TASK TRACER

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

TASK TRACER

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?"

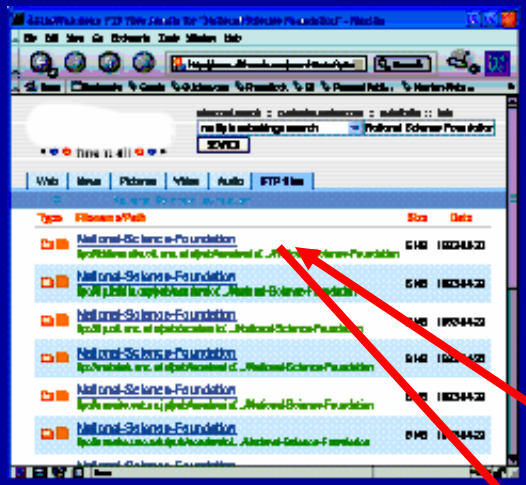
2. Jane describes her current task to TaskTracer

"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 TRACER

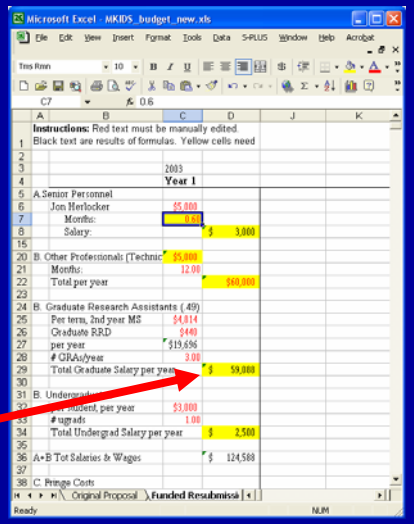


4. A list of resources Jane used while writing her last grant is displayed: files, web pages, email addrs, phone numbers.

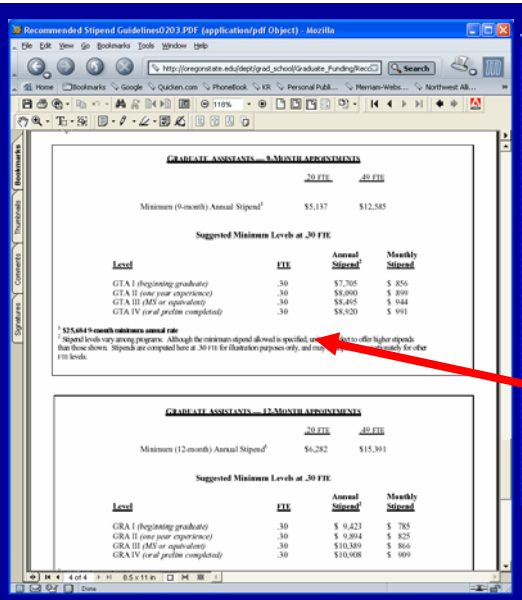
"Aha! I'll start with the budget I used last time as a template."

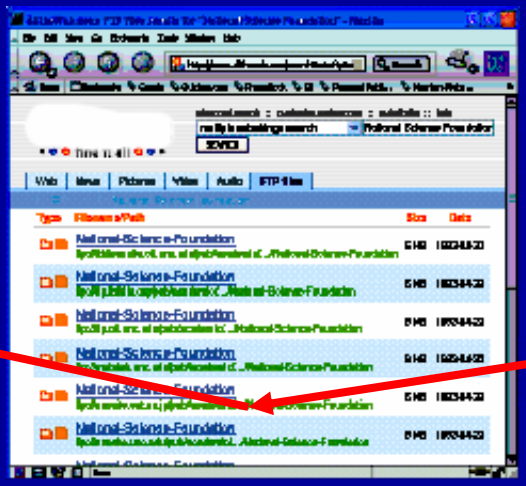
5. The budget spreadsheet is opened. The cells that were edited in the previously selected task are highlighted.

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



TASK TRACER



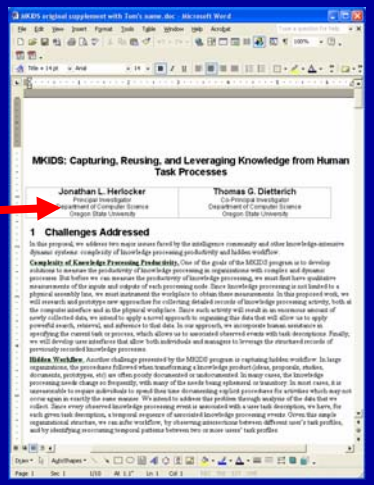


6. TaskTracer remembers the previous cut/paste of this information. The spreadsheet cells have right-click option: "Show source URL…"

"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!"



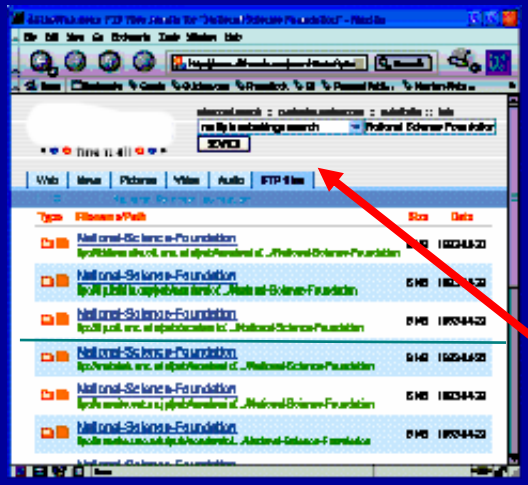
8. Interruption! Spouse calls and says he's outside.

"Need to log out and head home…"

TASK TRACER

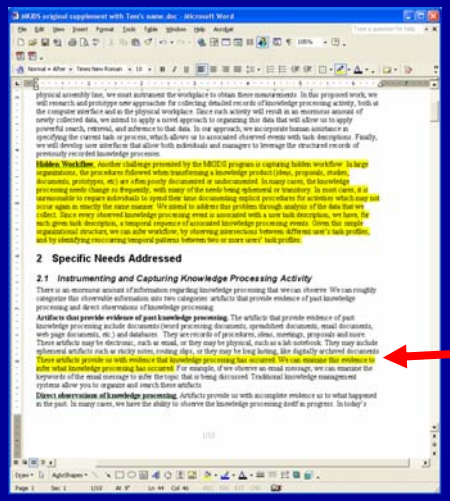
9. Next day after she logs in.
"What was I working on???"

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"

TASK TRACER



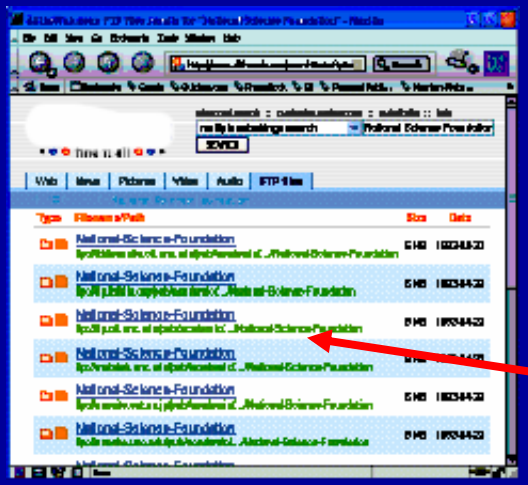
12. TaskTracer opens the Word document and highlights the text that she most recently edited.

"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"

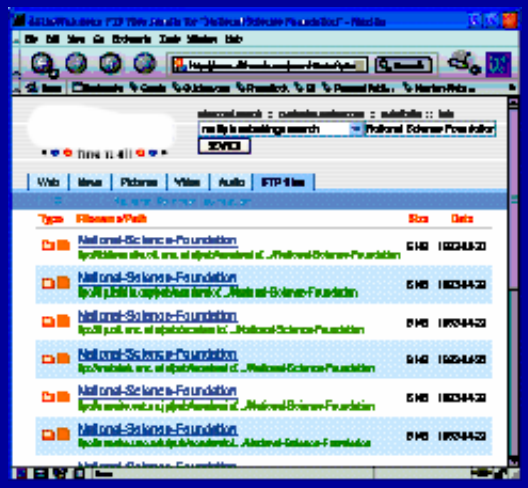
TASK TRACER

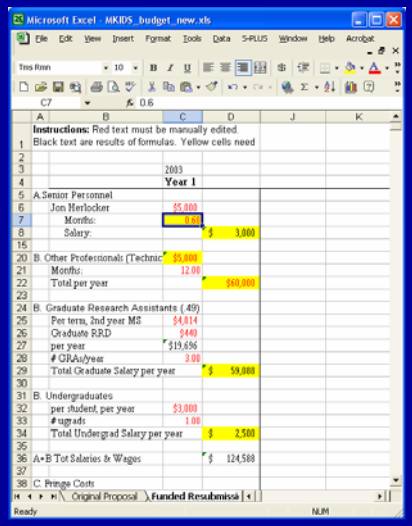


14. TaskTracer pulls up a record of the task and lists past emails and phone calls to and from that caller. *“I can quickly scan the recent email messages to and from Chris to recall what we last discussed”*

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”





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

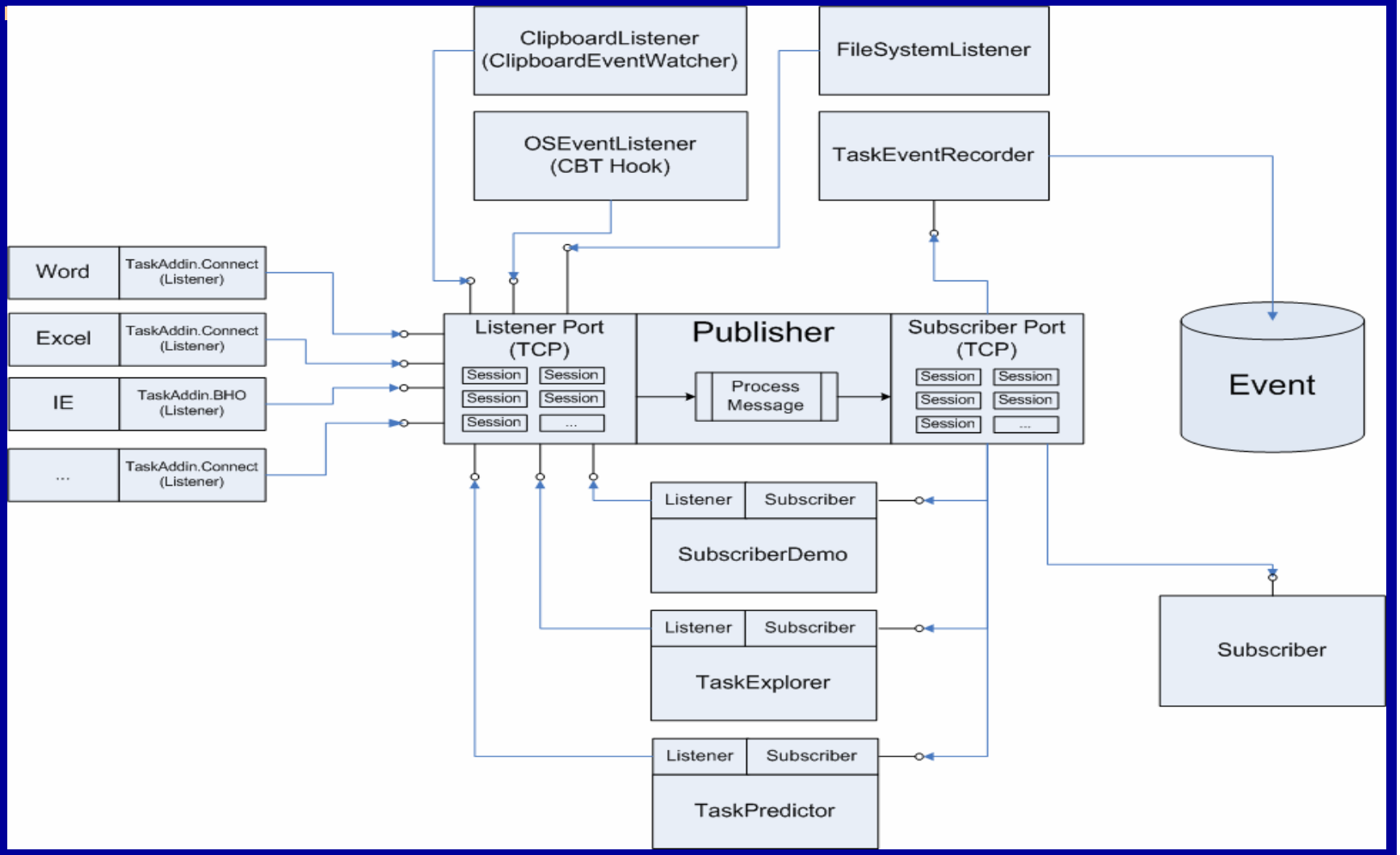
The Activity Hypothesis

- 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



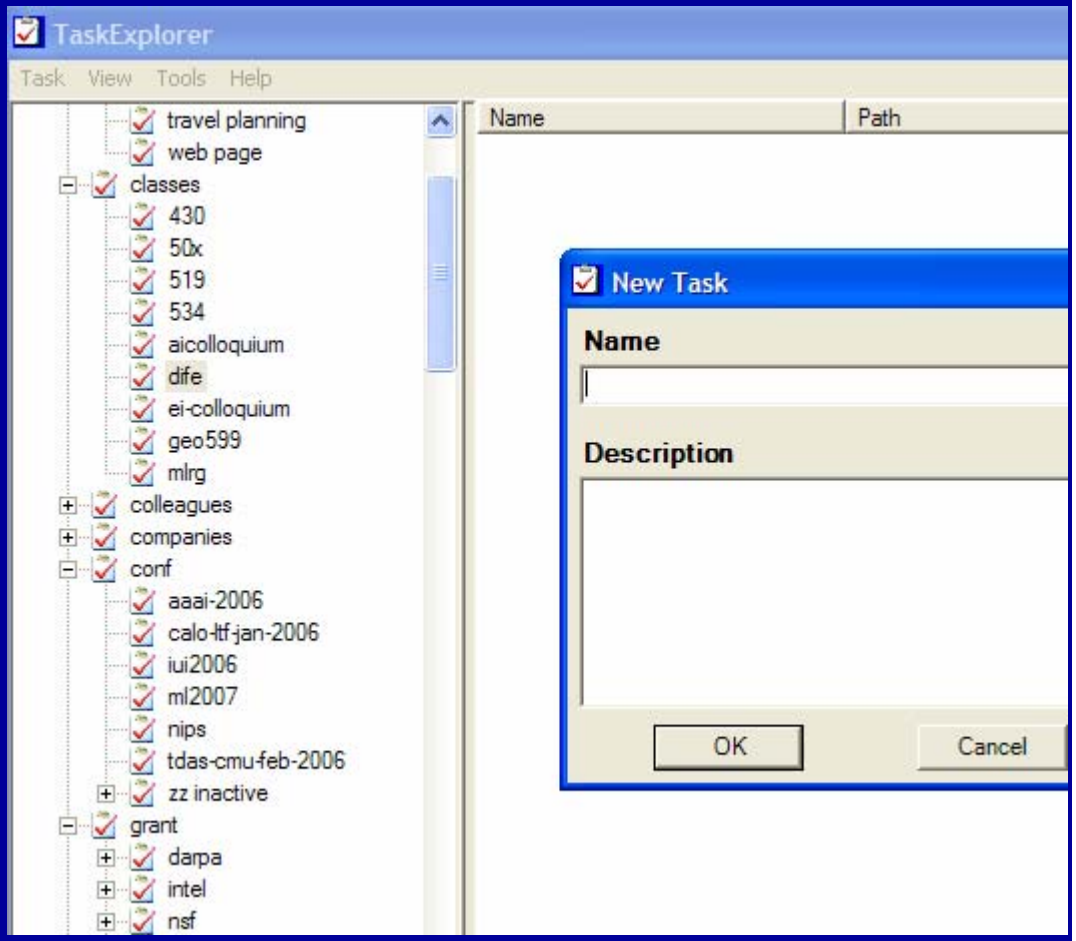
TASK TRACER

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)

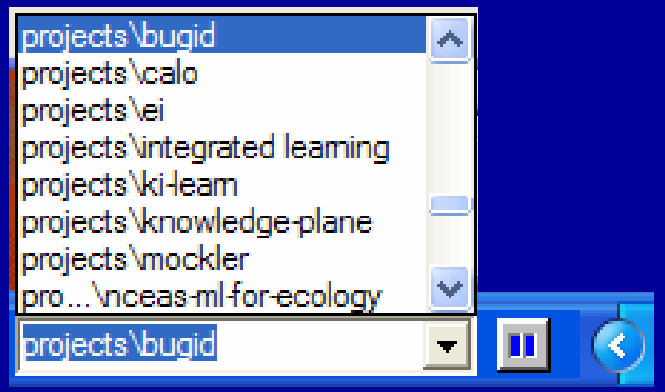
See: Huang, D. Govindaraju, T. M. Mitchell, V. R. Carvalho, W. W. Cohen (2004) Inferring Ongoing Activities of Workstation Users by Clustering Email

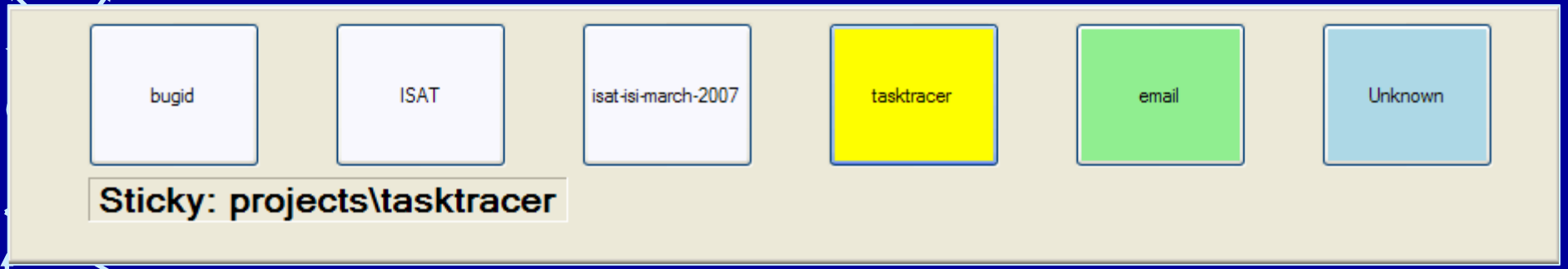
3. Associate events/resources with tasks

- 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)

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

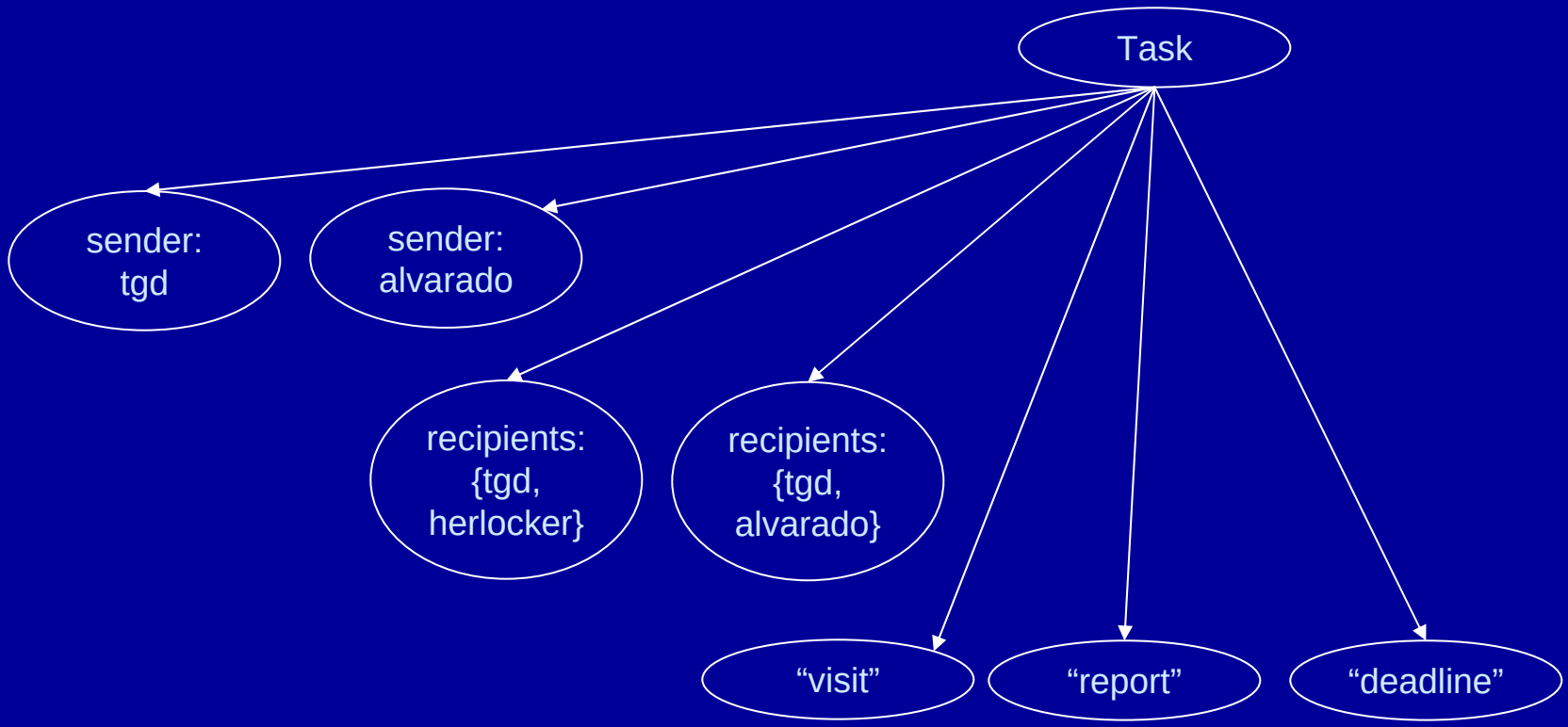
Machine Learning Challenges

- 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



$$P(task \mid X) = \alpha\ P(x_1|task)\ P(x_2|task) \cdots\ P(x_n|task)\ P(task)$$

Hybrid Learning Algorithm

- Train Naïve Bayes algorithm
- Train Support Vector Machine algorithm
- To classify:
- Compute $P_{NB}(X) = \sum_{task} P(task)\ P(X|task)$

  if $P_{NB}(X) > \theta$ then use SVM prediction

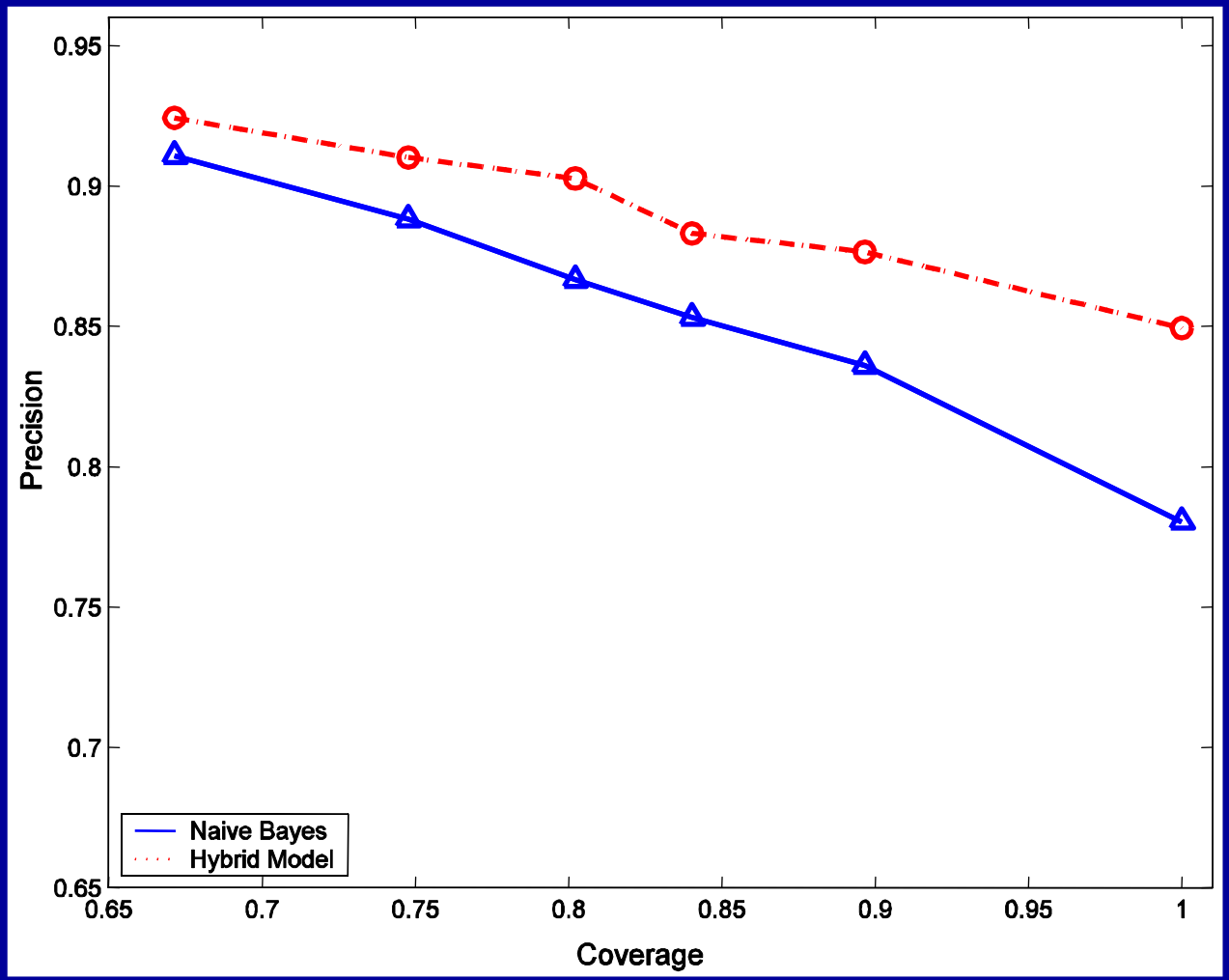
- Naïve Bayes identifies data points that are unfamiliar and should not be predicted

Email Experiment

- Data Set

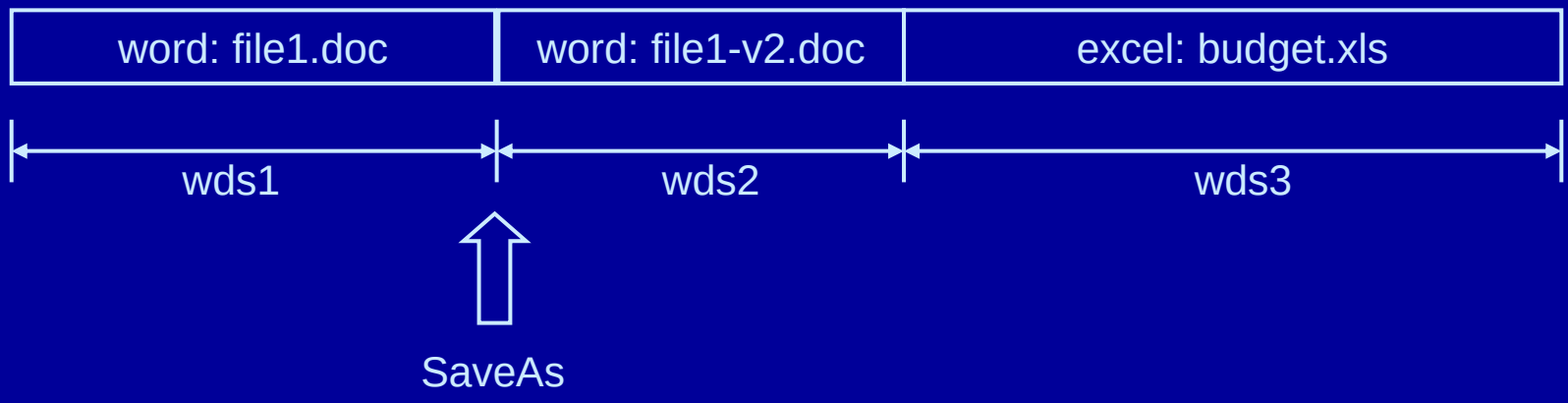
Subjects:	FA	RA	RB	SA	SB	SC	SD	SE
# messages	459	416	244	289	869	243	458	305
# tasks	21	23	12	9	8	14	5	15
# features	934	721	379	613	1158	598	448	349

Results


0.95
0.9
0.85
0.8
0.75
0.7
0.65
Precision
0.65
0.7
0.75
0.8
0.85
0.9
0.95
1
Coverage
Naive Bayes
Hybrid Model

Task Predictor

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



WDS Features

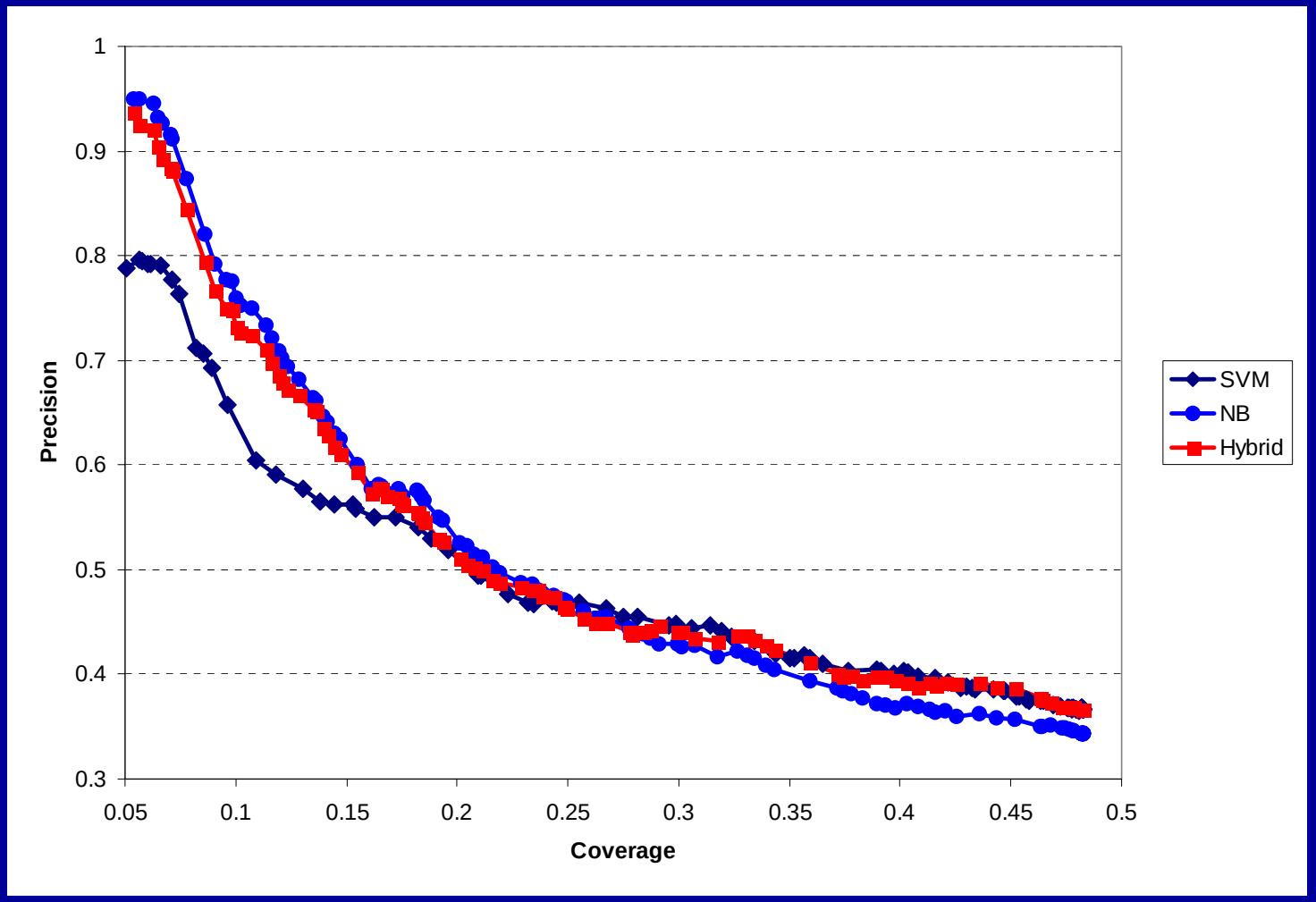
- Words in
  - window title
  - pathname of file
  - web site name
  - URL pathname of web page

WDS Data

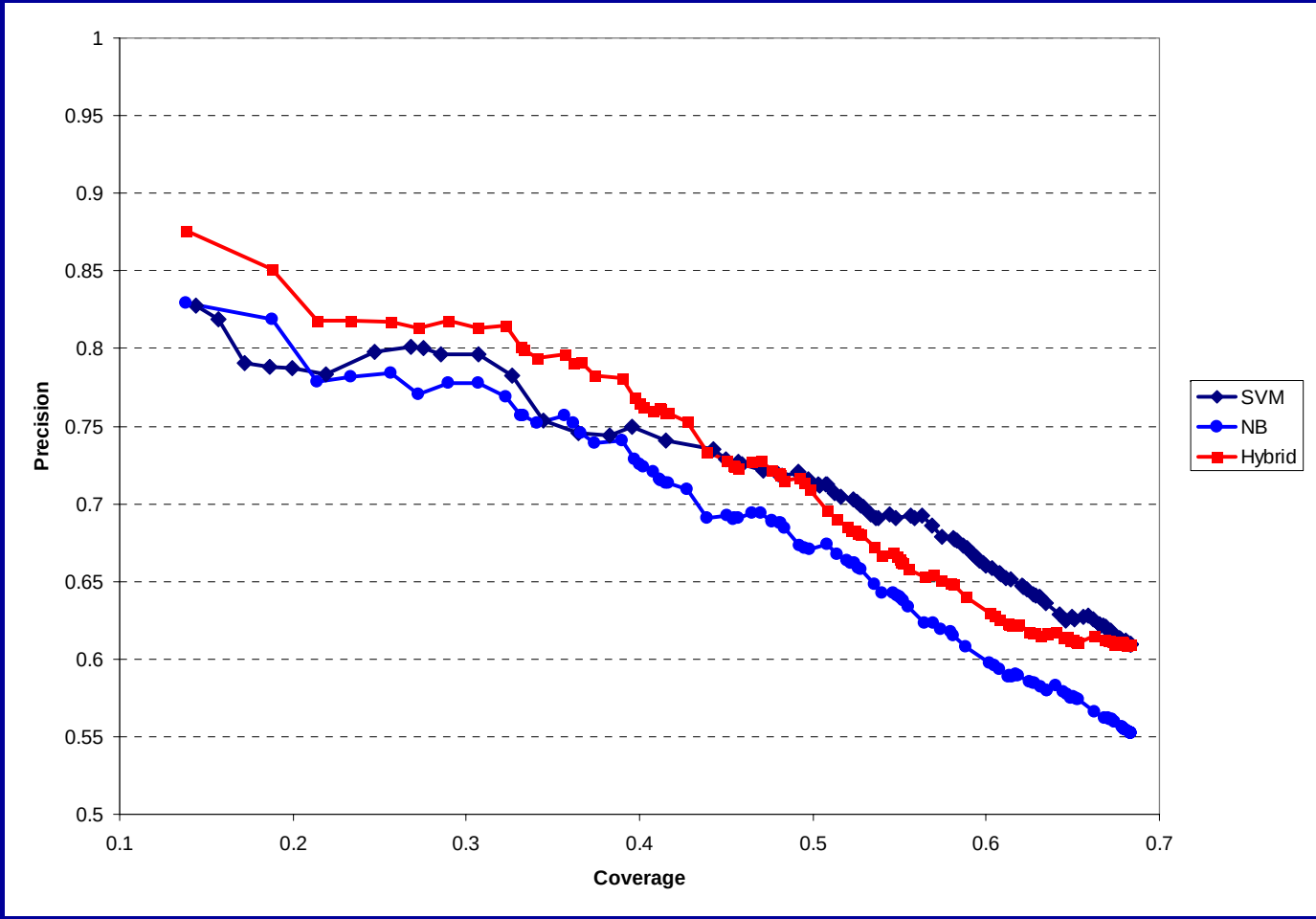
- Data sets:

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

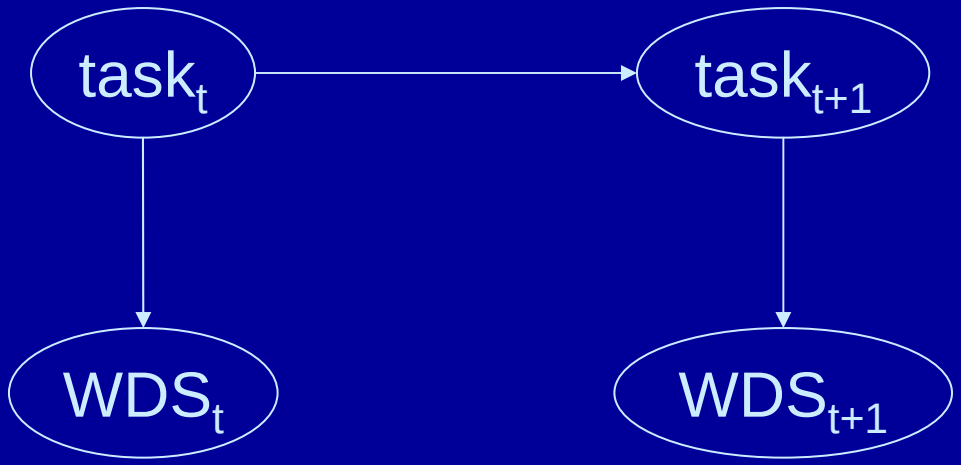
Task Predictor Results (FA)



Task Predictor Results (FB)



Combining Multiple WDS Predictions: Hidden Markov Model



$P(WDS_t \mid task_t)$: Naïve Bayes model

$P(task_{t+1} \mid task_t)$: assume fixed probability of task switch

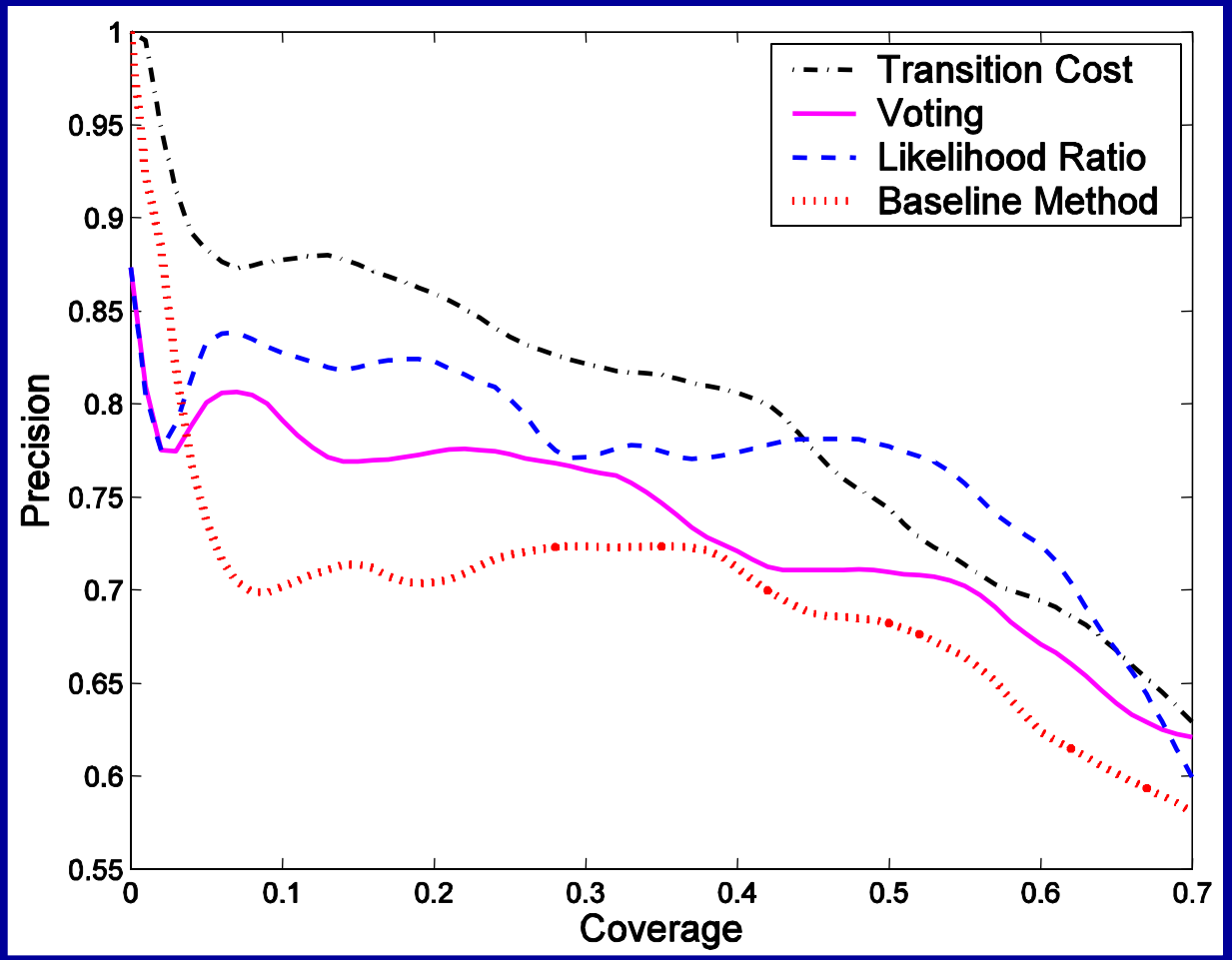
Combining Multiple WDS Predictions

- Consider a sequence of task predictions

  $P(task_1 \mid WDS_1)$, $P(task_2 \mid WDS_2)$, …

  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



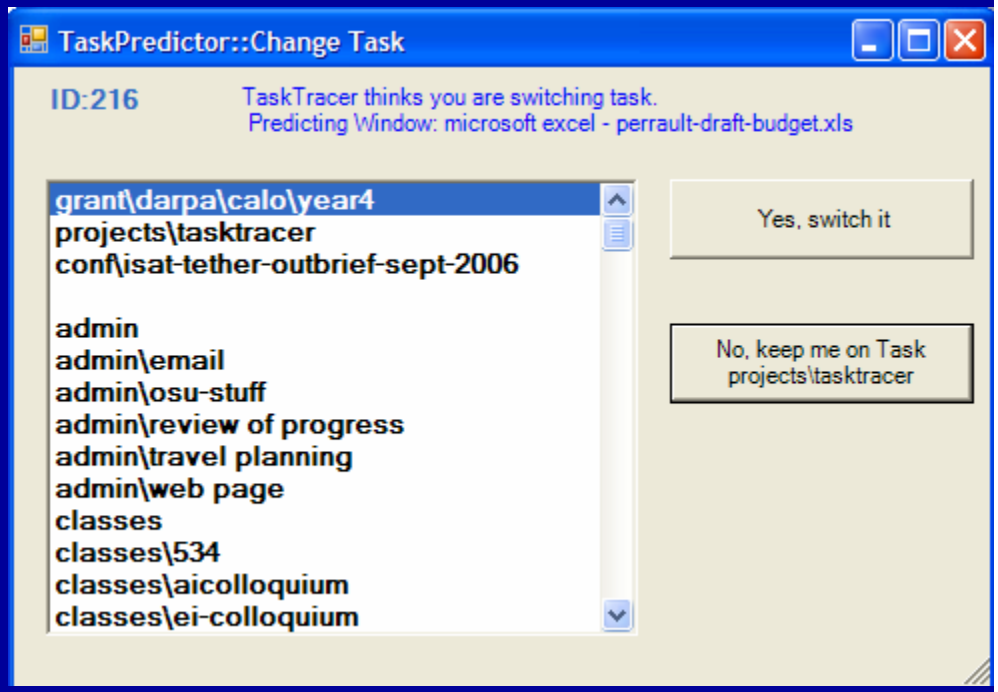
Other Sources of Information

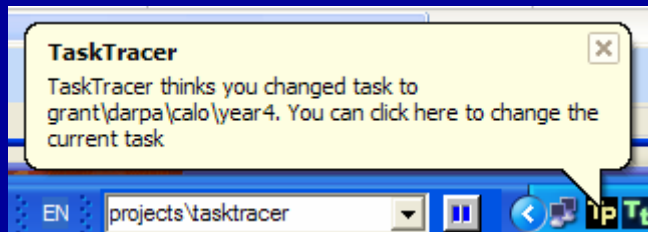
- Memorized document—task associations
- Hierarchical path name analysis
  - classes/cs/534
  - classes/cs/561
- Time since last task switch + episode duration models
- Generic indicators of task switching
  - save, close window
  - attach, send email
  - type in new URL (versus clicking on link)

TASK TRACER

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



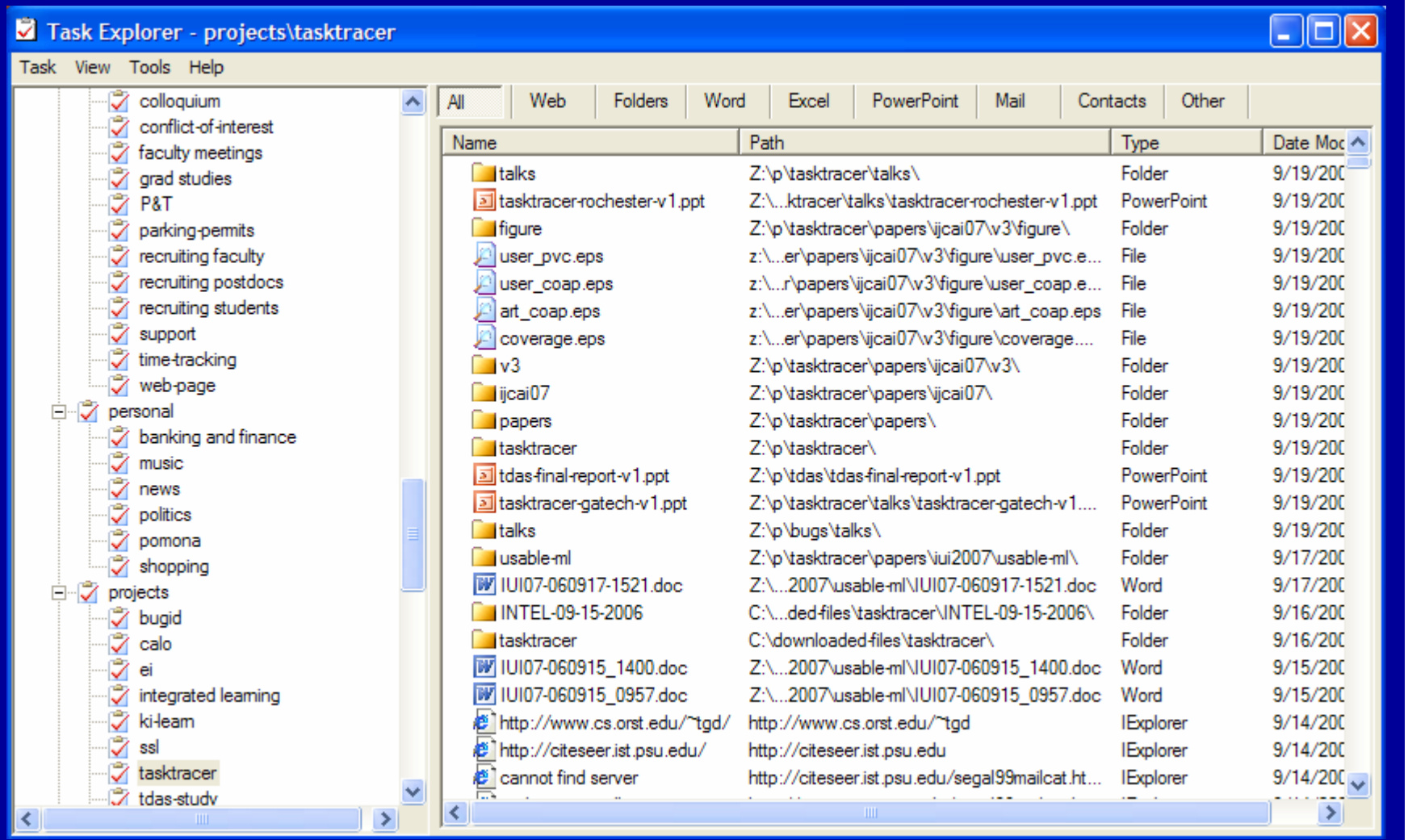


4. Task-Aware User Interfaces

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

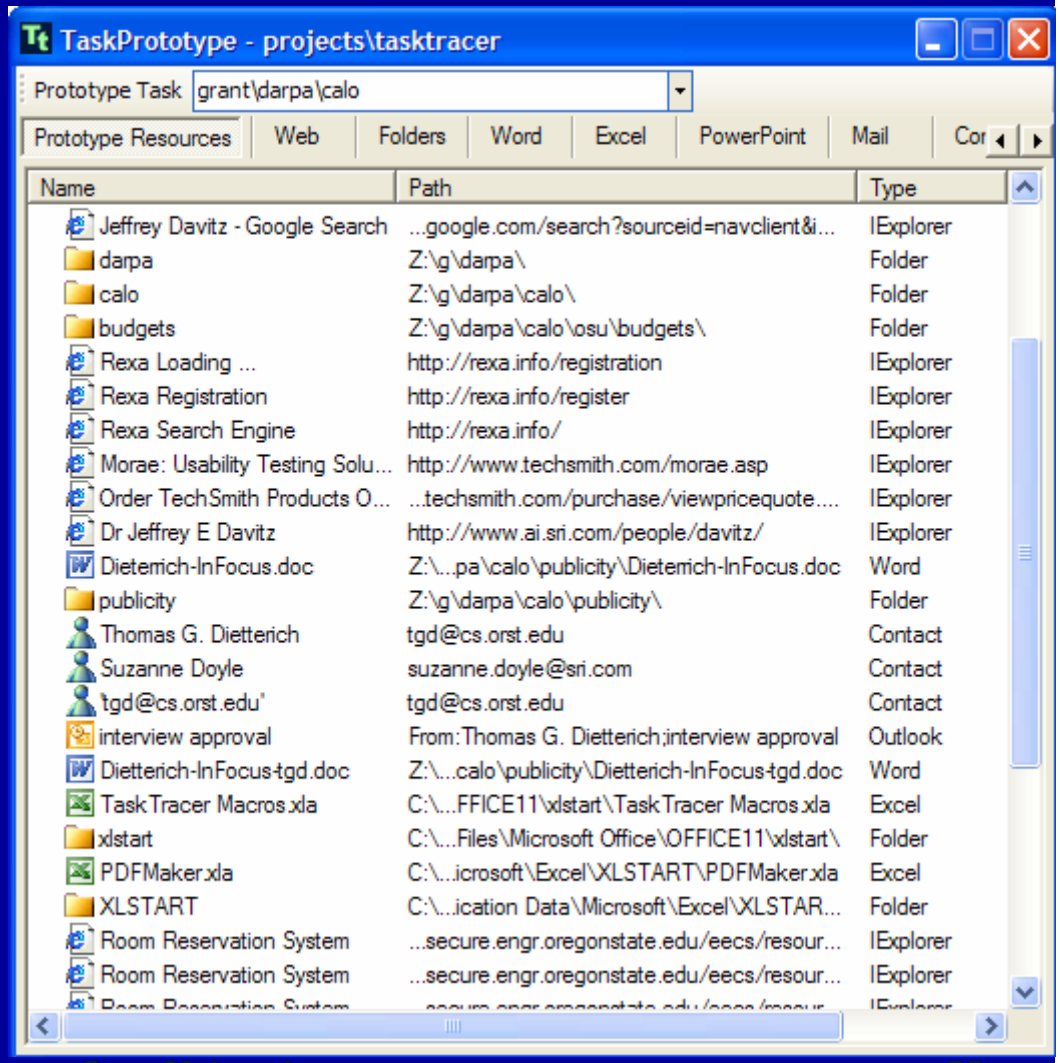
TASK TRACER

TaskExplorer Provides Easy Access to Task Resources



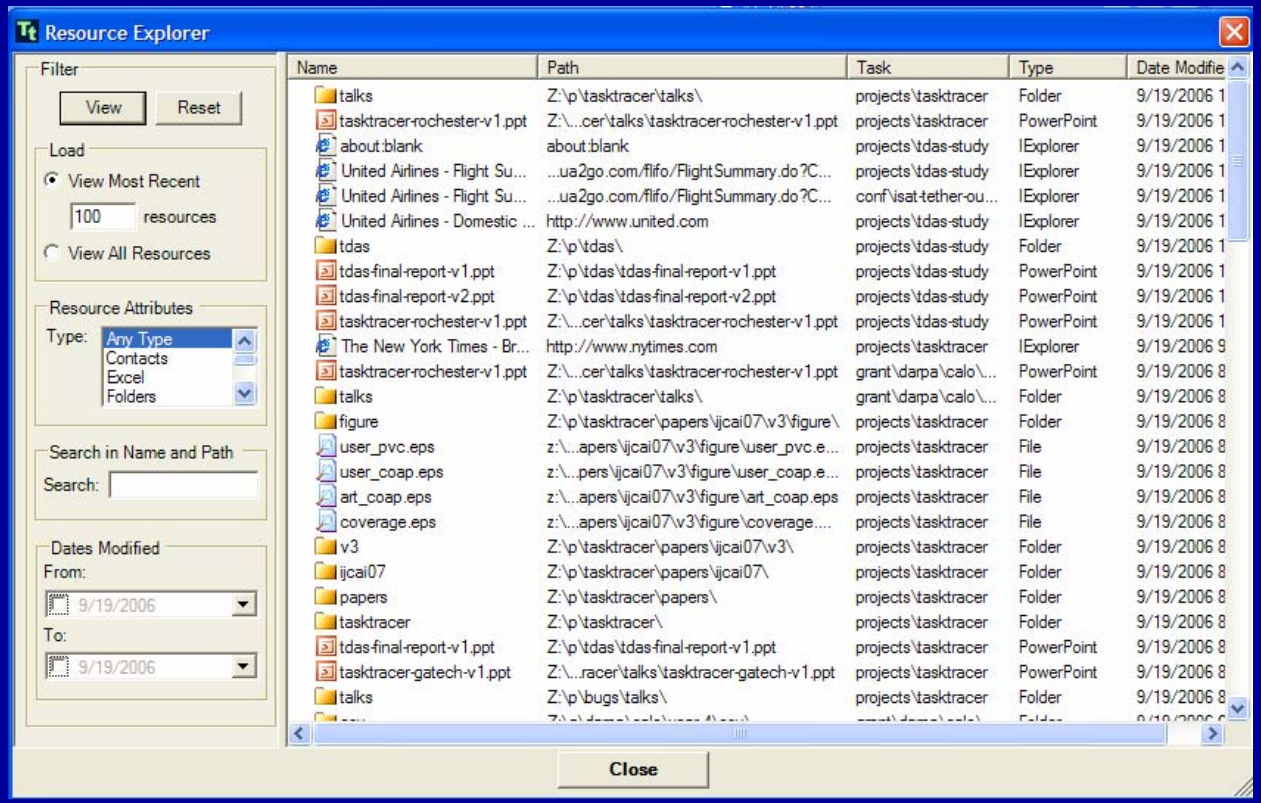
Task Prototypes

- 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



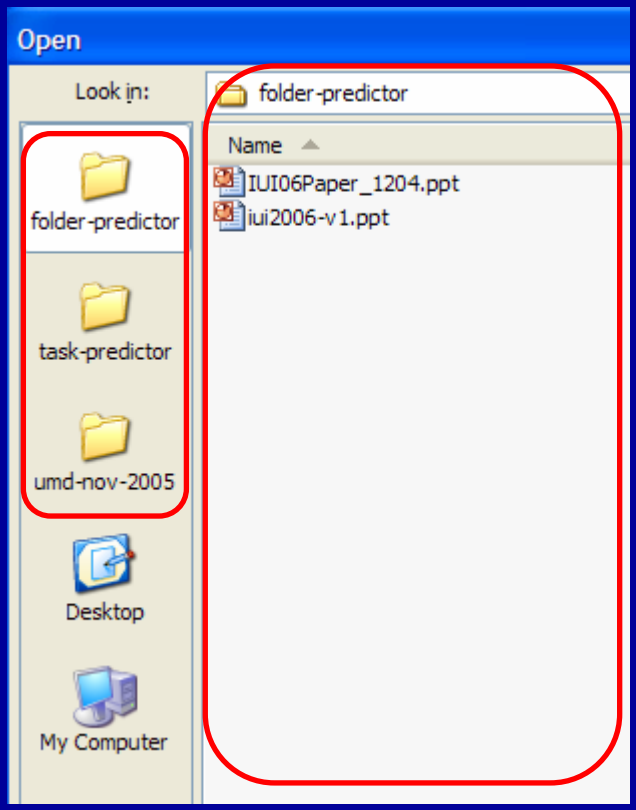
Resource Explorer

- 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 foldler



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

Experiment

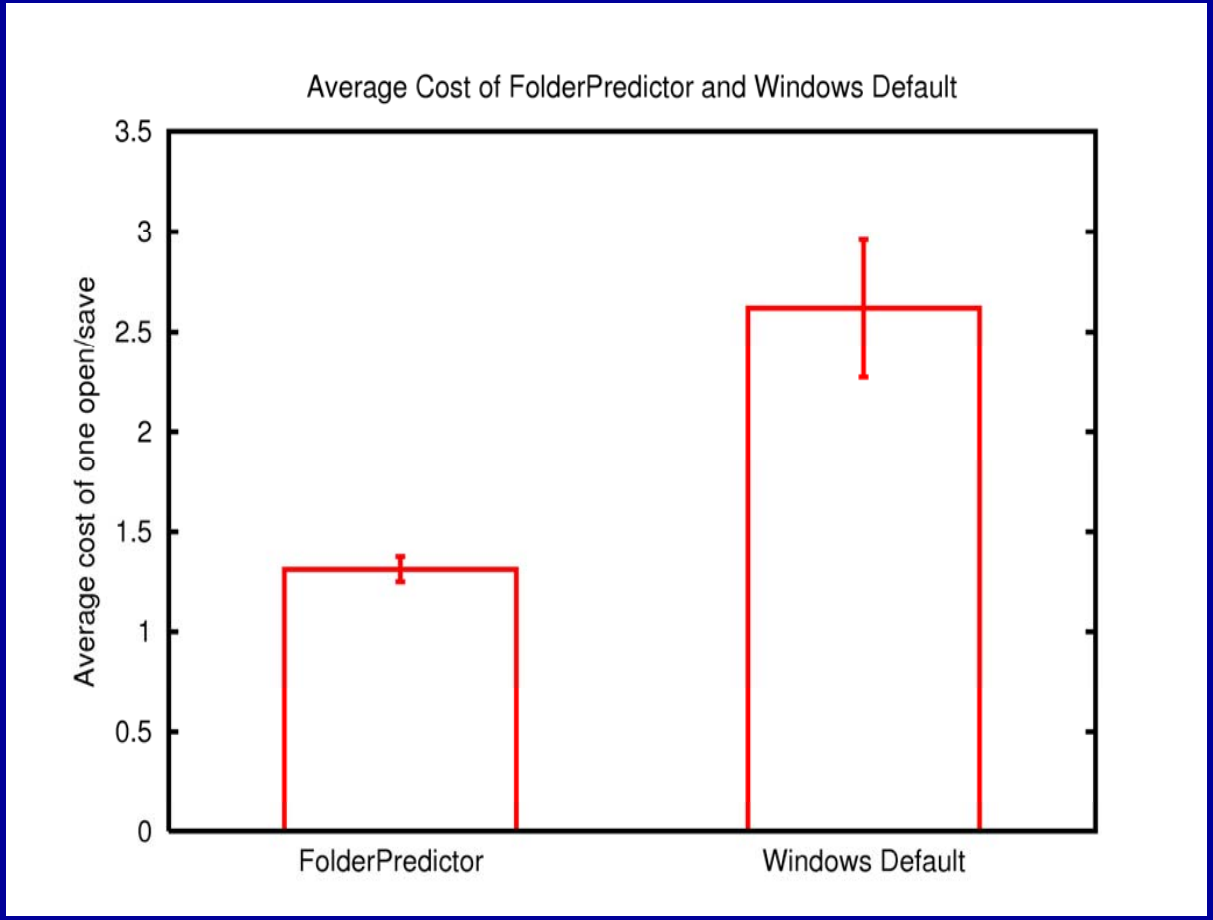
- Data Sets:

#	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

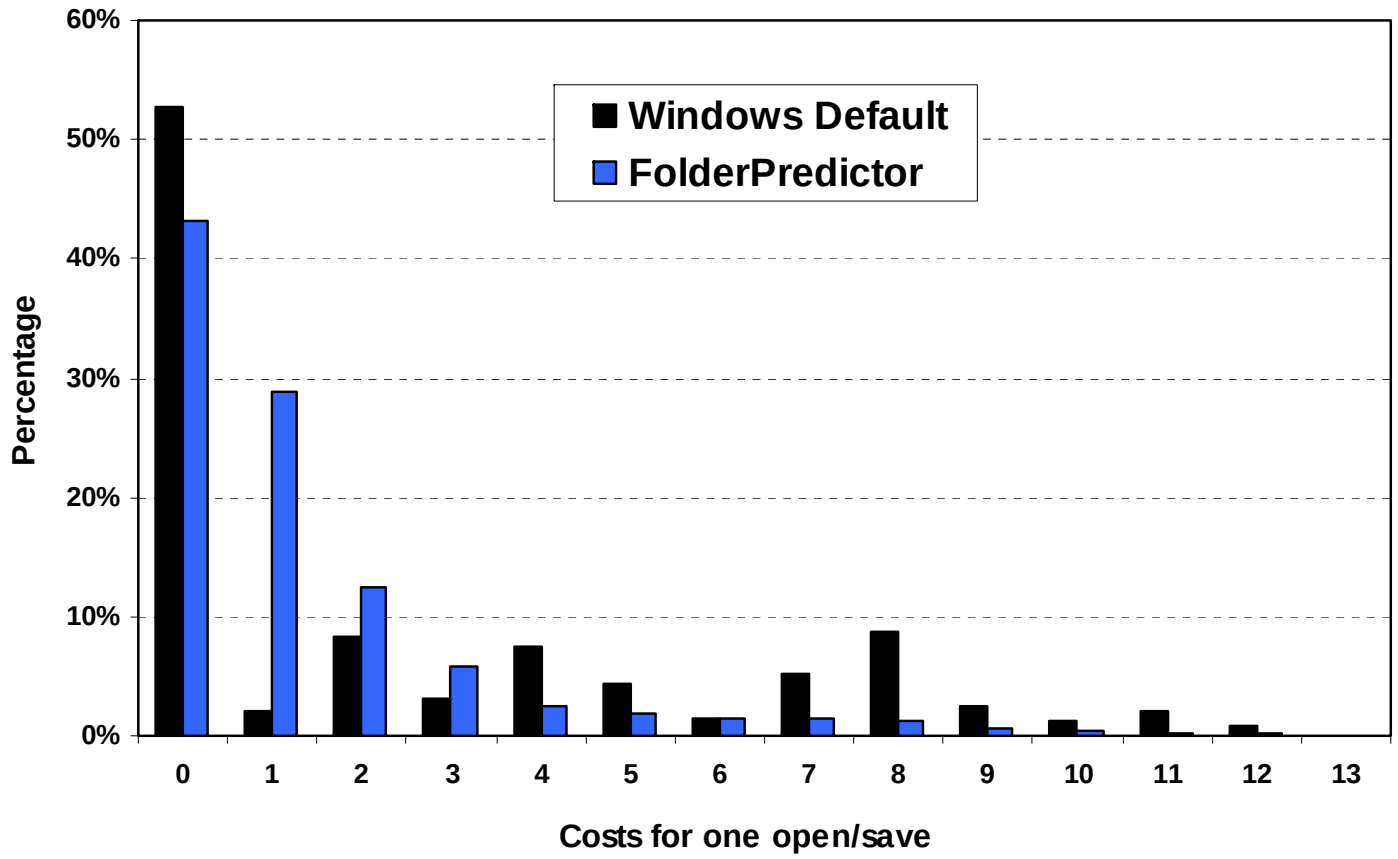
- Discount Factor $\gamma = 0.85$

TASK TRACER

Average Cost to Reach Target Folder

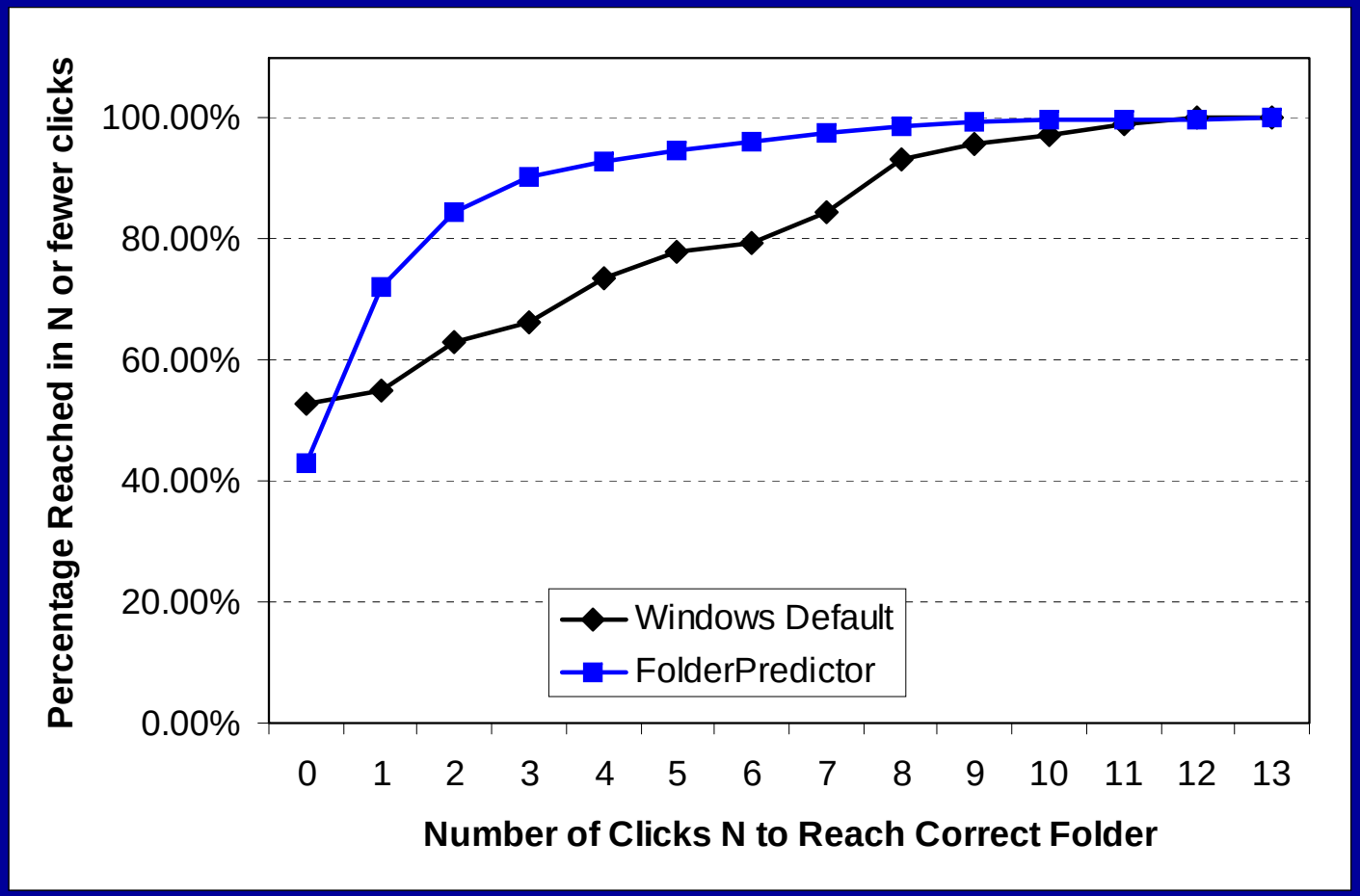


Results: Many Fewer Clicks Required



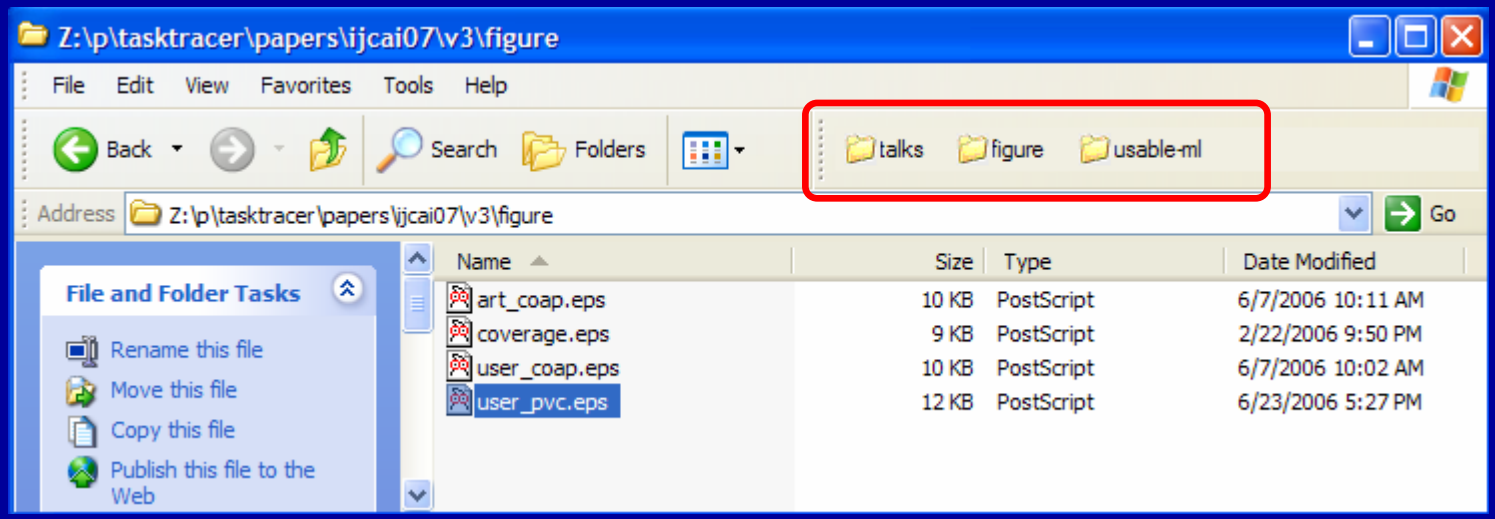
TASK TRACER

Cumulative Clicks Required



TASK TRACER

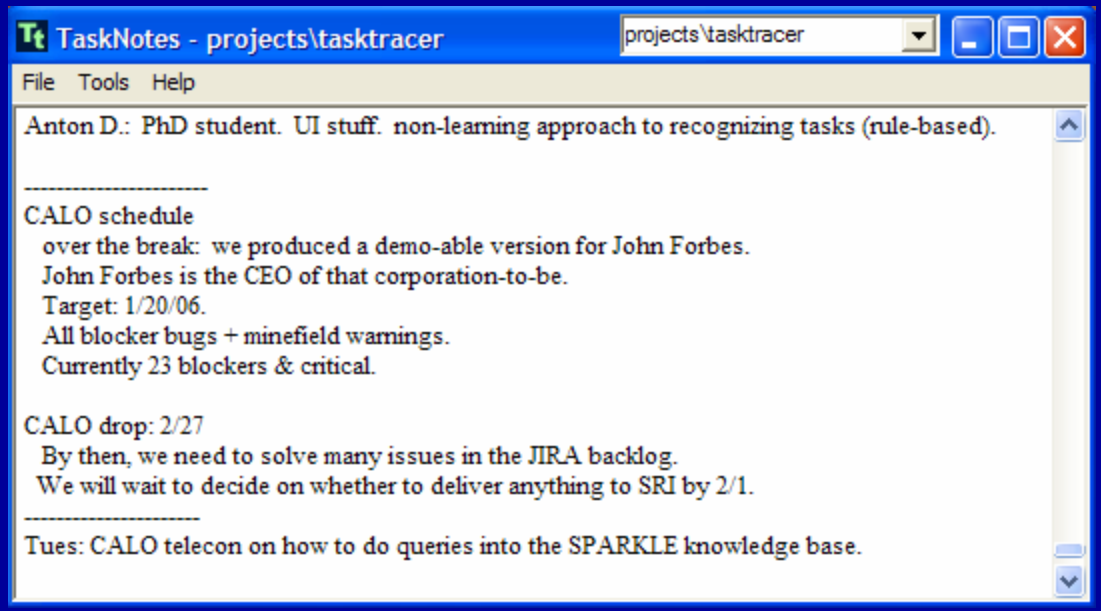
Folder Predictor Toolbar in Windows Explorer



TASK TRACER

Task Notes

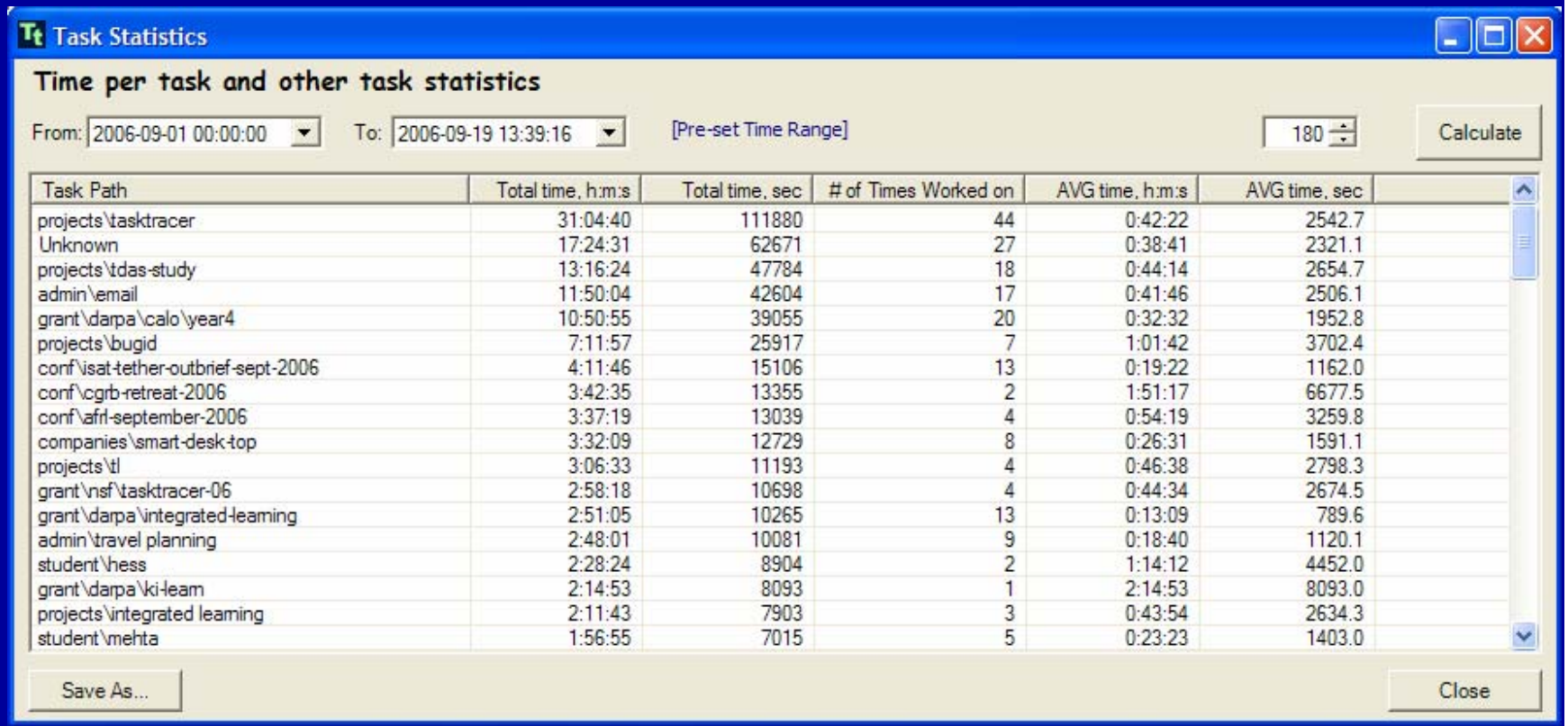
- Notepad associated with the current task
- Time stamp automatically inserted each time you change tasks



TASK TRACER

Time Tracking

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



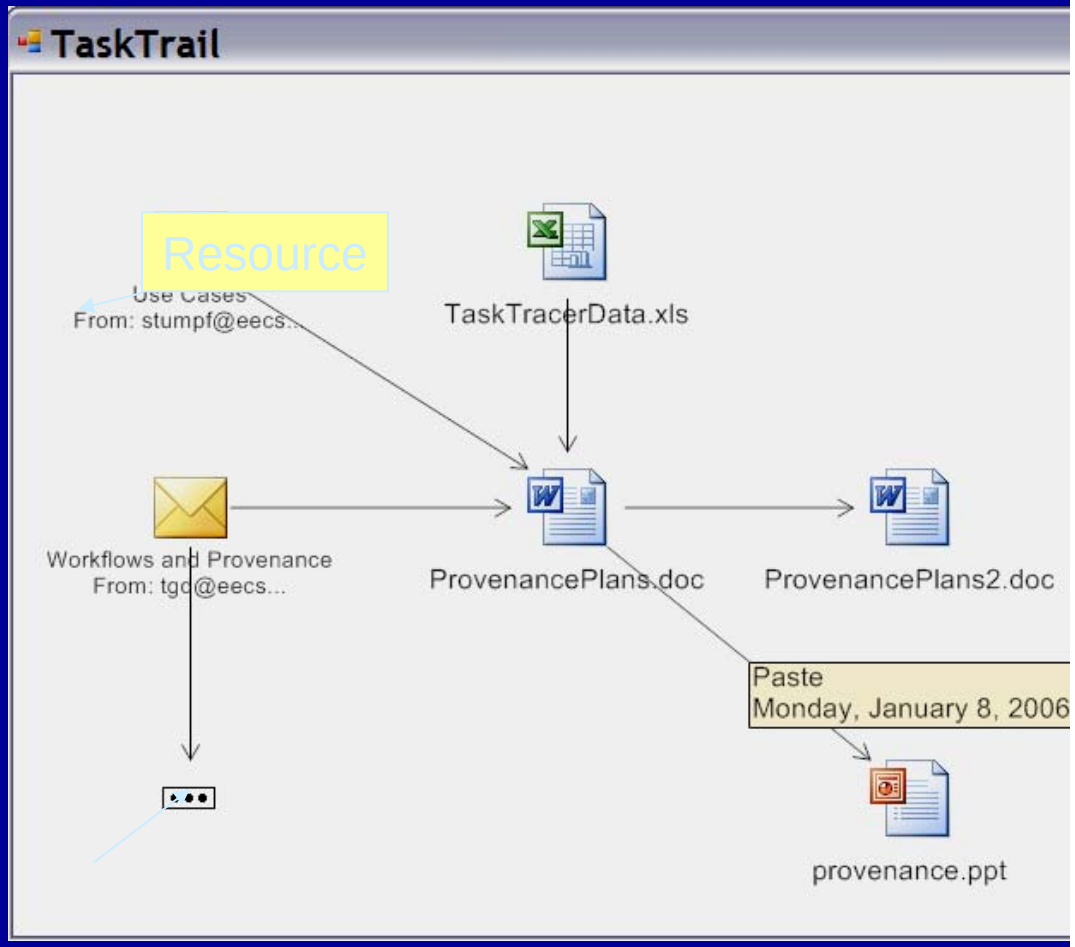
Task Path	Total time, h:m:s	Total time, sec	# of Times Worked on	AVG time, h:m:s	AVG time, sec
projects\tasktracer	31:04:40	111880	44	0:42:22	2542.7
Unknown	17:24:31	62671	27	0:38:41	2321.1
projects\tdas-study	13:16:24	47784	18	0:44:14	2654.7
admin\email	11:50:04	42604	17	0:41:46	2506.1
grant\darpa\calo\year4	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\afrl-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

Future Work

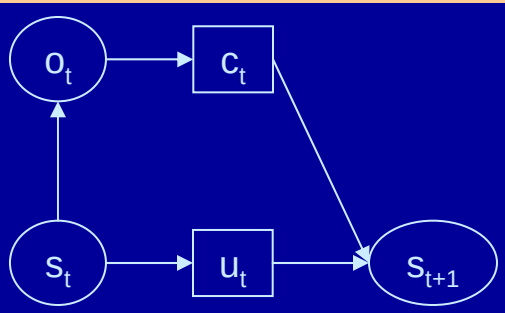
- Provenance-Based Information Access
- Activity Recognition and Proactive Assistance
- Combining Logical and Probabilistic Reasoning

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



Computer Assistant

- 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(truth assignment) =

  1/Z exp[$\sum$ weight of satisfied formulas]
- Inference
  - Find most likely truth assignment

    Weighted Max Satisfiability
  - Compute probability of a ground formula or ground literatal

    Markov Chain Monte Carlo (MCMC) method based on slice sampling (Gibbs Sampling)

TASK TRACER

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