The NASA Personnel Security Processing Expert System

David Silberberg

The Johns Hopkins University Applied Physics Laboratory
Milton S. Eisenhower Research Center
Johns Hopkins Road
Laurel, MD 20723
(301) 953-6231
David.Silberberg@jhuapl.edu

Robert Thomas

NASA Headquarters
300 E Street, SW
Washington, D.C. 20546
(202) 358-2456
rthomas@hqops.hq.nasa.gov

Abstract

The NASA Personnel Security Processing Expert System is a tool that automatically determines the appropriate personnel background investigation required for a civil servant or contractor occupying a position of national security or public trust. It also instructs the personnel security processing staff to perform special checks based on a specific position.

The system is implemented using a rule-based expert system and a World Wide Web interface. The system design separates the user interface, knowledge base and control structure to simplify system evolution. When one subsystem is modified, the others are impacted minimally.

This system provides many benefits to the NASA Personnel Security Program. First, it frees the agency personnel security specialist from trouble-shooting and correcting all investigative problems. It also provides a learning tool for security processing staff at each installation. The system ensures that each installation security office is in compliance with all applicable laws, regulations and policies. Finally, eliminating overlapping, inappropriate and duplicative efforts to process employees saves many resources.

The system was deployed less than a year ago. To date, it saved $1.2 million of the $1.5 million agency-wide personnel security budget.

Problem Description

One of the mandates of the NASA Security Office is to ensure that the appropriate personnel background investigations are performed for civil servants and contractors occupying designated positions. Some of the common position duties require the development of or access to automated data processing systems, personnel reliability program information, national resource protection program facilities, and classified national security information. The level and type of background investigation required varies with the position and the employee. For instance, positions involving national security require screening at multiple levels to ensure that classified information is not compromised; the level depends on the sensitivity of the program and the employee's duties. Similarly, positions of public trust require screening at multiple levels commensurate with the amount of financial or resource damage that the employee potentially could cause. Other positions require specific background investigations mandated by the nature of the position, program and applicable laws. Finally, background investigations of employees who have had recent background investigations may reuse some of the information from the previous investigation, thus saving resources.

Currently, personnel security specialists at each NASA installation initiate the required investigations for applicants and employees at their respective centers. There are manuals, laws, executive orders, and federal regulations that stipulate the required background investigation to be conducted for each case. However, the manuals are not always utilized in an efficient manner to ensure compliance with current national policies pertaining to personnel security investigations. Therefore, applicants and employees are subject to being processed inappropriately, which may result in increased costs to the agency.

There is a formidable learning curve for new personnel security specialists. There are many positions and programs with unique requirements as well as laws,

procedures and regulations that need to be understood and analyzed to determine the appropriate investigation level. Often, both new and experienced security specialists consult with the NASA personnel security expert at headquarters for guidance, direction and final arbitration on the appropriate background investigation. This process is costly and diverts the expert from other matters of national security.

To address these problems, we created an expert system with an interface available through the World Wide Web. An installation personnel security staff member loads the NASA Personnel Security Program homc pagc (initial screen) and answers questions about the employee and position. The back-end expert system evaluates the questions, determines the appropriate background investigation for the position and employee, and displays the results and special instructions to the personnel security processing staff. If contradictory answers are entered, the user is encouraged to resolve the contradictions.

This paper describes the development of the NASA Personnel Security Processing Expert System. In the following sections, we describe the motivations for building an expert system available through the World Wide Web, the knowledge engineering process with respect to requirements gathering and rule acquisition, the system design, the user interface design, the structure of the expert system with special emphasis on the rule base, and the system's cost benefit to NASA. We also emphasize how the flexibility of the design approach made it easy to transition to a second version of the system. Finally, the conclusion describes the success of the system and future work anticipated.

Application Description

The following subsections discuss the alternatives considered and why a WWW-based expert system was chosen.

The Requirements

The NASA Security Office identified the need for an automated tool to aid in determining the appropriate background investigation. The system needed to capture knowledge of the applicable laws and regulations, and perform as well as the NASA personnel security expert. It also needed to ensure uniformity in investigation processing. Furthermore, the tool needed to be accessible by the NASA personnel security community on IBM-compatible PC's, Macintoshes and UNIX workstations. Finally, this system was a prototype effort and the budget was limited. Therefore, the system needed to be built with inexpensive software tools.

Why an Expert System

An *expert system* paradigm was chosen because it represents and reasons with knowledge of some specialist subject with a view to solving problems or giving advice. If the decision maker uses an expert system, it improves the decision maker's productivity. Also, if the decision maker is not yet an expert, an expert system can help the decision maker reach the level of an expert (Jackson 1990) (Luger 1993).

Issues of Expert Systems

Rule-based expert systems are appropriate for applications in which rules play a significant role. If the knowledge of the problem can not easily be expressed in terms of productions rules, than the expert system introduces overhead with little benefit. In our application, the information about what procedures apply was easily expressed in terms of rules.

Our application requires rules to be divided into *rule sets*, as will be explained later in the paper. There are two common ways to divide rules into sets. One is to write the software encapsulating the expert system shell in a procedural manner. The code loads a set of rules and calls the expert system shell to evaluate them. When an intermediate end state is reached, the code interprets the result, unloads the set of currently loaded rules, loads the set of new rules consistent with the result and calls the expert system shell to evaluate them. This process continues until the ultimate end state is reached. However, the software external to the expert system must be vested with some knowledge of the application. If the rule base changes or the method for determining appropriate rules is altered, this software must also be altered. One would like to separate completely the knowledge base from the control software so that the rules can be modified without recompiling the application.

Another alternative is to use *control rules* to determine when control passes from one rule set to another. For example, when all rules of a rule set have had the opportunity to fire, control rules fire to set facts indicating that another set of rules are ready to be considered. To ensure that the control rules fire after all other rules of a rule set have had the opportunity to fire, *salience* is utilized. The control rules are defined with the lowest salience in their rule set to ensure they fire last. This approach removes the control responsibility from the encapsulating program and places it on the rule set. This is good from the standpoint that the encapsulating code need not be altered and recompiled when the knowledge base is changed. However, the rules must have inherent control information as well as knowledge of the domain. This adds a level of complexity to the knowledge base and makes it more difficult to maintain.

The Expert System Platform

We implemented the system using the CLIPS 6.0 expert system shell (Giarratano 1993a) (Giarratano 1993b). CLIPS was developed by NASA at the Johnson Space Center.

Why the World Wide Web

The World Wide Web has been described as a wide-area hypermedia information retrieval initiative aiming to give universal access to a large universe of documents. The World Wide Web project provides users on computer networks with a consistent means to access a variety of media in a simplified fashion. Using a popular software interface browser to the Web such as Mosaic or Netscape, the user can access much information available through a multitude of network protocols (Boutell 1995a) (Boutell 1995b). The Web project has changed the way people view and create information — it has created the first true global hypermedia network (Hughes 1994).

The WWW consists of documents and links. Indexes are special documents which, rather than being read, may be searched. The result of such a search is another document containing links to the documents found. A network protocol called the HyperText Transfer Protocol, or HTTP, is used to allow a browser program to request a keyword search by a remote information server. The Web contains documents in many formats. Those documents, which are hypertext (real or virtual), contain links to other documents, or places within documents. All documents, whether real, virtual or indexes, look similar to the reader and are contained within the same addressing scheme. To follow a link, a reader clicks with a mouse (or types in a number if there is no mouse). To search an index, a reader gives keywords (or other search criteria). These are the only operations necessary to access the entire world of data (Berners-Lee 1994).

The Common Gateway Interface, or CGI, is an interface for running external programs, or gateways, under an information server. Currently, the supported information servers are HTTP servers. A gateway is really a program which handles information requests and returns the appropriate document or generates a document on-the-fly. With CGI, a server can serve information which is not in a form readable by the client (such as an SQL database), and act as a gateway between the two to produce something which clients can use (McCool 1993).

A requirement of the personnel screening application is that it be available to the security specialists at all NASA sites across the country. The specialists use a wide variety of platforms including UNIX workstations, IBM-compatible PC's and Macintoshes. An option would have been to write a program using products supported on all the platforms and distribute it to all sites. Certainly, upgrades and changes to the user interface and rule set would have been difficult to manage and maintain. Alternatively, we could have written the application using a client/server architecture with a proprietary interface package supporting the various platforms. While the rule set would have been easily managed, the user interface would not. Therefore, we chose to implement the system using the World Wide Web using the Common Gateway Interface. This simplifies the management of the user interface and the rule set in that it can be maintained at one location, while remaining accessible to a wide set of users and platforms. Furthermore, the costs of using these products are minimal.

Issues of the World Wide Web

A significant issue of the World Wide Web using the Common Gateway Interface protocol is that it is stateless. This means that once a request is processed by a server, the server program is terminated. If a user interface requires a series of screens for which to enter data, the program required to process the data of one screen knows nothing about the state of previous screen inputs. Generally, information passed from one screen state to another is crucial in user interfaces. Solutions to this problem are described later in the paper.

Another issue of developing World Wide Web interfaces is its limited user interface flexibility. Screens are created with a markup language, which provides a limited set of commands to represent common text, images and screen field formats on a Web browser. The most common markup language is the standard version of the HyperText Markup Language, or HTML; however, vendors and other software providers have created extensions to the standard HTML as well as other more powerful markup languages. While the features provided by the markup languages are limited, they did not significantly limit the development of our application.

Knowledge Acquisition

Our expert has been recognized in the area of background investigations at NASA for more than a decade. During his years of service, he has written several agency manuals and documents to formally express the rules of determining appropriate background investigations. However, these documents often were not read thoroughly by other security specialists. Therefore, he wrote a general questionnaire of approximately 50 questions to lead a personnel security officer through the expert's decision process. The responses would provide enough information to determine the correct background check in foreseeable circumstances. The questionnaire was also the vehicle by which the expert expressed his knowledge domain to the software engineer.

The initial interactions between the expert and the software engineer were driven by the expert. Much time was spent acquainting the software engineer with his world. The expert gave a complete overview of the personnel security level determination process, and placed the software engineer in the frame of mind of the expert.

During this time, much of the focus was placed on the questionnaire. The software engineer spent many hours asking about the meaning of the questions and their implications. Through the questions posed by the software engineer, the expert obtained a good sense of the structure of rules in an expert system and the models that were built to reflect the concepts. Slowly, as more questions were asked, the expert modified his questionnaire and the model of his own process. He refined his categorizations of the process which led to a refinement of the questions on the questionnaire. Both parties slowly developed insights into the other's world. The result of the process was a common structure, captured on paper, that defined the structure of the expert's thought process in an ontology that was understandable to the software engineer. The ontology was the interface between the expert and the software engineer (Farquhar et al. 1995).

System Design

In the development of a prototype system, what is initially conceived is not usually close to the end product. Therefore, the design of a successful prototype must account for the inevitable evolution of the system. The software engineer must design the system so that any change has minimal impact. Therefore, the system elements were divided into the interface, the underlying model and the control code. The interface is all the software and HTML files specific to the user interface display. The model is the expert system, which includes an inference engine and the knowledge base. The control code drives the interaction between the expert system and the user interface.

The expert system design also is divided into the model, the user interface and the control code. CLIPS has an X-based, C-callable and command-line interface. Its model is its knowledge base, and its control code is its inference engine.

The benefit of this approach is that the main aspects of the system are encapsulated. Generally, changes to the model do not affect the interface and changes to the interface do not affect the model. If control aspects need modification, neither the model nor the interface need to be altered. In practice, cosmetic changes to the interface require no modifications to the model. However, some changes to the interface require corresponding changes to the model. For instance, consider the case where a new question is added to a user interface screen to capture previously unaccounted for user knowledge. A corresponding rule is added to the model to process it. However, nothing in the paradigm of the user interface processing or expert system code needs modification. Also, the control code needs no modification in an instance such as this.

User Interface Approach

Two aspects of the user interface are described in this section. The first is the software structure of the user interface code. The second is the actual screen flow and design.

Software Structure

The user interface code was designed so that any change to the view and its interactions would only affect the view code and the HTML files. When the screen format changes, nothing else needs to be modified. If the screen captures new information to be interpreted by the expert system, then a corresponding modification is made to the rule base. However, no other part of the model requires modification.

The most common protocol for displaying files with a Web browser is the HyperText Transfer Protocol (HTTP). The Web browser, using the HTTP protocol, translates a Uniform Resource Locator (URL) to the IP (Internet Protocol) address of a computer and either a name of a file containing HTML code or a name of a server program which produces HTML code. A message is sent to the addressed computer requesting the file or the server program. The server either returns the contents of the fixed HTML file or an HTML stream generated by the server program for display on the browser. The protocol used for passing screen inputs to the server program is known as the Common Gateway Interface.

HTML that originates from a fixed file is relatively easy to debug. However, there is little flexibility for dynamic screen generation based on the user's inputs. HTML originating from a program is more difficult to debug because a server program can not be run in debug mode from the Web browser. If one attempts to debug the program on the server, one must run the program without a Web browser. The results is a stream of HTML text is written to the standard output device that can be not viewed with a Web browser. Despite the drawback of limited debugging capabilities, a server program offers maximum flexibility for dynamic screen generation.

To overcome these problems, screens in our application have fixed and variable areas. The fixed areas consist of text, images, and fields which always are visible when the screen is displayed. The variable areas are the portions of the screen that may be displayed depending on the user input. The screens exist as disk files; the fixed areas are written using standard HTML and the variable areas are written using macros embedded within HTML comment lines. When a disk file address is entered on a Web browser, it displays only the fixed portion of the screen. When our server program is addressed in the Web browser, it reads each line of HTML disk file and writes them to the standard output. When the server program encounters a macro, it creates the appropriate HTML based on the user input and inserts into the output stream.

When the developer debugs the fixed portion of the HTML code, the URL is displayed without executing the server code. The screen design and layout can be modified independent of the software that produces the variable section of the screen. When the variable portions of the screen need modification, the code can be modified independently of the fixed HTML. This strategy provides flexibility for debugging fixed and variable HTML files.

To overcome the stateless nature of the HTTP protocol, we carry information from screen to screen through HTML hidden fields. The hidden fields and corresponding values are embedded in the output stream of the next screen by the server program. When the user interacts with the next screen and selects the submit button, the values of the user selections are sent to the server along with the values of the hidden fields.

Screen Flow and Design

The interface of the initial version of the system was a single screen that resembled the initial questionnaire of approximately 50 questions. Because of its size, the user could view only four or five questions at a time, at most. After completing the questions in view, the user had to scroll down the questionnaire to answer subsequent questions. Also, the user had to complete the entire questionnaire before submitting the answers. Most of the security specialists were only mildly comfortable with computers; thus, the interface was somewhat intimidating. We realized that the interface had to be modified to make it usable by the community.

After re-examination of the questions, we realized that not all questions were appropriate for all employees or for all positions. We concluded that the questions needed to be categorized by employee and position type. In a sense, re-categorizing the questions meant re-categorizing the rules. We gathered old rules into new rule sets that categorized employees and position types. Then we modeled a new screen interaction according to the new rule sets.

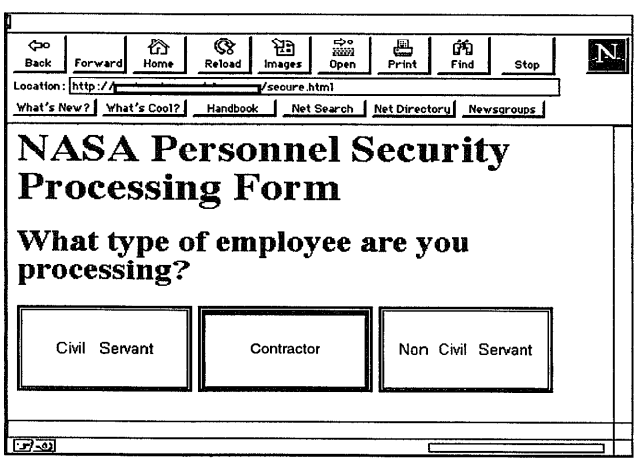

Figure 1. The Initial Screen

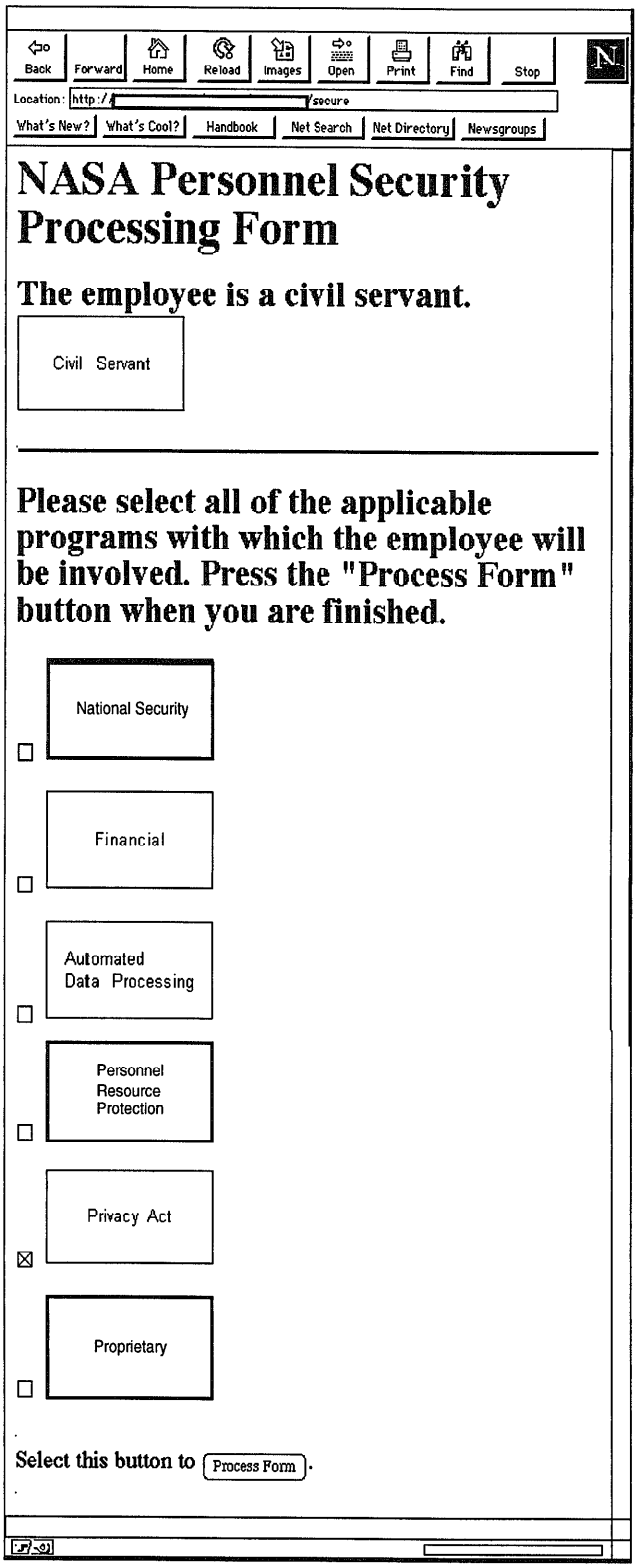

Figure 2. Selection of the Applicable Programs

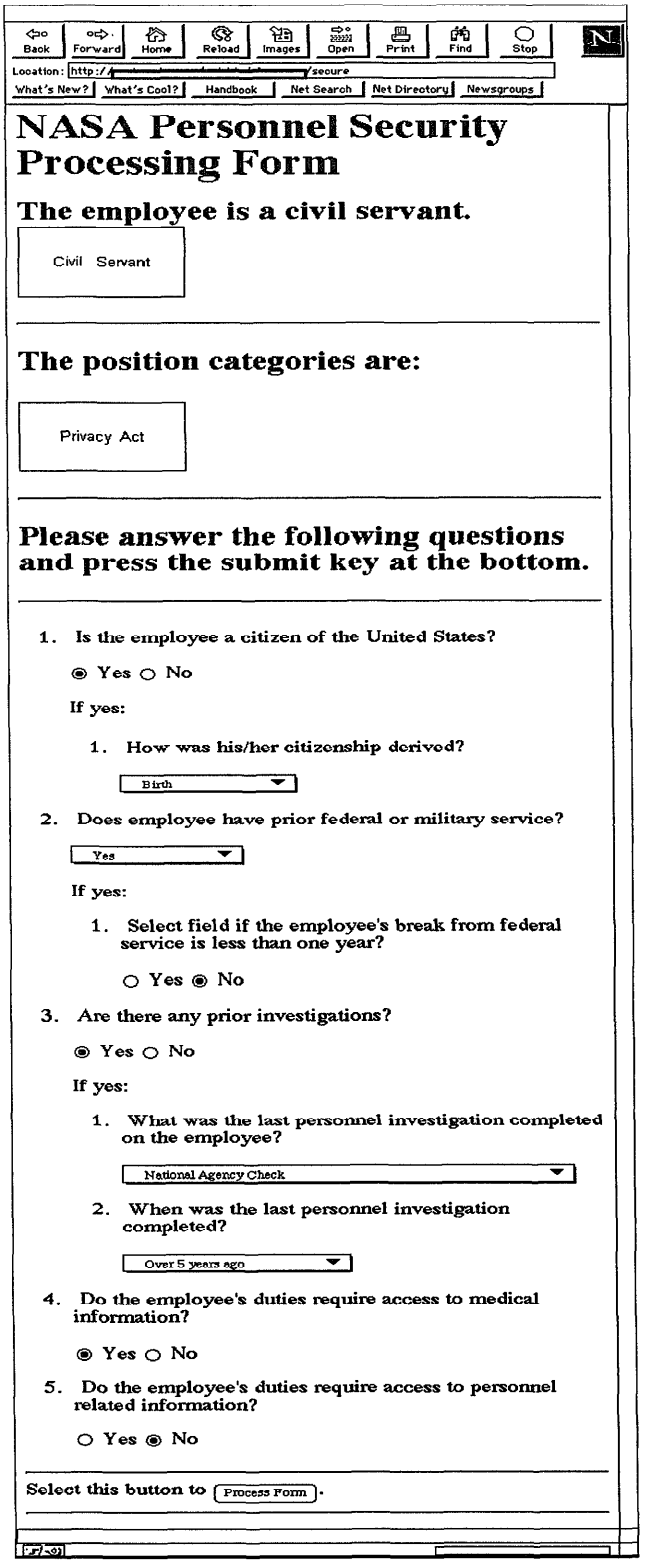

Figure 3. The Questionnaire

NASA positions are filled with civil servants, contractors and non civil servants (e.g., foreign nationals). There are many position types including positions of national security, automated data processing, financial management, child care, access to privacy act information, personnel resource protection, national resource protection, general services and general access to the site. Not all positions are available to each employee type. For example, civil servants are not hired to perform general services duties, and contractors are not hired to perform duties requiring access to proprietary information. Non civil servants must be escorted and will not have access to classified information.

The new interface is a series of three input screens. They are illustrated in Figs. 1, 2 and 3. The initial screen asks the user to select the type of employee being processed by "clicking" on a graphical icon representing the type of employee. The information is sent to the server, which calculates the positions appropriate for the selected employee type. The server returns a screen displaying the type of employee selected and a list of the potential duties that can be performed by that employee. Each duty is displayed with an icon and a check box. The user is asked to select all those duties that the particular employee being processed will perform. The user submits this form; the server calculates the appropriate questions applicable to the duties selected and displays a questionnaire to the user. This questionnaire may contain anywhere between two and ten questions, tailored to the duties selected. After answering the questions and submitting the form, the answers are sent to the server to calculate the correct background investigation. The server sends a screen back indicating the required background investigation and other checks that must be performed. This is illustrated in Figure 4.

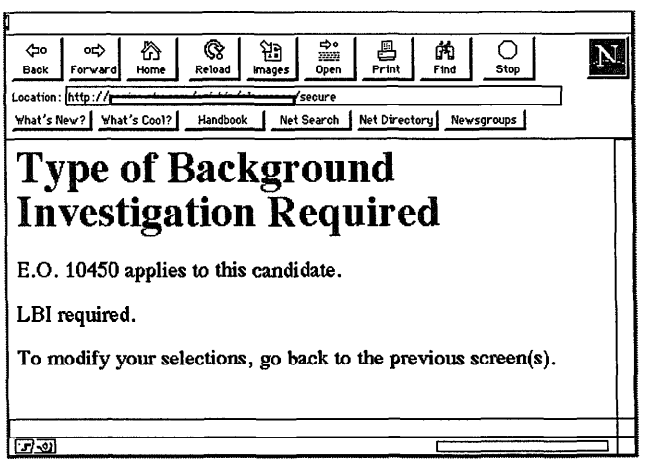

Figure 4. The Results

The new interface proved to be more user friendly. Users preferred to be presented with several short screens containing icons and a few questions rather than a long

screen containing many questions. It was easier to use and faster to interact with. More fundamentally, the new approach gave the user clear understanding of the expert's process used to evaluate appropriate background investigations.

When the overall structure of the rules was examined, better rule classifications were determined. We took the classifications initially handled by the rules and transformed them into a graphical interaction to produce a new user interface. In a sense, we moved some of the knowledge base represented by the rules to the user interface. While it is not clear that this process can be formalized or generalized, it does provide an area for future exploration. Also, the nature of the user interaction itself provides insight into the expert's process and thus, serves as a learning tool for the users of the system. This functionality was previously accomplished by the rule base.

Expert System Rule Design

This section discusses the importance of rule categorization and the flexibility of our approach.

Rule Categorization

The initial knowledge acquisition process identified over 100 rules. They were categorized into rule sets that identify conflicting questionnaire responses, that inform users of special checks that must be performed, that determine the sensitivity of the position, and that determine the appropriate background investigation. In general, the rule sets are considered serially. In the first version of the system, the *Conflict* rules were very important for uncovering inconsistent responses to the 50+ questions with which users were faced. The new user interface presents the questions to the user in a limited and focused way. This dramatically reduces the amount of possible *Conflict* rules the system needs. In both user interface versions, *Special Checks* rules are important for highlighting official documents and other special checks required due to the uniqueness of the employee and/or position. The *Position Sensitivity* rules infer the sensitivity of the position based on a number of interdependent conditions specified by the user input. Finally, the *Background Investigation* rules infer the type of background check required based on the position sensitivity and the past security history of the employee.

A simplified subset of the rules appear in Figure 5 using the CLIPS syntax. For example, the first rule defined is the *prior-service* rule. The conditions that must be met to activate this rule are that the position type is a civil servant, the employee has a military or federal work history, and that there was no prior investigation. If the inference engine fires this rule, it takes the actions following the "=>" symbol. In this case, it creates a fact stating that a conflict exists and prints a corresponding message.

After the user completes the user interface questionnaire,

Conflicts

```
(defrule prior-service
   (position-type "Civil Servant")
   (military-or-federal-history Yes)
   (prior-investigation No)
 => (assert (conflict TRUE))
   (printout t "If there was prior
      military or federal history,
      there must have been a prior
      investigation.<p>"))
(defrule conflict-summary
   (declare (salience -100))
   (not (conflict TRUE))
 => (assert (inputs-OK TRUE)))
```

Special Checks

```
(defrule 10450-applies
   (inputs-OK TRUE)
   (position-type "Civil Servant")
 => (printout t "E.O. 10450 applies to
      this candidate.<p>"))
```

Position Sensitivity

```
(defrule medical-info-sensitivity
   (inputs-OK TRUE)
              (access-to-medical-info Yes)
 => (assert (public-trust-level Medium))
   (assert (national-security No))
   (assert (public-trust Yes))
   (assert (non-sensitive No)))
(defrule is-public-trust-rule
   (declare (salience -100))
   (inputs-OK TRUE)
   (not (sensitivity-conflict TRUE))
   (or (public-trust Yes)
       (and (national-security No)
       (public-trust Possible)))
 => (assert (is-public-trust TRUE)))
```

Background Investigation

```
(defrule public-trust-BI
   (is-public-trust TRUE)
   (position-type "Civil Servant")
   (public-trust-level High)
 => (printout t "BI required.<p>"))
(defrule public-trust-LBI
   (is-public-trust TRUE)
   (position-type "Civil Servant")
   (public-trust-level Medium)
 => (printout t "LBI required.<p>"))
```

Figure 5. Sample Rule Set

the form fields and values are sent to the server. The server initializes the CLIPS expert system shell and sets the initial set of *facts* in accordance with the user interface responses. In the interaction presented in sample screen figures 1-3, some initial facts are that the position type is a civil servant (position-type "Civil Servant"), the employee has a military or federal work history (military-or-federal-

history Yes), (prior-investigation Yes), the position requires access to medical information (access-to-medical-info Yes) and the position does not require access to personnel information (access-to-personnel-info No). CLIPS operates on the facts by searching its rule base and *activating* all rules whose predicates match the set of facts. Our application first checks the rule set for conflicting facts and then processes special checks. Next, it determines the position sensitivity; and finally determines the appropriate investigation for the specified employee in the specified position. Each rule set contains *control rules* that fire only when all other rules in the set have had the opportunity to fire. When a control rule fires, it sets a *control fact* indicating that the processing of the current rule set is complete and the next appropriate set of rules can be considered.

In the knowledge base represented in Figure 5, the control fact (inputs-OK TRUE) indicates that all conflict rules have had the opportunity to fire, but that no conflicts exist. Similarly, the control fact (is-public-trust TRUE) indicates that all position sensitivity rules have had the opportunity to fire and that the position is one of public trust. Initially, only the *prior-service* and *conflict-summary* do not require the existence of control facts to fire. The *conflict-summary* rule is the control rule which is guaranteed to fire after all other rules in the conflicts rule set because it is declared with a lower salience. If facts are in conflict, the fact (conflict TRUE) is set; its existence prevents the *conflict-summary* rule from firing. If conflicts do not exist, *the conflict-summary* rule fires and produces the (inputs-OK TRUE) fact. This control fact is a predicate for all special checks and position sensitivity rules. In our example, there is no conflict in the user inputs. The control fact (inputs-OK TRUE) is set which permits the rules in the special checks and position sensitivity rule sets to be considered. When the *10450-applies* fires, a message containing embedded HTML code (<p> means start a new paragraph) is printed to the output stream, and thus to the Web browser. Also, when the *medical-info-sensitivity* fires, facts are set to indicate that the position is one of public trust. Finally, after all the rules in these sets have the opportunity to fire, the control rule *is-public-trust-rule* fires. It sets the *control fact* (is-public-trust TRUE) which, in turn, is the predicate for the background investigation rules *public-trust-BI* and *public-trust-LBI.* (BI stands for a particular investigation called Background Investigation; LBI stands for Limited Background Investigation). In our example, the rule public-trust-LBI fires and a message informing the user that an LBI is required is printed to the output stream, and thus to the Web browser.

Flexibility

The evolution from the first version (a questionnaire presented on a single page) to the second version (a more user friendly set of screens) required little modification to the rules. Only the rules made obsolete by the new user interface needed to change. Some of the original questions, and therefore some of the original rules, established a context identifying the type of employee and position. The second version established the employee and position type contexts via the user interface flow.

Application Use and Payoff

The system was deployed in July 1995. It is actively used by approximately 15-20 employees across 11 NASA field installations. The cost to produce the system was $60 thousand. Of the $1.5 million agency budget for personnel security investigations, it saved $1.2 million, or 80%, in less than one year. The head of personnel security for NASA used to spend 80% of his time helping personnel security specialists determine appropriate background investigations; now that the expert system is in use, he devotes less than 5% of his time to this activity. Personnel security specialists used to require several weeks to determine the appropriate background investigation for an employee; with the use of the expert system, they now complete their tasks in 10-15 minutes.

Soon, the White House Security Policy Board will issue new policies which will standardize security policies across all government agencies. We estimate that it will be trivial to incorporate the new rules into our system, thus saving $30 thousand in training costs for the personnel security specialists.

Conclusion

The system was well received by the user community and the NASA Security Office. It provides many benefits to the NASA Personnel Security Program. Since installation security staff were not utilizing manuals and applicable regulations appropriately, the Agency Personnel Security Specialist, or expert, was responsible for trouble-shooting and correcting all investigative problems. This tool saves the expert's time and allows him to pursue other pressing security issues. This system also provides a learning tool for security processing staff at each installation, by familiarizing them
with the appropriate questions to ask when processing investigations. The system ensures that each installation security office is in compliance with applicable laws, regulations and policies. Finally, $1.2 million, or 80%, of the agency's budget was saved in less than a year by eliminating overlapping, inappropriate and duplicative efforts to process employees.

The system design provided flexibility. By separating the underlying model from the user interface, both were able to be developed and modified independently. This made system modification easy and fast.

The rules were classified into rule sets. Through the classification process, we were able to remove the rules that established context from the rule set and design the user interface to capture this knowledge in its presentation and screen flow. This clarified the user interface in that the users now understand the process of determining

background investigations more clearly. Furthermore, it eliminated some of the rules established to perform classification.

Due to the system's success, plans are being developed to implement additional expert systems to aid in other areas of the NASA Security Program.

Acknowledgments

This work was performed for the NASA Security Office by Hughes STX. We would like to thank Rick Carr of the NASA Security Office and Frank Husson of Hughes STX for their support of this project. We would also like to thank Dr. Ralph Semmel and Marty Hall of the Johns Hopkins University Applied Physics Lab for their helpful review of this paper.

References

Berners-Lee, T. 1994. An Executive Summary of the World Wide Web.
http://www.w3.org/hypertext/WWW/Summary.html

Boutell, T. 1995b. What are WWW, Hypertext and Hypermedia?
http://sunsite.unc.edu/boutell/faq/hypertext.html..

Boutell, T. 1995c. "WWW FAQ Introduction: How Can I Access the Web?"
http://sunsite.unc.edu/boutell/faq/introaccess.html.

Farquhar, A., Fikes, R., Pratt, W., Rice, J. 1995. Collaborative Ontology Construction for Information Integration. Knowledge Systems Laboratory Department of Computer Science, KSL-95-63, Dept. of Computer Science, Stanford Univ.

Gruber. T. R. 1993. A Translation Approach to Portable Ontologies. *Knowledge Acquisition*, 5(2):199-220.

Giarratano, J.C. 1993a. *CLIPS User's Guild (CLIPS Version 6.0)*, NASA Lyndon B. Johnson Space Center.
Giarratano, J.C. 1993b. *CLIPS Reference Manual*, Vol. I and II, NASA Lyndon B. Johnson Space Center.

Hughes, K. 1994. Entering the World Wide Web - A Guide to Cyberspace." *http://www.eit.com/web/www.guide*.

Jackson, P. 1990. *Introduction to Expert Systems*. Wokingham, England: Addison-Wesley Publishing Company.

Luger, G.F. 1993. *Artificial Intelligence Structures and Strategies for Complex Problem Solving*, Redwood City, California: The Benjamin/Cummings Publishing Company, Inc.

McCool, R. 1993. "CGI - Overview and Introduction." *http://hoohoo.ncsa.uiuc.edu/cgi/* .