Task Aware Information Access For Diagnosis of Manufacturing Problems

Larry Birnbaum[1], Wallace Hopp[2], Seyed Iravani[2], Kevin Livingston[1], Biying Shou[2], Thomas Tirpak[3]

[1]Dept. of Computer Science
Northwestern University
1890 Maple Ave
Evanston, IL 60201 USA
+1 847 491 3500
{birnbaum, livingston}@cs.northwestern.edu

[2]Dept. of Industrial Engineering and Management Sciences
Northwestern University
2145 Sheridan Rd, C210 Tech
Evanston, IL 60208 USA
+1 847 491 3383
{hopp, s-iravani, b-shou }@northwestern.edu

[3]Motorola Labs Physical Realization Research Center
Motorola
1301 E. Algonquin Rd.
Schaumburg, IL 60196 USA
+1 847 576 5414
T.Tirpak@motorola.com

ABSTRACT

Pinpoint is a promising first step towards using a rich model of task context in proactive and dynamic IR systems. Pinpoint allows a user to navigate decision tree representations of problem spaces, built by domain experts, while dynamically entering annotations specific to their problem. The system then automatically generates queries to information repositories based on both the userís annotations and location in the problem space, producing results that are both task focused and problem specific. Initial feedback from users and domain experts has been positive.

Categories and Subject Descriptors

H.3.3 [**Information Storage and Retrieval**]: Information Search and Retrieval ñ *query formulation, retrieval models, search process, selection process*

General Terms

Design, Experimentation, Human Factors.

Keywords

User task and context modeling, dynamic information retrieval.

1. INTRODUCTION

Notwithstanding the genuine revolution in information access over the past two decades, peopleís everyday experience of trying to get the information they need to carry out their work is often one of incredible frustration. Even though what they need probably resides *somewhere* on their organizationís website, they can only find it -- *if* they can find it -- by crafting complicated queries to a search engine and hoping that what they are looking for is somewhere in the top ten results, or by combing through some gigantic index accessible through a corporate portal.

This all-too-common experience has led us to investigate technologies that assist the user in carrying out information search and access tasks -- or, even better, that take the burden off the user entirely. For example, Watson [2] is a system that analyzes the document that you are currently reading or writing in a variety of applications (e.g., word processors, email clients, web browsers, etc.) and automatically retrieves information relevant to that document from a variety of sources (the public web, corporate intranet, or proprietary data services).

Although it has proven quite effective in a number of studies [1], Watsonís notion of the userís context is limited to the contents of the documents that the user is currently manipulating. In other words, Watson has no notion of the real-world task being undertaken by the user (e.g., writing a proposal, developing a project plan, critiquing a product design, etc.), which may be only indirectly reflected in those documents. We believe that it is ultimately necessary to develop a richer characterization of the userís context -- in terms of the real-world task the user is carrying out, and where the user currently stands in that task -- in order to determine his or her information needs with enough specificity to be able to meet them automatically.

Pinpoint represents our first effort to develop such a task-centered approach to proactive and dynamic information delivery. It supports the task of diagnosing and repairing factory assembly-line problems. This domain was chosen because of the existence of *factory physics* [3], a body of science that describes the underlying principles governing behavior of manufacturing systems. Factory physics forms the basis of a powerful yet general characterization of the userís problem-solving context in the course of diagnosing and repairing problems in manufacturing systems. This characterization can then be used to support the automatic retrieval and presentation of task-relevant information.

2. SYSTEM ARCHITECTURE

Based on factory physics, we have developed a decision-tree model of the task of diagnosing and repairing production problems. (A small portion of the decision tree can be viewed in the top frame of Figure 2.) The categories of problems and repairs within this decision tree are entirely abstract. For example, a problem category within the decision tree is ìinsufficient utilization of bottleneck process.î The key point is that none of these diagnostic categories refer to the particular nature of the processes or the actual, physical causes for the problems described. This permits the application of the decision tree model to a wide variety of production processes.

IUIí05, January 9-12, 2005, San Diego, California, USA.
ACM 1-58113-894-6/05/0001.

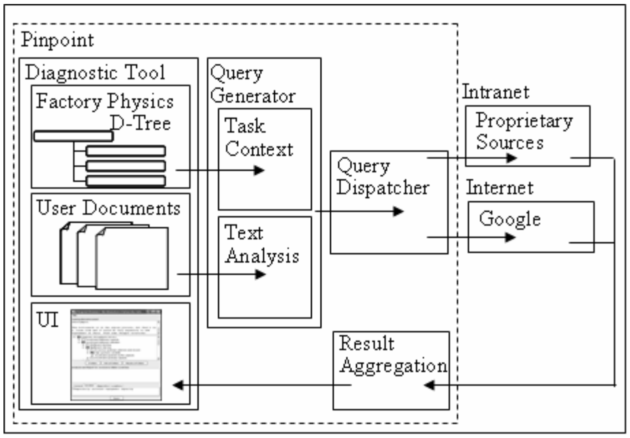

Figure 1: General architecture of the Pinpoint system.

Pinpoint supports users in traversing the decision tree as they diagnose production problems. The particular diagnostic category (hypothesis) currently under consideration forms one portion of the userís context -- a relatively abstract characterization of the current state of the userís diagnostic task. The other portion consists of documents reflecting the specifics of the current case, initially a ìtasking memoî describing the ìpresenting symptomsî of the problem, subsequently augmented by the userís analyses and annotations recorded as he or she traverses the decision tree (which ultimately constitute the final report or work product).

These two aspects of the userís context -- position in the decision tree (reflecting where the user stands in the diagnostic task) and user annotations (reflecting specifics of the current case) -- are used in two different ways to automatically generate queries that are likely to yield relevant information. The user-authored content is analyzed statistically and heuristically (as in Watson) to develop a set of keywords relevant to the specific problem domain (e.g., circuit board manufacturing). In addition, associated with each node in the decision tree is a series of abstract key words related to the problem represented by that node. These two sets of keywords are then combined to form an integrated characterization of the userís context that reflects both the state of the diagnostic task and the particular problem at hand.

This integrated characterization is then used to automatically generate queries to relevant information repositories, both public and proprietary, and the results are presented to the user. Our expectation is that this will result in the retrieval of more relevant information than could be found given either the abstract specification of the task state, or the domain-specific elements that can be gleaned from user documents, alone. The user can then employ these results to support his or her thinking, for example, to determine whether the current diagnostic category truly applies to the situation, or to help determine which sub-category to consider next, as well as directly incorporating them into the report being generated.

In sum, Pinpoint integrates a decision tree-based performance support tool with a proactive and dynamic information access system. The architecture of Pinpoint is depicted in Figure 1.

3. SAMPLE INTERACTION

A user invokes Pinpoint after being tasked to solve a manufacturing problem. Upon opening the Pinpoint browser

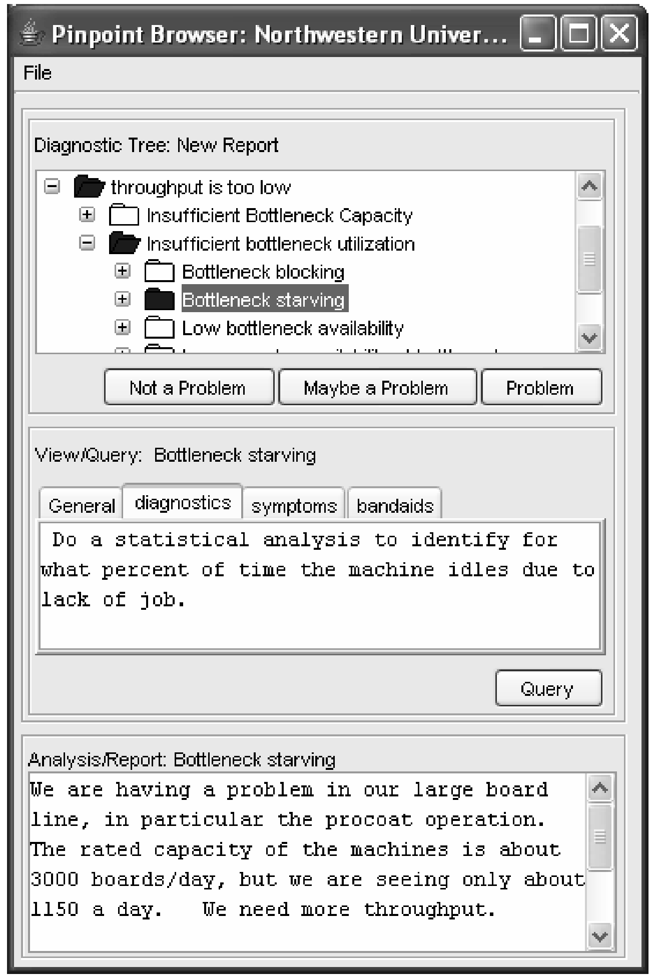

Figure 2: Pinpoint user interface. The interface is divided into three sections. The top allows the user to navigate between nodes in the decision tree. The middle allows users to select different ìviewsî of the currently selected node and to launch queries; here the ìdiagnosticsî view of the ìBottleneck starvingî node is selected. The bottom allows users to enter text annotations associated with the currently selected node.

(loaded with the factory physics decision tree) the user will be presented with an interface much like that shown in Figure 2. The user may begin by immediately starting to browse the tree. However, in general we would expect the user to enter any tasking documents or other specific information about the problem in the annotation box. This content would then be associated with the root node of the decision tree. Pinpoint records this information and all other annotations for inclusion in a final report, organized by the userís path through the tree.

As the user navigates the decision tree by expanding and selecting nodes in the top section of the interface, various views are presented in the middle section of the interface. These views provide different ways of looking at the problem represented by the current node. For example, ìdiagnosticsî views typically explain how to evaluate the userís current situation with respect to the selected potential problem node. Likewise, ìbandaidsî views provide quick fix information which may be possible given the current level of problem specificity. Not all nodes will have all views; views vary from node to node depending on how the tree is built by the tree designers (domain experts). One view that is always available is the ìGeneralî view, which is a default view representing the problem node itself.

As the user interacts with the decision tree, he or she can also annotate nodes by entering text in the bottom section of the interface. The user may tag nodes as being ìpart of the problem,î ìnot part of the problem,î or ìmaybe part of the problem.î These tags are represented visually in the tree through color.

Finally, the primary feature of the Pinpoint system is the ability of the user to issue queries simply by pressing the ìqueryî button located just under the view section of the interface. When a query is executed, Pinpoint identifies a set of relevant information repositories, based on the userís current location in the problem space, and then automatically generates queries for those information sources based on the contents of the decision tree, where the user is in that tree, and the annotations he or she has provided. The results of these queries are then displayed to the user in a web browser. The user can then browse these results, and upon finding useful information he or she can copy and paste either the information itself or URLs into the annotation field associated with the current node. These actions of browsing, querying, and annotating may be repeated throughout the tree.

4. CURRENT STATUS & CASE STUDIES

The Pinpoint diagnostic tree representation of the factory physics domain, at the time of writing, contains 265 nodes, and 641 views, with 265 of those views being the ìGeneralî or default views for the node which are automatically generated by the Pinpoint system. The number of additional views per node varies from zero (0) to four (4).

Initial user testing was done using a set of Industrial Engineering graduate students at Northwestern. They were presented with a realistic scenario (which one of the authors uses as an exam question for an upper-level Industrial Engineering class), and then asked to use Pinpoint as they worked through the problem. The students were given a version of Pinpoint that queried Google for relevant documents. The users identified the majority of the content present to them by Pinpoint as very on point, but not that useful given their existing level of knowledge.

Given these results we are inclined to believe that when equipped with general information repositories Pinpoint would be more useful as a pedagogical tool for novices than as a support tool for experts. Thatís not to say that the system using only general information repositories provides no leverage to sophisticated users. Although the bulk of the documents returned did not contain information of which the users were previously unaware, several gems were uncovered. For example, unbeknownst to the author of the scenario, a solution had been posted on the web, and this was quickly found by Pinpoint and ranked highly.

Pinpoint is currently under evaluation by several groups at Motorola. It is being extended to enable queries of Motorolaís Compass System for document knowledge management. Work with the current prototype has focused on applications in Motorolaís manufacturing lines. Other domains are also being investigated concurrently at Motorola, including supply chain management, and mechanical design.

With respect to Pinpointís effectiveness with general information repositories, users have told us that they feel the results they are retrieving from Google using Pinpoint are better than those they would receive if they were to use Google themselves.

5. RELATED WORK

Watson [1, 2] as described earlier, proactively and dynamically retrieves relevant information based on the document currently being manipulated by the user. Rhodesís Remembrance Agent [7] provides a similar service where the primary information being accessed consists of user authored prior documents. Liebermanís Letizia [6] automatically searches links from web pages being viewed by a user.

Previous task-based information systems include the Air Campaign Planning Advisor, or ACPA [5], a task context-based help and advice system for the air campaign planning task. However ACPA, while heavily based on task context, does not dynamically formulate and generate queries. Instead, task state is mapped to a set of fixed nodes in a hypermedia system.

Also highly relevant is Horvitz and colleaguesí work on the use of statistical analyses of user on-line activity to anticipate user needs, e.g., by retrieving relevant help and advice [4]. The statistical analysis is used to categorize userís goals or tasks in the current context, which in turn drives the delivery of content.

6. ACKNOWLEDGMENTS

The authors thank Joe Woods for help with the programming and interface design, Juan Lopez and Rich Anderson for help on deployment and testing at Motorola, and Kris Hammond and Jay Budzik for many useful discussions. We also thank the NSF for supporting this research.

7. REFERENCES

[1] Budzik, J. 2003. *Information Access in Context: Experiences with the Watson System.* PhD thesis, Northwestern University, Dept. of Computer Science, Evanston, IL.

[2] Budzik, J., Hammond K., and Birnbaum, L. 2001. Information access in context. *Knowledge-based systems*, vol. 14, pp. 37-53.

[3] Hopp, W.J., and Spearman, M.L.. 2000. *Factory Physics: Foundations of Manufacturing Management.* McGraw-Hill, New York.

[4] Horvitz, E., Jacobs, A., and Hovel, D. 1999. Attention-sensitive alerting. *Proc. Conf. on Uncertainty and Artificial Intelligence*, pp. 305-313.

[5] Johnson, C., Birnbaum, L., Bareiss, R., and Hinrichs, T. 2000. War stories: Harnessing organizational memories to support task performance. *ACM Intelligence*, vol. 11, pp. 16-31

[6] Lieberman, H. 1995. Letizia: An Agent That Assists Web Browsing. *The International Joint Conference on Artificial Intelligence.*

[7] Rhodes, B. J. and Starner, T. E. 1996. Remembrance agent: A continuously running automated information retrieval system. *Practical Applications of Intelligent Agents and Multi-Agent Technology (PAAM).*