Process Improvement for Traceability: A Study of Human Fallibility

Wei-Keat Kong, Jane Huffman Hayes
Department of Computer Science
University of Kentucky, Lexington, KY, USA
wkkong1@uky.edu, hayes@cs.uky.edu

Alex Dekhtyar*, Olga Dekhtyar+
Department of Computer Science*,
Department of Statistics+,
CalPoly, San Luis Obispo, CA, USA
{dekhtyar, odekhtya}@calpoly.edu

***Abstract*—Human analysts working with results from automated traceability tools often make incorrect decisions that lead to lower quality final trace matrices. As the human must vet the results of trace tools for mission- and safety-critical systems, the hopes of developing expedient and accurate tracing procedures lies in understanding how analysts work with trace matrices. This paper describes a study to understand when and why humans make correct and incorrect decisions during tracing tasks through logs of analyst actions. In addition to the traditional measures of recall and precision to describe the accuracy of the results, we introduce and study new measures that focus on analyst work quality: potential recall, sensitivity, and effort distribution. We use these measures to visualize analyst progress towards the final trace matrix, identifying factors that may influence their performance and determining how actual tracing strategies, derived from analyst logs, affect results.**

***Keywords*-Traceability; Human Factors; Performance Measures; Process Improvement; Tracing Strategies**

I. INTRODUCTION

Trace Matrices (TM) support the verification of mission- and safety-critical software systems. Even though TMs can be automatically generated, analysts are still required to vet TMs and ensure that critical requirements have been satisfied. Analysts, however, can sometimes make incorrect decisions that lower the quality of vetted TMs. Despite the subjectivity of analysts' decisions, it is not possible to do away with the analyst in the tracing process and rely only on software-generated TMs [1, 2, 3, 4]. To move toward improvement of tracing as a practice, it is necessary to consider the human in the tracing process improvement feedback loop.

Despite their vital role, the study of analysts in the context of working with tracing software [5, 6], calculating tracing effort and quality [7], and understanding how analysts perform tracing [1, 2, 3, 4] has only just begun. Important information about how analysts work with TMs has not yet been studied in detail and empirically validated. For example, how accurately do analysts perform tracing tasks? How often do analysts make correct decisions? How often and why do they make incorrect decisions? How do analysts spend their time during the tracing task and are they making the best use of their time? While we may have some intuitive answers, this work empirically validates those questions.

We know that automated tracing methods do not retrieve perfect TMs [2]. We know that analysts are not perfect either, and can often make a high quality TM worse [2, 3]. We, however, **need** analysts to properly validate TMs and improve their accuracy. Our overarching goal is to develop procedures and software that facilitate **accurate** assisted tracing, where analysts work with the output of an automated tracing tool [1]. To that end, we need to identify things that analysts do well and things with which they struggle. Based on this knowledge, we can make improvements (better tracing methods, better user interfaces, better procedures that capitalize on analyst strengths) and avoid things that challenge analysts (or handle these challenges).

Recall and precision are measures frequently used to evaluate the accuracy of a TM from a researcher's perspective [8, 9, 10, 11, 12], while measures such as lag, selectivity, and mean average precision have been used to evaluate the quality of a TM from an analyst's perspective [5, 13]. In general, automated methods return candidate TMs ("candidate" until a human analyst vets them) with high recall and low precision. New or improved techniques attempt to maintain high recall while increasing precision over existing techniques. These measures, however, indicate the accuracy of the final tracing product and not the accuracy of the analyst. We need measures to capture information about analyst behavior in order to understand how analysts perform tracing tasks and what factors affect their work quality.

We posit that recall may not always be preferred over precision when evaluating analyst quality. Recall only tells us how many true links an analyst added to the final TM and not how many they did not find or incorrectly rejected. Analysts' performance should reflect all their decisions on true and false links. An analyst that rarely rejects a true link, rarely accepts a false link, and spends less effort on false links produces a high quality final TM. Analysts need to be able to observe all the true links in the candidate TM in order to maximize the likelihood of accepting those links into the final TM.

This paper introduces three new measures that target the study of the tracing process *in addition* to the accuracy (recall and precision) of the final TM: *potential recall*, *sensitivity*, and *effort distributio*n. We apply these measures in a multi-site and multi-dataset study of assisted

978-1-4673-2785-5/12/$31.00 © 2012 IEEE

requirements tracing. We study when analysts make correct and incorrect decisions by logging analyst actions during a tracing task. We also introduce a matrix visualization that provides an at-a-glance view of analyst decisions on true links. To support trend analysis, we visualize analyst logs using a lattice chart that tracks the state of the TM and analyst measures over time. We analyze log and survey data to identify actual participant tracing strategies.

The paper is organized as follows. Section II discusses background and related work while Section III presents the new analyst-specific measures. Section IV describes the study setup. Section V presents threats to validity. Section VI discusses results and Section VII presents observations. Section VIII concludes and addresses future work.

II. Background and Related Work

The *assisted tracing* process is best described as follows: an analyst uses an automated method to generate a candidate TM, reviews it, makes any desired changes, and *certifies* the final TM. We know human analysts are not perfect and cannot possibly review every link in the candidate TM without investing significant time and effort. The analyst has to decide how to best spend their time in order to produce a high quality final TM. We measure the quality of the final TM against an answer set TM (an independently validated TM that contains all the true links in a document collection) using recall and precision, defined below:

$$\text{Recall} = TL_a / TL_t \,, \tag{1}$$

$$\text{Precision} = TL_a / (TL_a + FL_a) \,, \tag{2}$$

where TL_a is the number of true links *accepted*, TL_t is the *total* number of true links in the collection, and FL_a is the number of false links *accepted*.

The study of automated tracing methods has resulted in much progress toward the automation of candidate trace link generation. Techniques using latent semantic analysis [11], key phrases [14], unsupervised learning [15], and term proximity [13] exploit the structural relationship between words in a document. The use of thesauri [10, 16] and web queries [17] supplement trace link generation with external information to improve weak links. Automated tracing techniques generate TMs that include most of the true links that should be found (80 – 90% recall). These techniques, however, also retrieve many false links (below 10% precision on large datasets and 20-40% on smaller datasets with unfiltered results [8, 10, 11, 12]).

Analyst simulations provided a means to test tracing strategies prior to studies of actual human analysts performing tracing. Relevance feedback with multiple iterations and filtering to validate candidate TMs showed improvements in final TMs (results included links used for feedback.) [10]. Incremental approaches using document cut or threshold weight filtering with various feedback strategies showed that a significant amount of effort is required to retrieve all true links in the TM (results excluded links that were used for feedback, and in some cases feedback made results worse) [18]. This study identifies actual tracing strategies from the logs of analyst actions, providing guidance on tracing strategies for future simulation studies.

Another analyst simulation study looked at how link ordering and analyst feedback affected results, measuring the effort required to achieve either a fixed recall level or to measure the recall achieved using a fixed amount of effort. Results showed that local ordering with feedback performed the best. It was observed that determining the stopping point is crucial, using feedback helps, and using a systematic approach helps [6]. Simulated analysts produced better results when evaluating links incrementally instead of the entire ranked list at a time [12].

While the simulation studies above assumed that analysts made perfect decisions, studies of actual human analysts showed otherwise. Given higher accuracy candidate TMs, analysts produced slightly lower accuracy final TMs. Given lower accuracy candidate TMs, analysts produced *significantly* higher accuracy final TMs [1, 2, 3]. Analysts tended to produce final TMs that were near the precision = recall line, meaning they had final TMs that were about the size of the true TM [2]. Analysts were better at validating links as opposed to searching for missing links [4] and their accuracy did not depend on whether they had industrial experience or not (while experienced analysts were more correct on true links than those with less experience, both achieved less than 50% precision) [7]. Decisions were more likely to be correct when made quickly and most decisions were made on false links [4, 7]. Effort spent validating links did not correlate with trace accuracy [2, 7]. This study of the analyst differs from prior work in that we study analyst decisions during the tracing task through the logs of their actions.

Recall and precision measures of the final TM reflect the number of links accepted by the analyst. These measures, however, do not indicate how many links the analyst actually examined and rejected. A particular measure of interest would be the number of true links rejected by the analyst as this indicates that the analyst did not acknowledge the relevance of the link. The following section proposes new measures that capture additional information about analyst decisions.

III. New Analyst Measures

Throughout the tracing process, the analyst observes, or *sees,* numerous candidate links. As stated earlier, however, the analyst is not expected to examine every link. Some true links may be among the candidate links not seen by the analyst. Thus, when it comes to validating true links, analyst accuracy is limited by the percentage of the true links seen. This percentage, dubbed ***potential recall,*** represents the upper bound on recall. It is defined as follows:

$$\textbf{\textit{Potential recall}} = TL_s / TL_t \,, \tag{3}$$

where TL_s is the number of true links *seen* (accepted, rejected, or left undecided), and TL_t is the *total* number of true links in the collection.

When we measure analyst accuracy with respect to the number of true links actually observed, we obtain a new measure, which we call ***sensitivity.*** Sensitivity is defined as:

$$\textbf{\textit{Sensitivity}} = TL_a / TL_s\ , \qquad (4)$$

where TL_a is the number of true links *accepted* and TL_s is the number of true links *seen*. Note that while recall is a measure of accuracy of the final TM, sensitivity *measures the quality of analyst decision-making on true links.* For example, an analyst who sees 90% of the true links but accepts only 50% of them (50% sensitivity) has 45% recall. Contrast this to another analyst that sees 45% of the true links and accepts all of them (100% sensitivity) yielding 45% recall as well. Between these two analysts, the one with higher sensitivity does a better job at deciding on true links. High sensitivity, however, can easily be achieved by accepting all the links in the candidate TM (which would likely not be a good approach as tracing tools also retrieve many false links). Precision balances sensitivity in the same way it balances recall, by measuring how selective analysts are at accepting links into the final TM.

Additionally, we want to measure analyst effort and how it is spent throughout the tracing process. In order for analysts to make the best use of their time, the effort spent reviewing false links should be balanced by the effort spent reviewing true links. The following measure can be used to indicate how analysts spend their time during a tracing task in terms of the number of links seen:

$$\textbf{\textit{Effort distribution}} = FL_s / TL_s\ , \qquad (5)$$

where FL_s is the number of false links *seen* and the TL_s is the number of true links *seen*. An analyst that sees an equal number of true links and false links has an ***effort distribution*** of one (1). We posit that analysts who view many false links are more likely to accept some of those links into the final TM, decreasing precision. Note, however, that an analyst may not go through the trouble of rejecting false links if they know that only accepted links are included in the final TM, which could result in higher ***effort distribution*** if they are skimming through links looking for specific keywords.

Each analyst, without specific traceability training or guidance, approaches tracing in their own way. Often, an analyst uses some sort of strategy, either consciously or unconsciously, to complete the tracing task. Capturing these strategies (without detracting from the actual tracing task) provides insight as to which strategies produce the best results in terms of ***potential recall***, ***sensitivity***, and ***effort distribution***. These strategies could also indicate the threshold that an analyst applies to what they consider to be a true link, which influences the precision of the final TM.

In order to design reliable and accurate *assisted tracing* processes, we need to understand what factors contribute to analyst performance in tracing tasks. In our prior studies [1, 2, 3], we varied the accuracy of the starting candidate TM for the tracing task and discovered that it strongly influenced the accuracy of the final TM. Meanwhile, almost no other factors related to individual analyst qualities, their environment, and their approach to tracing had any significant influence. In this study, we focus on the link validation task and drill down into analyst actions using logs of their tracing activity. By having participants work with the same starting candidate TM, any variability in responses can be attributable to other factors.

We identify three categories of factors that can influence analyst performance: *(i) personal characteristics*, *(ii) environmental characteristics*, and *(iii) tracing behavior.* These sets of characteristics are measured in different ways in the data we collected. These characteristics, however, are not independent. In particular, the tracing behavior of analysts can be motivated by **both** their personal characteristics and environmental factors.

Among the personal characteristics of the participants, we look at their *grade level, software engineering experience, tracing experience,* and *confidence in tracing.* Environmental characteristics in our study are essentially the study *dataset* and the *study location/group*. Logs and post-study surveys allow us to extract information about the tracing behavior of the participants. In this work we consider four tracing behaviors: *time to complete* the tracing task, *link selection strategy, use of feedback,* and *average number of links viewed* per high-level element.

These motivations lead to the following questions:

RQ1: How do analysts creating the final TM perform using these new measures?

RQ2: What are statistically significant factors that affect analyst performance?

RQ3: Do better-performing analysts exhibit certain trends using these new measures?

RQ4: How do tracing behaviors affect the results of the tracing task?

IV. Study Details

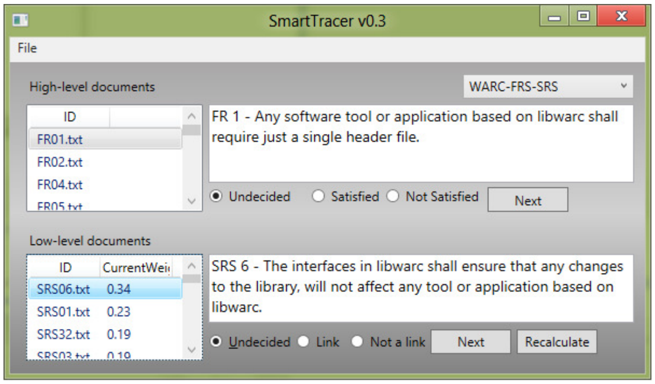
This section describes instrumentation, datasets, participants, study design, and data collection for the study.

A. Instrumentation

To address the research questions in the previous section, an experimental tool called SmartTracer was created to log participant actions while performing a tracing task. Fig. 1 shows a screenshot of SmartTracer. SmartTracer presents a set of high-level documents (HDs) and a set of low-level documents (LDs) to the participant, allowing them to make decisions on each retrieved pair of documents. SmartTracer also allows the participant to make a decision on whether an HD is satisfied by the linked LDs. The simple user interface is designed to allow the participant to concentrate on the task of making decisions on trace links.

A “Recalculate” button in the tool allows the participant to use positive feedback they’ve already given to reorder the

LDs. The Rocchio feedback algorithm [19] with parameters α=1, β=1, γ=0 is used in SmartTracer, meaning that the full term weights of links provided through positive feedback (β=1) are used in the feedback calculation. Negative feedback (γ=0) is not used as studies have shown that standard relevance feedback techniques perform poorly with negative feedback [20, 21]. After the LDs are reordered, the next undecided LD is shown to the participant. The participant can choose not to use the "Recalculate" button and proceed to the next document in the list by clicking on the "Next" button or by directly clicking on another LD in the list. SmartTracer records a number of actions that can be performed by the participant: select an HD or LD, decide on an HD or LD, and press the recalculate button. SmartTracer also records a timestamp for each individual action.



FR01.txt
FR02.txt
FR04.txt
FR05.txt

FR 1 - Any software tool or application based on libwarc shall require just a single header file.

ID	CurrentWei
SRS06.txt	0.34
SRS01.txt	0.23
SRS32.txt	0.19
SRS03.txt	0.19

SRS 6 - The interfaces in libwarc shall ensure that any changes to the library, will not affect any tool or application based on libwarc.

Figure 1. Screenshot of SmartTracer.

B. *Datasets*

Two datasets are used in the study. The first is a set of 42 functional requirements (FRs) and 89 software requirements (SRs) for open source web archive file manipulation tools called WARC[1]. Eighteen (18) FRs that have two or more relevant SRs and all 89 of the SRs are used for the study. The excluded FRs have either one relevant SR that is phrased roughly the same as the FR or do not have any relevant SRs. The candidate TM contains 1535 links with 100% recall and 3.6% precision. The answer set contains 55 links. FRs are used as HDs while SRs are used as LDs.

The second dataset consists of 123 operational requirements (ORs) and 503 system specifications (SSs) for an Unmanned Aerial Vehicle Tactical Control System (UAVTCS)[2]. A subset of 20 ORs and 264 SSs is used for the study. The candidate TM contains 4621 links with 100% recall and 1.8% precision. The answer set contains 81 links. ORs are used as HDs while SSs are used as LDs.

Candidate TMs are generated using a vector space model with term frequency and inverse document frequency weighting. The original TMs included in both datasets were revised by multiple graduate and undergraduate students until full consensus was reached on each link in the answer set. The original authors of the artifacts were not available to provide feedback on the revisions.

C. *Participants*

A convenience sampling procedure is used to recruit study participants. Participants are mostly junior- and senior-level undergraduate and graduate students in computer science from the University of Kentucky (UK) and graduate students in computer science from DePaul University and Cal Poly. The graduate students at UK and DePaul are mostly part-time graduate students that work full time in industry. Most graduate students at Cal Poly are full-time students with prior experience in industry through part-time or full-time employment or summer internships. The study was conducted during regular class time in a lab for three groups at UK. Participants at DePaul and Cal Poly were given instructions in a group setting but performed the tracing task on their own time.

D. *Study Design*

Table I presents the distribution of participants and datasets for the study. Participants were given the same starting candidate TMs. Participants were blocked on grade level (graduate and undergraduate) and dataset (WARC and UAVTCS) to reduce the effects of those factors on the dependent variables in Table II. Recall is not included as a dependent variable as it is the product of potential recall and sensitivity. A fourth university was to participate in the study (using the UAVTCS dataset) but was unable to recruit enough student participants, resulting in the unbalanced study groups. Almost twice as many graduate students worked with the WARC dataset compared to the UAVTCS dataset.

TABLE I. PARTICIPANT INFORMATION

Location	*# of participants*	*Dataset*
University Y Group A (grad)	6	WARC
University X Group B (und)	10	WARC
University Z Group E (grad)	8	WARC
University X Group C (und)	15	UAVTCS
University X Group D (grad)	8	UAVTCS

TABLE II. DEPENDENT VARIABLES

Variable	*Scale*
Potential recall	Ratio
Sensitivity	Ratio
Precision	Ratio
Effort distribution	Ratio

Table III presents independent variables used in the study. *Software engineering (SE) experience* is based on the number of SE courses taken in college and industry experience. *Tracing experience* is based on the number of

[1] http://code.google.com/p/warc-tools/

[2] http://www.fas.org/irp/program/collect/uav_tcs.htm

tracing tasks performed to date. *Confidence in tracing* is a 5-point Likert scale of the participant's confidence in performing the tracing task with one being the lowest and five being the highest.

TABLE III. INDEPENDENT VARIABLES

Variable	*Abbreviation*	*Scale*
Grade Level	Grade	Nominal {Undergrad, Grad}
Software Engineering Experience	SEExp	Ordinal {0, 1, 2}
Tracing Experience	TRExp	Ordinal {0, 1, 2}
Confidence in tracing	Confidence	Ordinal {1, 2, 3, 4, 5}
Dataset	Dataset	Nominal {WARC, UAVTCS}
Location	Location	Nominal {UK, CP, DP}
Time to perform tracing task	Time	Ratio {Minutes}
Link Strategy	LinkStrategy	Nominal
Level of relevance feedback	Feedback	Ordinal {0, 1}
Average number of links viewed	LinksViewed	Ratio {Links}

E. Data Collection

Prior to the study, participants were given a pre-study survey with questions regarding their software engineering background, prior software engineering classes taken, their tracing experience, as well as an assessment of their confidence in performing the tracing task. Each participant was given a user ID to identify them in the study. Each participant was given a short training session on how to use the tracing tool. The overall goal of the study was explained and instructions were given for them to be mindful of how they perform the task.

After completing the training, participants at University X were given 45 to 60 minutes to complete the tracing task. Upon completing the tracing task, participants submitted the final TM and trace logs. The logs track the time spent on each action and record the number of feedback recalculations per HD. Participants at University Y and Z performed the task on their own time.

A post-study survey was given after completing the task, asking each participant to record: their overall tracing strategy, when they decided to stop looking for additional links, feedback on what additional tool features might be useful, and their confidence in performing tracing after performing the task.

1) Data collection for RQ1 and RQ3: Potential recall, sensitivity, recall, precision, effort distribution, and final TM size are calculated at each participant's decision point. Snapshots of participant decisions are captured at the nearest five-minute mark with the time of the last decision rounded down to the nearest five-minute mark to plot the charts in Figs. 3 and 4.

2) Data collection for RQ2: Pre-study surveys are reviewed and coded into the scales in Table III. The level of relevance feedback is coded based on whether or not participants consistently used the "Recalculate" button.

3) Data collection for RQ4: Trace logs and post-study surveys are analyzed to identify strategies used by participants and compared with data collected for RQ1.

V. THREATS TO VALIDITY

Threats to conclusion validity are issues that affect the credibility of the conclusions reached from the results. A possible Hawthorne effect (change in behavior when one is being observed) was introduced when participants were told that their actions were being recorded and that they were to be mindful of how they performed the tracing task. The small number of participants per group possibly limits the significance of the results, which is partially mitigated by running the study at multiple sites.

Threats to internal validity relate to whether the trends we are seeing are indeed causal. Explaining the study procedures and having the participants perform the tracing task in a single session possibly influences the results of the study, partially mitigated by having participants in two of the three locations perform the task on their own time.

Threats to construct validity involve questions of whether the study is designed to correctly measure what we set out to measure. A possible bias would be the use of a simple tracing tool that is not representative of full-featured tracing tools in use today. This decision was intentionally made to reduce possible nuisance factors that may arise from tool usage. A possible selection threat exists due to the selection of HDs used in both datasets in order to influence the performance of the relevance feedback mechanism.

Threats to external validity deal with the generalization of results to other domains. Threats of this nature are mitigated through the use of two datasets from very different domains; a mission-critical system and a web content archival tool. The results of this study may not be generalizable to tracing tasks using other software engineering artifacts. Use of student participants does not significantly affect results as found in previous studies [3], though this study includes a number of participants who have industry experience.

VI. RESULTS

This section provides answers to the research questions formulated in Section III[3]. In group C, three participants were dropped from the study due to partial loss of results e.g., results were submitted without log files.

A. Results for Research Question 1

Table IV shows the average potential recall, average sensitivity, average recall, average precision, and average effort distribution by dataset and grade level. Each

[3] Due to space restrictions, detailed tables representing results of analysis were not included. They can be found in the full version of the paper at http://selab.netlab.uky.edu/TechReports/techreport520-12.pdf

participant, on average, saw 79% of all true links in the candidate TM but only accepted 77% of them, resulting in the average final TM having 61% recall. This is a significant 18 percentage point drop due to participants not reviewing some of the true links and rejecting some of the true links. The final TMs had an average 54% precision, meaning that 46% of the links in the TM were false links incorrectly accepted by the participants. Participants viewed, on average, close to five times as many false links as true links.

A significant difference in sensitivity exists between WARC and UAVTCS datasets (two-sample t-test, alpha=0.05, p=0.042), while the differences in other measures (recall, potential recall, precision, and effort distribution) are not statistically significant. A statistically significant difference in *sensitivity* and *recall* exists between grade levels (A, D, E vs. B, C), with undergraduates having higher averages (two-sample t-test, alpha=0.05, p=0.02 for sensitivity and p=0.004 for recall). Between datasets, grade level had *no statistically significant effect on any of the dependent variables* for UAVTCS. Grade level had a statistically significant effect on *sensitivity, recall,* and *precision* on WARC: graduates had higher average precision while undergraduates had higher average recall, which indicates that undergraduates tended to accept more links than graduates. For the UAVTCS dataset, however, graduate and undergraduate students performed similarly without any significant difference in any of the measures.

TABLE IV. STATISTICS FOR EACH PARTICIPANT GROUP

	Pot. Recall	*Sensitivity*	*Recall*	*Precision*	*Eff. Dist.*
Overall	**0.79**	**0.77**	**0.61**	**0.54**	**4.8**
Dataset					
WARC	**0.81**	**0.73**	**0.60**	**0.56**	**4.4**
Undergrad. (B)	0.83	0.78	0.65	0.46	5.8
Grad. (A, E)	0.79	0.70	0.56	0.63	3.4
UAVTCS	**0.78**	**0.82**	**0.63**	**0.51**	**5.3**
Undergrad. (C)	0.82	0.85	0.70	0.52	2.8
Graduate. (D)	0.71	0.78	0.53	0.49	9.0
Grade Level					
Undergrad.	**0.83**	**0.82**	**0.68**	**0.50**	**4.2**
WARC (B)	0.83	0.78	0.65	0.46	5.8
UAVTCS (C)	0.82	0.85	0.70	0.52	2.8
Grad.	**0.76**	**0.73**	**0.55**	**0.58**	**5.4**
WARC (A, E)	0.79	0.70	0.56	0.63	3.4
UAVTCS (D)	0.71	0.78	0.53	0.49	9.0

B. Results for Research Question 2

We observed differences in analyst performance based on environmental factors: the combination of the dataset they were working with and, for WARC, their specific group. Among the *personal characteristics* of participants, grade level had statistically significant effect on participant performance. Additionally, for the UAVTCS dataset, *tracing experience*, when controlled for *software engineering experience* and *post-study tracing confidence,* had a significantly negative effect on sensitivity.

Statistical analysis of precision, time spent tracing, and effort distribution revealed a significant relationship between those three measures. Multiple regression showed that when considering both datasets, time to trace and effort distribution jointly explain 41.6% of precision (with r^2_{adj} = 38.7), which is statistically significant. A significant negative correlation with precision exists between both time to trace (-0.52) and effort distribution (-0.57).

Looking at individual datasets, however, provided some additional insight. For the WARC dataset, multiple regression showed effort distribution to be significant for precision (r^2 = 36.7, r^2_{adj} = 30.6) when controlling for time. At the same time, when controlling for effort distribution, time spent tracing ***was not a significant influence*** on precision. For UAVTCS, the situation was reversed. Controlling for time, multiple regression showed effort distribution to be not significant for precision, while controlling for effort distribution, time spent tracing ***was a significant influence***. We observed a similar discrepancy between graduates and undergraduates. For graduates, multiple regression showed effort distribution to influence precision significantly when controlling for time (r^2 = 58.1, r^2_{adj} = 52.9), while time was not a significant influence on precision. For undergraduates, the opposite held.

We observed that on the WARC dataset, the increase in the number of observed links and thus the decrease in precision *primarily came* from participants who viewed more false candidate links, but it was not affected by how long the participants worked on the tracing task. On the other hand, for UAVTCS dataset, increase in the number of links viewed and decrease in precision primarily came from participants electing to spend more time viewing links, but not necessarily viewing more false candidate links percentage-wise. Similarly, graduates decreased their precision whenever they wound up viewing more false candidate links, but not when they worked longer. Undergraduates decreased their precision with time spent tracing, but not with how many more false candidate links they saw.

C. Results for Research Question 3

Fig. 2 is a matrix visualization of the decisions that participants made on true links for both datasets, allowing us to explore individual participant behavior. Each row represents a participant and each column represents a true link in the candidate TM (20x81 for UAVTCS, 24x51 for WARC). True links that were never seen are marked in black and true links that were seen but rejected are marked in gray. The remaining 'white space' represents true links that were correctly marked.

For the UAVTCS dataset, twelve links were never seen by more than half of the participants, of which three links were never seen by all participants, and one link was only seen by one participant (as indicated by black vertical line segments). Most of these links had low weights and the HD in each of these links was also linked to a number of other

LDs that fully satisfied each respective HD. One participant did not see more than 90% of the true links and another missed about 45% of the true links (both from Group D). Both participants spent most of their time on a few HDs and responded in the post-study survey that they did not feel sufficiently trained on the task. Two other participants each did not see about 25% of the true links but the missing links were spread out over the dataset (as indicated by black horizontal line segments). The logs show that both participants viewed an average of 6-7 LDs per HD, missing any additional links further down the list. These twelve links and four participants together account for about 18% out of the 22% of lost potential recall.

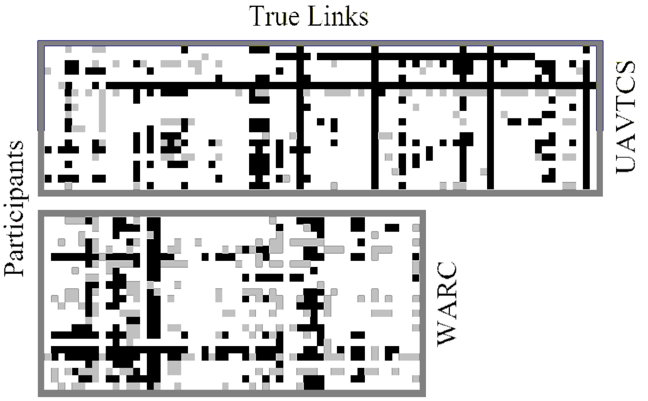


Figure 2. Matrix visualization of participant true link decisions

For the WARC dataset, all true links were seen by at least one participant, but six of those links were never seen by more than half of the participants (also due to the same reason as the twelve links in UAVTCS, although some were somewhat related). Three participants did not see more than half of the true links and two participants did not see about 35% of the true links (also due to viewing anywhere from 4-8 LDs per HD). Five participants rejected at least one-third of the true links that they saw, and fourteen true links were rejected by at least 25% of the participants. Most of these rejections were because the LDs in each link were only somewhat relevant to their respective HDs, causing some participants to waver in their decision.

Fig. 3 shows individual participant performance on the WARC dataset by group on a lattice chart, tracking potential recall, distribution and TM size (on secondary vertical axis) on the lower cell at five-minute intervals. The number of links in the answer set is represented as a line intersecting each bar representing TM size at each time interval. Participant results are sorted by increasing TM size.

For example, participant B4 had about 5% recall and 65% precision five minutes into the tracing task and correctly identified all the true links seen up to that point. Thirty minutes into the task, recall went up to about 30% while precision dropped to about 30% as well. At about 50 minutes (at the end of the task), recall went up to 60%, precision increased to about 40%, but potential recall was about 90%, i.e., the participant missed about 30% of the true links they saw (66% sensitivity). Effort distribution steadily increased but leveled off half way through the tracing task, coinciding with the increased recall and decreased sensitivity (seeing more true links but rejecting some of them as well).

Similarly, Fig. 4 shows participant performance on the UAVTCS dataset. Participant D8 achieved about 5% recall and 60% precision five minutes into the task with 100% sensitivity. After 30 minutes, precision and sensitivity plunged to about 20% and 30%, respectively. Additional log analysis revealed that the participant spent about ten minutes on the first two HDs looking through many LDs, as indicated by the spike in effort distribution. The participant then started skimming through the remaining HDs, as indicated by the plunge in sensitivity, adding false links into the final TM, as indicated by the plunge in precision, before spending another 20 minutes on the first two HDs, as indicated by the stagnant recall. The second half of the time saw a sharp increase in recall as the participant went through the remaining HDs much faster, accepting many of the true links seen earlier but continuing to accept many false links, as indicated by increasing recall and sensitivity while lowering precision. The participant ended the task with a final TM containing 246 links with about 80% recall, 94% sensitivity, and 30% precision. A number of participants showed similar trends where significant differences between potential recall and recall early in the tracing task (B1, E2, D4, D8) can be attributed to participant actions of reading through each HD first before starting to mark links. This can be seen mostly when sensitivity starts low or drops suddenly before increasing steadily as the task progresses.

WARC participants who performed well (A4, B3, E2) averaged about 75% recall, 59% precision, and 83% sensitivity while UAVTCS participants who performed well (D1, C1, C2) also averaged about 76% recall, 58% precision, and 84% sensitivity. These participants increased recall at a consistent pace, while keeping other measures stable.

In Fig. 4, participants D2 and D6 did not complete the tracing task as they spent most of their time on the first few HDs, as indicated by the rapid increase in effort distribution. Participant D2 changed strategies about 35 minutes into the task (effort distribution peaked and started coming down) and managed to achieve about 50% recall at the end of the task. Participant D6, however, spent almost all of their time reviewing false links. Both participants had low precision from adding many false links into the final TM.

D. Results for Research Question 4

SmartTracer directs its users to consider candidate links by HD, consistent with other tracing software used in similar studies [4, 10, 22]. We observed that participants articulated a number of different strategies for selecting links. The observed strategies are briefly outlined below.

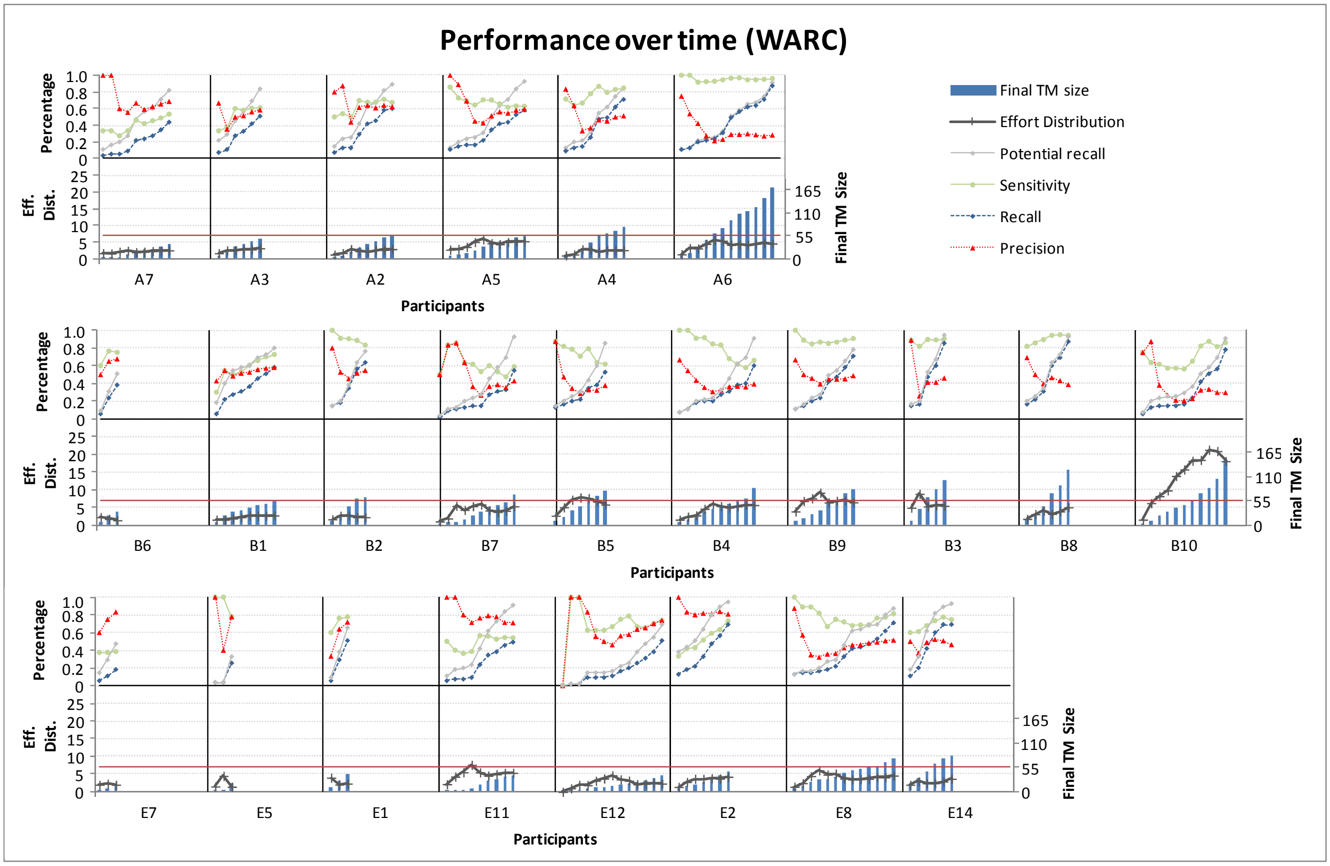


Figure 3. Participant performance over time on WARC.

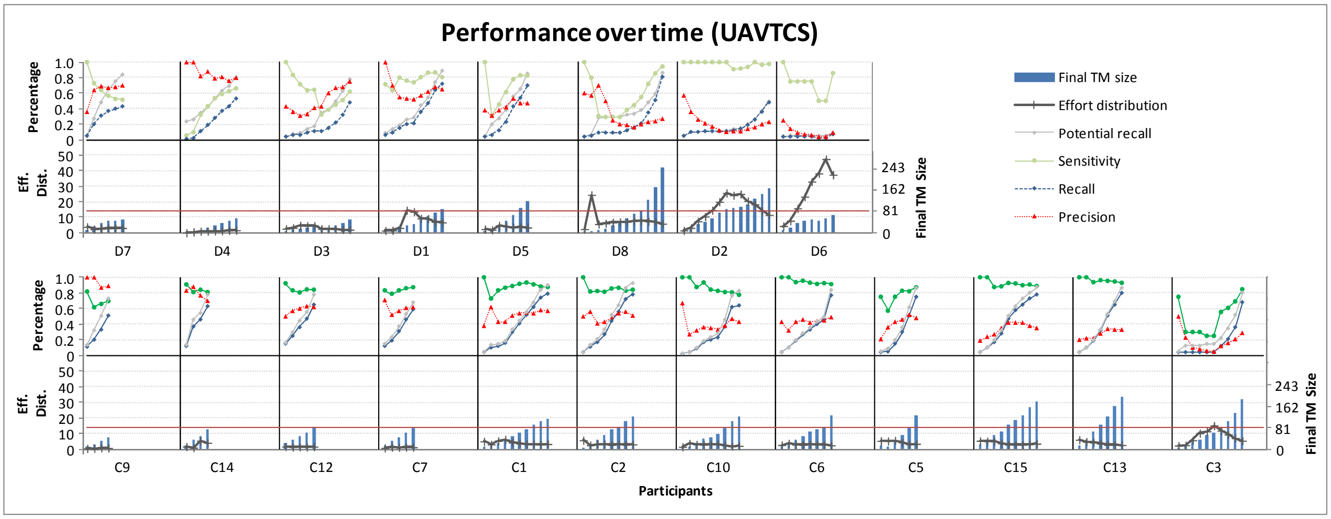


Figure 4. Participant performance over time on UAVTCS.

First good link. Participants looked through the list of candidate links associated with a single HD only until they discovered the *first good link* (one they think satisfies the HD) before switching to the next HD.

Accept-focused. Participants tended to only submit *accept* decisions for candidate links, not bothering to reject links in SmartTracer, possibly understanding that not accepting a link is essentially equivalent to rejecting it.

Preview. Participants *previewed their task* by reading through the list of HDs and some LDs before starting to make any decisions on links.

Iterative. Participants revisited most of the HDs *more than once* to review or change their decisions.

Some participants used ***multiple*** strategies. For some, a distinct strategy could not be established (***Unknown***). We also looked at whether participants ***used feedback*** ("Recalculate" button) during their work. Participants were divided into three categories based on the ***average number of links*** per HD they considered: *less than 10, 10 to 20,* and *more than 20.*

Table V presents the results of the study broken down by participant strategy. For example, two participants using the *First good link* strategy achieved, on average, 40% potential recall, 22% recall, 81% precision, and 1.9 effort distribution. This strategy led to fast task completion (average 15 minutes) but at the cost of not observing a significant number of true links. On the other hand, participants who used multiple strategies were able to achieve high potential recall (87% on average) with moderate (4.4. on average) effort distribution.

A significant difference in potential recall and recall exists between those that used feedback and those who didn't, but most of the difference can be attributed to the two participants who used the *First good link* strategy and the participant who only observed two HDs (neither used feedback.) When comparing participants by the average number of links viewed, the *10-20* strategy was most common and achieved high potential recall and moderate effort distribution.

TABLE V. RESULTS FROM TRACING STRATEGIES

Strategy	Pot. Recall	Recall	Precision	Eff. Dist.	# of particip.	Time Spent
Link Selection						
First good link	40%	22%	81%	1.9	2	15
Accept-focused	79%	65%	64%	2.3	4	30
Preview	81%	47%	67%	3.4	2	40
Iterative	85%	67%	53%	2.9	4	34
Multiple	87%	68%	60%	4.4	5	43
Unknown	80%	62%	49%	5.9	27	44
Feedback						
Used feedback	84%	66%	53%	4.3	31	43
No feedback	68%	47%	56%	5.9	13	33
Links Viewed						
Under 10	67%	46%	72%	1.8	11	28
10-20	87%	67%	51%	3.9	26	42
20+	72%	60%	38%	12.6	7	54

VII. OBSERVATIONS

From the results of the previous research questions, we observed that links are more likely to be missed when there are multiple LDs for an HD and when some of those LDs fully satisfy the HD. This possibly causes participants to decide at some point that they have enough LDs to mark the HD "satisfied." This is especially characteristic of those who never investigate far down the ranked candidate link list. This insight may provide guidance for when other software artifacts are generated from requirements. As a software artifact is generated to satisfy a requirement, a search for other similar artifacts would determine if there is a need for the additional artifact or a modification of an existing artifact will suffice.

We also observed that participant decisions fell into three categories: obvious true links, obvious false links, and troublesome gray links, i.e., links that seem to cause significant amount of deliberation for the analysts. Without proper training and direction, analysts may spend too much time on these links and may vary in how selective they are in determining what constitutes a link, possibly because they do not really know how the TM is to be used. The issue of gray links is also a concern for researchers when building answer sets (Does the answer set include gray links or not?). We posit that an analyst that has knowledge of how the final TM is going to be used would be better equipped to reject or accept gray links they find to trigger the appropriate successor activities to resolve those concerns. One way to study these decisions would be to have a third decision option that separates these gray links from the "Yes it's a link" and "No it's not a link" decisions. We can then measure how accurate the analyst is at making decisions on links that they think are obvious versus links they think are "suspect."

One of the things we can do about the analyst other than to *embrace* them is to *change* them [1]. When TM usage is defined, analysts can be trained to produce final TMs that fit the desired final TM characteristic based partially on the final TM size. A final TM size that is close to the true TM size will have nearly equal precision and recall. Given an estimate of the true TM size (based on historical data or a starting estimate), analysts may be more aware of their selectiveness when adding links into the final TM, adjusting the thresholds they apply to links as they proceed through the tracing task. Learning and applying tracing strategies to tracing tasks is another way to *change* the analyst. Once we know how tracing strategies affect results, analysts will be able to apply appropriate strategies for the desired tracing task outcomes.

VIII. CONCLUSIONS AND FUTURE WORK

An important step in improving traceability practice is to understand how analysts work with TMs. In this study, we presented a set of measures that focus on the quality of the analyst working to produce final TMs, visualizing and analyzing analyst trace logs to detect trends. We measured how environmental, personal, and behavioral factors affected results, finding significant interactions between time spent on tracing, effort distribution, and precision. We visualized analyst trace logs to show where analysts make correct and incorrect decisions on true links, and then analyzed the possible causes for the links that were never seen and the links that were rejected. Actual analyst tracing strategies obtained from trace logs provide insight into how analysts performed the tracing task.

Based on these results, we now have an initial measure of the imperfect analyst that misses roughly one out of every four true links they observe (77% sensitivity). Future studies using relevance feedback will measure how simulated techniques fare using the tracing strategies mined from trace logs along with imperfect feedback to validate technique effectiveness. Analysts are more likely to miss links when TMs have multiple relevant LDs per HD. Future studies will focus on ways to encourage the analyst to continue looking for these additional links. Future studies will also include the investigation of a "gray link" decision as a possible decision during the tracing task where the analyst is given guidance on final TM usage. There is still much to be done in the study of the analyst, and further findings will continue to lead to the improvement of tracing as a practice.

Acknowledgment

We thank Jane Cleland-Huang, Maureen Doyle, and all the study participants. This study was funded in part by the National Science Foundation under grant CCF-0811140.

References

[1] D. Cuddeback, A. Dekhtyar, J. H. Hayes, J. Holden, and W.-K. Kong, "Towards Overcoming Human Analyst Fallibility in the Requirements Tracing Process (NIER Track)," in *Proceedings of the 33rd International Conference on Software Engineering*. New York, NY, USA: ACM, 2011, pp. 860–863.

[2] D. Cuddeback, A. Dekhtyar, and J. H. Hayes, "Automated Requirements Traceability: The Study of Human Analysts," in *Requirements Engineering Conference (RE), 2010 18th IEEE International*, Oct. 2010, pp. 231 –240.

[3] A. Dekhtyar, O. Dekhtyar, J. Holden, J. H. Hayes, D. Cuddeback, and W.-K. Kong, "On Human Analyst Performance in Assisted Requirements Tracing: Statistical Analysis," in *Requirements Engineering Conference (RE), 2011 19th IEEE International*, Sept. 2011, pp. 111 –120.

[4] W.-K. Kong, J. H. Hayes, A. Dekhtyar, and J. Holden, "How Do We Trace Requirements: An Initial Study of Analyst Behavior in Trace Validation Tasks," in *Proceedings of the 4th International Workshop on Cooperative and Human Aspects of Software Engineering*, ser. CHASE '11. New York, NY, USA: ACM, 2011, pp. 32–39.

[5] J. H. Hayes, A. Dekhtyar, S. Sundaram, and S. Howard, "Helping Analysts Trace Requirements: An Objective Look," in *Requirements Engineering Conference, 2004. Proceedings. 12th IEEE International*, Sept. 2004, pp. 249 – 259.

[6] A. Dekhtyar, J. H. Hayes, and J. Larsen, "Make the Most of Your Time: How Should the Analyst Work with Automated Traceability Tools?" in *Predictor Models in Software Engineering, 2007. PROMISE'07: ICSE Workshops 2007. International Workshop on*, May 2007, p. 4.

[7] A. Egyed, F. Graf, and P. Grünbacher, "Effort and Quality of Recovering Requirements-to-Code Traces: Two Exploratory Experiments," in *Requirements Engineering Conference (RE), 2010 18th IEEE International*, Oct. 2010, pp. 221 –230.

[8] G. Antoniol, G. Canfora, G. Casazza, A. De Lucia, and E. Merlo, "Recovering Traceability Links Between Code and Documentation," *IEEE Trans. Softw. Eng.*, vol. 28, no. 10, pp. 970 – 983, Sept. 2002.

[9] A. De Lucia, F. Fasano, R. Oliveto, and G. Tortora, "ADAMS Re-Trace: A Traceability Recovery Tool," in *Software Maintenance and Reengineering, 2005. CSMR 2005. Ninth European Conference on*, Mar. 2005, pp. 32 – 41.

[10] J. H. Hayes, A. Dekhtyar, and S. Sundaram, "Advancing Candidate Link Generation for Requirements Tracing: the Study of Methods," *Software Engineering, IEEE Transactions on*, vol. 32, no. 1, pp. 4 – 19, Jan. 2006.

[11] A. Marcus, J. I. Maletic, and A. Sergeyev, "Recovery of Traceability Links between Software Documentation and Source Code," *International Journal of Software Engineering & Knowledge Engineering*, vol. 15, no. 5, pp. 811 – 836, 2005.

[12] A. De Lucia, R. Oliveto, and G. Tortora, "IR-Based Traceability Recovery Processes: An Empirical Comparison of "One-Shot" and Incremental Processes," in *Automated Software Engineering, 2008. ASE 2008. 23rd IEEE/ACM International Conference on*, Sept. 2008, pp. 39 –48.

[13] W.-K. Kong and J. H. Hayes, "Proximity-based Traceability: An Empirical Validation using Ranked Retrieval and Set-based Measures," in *Empirical Requirements Engineering (EmpiRE), 2011 First International Workshop on*, Aug. 2011, pp. 45 –52.

[14] X. Zou, R. Settimi, and J. Cleland-Huang, "Phrasing in Dynamic Requirements Trace Retrieval," in *Computer Software and Applications Conference, 2006. COMPSAC '06. 30th Annual International*, vol. 1, Sept. 2006, pp. 265 –272.

[15] J. Cleland-Huang, A. Czauderna, M. Gibiec, and J. Emenecker, "A Machine Learning Approach for Tracing Regulatory Codes to Product Specific Requirements," in *Software Engineering, 2010 ACM/IEEE 32nd International Conference on*, vol. 1, May 2010, pp. 155 –164.

[16] R. Settimi, J. Cleland-Huang, O. Ben Khadra, J. Mody, W. Lukasik, and C. DePalma, "Supporting Software Evolution through Dynamically Retrieving Traces to UML Artifacts," in *Software Evolution, 2004. Proceedings. 7th International Workshop on Principles of*, Sept. 2004, pp. 49 – 54.

[17] M. Gibiec, A. Czauderna, and J. Cleland-Huang, "Towards Mining Replacement Queries for Hard-to-Retrieve Traces," in *Proceedings of the IEEE/ACM international conference on Automated software engineering*, ser. ASE '10. New York, NY, USA: ACM, 2010, pp. 245–254.

[18] A. De Lucia, R. Oliveto, and P. Sgueglia, "Incremental Approach and User Feedbacks: a Silver Bullet for Traceability Recovery," in *Proceedings of the 22nd IEEE International Conference on Software Maintenance*. Washington, DC, USA: IEEE Computer Society, 2006, pp. 299–309.

[19] J. Rocchio, *Relevance Feedback in Information Retrieval*. Prentice-Hall Inc., 1971, ch. 14, pp. 313–323.

[20] M. D. Dunlop, "The Effect of Accessing Nonmatching Documents on Relevance Feedback," *ACM Trans. Inf. Syst.*, vol. 15, pp. 137–153, Apr. 1997.

[21] X. Wang, H. Fang, and C. Zhai, "A Study of Methods for Negative Relevance Feedback," in *Proceedings of the 31st annual international ACM SIGIR conference on research and development in information retrieval*, ser. SIGIR '08. New York, NY, USA: ACM, 2008, pp. 219–226.

[22] J. Lin, C. C. Lin, J. Huang, R. Settimi, J. Amaya, G. Bedford, B. Berenbach, O. Khadra, C. Duan, and X. Zou, "Poirot: A Distributed Tool Supporting Enterprise-Wide Automated Traceability," in *Requirements Engineering, 14th IEEE International Conference*, Sept. 2006, pp. 363 –364.