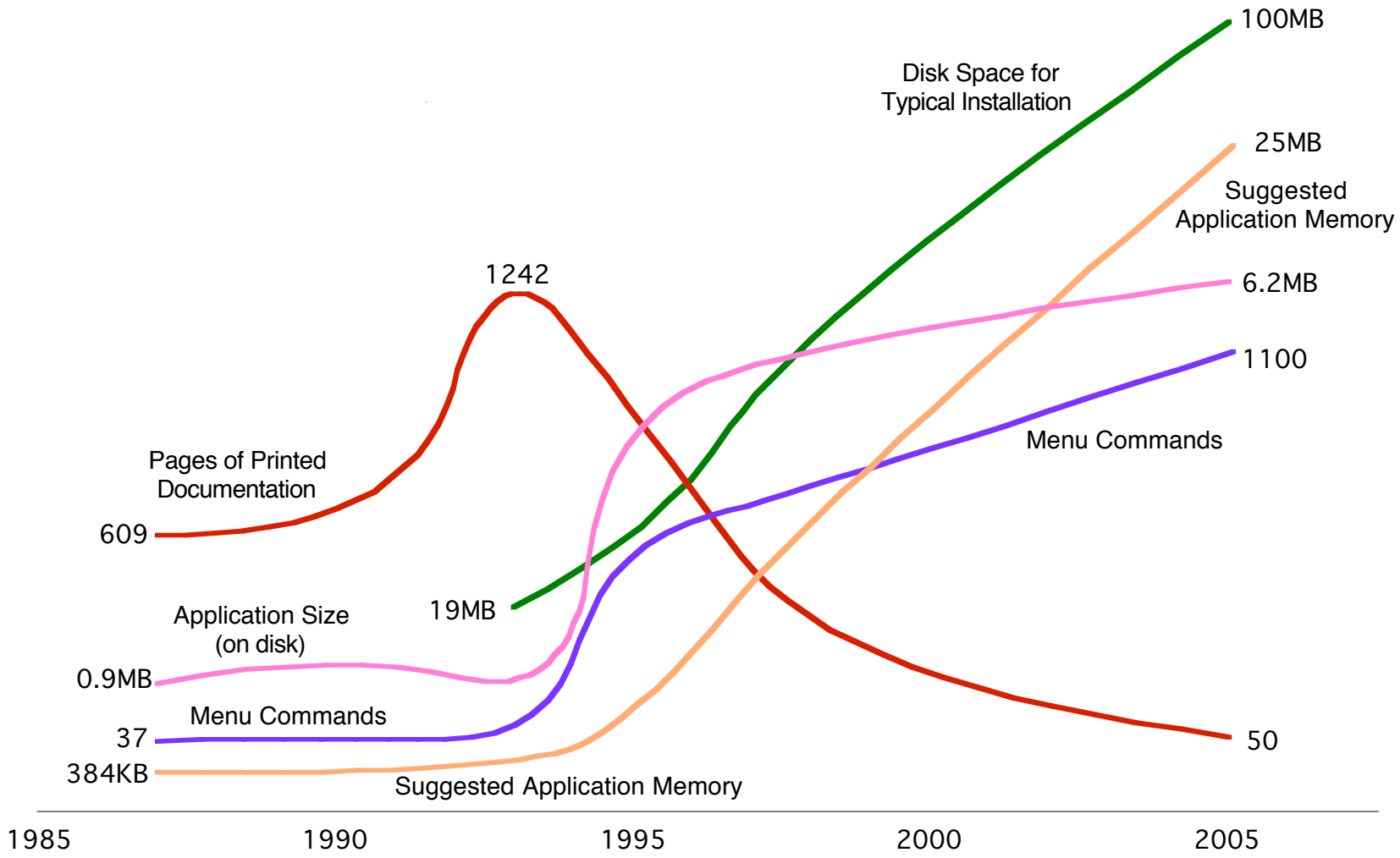
Usage of Sophisticated Help Systems



Problems with HFA: Microsoft's View and Objectives

- some "routine" tasks could be and needed to be **automated (→ Autocorrect)**
- 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 (→ **use on demand**)
- 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 (→ **personalization**)

A qualitative view of HFA trends (as observed in Microsoft Word)



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

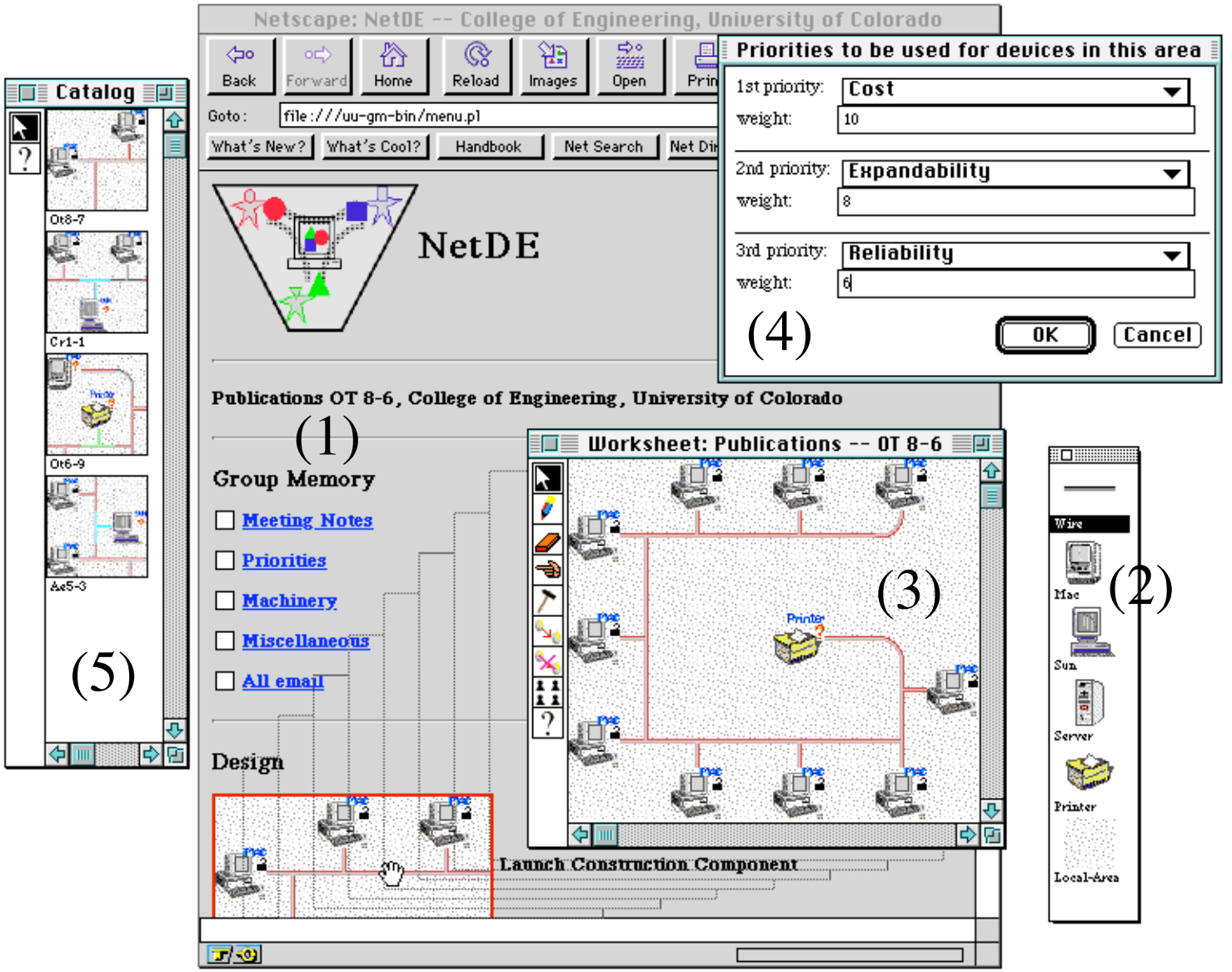
IntelliSense's Features

- **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)

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
- **support learning on demand**

A DODE for Computer Network Design



Example— Knowledge-Based Help Systems: Activist

- **Activist — an active help system** for an EMACS-like editor, deals with two different kinds of sub-optimal behavior:
  - the user does not know a complex command and uses “sub-optimal” commands to reach a goal (“sub-optimal”: 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**

- **for details see:** Fischer, G., Lemke, A. C., & Schwab, T. (1985) "Knowledge-Based Help Systems." In L. Borman & B. Curtis (Eds.), *Proceedings of CHI'85 Conference on Human Factors in Computing Systems,* ACM, New York, pp. 161-167.

Some Challenging Research Problems

- **identify user goals from low-level interactions**
  - active help systems
  - data detectors
- **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)

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 how to adapt)
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