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.
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”
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”
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…”
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.”
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”
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
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)

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

\[ P(\text{task} | X) = \alpha P(x_1 | \text{task}) P(x_2 | \text{task}) \cdots P(x_n | \text{task}) \ P(\text{task}) \]
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

<table>
<thead>
<tr>
<th>Subjects:</th>
<th>FA</th>
<th>RA</th>
<th>RB</th>
<th>SA</th>
<th>SB</th>
<th>SC</th>
<th>SD</th>
<th>SE</th>
</tr>
</thead>
<tbody>
<tr>
<td># messages</td>
<td>459</td>
<td>416</td>
<td>244</td>
<td>289</td>
<td>869</td>
<td>243</td>
<td>458</td>
<td>305</td>
</tr>
<tr>
<td># tasks</td>
<td>21</td>
<td>23</td>
<td>12</td>
<td>9</td>
<td>8</td>
<td>14</td>
<td>5</td>
<td>15</td>
</tr>
<tr>
<td># features</td>
<td>934</td>
<td>721</td>
<td>379</td>
<td>613</td>
<td>1158</td>
<td>598</td>
<td>448</td>
<td>349</td>
</tr>
</tbody>
</table>
Results
Task Predictor

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

```
<table>
<thead>
<tr>
<th>word: file1.doc</th>
<th>word: file1-v2.doc</th>
<th>excel: budget.xls</th>
</tr>
</thead>
<tbody>
<tr>
<td>wds1</td>
<td>wds2</td>
<td>wds3</td>
</tr>
</tbody>
</table>
```

SaveAs
WDS Features

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

* Data sets:

<table>
<thead>
<tr>
<th>Subjects:</th>
<th>FA</th>
<th>FB</th>
</tr>
</thead>
<tbody>
<tr>
<td># tasks</td>
<td>96</td>
<td>81</td>
</tr>
<tr>
<td># WDSs</td>
<td>5894</td>
<td>4151</td>
</tr>
<tr>
<td># features</td>
<td>1202</td>
<td>983</td>
</tr>
</tbody>
</table>
Task Predictor Results (FA)

Precision vs. Coverage for SVM, NB, and Hybrid models.
Task Predictor Results (FB)

![Graph showing Task Predictor Results for SVM, NB, and Hybrid methods. The x-axis represents Coverage, ranging from 0.1 to 0.7, and the y-axis represents Precision, ranging from 1 to 0. The graph compares the performance of SVM, NB, and Hybrid methods across different coverage levels.]
Combining Multiple WDS Predictions: Hidden Markov Model

\[ P(WDS_t \mid task_t): \text{Naïve Bayes model} \]
\[ P(task_{t+1} \mid task_t): \text{assume fixed probability of task switch} \]
Combining Multiple WDS Predictions

- Consider a sequence of task predictions:
  \[ P(\text{task}_1 \mid \text{WDS}_1), P(\text{task}_2 \mid \text{WDS}_2), \ldots \]
  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
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

\[
\arg\min_{\{f_1, f_2, f_3\}} \sum_f P(f \mid \text{task}) \cdot \min \{\text{clicks}(f_1, f), 1 + \text{clicks}(f_2, f), 1 + \text{clicks}(f_3, f)\}
\]
Experiment

- Data Sets:

<table>
<thead>
<tr>
<th>#</th>
<th>User Type</th>
<th>Data Collection Time</th>
<th>Set Size</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Professor</td>
<td>12 months</td>
<td>1748</td>
</tr>
<tr>
<td>2</td>
<td>Professor</td>
<td>4 months</td>
<td>506</td>
</tr>
<tr>
<td>3</td>
<td>Graduate Student</td>
<td>7 months</td>
<td>577</td>
</tr>
<tr>
<td>4</td>
<td>Graduate Student</td>
<td>6 months</td>
<td>397</td>
</tr>
</tbody>
</table>

- Discount Factor $\gamma = 0.85$
Average Cost to Reach Target Folder

![Average Cost of FolderPredictor and Windows Default][1]

1. TASK TRACER

---

[1]: Average_Cost_of_FolderPredictor_and_Windows_Default.png
Results: Many Fewer Clicks Required
Cumulative Clicks Required

![Graph showing the cumulative number of clicks required and the percentage reached in N or fewer clicks. The x-axis represents the number of clicks N to reach the correct folder, ranging from 0 to 13. The y-axis represents the percentage reached, ranging from 0.00% to 100.00%. Two lines are shown: one for the Windows Default and another for the FolderPredictor. The percentage increases as the number of clicks increases.]
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.

### Time per task and other task statistics

<table>
<thead>
<tr>
<th>Task Path</th>
<th>Total time, h:m:s</th>
<th>Total time, sec</th>
<th># of Times Worked on</th>
<th>AVG time, h:m:s</th>
<th>AVG time, sec</th>
</tr>
</thead>
<tbody>
<tr>
<td>projects\tasktracer</td>
<td>31:04:40</td>
<td>111880</td>
<td>44</td>
<td>0:42:22</td>
<td>2542.7</td>
</tr>
<tr>
<td>Unknown</td>
<td>17:24:31</td>
<td>52671</td>
<td>27</td>
<td>0:38:41</td>
<td>2321.1</td>
</tr>
<tr>
<td>projects\tdas-study</td>
<td>13:16:24</td>
<td>47784</td>
<td>18</td>
<td>0:44:14</td>
<td>2654.7</td>
</tr>
<tr>
<td>admin\email</td>
<td>11:50:04</td>
<td>42604</td>
<td>17</td>
<td>0:41:46</td>
<td>2506.1</td>
</tr>
<tr>
<td>grant\darpa\colo\year4</td>
<td>10:50:55</td>
<td>39055</td>
<td>20</td>
<td>0:32:32</td>
<td>1952.8</td>
</tr>
<tr>
<td>projects\bugid</td>
<td>7:11:57</td>
<td>25917</td>
<td>7</td>
<td>1:01:42</td>
<td>3702.4</td>
</tr>
<tr>
<td>conf\usat-tether-outbrief-sept-2006</td>
<td>4:11:46</td>
<td>15106</td>
<td>13</td>
<td>0:19:22</td>
<td>1162.0</td>
</tr>
<tr>
<td>conf\cgrb-retreat-2006</td>
<td>3:42:35</td>
<td>13355</td>
<td>2</td>
<td>0:51:17</td>
<td>6677.5</td>
</tr>
<tr>
<td>conf\aifl-september-2006</td>
<td>3:37:19</td>
<td>13039</td>
<td>4</td>
<td>0:54:19</td>
<td>3259.8</td>
</tr>
<tr>
<td>companies\smart-desk-top</td>
<td>3:32.09</td>
<td>12729</td>
<td>8</td>
<td>0:26:31</td>
<td>1591.1</td>
</tr>
<tr>
<td>projects\tl</td>
<td>3:06:33</td>
<td>11133</td>
<td>4</td>
<td>0:46:38</td>
<td>2793.3</td>
</tr>
<tr>
<td>grant\vista\tasktracer-05</td>
<td>2:53:18</td>
<td>10638</td>
<td>4</td>
<td>0:44:34</td>
<td>2674.5</td>
</tr>
<tr>
<td>grant\darpa\integrated-learning</td>
<td>2:51:05</td>
<td>10255</td>
<td>13</td>
<td>0:13:09</td>
<td>789.6</td>
</tr>
<tr>
<td>admin\level-planning</td>
<td>2:43:01</td>
<td>10001</td>
<td>9</td>
<td>0:18:40</td>
<td>1120.1</td>
</tr>
<tr>
<td>student\hess</td>
<td>2:28:24</td>
<td>8904</td>
<td>2</td>
<td>1:14:12</td>
<td>4452.0</td>
</tr>
<tr>
<td>grant\darpa\sleetem</td>
<td>2:14:53</td>
<td>8093</td>
<td>1</td>
<td>2:14:53</td>
<td>8093.0</td>
</tr>
<tr>
<td>projects\integrated-learning</td>
<td>2:11:43</td>
<td>7903</td>
<td>3</td>
<td>0:43:54</td>
<td>2634.3</td>
</tr>
<tr>
<td>student\mehta</td>
<td>1:58:55</td>
<td>7015</td>
<td>5</td>
<td>0:23:23</td>
<td>1403.0</td>
</tr>
</tbody>
</table>
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