

# Evaluating Assistance of Natural Language Policy Authoring

Kami Vaniea<sup>2</sup>, Clare-Marie Karat<sup>1</sup>, Joshua B. Gross<sup>3</sup> John Karat<sup>1</sup>, and Carolyn Brodie<sup>1</sup>

<sup>1</sup>IBM T.J. Watson Research  
Hawthorne, NY 10532 USA  
ckarat, jkarat,  
brodiec@us.ibm.com

<sup>2</sup>Carnegie Mellon University  
Pittsburgh, PA 15217 USA  
kami@cs.cmu.edu

<sup>3</sup>The Pennsylvania State University  
University Park, PA 16802  
jgross@ist.psu.edu

## ABSTRACT

The goal of the research study reported here was to investigate policy authors' ability to take descriptions of changes to policy situations and author high-quality, complete policy rules that would parse with high accuracy. As a part of this research, we investigated ways in which we could assist policy authors in writing policies. This paper presents the results of a user study on the effectiveness of providing syntax highlighting in a natural language policy authoring interface. While subjects liked the new interface, they showed no improvement in accuracy when writing rules. We discuss our results in terms of a three phase authoring process that users move through when authoring or modifying policies. We describe this process, discuss why and how our interface failed to support it and make recommendations to designers on how to better support this process.

## Categories and Subject Descriptors

H5.2. Information interfaces and presentation: User interfaces.  
K4.1. Public policy issues: Privacy

## General Terms

Design, Security, Human Factors.

## Keywords

Privacy, privacy policies, natural language interfaces, social and legal issues, design process, syntax highlighting.

## 1. INTRODUCTION

Policies, defined as collections of goals and processes for achieving them, can be seen as important elements in the operation of any organization. An organization's policy is vital since it functions as both the organization's external reassurance to its customers that their data will be protected and an internal guide for procedure and practices. Having good tools that allow organizations to author, implement and monitor their policies help them to maintain the balance between high level policies and organizational practices. In this work we discuss the use of syntax highlighting to augment privacy policy authoring using the

SPARCLE Policy Workbench.

While policies support a wide range of aspects of operations in an IT organization (e.g., security, network management, business process), we have focused on privacy policies. This is partly because privacy policy rules have been subjected to analysis resulting in the identification of standard components of such rules, and partly because large gaps still exist between high level policies and operational practices. A goal of our research has been to develop approaches which can assist an organization in going from high level policy specifications (e.g., We will respect the privacy rights of our customers.) to low level representation such as XML which can be used to implement the policies through automation [5], [9], [10].

Our work to date suggests that allowing policies to be expressed in natural language, and iteratively transforming them into machine readable formats is a valuable approach [14]. Authoring a policy becomes a process of writing and modifying rules. This research builds on our initial research which focuses on unexamined aspects of the authoring process such as how differing interaction methods can help users to detect errors and modify rules. This is a wide area of research and the results presented in this paper represent our first steps in this direction. The primary design approaches employed were (1) to provide more immediate feedback to authors on how well their rules were understood (i.e., parsed) by the system, and (2) to provide syntax highlighting of elements of the text entered by authors, that is, the tool would distinguish different elements of a rule through differential highlighting.

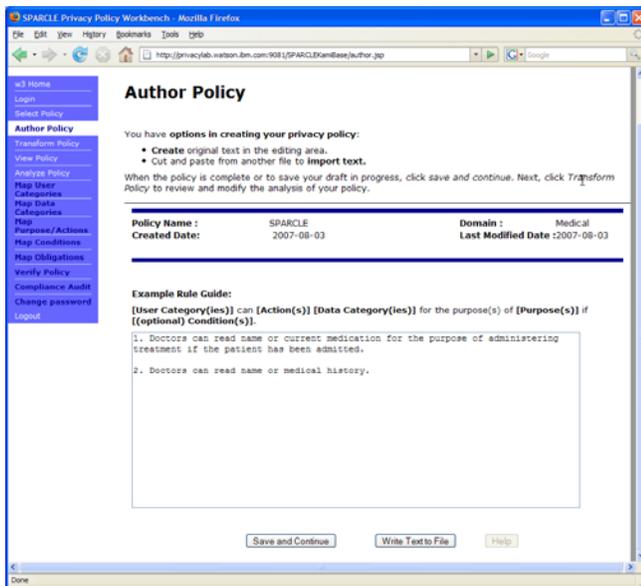
The remainder of this paper is structured as follows. We describe the policy structure used in this work (Section 1.1). We describe the SPARCLE Policy Workbench and the various interfaces associated with policy authoring (Section 1.2). We propose a new interface that uses syntax highlighting (Section 1.3). The study methodology is described (Section 2) followed by a description of the Qualitative (Section 3.1) and Quantitative (Section 3.2) results. Finally, we discuss our results and put forth a theory on the process of policy authoring (Section 4).

### 1.1 Policy Rule Elements

In this section we will review the different elements of a privacy policy rule and introduce terminology that will be used in the remainder of the paper.

Copyright is held by the author/owner. Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee.

Symposium On Usable Privacy and Security (SOUPS) 2008, July 23-25, 2008, Pittsburgh, PA, USA.



**Figure 1. The Author page allows a user to write policies using guided natural language.**

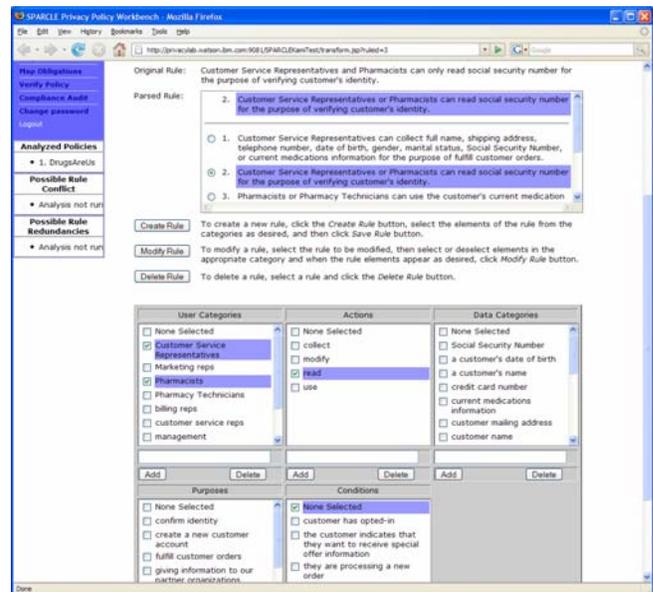
We define an implementable privacy policy rule as a rule that can be defined for automated enforcement through technology. In our model, a privacy policy rule is a single sentence constructed using five different elements: user, action, data, purpose and conditions. Each element may be composed of one or more values. In the example “Doctors or nurses can access medical records for the purpose of administering treatment” Doctors and nurses are both values of the element User.

The User element describes the individual or role given access permissions by the rule. An Action is defined as the things that a user can do with the data specified in the rule. Data describes the electronic record or data field(s) on which the user can perform the action. The Purpose specifies the reason that the user can perform the stated action on the data. Finally, the Condition, which is optional, specifies the pre and post permission requirements (conditions and obligations) for the rule to be applicable. Previous research found that most users do not differentiate between conditions and obligations. Therefore, both items are referred to as conditions in the user interface, however, they are handled separately in implementation [5].

## 1.2 Overview of the SPARCLE Policy Workbench

The SPARCLE Policy Workbench is designed to help organizations successfully implement their privacy policies. The workbench provides them with tools to help create understandable policies, link their written privacy policies with the implementations of the policy across their IT configurations, and then help them to monitor the enforcement of the policy through internal compliance audits. In this paper we will be focusing on the policy authoring portion of the SPARCLE Workbench.

The authoring portion of the SPARCLE Policy Workbench consists of three pages which we will refer to as *Author*, *Transform* and *View*. In this section we will provide a short overview of each page. For a more in-depth discussion and description, the reader is referred to prior work on the SPARCLE Policy Workbench [9], [11].



**Figure 2. The Transform page allows a user to view the output of the natural language transformation and make changes as needed.**

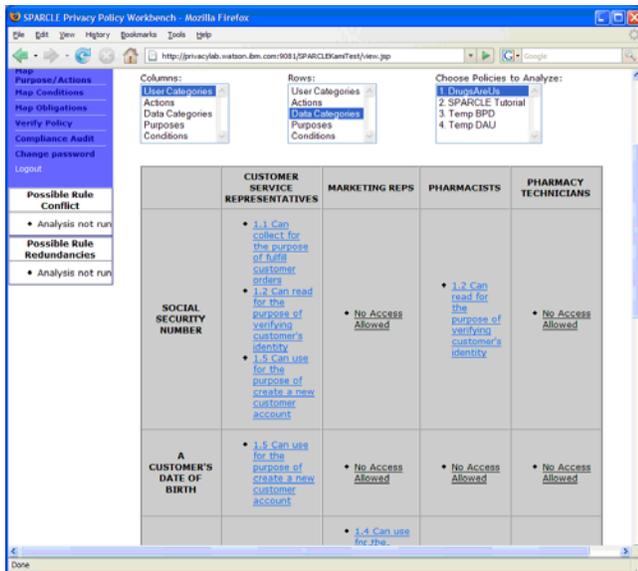
The *Author* page, pictured in **Error! Reference source not found.**, allows a policy author to compose or modify a policy using guided natural language. The page features a text area where the author can enter natural language policies. The page also provides an example rule guide which serves the dual purposes of reminding the author of the five elements in an implementable rule and the recommended sentence structure for rules to parse with high accuracy and precision.

Once the policy text meets her satisfaction the user can use the *Transform* page, pictured in **Error! Reference source not found.**, to see the result of the policy transformation. For each rule, this page shows the results of the natural language parsing of the policy text and the identification of policy rule elements. The page contains five boxes, one for each policy rule element, containing a list of possible values for that element. If the author uses a value not previously on the list, it is automatically added to the list. The author can also manually add values to the list. The author can then modify the parsed policy rules by selecting new values for the elements. Changes made in the *Transform* page are reflected in the *Author* page allowing the user the flexibility of writing and editing in either page.

When the user is happy with the content of her rules she can use the *View* page, pictured in Figure 5, to review a visualization of the entire resulting policy. The visualization consists of a 2-dimensional table with user and data categories as the default policy elements on the axes. The cell then contains the content of the rule based on the other three elements. The user can control which elements are on the axes through use of drop-down lists at the top of the table.

## 1.3 Experimental Interface

Previous studies of the SPARCLE interface have shown it to be an effective authoring tool [9], [10]. Specifically, users enjoyed the ability to choose between authoring in natural language and using a more structured interface. They also appreciated the value of



**Figure 3. The View page assists the user in visualizing a policy by showing it in a configurable 2-dimensional table.**

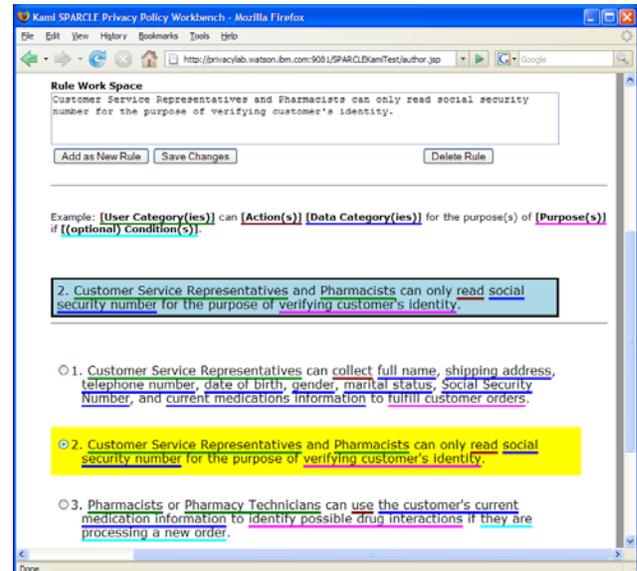
being able to write a machine readable privacy policy using natural language.

Even though users liked the natural language interface, they occasionally became frustrated with the tool when their natural language rules failed to parse as accurately as they would like them to. The tool was designed and developed based on an expected syntax for privacy and security policy access-control policies, with some variations for flexibility. Based on these observations, a goal was identified to design an interface that would both support natural language input and allow users to more quickly become proficient in writing policy rules with high parsing accuracy.

We hypothesized that providing syntax highlighting on the *Author* page as a form of rapid feedback about rule parsing accuracy would improve user performance in authoring high quality rules. Syntax highlighting has been used to aid in the development of natural language grammars in the past. It has also been used for indicating confidence in speech recognition [16], highlighting syntax in code development environments [15] and is commonly used by such development environments as Eclipse, Rational Application Developer and Emacs. To the best knowledge of the authors, there has been no study on the effectiveness of using syntax highlighting for policy authoring tasks.

We hypothesized that providing rapid and clear feedback on the parsing accuracy would improve authoring performance in two ways. First we reasoned that by providing quick feedback the user would be able to experiment with different wordings of the rules and learn what wordings worked and which didn't. Second, we also hypothesized that this rapid feedback would reduce the amount of time the user spent reviewing rule parsing results as this review could occur in parallel during the policy authoring task.

The syntax highlighting consists of colored underlining of elements in the policy rules. Each color corresponds to a specific element and all values associated with that element are underlined in the same color. If any required element is missing from a rule,



**Figure 4. The new Author interface makes use of syntax highlighting to indicate how rules were parsed.**

the words “(No [Element Name] Selected)” are added in red the end of the rule. We choose to use colored underlining instead of other common methods such as colored text or backgrounds for several reasons. First, a single element value may be several words in length and we wanted to make it unambiguous how phrases were being grouped by the parser. Second, issues such as color blindness and poor eyesight can cause color-based interfaces to perform sub-optimally. By creating a clear line under the text we hoped that the interface would still provide assistance to those who had difficulty distinguishing color. Finally, we didn't want to overly distract the user by creating a loud and busy interface. By underlining phrases we hoped to provide an additional tool without overly distracting the user.

The resulting *Author* interface can be seen in Figure 6. The interface features a text box labeled “Rule Work Space” in which the user can enter or modify their natural language policy rules. Below the “Rule Work Space” is the same rule guide as in the original interface with the addition of information to tell the user about the color-coding of different policy elements. Next is a blue box which displays the last rule added, altered or selected. As soon as new policy authoring changes are saved, the rule appears in this box allowing the user to immediately see the parsing results. At the bottom of the screen is the policy rule list. This is the list of rules which compose the current policy rule set. Each rule is syntax highlighted so that a user can easily scan a document for errors. Clicking on any of the rules causes that rule to appear in the “Rule Work Space” and the reviewing box, allowing the user to reference the rule's previous parsing while they are making edits.

## 2. METHODOLOGY

### 2.1 Participants

Study participants were recruited via email sent to researchers and interns at an industrial research site in North America. A total of 30 participants were recruited for the study. Ten participants completed a pilot test. Twenty participants completed the

empirical study, however, the data from two participants was dropped prior to analysis due to insufficient English skills, and one participant was removed because of prior knowledge of the system. The research reported here is based on 17 participants in the study, 9 in the Experimental condition and 8 in the Control condition.

The users in the empirical study ranged in age from 24 to 53 years old with an average age of 35, and users employed computers an average of 8 hours a day. None of the participants had experience writing privacy policies. Two participants were technical writers, and the rest were in electrical engineering and computer science related fields.

The participants were knowledge workers who were novices in the policy authoring area. The participants had knowledge of the task domains (interacting with pharmacies and financial organizations, and the topic of privacy policies in general). We chose to study participants from this group because there are very few policy authoring experts and their time is very valuable. It was felt that the participants' general experience and the study topic of learning how to write high-quality policy rules with different interfaces qualified them as an appropriate initial study population. Policy authors in organizations are a select set of experts. We plan to follow-up with experimental studies of that population in the future as the field grows in size and availability. Participants received a coupon for a free lunch as compensation for their participation in the research study.

## 2.2 Design

The study employed a between subject design, with random assignment to groups. The Control group was assigned to use the original SPARCLE authoring interface. The Experimental group was given the new authoring interface featuring syntax highlighting.

### *Pilot Study*

A pilot study was used to validate the study instruments, methodology and provide an initial test of the system. After the pilot study we made several simplifications to the study tasks and altered the tutorial to both be more controlled and informative. The interface was also altered in minor ways to address several system specific issues encountered by participants. The study design presented in the following was partially informed by the pilot study.

### *Training*

Participants were given a tutorial on how to write privacy policy rules and how to use the SPARCLE system. The tutorial included an example rule the participant typed in, reviewed parsing on, and made modifications to. The session administrator guided the participant through the tutorial, ensuring that participants received consistent training content. The training session required 10 minutes of the 60 minute study session.

### *Tasks*

The goal of the research study was to investigate people's ability to take descriptions of changes to policy situations and author high-quality, complete policy rules that would parse with high accuracy. We intentionally gave users tasks with some flexibility in the solutions so that they could demonstrate how they proceeded to construct the rules with the experimental and original versions of the user interface of the policy authoring tool.

Participants were given two scenarios which were presented in random order. The scenario described an organization and asked the user to make two modifications and two additions to the organization's existing privacy policy. One task concerned the privacy policy for an online pharmacy and the other was about an online financial organization. The existing privacy policy for each task was composed of five rules. All rules were written in active voice in the same format as the example rule guide.

The following is an excerpt from the user's task describing the requested changes to DrugsAreUs's privacy policy.

Your current task is to update the current DrugsAreUs privacy policy using the SPARCLE tool. These updates have to do with new organizational procedures about handling personal information on the website. As Chief Privacy Officer, you want to reassure our customers that DrugsAreUs continues to be a privacy sensitive company.

As part of these changes in procedure, Customer Service Representatives will begin collecting email addresses from customers in addition to other personal information which is currently collected. This information will be used to fulfill customer orders. If customers indicate that they wish to receive special offers, Marketing Reps can provide partner organizations with the customers' email addresses in addition to the information already provided.

If a user forgets her password, Website Administrators or Customer Service Representatives can use email to send a temporary password. DrugsAreUs customers will also be allowed to view and modify their own personal information (such as home address, phone number, email address and credit card information) so that they can verify its correctness and make changes. However, customers can only make changes if they are successfully logged in to the web site.

The scenarios and original policies for both fictional organizations were heavily based on materials previously used by Reeder, Karat, Karat, and Brodie in their study of common issues experienced in policy authoring [14].

### *Data Collection*

We collected both qualitative and quantitative data during the study session. Quantitative data collected included Time on Task and the final text of the rules. Screen captures and audio recordings as well as experimenter notes were used to capture all of the participants' comments and thoughts during the sessions. Screen capture and audio data was collected using a program called Camtasia.

## 2.3 Procedure

The study session ran approximately 60 minutes with each participant. When participants first arrived for the study, they completed a short demographics questionnaire. They were then informed that the session would consist of a tutorial where they could ask questions, a set of two tasks during which they could not ask questions, and a debriefing interview. The participant was then given a copy of the tutorial and completed it with the researcher. During the tutorial the session administrator answered any questions the participant had concerning the interface.

After the tutorial the researcher gave the participant two tasks and asked the participant to complete the tasks in order. After each task the participant was asked to say "I'm Done." The administrator left the room and the participant then completed the two tasks. At the end of the session, the administrator joined the participant to conduct the debriefing interview. Participants were first asked for their impression of the system and what they

thought were its good and bad points. The researcher then asked about events that occurred during the session such as interface issues, parsing errors or anything that appeared to confuse the user. Finally the participant was given an eight dollar lunch certificate for use in the building's cafeteria.

### 3. RESULTS

We completed several different data analyses on the data collected. Qualitative data was analyzed using Contextual Design techniques such as Affinity Diagrams to find common themes and connections among them [8]. Quantitative data was analyzed to determine whether there were differences in user performance across the conditions.

#### 3.1 Qualitative

As part of the analysis of the qualitative data collected, we built an affinity diagram using transcribed comments from participants, researcher notes, usage patterns and rule errors. We grouped concepts based on common themes.

Several important themes emerged during this analysis. These themes include the use of syntax highlighting, reasons for using particular interfaces for sub-tasks, and task interruptions. Using the affinity diagram we discussed and analyzed these themes to better understand why participants had trouble with certain interface features.

Participant policy rule authoring was analyzed to identify rule modification, creation, and deletion patterns. For each user, the number of rule modifications, creations and deletions that occurred on each page were computed. This data was then analyzed to determine which interface capability the user was using for various sub-task actions. Estimates of these results were collected during the participant sessions and informed the debriefing sessions. The sums were later verified through analysis of the screen capture data.

##### *Syntax highlighting*

The first theme identified addressed the syntax highlighting added to the *Author* page for the experimental users. In general the highlighting was well liked and considered useful by participants. They liked the use of colors to differentiate the types of parsed elements and found the immediate feedback to be very useful. One participant commented that the colors were hard for him to differentiate but even he found the underlining useful.

Despite these perceptions, participants still had trouble with various aspects of the experimental interface. First, they seemed to have trouble noticing when parsing results identified significant issues. Even participants who were confident that they understood the interface were surprised when the session administrator pointed out errors to them during the debriefing session. Participants in the control condition had fewer problems with this since the *Transform* interface expressed very clearly the parser's interpretation of the rule. Though the *Transform* interface was available to participants in both groups, participants in the experimental group used it less often because the *Author* interface showed them the results of the parsing

Another goal of the syntax highlighting was to increase authors' ability to quickly write rules that the natural language parser could parse accurately. While the parser has high accuracy for the constrained natural language on which it is based [9], when it is used outside those constraints, results are mixed. For participants

in the control condition, parsing results were a source of minor annoyance when they discovered that their rule had not parsed perfectly and would need to be fixed. This was not overly concerning for them and most participants were able to fix the parsing errors with minimal problems. The feedback given to experimental group users, however, allowed them to recognize and fix the error immediately. For most participants this worked well, a simple rewording of the rule would fix the problem. For some users, this was a source of great frustration. These users either chose to not use the recommended rule language or simply were unable to formulate the rule into a well formed privacy rule. Often these users would give up and instead make their changes in the *Transform* page.

##### *Reasons for using particular interfaces for sub-tasks*

A second theme that emerged involved the pattern of use displayed by participants when choosing the interfaces they preferred to complete a particular sub-task. Participants in both conditions were presented with a guided natural language interface and a structured interface where they could select rule elements. Participants in the Control group used the *Author* interface for creating new rules and the *Transform* interface for resolving parsing issues and making other small modifications. Participants in this condition made extensive use of both interfaces, often switching between them to obtain the desired effect. Only one user in the Control condition chose to complete both task scenarios using only the *Transform* interface, all other users made use of both interfaces. The participant (a non-native English speaker) who did use the *Transform* page extensively for authoring explained that she did so because she wanted the tool to compose the sentence for her to eliminate errors on her part in writing.

In contrast, five of the experimental group participants, chose to complete both task scenarios using the *Author* interface only. When asked why they didn't use the *Transform* page, most said that they simply didn't need it since all the information they required was provided on the *Author* page.

Participants in both conditions rarely used the *View Policy* interface. Though participants felt the interface was valuable, only one person actually used it to identify a problem with coverage of the policy. This is understandable since the original policy rule set the participants were given was fairly small, and they were asked to create new rules to cover a self-contained new area and otherwise make straightforward changes to existing policy rules.

##### *Task interruption*

A third theme emerged regarding the problems encountered when the user became interrupted by the interface. Users in both conditions discussed issues that interfered with their sub-tasks. Users in the Experimental condition expressed difficulty in staying focused while writing and editing policy rules due to the feedback on parsing syntax issues which would motivate them to focus on the syntax over the content.

Users in the control condition experienced difficulty in using the *Transform* page because a system bug in the prototype developed for the study interrupted their ability to create policy rules. Whenever the user added a new value using the *Transform* page all unsaved changes on the page would be reset. This meant that anyone using the page to create rules would first have to make sure all the values they wanted were present, and then select them all at once. A control user told us that he "wasn't able to think

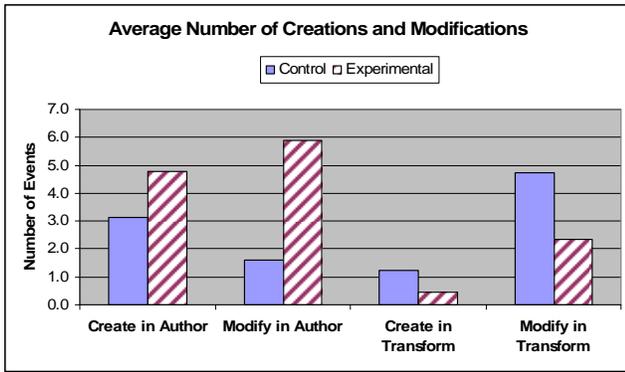


Figure 5. Policy rule creations and modifications.

about the problem in a natural way because of terms re-setting.” Experimental users rarely encountered the issue since they primarily made edits and additions on the *Author* page.

Several participants in both conditions commented about the key sub-task goal of just “getting ideas out”. They expressed that this creative process needed to occur first in completing the task.

### 3.2 Quantitative

#### Time on task

Time on task was computed based on the time the user received the task to the point at which they said “I’m done.” These timings were calculated using the screen capture files. The times for the tutorial and the debriefing session were also computed to ensure that users had received similar treatment.

Analysis of the Time on Task data found no statistically significant difference in user performance in the two conditions. On average the tutorial took ten minutes and the debriefing session took 9.5 minutes. The scenarios took an average of 28.9 minutes to complete. Participants spent an average of 16.3 ( $\sigma=3.4$ ) minutes completing their first scenario and 12.6 ( $\sigma=4.6$ ) minutes on their second scenario. There was no significant difference between the times for the first task scenario and the second one.

#### Interface usage

The data analysis showed there was a main effect for overall use of the *Author* interface by the two groups (see Figure 7). There were also 82 new rules created, 35 by the Control group and 47 by the Experimental group. Experimental group participants made significantly more use of the *Author* interface than Control group participants ( $F=11.5, p=.004$ ).

There were a total of 125 rule modifications, 51 by Control condition participants and 74 by Experimental condition users. There was a statistically significant interaction effect in the use of the *Author* and *Transform* interfaces for policy rule modification. As can be seen in Figure 7, the Experimental group modified significantly more policy rules using the *Author* page with syntax highlighting ( $F=8.3, p=.01$ ), whereas the Control group used both pages to make modifications. Few participants in either condition chose to create rules using the *Transform* page. However, this was unsurprising since in the tutorial participants were only shown how to create rules in the *Author* interface.

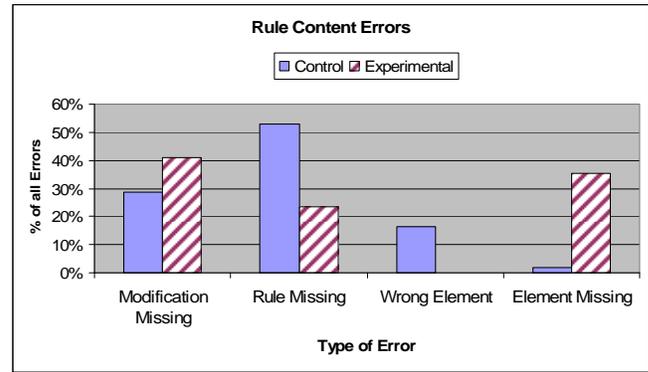


Figure 6. Rule content errors as a percentage of all errors.

#### Content and parsing accuracy

Policy rules were scored to reflect their accuracy. The two modifications were worth one point each. The simpler rule with four elements was worth four points and the more complex rule which contained a condition was worth five points. One point was deducted for each missing or incorrect rule element. For example if a five point rule was missing a single element one point would be deducted. Each of the two scenarios was worth a maximum of eleven points so a perfect score would be twenty two.

Each policy rule was scored twice, once for content accuracy and the second time for parsing accuracy. Content accuracy measured the presence of all required elements within a rule over the total number of expected required elements. For the purposes of this paper, we define parsing accuracy as the number of elements that were correctly parsed over the total number of expected elements.

There were no statistically significant differences in content accuracy scores for the policy rules created and modified by the two groups (see Table 1). In analyzing the errors that did occur, one small pattern emerged. More participants in the Experimental group made content errors but the errors they made resulted in smaller scoring deductions. Only two Control group participants did not score the full points possible on Content Accuracy, however, their errors were more costly.

Table 1. Rule accuracy for content.

	Control	Experimental
<b>Average accuracy across all users (N=17)</b>	86.1%	91.4%
<b>Average accuracy for users who made errors (N=6)</b>	44.3%	80.7%

The user errors were analyzed to determine if there were any design implications that could be gained from them. The majority of the errors made by the two Control group users and three of the four Experimental group users seemed to be the result of a lack of understanding of the study task. Both control and experimental users received scoring deductions due to failure to make requested policy rule modifications (Modification Missing) and to create new policy rules where warranted (Rule Missing) (see Figure 8). For example, one Control group participant explained during the debriefing session that he thought one of the rule additions was self-obvious and didn’t need to be added to the privacy policy for it to be considered part of the organizational policy. Similarly,

Control group participants received scoring reductions for attempting to write different rules (Wrong Element) than those specified.

There were no statistically significant differences in the parsing accuracy of the policy rules created by the two groups (see Table 2). To measure the effectiveness of the interfaces in assisting users to create implementable rules, we analyzed policy rule parsing problems that users encountered. In the following analysis, we assumed that the policy rule content written by the user was correct and analyzed how accurately their rules parsed.

**Table 2. Parsing Accuracy of policy rules.**

	Control	Experimental
<b>Average accuracy across all users (N=17)</b>	94.9%	94.5%

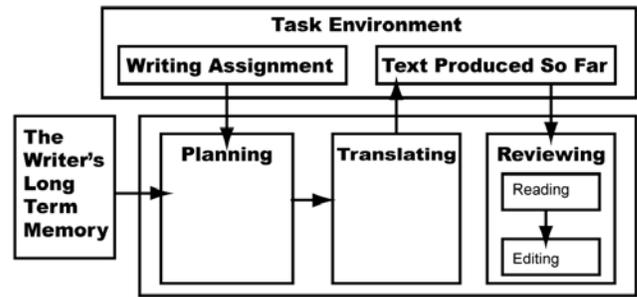
Only one Control group user wrote a policy rule that failed to parse accurately. Four of the five elements in this policy rule did not parse correctly. These errors occurred because the user forgot to review this rule on the *Transform* page.

Five Experimental group users wrote a total of eight policy rules that failed to parse accurately. The majority of the errors were due to the use of improper syntax for a single rule element in a policy rule.

#### 4. DISCUSSION & SUMMARY

When we added syntax highlighting to the authoring interface we started with the hypothesis that the highlighting would better support learning, enabling users to learn how to write parsable rules. We anticipated that participants would use the feedback to both learn to write better rules and perform an immediate review and correction of the rule. We thought users would be able to author the rules and work on the syntax simultaneously without creating cognitive overload. This hypothesis turned out to be incorrect as there was no significant decrease in the task time or errors between conditions. While Experimental condition users said during the debriefing session that they liked the new interface and appreciated being able to write their rules all in one place, their policy authoring performance using the interface was no better than the Control group.

To better understand why the interface failed to improve participant performance, we apply a theoretical model from the study of human writing processes. The model depicted in Figure 9 is a simplified version of the model from Hayes and Flower's work on the cognitive process of writing as described in [1]. (See [1] or [7] for a complete description of the model.) In their model they describe the writer's process as made up of three components: the writer's long term memory, environment and active processes. A writer goes through three active mental processes when writing a document: Planning, Translating and Revising. Planning, the first process is where the subject decides what needs to be done, synthesizing her writing goals into a clear plan of what should be written. In the next process, Translating, the subject uses her plan and knowledge of the subject area to create natural language. In the last process, Revising, the subject reads through the currently written text and makes corrections where necessary. In the following sections we will examine each of these mental processes of writing and how they affected our study participants.



**Figure 7. Simplified version of the Hayes and Flower model of the writing process.**

#### 4.1 Planning/Translating

The Planning process of writing involves generating a plan, organizing thoughts and setting writing goals. This process produces a content plan which contains the writing strategies, structure and intention to be used but has no details concerning the exact content. The Translation process takes in the plan outputted by the Planning process and uses it to convert information from the writer's memory into natural language. Information stored in memory is structured semantically and during the Translation process must be converted to a linear representation, natural language. The result of the Planning and Translating processes is a draft of the user's attempt to verbalize the intended text [1]. In SPARCLE the Planning and Translation processes can be accomplished in either the *Author* or the *Transform* interfaces.

The reason Experimental participants didn't experience reduced error rates may be because the syntax highlighting on the *Author* interface interrupted users' verbalization of their writing plan in the Translation process. By providing quick feedback to users about policy rule content and parsing accuracy, the interface motivated them to fix the errors immediately, causing them to stop the Translating process and move directly to Revising. While this was only a minor irritation for most users, a few found it highly annoying. One user became frustrated with the switching and instead went and made all her additions, and then went through and revised the policy rules for proper syntax. Experimental users also had difficulty noticing that elements of their rules were either missing or were composed with poor syntax. One experimental user expressed this situation well in his debriefing interview when he said:

When I type I read, maybe after some time of use I will adapt to the system to think of 'maybe I missed some slot.' But when I was typing I only think of what type of effect I want to achieve by this rule. I didn't really think maybe this is not considered a purpose by the computer.

Participants in the control condition did not encounter these frustrations because they were not being prompted to switch processes before they were ready. Since the interface provided no error prompting the control users were able to complete their Translation process uninterrupted and then naturally moved onto to Revising.

#### 4.2 Revising

In the Revising process, the writer compares the current text to the intended text and makes any necessary changes to correct their work. They look for and correct potential problems, including parsing issues, semantic problems, awkward phrasing, spelling

and syntax errors. The Revising process is composed of two sub-processes: Reading and Editing which we will address separately in the two following subsections [1]. In SPARCLE Revising can be accomplished by either experimental group on any of the three pages (*Author*, *Transform*, or *View*). In the following sections we address the Reading and Editing sub-processes separately.

#### 4.2.1 Reading

In the Reading sub-process, users compare their work with the intended text. The writer may identify an error that requires action at which point they will transition to the Editing sub-process. In our study users needed to be able to recognize errors in both content and parsing. Identifying content errors required a user to read through their rules and determine that they differed from the intended rules. Parsing errors could be identified by using the syntax highlighting or the *Transform* interface to determine that intended parsing did not match the actual parsing.

Experimental users seemed to do the Revising process for each rule immediately after they Translated it or not at all. With the exception of a few users who intentionally did not complete the parsing review until they had created all their rules, users in the Experimental condition reviewed rules in detail immediately after creating them. After writing all their rules (Planning/Translating) Experimental group participants would scroll up and down the rule list glancing at each rule. Sometimes they would then click on the *View* or *Transform* page and scroll around briefly. They very rarely looked at any rule in any great depth. From the interviews we learned that users who did all their editing in the *Author* page were fairly confident about the policy rules they had written and in their reviews were scanning for obvious errors only. Since they were not reading through each rule they missed the subtle syntax highlighting that would have indicated errors and falsely concluded that no errors existed.

In contrast, participants in the Control group waited until they had written all their rules before reading through them for parsing errors. This gave them a clear transition between the Translating and Revising processes when they moved from the *Author* to the *Transform* pages. The only way to review parsing was to use the *Transform* page which required the user to examine each rule individually or not at all therefore there was no way to skim looking for errors. The *Transform* page also explicitly specified how each rule was parsed and clearly highlighted any missing element in red. Consequently, as can be seen in Figure 8, Control group users had extremely few issues with missing elements.

Participants had the option to review rules using the *View* interface in addition to the other interfaces but few users in either condition made use of it. A Control user explained that the *View* interface wasn't helpful to her when reviewing rules because she did not have a good understanding of what the effective policy should look like. She felt the interface would be very useful if she was more familiar with the intended policy. In terms of the model, users did not have a good understanding of the intended effective policy and therefore had trouble using the *View* interface, which displays the effective policy, to compare implemented and intended policies.

#### 4.2.2 Editing

The Editing sub-process concerns the user making changes to the text to address issues found in the Reading sub-process. At this point participants are no longer trying to synthesize an idea;

instead they are trying to make a very specific change to the text to address a specific issue. In our case users were making edits to either adjust the rule content or address rule parsing issues. They changed the syntax and wordings of policy rules to achieve higher accuracy parsing and also added or removed values of elements to improve policy rule content.

Control group participants used the *Transform* page to review the parsing and if they found something they didn't like they would change it right there. This was easy for them since the interface allows users to edit the parsing directly. Control group participants also used the pre-defined values in the element boxes as ideas about what they might have missed. Users would scan through a list of values and add ones they thought were missing or they had previously forgotten.

Experimental users make the majority of their edits in the *Author* interface though, as can be seen in Figure 8. With the exception of a few users who intentionally did not complete the parsing review task, users in the Experimental condition edited rules immediately after creating them.

In addition to performing Revising tasks to correct parsing issues, users also felt a strong need to correct grammar and spelling issues. Those users who completed the Revising phase primarily in the *Transform* page were particularly concerned about this. The *Transform* page converts policy rules into a standard active phrasing, and there are occasionally issues with awkward sentence structures and improper verb usage. One participant in the Control group explained that he preferred that the policy rules be written in his wording rather than the sentence structure used by the system. Consequently he, and several other users, chose not to use the *Transform* page and made edits using the *Author* interface.

### 4.3 Recommendations

Based on the results of this study we would like to put forward a few recommendations to designers.

In regard to the design of the policy authoring interface and interaction methods, syntax highlighting may be a valuable tool for giving users a quick understanding of how natural language is parsed. However, it is important to provide this information in a way that integrates with the user's writing process. Providing feedback too early in the process can cause the user to switch mental processes before they are ready.

We learned that for policy authoring there needs to be a clear and smooth transition between the Planning/Translating processes and the Review process. We also learned about the need to provide a smooth and simple transition after identifying an error to revising it. This transition must occur without switching screens or having to enter values in a separate area apart from the Review area.

## 5. CONCLUSIONS

In the research reported in this paper, we hypothesized that syntax highlighting within the *Author* interface would allow policy authors to more quickly recognize and fix errors and create higher quality policies but instead measured no significant improvement. From the analysis of the qualitative data collected we have shown how our results align with a common writing process model. We have also presented more detailed results from our user study. In addition to the policy authoring process, we also learned that although the participants reported that they liked the syntax highlighting interaction method, they often do not want to find

and fix policy errors of any type during the Translation process because they are concentrating on recording their ideas. In other words, we learned that the Translation and revision subtasks can not be done in parallel effectively. Further, the analysis of the quantitative supported this finding.

In the future we plan to use the results of this study and to apply the recommendations we have proposed to the design of SPARCLE in order to provide a better user experience for policy authors and to help them create higher quality policies more quickly. In particular, we believe that syntax highlighting is a valuable tool that can assist in the creation of policies and is well received by policy authors, but that it should be applied during the Review phase rather than at the Translation phase of the policy authoring process. In SPARCLE this means that syntax highlighting could be more useful when incorporated into the *Transform* page which contains the functionality most often employed by users when they have transitioned to the Review phase of the authoring process. We hypothesize that this could help users to more easily detect errors and learn how to quickly fix them.

Overall, we believe that the research reported in this paper highlights how crucial the proper use of HCI design, analysis, and evaluation methodologies are to the creation of usable applications. In the course of the research reported in this paper, we learned about the importance of understanding the policy authoring task sub-goals and applying this knowledge to the design of the application. Even more importantly, this work shows how the quantitative and qualitative analysis of data collected in a usability study allows designers to understand whether a given interaction method is truly useful, and, if so, how and where it should be applied in order to give the intended audience the most value. At its heart, the success of the user-interface design remains in understanding the users, their tasks, and the context of use and for many designs, that understanding needs to be at a fairly detailed level of the user's model of the activity.

## 6. ACKNOWLEDGMENTS

Our thanks to all the students and researchers who have worked on SPARCLE to make it the comprehensive and valuable tool it is today.

## 7. REFERENCES

- [1] Alamargot, D. and Chanquoy, L., *Through the Models of Writing*. Kluwer Academic Publishers, Dordrecht, The Netherlands, 2001.
- [2] Anton, A., He, Q., and Baumer, D., The complexity underlying JetBlue's privacy policy violations, *IEEE Security & Privacy*. August/September, 2004.
- [3] Breaux, T.D., and Anton, A.I. Analyzing Goal Semantics for Rights, Permissions, and Obligations. Technical Report #TR-

2005-08, North Carolina State University, Department of Computer Science, February 15, 2005.

- [4] Baumer, D., Earp, J.B. and Payton, F.C., Privacy in medical records: IT implications of HIPPA. *Computers and Society*, 40-47. December, 2000.
- [5] Brodie, C., Karat, C.M., Karat, J., An empirical study of natural language parsing of privacy policy rules using the SPARCLE policy workbench. In *Proceedings of the Symposium on Usable Privacy and Security (SOUPS)*, New York, NY, ACM Press 8-19. July 2006.
- [6] Brooks, K. Migrating to role-based access control. In *Proceedings of the Fourth ACM Workshop on Role-Based Access Control, (RBAC)*. ACM Press, New York, NY, 71-81. 1999.
- [7] Hayes, J.R. and Flower, L.S. Identifying the organization of writing processes. In L. W. Gregg, and E.R. Steinberg (Eds.), *Cognitive processes in writing: An interdisciplinary approach* (pp. 3-30). Hillsdale, NJ: Lawrence Erlbaum Associates, 1980.
- [8] Holtzblatt, K. and Beyer, H., *Contextual Design*. Morgan Kaufmann, San Francisco. 1998.
- [9] Karat, C., Karat, J., Brodie, C., and Feng, J. Evaluating interfaces for privacy policy rule authoring. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*. (ACM Press, New York, NY, 83-92. April 22 - 27, 2006.
- [10] Karat, C., Karat, J., and Brodie, C. Why HCI research in privacy and security is critical now, *International Journal of Man-Machine Studies* 63(1-2), 1-4. 2005.
- [11] Karat, J., Karat, C., Brodie, C. and Feng, J., Privacy in information technology: Designing to enable privacy policy management in organizations. *International Journal of Human-Computer Studies*, 63(1-2), 153-174. July, 2005.
- [12] Karjoth, G. and Schunter, M. A privacy policy model for enterprises. In *Proceedings of the 15th IEEE Computer Security Foundations Workshop*, 271-281. 2002.
- [13] Smith, J., Privacy policies and practices: Inside the organizational maze. *Communications of the ACM*, 36, 12, 105-122. 1993.
- [14] Reeder, R., Karat, C., Karat, J. and Brodie, C., Usability challenges in security and privacy policy-authoring interfaces. *INTERACT*, 2007.
- [15] Rubin, L. F., Syntax-directed pretty printing – a first step towards a syntax-directed editor. In *Proceedings of the IEEE Transactions on Software Engineering*. 119-127. March, 1983.
- [16] Vertanen, K., and Kristensson P. O., On the benefits of confidence visualization in speech recognition. In *Proceedings of the 26<sup>th</sup> Annual SIGCHI Conference on Human Factors in Computing Systems*. 1497-1500, April, 2008.