Automatic Letter Composition for Customer Service

Stephen Springer, Paul Buta, and Thomas C. Wolf

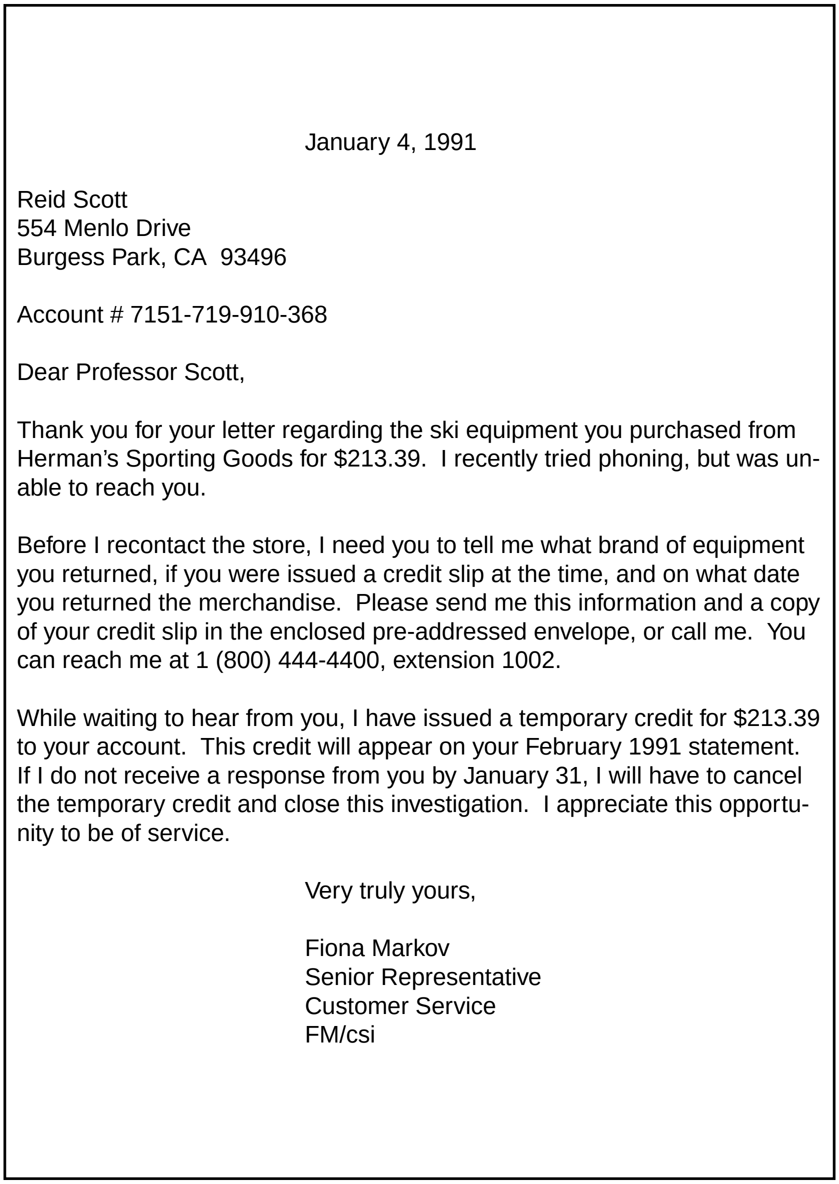
Corporations worldwide have seen a sharply increasing need for customized client correspondence in recent years. Financial institutions, mail-order companies, legal firms—any corporation that maintains a Customer Service Department—must generate written correspondence that clearly communicates specific, individualized information to clients. Indeed, in a slumping economy, much has been made of the need to differentiate one's business from the competition by providing higher-quality customer service (Agins et al. 1990). However, the industry continues to express this commitment through a combination of awkward form letters and expensive original letters composed by hand. This chapter describes an application of Cognitive Systems' Intelligent Correspondence Generator (ICG), which has increased the quality, reduced the complexity, and drastically cut the turnaround time associated with the production of personalized letters at its installation site. Users with little or no training in business correspondence can invoke the system to automatically compose complete, high-quality letters specifically tailored to the addressee's situation.

The Correspondence Problem

The application has been deployed at a major credit card organization for approximately 1 year. The needs of this organization are both complex and representative of most customer service organizations. At heart is a familiar trade-off between quality and cost. An automated form-letter system originally formed the core of this organization's correspondence facility. Because its letters must specifically discuss different kinds of financial transactions, the system has grown to include close to 1,000 different form letters to address the simplest divisions of common problems. However, in practice, most of these letters are never used: Customer service representatives, working under pressure to handle as many cases as quickly as possible, tend to use 10 to 20 letters that they are familiar with and that are close enough to describing the client's situation rather than take the time to discriminate between slight variations within the form library. When a client's situation even slightly varies from these forms or encompasses a combination of topics addressed in separate form letters, a new letter must be composed by hand if the client is to be convinced that s/he has received individual attention. Form-letter systems might come cheap, but they don't always stay that way, and the quality of output for any particular situation can never be very high.

High-quality correspondence, in fact, is difficult to guarantee even when left to human beings. Figure 1 is an example of the desired standard for writing quality, in this case, a typical request by the credit card organization for more information from the card holder. Being both professional and polite, the writer must manage to clearly discuss the nature of the problem, the amount in dispute, prior communications, the exact information required, how to contact the company, and several other transactions and events related to the central problem.

Customized letters such as the one shown in figure 1 are pleasing to the card holder: They are clear, concise, and polite and convey a sense that a particular representative has handled one's situation individually. However, it has proven extremely difficult for our credit card organization and other corporations we spoke with to consistently produce this level of quality. We noted numerous examples of customer service representatives slipping into an inappropriate tone, for example, when they couldn't bring themselves to be as polite as expected to a truly unreasonable card holder or when they became inadvisably familiar with a more deserving client. Customer service representatives also often operate under a quota system, having to handle a minimum number of cases each day, thus increasing the pressure to finish a letter and move on. In short, corporate letters written by hand might be as good as the



January 4, 1991

Reid Scott
554 Menlo Drive
Burgess Park, CA 93496

Account # 7151-719-910-368

Dear Professor Scott,

Thank you for your letter regarding the ski equipment you purchased from Herman’s Sporting Goods for $213.39. I recently tried phoning, but was unable to reach you.

Before I recontact the store, I need you to tell me what brand of equipment you returned, if you were issued a credit slip at the time, and on what date you returned the merchandise. Please send me this information and a copy of your credit slip in the enclosed pre-addressed envelope, or call me. You can reach me at 1 (800) 444-4400, extension 1002.

While waiting to hear from you, I have issued a temporary credit for $213.39 to your account. This credit will appear on your February 1991 statement. If I do not receive a response from you by January 31, I will have to cancel the temporary credit and close this investigation. I appreciate this opportunity to be of service.

Very truly yours,

Fiona Markov
Senior Representative
Customer Service
FM/csi

Figure 1. Typical Correspondence Example.

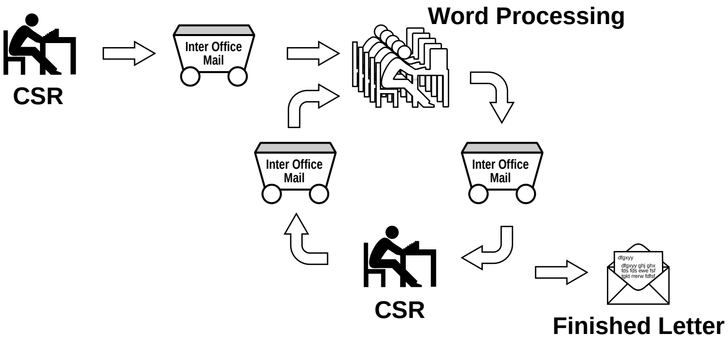


Figure 2. Standard Correspondence Work Flow.

example in figure 1, but they can just as easily contain anything from typographic errors to significant omissions to rude treatment. They can also appear formulaic if the customer service representative reverts to “corporate-speak” rather than carefully writes each new letter.

Customized letters have also been expensive to produce. Customer service representatives must be trained in corporate writing style, which can involve a combination of a few days in a training class and many hours of on-the-job review at our credit card organization. The customer service representatives must take the time (on the average, 45 minutes) to compose each letter from scratch—even though the letters almost always address relatively slight variations on familiar problems. A separate word processing staff must be maintained to type in, edit, and print the letters. The *turnaround time*—the time between receiving the client’s request and mailing a reply—was never less than three days and could be as long as two weeks if there were errors or miscommunication with the word processing staff. The general confusion of the standard correspondence work flow is detailed in figure 2.

As figure 2 shows, each letter is drafted and written in longhand by a customer service representative; forwarded to word processing; printed; returned to the customer service representative; reviewed; sent back to word processing if errors are discovered (or if circumstances have changed since the letter was first drafted!); and, eventually, printed and mailed.

Our task was twofold. First, we had to design a text-generation system capable of generating a wide variety of quality output in a specific corporate style in a few seconds. Second, we had to ensure that such a system could be integrated into the credit card organization’s work environment, could effectively be used by designated users with limited

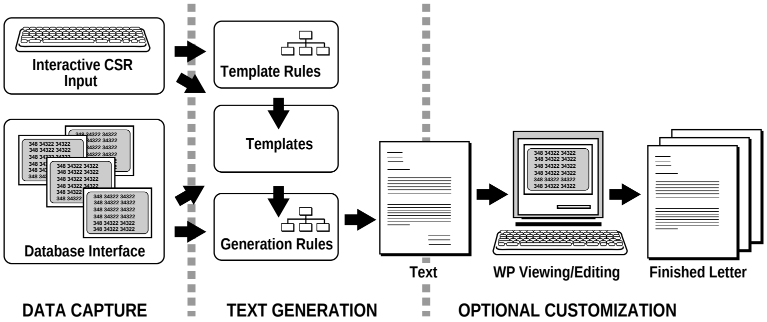


Figure 3. ICG Functional Breakdown.

writing skills, and could improve the existing production routines required by the high-volume individualization of letters.

In fact, the entire letter shown in figure 1 was composed, printed, and mailed by a customer service representative using the ICG system. (The only changes made to the letter involve the account number and some direct word substitutions to protect the anonymity of the card holder and the organization.) For a limited group of customer service representatives, ICG has already replaced the entire form-letter system and the write-from-scratch method. It is currently producing letters daily that exceed the organization's standards for handwritten letters and reduced or eliminated the need for separate word processing and quality control staffs.

The Text-Generation Technology

The functional breakdown of the ICG technology is shown in figure 3.

ICG's functional architecture is similar to the Penman system (Mann 1983). Internally, it is organized around a blackboard-based expert system, supplemented by a text-generation controller and utilities. All data and inferences involved in generating a document are placed on the blackboard, which provides a common place to organize information from all the modules.

When composing a document, ICG executes the following steps: First, the information that is needed to generate the document is collected from the credit card organization's existing client database and the customer service representative. This information is placed on the blackboard. Second, inference rules operate on the blackboard to organize the information, building a frame-based model of the situation to be discussed as well as other related concepts. This model is independent of the output language. Third, template rules determine the basic structure of the document to be generated. Fourth, based on the set of templates selected, generation rules perform lower-level text planning and realization, building a complete letter. Fifth, the final document is formatted and presented to the user in a Microsoft Word document.

Data Capture

The system's input is collected through both an interactive user interface and a transparent interface to the organization's IMS database. When the user invokes the system, ICG first scans the database fields for the account most recently displayed and makes inferences from what is available, retrieving baseline information such as name and address, identifying relevant financial transactions and recorded prior communications, and performing a rough problem categorization based on the pattern of transactions identified. That is, before the first question is asked, ICG can already have modeled much of the situation to be discussed. Then, the user interface only needs to request a basic confirmation and a few data items not available online. This information is collected by asking the user to supply multiple-choice answers to a series of questions presented in an on-screen, color-coded form. Because the form modifies itself based on each multiple-choice answer as it is supplied by the customer service representative, interactive dialogs are usually no longer than 15 queries.

The Blackboard

We chose a blackboard-based design (Engelmore and Morgan 1988) to facilitate communication between the different ICG modules. The blackboard serves as a repository for information retrieved by the database and dialogs as well as a place to organize inferences about the information and the generation process. The information is stored on the blackboard as object-based frames that can be retrieved by unification patterns similar to those found in OPS5 (Brownston et al. 1985).

The objects described in frames are organized in a knowledge hierarchy that allows the inheritance of information such as lexical reference methods, origin of information, and relationships to other objects. The installed ICG application includes definitions for well over 600 objects, including the associated methods for generating text to describe many of them. These concepts include people (card holders, customer service representatives, store employees, and so on), institutions (the credit card organization itself, stores and other types of service providers, branches of the federal government, and so on), various kinds of merchandise and services that can be purchased with a credit card, events associated with using a credit card (purchase, signing of receipt, third-party shipments, merchandise return, and so on), account transactions (credits, debits, payments, and charges that can be temporary, permanent, and periodic), policies and contracts (of both the organization and the service providers), dispute classifications and problem resolutions, documents (account statements, sales receipts, shipping receipts), dates (specific and month-long time periods), and money amounts.

The system also includes representations of generation-specific concepts, including tone (both of the card holder and the reply being generated), stylistic connectives in text, and the structure of the letter being generated (objects for sentences, paragraphs, and the letter).

Text Generation

Text generation has traditionally been considered a three-part process: content determination, text planning, and realization (Grosz, Sparck Jones, and Webber 1986). *Content determination* in ICG roughly corresponds to the rule system's task of organizing the information collected into the appropriate frames on the blackboard. *Text planning* in ICG occurs at two levels: First, *template-selection rules* are checked to determine the outline of the letter to be generated. These rules can combine several predefined groupings of topics to assemble a basic ordering of topics to be discussed. Based on this ordering, *generation rules* then organize the paragraphs and sentences. Unlike most systems (McKeown 1985; Nirenburg, Lesser, and Nyberg 1989), ICG does not have a formal planning mechanism to control text organization. Forward- and backward-chaining rules simply determine the relevant content and its general organization.

In the *realization* process, ICG traverses a large library of generation rules. Generation rules are indexed off particular topics, as described in the letter outlines. Their associated actions modify the letter object being built on the blackboard, adding a date line or sentence or starting a new paragraph. Unlike the vast majority of generation systems, ICG deliberately does not require that text be generated from an internal knowledge representation. All text to be generated is expressed in generation rules as strings (figure 4). Although ICG prescans these strings to recognize references to blackboard objects, variable text, and limited syntactic structure, it does so only to identify sufficient information to make the addition of these nuggets of text flow smoothly in the

```
defgenrule   deposit-as-additional-charge
  :template  additional-charge-template
  :section   bill-explanation
  :test      (and (deposit-on-separate-receipt    :deposit ?deposit
                                                  :total-charge ?amt)
                  (?sp = service-provider)
                  (?service = service)
                  (case   :disputed-charges (? :charge ?charge)
                          :account ?account))
  :action    START-NEW-PARAGRAPH
             ADD-SENTENCE
               "the ?sp's itemization shows a total charge of ?amt"
             ADD-SENTENCE
               "as you know, you initially left a ?deposit with ?sp"
             ADD-SENTENCE
               "~because the total amount of the ?service was
               more than the deposit, ?sp has submitted an
               additional ?charge to your ?account"
```

Figure 4. Sample Generation Rule.

final letter. In this way, we avoided the pitfall of having to implement a complete grammar for text generation, which is dangerous both because of its inherent difficulty and because we are not aware of any other current generation research that is flexible enough to be precisely adapted to the writing style of a given corporation.

Our philosophy resembles that of a phrasal lexicon, which was first proposed by Becker (1975) for language understanding. The use of a phrasal lexicon was more recently adapted to the problem of text generation in Hovy's (1988) PAULINE system (see also Kukich [1988]). We have essentially taken the notion of generating text from a phrasal lexicon to the extreme: The *phrases* are unusually long (generally sentence-length) snippets of the corporation's specific wording preferences regarding familiar aspects of its business. The phrasal lexicon's keys are complex unification patterns that describe the situations the phrases apply to; these patterns are organized by the system's generation templates. We do not mean to say, however, that we are simply stringing together canned sentences. A combination of shallow parsing by the system and judicious analysis by the application's knowledge engineer allows us to abstract out the conceptual aspects of each phrasal unit. The ICG system then supplies the utilities for reconstructing cohesive text from these processed phrases. For example, the most significant system utility is ICG's noun phrase generator. With each new reference to an object in the letter, the noun phrase generator consults the system's discourse model to see how it previously referred to it as well as to other similar objects. Based on the model, it either chooses to use a pronominal reference or selects the minimum set of attributes necessary to uniquely and naturally identify the object. Cognitive Systems' ICG technology is described in further detail in Buta and Springer (1990).

Implementation and Deployment

In this section, we discuss the project schedule, the platform and environment, and the knowledge engineering process.

Project Schedule

Our contract with the credit card organization allowed six calendar months of development followed by rigorous acceptance testing and a limited production rollout to three representatives in a designated customer service unit. This development and deployment stage was to be followed by a two-month period of postrelease bug fixing and support by Cognitive Systems. The development period included three phases with interim deliverables at three months, five months, and six months from project start. Cognitive Systems had already developed and tested a working ICG engine at the start of the project, although it had done no modeling of credit card situations in general or of the credit card organization's specific environment, problems, or policies. Thus, each delivery consisted mainly of a new library of objects, rules, and templates to cover a new domain area. (The first interim deliverable also included the system integration necessary for initial installation.) This approach ensured that with each deliverable, the customer service representatives could use the entire system, from dialog to printed letter, and would know in advance of the extent of the system's knowledge. In our planning, we took care both that knowledge engineering was equally divided among the deliverables and that users would intuitively grasp what types of letters they could expect the system to generate at any given time.

Platform and Environment

The original working environment for the customer service representatives was a mainframe computer, with a separate word processing system and typing staff. In the new environment provided with ICG, the customer service representatives have full control over the entire dispute-resolution process. The credit card organization had already begun to upgrade the customer service representatives' platform before the ICG project. The customer service representatives involved in the ICG project work on IBM PS/2s (Model 70) running MS-DOS (with an extended memory manager) that are linked to the mainframe. With ICG, each customer service representative has conventional terminal emulation with full mainframe database access to continue standard problem-resolution tasks; the ICG user interface; Microsoft Word for viewing the generated document; Grammatik, the grammar and spelling checker (not needed for ICG text—the letter generated by ICG includes control characters to actually block Grammatik from spell checking proper names—but used for any changes made by the customer service representative); direct hookup to an easily accessible laser printer; and function-key support to facilitate movement among these elements.

The dispute-resolution process is generally the same as before ICG. The customer service representative studies the case and takes whatever actions s/he thinks are appropriate. These actions can include issuing a temporary credit, contacting a service provider, or ordering a copy of a document. As before, the specific actions taken are stored in a mainframe database entry that describes the case. When correspondence is required, the writer invokes ICG. As previously outlined, ICG reads the database information about the current case and infers as much as possible from the prior actions recorded in the case database. In the example letter in figure 1, ICG noted the logging of an attempted telephone call prior to its invocation and the issuing of a temporary credit. Next, ICG runs through its interactive dialog with the customer service representative, verifying unclear database references and asking for any new actions the customer service representative intends to take as well as for composition details such as the appropriate tone for the outgoing letter. ICG then switches the customer service representative into Microsoft Word with the letter displayed. If any changes are made, they are automatically passed through Grammatik. Additionally, the user has the option of returning to the dialog screen to change any of his(her) earlier responses. When finished, the letter is printed on a local laser printer. The customer service representative simply folds the letter and places it in a windowed envelope along with any relevant enclosures.

In addition to generating the letter, the ICG system also generates a cover sheet that is printed along with the letter. The cover sheet records background information relevant to the credit card organization, including the card holder, the customer service representative, and the general dispute information; the amount of time spent in different ICG tasks, such as interactive dialog and database retrieval and generation time; important internal inferences used in producing the letter; and other related information. These tracking data are produced, formatted, and printed by the same utilities that generate the card holder's letter.

In short, the customer service representatives had to learn a total of two new screen environments: the ICG dialog screen (which always displays relevant choices and provides context-sensitive help for each query) and Microsoft Word.

The maintenance interface to the system is a standard programmer's interface (file editing and compiling, and so on). Maintenance does not require knowledge of any programming language other than the syntax of ICG's dozen or so restricted forms. Significant changes to the system currently require a person at the level of an expert system engineer. Cognitive Systems is developing a graphic user interface for the maintenance and development of ICG applications to simplify these tasks.

Knowledge Engineering

Knowledge engineering for ICG—the principal task across the six months of development—presented a challenge to us for two different reasons:

First, substantial effort was required to simply model the situations that would be discussed in letters. A credit card domain requires representations for all possible relationships between at least three actors: the credit card organization, the card holder, and the provider of the service or merchandise. When it was necessary to distinguish between the credit card organization and the customer service representative or a department store and a store employee, the situation grew even more complex. Although we had experience building a much narrower and deeper application (which discussed billing misunderstandings for a mail-order company), we found the credit card domain to be considerably broader and somewhat shallower. The first two full months of the project were spent in designing the overall structure of the knowledge to be represented.

Second, knowledge engineering for text generation in particular requires the consideration of additional factors, involving the communication of information. Some of these factors include the overall content and organization of the letter, the importance of the writing-style components and the manner in which they are modeled, the organization and presentation of the events, the handling of a combination of topics, the referencing of the parties, contributions to the card holder's satisfaction on reading the letter, and the company's efforts to guarantee this satisfaction.

Conveniently, for the credit card organization's writing experts, even a knowledge engineer with limited writing skills can appreciate what an expert correspondent includes in a letter. In addition, most corporations have a vast supply of reference material for such a system in the form of existing form letters and recent original letters. The Cognitive Systems knowledge engineers took a large collection of the credit card organization's recent correspondence and reduced the letters to their component topics and wordings (introduction, account actions, and so on), deducing the appropriate text rules in many cases. However, as we expected, many inconsistencies and questions arose. Resolving these writing issues formed a substantial part of the knowledge engineering process throughout development. We had little research to fall back on in this case. To our knowledge, although many text-generation investigations have considered questions of *linguistic competency*—the ability to generate various constructions and communicate various intents—few have attempted to build a system within the extremely tight *stylistic constraints* imposed by the correspondence standards of a corporation. (See, however, Hovy [1987] for a discussion of pragmatic criteria in text generation.)

The final ICG application contains over 100 composition templates, which can be nested in different ways to produce an overall letter outline. It contains almost 900 rules, about half of which are generation rules, whose actions construct the final document.

Results

In this section, we discuss project results in terms of system acceptance, speed enhancements, and quality enhancements.

Acceptance

We evaluate acceptance of the system by how the ICG users responded to it as well as by formal software acceptance procedures.

As it turned out, the customer service representatives liked the system so well that they started sending letters out to card holders after the first deliverable. That is, the system was put into production far ahead of schedule, after only two months of design work and one month of knowledge engineering. A number of factors contributed to the rapid deployment:

First, because ICG output directly appears on the customer service representative's screen in a word processor, the customer service representative could review and modify all generated text. Second, considerable general knowledge was included in the initial deliverable that was applicable across all problem classifications. For example, if the customer was receiving a credit, ICG could generate a general description of the account action regardless of missing knowledge about the problem classification. Third, when the system failed to fully understand part of a situation, it left this section of the letter blank. The writer could simply fill in the missing details.

Now that the project is complete, the ICG users almost never modify text generated by the system. Changes to letters only occur when the customer service representative wants to add some piece of information beyond the system's stated capabilities.

Formal acceptance required that the system generate a set of test cases with 100 percent accuracy, that is, with no errors. Each letter was judged using the following categories within a quality matrix: opening and closing paragraphs, grammar, spelling and punctuation, tense, length of clauses (clarity of individual sentences), succinctness of entire letter, clarity, the addressing of each of the card holder's issues, and correct information extracted from the database.

In addition to receiving a perfect score from the set of test cases, the system had to achieve an 80 percent hit rate on a random sample of 100 cases. That is, it was expected to generate 80 percent of the letters without a single error.

ICG successfully passed both acceptance tests by the end of the 8-month contract. At the time of this writing, the system has been deployed for over 1 year. No changes in system behavior have been requested since the project was completed over 6 months ago. Currently, 3 customer service representatives use ICG exclusively to generate their correspondence, averaging a total of 20 letters a day.

Training

ICG required little training. The system was designed by implementing a user model to index the various system templates and specific textual-content decisions. Therefore, if the customer service representative understood enough to resolve the card holder's dispute, s/he should know enough to go through ICG's dialog without referencing a manual. The only training necessary was syntactic—which button to press to invoke the system, how to perform various edit operations in the word processor, and how to use the grammar checker. The representatives received a cheat sheet of about 10 operations that are needed to use the system.

The customer service representatives who initially used the system received 4 hours of introduction and training before they started sending letters out live on their own. They received no additional training from us after this first training session. In fact, these representatives then trained other users to use the system. The credit card organization informed us that customer service representatives can now be productive with only 30 minutes of training. This situation can be com-

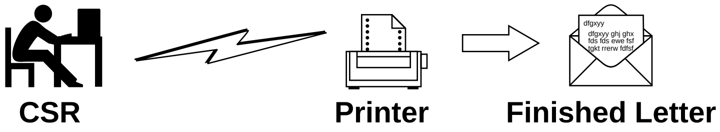


Figure 5. Correspondence Work Flow with ICG.

pared with a several-day course needed for certification in the use of the credit card organization's form-letter system.

Another major saving in training came from the system's ability to replace many of the previously handwritten letters and to frame letters that it can't produce in detail. As mentioned previously, one of the unit's biggest problems involved training its representatives to become quality letter writers. Writing high-quality letters is a skill that the credit card organization taught in the production environment by making supervisors randomly review the outgoing letters of the customer service representatives. This process was time consuming for the supervisor and complicated the education of the representative because s/he did not have samples of high-quality letters with which s/he could compare his(her) writing. However, because ICG continually illustrates good letter-writing style by example, a customer service representative can more quickly become proficient at writing in the same style. Thus, customer service representatives can produce better-quality letters with less training when they need to add text to an ICG letter or write a letter from scratch if it is outside ICG's target domain.

Production Turnaround

We mentioned previously that customer service representatives began using the system after our first three-month deliverable. In fact, acceptance went beyond just occasional use of ICG. The supervisor of our customer service representatives instructed them to stop using both the form-letter system and the word processing staff within a few weeks of the initial ICG installation. Even during development, using ICG to help write a letter was preferable to drafting one from scratch and relying on the word processing department. The high-level architecture of the system allowed the standard work flow to be drastically simplified. Compare figure 5 with the old work-flow pattern shown in figure 2.

The turnaround for a handwritten letter before ICG averaged three days, possibly taking as long as two weeks. Generating a letter with ICG now takes a representative—from the time s/he invokes the system to the time s/he stuffs the envelope—an average of five minutes.

Employees' Time

The actual time the customer service representative spent writing a letter before ICG was installed varied considerably, depending on how proficient s/he was at writing. Most representatives would spend a minimum of 45 minutes writing a letter by hand. A typist in word processing would then take a few minutes to enter the letter and would frequently need to contact the representative because of illegible handwriting or unclear instructions. ICG cut the customer service representative's involvement to the same 5-minute average previously mentioned (once all queries are answered, the finished letter is generated in under 30 seconds) and eliminated the need for word processing and interoffice mail personnel to even know that a letter was sent.

Unfortunately, at this time, we cannot estimate the cost savings to the credit card organization based on these reductions. There has been an increase in productivity, but computing a dollar amount would require, among other things, statistics such as the percentage of the word processing staff's time spent preparing these letters. The credit card organization is currently preparing such estimates. Meanwhile, we caution that such savings are only one part of ICG's contribution to profit. There is, perhaps more importantly, the harder-to-quantify goal of increasing client satisfaction and loyalty through higher-quality customer service.

Quality

In the first random sample by the credit card company's Quality Control Department (at the beginning of acceptance testing), 90 percent of ICG's letters were rated errorless. This percentage compared to about an 80 percent errorless score for handwritten letters sent without ICG. Many of these original errors dealt with capitalization or expansion of abbreviations of fields from the client database. Because ICG converts all name and address fields to mixed-case text and expands abbreviations, mistakes in either of these routines were considered an error. It is important to note that the form-letter system did not need to worry about converting to mixed case. The system always left all database fields in capitals. The quality control group did not consider this problem an error! These and other errors are now fixed, and over 95 percent of the letters produced with ICG are error free.

Conclusion

This installation of ICG bodes well for future applications. The system is well liked by both its end users and the credit card organization's management. It has made the customer service representatives' job more enjoyable, improved letter quality, and slashed turnaround time. Most importantly, we found the credit card organization site to be highly representative of other customer service organizations, both in form and in the types of correspondence issues it must address.

Much of what we learned building this application was general. We have now developed a substantial knowledge base associated with general customer service, mail-order, and credit card business. Building the next application in any of these fields will be substantially easier than our first effort. In fact, long-term plans include the development of several market-specific knowledge base modules.

We also found that ICG's design philosophy agrees extremely well with the way people actually think about and undertake the task of producing institutional correspondence. At the same time, its functional behavior dovetails nicely with the way many customer service operations go about their business. Future work will also include designing an advanced developers' interface to the system to allow design and enhancement by analysts with less formal training and in less time as well as run-time modifications to support ICG's deployment in different customer service environments.

Finally, we continue to investigate the somewhat delicate trade-off between providing a generation system that adheres to any level of stylistic constraint and providing a system that demonstrates the most power to create original compositions. We are convinced that any institutional use of text-generation technology requires both capabilities.

Acknowledgments

The authors want to gratefully acknowledge all who contributed to the success of this project, especially Mark Kriegsman, Vic Lanzillotti, Mark Wachsler, Park Walker, and Philip Werner.

References

Agins, T.; Bennet, A.; Deveny, K.; Milbank, D.; Fuchsberg, G.; Solomon, J.; and Rigdon, J. E. 1990. Customer Service: Challenge for the '90s. *The Wall Street Journal,* November 12–20 (multiple-issue series).

Becker, J. 1975. The Phrasal Lexicon. In *Theoretical Issues in Natural Language Processing,* 70–73. Cambridge, Mass.: Association of Computational Linguistics.

Brownston, L.; Farrell, R.; Kant, E.; and Martin, N. 1985. *Programming Expert Systems in OPS5*. Reading, Mass.: Addison-Wesley.

Buta, P., and Springer, S. 1990. OMBUDSMAN: The Correspondence Generation System. In Proceedings of the Tenth International Workshop on Expert Systems and Their Applications: Natural Language Processing, 195–204. Avignon, France: EC2.

Engelmore, R., and Morgan, A., eds. 1988. *Blackboard Systems*. Reading, Mass.: Addison-Wesley.

Grosz, B.; Sparck Jones, K.; and Webber, B., eds. 1986. *Readings in Natural Language Processing*. San Mateo, Calif.: Morgan Kaufmann.

Hovy, E. 1988. Generating Language with a Phrasal Lexicon. In *Natural Language Generation Systems,* eds. D. McDonald and L. Bloc, 353–383. New York: Springer-Verlag.

Hovy, E. 1987. Some Pragmatic Decision Criteria in Generation. In *Natural Language Generation*, ed. G. Kempen, 3–17. Dordrecht, The Netherlands: Martinus Nijhoff.

Kukich, K. 1988. Fluency in Natural Language Reports. In *Natural Language Generation Systems*, eds. D. McDonald and L. Bloc, 280–309. New York: Springer-Verlag.

McKeown, K. 1985. *Text Generation*. Cambridge: Cambridge University Press.

Mann, W. 1983. An Overview of the PENMAN Text-Generation System. In Proceedings of the Third National Conference on Artificial Intelligence, 261–265. Menlo Park, Calif.: American Association for Artificial Intelligence.

Nirenburg, S.; Lesser, V.; and Nyberg, E. 1989. Controlling a Language Generation Planner. In Proceedings of the Eleventh International Joint Conference on Artificial Intelligence, 1524–1530. Menlo Park, Calif.: International Joint Conferences on Artificial Intelligence.