34067
P- 7

AI & Workflow Automation:
The Prototype Electronic Purchase Request System

Michael M. Compton
compton@ptolemy.arc.nasa.gov
(415) 604-6776

Shawn R. Wolfe
shawn@ptolemy.arc.nasa.gov
(415) 604-4760

AI Research Branch / Recom Technologies, Inc.
NASA Ames Research Center
Moffett Field, CA 94035-1000

Abstract:

Automating "paper" workflow processes with electronic forms and email can dramatically improve the efficiency of those processes. However, applications that involve complex forms that are used for a variety of purposes or that require numerous and varied approvals often require additional software tools to ensure that 1) the electronic form is correctly and completely filled out, and 2) the form is routed to the proper individuals and organizations for approval. The Prototype Electronic Purchase Request (PEPR) system, which has been in pilot use at NASA Ames Research Center since December 1993, seamlessly links a commercial electronic forms package and a CLIPS-based knowledge system that first ensures that electronic forms are correct and complete, and then generates an "electronic routing slip" that is used to route the form to the people who must sign it. The PEPR validation module is context-sensitive, and can apply different validation rules at each step in the approval process. The PEPR system is form-independent, and has been applied to several different types of forms. The system employs a version of CLIPS that has been extended to support AppleScript, a recently-released scripting language for the Macintosh. This "scriptability" provides both a transparent, flexible interface between the two programs and a means by which a single copy of the knowledge base can be utilized by numerous remote users.

Introduction

The Procurement Division at NASA Ames Research Center processes up to twenty thousand purchase requests (PRs) every year. These PRs, which all use a common form, are used to procure virtually anything used at the Center: computers, hazardous chemicals, office equipment, scientific instruments, airplane parts, and even funding for external research projects. PRs can be submitted by any civil servant employee at the Center, and must be approved by anywhere from three to twenty different individuals and offices. The average time required to submit a PR and obtain the necessary approvers' signatures is eighteen business days. Worse yet, approximately half of the PRs that are submitted are either incorrectly filled out, lack some required additional paperwork, or are routed to the wrong group for approval and must be returned to the originator. This not only delays procurement of the requested items but also burdens the system with a significant amount of paper flowing in the "wrong direction". In addition, the paper system lacks any mechanism for tracking a submitted PR, so people who originate these purchase requests often try to track them manually by picking up the telephone and calling around until they find where the PR is in the approval process. This, along with the numerous "walk-though" PRs, contribute significantly to the delays involved in processing the requests.

In 1991, the AI Research Branch at NASA Ames undertook a "weekends and evenings" effort to see whether a knowledge-based system, in conjunction with other advanced computing tools, could help expedite the process by which purchase requests are submitted, routed, and approved. The resulting system, called the Prototype Electronic Purchase Request system (PEPR), combines a commercial electronic forms package with a knowledge-based system that both ensures that the

submitted forms are correct and complete, and generates an electronic "routing slip", based on the content of various fields on the form, that reflects the approvals that particular PR requires. The PEPR system currently operates in a Macintosh environment and takes advantage of several new collaborative features of the latest release of the Macintosh OS, including digital signatures and "OS-level" electronic mail.

The system is now being used by several different groups at Ames to process a particular class of PR, namely those that apply to the funding of external research at colleges and universities. Initial results indicate that the system can dramatically reduce the time required to originate and process PRs and their supporting paperwork by ensuring that PRs entering the system are error-free and automatically routing them to the proper individuals. The system also provides a tracking capability by which the originator of a PR can query the system about the status of a particular PR and find out exactly where it is in the approval process.

Figure 1: An Example Purchase Request

Figure 1 shows an example PR. Because of its size and complexity, we have focused on the Ames Purchase Request form for the development of the PEPR system. However, our implementation is largely form-independent, and can be applied to other forms that require "complex routing". We have also applied the PEPR system to approximately six other types of forms and are actively pursuing other potential applications of the system both inside and outside of NASA Ames.

Why AI?

The knowledge-based component of the PEPR system utilizes a fairly straightforward rule- and object-based mechanism to provide its validation and routing capabilities (although we are

investigating machine learning techniques to ease the knowledge-acquisition problem -- see the section entitled Future Plans, below). There are two main reasons that a knowledge-based approach is appropriate to the problem of validation and routing of electronic forms.

First, the knowledge required to ensure the PR's correctness and completeness is quite diverse and very widely distributed among the various groups at Ames. Different validation "rules" come into play depending on what items or services are being ordered and what offices are required to approve a particular purchase. Early on in the project it became clear that these validation rules would have to be acquired and revised incrementally. In addition, the fact that different validation rules come into play at different stages of the approval cycle meant that the validation mechanism had to be both "item-sensitive" and "context-sensitive". By adopting a rule-based approach, we were able to design and implement a general mechanism for applying validation rules of varying complexity and then add and/or refine form-specific validation rules as they were discovered.

Second, the knowledge that we required to generate a correct "approval path" for a particular PR was not well-defined and distributed among a wide variety of people. We also recognized that in order to guarantee that the system would always generate a correct set of approvers, we would need to be able to incrementally add routing knowledge as "holes" in the existing routing knowledge became apparent. The inherent separation of "inference engine" and "knowledge base" in rule-based systems offered a clear advantage over a conventional procedural approach.

Why CLIPS?

We decided to use the CLIPS shell to implement the knowledge-based portion of the PEPR system for a variety of reasons. First, the data-driven nature of forms processing seemed to suggest that a forward-chaining inference engine would be appropriate. Second, CLIPS runs in a Macintosh environment, which is the platform of choice among our targeted pilot users. Third, the availability of CLIPS source code meant that we could tailor it to our specific needs (see [2] for a more detailed description of the modifications we made to CLIPS). Several other projects within our branch had successfully applied CLIPS to a variety of problems, so there was a fair amount of local expertise in its use. Also, the fact that it is available to government projects at no cost made it particularly appealing.

Key Design Requirements

To evaluate the suitability of a knowledge-based system in an automated workflow environment, it was of course necessary to provide other components of the workflow system. As a result, certain design issues and requirements were identified early in the project. The following represent key assumptions and design desiderata that probably apply to any automated workflow system:

• **Familiar user interface:** The electronic version of the form had to look very much like the paper form with which the users were already familiar. Also, the electronic form needed to perform the rudimentary operations that users have come to expect from any automated application (printing support, automatic calculation of numeric fields, etc.)

• **Reliable data transport mechanism:** In order to get the forms from user to user, the system had to utilize an easy-to-use and reliable electronic mail system.

• **User authentication:** Once users are expected to receive and process sensitive data electronically, they must be assured that the people who sent the data are who they claim to be. Therefore, our system needed to ensure authentication of its users.

• **Data integrity assurance:** Likewise, the users needed to be sure that the data they received had not been altered, either accidentally or intentionally, while in transit.

• **Seamless integration:** We wanted the operation of the knowledge-based component of the system to be completely invisible from the user and have its output appear as data on the form.

• **Tracking capability:** Enabling a user to determine where in the process a particular form is at any particular moment, without having to bother other users, is very important to the acceptance of an automated workflow system. We wanted our users to be able to determine the status of their submitted forms automatically.

Of course, we did not want to have to develop all of the mechanisms required to meet these key needs. Thankfully, most of the requirements described above had been provided by recently-released commercial products. Therefore, our goal in the development of the PEPR system was to make use of commercially-available technology to fulfill these requirements whenever possible, and to integrate the various software components as cleanly as possible. While this approach required us to make use of pre-release versions of some software components of the system (with many of the frustrations inherent in "beta testing"), it enabled us to focus on the development and integration of the knowledge-based component and also led to mutually beneficial relationships with the vendors whose products we utilized.

System Components

Because of the workflow-enabling capabilities of the latest release of the operating system, the availability of workflow-related products, and the relative popularity of the platform at Ames, we chose to implement the first version of the PEPR system on the Apple Macintosh. The PEPR system is comprised of several commercial software tools:

• **Expert System Shell:** As described above, we selected CLIPS with which to implement the knowledge-based portion of the PEPR system.

• **Electronic Forms Package:** The Informed™ package from Shana Corporation is used to produce high-fidelity electronic forms. This package is comprised of the Informed Designer™ program, which permits a forms designer to define the layout and functionality of the form, and the Informed Manager™ program which permits filling out the form by end-users.

• **Scripting Language:** The various software modules that comprise the PEPR system share data by means of AppleScript™, a scripting language for the Macintosh that allows the programs to interact with each other and share data, even across an AppleTalk network.

• **DBMS:** The PEPR system currently utilizes 4th Dimension™, a "scriptable" data base management system from ACIUS, to hold data associated with the routing and tracking of forms as they are sent from user to user.

These applications all operate together under version 7.1.1 of the Macintosh operating system (also known as "System 7 Pro") which provides system-level capability for electronic mail, digital signatures (for user authentication and data integrity assurance) as components of Apple's PowerTalk™ software product. The PowerShare™ system, which provides the store-and-forward mail service and user catalog support, operates on a centrally-located server system and supports all client users.

Each user of the system is required only to be running System 7 Pro and the Informed Manager application. CLIPS and 4th Dimension reside on a central "server" and can be accessed remotely by all users.

Knowledge Base Structure

The PEPR knowledge base is comprised of four main modules; the Validator, the Classifier, the Approval-path Generator, and the Organization "data base". In addition, each form that the PEPR system supports has its own set of form-specific validation rules that are loaded dynamically as the form is processed.

   • **Validator:** The PEPR validator is responsible for ensuring that the various fields on the form are correct and complete. The validation rules are represented as CLIPS classes, and are organized hierarchically with their own "apply" methods. Actual form-specific validation rules are represented as instances of these classes and are loaded dynamically from disk files when a particular form type is to be validated. If a validation rule is violated, the validator creates an instance of an error object with a suitable error message that eventually gets returned to a field on the form.
   • **Classifier:** If the validator finds no errors on the form, the Classifier is invoked. The Classifier uses the contents of specific fields on the form to construct hypotheses about potential categories to which the specific form might belong. The Classifier loads a group of form-specific "clues" that are comprised of a text string, a field name, a classification category, and a certainty factor. These clues are evaluated in turn; if the clue's text string is found within the associated field, then membership in the given category is established with the given certainty factor. These certainty factors can be positive or negative, and are combined using the CF calculus defined by Shortliffe *et al* for the MYCIN system. If the resulting certainty associated with a certain hypothesis exceeds a threshold value, then the form is said to belong to that category.
   • **Approval-path Generator:** Once all of the applicable categories for a given PR have been determined, the approval-path generator looks at specific fields on the form and determines the originating organization. It then loads the form-specific routing rules, and determines both the "management" approvals that are required (which depend upon the originating organization and, often, the total dollar amount associated with the PR) and the "special" approvals that are required (which are dependent on the classification categories to which the PR was assigned). These approvals are represented as the various organizations that must approve the form. The approval-path generator then looks up the "primary contact" for each of these organizations in the "organization data base" and inserts that person's name in the forms electronic "routing slip". (Note that by updating the "primary contact" for an organization periodically allows forms to be routed to designated alternates should the real primary person be on vacation or otherwise unavailable).
   • **Organization Data Base:** This portion of the knowledge base contains CLIPS objects that correspond to the various managerial groups and hierarchies at Ames, and is used to help generate approval paths, as described above. (Of course, this module is currently a very good candidate for re-implementation in some other format as an external data base, and the PEPR team is currently negotiating with other Ames groups who maintain similar data bases for other purposes).

Development History

The PEPR system has been under development on a part-time basis for the past three years. Since then, the system has undergone various changes, both in its architecture and functionality. In Early 1991, work began in investigating the problems associated with the procurement process and the potential applicability of software tools to help address those problems. The team identified a useful sub-class of purchase requests on which to begin work, namely those PRs associated with

the funding of research at external universities (this sub-class had the advantages of being reasonably straightforward with respect to the routing required and of providing an immediate near-term benefit -- the AI Research Branch submits a substantial number of these PRs and would therefore be in a good position to evaluate the utility of such a system). In June of 1991, the forms required to support university grants were distributed to a small number of users. These forms included only the evaluation forms (not the PR), and did not utilize the knowledge-based component. Early in 1992, the electronic forms were re-implemented in Informed, which proved to be a superior forms package to the that which had been used previously. These forms (except the PR) were given to numerous users around the Center, and were well-received. The knowledge-based validator, although working, was not deployed to end-users because we lacked a mechanism to efficiently share data between the form and the knowledge system. By the fall of 1992, we had initial versions of both the validator and the approval-path generator working, but they were only usable as a demonstration because they were not well-enough integrated with the forms system to be given to end-users. This "integration" was by means of a popular keyboard-macro package that allowed the two applications to clumsily share data via a disk file. However, this approach had two serious drawbacks. First, the keyboard macro package merely simulated the manipulation of the user interface, and so the user would have been subjected to a very distracting flurry of dialog boxes and simulated mouse-clicks. Second (and more importantly), that integration required that both the forms package and the knowledge base be running on the user's machine. That was an unacceptable limitation and would have undoubtedly "turned off" more users that it would have helped. We were, however, able to give the end-users electronic versions of the grant evaluation forms, which were somewhat helpful to the more experienced users even without the knowledge-based components. In early 1993, the team signed on as pre-release users of AppleScript, and modified the CLIPS shell to be "scriptable". This not only enabled a more "seamless" and less distracting integration between the forms package and the knowledge base, but more importantly it enabled us to set up a single copy of the knowledge system on a central server and permit users to access it over the network. By the summer of 1993, we became pre-release users of the new operating system software (part of the Apple Open Collaboration Environment) that provided support for digital signatures, system-level electronic mail, and other workflow-facilitating features. With these new features came the ability not only to give real users access to the knowledge base validation and routing capability, but also the data integrity assurance that would be required to support electronic submission of the sensitive data contained on financial instruments such as a purchase request form.

In December 1993, the PEPR system "went live", and is now in daily use within the Aerospace Systems Directorate at Ames for the electronic submission, approval, and routing of purchase requests and university grant evaluation forms. The system is even being used to electronically send grant award forms to selected universities, something that had previously been done manually by the University Affairs Office at Ames.

Future Plans

The PEPR team is currently supporting the University Grant pilot testers, and is in the process of making small refinements to the system as the users report problems and suggest improvements. In the coming months, we expect to be able to expand both the user base of the system and the scope of the purchase requests to which it is applied. We are also investigating other related workflow applications, both within Ames and at other government laboratories and within industry.

The PEPR team is also working very closely with researchers at Washington State University who are applying machine learning techniques to electronic forms. Our hope is that as our data base of "correct and complete" forms grows, we will be able to utilize these techniques to automatically generate new validation and routing rules.

References

[1] Compton, M., Wolfe, S. 1993 *Intelligent Validation and Routing of Electronic Forms in a Distributed Workflow Environment..* Proceedings of the Tenth IEEE Conference on AI and Applications.

[2] Compton, M., Wolfe, S. 1994 *CLIPS, Apple Events, and AppleScript: Integrating CLIPS with Commercial Software.* Proceedings of the Third Conference on CLIPS.

[3] Compton, M., Stewart, H., Tabibzadeh, S., Hastings, B. 1992 *Intelligent purchase request system,* NASA Ames Research Center Tech. Rep. FIA-92-07

[4] Shortliffe, E. H. 1976 *Computer-based medical consultations: MYCIN.* New York: American Elsevier.

[5] Hermens, L.A., Schlimmer, J.C. 1993 *Applying machine learning to electronic form filling.* Proceedings of the SPIE Applications of AI: Machine Vision and Robotics