Leveraging Characteristics of Task Structure to Predict the Cost of Interruption

Shamsi T. Iqbal and Brian P. Bailey
Department of Computer Science
University of Illinois, Urbana, IL 61820
{siqbal,bpbailey}@uiuc.edu

ABSTRACT

A challenge in building interruption reasoning systems is to compute an accurate cost of interruption (COI). Prior work has used non task-specific cues to predict COI, but ignore characteristics related to the *structure* of a task. This work investigates how well characteristics of task structure can predict COI, as objectively measured by resumption lag. In an experiment, users were interrupted during task execution at various subtask boundaries to collect a large sample of resumption lag values. Statistical methods were employed to create a parsimonious model that uses characteristics of task structure to predict COI. A subsequent experiment with different tasks showed that the model can predict COI with reasonably high accuracy. Our model can be expediently applied to many goal-directed tasks, allowing systems to make more effective decisions about when to interrupt.

KEYWORDS

Attention, Interruption, Learning, Task Models, Workload.

CATEGORIES AND SUBJECT DESCRIPTORS

H.1.2 [Models and Principles]: User/Machine Systems – Human Information Processing

INTRODUCTION

As users increasingly multi-task among proactive systems, their tasks are being interrupted more often [10, 18, 22, 27]. Though proactive delivery of information can benefit users, studies show that interrupting primary tasks can negatively impact productivity [4, 7, 9, 29] and affective state [1, 39].

To enable users to maintain benefits of proactive systems while mitigating these interruption costs, systems are being developed that can reason about appropriate moments for interruption [14, 15]. To make effective decisions, systems must be able to accurately predict the cost of interruption (COI). Systems currently predict COI using cues related to desktop activity, scheduled user activities, and visual and acoustical analysis of the task environment [13-15, 17, 19].

Systems could predict a more accurate COI if they also considered characteristics related to the structure of a task. *Task structure* refers to the subtasks and boundaries within a task decomposition [5]. *Characteristics* of task structure include depth of decomposition, types of subtasks, mental carryover, etc. Consideration of task structure is imperative since it can affect workload [5], which affects the COI [37]. Specifically, systems need to consider *subtask boundaries* since they are posited to reflect lower workload [28], which typically results in lower COI [37], and since boundaries are present in almost every goal-directed task [5].

Our prior work empirically demonstrated that interrupting at subtask boundaries results in much lower COI than non-boundary moments [4, 21]. We also found that interrupting at boundaries with *lower* workload results in meaningfully lower COI than at boundaries with *higher* workload. However, differentiating among subtask boundaries based on workload required the use of a physiological measure. This process was overly laborious and required access to specialized hardware. However, analysis of the data hinted at a possible alternative: leverage characteristics related to task structure to predict workload (COI) at boundaries.

This work investigates how well characteristics of a task's structure can predict the COI at subtask boundaries. In an experiment, users performed representative primary tasks and were interrupted at various boundaries with peripheral tasks. Resumption lag, time to resume primary tasks after an interruption, was used to provide ground truth for COI.

From a candidate set of characteristics, stepwise regression was applied to identify the best predictors of resumption lag (COI). Resumption lag values were then clustered into three classes to allow better interpretation of the COI. A multi-layer perceptron (MLP) was constructed to learn a mapping from the predictors to the COI classes. The generalizability of our COI model, consisting of the MLP plus heuristics for assigning its inputs, was evaluated in a subsequent user experiment using a *different* set of primary tasks and achieved reasonably high classification accuracy.

The benefit of our COI model is that it can be expediently applied to approximate COI at subtask boundaries in many goal-directed tasks. Having access to these values would allow interruption reasoning systems to differentiate among boundaries, absent the use of a physiological measure of workload. Our model could be extended to parts of a task other than boundaries as well as integrated into frameworks that also consider social and environmental cues [15, 19].

RELATED WORK

We situate our work within broader strategies for managing interruption, review empirical costs of interruption to justify our direction of work, discuss models for predicting the COI, and relate characteristics of task structure to the COI.

Strategies for Managing Interruption

There are four known strategies for managing interruption [26]; immediate, scheduled (defined intervals), negotiated (user determined), and mediated (third party decides). Our work contributes to the mediated strategy, where a system attempts to determine low cost moments for interruption. Mediated strategies can be used to meaningfully mitigate the COI [4, 26] and the ability to implement such strategies within computational systems has been demonstrated [16].

Empirical Cost of Interruption

Studies show that interrupting tasks at random moments can cause users to take up to 30% longer to resume the tasks, commit up to twice the errors, and experience up to twice the negative affect than when interrupted at boundaries [1, 4, 21]. Other studies show similar results [2, 8, 26, 39] and the differences in COI are typically attributed to differences in workload at the point of interruption [4]. Field studies also show that interruptions similar to those typically used in controlled studies are common in practice [10, 22].

It is important to mitigate these costs, as response delays or errors committed due to interruption can cost human life in safety critical domains [27], and unnecessary increases in negative affect degrades the user experience in others [34].

Interruption reasoning systems seek to deliver information when the costs would be low. To achieve this, systems must have an accurate model of COI during task execution. Our work contributes such a model, based on resumption lag.

Characteristics of Task Structure and COI

A model for the COI should consider task structure, as task structure generally affects mental workload [5], which affects COI [37]. Prior work has sought to further elucidate the relationship between characteristics of task structure and the COI. For example, Monk et al. [29] and Czerwinski et al. [9] decomposed a task into temporal phases and found that interrupting during earlier phases had lower cost.

Our work focuses specifically on one component of task structure - subtask boundaries. This is because systems can detect them [3] and because they are present in almost every goal-directed task [5]. Most importantly, our prior work showed that interrupting at subtask boundaries results in much lower COI than at non-boundary points [4, 21]. We also found that interrupting at subtask boundaries with lower mental workload results in meaningfully lower COI than interrupting at boundaries with higher workload.

However, a limitation of our approach was that it required the use of a physiological measure to differentiate among subtask boundaries. To overcome this limitation, analysis of the data suggested that certain characteristics of boundaries, e.g., level in the task model, could be used to predict workload (COI), *absent the use of a physiological measure*.

This work further investigates which characteristics of a task's structure best predict the COI at boundaries and the prediction accuracy of the resulting COI model.

Predicting the Cost of Interruption

The foremost method for predicting the COI is to build a probabilistic model that uses input cues related to desktop activity, visual and acoustical analysis of the physical task environment, and scheduled activities of the user. To train such a model, ground truth for the COI may be defined by users [15, 19] or determined empirically [14]. For example, Fogarty et al. built a statistical model that maps interface events (typing, scrolling, navigating, etc.) to one of three classes of task engagement (COI), where ground truth was determined using response time to a secondary task [14].

Our method of statistical modeling follows prior work, but our work differs in that we are using characteristics related to the *structure* of a task and are using resumption lag as the ground truth for COI in the model building process. Our selection of resumption lag is important, as it provides a direct, empirical cost of interruption. Our model can be used alone or can complement the use of existing models to predict COI more accurately than either could predict alone.

EXPERIMENT 1: COLLECTING COI DATA

The purpose of our first experiment was to collect a sample of COI data. This was achieved by having users perform primary tasks, interrupting the tasks at various boundaries with peripheral tasks, and measuring the resumption lag.

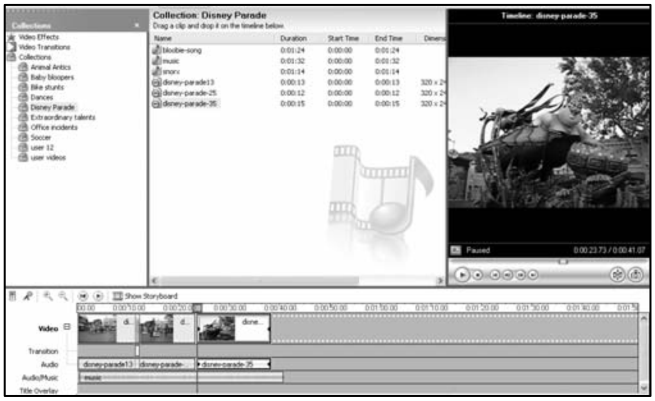
Users

12 users (7 female) participated in the study. Users ranged from 23 to 33 years of age (M=26.33, SD=2.839). Users were compensated with a $5 coupon to a local coffee shop.

Primary Tasks and Models

Three categories of primary tasks were developed:

- *Video Editing*. As shown in Figure 1, users were asked to use Windows MovieMaker to compose a short (~ 1 min) digital video from provided clips, each about 15-20s. Themes of clips included Disney parade, animal antics, soccer highlights, baby bloopers and bicycle stunts. The user reviewed the clips, added a subset of the clips to the editing timeline and edited length/content as desired. Any visual transitions of their choice were then added between clips. Next, the user reviewed provided audio tracks and added the desired track to the video, and then compiled



Collection: Disney Parade

Drag a clip and drop it on the timeline below

Name	Duration	Start Time	End Time	Dimens
bloobie-song	0:01:24	0:00:00	0:01:24	
music	0:01:32	0:00:00	0:01:32	
snorx	0:01:14	0:00:00	0:01:14	
disney-parade13	0:00:13	0:00:00	0:00:13	320 x 2
disney-parade-25	0:00:12	0:00:00	0:00:12	320 x 2
disney-parade-35	0:00:15	0:00:00	0:00:15	320 x 2

Figure 1: The video editing task. A user creates a short digital video by composing and editing provided clips, inserting transitions between clips, and adding a suitable audio track.

and saved the final video. Users were encouraged to be as creative as possible while still following instructions.

- *Route Planning*. An interactive map with two routes connecting two cities was displayed along with two tables. Each route had three segments and each segment had distance and fare data associated with it, displayed in a tooltip balloon. For the task, the user moved the cursor over a route segment, retrieved the distance and fare data, entered it into the corresponding row in the table, and repeated for the other two segments. The distance and fare columns were then mentally added and the result was entered into the last row. The user repeated this process for the second route and table and then selected the shorter and the cheaper routes from drop down lists.
- *Document Editing.* Users edited a manuscript annotated with three comments that varied in the complexity of the edit required. Content included contemporary topics such as global warming, legal issues regarding digital media, endangered species, the education system in the U.S., etc. We felt these topics would be interesting and familiar to users. The user edited the document according to each comment, stored in a tooltip. After reading a comment, the user located the text, made the appropriate edit, and repeated this process twice more. Once edited, the user saved the document with a name of their choice.

These tasks were designed to be engaging as well as to have meaningful subtasks requiring varying mental effort, salient boundaries between the subtasks, and largely prescribed execution sequences. The latter constraint was necessary to be able to interrupt task execution at specific points for data collection. Each task lasted about 5-6 minutes. The latter two tasks were adapted from our prior work [20].

Since a within-subjects design was used, multiple instances of each task were created and we were careful to alter just the content, not the basic execution structure of the tasks. For example, video editing used different video and audio clips, document editing used different content, and route planning used different city names and route data.

To define the structural characteristics of the tasks, GOMS models were developed, one per category (see Figure 2 for the task model for video editing). Following [5], initial models were built based on our own understanding of the task's execution. The models were iteratively refined by having users (in a pilot study) perform the tasks and matching the models to the observed execution sequences. This continued until the models achieved high accuracy.

In developing the task models, we tried to balance having enough detail to identify lower-level boundaries with being able to allow for variability in the execution sequences. For example, the model in Figure 2 includes a subtask for insert transition at level 3, but whether a user drags or copies and pastes a transition to the timeline is not explicitly modeled. This allowed us to model the adjacent boundaries yet still capture some variability. We found that decomposing a task to about 3-5 levels typically achieved the desired balance.

The final models were evaluated against the interaction sequences from the actual study. Each model achieved more than 90% accuracy with no obvious patterns in the errors. Models for route planning and document editing were adapted from our prior work [20] while the model for video editing was newly constructed for this work.

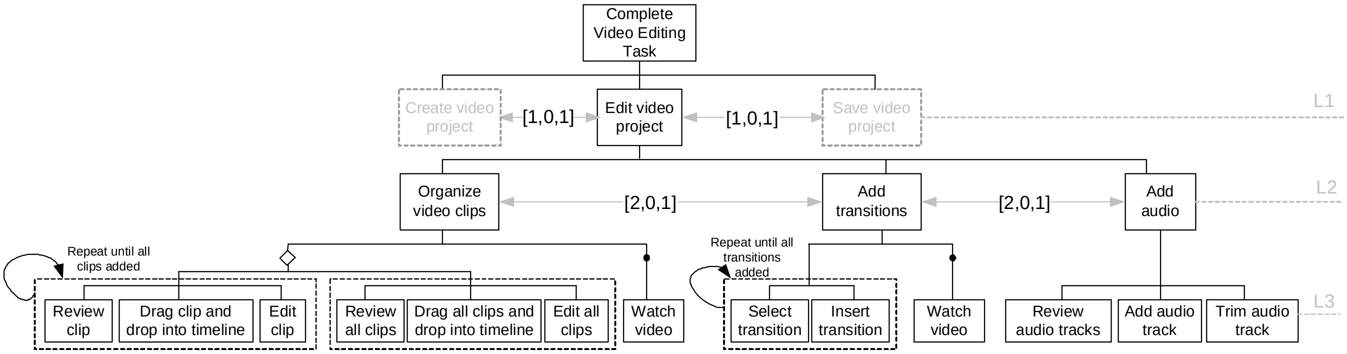


Figure 2: Part of the GOMS model for the video editing task, showing details for the Edit Video subtask. Time moves from left to right. The models were developed to strike a balance between having an appropriate level of detail and allowing for variability in execution sequences. Diamonds indicate common alternative sequences that were explicitly modeled and solid dots indicate optional subtasks. Values for predictors ([*level, carryover, difficulty of next subtask*]) are shown at a sample of the subtask boundaries.

Peripheral Tasks

A realistic stock scenario was presented, adapted from prior work [4]. Each scenario consisted of a fictitious company's name along with the quantity, date, and price of shares that the user hypothetically purchased from that company. Each scenario also contained the price of the stock and a one sentence "news-flash" about the company. After analyzing the scenario, the user selected one of five trading actions; do nothing, buy a few more shares, buy many more shares, sell a few shares or sell all shares. Multiple instances of the task were created and each required about 20s to perform.

This task was used because it is representative of peripheral information that users often receive [24, 25] and because analyzing the scenarios taps cognitive resources [37].

Moments for Interruption

For each task, we selected a sample of ten representative subtask boundaries from the corresponding GOMS model. For example, for video editing, boundaries included the point after dragging a clip and dropping it on the timeline, but before making any edits; after making the last edit on the timeline, but before adding transitions; after completing the video editing but before saving it, etc. The set of selected boundaries sampled different levels and temporal positions in the task model. Boundaries for the other tasks were selected using a similar strategy.

Experimental Setup

A Wizard of Oz technique was used to time delivery of the peripheral task. The experimenter monitored a user's task execution using a Real VNC client and delivered peripheral tasks at the selected boundaries using a remote command.

For each selected boundary, the experimenter waited for the user to make a directed action signifying the start of the subsequent subtask, based on the task model. This method mimicked how systems may identify boundaries in practice [3]. Since a high speed LAN connection was used, there was negligible latency from when the peripheral task was commanded to when it actually appeared on a user's screen.

Procedure

Upon arrival at the lab, a user went through an informed consent process and received general instructions for the study. Since a within-subjects design was used, the primary task categories were presented using a Latin Square design.

For each category, the user received specific instructions, performed a practice and then performed the actual trials. A user performed five task trials and was interrupted twice during each trial. Interruption moments were randomly selected from the defined set of ten, without replacement. The peripheral task was presented in a modal window and covered the main work area, prompting a task switch at the defined moment. Users were asked to begin the peripheral task as soon as it appeared and, once complete, to resume the primary task as quickly as possible. This process was repeated for each task category. The study lasted 90 min.

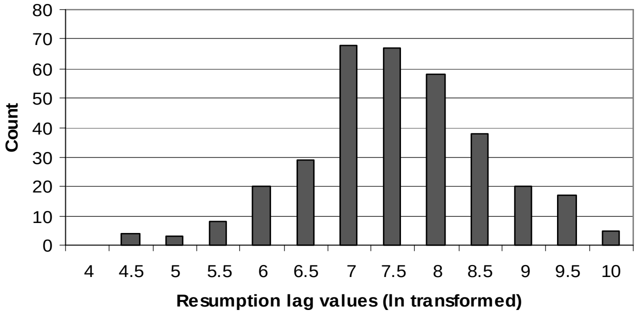


Figure 3: Histogram of the transformed resumption lag data. The distribution is near normal, with more values in the middle and fewer at either end of the scale. Each bar represents the number of values that fall between the value corresponding to the previous bar and itself.

Measurements

Resumption lag was used as the cost of interruption. It was measured as the time difference from when the peripheral task window was closed to when the user made the first observable action in the resumed primary task [2].

Other measures could have included error rate and affective state. Errors were not used because they would be difficult to judge for creative tasks such as video editing and it is unclear what time window should be used for attributing errors to the interruptions. Affective state was not used since this is often measured using a subjective rating, which would likely change based on the interruption's content. Resumption lag is objective, continuous, and well defined.

RESULTS AND DEVELOPMENT OF THE COI MODEL

A total of 360 resumption lag samples were collected. To normalize the resumption lag data, a natural log transform was applied, which is common for performance data. Outliers and data values corresponding to errors (e.g., the experimenter missed a boundary due to the user deviating from the model) were removed from the data (~ 6%). This left a total of 337 samples in the data set. As shown in Figure 3, the resulting transformed data set ranged from 4.10 to 9.86 (M=7.3, SD=1.04). These results are consistent with resumption lag data reported in prior work [2, 21, 36].

With this transformed data, we proceeded to building the COI model. This consisted of identifying candidate factors, using stepwise regression to determine the most predictive ones, clustering the data into discrete classes, and learning a mapping from the predictive factors to those classes.

Identify Candidate Factors

The first step was to propose a candidate set of structural characteristics related to boundaries. Based on prior work and our own experience, we identified these factors:

- *Level*. The level of a subtask boundary is defined as (1 +) the depth of the shared ancestor of the adjacent subtasks.

Difficulty	Category	Example
1 (Least)	Motor movements	Move mouse towards a menu item or select a menu item
2	Easy content generation	Enter a new filename or select a transition for a video clip
3	Comprehension or store information in memory	Read text or comments, retrieve a route segment's distance and fare information and commit it to memory
4	Recall information	Recall a route segment's distance and fare information
5	Creative content generation	Edit document text or edit video clips
6 (Most)	Mathematical reasoning	Add distance or fare information

Table 1: The six levels of subtask difficulty in our tasks, their corresponding categories, and examples of each.

This factor was selected because prior work has shown that level affects how much mental workload decreases at boundaries [20] and that interrupting at boundaries at different levels in the task model influences COI [21].

- *Presence of a visual resumption cue*. This factor was a binary value (0=no cue) indicating whether the state of the primary task at the point of interruption provides an obvious visual cue for resuming it. The *presence* (not saliency) of cues has been thought to reduce COI [2, 6].
- *Percent of task complete*. This refers to how much of the overall task is complete at the boundary. To provide an accurate value, we timed a few users performing the tasks and then mapped the percent complete to each boundary in the model. Temporal position in a task (e.g., beginning, middle, end) has been shown to affect COI [9, 28, 29].
- *Percent of parent subtask complete*. This factor was similar to the previous one, except that percent complete was now measured relative to the parent of the subtasks adjacent to the boundary. This factor was considered because users often chunk execution of tasks [30].
- *Difficulty of preceding subtask*. There is no standard method for computing the difficulty of *subtasks*. We thus adapted a heuristic often used to approximate difficulty when predicting resource conflicts between tasks [37]. The leaf subtasks (operators) were categorized based on presumed difficulty and the categories were qualitatively ordered based on their presumed cognitive demands. As shown in Table 1, this produced 6 categories, with '1' being least demanding. For example, a mouse movement was assigned 1 whereas mental calculation was assigned 6. If the preceding subtask was a goal subtask, the difficulty of its last operator was used. For example, the boundary between edit video project and save video project in the video editing task was assigned 1, as this was the difficulty of the last operator (trim audio track). Difficulty of preceding subtask was considered because COI is thought to depend on a user's mental workload at the point of interruption [4].
- *Difficulty of next subtask*. This factor was included for the same reason as the previous one and its value was computed analogously. If the next subtask was a goal subtask, the difficulty of its first operator was used.
- *Carry over at boundaries*. This factor refers to how much data must be maintained across a boundary. Similar to difficulty, we categorized boundaries based on presumed carryover, resulting in four categories, and qualitatively ordered them by assigning values of 0 (no carryover) to 3 (high carryover). For example, maintaining a seven digit value across a boundary in route planning was assigned 3, while retaining where to position a clip in a video after selection was assigned 1. We included this factor since it provided another estimate of workload at a boundary.

These values were computed for each boundary in the task models. Though additional characteristics could have been included, we restricted this first set to those identified in prior work and that could be computed relatively easily.

Determine the Most Predictive Factors

The next step was to determine which of the candidate factors were the most predictive of resumption lag. The technique employed was stepwise multiple regression.

We first checked the global utility of the regression model. A multiple regression analysis with Resumption Lag as the dependent variable and all candidate factors as independent variables was performed. The linear regression model was predictive (F(12,336)=11.23, p<0.0001, adjusted R^2=0.25) and the residuals of the regression model met the normality assumption. Passing the global utility test strongly suggests that at least one of the candidate factors has a non-zero coefficient and is predictive of resumption lag.

To create a parsimonious model (the least number of factors that explain as much of the variance in the data as possible), a stepwise model building technique was employed. As summarized in Table 2, this technique showed that Level, Carry Over, and Difficulty of Next Subtask were the most

Model	*β*	Std Err	*t*	*p*
Constant	6.197	0.138	44.92	0.0001
Level	0.38	0.068	5.581	0.0001
Carry Over	0.158	0.067	2.351	0.019
Difficulty	0.077	0.038	2.063	0.040

Table 2: Regression model with the three predictive factors.

predictive of Resumption Lag, with adjusted $R^2 = 0.26$. This means that 26% of the variance in Resumption Lag can be explained by these three characteristics alone, a very positive result given the innate complexities of the human information processing system [5].

When compared to the full model, the amount of variance explained changed little, yet the number of factors was reduced to three. Also, the reduction to these three particular factors suggests that COI depends more on the characteristics that reflect current and prospective allocation of mental resources (workload) than on those that reflect temporal position or cue availability.

An interruption reasoning system could use this model to predict resumption lag for each boundary in a task model. However, given the model's modest correlation coefficient, a challenge for systems is to interpret the meaningfulness of differences among predicted values (e.g., how much better is 7 than 7.5), as each prediction has error associated with it. Though a z-score accounts for variance in the data set, it does not account for the error in the predicted value itself.

Thus, we decided to cluster resumption lag into classes such that there was a meaningful difference between them. This would allow the model to be adapted to predict the *classes* rather than specific values, which would enable increased prediction accuracy, at the price of decreased sensitivity.

Determine Cost Classes

K-means cluster analysis was applied to the data. The goal was to identify the largest number of clusters such that meaningful differences would be maintained between them.

Based on several data visualization techniques, we found that about 3-5 clusters would be appropriate. Analyzing each number, we found the use of 3 clusters (COI_L, COI_M, and COI_H) to be most appropriate. With these clusters, 75 values fell into COI_L ($\underline{M}$=5.938, $\underline{SD}$=0.612), 177 values into COI_M ($\underline{M}$=7.252, $\underline{SD}$=0.376) and 85 values into COI_H ($\underline{M}$=8.628, $\underline{SD}$=0.505). An ANOVA showed that the means differed (p<.0001 between all pairs). Three clusters were the most that maintained these differences between pairs.

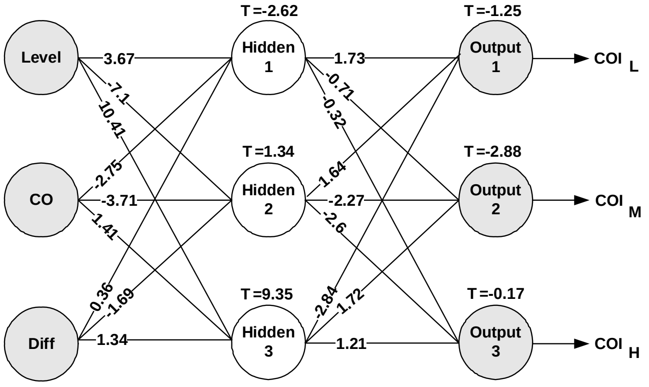


Figure 4: The MLP model that maps the predictors (Level, Carryover, and Difficulty of next subtask) to the COI classes. The input nodes are the predictors identified from the stepwise regression analysis while the output nodes correspond to the COI classes determined from the cluster analysis.

		Predicted Cost			
		COI_L	**COI_M**	**COI_H**	**Total**
Actual Cost	**COI_L**	42 (56%)	32 (42.7%)	1 (1.3%)	75 (100%)
	COI_M	24 (13.6%)	136 (76.8%)	17 (9.6%)	177 (100%)
	COI_H	4 (4.7%)	46 (54.1%)	35 (41.2%)	85 (100%)

Table 3: Distribution of predicted vs. actual COI classes for the model building tasks.

This is consistent with [14], where it was reported that a user's interruptibility could be best classified into at most 3 classes. Our result is an interesting parallel, as it suggests that, absent a physiological measure, a system may only be able to effectively classify the COI into at most 3 classes.

Learn a mapping from predictors to COI classes

Finally, we needed a mechanism to map from the predictors to these COI classes. Unfortunately, the regression equation could not be used since the constant term (6.197) was greater than the mean of COI_L (5.938), thus it could not always map predictors to this class. After analyzing several methods, we settled on the use of a multi-layer perceptron (MLP). Unlike a Naïve Bayes model, for example, an MLP model does not require the predictors to be independent.

Back propagation was used to learn an MLP model, with Level, Carry Over and Difficulty of Next Subtask as input. There was one hidden layer and three outputs, one for each COI class. Figure 4 shows the resulting MLP.

A 10-fold cross validation technique was used to evaluate the model. Table 3 shows the distribution of predicted vs. actual COI classes, where the diagonal represents correct predictions. The overall number of correct predictions was 63.2%, much better than chance (*N*(0.33, .000656)=24.67, p<.0001). Inspection of the table shows that the model is most accurate for COI_L and COI_M, but slightly less accurate for COI_H. However, the most egregious type of error, predicting COI_L when it is actually COI_H, is low (4.7%).

The next step was to evaluate how well the model predicted COI classes when applied to boundaries within tasks that are *different* from those used in the model building process.

EXPERIMENT 2: EVALUATING THE COI MODEL

A second experiment was conducted to evaluate how well our COI model (the MLP plus heuristics for assigning its inputs) predicted COI classes when applied to boundaries within different tasks. Specifically, we wanted to (i) evaluate the accuracy of the predicted COI classes; and (ii) test whether there are differences in resumption lag between predicted COI classes, which would validate that reasoning systems should integrate the use of our model.

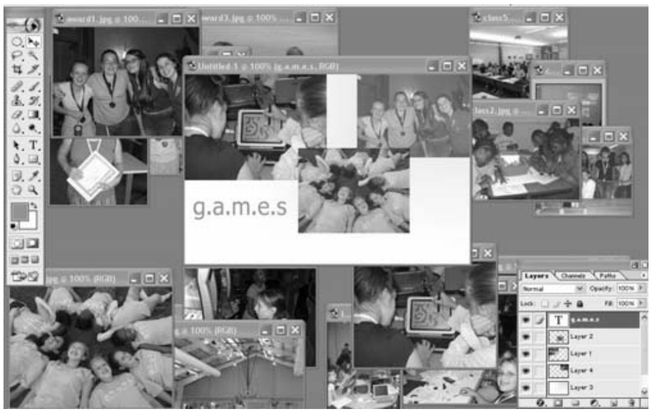


Figure 5: Collage generation task. A user created a collage by composing images from several categories depicting a certain theme. Users included at least one image from each category, manipulated the layers, and add visual effects to the collage.

Users

A different set of 12 users (2 female) participated in the study, with ages from 21 to 26 (M=24.2, SD=1.8). Users were compensated with a $5 coupon to a local coffee shop.

Tasks and Models

Two new tasks were developed for this experiment:

- *Collage Generation.* As shown in Figure 5, users were asked to create collages in Adobe Photoshop that would communicate a given theme. Themes included activities in amusement parks, life as a CS student, and experiences in summer camps. To foster engagement in the task, users were told that the collages would be used for marketing. For each theme, four categories, each with four images, were provided (e.g., outreach, research, campus, and fun for *life as a CS student*). To create the collage, users created a blank image with a specified width and height, and then opened all of the source images (16 in total) that could be used in the collage. Users were to select at least one image from each category, paste it into the collage, and size and position it as necessary. After integrating the images, users manipulated their layer ordering to create the desired look and added at least one visual effect (e.g., blurred edges around each layer) to the collage. Finally, they saved the collage in a directory with a desired name.
- *Electronic Form Design.* Users designed electronic forms using Adobe Designer. Forms included a registration form for an HCI workshop, customer satisfaction survey, and purchase order. Users were provided with a partially completed form and asked to complete it based on given requirements. For example, users were asked to construct fields for collecting payment information, feedback about customer service, and product information as efficiently as possible. Widgets such as text fields, check boxes, radio buttons, and drop-down lists were available for use. Users selected any widgets they felt were most suitable for collecting the needed information, placed them on the form, and added the appropriate text.

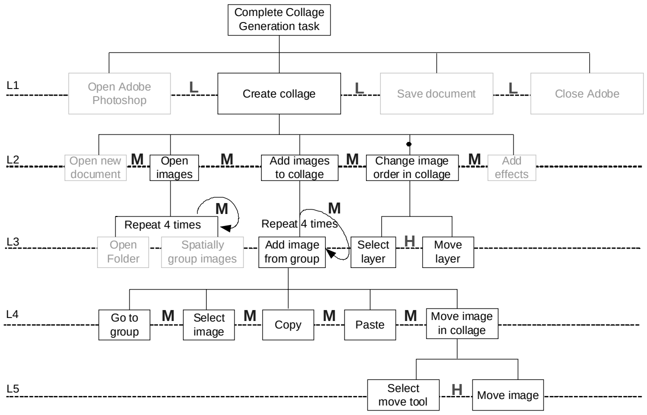


Figure 6: Part of the GOMS model for the collage generation task, showing details for just the Create Collage subtask. The predicted COI classes are shown at each subtask boundary.

Similar to the tasks used in the first experiment, these tasks were designed to be engaging and to have subtasks requiring varying mental effort, observable boundaries, and mostly prescribed sequences. Each task took about 5-6 minutes to perform. Task models were iteratively developed and validated using techniques as before. Error rates of the task models were low, consistent with the first experiment.

Predicted COI Classes and Moments for Interruption

We applied our COI model, consisting of the heuristics for assigning values to the predictors and the MLP shown in Figure 4, to predict the COI class at each subtask boundary. For each subtask boundary, we used our heuristics to assign values for the three predictors (Level, Carry Over, and Difficulty of Next Subtask). The values were then used as input to the MLP, which computed the predicted COI class.

Figure 6 shows part of the task model for collage generation with the predicted COI classes. Including both task models, there was a total of 38 subtask boundaries, of which 7 were assigned to COI_L, 26 to COI_M, and 5 to COI_H.

For the specific moments to interrupt, we randomly selected a sample of six boundaries from each task model, two from each of the three COI classes. The peripheral task, experimental setup and procedure, and resumption lag measurements were the same as in the first experiment.

RESULTS OF EXPERIMENT 2

A total of 144 data samples were collected. Prior to analysis, we filtered outliers and any data resulting from experimental errors, resulting in 7% of the values being removed. This left 134 samples in the data set. Once filtered, a natural log transform was applied to normalize the resumption lag data.

Compare Predicted to Actual COI Classes

The resumption lag values at each boundary were classified into their *actual* COI classes using the cluster information determined in Experiment 1.

		Predicted Cost			
		COI_L	COI_M	COI_H	Total
Actual Cost	COI_L	35 (55.56%)	19 (30.16%)	9 (14.29%)	63 (100)%
	COI_M	2 (4.26%)	23 (48.94%)	22 (46.81%)	47 (100)%
	COI_H	5 (20.83%)	6 (25%)	13 (54.17%)	24 (100)%

Table 4: Distribution of predicted vs. actual COI classes for the tasks in the second evaluation.

Table 4 shows the distribution of predicted vs. actual COI values. As in Table 3, values along the diagonal, where the predicted costs match the actual costs, were the highest – 56% for COL_L, 49% for COL_M, and 54% for COL_H. Our model correctly predicted 53% of COI values, much higher than chance ($N(0.33, 0.00165)=13.05$, $p<0.0001$).

The most egregious type of error (predicting COI_L when it is actually COI_H) was higher than for the model building tasks (20.8% vs. 4.7%), though it was still reasonably low overall. One plausible explanation is that there may have been increased mental workload across higher-level boundaries in these tasks. This would likely cause greater resumption lag, while the model would predict a lower cost.

The classification accuracy for COI_L (~56%) is identical to what was obtained for the model building tasks, while the classification accuracy for COI_M decreased from 77% to 49%, and accuracy for COI_H increased from 40% to 54%. As before, most errors were made to an adjacent class and fewer were made between COI_L and COI_H.

Though changes in the distribution occurred, they were not unexpected as our heuristics can only approximate values for the predictors of Carryover and Difficulty Next Subtask. The most important outcome, however, was that the overall accuracy and pattern of distribution was very similar to the model building tasks. This suggests that our COI model can be reasonably generalized to other goal-directed tasks.

Differences in Resumption Lag among Predicted COI

For this analysis, we grouped the resumption lag values by their *predicted* (not actual) COI values. An ANOVA showed that resumption lag was different among predicted COI classes ($F(2,131)=25.23$, $p<0.0001$). Post hoc tests showed that COI_H ($M=7.44$, $SD=0.92$) had greater resumption lag than COI_M ($M=6.92$, $SD=0.66$, $p<0.013$) and COI_L ($M=6.14$, $SD=0.97$, $p<0.0001$) and that COI_M had greater resumption lag than COI_L ($p<0.0001$). The means of each predicted COI class translates into 1702ms (COI_H), 1012ms (COI_M), and 464ms (COI_L) respectively, which represents meaningful differences for resumption lag data, especially when extrapolated over many interruptions.

Using predicted COI to group resumption lag values was important, as the results show that even with some errors, the model is accurate enough such that predicted values still correspond to empirical, meaningful differences in the cost of interruption. This validates that a system can and should use our model to differentiate among subtask boundaries, enabling more effective decisions about when to interrupt.

DISCUSSION

This research explored how well structural characteristics of a task could be used to differentiate COI among subtask boundaries. By employing a series of statistical methods, we showed that three characteristics of task structure (Level, Carryover, and Difficulty of Next Subtask) can be used to predict COI at boundaries with reasonably high accuracy. We also showed that our model's predictions of differences in COI correspond to differences in resumption lag.

Cognitive theory argues that lower COI should result when a primary task is interrupted at moments of lower workload, as fewer mental resources must be re-acquired to resume the task [37]. The efficacy of our model thus derives from its ability to capture the current (Level and Carryover) and prospective (Difficulty of Next Subtask) allocation of resources (workload) at subtask boundaries. The advantage of our model is that it can differentiate subtask boundaries based on workload, absent use of a physiological measure.

In the next sections, we briefly describe how the COI model could be used in practice and then discuss its limitations.

Applying the COI Model in Practice

We have recently developed a task framework that includes a language for describing tasks and a system for monitoring execution of those tasks [3]. Our COI model is intended to be used with this type of task monitoring framework.

The language allows the structure and execution sequences of a task to be concisely described, but in much less detail than those used for user simulation [32]. COI values can be assigned to any point in a description, including boundaries. To apply our COI model, a person computes values for the predictors (Level, Carryover, Difficulty of Next Subtask) at each boundary, inputs the values into the MLP (figure 4), and encodes the COI predictions within the description.

During task execution, interface events are matched to the task descriptions. When a user reaches a subtask boundary, as indicated in the description, the encoded COI value is retrieved and can be sent to a broader reasoning framework. The framework could then consider this value along with social and environmental cues to determine an overall COI.

The benefit of using our COI model and task framework is that it will enable reasoning systems to ground at least part of their COI prediction in cognitive theories of resource allocation related to task structure, which has not been directly considered in existing systems. This is important since resource allocation strongly influences the cognitive cost of interruption [37] and other types of task switching [33]. By considering this information, systems can make more effective decisions about when to interrupt, mitigating competition for resources and thus the COI.

Our current COI model would yield the most benefit if it were applied to high frequency, routine, or safety critical tasks, which often have prescribed execution sequences [11]. For tasks with less prescribed sequences, task models for significant variations could be created or learned and our COI model could be applied to each of them. Though this would require a large effort, it may soon be possible to develop or adapt tools to automate much of the process (e.g., see task modeling tools discussed in [12, 23, 35]).

Limitations

Our COI model currently considers only one component of task structure – subtask boundaries. This is because systems can detect them [3], the cost of interrupting at boundaries is lower than at other points [21], and they are present in almost every goal-directed task [5]. However, systems may also want to differentiate among non-boundary points, i.e., subtasks. For example, this could be useful when the temporal distance between boundaries is large. To allow differentiation among subtasks, our model building process could be utilized to extend the current COI model or create a complementary model for different types of subtasks.

The presence, location, or utility of boundaries may change as a user's knowledge of performing a task transitions from novel to skilled behavior. As a task becomes skilled, the mental representations are thought to become coarser [31], eliminating or reducing the utility of some boundaries. However, experimental studies have shown that familiarity with a task seems to have little effect on how users perceive its hierarchical structure [38], suggesting that the mental representations for tasks remain fairly stable. Still, skilled tasks are typically performed in larger chunks [35] and COI models should consider this effect. A possible solution is to extend our current COI model to include skill level as a predictor and to encode a COI value for each skill level at each boundary or other salient point in the task.

Our current COI model assumes a stable goal structure and mostly prescribed execution sequence, as these impact the values of the predictors. This means that our current COI model is best suited for tasks that meet these constraints, e.g., high frequency, routine, or safety critical tasks. One approach for addressing this limitation is to create multiple task models, apply our COI model to them, and adapt or develop tools that can automate much of the process, e.g., [12 , 23]. Also, when simpler tasks are composed into more complex activities, the COI values assigned to the simpler tasks cannot be directly applied to the composition. The current solution requires that the COI values be recalculated by fully applying the COI model to the broader activity.

FUTURE WORK

Our future work is to:

- *Investigate automated methods for building task models and predicting COI.* In this work, we developed the task models by hand and manually applied the COI model. Though this process works, it requires a fair degree of effort. We are thus investigating how to adapt automated task modeling tools (e.g., [23]) to not only fully support building hierarchical task models, but also to automate the process of predicting COI at the boundaries.
- *Extend the COI model to include non-boundary points*. A system may need to interrupt at non-boundary points, e.g., when the time until the next low-cost boundary is too long. Since different types of subtasks, e.g., language comprehension, memory store/recall, mental reasoning, etc. typically induce different workload, they would also have different COI. We would like to follow a similar process to extend our COI model for different subtasks.
- *Implement our COI model within an existing interruption reasoning system.* As discussed earlier, COI values can be encoded within machine-parsable task descriptions. A user's task execution can be matched to the descriptions to identify when a subtask boundary is reached, retrieve its COI value, and pass it along to a broader framework. We are more fully implementing this process within [3] and will soon be testing its efficacy in practice.

CONCLUSION

Our work has made several contributions towards enabling systems to compute an accurate cost of interruption (COI). First, we drew upon literature in cognitive psychology and our prior work to establish that systems need to consider task structure when reasoning about when to interrupt.

Second, using data collected in an experiment, we showed *which* characteristics of boundaries are most predictive of resumption lag and then developed a parsimonious model that maps these predictors to a set of discrete COI classes.

Third, our model was applied to predict COI at boundaries within different tasks. Results showed that reasonably high classification accuracy was achieved. Also, results showed that predicted COIs corresponded to meaningful differences in resumption lag, validating that systems can and should use our model to differentiate among subtask boundaries.

Finally, we described how our model could be integrated into frameworks that consider cues beyond the context of the current task. This would allow systems to make better decisions about when to interrupt than is possible today.

REFERENCES

1. Adamczyk, P.D. and B.P. Bailey. If Not Now When? The Effects of Interruptions at Different Moments within Task Execution. *CHI*, 2004, 271-278.
2. Altmann, E.M. and J.G. Trafton. Task Interruption: Resumption Lag and the Role of Cues. *CogSci*, 2004.
3. Bailey, B.P., P.D. Adamczyk, T.Y. Chang and N.A. Chilson. A Framework for Specifying and Monitoring User Tasks. *Journal of Computers in Human Behavior*, to appear, 2005.
4. Bailey, B.P. and J.A. Konstan. On the Need for Attention Aware Systems: Measuring Effects of Interruption on Task Performance, Error Rate, and Affective State. *Journal of Computers in Human Behavior*, to appear, 2005.

5. Card, S., T. Moran and A. Newell. *The Psychology of Human-Computer Interaction*. Lawrence Erlbaum Associates, Hillsdale, 1983.

6. Chung, P.H. and M.D. Byrne. Visual Cues to Reduce Errors in a Routine Procedural Task. *CogSci*, 2004.

7. Cutrell, E., M. Czerwinski and E. Horvitz. Effects of Instant Messaging Interruptions on Computing Tasks. *CHI*, 2000, 99-100.

8. Czerwinski, M., E. Cutrell and E. Horvitz. Instant Messaging and Interruption: Influence of Task Type on Performance. *Annual Conference of the Human Factors and Ergonomics Society of Australia (OZCHI)* , 2000, 356-361.

9. Czerwinski, M., E. Cutrell and E. Horvitz. Instant Messaging: Effects of Relevance and Timing. *People and Computers XIV: Proceedings of HCI*, 2000, 71-76.

10. Czerwinski, M., E. Horvitz and S. Wilhite. A Diary Study of Task Switching and Interruptions. *Proceedings of the ACM Conference on Human Factors in Computing Systems*, 2004, 175-182.

11. Degani, A. and E. Wiener. Cockpit Checklists: Concepts, Design, and Use. *Human Factors*, 35 (2), 345-359, 1993.

12. Dragunov, A.N., T.G. Dietterich, K. Johnsrude, M. McLaughlin, L. Li and J.L. Herlocker. Tasktracer: A Desktop Environment to Support Multi-Tasking Knowledge Workers. *IUI*, 2005, 75-82.

13. Fogarty, J., S.E. Hudson and J. Lai. Examining the Robustness of Sensor-Based Statistical Models of Human Interruptibility. *CHI*, 2004, 207-214.

14. Fogarty, J., A.J. Ko, H.H. Aung, E. Golden, K.P. Tang and S.E. Hudson. Examining Task Engagement in Sensor-Based Statistical Models of Human Interruptibility. *CHI*, 2005, 331-340.

15. Horvitz, E. and J. Apacible. Learning and Reasoning About Interruption. *ICMI*, 2003, 20-27.

16. Horvitz, E., A. Jacobs and D. Hovel. Attention-Sensitive Alerting. *Conference Proceedings on Uncertainty in Artificial Intelligence*, 1999, 305-313.

17. Horvitz, E., P. Koch and J. Apacible. Busybody: Creating and Fielding Personalized Models of the Cost of Interruption. *CSCW*, 2004, 507-510.

18. Hudson, J.M., J. Christensen, W.A. Kellogg and T. Erickson. "I'd Be Overwhelmed, but It's Just One More Thing to Do": Availability and Interruption in Research Management. *CHI*, 2002, 97-104.

19. Hudson, S.E., J. Fogarty, C.G. Atkeson, D. Avrahami, J. Forlizzi, S. Kiesler, J.C. Lee and J. Yang. Predicting Human Interruptibility with Sensors: A Wizard of Oz Feasibility Study. *CHI*, 2003, 257-264.

20. Iqbal, S.T., P.D. Adamczyk, S. Zheng and B.P. Bailey. Towards an Index of Opportunity: Understanding Changes in Mental Workload During Task Execution. *CHI*, 2005, 311-320.

21. Iqbal, S.T. and B.P. Bailey. Investigating the Effectiveness of Mental Workload as a Predictor of Opportune Moments for Interruption. *CHI*, 2005, 1489-1492.

22. Jackson, T.W., R.J. Dawson and D. Wilson. The Cost of Email Interruption. *Journal of Systems and Information Technology*, 5 (1), 81-92, 2001.

23. John, B.E., K. Prevas, D.D. Salvucci and K. Koedinger. Predictive Human Performance Modeling Made Easy. *CHI*, 2004, 455-462.

24. Maglio, P. and C.S. Campbell. Tradeoffs in Displaying Peripheral Information. *CHI*, 2000, 241-248.

25. McCrickard, D.S., R. Catrambone, C.M. Chewar and J.T. Stasko. Establishing Tradeoffs That Leverage Attention for Utility: Empirically Evaluating Information Display in Notification Systems. *IJHCS*, 58 (5), 547-582, 2003.

26. McFarlane, D.C. Coordinating the Interruption of People in Human-Computer Interaction. *Proceedings of the IFIP TC.13 International Conference on Human-Computer Interaction*, 1999, 295-303.

27. McFarlane, D.C. and K.A. Latorella. The Scope and Importance of Human Interruption in Hci Design. *Human-Computer Interaction*, 17 (1), 1-61, 2002.

28. Miyata, Y. and D.A. Norman. Psychological Issues in Support of Multiple Activities. In Norman, D.A. and Draper, S.W. (eds.) *User Centered System Design: New Perspectives on Human-Computer Interaction*, Lawrence Erlbaum Associates, Hillsdale, NJ, 1986, 265-284.

29. Monk, C.A., D.A. Boehm-Davis and J.G. Trafton. The Attentional Costs of Interrupting Task Performance at Various Stages. *HFES*, 2002.

30. Navon, D. and D. Gopher. On the Economy of the Human Processing Systems. *Psychological Review*, 86, 254-255, 1979.

31. Newell, A. and P.S. Rosenbloom. Mechanisms of Skill Acquisition and the Law of Practice. In Anderson, J.R. ed. *Cognitive Skills and Their Acquisition*, Erlbaum, Hillsdale, NJ, 1981, 1-55.

32. Ritter, F.E., G.D. Baxter, G. Jones and R.M. Young. Supporting Cognitive Models as Users. *ACM Transactions on Computer-Human Interaction*, 7 (2), 141-173, 2000.

33. Rubinstein, J.S., D.E. Meyer and J.E. Evans. Executive Control of Cognitive Processes in Task Switching. *Journal of Experimental Psychology: Human Perception and Performance*, 27 (4), 763-797, 2001.

34. Shneiderman, B. *Designing the User Interface*. Pearson Addison Wesley, Third Edition, 1997.

35. Tollinger, I., R.L. Lewis, M. McCurdy, P. Tollinger, A. Vera, A. Howes and L.J. Pelton. Supporting Efficient Development of Cognitive Models at Multiple Skill Levels: Exploring Recent Advancements in Constraint-Based Modeling. *CHI*, 2005, 411-420.

36. Trafton, J.G., E.M. Altmann, D.P. Brock and F.E. Mintz. Preparing to Resume an Interrupted Task: Effects of Prospective Goal Encoding and Retrospective Rehearsal. *IJHCS*, 58, 583-603, 2003.

37. Wickens, C.D. Multiple Resources and Performance Prediction. *Theoretical Issues in Ergonomic Science*, 3 (2), 159-177, 2002.

38. Zacks, J., B. Tversky and G. Iyer. Perceiving, Remembering, and Communicating Structure in Events. *Journal of Experimental Psychology: General*, 130 (1), 29-58, 2001.

39. Zijlstra, F.R.H., R.A. Roe, A.B. Leonora and I. Krediet. Temporal Factors in Mental Work: Effects of Interrupted Activities. *Journal of Occupational and Organizational Psychology*, 72, 163-185, 1999.