Navigation in Degree of Interest Trees

Raluca Budiu
budiu@parc.com
Palo Alto Research Center
3333 Coyote Hill Rd.
Palo Alto, CA 94304

Peter Pirolli
pirolli@parc.com
Palo Alto Research Center
3333 Coyote Hill Rd.
Palo Alto, CA 94304

Michael Fleetwood
fleet@rice.edu
Rice University
P.O. Box 1892
Houston, TX 77005

ABSTRACT
We present an experiment that compares how people perform search tasks in a degree-of-interest browser and in a Windows-Explorer-like browser. Our results show that, whereas users do attend to more information in the DOI browser, they do not complete the task faster than in an Explorer-like browser. However, in both types of browser, users are faster to complete high information scent search tasks than low information scent tasks. We present an ACT-R computational model of the search task in the DOI browser. The model describes how a visual search strategy may combine with semantic aspects of processing, as captured by information scent. We also describe a way of automatically estimating information scent in an ontological hierarchy by querying a large corpus (in our case, Google's corpus).

Categories and Subject Descriptors
H.52 [**User Interfaces**]: Theory and methods; D.2.8 [**Software Engineering**]: Metrics—*complexity measures, performance measures*

General Terms
Measurement, Human Factors, Theory

Keywords
User studies, user models, ACT-R, DOI trees, information scent, information visualization

1. INTRODUCTION
Visualizing large hierarchical information structures is a pervasive task in the real world. Whether the structures are organization charts, web pages, chronological information, or even file systems, we need to see as much as possible on a limited size screen. With that aspiration in mind, one successful and inventive technique was focus + context [2]. Focus + context is a name that covers several visualization methods (e.g., fisheye views [2], hyperbolic trees [6], degree-of-interest trees [3]) aiming to increase the amount of information displayed to a user. These visualizations generally present in greater detail the information around the user's (assumed) point of focus and offer a more schematic view of the other parts of the information structure. However, in spite of their intuitive efficacy, it is not clear that focus + context techniques present a real advantage to the everyday user. Performance in focus + context visualization appears to be largely a function of the relation between the semantics of the navigation cues and the user's tasks. (This relation is called information scent and is described in greater detail below.) For instance, [12] examines users' performance on search tasks performed with a hyperbolic tree browser and a more common, Windows Explorer-based browser. They conclude that, although in some cases (e.g., expert users or low information scent tasks) the hyperbolic tree may produce faster results, the interaction between the semantic qualities of the labels and the visual layout is quite complex and a clear advantage cannot be predicted for the hyperbolic tree.

The purpose of this paper is to understand how the combination between visual and semantic information in the focus + context browser affects users' performance in a search task. To achieve this goal, we first collected experimental data about how users perform simple search tasks. We looked at two kinds of search tasks: high-scent tasks, in which the target item was on a highly predictable, common-sense path, and low-scent tasks, in which the target item was hidden under less plausible labels. We collected the times that people needed to complete the search tasks for two different browsers, as well as finer grain eye-movement data such as number of nodes looked at, number of revisitations per node, or direction of eye gaze. Then, based on these data, we built a detailed cognitive model of navigation, that took into account visual search phenomena, as well as semantic aspects of the information processing.

One of the problems in evaluating interfaces or in devising computational models of browsing is an automatic measure of information scent: how do we know which nodes look highly relevant to people? This paper also describes a way of automatically determining scent in an ontological hierarchy. Our method involves extracting category relationships from a large corpus (in our case the corpus that Google makes accessible through their API).

1.1 Degree of Interest Trees

Degree of Interest (DOI) trees [3] are an instance of focus + context visualization and use degree-of-interest calculations to decide what gets displayed on the screen. Figure 1a shows a degree of interest tree that represents an ontological hierarchy of concepts. The hierarchy was first used in the Great CHI'97 Browse-off [10]. In Figure 1a the node in focus is *Artificial*, the node that was last clicked by the user. Note that only the nodes close (in terms of tree distance) to the node in focus or to its ancestors are displayed.

Figure 1b shows a a traditional Windows-Explorer–like type of file browser (henceforth called Explorer) — specifically, the image that the user sees after clicking on the node *Artificial* in this browser. Fewer nodes are displayed and the user cannot get any sense about the kind of information hidden under the other labels except for the ones he has already clicked. The tree structure is harder to grasp in this display, too.

The assumptions underneath the DOI tree visualization are (1) that the users will navigate more easily through the information structure because its skeleton is clearly exposed; and (2) that they will learn faster about the information in the tree, because more (possibly unnecessary) information is available to them.

1.2 Information Scent

Throughout the paper we refer to the following search task: *find the node with the label A* (abbreviated as *find A*) in a hierarchical ontology, as displayed by some browser. A is called the target node. In solving this task, the user must click on a sequence of nodes to find the descendants of those nodes that are not already displayed on the screen. To understand how the semantic quality of the labels affects the navigation, we looked at two kinds of tasks that differed in how predictable the path to the target was: tasks with high information scent and tasks with low information scent. In a search task, the information scent [11] of a label is an informal estimate of the likelihood that the label hides the target node. A task is high scent if all the labels on the path to the solution have high scent; it is low scent if at least some of these labels are low scent. For instance, for the search task *find a banana* in an ontology like the one in Figure 1, the path to the node *banana* is *Categories* → *Things* → *Natural* → *Vegetable* → *Fruits* → *Tropical* → *Banana*, so the task is high scent because all nodes on the path make sense. However, our ontology is not perfect: an item such as *Library of Congress* is under *Categories* → *People* → *Specific People* → *Organizations* → *Governmental* → *United States* → *Legislative Branch* → *Library of Congress*; this task is low scent because it is less intuitive that *Library of Congress* should be under the labels *People* or *Specific People*. In our experiments we actually use human ratings to decide whether a task is low or high scent.

2. EXPERIMENT

This experiment examines whether the DOI tree indeed provides substantial performance gains for its users. Specifically, we looked at how participants performed low- and high-scent search tasks using two browsers: a DOI browser (see Figure 1a) and a traditional, Explorer browser (see Figure 1b). We collected reaction times, mouse movements and clicks, eye movements, but we do not discuss all of these measures in this article.

Table 1: Average response times (s) for the search tasks.

Browser	High Scent	Low Scent
DOI	24.54	69.01
Explorer	28.42	59.83

Participants. Eleven participants were recruited from Stanford University and from PARC; the participants from Stanford were paid $40. Two additional recruits were eliminated due to eye tracking problems.

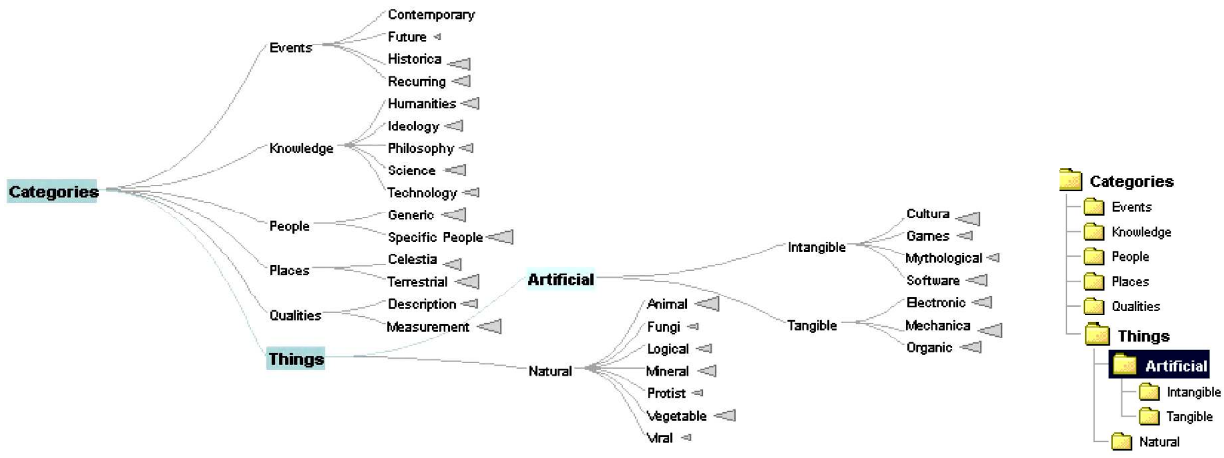
Materials. Sixteen search tasks were used. All these tasks involved finding one node in the ontological hierarchy used in [12]. Out of the 16 tasks, 8 were low scent (e.g., *find the Library of Congress*, discussed in Section 1.2) and 8 were high scent (e.g., *find a banana* in Section 1.2). All these tasks were used before in Experiment 1 from [12]. The decision whether a task was high or low scent was based on normative data collected in [12]. An ISCAN RK-426PC eye-tracker was used to record eye movements.

Procedure. The participants proceeded through (a) a familiarization phase, (b) a practice phase, and (c) a test phase. During the familiarization phase, the experimenter demonstrated the During the practice phase, each participant completed two practice tasks with each browser. Then the participants' eyes were tracked. During the test session, each participant completed the two sets of 32 tasks. For each participant, one test list was presented with one browser, and then the second test list of 16 items with the other browser. List order and browser order were counterbalanced across participants. The presentation order of test items within each list was randomized for each participant.

2.1 Results and Discussion

Our design was repeated measure. As suggested by Lorch and Myers [9], to make sure that all the error terms are taken into account, the correct statistical method to analyze this kind of design is to run separate regressions on the data from each subject that include all the variables of interest. Then, for each variable, one should use a t-test to check if the regression coefficients from all subjects are significantly different from 0.

Response Times. The average response times are shown in Table 1. The only significant variable that affected the response times was the scent ($t(10) = 3.75, p < 0.05$): participants completed the high scent tasks faster than the low scent tasks. The browser had no effect on how fast the participants performed the tasks ($t(10) = -1.23, p > 0.2$). Moreover, unlike in studies of the hyperbolic browser, there was no interaction between the scent and the browser type; namely, participants did not behave differently for different types of tasks in different browsers. The lack of a browser effect contradicted the common sense expectation that the DOI tree may actually facilitate search due to more nodes being exposed to the participants. In what follows, we explore some other measures of participants' performance in



(a) (b)

Figure 1: Two different visualizations for the same ontological hierarchy: (a) DOI tree (b) Explorer.

an attempt to understand why no performance difference was obtained for the two browsers.

Number of nodes visited. This measure refers to the number of distinct nodes either looked at or clicked on) by each participant in each tasks. There were far fewer nodes ($t(10) = -4.47, p < 0.005$) visited in the Explorer browser (on average, 53 nodes per task) than in the DOI tree browser (80 nodes). The number of nodes visited in the high scent task was lower than for the low scent tasks (47 versus 85 nodes: $t(10) = 4.21; p < 0.005$), confirming our intuition that the low scent tasks were harder and that the participants had to wander around quite a while until they found the correct path. The interaction between browser and scent was not significant. That people visit more nodes in the DOI tree is not surprising: by definition, the DOI tree attempts to expose the users to as much as possible relevant information. However, the fact that they are not faster with the DOI browser suggests that this extra amount of information possibly means time needlessly spent.

Average Revisitation. We also looked at how many times a participant revisited the same node on average. People tended to revisit more nodes in the Explorer than in the DOI browser ($t(10) = 4.34; p < 0.005$): on average they revisited a node about 6.6 times in the DOI tree, as indicated by the mouse clicks and eye movements. The same number is 8.15 for the Explorer. Since each revisit costs time and based on the number of nodes visited, one can hypothesize that the time spent in the Explorer for revisitation (needed to backtrack when a wrong path has been taken) is actually used in the DOI browser for visiting new nodes. So in fact people do get some extra knowledge of the information structure with the DOI tree; however, getting that extra knowledge acts as a distraction at the level of task completion.

Distance to Solution Path. This measure refers to the average distance from the nodes visited to the actual solution path[1]. Not surprisingly, participants went further away ($t(10) = -6.59; p < 0.0001$) from the solution path in the DOI browser (on average, 4.57 nodes away) compared to the Explorer browser (only 3.96 nodes away). Moreover, in the high scent tasks participants wandered less ($t(10) = 4.2; p < 0.005$) from the solution path (3.62 nodes on average) compared with the low scent tasks (4.83 nodes).

By examining these data and also the individual participants' data we got more insights into the advantages and disadvantages of the DOI tree. First, the DOI tree seemed to be more forgiving if participants made a mistake. Namely, even if the participants clicked on a wrong node, if the wrong node was close to the correct one, clicking on that node would also expand some of the children of the correct node and would offer the opportunity to recover from that mistake. For instance, in Figure 1a, the participant looking for the Ebola virus wrongly clicked on the node *Artificial*. However, that click also expanded the children of the correct node (*Natural*), thus giving the participant the occasion to correct themselves and choose the correct node (i.e., *Viral*). As seen from Figure 1b, in the Explorer browser this is not possible because the children of *Natural* are not visible.

Whereas the DOI tree was more forgiving for users' errors, its visual characteristics also had a greater potential for distraction. One example comes from looking at how participants completed the high scent task *Find the lobster*. Surprisingly, this task took a lot more time in the DOI browser (65s on average) than in the Explorer browser (only 25s). When we examined the data, we noticed that participants in the DOI browser tended to explore the node *Fishes* and related nodes more than in the Explorer. One explanation is that, because the node *Fishes* is visible on the screen early in the task (even after clicking on the node *Animals*) and, due to its high semantic similarity with *lobster*, it acts as a powerful distractor. *Fishes* does not have the same visibility in the Explorer, so users are less likely to make the mistake of choosing it. Figure 2 shows an instance where the node *Fishes* is displayed in a highly visible position.

[1]By distance from a node to a path in the tree we mean the length of the minimum path from that node to a node on the path to the solution.

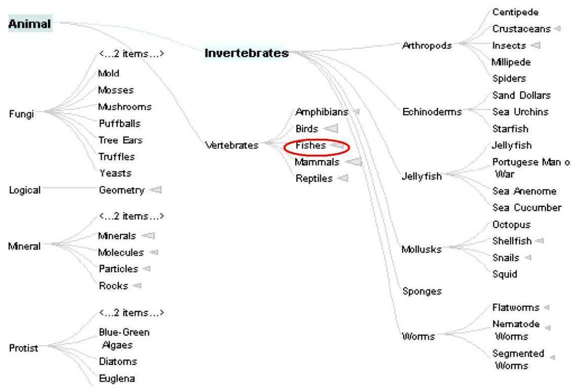


Figure 2: The node *Fishes* acts as a distractor in the task *Find a lobster.*

These data indicated that, as suggested by our initial intuition, a greater amount of information is exposed and processed in the DOI browser. However, this extra information has the potential of distracting the users in their search for a solution to their query.

Direction of eye movements and mouse clicks. We report here only the data from the DOI browser. We split the eye movements and the mouse clicks into two categories: movements/clicks made immediately after the mouse had been clicked (i.e., immediately after the screen has changed and new information has appeared on the screen) and movements/clicks made in between two mouse clicks, when the screen was static. We analyzed the direction of the eye movement/mouse click: (1) down, if the next action was on a node at a level lower in the tree than the current node; (2) up, if the next action was on a level higher than that of the current one; and (3) lateral,if the next action was at the same level. Participants' visual strategy was different in the high and low scent tasks. Immediately after clicking the mouse, participants had a marginal tendency to look up (i.e., back to the root) more in the low scent cases ($t(10) = 1.84, p < 0.1$). After a given mouse click, the next mouse click was more likely to be down in the tree for high scent than for low scent tasks ($t(10) = 4.93, p < 0.001$) and up or lateral in the tree for low scent tasks (up: $t(10) = -3.77; p < 0.005$; lateral: $t(10) = -4.37; p < 0.001$). These data showed that participants were more likely to follow the path they were on in the high scent cases, whereas in the low scent cases they may have preferred to go back or wander around the tree more in search for a label with a stronger scent. In between mouse clicks, participants tended to look back up in the tree more for low scent tasks than for high scent tasks ($t(10) = -1.93; p < 0.1$). However, for both types of tasks the majority of movements (more than 55%) were made in the lateral direction (i.e., on the same tree level).

3. MODELING NAVIGATION

In this section we present an ACT-R [1] computational model of navigation in degree of interest tree. The ultimate goal of such a model is to capture the data at the very fine grain level of eye movements. While we are not still at the stage where we could report such an accomplishment, we believe there are many lessons to be learned even from trying to model participants' data at a coarser grain level such as response times and nodes clicked or looked at. Our ACT-R model has two components: a visual search component and a semantic component. The model first decides which part of the screen it needs to focus on (based mostly on visual cues such as screen position, density, etc.) and then it examines the nodes in that part of the screen. The node examination is based on a scent type of function: for each node, it evaluates how good a match it is for this particular target. Then it selects the best node and clicks on it.

3.1 Estimating Individual Node Scent

The results of our experiment showed that semantic factors (low versus high scent) are highly relevant for both low and high scent But how should we estimate the scent of a node (e.g, *Vegetables* or *People*) with respect to a particular target (e.g., *Banana*)? One way is to ask people directly. This is how it has been done in our experiment to decide the scent of different tasks. However, the ratings that were collected referred to only the first four levels in our hierarchy. But when solving the task, since a model (or a person) could get to a random point in the tree, it would be useful to estimate the scent of any node in the tree with respect to the particular task. There are about 7000 nodes in the tree; therefore, a method for automatically estimating scent would be highly valuable.

Measures of semantic similarity may look as obvious choices for estimations of scent. We first decided to look at Latent Semantic Analysis (LSA)[7] and Pointwise Mutual Information (PMI)[14] as measures of semantic similarity. Both of these measures compute the similarity between two words based on their co-occurrences in the same documents. Unfortunately, these measures proved to be quite poor choices. Indeed, when we compared them with the human data that we had collected, neither PMI or LSA fared well.(Out of the 16 targets, only once the humans' top choice matched one of these measures' top choice.). But after looking at the task in more depth, we realized that for our ontology similarity was not a fair measure of scent. Indeed, the structure of the CHI ontology is mostly categorical, whereas co-occurrence metrics usually estimate semantic associatedness. So, instead of picking on associatedness cues, in a search task people may actually pick on category membership and respond questions such as: is *banana* a *thing* more than a *place*, a *monkey* or a *person*?

Unfortunately, although there are quite a few programs that automatically discover hypernymy, to the best of our knowledge there is none that answers this kind of question. With that in mind, we decided to build our own program. Much of the work on automatic hypernym discovery [5, 13] relies on the observation that there are certain textual patterns that mark the description of a category relationship (e.g., *banana is a (kind/type of) fruit, fruits such as bananas, fruits especially banana, fruit called banana*). We do not have a corpus rich in such examples (and that is one of the problems of automatic hypernym discovery research), so we decided to use the Google's corpus (available via Google API) to collect data about the number of co-occurrences of such strings. For each of our tasks we generated a Google

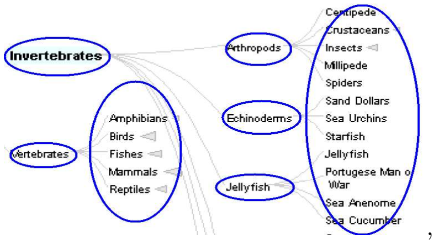


Figure 3: Different proximity-based groups as defined by our model.

query[2]). This is a shortened example: *"arthropods including lobster" OR "arthropods especially lobster" OR "lobster is a kind of arthropod" OR "lobster is a type of arthropod OR "arthropods like lobster" OR "arthropod called lobster" OR "lobsters and other arthropods" "lobsters or other arthropods" OR "arthropods such as lobster" OR "lobster is an arthropod" OR "lobster an arthropod"*. Then we recorded the estimated number of results that Google returned. (Note that this method is applicable with any other big enough corpus and search engine.)

To estimate scent from the numbers returned from Google, for each task we ranked the nodes in the reverse order of the number of results and then assigned a co-occurrence rank to each node. For each node we also computed PMI-like scores by dividing the Google results to the frequency of the node label (this frequency was obtained as the results returned from querying Google with the node label — e.g., *"lobster" OR "lobsters"*) and ranked these numbers in the same way, from highest to lowest. Then we averaged the co-occurrence and the PMI ranks to obtain a scent indicator[3]. This measure resulted in Google's top choice agreeing with humans's top choice in 7 out of 16 tasks. Out of the remaining 9 tasks, six of them presented some overlap between Google's top three choices and humans' top three choices. For only three tasks there was no relationship between this method's top three choices and subjects' top three.

3.2 The Model

Our navigation model was implemented in ACT-R[1]. ACT-R is a production-system–based cognitive architecture that has been extensively used to model various aspects of human cognition, from memory and problem solving, to language and web browsing.

According to work in the visual search literature [8, 4], when processing a visual scene, people often tend to group visual items together, select one such group and attend the items within it. These visual groups may be formed based on proximity or based on other common feature that they share to a large degree (e.g., color). In our model, we assume that the visual display is partitioned according to (vertical) closeness. Figure 3 shows an example of how the visual items are grouped together. Given a display, the model selects the best group according to several heuristics. Groups that contain nodes that are children of the node clicked on are preferred; also sparse groups [4] and groups close to the current eye position (i.e., the node that has been attended last). Once a group is selected, the nodes within that group are attended one by one, from top position to bottom position. The processing of each node includes three stages: (1) finding the location of the next item; (2) attending that location; (3) encoding the semantic information existent at that location. This last step is equivalent with processing the text of the label; it involves assessing whether the label stands for a hypernym of the search target (that is done essentially by estimating whether the degree of hypernymy is high enough). As the model scans the nodes, if it does not encounter any hypernyms and if it's deep enough in the tree[4], it decreases the probability of continuing to attend nodes in the same visual group. When all the nodes in the group have been visited or when the probability of continuing becomes too low, the model makes a selection: it chooses the node among the previously visited nodes that is the best hypernym of the target. If the model can find a best hypernym in the current visual group, it is the node that will be next clicked on. Otherwise, if no such node exists or if the best hypernym is part of a different visual group[5] the model makes a probabilistic decision: either leaves the group without clicking and looks for another group to attend or returns to the best hypernym found so far (if it is still displayed on the screen). The model then goes back to selecting a visual group.

Results of the simulations. Here we present some of the results obtained so far regarding response times and mouse clicks per task and we compare these data with the human participants data. Given that the model attempts to predict solution for complex search problems and no correction to the model are made when it makes a "wrong" move (i.e., a move that a human did not do), even finding the solution to a problem is a big accomplishment, especially in a world where the scent of the nodes (as we computed them) can be only approximations of the knowledge that people actually have. We stopped the model after it ran 300s without finding a solution. In this circumstances, the model found solutions to 11 out of the 16 tasks (7 high scent, 4 low scent). Part of the problem was our estimation of hypernymy: for instance, the phrase "crosscut saw is a type of $< A >$" is quite rare even for the huge Google corpus. In such cases we tried to estimate the hypernymy of the parts, but again "saw", with its more frequent verb meaning, drove us into problems.

When we looked at the response times, the model was able to capture the difference between low and high scent task: the high scent tasks took on average 41.17s and the low scent tasks were slower: 74.88s for the model. (Human data can be found in Table 1.) On average the model clicked on 69% of the nodes that humans clicked on[6]. In the case of high

[2]In fact, each Google query needed to be split in several queries due to Google-imposed constraints on the length of the query.

[3]The actual scent indicator was the difference between a large constant and that average.

[4]The nodes that are close to the root are general and unrelated. A node not being in one of these categories does not predict well whether it is going to be part of another.

[5]This is equivalent with the model remembering that elsewhere it has met a hypernym better than any node in this group.

[6]But some of the clicks that only the model made are perfectly plausible: for instance, when searching for *the Pawnee Indian tribe* it clicked on *Native American*, although none

scent tasks, the model clicked on 68% of the nodes that the participants clicked on; whereas this number was 73% for the low scent tasks.

4. CONCLUSIONS

We have presented an experiment and a computational model of how people navigate through large hierarchical information structures such as DOI trees. The experiment studied how people complete search tasks in a DOI tree browser and a Windows-Explorer like browser. The results suggested that, in terms of task completion speed, the two browsers are not different. What really made a difference in the task completion speed was the semantic aspect captured by the information scent: how well chosen were the labels for the nodes. People tend to visit more new nodes in the DOI tree than in the Explorer browser, whereas in the Explorer browser they tend to backtrack and revisit the same nodes more. By displaying more information, the DOI browser seems to encourage users to wander more off the solution path and to gather more knowledge than necessary for task completion. Thus, the DOI browser may prove beneficial over long term usage because it allows users to gather more information about the hierarchy structure. The DOI browser also allows users to recover more easily when they made certain types of errors; however, it also can distract the participants with highly salient but irrelevant items that are placed close to the users' focus of attention.

We also sketched an ACT-R computational model of navigation in DOI trees. The model combined a detailed visual strategy in which a region of the screen was located as relevant with a semantic process that made the selection of which node to click next. The semantic part of the model was based on an automatic estimation of the information scent of the nodes in the tree. In fact, one of the lessons from this model is that, to capture the human performance even in the roughest detail (e.g., which tasks get completed in a useful time) we need a good measure of information scent. We presented a measure that captures the hypernymy relationship: to what degree the search target A is member of the category denoted by the node label B? That measure was computed by querying a large corpus (in our case Google's database) with queries describing category membership.

Our goal was to understand the interplay between visual search and semantic information. We noted earlier that information scent plays the most important part in search. However, even good information scent will not be able to salvage a poor search strategy. Simple facts (such as the fact that people know that if they clicked a node, they should attend to new parts of the screen first) are not trivial to capture in a visual attention model when the visual environment is so rich in items.

Stepping back into the world of focus + context visualizations, our results with the DOI tree browser corroborated those in [12]: cramming more information on the screen does not necessarily improve performance. Putting the right labels on the information that you display is a better way to speed up the task.

---

of the six participants who completed that task clicked on that node.

5. ACKNOWLEDGMENTS

Portions of this research have been funded by an Advanced Research and Development Activity, Novel Intelligence from Massive Data Program, Contract No. MDA904-03-C-0404 to S.K. Card and P. Pirolli. We thank Duncan Brumby and Jeff Heer for their help with integrating the DOI browser with ACT-R.

6. ADDITIONAL AUTHORS

Additional authors: Julie Heiser (Stanford University, Stanford CA 94305, email: `jheiser@psych.stanford.edu`).

7. REFERENCES

[1] J. Anderson and C. Lebiere. *The atomic components of thought*. Erlbaum, Mahwah, NJ, 1998.

[2] S. Card, J. Mackinlay, and B. Schneiderman. *Information visualization*. Morgan Kaufmann, 1999.

[3] S. Card and D. Nation. Degree-of-interest trees: a component of an attention-reactive user interface. In *Advanced Visual Interface '02*, Trento, Italy, 2002.

[4] T. Halverson and A. J. Hornof. Strategy shifts in mixed-density search. In *The 26th Annual Meeting of the Cognitive Science Society*, Chicago, IL, 2004.

[5] M. Hearst. Automatic acquisition of hyponyms from large text corpora. In *Fourteenth International Conference on Computational Linguistics*, 1992.

[6] J. Lamping, R. Rao, and P. Pirolli. A focus + context technique based on hyperbolic geometry for visualizing large hierarchies. In *Conference on Human Factors in Computing Systems CHI '95*, 1995.

[7] T. K. Landauer and S. T. Dumais. A solution to Plato's problem: the Latent Semantic Analysis theory of acquisition, induction and representation of knowledge. *Psychological Review*, 104:211–240, 1997.

[8] G. Logan. The code theory of visual attention: An integration of space-based and object-based attention. *Psychological Review*, 103:603–649, 1996.

[9] R. Lorch and J. Myers. Regression analyses of repeated measures data in cognitive research. *Journal of Experimental Psychology: Learning, memory and cognition*, 16(1):149–157, 1990.

[10] K. Mullet, C. Fry, and D. Schiano. On your marks, get set, browse! In *Human Factors in Computer Systems, CHI '97 (Extended abstracts)*, Atlanta, 1997.

[11] P. Pirolli and S. Card. Information foraging. *Psychological Review*, 1999.

[12] P. Pirolli, S. Card, and M. Van Der Wege. The effect of information scent on searching information visualizations of large tree structures. In *AVI*, Palermo, Italy, 2000.

[13] R. Snow, D. Jurafsky, and A. Y. Ng. Learning syntactic patterns for automatic hypernym discovery. In *NIPS 17*, 2005.

[14] P. Turney. Mining the web for synonyms: PMI-IR versus LSA on TOEFL. In *Twelfth European Conference on Machine Learning (ECML-2001)*, 2001.