



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**

Learning on Demand and High-Functionality Applications

**Gerhard Fischer and Leysia Palen
Spring Semester 1999**

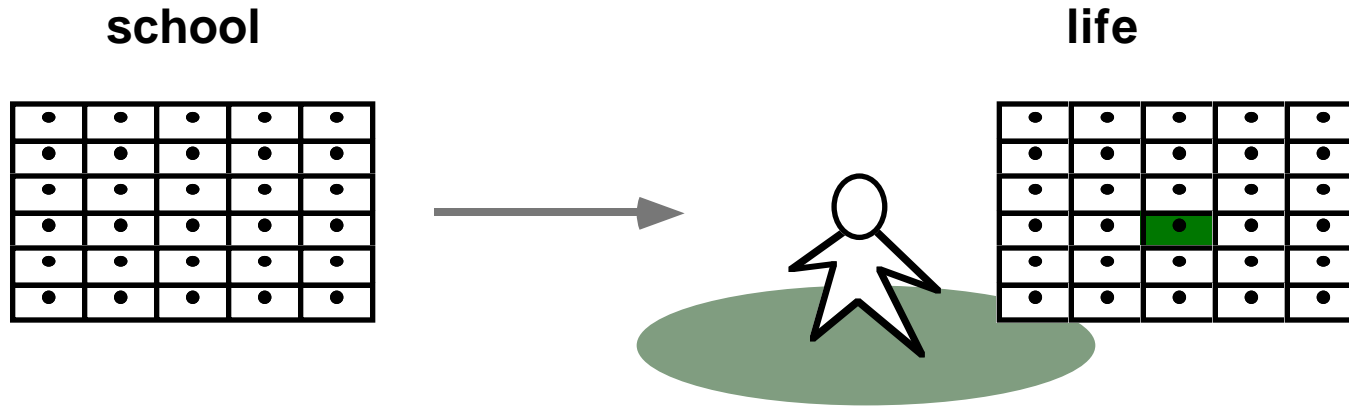
March 17, 1999

Why Learning on Demand

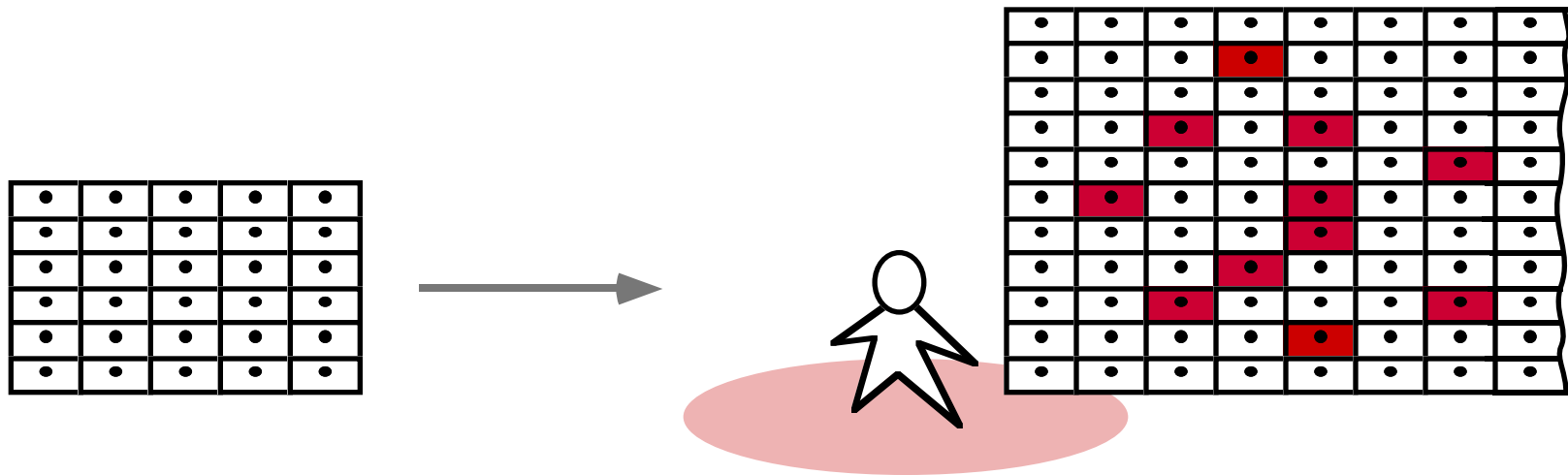
- **information overload, high functionality systems, and the rapid change** of our world have created new problems and new challenges for learning and education
- humans settle on plateaus of **suboptimal behavior**
- new instructional approaches are needed to circumvent the unsolvable problems of **coverage** and **obsolescence**
- education needs to be a distributed lifelong process, where **one learns the material as one needs it**

Coverage and Obsolescence

old model:



new model:



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), and
 - human capacities for absorbing information (has not increased at all)
- as long as information is in large supply and human attention is in short supply: we will “miss something”
- 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 (they often lack a strong domain-orientation; rarely contain domain-oriented knowledge; **but:** there are domain-oriented wizards in Word, e.g. for writing a letter)
 - domains: kitchen design, network design, transportation system design,
- **not suited:**
 - flying an airplane (but: we learn on demand in a flight simulator???)
 - being dropped by a helicopter with skis on top of a high mountain

Issues Associated with Learning on Demand

- demands originating from the user versus from the system
- how do we make the demand evident?
 - how are failures, inadequacies and breakdowns perceived ?
 - how does the situation “talk back”?
 - Rittel: “Buildings to not speak for themselves”
- how can “breakdowns” lead to opportunities for learning?
- how can we supply the necessary information?
- how can the demand be described? (only in the situation model?)
- why learning on demand? is *using on demand* not good enough?

How do Design Environments Support Learning on Demand?

- the artifact is embedded in the design environments and can be computationally analyzed
- critics increase the “back-talk” of the situation
- access paths to relevant information are provided
- demands can be articulated in the situation model
- presentation of information is based on relevance to the task at hand

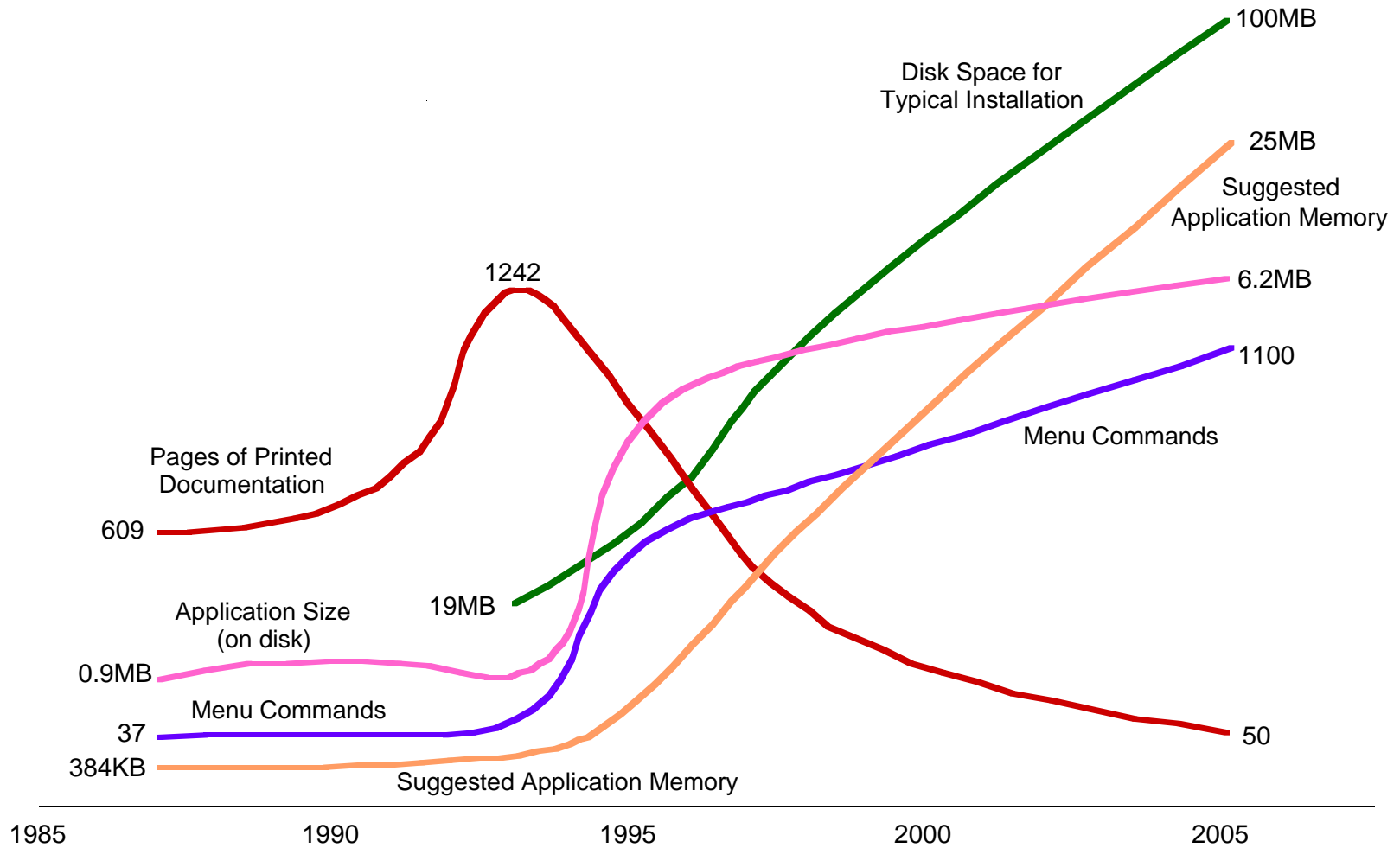
Learning on Demand and Critiquing Systems

- **claim:** as people take on more jobs that are more complex or more comprehensive, they need help accomplishing unfamiliar tasks that are part of an expanded job — e.g.: multi-media is a good example (charts, color,)
- **Kosslyn (in “Elements of Graph Design”, p 2):**
 - one reason for the abundance of bad graphs is the proliferation of low-cost microcomputers and “business graphics” packages, which often seduce the user into producing flashy, but muddled display
 - the ease of creating charts and graphs is a major selling point for personal computers, one rarely hears anything about the utility of the displays the machines produce
- **Travis (in “Effective Color Displays”):**
 - the standard IBM PC can now display 256 K colors and a Sun workstation can display 16.8 million — hardware is no longer a limiting factor to use color
 - *but:* when color is used inappropriately it can be very counter productive and few software designers have much experience with the use of color

High-Functionality Applications

- **examples:** MS-Word, MS-Office, UNIX, Canvas,
- **production paradox:** people are not always eager to learn new things, but they want to get their work done, i.e., getting the job done is the primary focus
- in HFA: there are no “experts” anymore

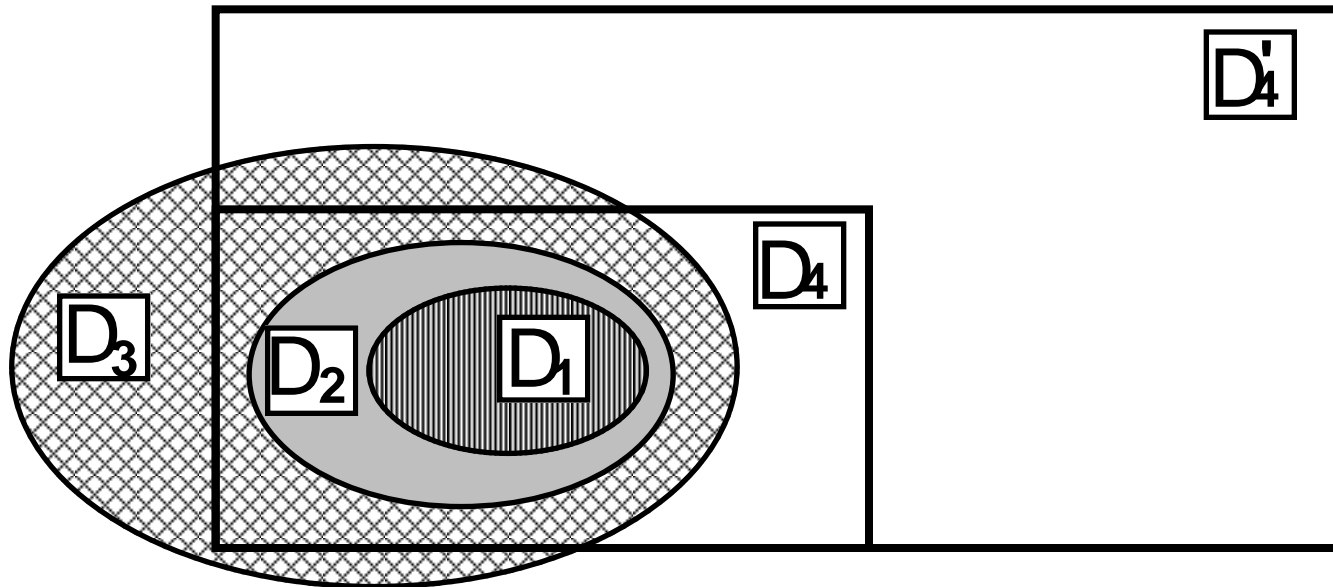
High Functionality Applications



Problems with High Functionality Applications

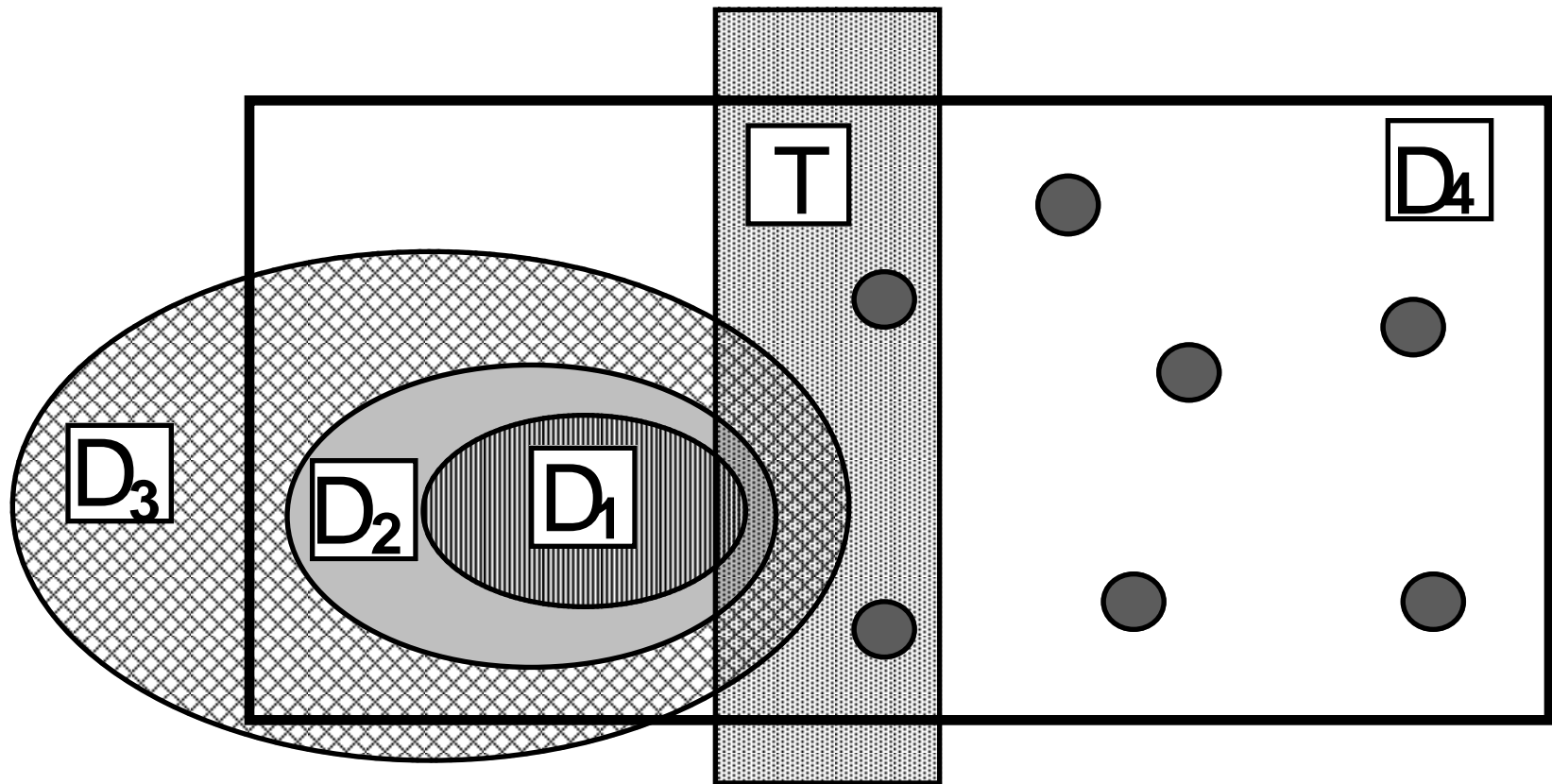
- users do not have **well-formed goals**
- users do not know about the **existence** of facilities provided by a system
- users do not know how to **access** the facilities
- users do not know **when** and **how** to use the facilities
- users do not understand the **results** that system facilities produce for them
- users cannot combine, adapt, and modify facilities according to their **specific needs**

Qualitative Analysis of Levels of Users' Knowledge

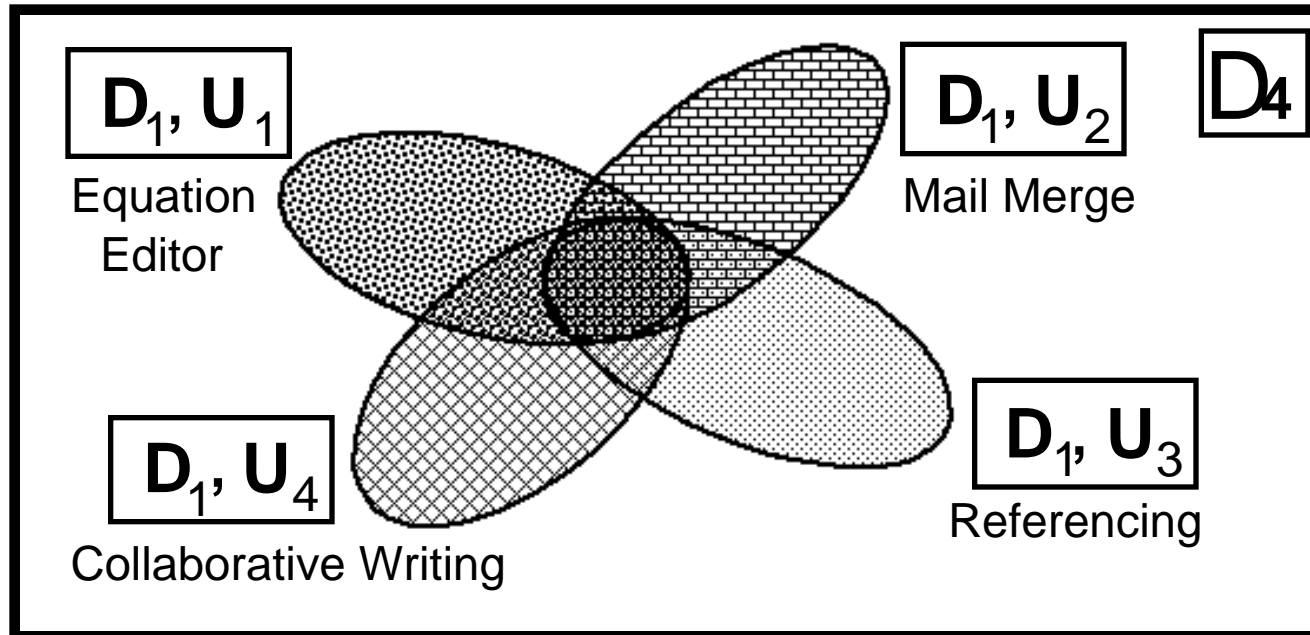


- D1: represents concepts well known
- D2: contains concepts known vaguely and used only occasionally
- D3: represents concepts users *believe* to exist in the system
- D4: represents the actual information space of a system
- D4': usage situation for future HFAs

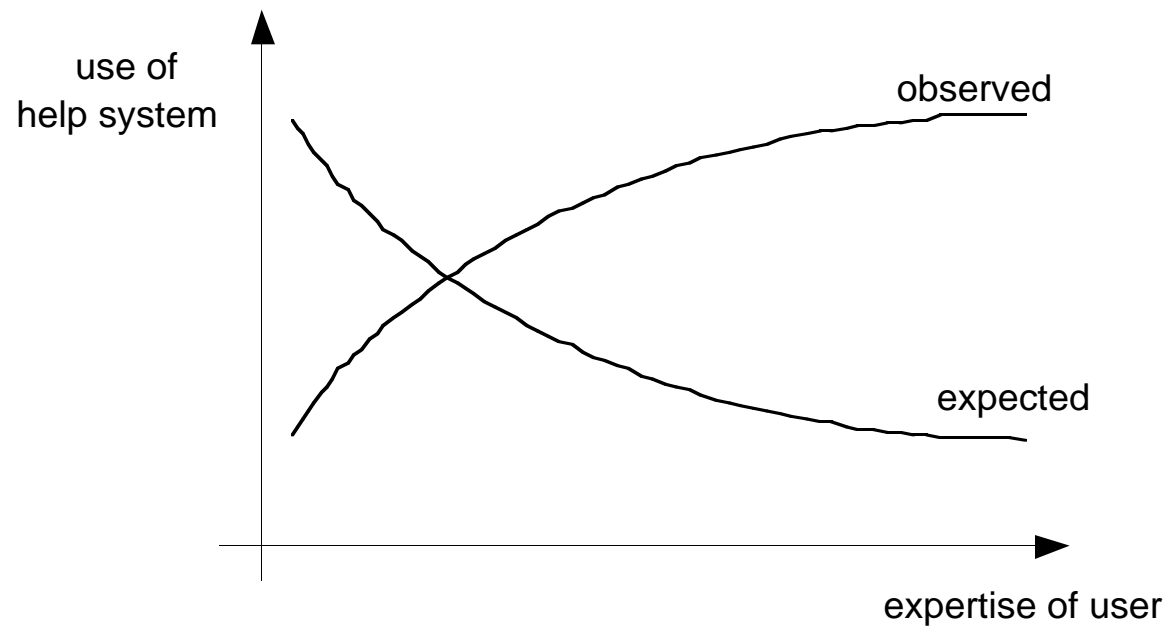
Functionality and its Relevancy to the Task at Hand



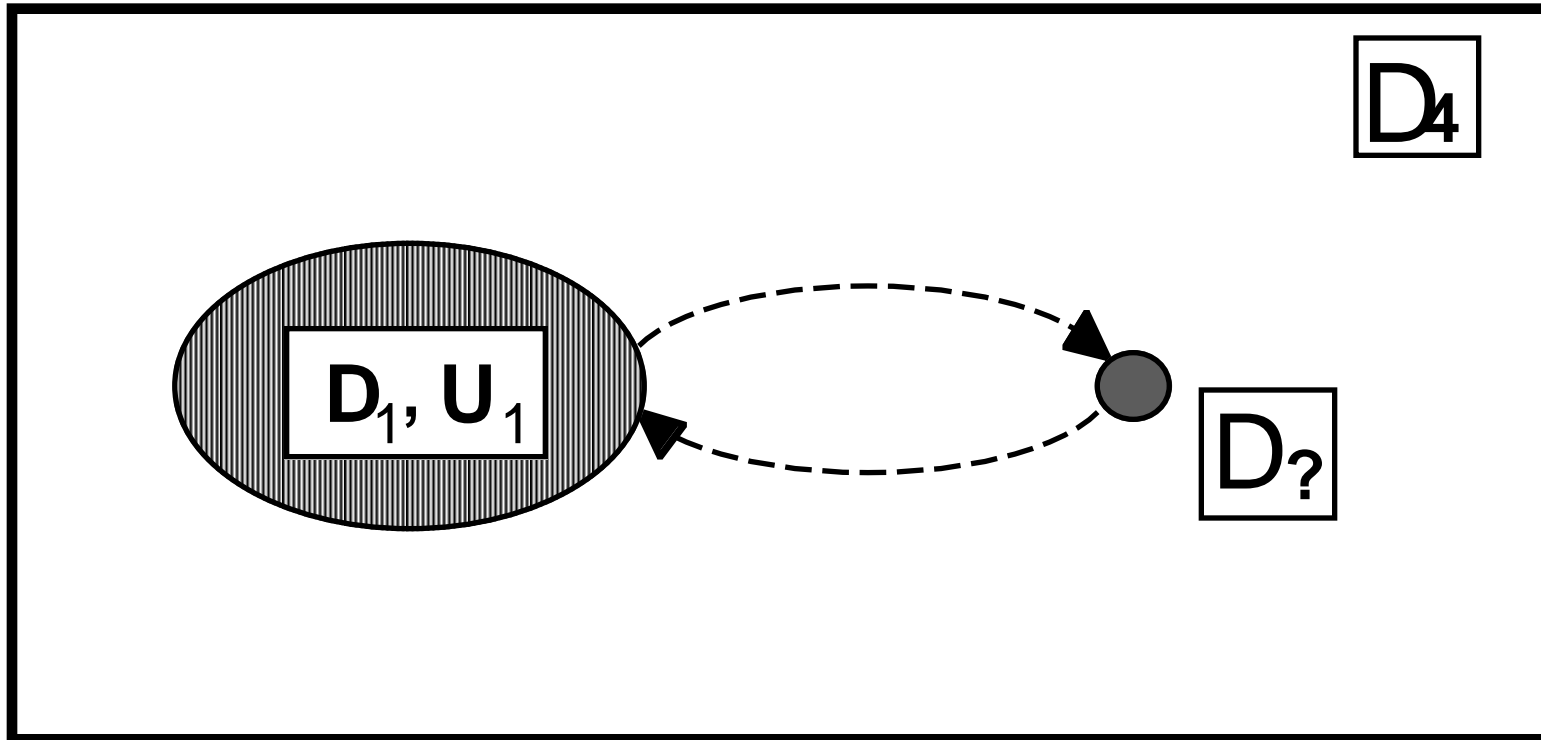
“Experts” and Expertise in HFA



Usage of Sophisticated Help Systems



Entering Unknown Parts of D4 — Opportunity or Problem?



Is End-User Modifiability the Answer to HFAs?

- HFA is that they contain too much unused functionality — at least in the abstract
- there is often not enough functionality for specific problems
- claim: no matter how much designers at design time try to anticipate the needs of users at use time, the effort will fall short because it is impossible to anticipate all future needs
- support in HFAs for end-user modifiability and end-user programming
 - extend the spelling dictionaries
 - Microsoft Word:
 - * tailor the system by setting different parameters as personal preferences
 - * write macros to create new operations;
 - * create programs in WordBasic to extend the functionality of the system

Commercial Developments

- newer software: has an increasing number of support mechanism that offer assistants, user tips and that diagnose user problems
- the “Tip of the Day” modeled after the “Did You Know (DYK)” system ---> *how make them more context-sensitive?*
- passive help (asking the Office Assistant) ---> *the vocabulary problem (drag and drop: how would I ask to copy instead of moving?)*
- active help (e.g., the flagging of misspelled words, the use of questionable grammar constructs) ----> *to which extent do their active help systems analyze the user’s work context?*
- wizards help users with doing standard operations (e.g., going from data in a spreadsheet to a visualization of the data
 - question: how successful are end-user modification feature in reality? ---> power-user, local developers

Empirical Study-1

- Sullivan, J. F. (1995) A Proactive Computational Approach for Learning While Working, Ph.D. Dissertation, Department of Computer Science, University of Colorado at Boulder, Boulder, CO.)
- user groups: novice, intermediate, power users
- **results**
 - no group was interested in attending training classes, reading documentation or using on-line help
 - users preferred to learn applications while working on tasks
 - becoming proficient requires several years' experience

Empirical Study-2

- Bob Gatewood: “Novice and Expert Learning in High-Functionality Applications” (course project, Fall 1996)
- user groups: novice (3) , intermediate (7) , power users (15)
- **results:**
 - number of years of experience and skill level are unrelated
 - advanced users have experience with more applications
 - all users rely of the help of others in learning HFAs, but experts and non-experts ask different questions
 - * beginners: ask a person for help even for simple tasks
 - * advanced: more use of help systems and documentation, FAQs,
 - advanced users employed a greater variety of learning strategies: structured experimentation, studied the work of others, ...
 - out-of-context information is not useful (“Tip of the Day”)
 - HFA’s functionality only matters relative to what the user is trying to accomplish
 - experienced users expected more functionality to exist which was not there (D3 & not D4)