

T  
A  
S  
K  
T  
R  
A  
C  
E  
R

---

# TaskTracer: Toward a Task-Oriented Desktop Interface

---

Thomas G. Dietterich  
Jon Herlocker  
Simone Stumpf  
TaskTracer Team

School of Electrical Engineering and Computer Science  
Oregon State University  
Corvallis, Oregon 97331  
<http://www.eecs.orst.edu/~tgd>

# An example scenario...

Warning: The scenario you are about to see has been only partially implemented.

# TASK TRACER

1. Jane has to write a grant proposal to the National Science Foundation (NSF)

*“I’ve done this before! How did I do it last time?”*



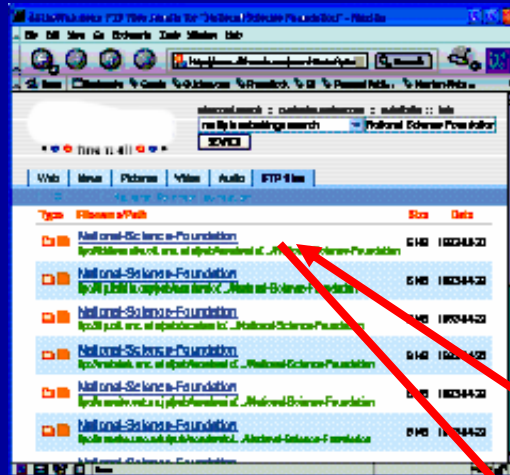
2. Jane describes her current task to TaskTracer  
*“Write NSF proposal”*

3. TaskTracer returns a list of past tasks that are relevant to the NSF.

*“Look here! – a record of the last time I wrote an NSF grant proposal. I’ll select that”*



# TASK TRACKER

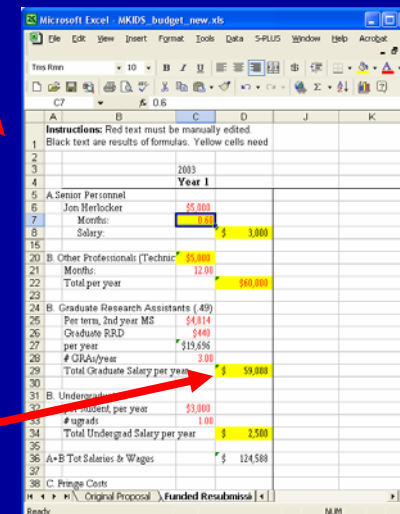


4. A list of resources Jane used while writing her last grant is displayed: files, web pages, email addr, phone numbers.

*“Aha! I’ll start with the budget I used last time as a template.”*

5. The budget spreadsheet is opened. The cells that were edited in the previously selected task are highlighted.

*“Looks like I need to find what the graduate pay rate will be in 2006-2007. It always changes”*



# TASKTRACER

Recommended Stipend Guidelines 2003.PDF (applications/pdf Object) - Mozilla

http://oregonstate.edu/dep/gnd\_school/graduate\_funding/0303/

GRADUATE ASSISTANTS - 9-MONTH APPOINTMENTS

	20 FTE	40 FTE	
Minimum (9-month) Annual Stipend*	\$5,137	\$12,585	
Suggested Minimum Levels at 30 FTE			
Level	FTE	Annual Stipend*	Monthly Stipend
GRA I (beginning graduate)	.30	\$7,705	\$ 856
GRA II (one year experience)	.30	\$8,090	\$ 899
GRA III (MS or equivalent)	.30	\$8,485	\$ 944
GRA IV (oral prof. completed)	.30	\$8,820	\$ 981

\* \$15,084 9-month minimum annual rate.  
 \*Stipend levels vary among programs. Although the minimum stipend should be specified, we do not offer higher stipends than those shown. Stipends are computed here at 30 FTE for illustrative purposes only, and may vary for other FTE levels.

GRADUATE ASSISTANTS - 12-MONTH APPOINTMENTS

	20 FTE	40 FTE	
Minimum (12-month) Annual Stipend*	\$6,282	\$15,391	
Suggested Minimum Levels at 30 FTE			
Level	FTE	Annual Stipend*	Monthly Stipend
GRA I (beginning graduate)	.30	\$ 8,223	\$ 785
GRA II (one year experience)	.30	\$ 8,804	\$ 825
GRA III (MS or equivalent)	.30	\$10,389	\$ 866
GRA IV (oral prof. completed)	.30	\$10,908	\$ 909

14116-Web-Home (77) This is the file for "National Science Foundation" - Firefox

ns:search :: search :: info  
run by: ns:search :: search :: info

SEARCH

Type	Filename/Path	Size	Date
File	Mail and Science Foundation	6140	10/28/02
File	Mail and Science Foundation	6140	10/28/02
File	Mail and Science Foundation	6140	10/28/02
File	Mail and Science Foundation	6140	10/28/02
File	Mail and Science Foundation	6140	10/28/02
File	Mail and Science Foundation	6140	10/28/02
File	Mail and Science Foundation	6140	10/28/02

6. TaskTracer remembers the previous cut/paste of this information. The spreadsheet cells have right-click option: "Show source URL..."

"Yes"

7. After editing the budget, Jane opens up the project summary from the last proposal and starts editing it. "Need to emphasize the artificial intelligence aspect of my research!"

MS-DOS original spreadsheet with Tom's name.doc - Microsoft Word

File Edit Format View Window Help

MSKID: Capturing, Reusing, and Leveraging Knowledge from Human Task Processes

Jonathan L. Herlocker Principal Investigator Department of Computer Science Oregon State University	Thomas G. Dietterich Co-Principal Investigator Department of Computer Science Oregon State University
--	--

1 Challenges Addressed

In this proposal, we address two major issues faced by the intelligence community and other knowledge-intensive domains: (1) the complexity of knowledge processing and (2) the hidden nature of knowledge processing. One of the goals of the MSKID program is to develop techniques to assess the productivity of knowledge processing, to organize with complex and dynamic processes. But before we can measure the productivity of knowledge processing, we must first have qualitative measurements of the inputs and outputs of each processing task. Since knowledge processing is not limited to a single domain, we must understand the complexity of these tasks. In this proposal, we will address and prototype new approaches for capturing detailed profiles of knowledge processing activity, both at the conceptual level and at the physical level. These tasks will include: (1) the development of a set of knowledge processing tasks, which will be used to apply a variety of approaches to capturing the data that will be used to apply processing the results to the process, which will be used to understand the hidden nature of the knowledge processing. We will develop new methods that allow both individuals and managers to leverage the structured results of previously used knowledge processing.

Hidden Workflows. Another challenge presented by the MSKID program is capturing hidden workflows. In large organizations, the processes followed when transferring a knowledge product (plans, proposals, studies, documents, prototypes, etc.) are often poorly documented or undocumented. In many cases, the knowledge processing tasks change so frequently with many of the tasks being repeated or reworked. In most cases, it is impossible to capture individuals by hand from low-level, unstructured procedures for activities which are not even open to nearly the same manner. We aim to address this problem through analysis of the data that we collect. Since every observed knowledge processing event is associated with a user task description, we learn the high-level task description, an integral sequence of associated knowledge processing events. Over this simple representational structure, we can also visualize, by drawing connections between different user task profiles, and by identifying recurring patterns between two or more users' task profiles.

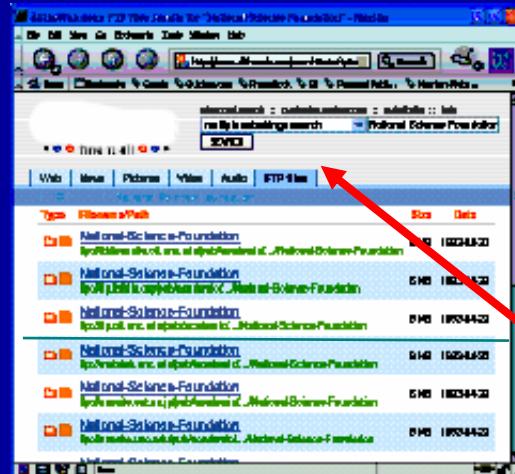
8. Interruption! Spouse calls and says he's outside. "Need to log out and head home..."

# TASKTRACER

9. Next day after she logs in.  
“What was I working on???”

10. On request, TaskTracer displays  
uncompleted tasks currently in  
progress.

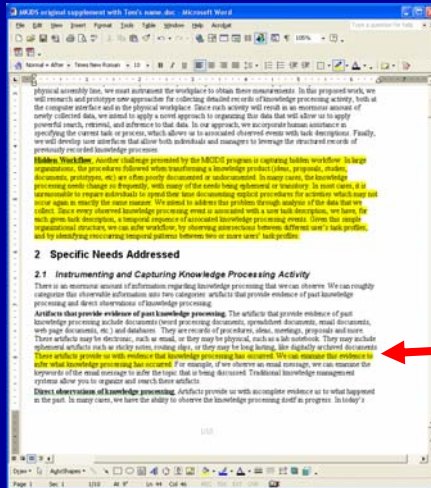
“Right! I was writing that NSF grant proposal.  
Let me select that task.”



11. TaskTracer shows documents  
touched in the previous day plus all  
the documents from the previous  
related task.

“The last thing I was doing was editing the  
project summary document – I’ll continue  
with that”

# TASKTRACER



12. TaskTracer opens the Word document and highlights the text that she most recently edited.

*"When I was interrupted, I was busy editing this paragraph here. I will continue from there"*

13. The phone rings – TaskTracer uses caller id to identify the caller, locates tasks associated with that caller, and displays a list of tasks that caller is associated with

*"Chris works with me on the conference committee, so it must be about that"*



# TASK TRACER



14. TaskTracer pulls up a record of the task and lists past emails and phone calls to and from that caller. "I can quickly scan the recent email messages to and from Chris to recall what we last discussed"

15. ... A month later, Joe, a new hire, asks Jane how the grant proposal process works at OSU "It's long and convoluted, Joe. Let me just send you the TaskTracer record from my most recent grant"



This screenshot shows a Microsoft Excel spreadsheet titled 'MKIDS\_budget\_new.xls'. It contains a budget breakdown with columns for 'A', 'B', 'C', 'D', 'J', and 'K'. The data includes various categories like 'Senior Personnel', 'Graduate Research Assistants', and 'Undergraduates' with associated costs and salaries.

Category	Item	Amount
A. Senior Personnel	Jon Herlocker	\$5,000
B. Graduate Research Assistants	Per term, 2nd year MS	\$4,814
C. Undergraduates	Per student, per year	\$3,000
D. Total	Total Graduate Salary per year	\$59,000
J. Total	Total Undergrad Salary per year	\$2,500
K. Total	A+B Tot Salaries & Wages	\$124,588



# The Windows™ Model

- ◆ Repeat
  - user randomly chooses program to run
    - user randomly chooses file to access
      - ◆ possibly one of the 4 most recent in that application
    - user randomly chooses where to save the file
- ◆ This model does not capture or exploit the coherent structure of the user's desktop activities

# The TaskTracer Model

## Repeat

- ◆ User chooses from a “working set” of ongoing “tasks” or “activities” (or possibly a new activity)
  - User chooses a resource associated with that activity
  - User works on that resource and then “delivers” it (print, fax, email, upload, etc.)
  - User communicates with other people involved in the activity
  - User attends meetings associated with the activity
- ◆ Activities tend to be interrupted by other activities
  - phone calls, IMs, scheduled appointments, trips (even eating and sleeping!), emergencies, opportunities
- ◆ New tasks are often similar to old tasks
  - Use old files as “templates” via copy-and-edit
  - Communicate with the same or similar people
  - Require similar amount of time and effort

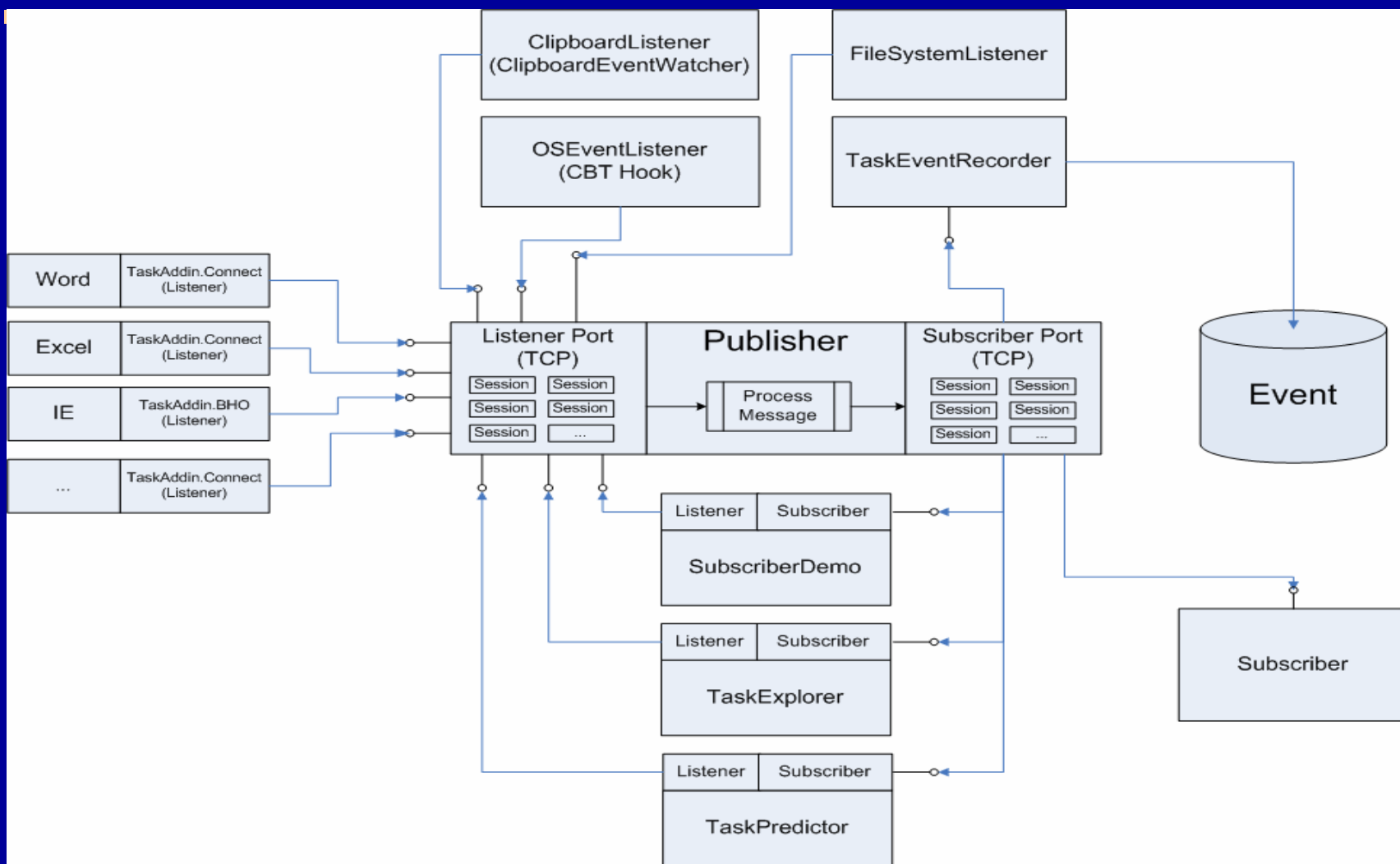
# The Activity Hypothesis

- ◆ Activities are the key abstraction for
  - understanding user behavior
  - organizing the resources needed by the user
  - helping the user
- ◆ In TaskTracer, “activity” = “task”

# Requirements for TaskTracer

1. Instrument desktop applications to capture events (accesses to files, folders, web pages, calendar; email, phone, and chat traffic)
2. Define/discover the user's "tasks"
3. Associate events/resources with tasks
4. Build/modify interfaces to provide easy access to relevant resources and events

# Instrumenting the Desktop: Publisher-Subscriber Architecture

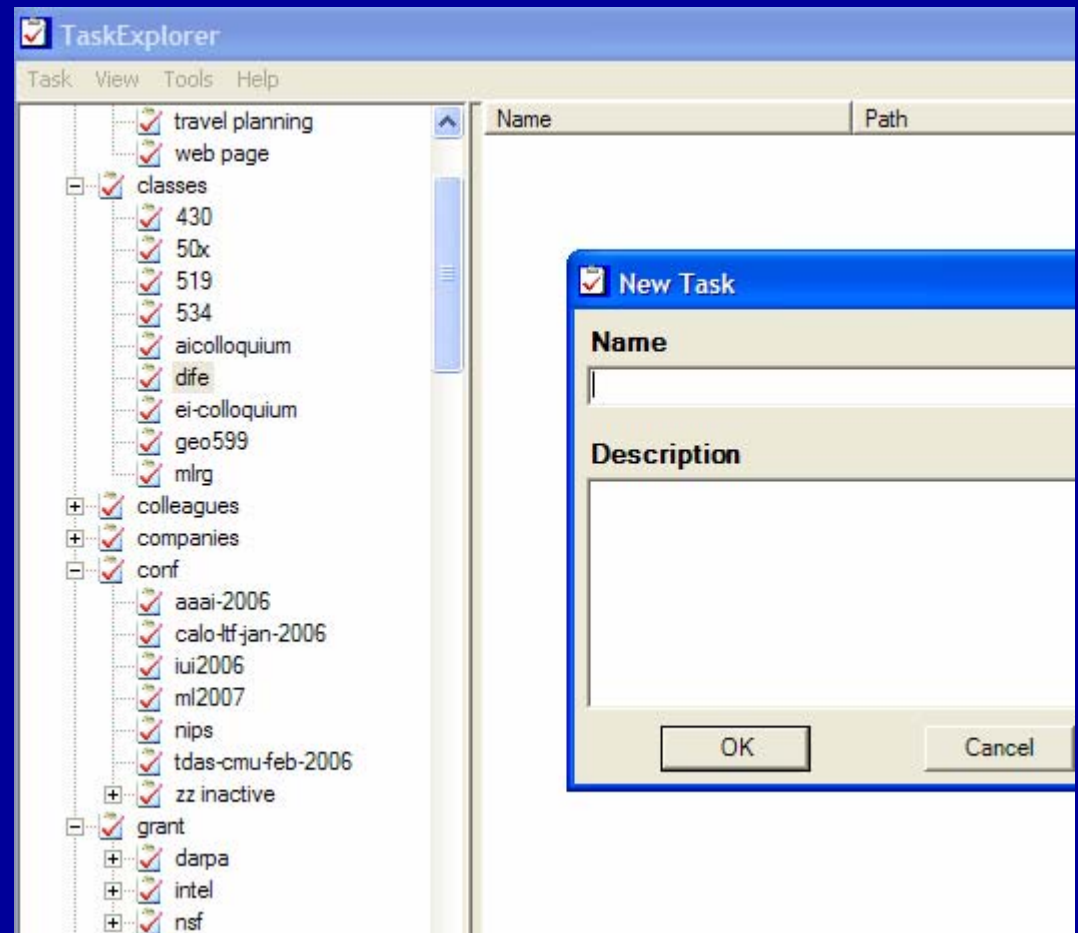


# 1. Instrument Desktop

- ◆ **Applications:**
  - Word, Excel, PowerPoint, Outlook, Internet Explorer, Windows Explorer, GSView, Acrobat, Visual Studio
- ◆ **Application Events:**
  - Documents: New, Change, Open, Print, Save, Save As, Close
  - Email: Open, Close, Send, Reply, Forward, Attach, Save Attachment, Open Attachment, Incoming Email
  - Web pages: Open, Navigate, Download File
- ◆ **OS Events:**
  - File create/delete/rename,
  - Window: Creation, Destroy, Focus
  - Copy/Paste
  - Suspend/Resume/Idle

## 2. Define/Discover User Tasks

- ◆ TaskExplorer application allows user to define a hierarchy of tasks
  - A task is just a name



# Other Ways of Discovering Tasks

- ◆ Cluster analysis of emails
  - “social” network of email correspondence
  - files attached to email messages
- ◆ Topic analysis of file contents, email contents, web page contents
- ◆ Files stored in same folder
- ◆ Cluster analysis of desktop activity (files, web pages, email messages co-occurring in time)

See: Huang, D. Govindaraju, T. M. Mitchell, V. R. Carvalho, W. W. Cohen (2004)  
[Inferring Ongoing Activities of Workstation Users by Clustering Email](#)



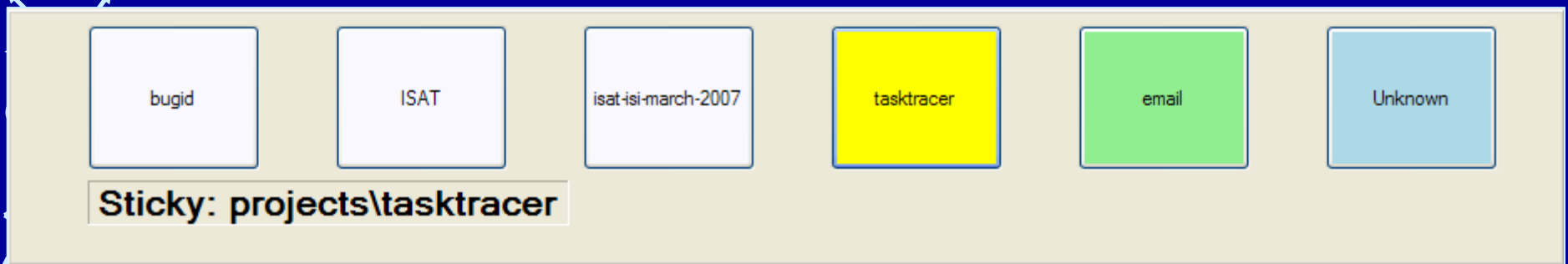
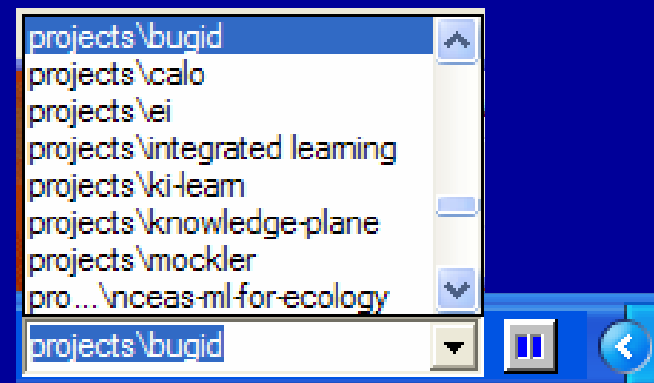
### 3. Associate events/resources with tasks

- ◆ Simplifying Assumption: The user is working on only one task at a time
- ◆ Require user to tell us what task they are working on
- ◆ Associate all items with the declared task (c.f. UMEA, 2003)

TRACER

# Declaring the Current Task

- ◆ Drop down menu in the task bar
- ◆ Control-backquote Quick Switch
- ◆ Creates a “TaskBegin” event that is sent to the Publisher



## Problem: Users Forget to Declare the Current Task

- ◆ Solution: Apply machine learning methods to predict the current task
- ◆ Two Predictors:
  - Email Predictor
    - Predict the task associated with an incoming email message
  - Task Predictor
    - Predict the task associated with the current window and document

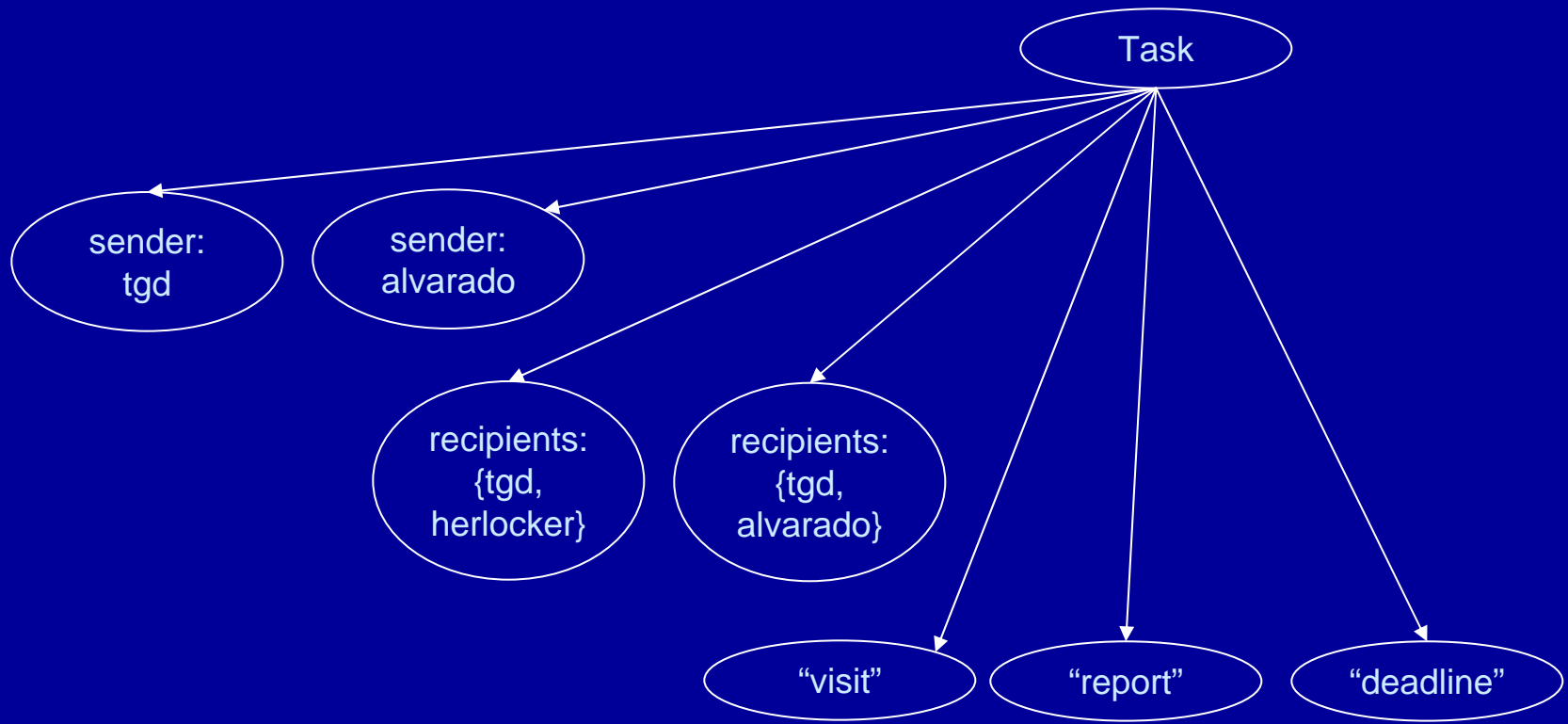
# Machine Learning Challenges

- ◆ Set of tasks is changing
- ◆ Distribution of task documents changes within a task over time
- ◆ Real-time online learning and prediction
- ◆ Must achieve very high accuracy to be acceptable

# Email Predictor

- ◆ Input features:
  - sender
  - union of From:, To:, CC:, and BCC: fields
  - words in subject
- ◆ Feature selection via mutual information
- ◆ Prediction based on probability threshold

# Naïve Bayes Classifier



$$P(\text{task} | X) = \alpha P(x_1|\text{task}) P(x_2|\text{task}) \dots P(x_n|\text{task}) P(\text{task})$$

# Hybrid Learning Algorithm

- ◆ Train Naïve Bayes algorithm
- ◆ Train Support Vector Machine algorithm
- ◆ To classify:
- ◆ Compute  $P_{NB}(X) = \sum_{\text{task}} P(\text{task}) P(X|\text{task})$   
if  $P_{NB}(X) > \theta$  then use SVM prediction
- ◆ Naïve Bayes identifies data points that are unfamiliar and should not be predicted

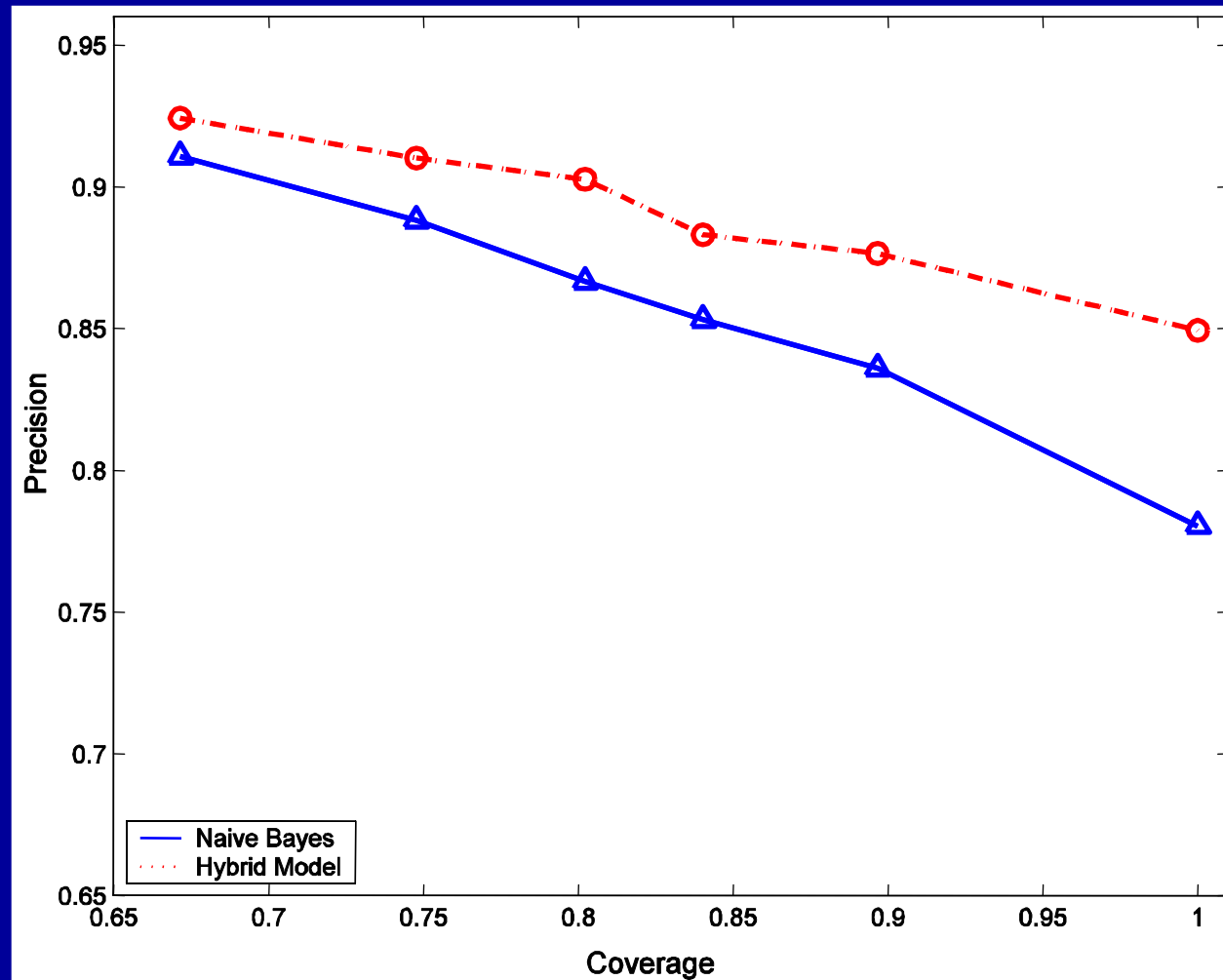
# Email Experiment

- ◆ Data Set

Subjects:	FA	RA	RB	SA	SB	SC	SD	SE
# messages	459	416	244	289	869	243	458	305
# tasks	21	23	12	9	8	14	5	15
# features	934	721	379	613	1158	598	448	349

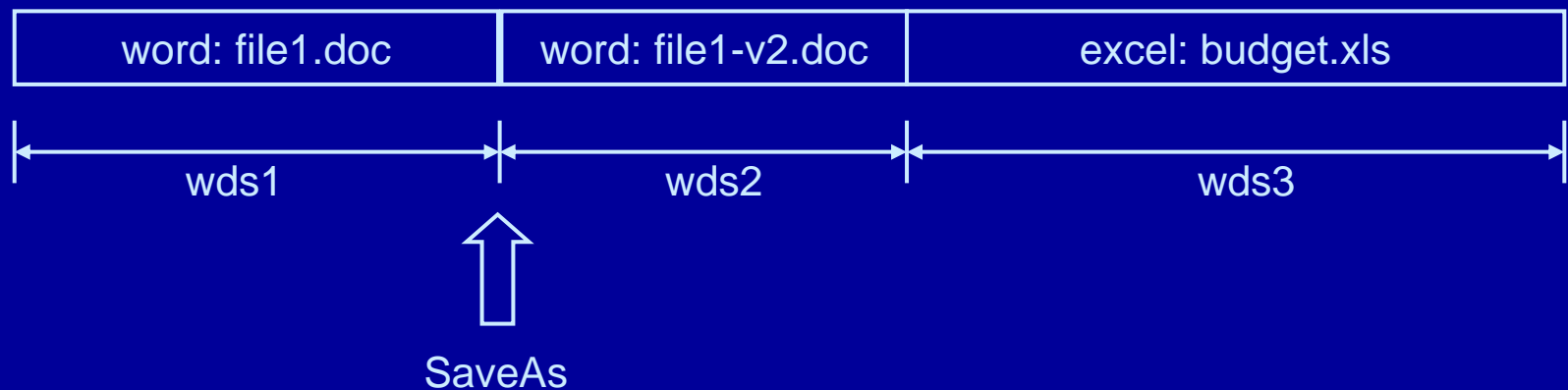


# Results



# Task Predictor

- ◆ WDS: Window Document Segment
  - time interval during with one window is open on one document



# WDS Features

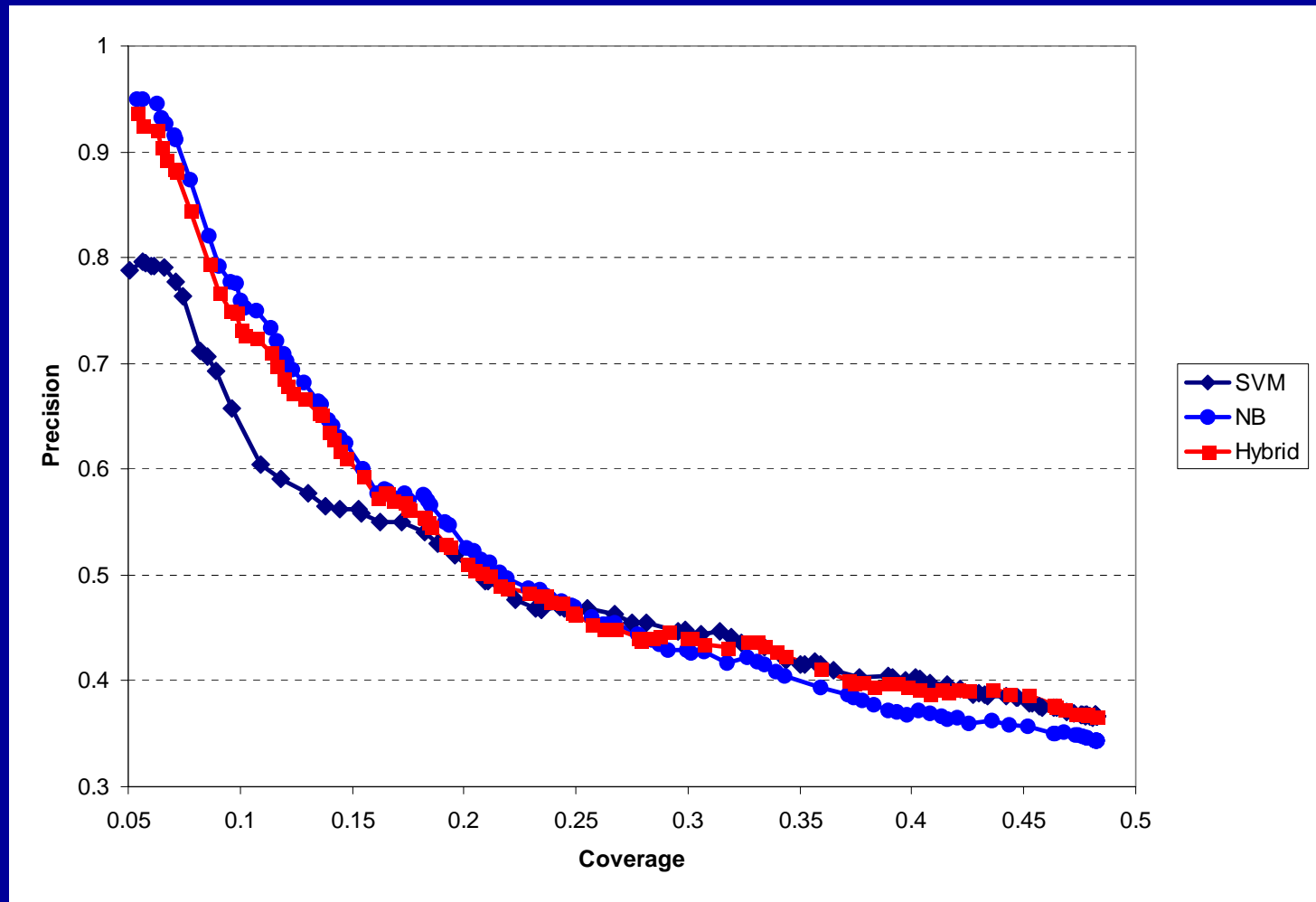
- ◆ Words in
  - window title
  - pathname of file
  - web site name
  - URL pathname of web page

# WDS Data

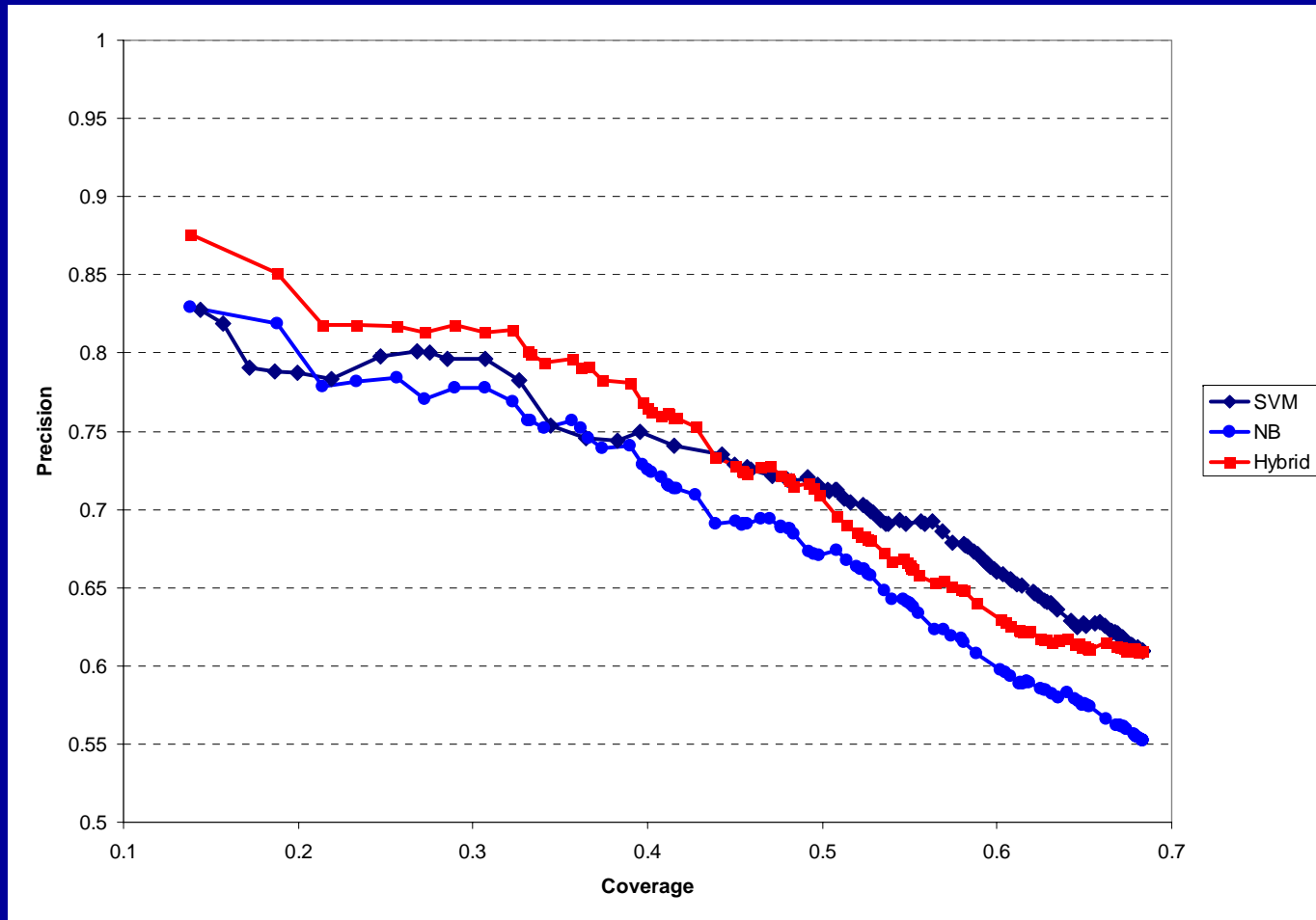
- ◆ Data sets:

Subjects:	FA	FB
# tasks	96	81
# WDSs	5894	4151
# features	1202	983

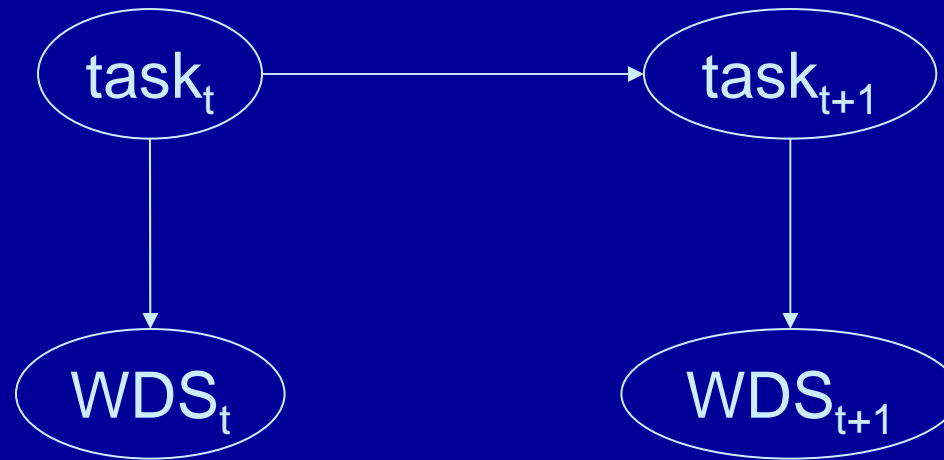
# Task Predictor Results (FA)



# Task Predictor Results (FB)



# Combining Multiple WDS Predictions: Hidden Markov Model



$P(\text{WDS}_t \mid \text{task}_t)$ : Naïve Bayes model

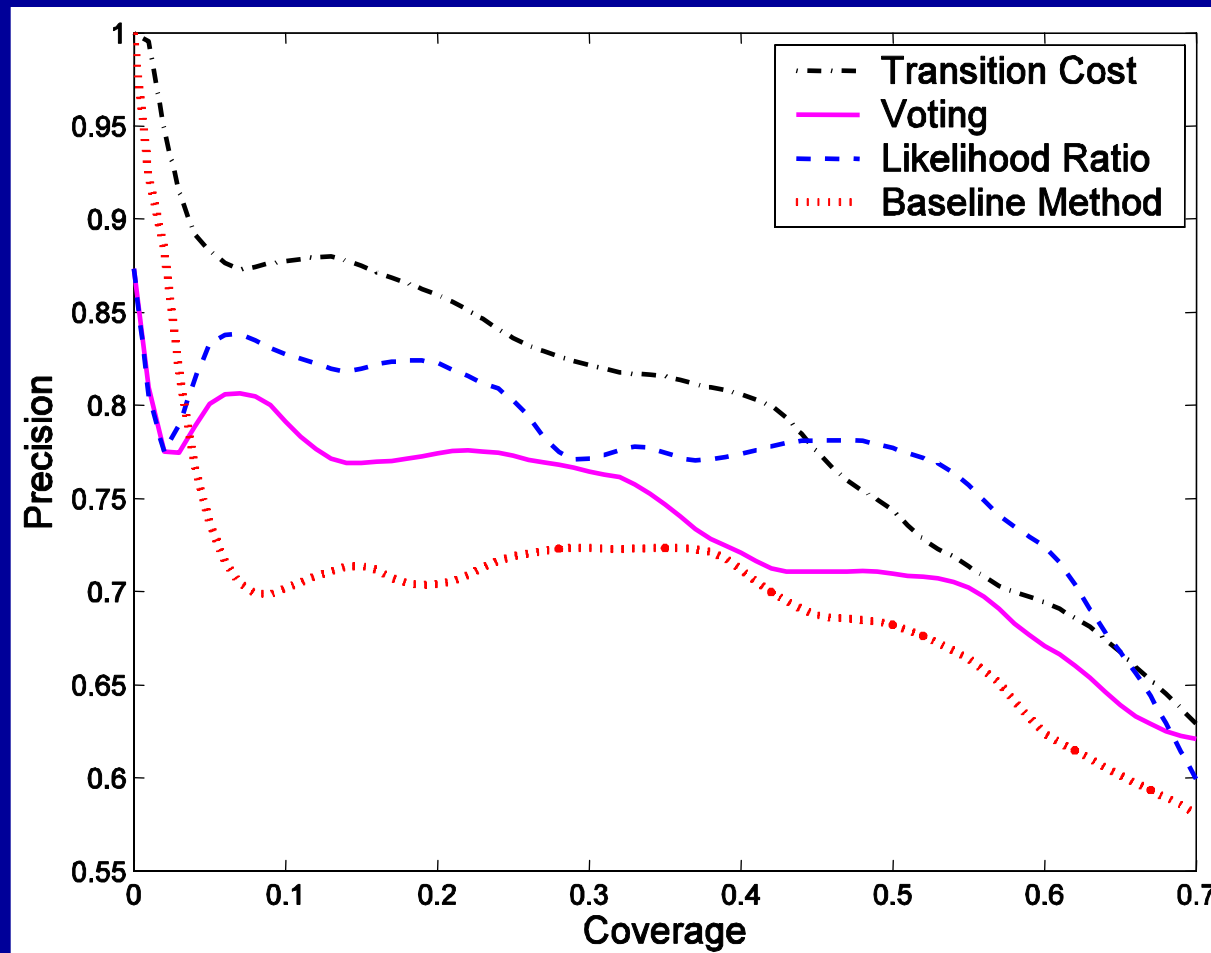
$P(\text{task}_{t+1} \mid \text{task}_t)$ : assume fixed probability of task switch

# Combining Multiple WDS Predictions

- ◆ Consider a sequence of task predictions  
 $P(\text{task}_1 | \text{WDS}_1), P(\text{task}_2 | \text{WDS}_2), \dots$   
How can we combine these to make more reliable predictions?
- ◆ Three methods studied
  - simple voting
  - likelihood ratio test (compare likelihood of single no-switch model to switch model)
  - HMM: transition cost + Viterbi



# Combining Multiple WDSs – Results

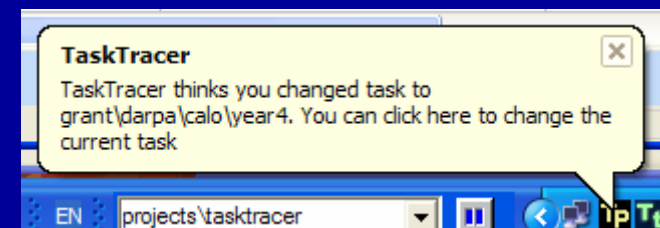
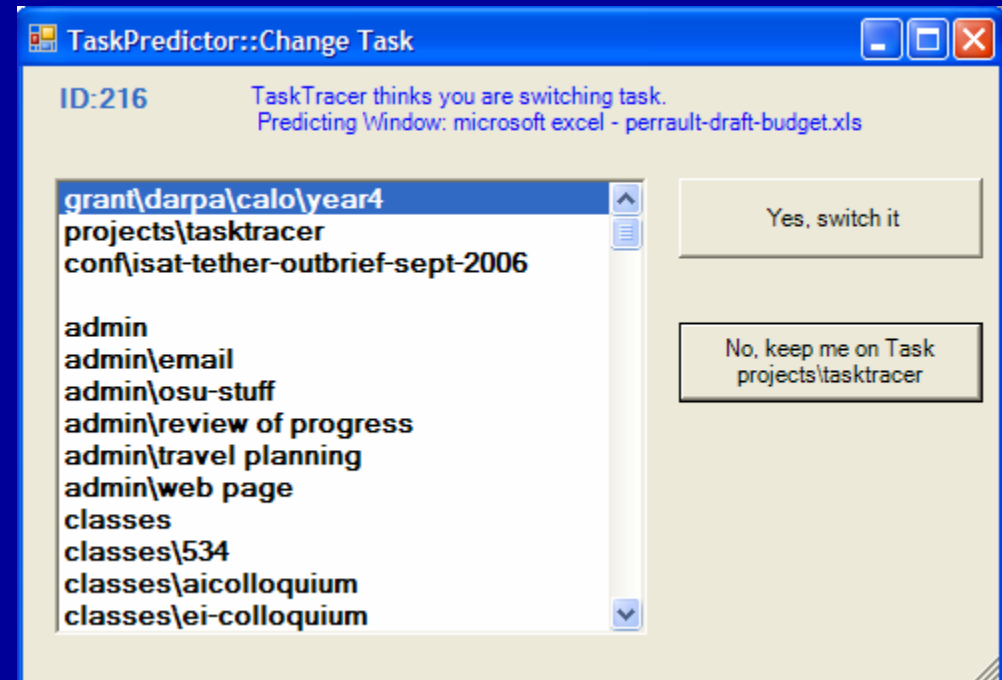


# Other Sources of Information

- ◆ Memorized document—task associations
- ◆ Hierarchical path name analysis
  - classes/cs/534
  - classes/cs/561
- ◆ Time since last task switch + episode duration models
- ◆ Generic indicators of task switching
  - save, close window
  - attach, send email
  - type in new URL (versus clicking on link)

# How to use the predictions?

- ◆ Interface: balloon alert in lower right corner of display
- ◆ Offers choice of
  - stay with current activity
  - switch to predicted activity
  - choose from menu of all activities

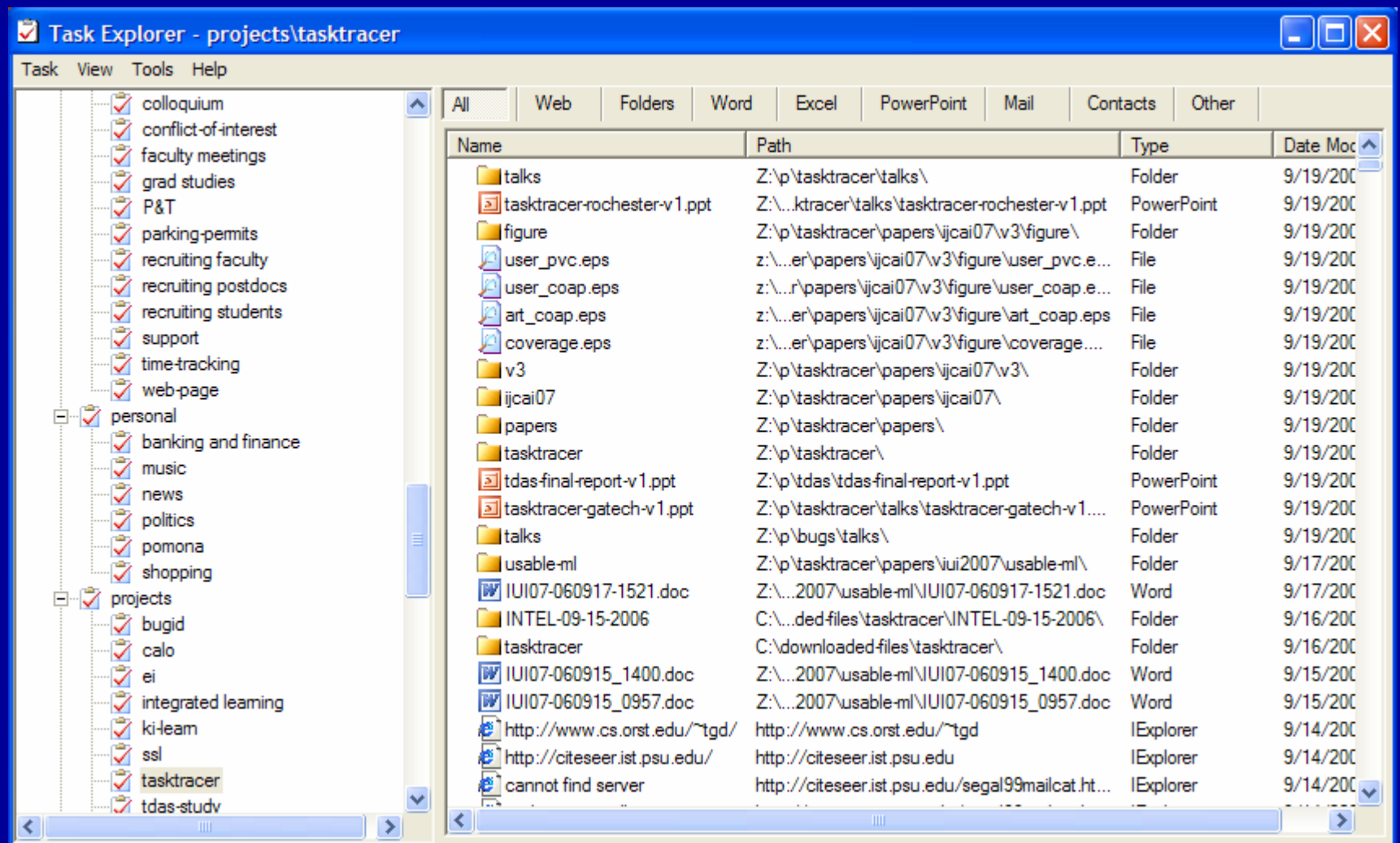


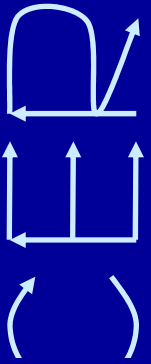
## 4. Task-Aware User Interfaces

- ◆ Task Explorer
  - Task Prototypes/Friends
  - Resource Explorer
- ◆ Folder Predictor
  - Windows Explorer Toolbar
- ◆ Task Notes
- ◆ Time Tracking

# TASKTRACER

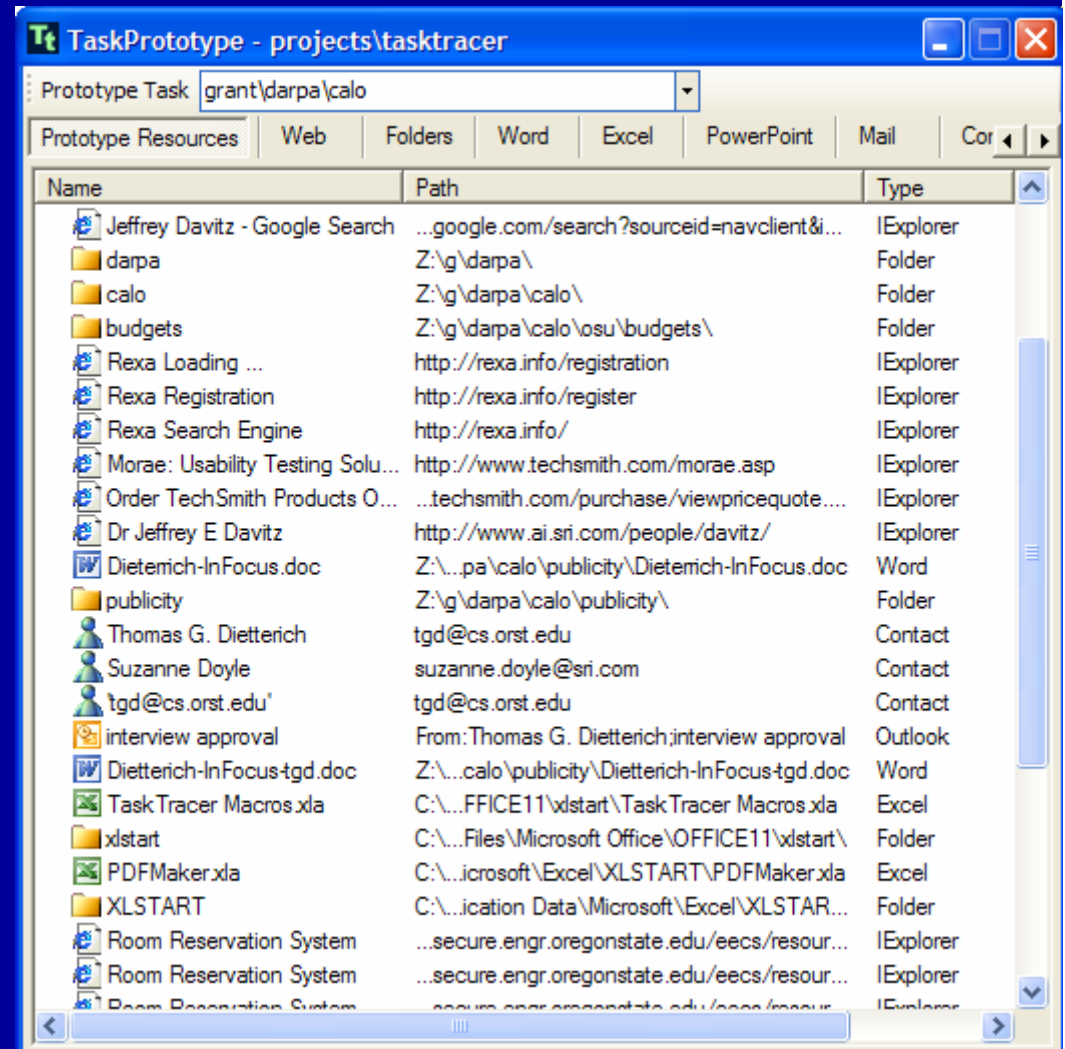
## TaskExplorer Provides Easy Access to Task Resources





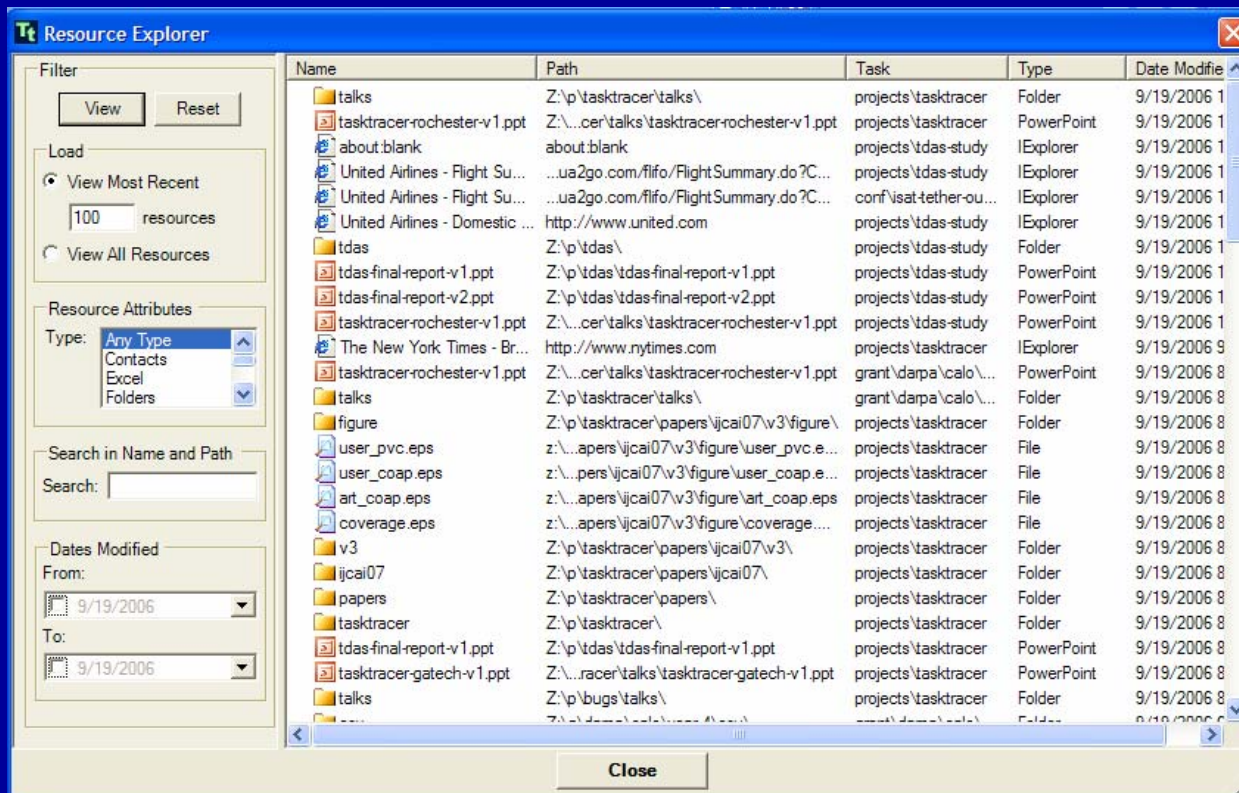
# Task Prototypes

- ◆ Makes it easy to access resources of related tasks
  - Example:
    - access classes/534-spring-05 when working on classes/534-spring-06
    - access projects/tasktracer when working on trips/rochester-sept-06
- ◆ Prototype docs are not auto-associated with the current task unless they are saved



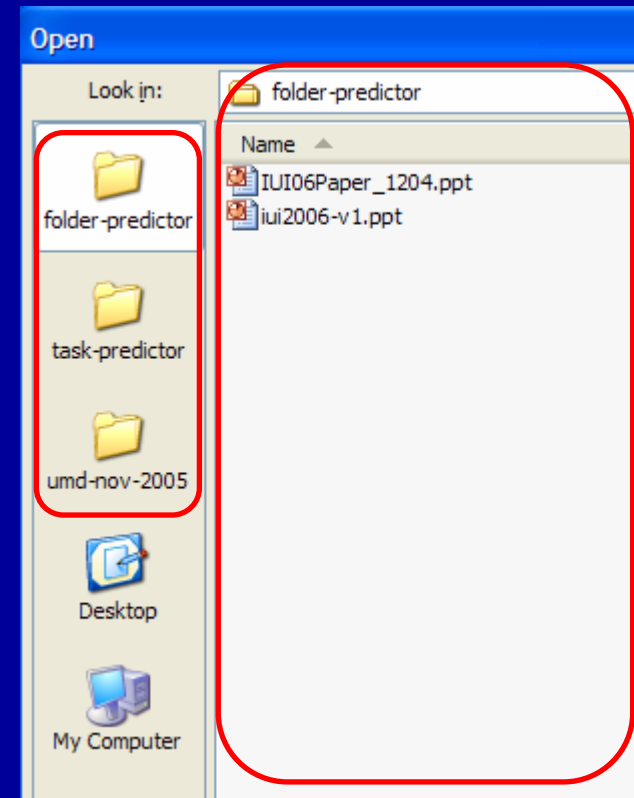
# Resource Explorer

- ◆ Sometimes need to find documents that you accesses recently but you don't know which task they were associated with



# Folder Predictor

- ◆ Maintain statistics on file opens and saves on a per-task basis
  - Recency-weighted count of saves and opens
- ◆ When user initiates open/save compute 3 folders to minimize expected number of clicks to get to the desired folder



$$\operatorname{argmin}_{\{f_1, f_2, f_3\}} \sum_f P(f \mid \text{task}) \cdot \min \{ \text{clicks}(f_1, f), 1 + \text{clicks}(f_2, f), 1 + \text{clicks}(f_3, f) \}$$



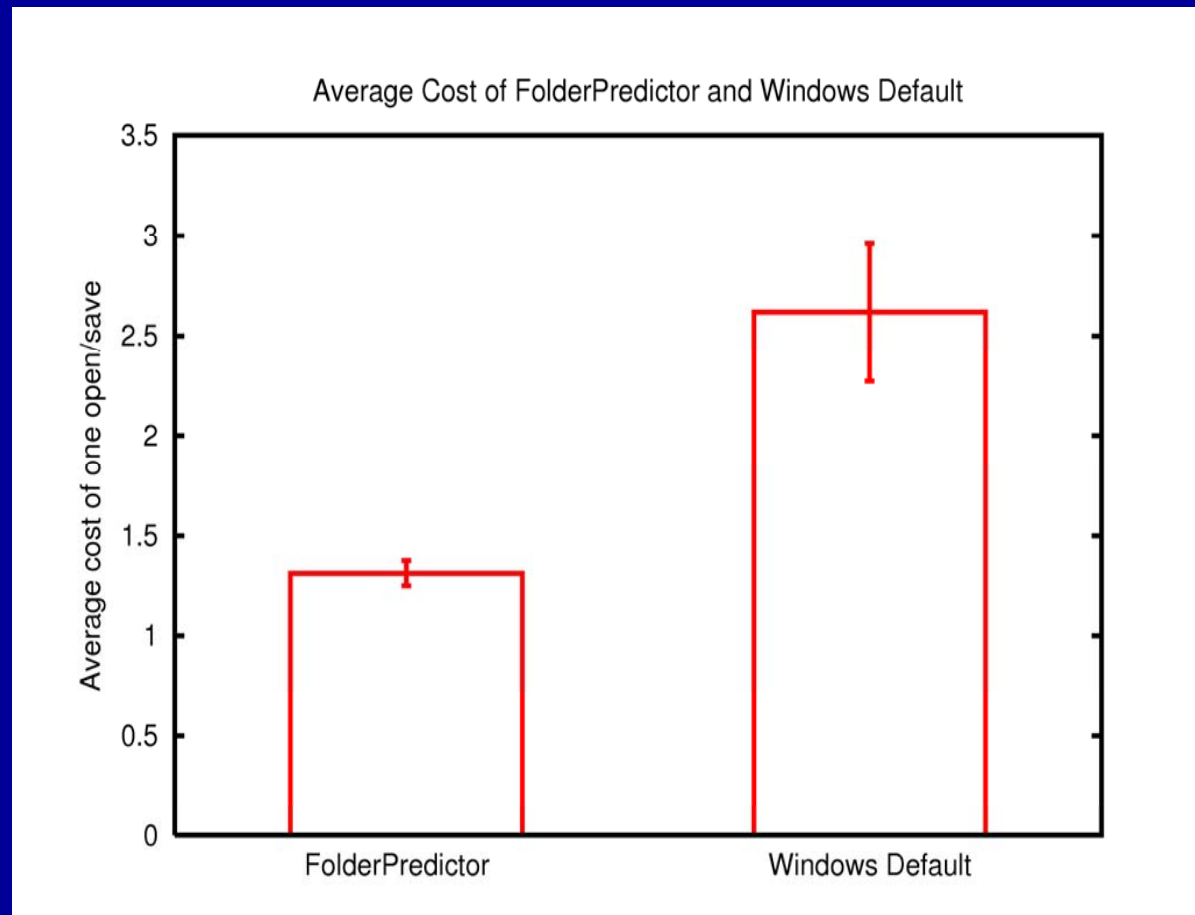
# Experiment

- ◆ Data Sets:

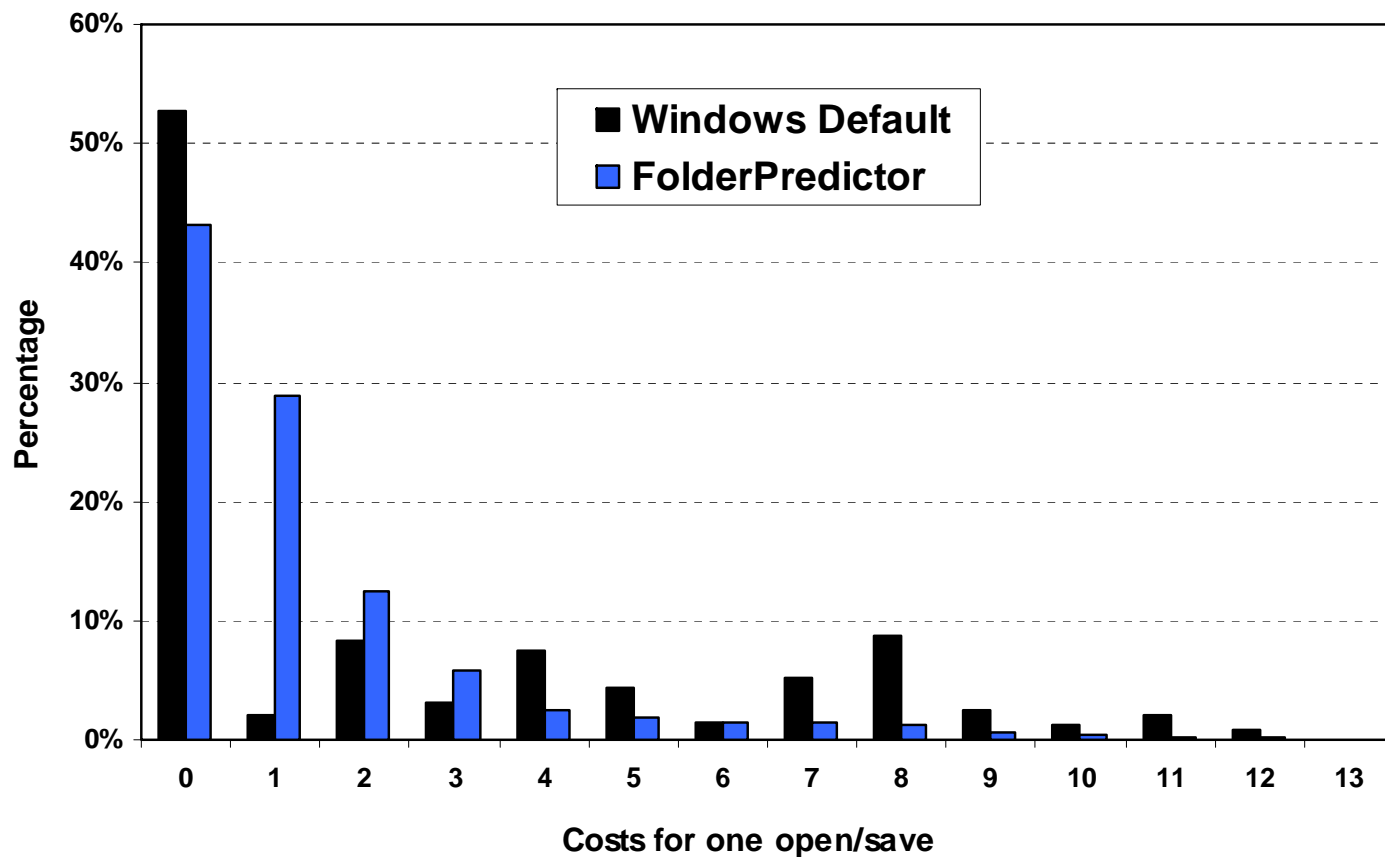
#	User Type	Data Collection Time	Set Size
1	Professor	12 months	1748
2	Professor	4 months	506
3	Graduate Student	7 months	577
4	Graduate Student	6 months	397

- ◆ Discount Factor  $\gamma = 0.85$

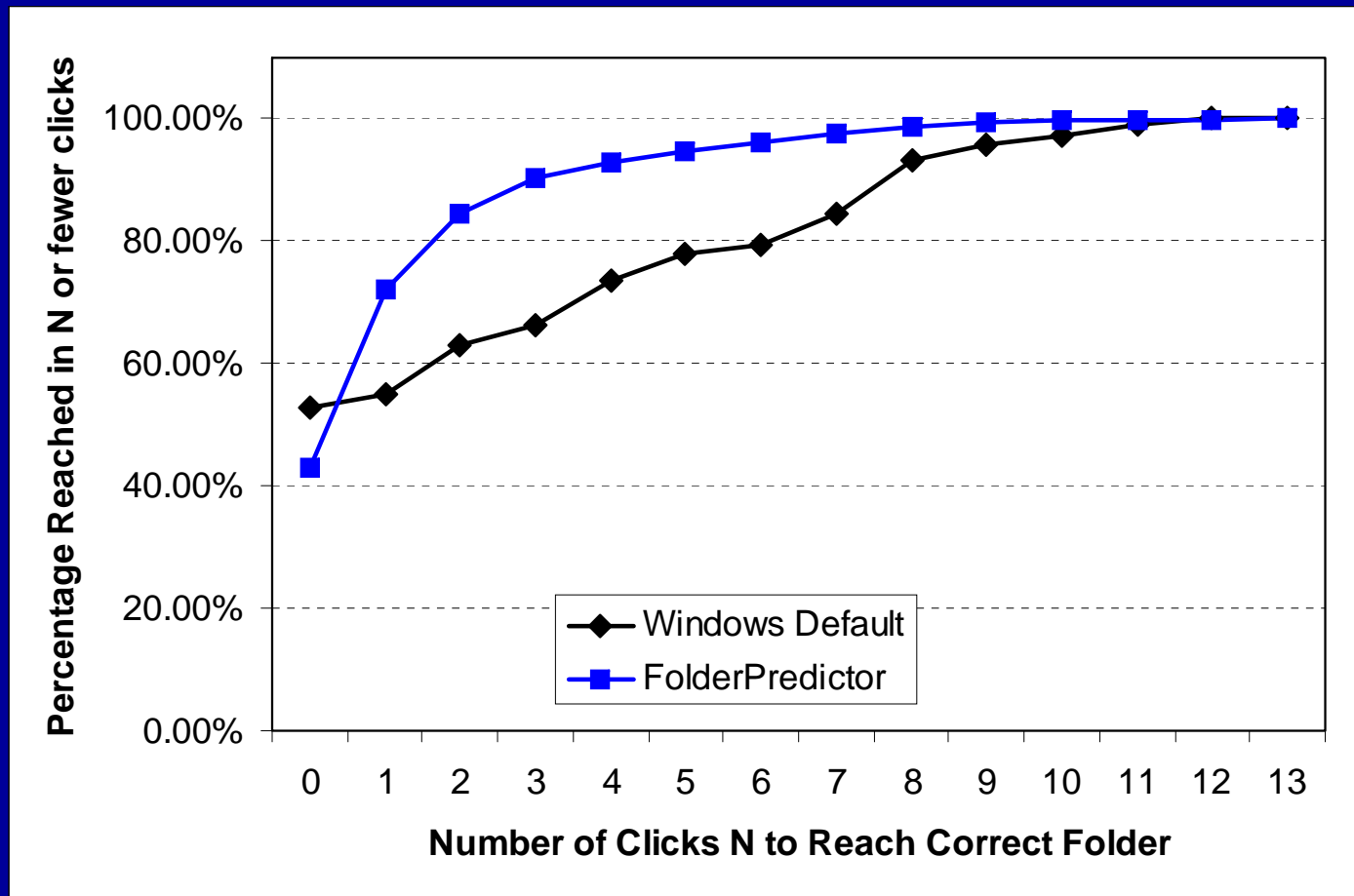
# Average Cost to Reach Target Folder



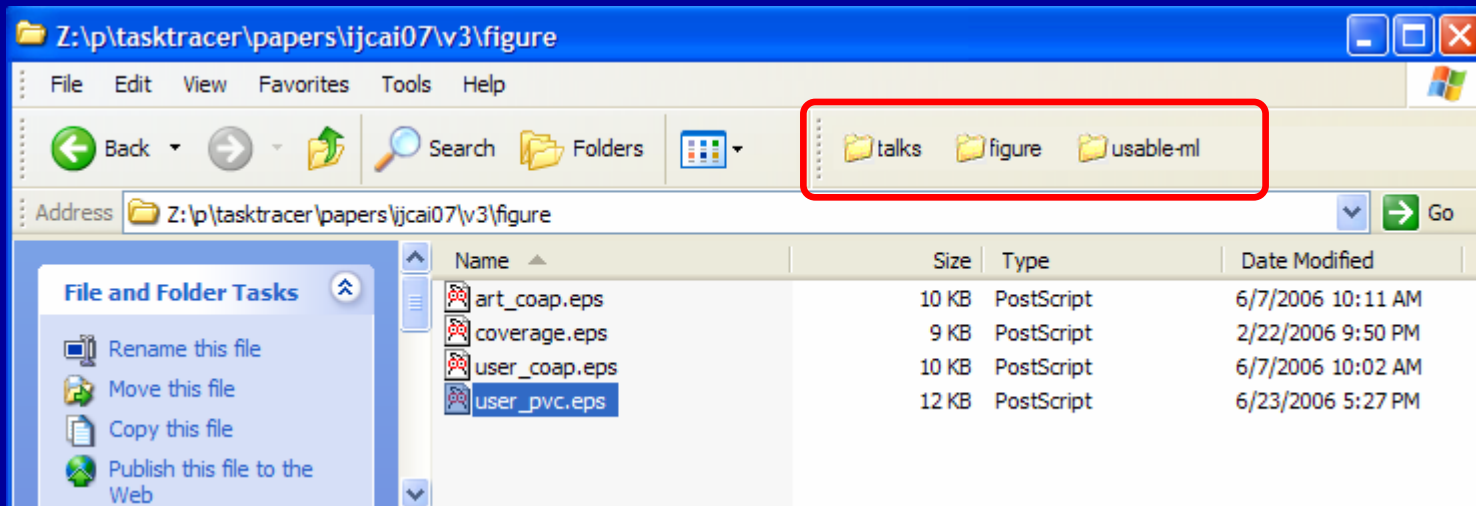
# Results: Many Fewer Clicks Required



# Cumulative Clicks Required

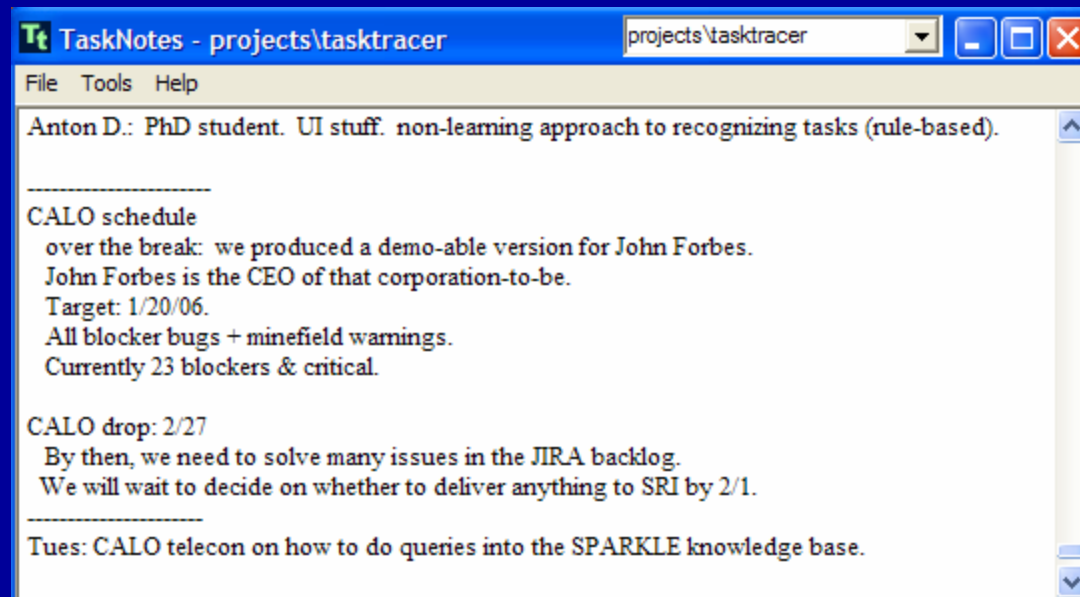


# Folder Predictor Toolbar in Windows Explorer



# Task Notes

- ◆ Notepad associated with the current task
- ◆ Time stamp automatically inserted each time you change tasks



# Time Tracking

- ◆ Where do you spend your time?
- ◆ Auditable for billing, etc.

**Task Statistics**

Time per task and other task statistics

From: 2006-09-01 00:00:00 To: 2006-09-19 13:39:16 [Pre-set Time Range] 180 Calculate

Task Path	Total time, h:m:s	Total time, sec	# of Times Worked on	AVG time, h:m:s	AVG time, sec
projects\tasktracer	31:04:40	111880	44	0:42:22	2542.7
Unknown	17:24:31	62671	27	0:38:41	2321.1
projects\tdas-study	13:16:24	47784	18	0:44:14	2654.7
admin\email	11:50:04	42604	17	0:41:46	2506.1
grant\darpa\calo\year4	10:50:55	39055	20	0:32:32	1952.8
projects\bugid	7:11:57	25917	7	1:01:42	3702.4
conf\isat-tether-outbrief-sept-2006	4:11:46	15106	13	0:19:22	1162.0
conf\cgrb-retreat-2006	3:42:35	13355	2	1:51:17	6677.5
conf\afri-september-2006	3:37:19	13039	4	0:54:19	3259.8
companies\smart-desk-top	3:32:09	12729	8	0:26:31	1591.1
projects\tl	3:06:33	11193	4	0:46:38	2798.3
grant\nsf\tasktracer-06	2:58:18	10698	4	0:44:34	2674.5
grant\darpa\integrated-learning	2:51:05	10265	13	0:13:09	789.6
admin\travel planning	2:48:01	10081	9	0:18:40	1120.1
student\hess	2:28:24	8904	2	1:14:12	4452.0
grant\darpa\ki-learn	2:14:53	8093	1	2:14:53	8093.0
projects\integrated learning	2:11:43	7903	3	0:43:54	2634.3
student\mehta	1:56:55	7015	5	0:23:23	1403.0

Save As... Close

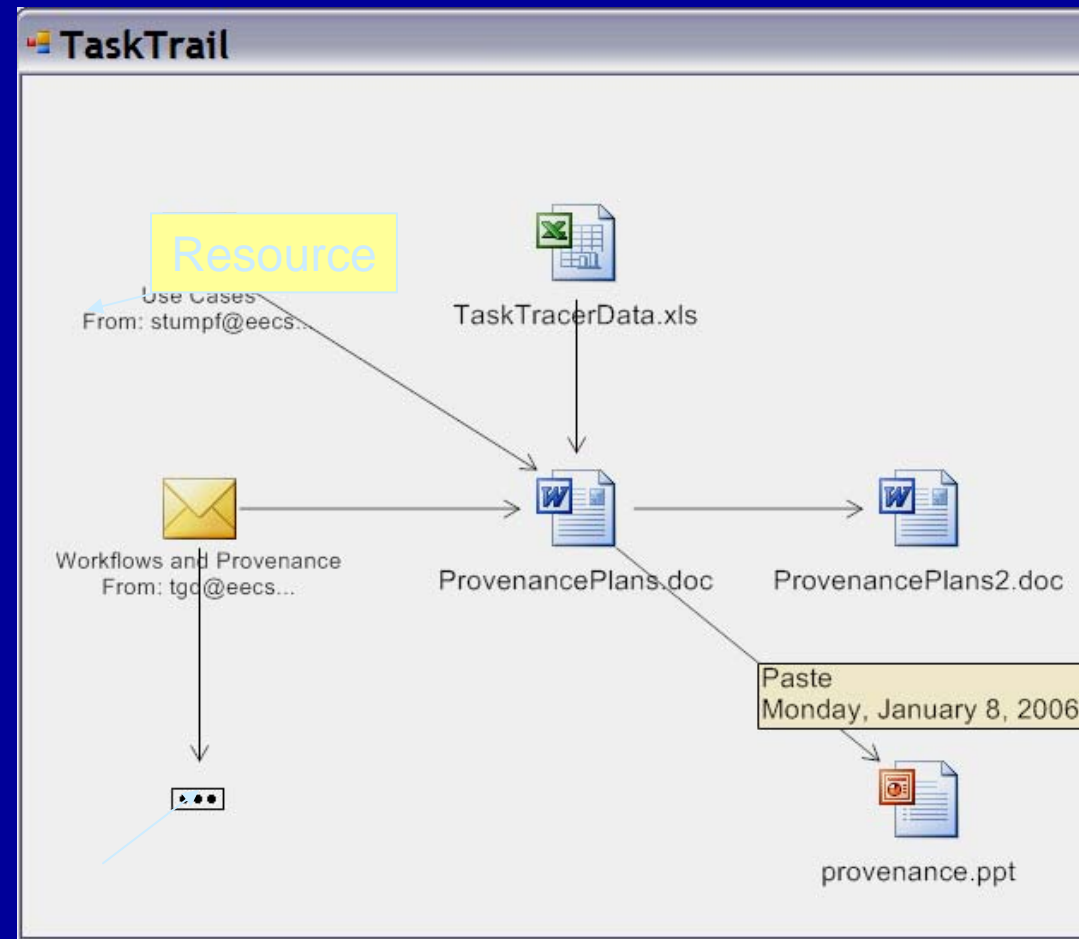
## Future Work

- ◆ Provenance-Based Information Access
- ◆ Activity Recognition and Proactive Assistance
- ◆ Combining Logical and Probabilistic Reasoning

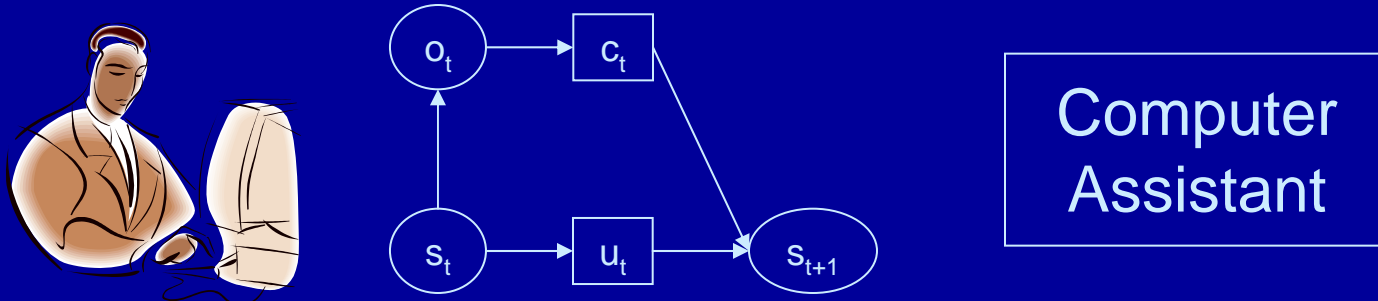


# Information Access via Provenance

- ◆ Right-click on object opens Provenance Graph
  - email header in Outlook
  - attachment in Outlook
  - file name in Windows Explorer



# Activity Recognition and Proactive Assistance



- ◆ Recognize new instance of known workflow (e.g., homework assignment)
  - course, deadline, URL
- ◆ Automatically add to the TODO list
- ◆ Automatically download assignment
- ◆ When commanded, upload solution
  
- ◆ Two Agent System:
  - User: state → action → state → action → state → action ...
  - CALO:
    - watches **observable** user behavior
    - infers **unobservable** state (goals, plans)
    - takes autonomous action to minimize expected cost to the user
  - Some actions are **coordination** actions

# Integrating Logic and Probability: Markov Logic

- ◆ Knowledge base: weighted formulas in first-order logic over finite domains
- ◆ Probabilistic interpretation:  
 $P(\text{truth assignment}) = \frac{1}{Z} \exp[\sum \text{weight of satisfied formulas}]$
- ◆ Inference
  - Find most likely truth assignment  
Weighted Max Satisfiability
  - Compute probability of a ground formula or ground literal  
Markov Chain Monte Carlo (MCMC) method based on slice sampling (Gibbs Sampling)

# Summary

1. Instrument desktop applications
  - Publish/Subscribe architecture; MySQL back end
2. Define/discover the user's "tasks"
  - User enters hierarchy of tasks
3. Associate events/resources with tasks
  - User declares current task
  - All events/resources are associated with that task
  - Task Predictors can predict current task instead
4. Build/modify interfaces to provide easy access to relevant resources and events
  - Task Explorer
  - Folder Predictor
  - Task Notes

# Acknowledgments

- ◆ Jon Herlocker: visionary and project leader
- ◆ Simone Stumpf: project manager
- ◆ Margaret Burnett: end-user design and evaluation
- ◆ Kevin Johnsrude and Jed Irvine: software architect, software engineer
- ◆ Lida Li, Jianqiang Shen: Task & Email Predictors
- ◆ Xinlong Bao: Folder Predictor
- ◆ Many other students, both undergrad and grad
- ◆ Funding Agencies:
  - NSF: MKIDS program
  - DARPA: PAL/CALO program
  - Intel