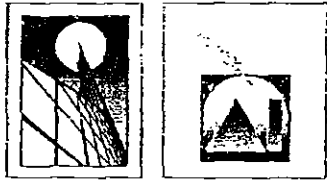# EXHIBIT B

Proceedings *of the*
Thirteenth National Conference on
Artificial Intelligence and the

Eighth Innovative Applications of
Artificial Intelligence Conference

Volume Two

Sponsored by the
American Association for
Artificial Intelligence

AAAI Press / The MIT Press

Menlo Park · Cambridge · London

Manufactured in the United States of America on acid-free paper.

BR 001251

EZ Reader: Embedded AI for Automatic Electronic Mail Interpretation and Routing

Amy Rice and Julie Hsu
Brightware, Inc.

Anthony Angotti and Rosanna Piccolo
Chase Manhattan Bank, N.A.

Abstract

EZ Reader is an intelligent electronic mail (email) reader that employs a unique combination of rule-based parsing and case-based reasoning to automatically and with a high level of accuracy classify and respond to large volumes of incoming email. EZ Reader reduces the time and human resources required to handle incoming email by selecting responses and adding attachments and advice to each incoming message based on how previous similar messages were handled. The application, developed for Chase Manhattan Bank using Brightware, Inc.'s ART*Enterprise® tool, answers emails automatically and decreases processing time for those requiring manual review. Phase 1 of EZ Reader was deployed in the first quarter of 1996, and handles up to 80% of incoming mail automatically, depending on message content. Later phases will enable automatic processing of a wider variety of messages. By dramatically reducing the effort associated with manual processing, EZ Reader will pay its own development costs within six months and will result in substantial, recurring dollar savings each year. This paper describes EZ Reader in detail, including its AI-based design, testing, implementation and development history.

Problem Description

Like other businesses that sought to expand access to their products and services through the Internet and other online channels[1], ChaseDirect, a unit of Chase Manhattan

---

Contact information follows. *Amy Rice:* 301 Tresser Blvd., 13th floor, Stamford, CT 06901, rice@brightware.com. *Julie Hsu:* 2010 Corporate Ridge, Suite 700, McLean, VA 22102, hsu@brightware.com. *Anthony Angotti:* One Chase Manhattan Plaza, 19th floor, New York, NY 10081, anthony.angotti@chase.com. *Rosanna Piccolo:* 15 E. 26th St. New York NY 10010, rosanna.piccolo@chase.com.

Bank N.A., Regional Bank, began to provide electronic banking services using phone and personal computer technology in 1995. Marketing campaigns advertised that email could be used to request information and services, opening a new electronic channel of communication with customers and prospects.

The success of its marketing campaigns created a challenge for ChaseDirect from the beginning to quickly and cost-effectively process email from multiple sources, including the Internet, Microsoft Money email, and another internal DOS-based money manager program with email capability. In addition to ChaseDirect's commitment to provide excellent, timely service to its customers, electronic commerce laws required the bank to respond to certain types of electronic correspondence within specific time frames. Although more than 80% of incoming messages were simple requests for product information, the staff often got backlogged and worked after hours and on weekends to keep up with the required analysis and responses. Faced with the huge projected increase in Internet email volume due to the planned introduction of a new World Wide Web server, as illustrated in Figure 1 below, ChaseDirect aggressively sought cost-effective, high-quality ways to process emails. This urgent business problem attracted attention from Chase's Regional Bank Knowledge Base (KB) technology team. The team's general charge was to apply artificial intelligence (AI) technology in key areas of the Regional Bank where appropriate to optimize operational decisions.

To address ChaseDirect's business problem, the Knowledge Base team created EZ Reader, an embedded AI application operating as an invisible layer between the Lotus Notes® email system and ChaseDirect. The application continuously retrieves incoming Internet email from Chase prospects and customers through an interface to Lotus Notes, and also acts as a filtering and routing

---

[1] Banks face significant threats to the retail banking franchise from advances in online banking (Taylor, Mehta & Wurster 1996)

Volume

2x

1x

Internet
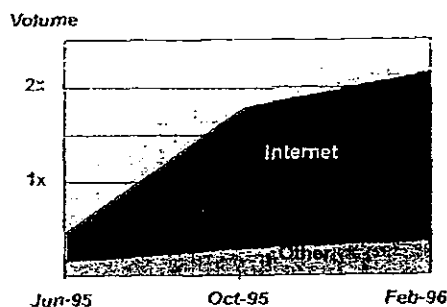
Other

Jun-95   Oct-95   Feb-96

Figure 1. Projected Growth of Incoming Email by Source

mechanism, either replying to the email automatically or attaching a suggested response and referring the message for manual review. The KB team's overall business goal for EZ Reader was to reduce the number of emails that needed manual processing by more than 80%, with an accuracy rate of 95%. Other goals were to provide rapid turnaround time for return messages to customers and prospects and consistency in responses. EZ Reader's automated reasoning capabilities enabled ChaseDirect to reach these goals and significantly reduce the manual effort needed for email processing.

EZ Reader Design

The ChaseDirect email processing problem offered several clear opportunities for the implementation of automated reasoning.

AI in EZ Reader

The email review process involved complex reasoning. The process was distinctly knowledge-intensive; specialists applied domain-specific knowledge using heuristics based on knowledge of Chase products and services. The reliance on knowledge as a process component was evident as junior team members routinely relied upon the experience of their supervisor and other senior team members to explain how to classify ambiguous messages and formulate responses for new classes of messages. Finally, common reasoning tasks were reflected in the process output: *classification* of the email and *configuration* of a response from a limited set of prepared text modules.

EZ Reader's AI reasoning component is a data-driven forward-chaining rule parser operating in concert with case-based reasoning written in Brightware, Inc.'s

ART*Enterprise, a commercially successful knowledge-based application development tool. As described below, the Knowledge Base team considered and rejected several alternative software applications and tools for implementing the reasoning component of EZ Reader.

In-house procedural application languages such as COBOL could not handle the reasoning component of the email process because the input to the process was so complex and unpredictable. An approach of coding branches for each potential input using procedural methods alone would have been impractically complex. The Rete algorithm used by ART*Enterprise's pattern matcher addressed this input complexity issue for EZ Reader. ART*Enterprise's integrated rule scripting language, case-based and object modeling capabilities enabled automated reasoning to reverse or change conclusions drawn throughout sequential parsing and to disambiguate text while ignoring irrelevant portions, such as signature lines, without coding explicit subroutines.

Although input was in the form of natural language, the output was a simple classification based on the application of heuristic rules and experience. For this reason, syntactic analysis and interpretation, discourse analysis and pragmatic inferencing in commercial natural language processing (NLP) products such as SRA or Logicon, or augmented transition networks in custom developed applications such as the Intelligent Banking System (Sahin & Sawyer 1989) were viewed as providing only an pre-processor function to the classification reasoning task, and not providing a value-added output *per se*. This is illustrated by the fact that the ChaseDirect human email reviewers were able to classify a large majority of messages by using a relatively limited set of key linguistic clues that could be expressed as simple rules. Furthermore, when human reviewers had trouble identifying common linguistic clues in a particular message, they relied upon team experience to help them classify the message and configure or synthesize a response. Compared with the time to create rules and case structures in ART*Enterprise, the time frame to fully analyze the semantics of the text interpretation domain and develop a comprehensive NLP module was deemed too long to meet business needs, and would have provided, at best, secondary functionality at greatly increased cost and technical complexity.

Fundamental technical requirements for an automated solution included compatibility with existing hardware and standards for run-time and network performance stability; scalability, vendor support and commercial viability. Widely-available shareware, proprietary intelligent programs specifically written for electronic forms processing (Compton & Wolfe 1993) and packaged email router/responder software such as Prolog-based

Mailbo ™ (Daxtron Laboratories, inc.) were rejected for these reasons. In contrast, the ART*Enterprise tool had already been deployed in another application at Chase where broad functionality and customizability, product reliability and vendor support necessary for fielding commercial applications was previously validated.

Since the content, number and ordering of concepts in incoming emails was unpredictable. automatic processing could not easily be accomplished by conventional procedural programming techniques. The full-featured AI capabilities of the in-house knowledge-based development tool afforded a quickly implementable, one-tool technique for transforming the linguistic clues in emails to output classifications.

Process Flow

Figure 2. illustrates the flow of an email through the EZ Reader system as described below:

1. The customer sends an email to Chase Manhattan Bank's Internet address.

2. Chase's corporate email router passes the message from the domain server to ChaseDirect's Lotus Notes server.

3. EZ Reader periodically checks the inbox (a Lotus Notes mail database) for new mail. When a new email arrives in the inbox, EZ Reader retrieves the message and "interprets" it by performing rule-based parsing and case-based retrieval. The outcome of its interpretation is one of two possibilities:

   a) EZ Reader can respond to the email automatically. An automatic response, which is routed directly to the ChaseDirect outbox, consists of the original email and one or more attachments, or prepared replies, that are retrieved from a Lotus Notes repository of standard responses.

   b) EZ Reader cannot respond to the email automatically. It refers the email to ChaseDirect for human review and response. Before placing the email in the manual review inbox, EZ Reader assigns a category and priority to the message and suggests one or more standard replies based on message content. (Categories and priorities are described in more detail later.)

4. ChaseDirect specialists review and write responses to all messages referred by EZ Reader and place these electronic replies in the outbox.

5. Chase's corporate email router routes the message from ChaseDirect's Lotus Notes mail management system and places it in Chase's domain server for reply back through the Internet.

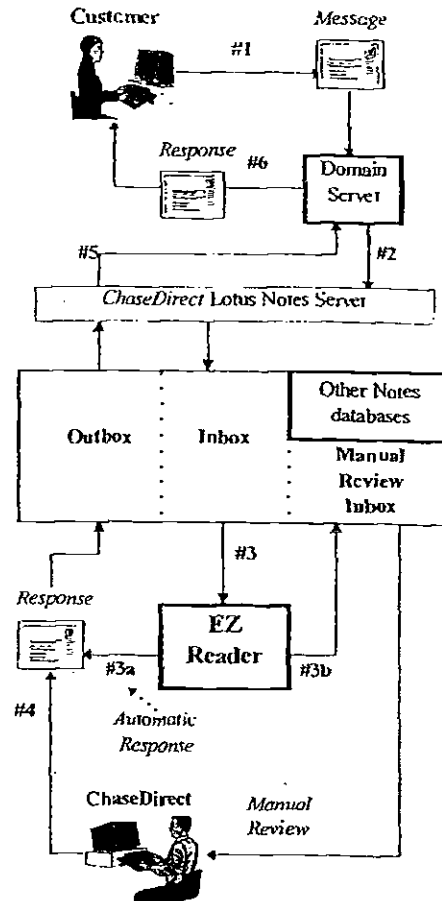6. The customer receives ChaseDirect's email reply.

Figure 2. Email Path through ChaseDirect

EZ Reader Hybrid Knowledge Base Approach

EZ Reader's hybrid reasoning approach reflects the actual interpretation process used by human email reviewers in ChaseDirect. The application combines pre-processing rules for parsing and case-based retrieval with a domain-specific knowledge-base. Other text interpretation applications have successfully used a hybrid approach (Sahin & Sawyer 1989) (Goodman 1991). A hybrid AI design provided both a functional and manageable programmatic representation of the business knowledge and rules for email interpretation.

The combined-rules and case-based approach was first evaluated after the team knowledge engineer observed and analyzed the ChaseDirect email interpretation process. Human email reviewers read each message from beginning to end while continuously evolving a final interpretation. The email reviewer recursively applied business knowledge to message content throughout the review. Reviewers modified their conclusions throughout the review, since an email contained any number of concepts in unpredictable order.

The application emulates the recursive nature of evolving interpretation by first detecting combinations of prominent words and patterns of text in any order throughout an incoming message, then setting object attribute values that both trigger and influence the case-based reasoning process. The application's case-based reasoning process then provides data to the rule-base to infer a classification by comparing the message content against the repository of messages in the case-base.

Internal Processing Flow

Within EZ Reader, program flow is controlled through the firing of declarative rules which trigger, monitor and control processing in the application programming interface (API), rule-base and case-base. Figure 3 depicts the knowledge-base processing flow, described in detail below.

Lotus Notes is Chase Manhattan Bank's corporate email standard; Chase's corporate email router routes emails to and from the Internet domain and Lotus Notes databases. Accordingly, EZ Reader was built to operate continuously and automatically in conjunction with Lotus Notes mail functions. The standard replies available for selection by EZ Reader are stored in a separate Lotus Notes database.

EZ Reader input and output is performed automatically through its connection to Lotus Notes, which was programmed via a Windows™ 3.1 API using the Vendor Independent Messaging (VIM) protocol and the API provided by Lotus Notes.

The EZ Reader API performs three important tasks:

1. It retrieves an email from the Lotus Notes inbox and returns it to EZ Reader.

2. It writes EZ Reader's output to either the manual review inbox (referral) or the outbox (auto-reply).

3. It marks the current email in the Lotus Notes inbox as read.

The API enables EZ Reader to send its output (the original message and EZ Reader's chosen response) to targeted Lotus Notes databases that can be viewed and edited by business users through a customized Lotus Notes interface that lists outputs by category. ChaseDirect business users have access to the Lotus Notes databases and all incoming

emails within the Lotus Notes system, so EZ Reader does not interfere with Lotus Notes as the standard Chase platform for email-related word processing, archiving, reference, and reporting functions.
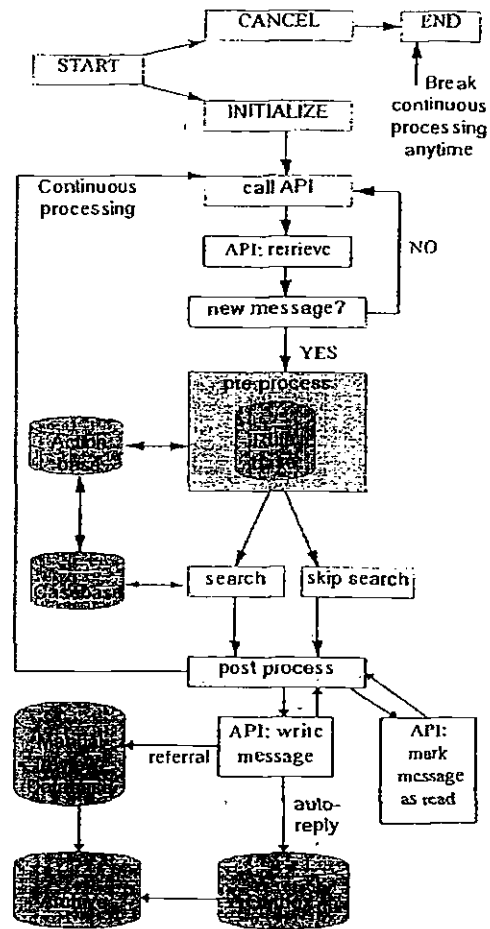

Figure 3. EZ Reader Internal Processing Flow

The API also contains functions to ensure transaction integrity in case of connectivity problems. Ordinary Lotus Notes processing tags a message as "read" when an unread mail message is opened for the first time and then closed. Typically, an API program that retrieves mail will mimic this action by marking a message as *read* immediately

after retrieval. However, in EZ Reader, message marking is deferred until after the response has been sent to ensure that no data will be lost. For example, if the EZ Reader client machine loses its connection to the Lotus Notes server in the middle of processing an email, upon application restart. EZ Reader will again retrieve that email and attempt to complete processing.

AI Enables Email Classification

EZ Reader uses classification rules and case-based reasoning to assign a business category and priority to each incoming email. EZ Reader then uses the inferred classification to select and attach a standard response from the Lotus Notes database of standard responses.

Categories. Using AI techniques described in the following sections. EZ Reader classifies each incoming email based on total message content into any of the following three categories:

Automatic Response. EZ Reader assigns a category of *Automatic Response* to items that can be associated with a response from the Lotus Notes repository of standard responses and directly mailed back to the sender without manual review or revision.

Referral. EZ Reader assigns a category of *Referral* to messages that cannot be processed solely as automatic responses. EZ Render then assigns a further sub-classification to the message to assist ChaseDirect staff with interpretation later. The sub-classifications reflect ChaseDirect's organization and operations, and are expected to change over time. Currently, there are two sub-categories for referred emails: Sales and Service, and 4 levels of priority.

Detected. EZ Reader assigns a category of *Detected* to emails that contain phrases or patterns that imply a particular manual handling procedure or interpretative aid for a referred message. In these cases. EZ Reader selects the appropriate remark to attach to incoming mail for manual review. Examples of Detected remarks include: "detected a phone number". "detected a foreign address".

Priority	Sub Category: Service	Sub Category: Sales
1	Fraud / lost cards	Promotional content (e.g., Microsoft Network)
2	Sensitive info (e.g., account number included)	Send Sign-Up Kit
3	Miscellaneous service	Multiple questions or lengthy messages
4	Comments, FYI	[none]

Table 1. Referrals - Categories and Priorities

Rule-base. EZ Reader uses rules to represent ChaseDirect business knowledge about how email content should be interpreted and handled. EZ Reader rules also control application processing flow.

EZ Reader's rules observe standard ART*Enterprise syntax. These rules are represented in an IF-THEN format with a left hand side containing a set of conditions and a right hand side containing conclusions. The rules *fire* whenever the conditions set forth in the left hand side of the rule are met regardless of the sequencing of conditions. The general syntax of a rule follows.

```
RULE RULE-NAME;
    condition₁
    condition₂
    . . .
=>    :: actions performed if all conditions are true
    action₁
    action₂
    . . .
```

An simple example of a rule for detecting foreign phone numbers is shown below. In this example, if the typical format of a foreign phone number is found in the message body (by calling a function called masked-member) or if other specified keywords are present, then all actions in the right-hand side of the rule will occur, including the printing of the text "Foreign Phone Number detected."

```
RULE foreign-phone
    (or
        (masked-member$ " +99 " ?message-body)
        (masked-member$ " +99-" ?message-body)
        (masked-member$ "(+99)" ?message-body)
        etc.
    )
    any other conditions...
=>
    (printout "Foreign phone number detected.")
    any other actions...
```

The left hand sides of the business knowledge rules in EZ Reader represent key linguistic clues that directly imply interpretive conclusions, including literals, wild card patterns, variables and segments, or choices of pattern sets. For example, one wild card pattern rule infers the presence of a foreign phone number by looking for patterns of text that resemble a phone number with a preceding plus sign. The inference of a foreign phone number is then used by the case-based search process to trigger an output classification.

Case-base. EZ Reader contains a case-base component that enables the application to emulate the reasoning of ChaseDirect staff when they use experience to determine how to handle an ambiguous email. When rule-base processing fails to clearly identify a classification for an ambiguous incoming email, EZ Reader attempts to find cases that closely resemble it. If a similar previous email is found, EZ Reader infers that the response used previously can be used (or adapted) for the incoming email.

Technically, the EZ Reader case-base is a searchable database of emails associated with specific actions and object instances stored within the EZ Reader application. It consists of an ART*Enterprise object model of an email, called a Case. The Case object class contains attributes, or slots, for the important features of emails as defined by the knowledge used by ChaseDirect to interpret and respond to the messages. Initial attributes of the Case object include references to addresses, specific types of computers, investment options, etc. The Case object also includes control attributes such as a title for the case instance. Another attribute of the Case object lists the results associated with the case, serving to link the case-base with the rule-base and also to direct Lotus Notes to retrieve specific standard responses. Each actual email sample in the case-base is defined as an object instance.

Within the ART*Enterprise development environment, an optimized case-base is prepared for searching by developer functions that create a case-base index, a highly optimized internal data structure that enables stored case and feature values to be matched very quickly with a an input case, called a *presented* case. EZ Reader searches the case-base assigning relative scores to each stored case based on the number of features, the mismatch of feature values and the absence of features as compared with the presented case using customizable case-based reasoning components supplied in the ART*Enterprise tool.

Character matching with trigrams was chosen to drive case-base scoring in EZ Reader. A trigram is a 3-character sequence. For example, the word "CHASE" generates 7 consecutive trigrams: _ _ C; _ CH; CHA; HAS; ASE; SE_; E_ _. When character matching is used, the value of the character feature is broken up into consecutive trigrams, and the trigrams of a stored case are matched against the trigrams of the presented case. The degree of partial matching is based on the proportion of the trigrams in the presented case that match trigrams in the stored case. The trigram matching technique minimizes the importance of the order of the individual words in the incoming message.

Standard case-base scoring for the message text of an email (as for all text type features) is driven by

ART*Enterprise's default trigram character-matching algorithm:

$$\text{feature-weight}(f)_i = ((tx-tm)/tx)^f \text{mismatch}_i + (tm/tx)^f \text{match}_i$$

Where
- tm is the number of trigrams in common between the presented case and the stored case
- tx is the total number of trigrams in the presented case feature
- $^f \text{mismatch}_i$ is the mismatch weight of feature f for the i-th case
- $^f \text{match}_i$ is the match weight of feature f for the i-th case.

The standard algorithm works as follows: if the value in a feature of the stored email matches the value in the corresponding feature of the incoming email, the feature's match weight is add to the stored email's score. If the feature's value mismatches, the feature's mismatch weight, typically a negative value, is added to the score.

In EZ Reader, each attribute, or feature, used by the case-base was assigned a default match-weight and a customized mismatch-weight of zero. In EZ Reader, the mismatch-weight of zero leads to better differentiation of scores, because of the incidence of misspellings in incoming emails, combined with the well-bounded knowledge domain. The actual weight that any feature contributes is meaningful only within the context of a particular case and relative to the weights of other features. Since stored cases can contain different numbers of features, a presented case's raw score is normalized by dividing the raw score by the maximum possible match score for the case.

Mismatches are not entirely ignored by EZ Reader. Another factor in scoring cases in EZ Reader is that a global absence weight is assigned to selected stored cases throughout the case-base. The total contribution of the absence weight to a stored case's match score is calculated by multiplying the value by the number of features in the presented case which are not in the stored case. The total absence weight is then added into the raw match score for the case prior to normalization. The default absence weight is -1; EZ Reader utilizes an absence weight of 0 to reduce the impact of missing features.

The case-base process is dependent upon rules to derive its presented case feature values. In EZ Reader, rules fire before the case-based reasoning process to extract features or characteristics of the email that help distinguish the content of the message. Depending on the content of the

message, any of the case-base search features may be set in the pre-processing rule phase. Any features set will then affect the scoring calculations performed by ART*Enterprise's case-based reasoning engine. Drawing out salient characteristics of the message content using rules combined with inexact case-based retrieval allows for more powerful and precise email interpretation than simple keyword parsing or case-based retrieval based on message body only.

For example, if EZ Reader infers from incoming email text that the sender does not want to be telephoned by ChaseDirect, the rule for do-not-call-customer? fires and sets that attribute in the case to "Yes". Features set to "Yes" then contribute to the case-based search by adding weight for similar stored cases during case-base retrieval.

A sample of EZ Reader hybrid processing flow, including the interaction between rule firings and case-base matching, is set forth below. The importance of set attributes for the case-base search is clearly illustrated in these two examples.

Two sample instances of the object class:case are shown below:

```
CASE001:
    title = "Sign-Up Kit request; Refer."
    subject = "chase direct"
    message text = "Please send me a
                    ChaseDirect sign-up kit.
                    My address is"
    address? = "Yes"
    action = refer:sign-up-kit.
      detected:address.
      auto:sign-up-ack
```

```
CASE002:
    title = "Sign-Up Kit request/no address;
    Auto Respond."
    subject = "chase direct"
    message text = "Please send me a
                    ChaseDirect sign-up kit."
    action = chase-direct-std
```

Suppose ChaseDirect receives an email with the body of the message as follows:

Dear ChaseDirect,
Please send the ChaseDirect Sign-Up Kit
to my home address.
    Thanks,
    John Doe
    123 Elm St.
    NY, NY 10001

A rule for detecting an address will fire, resulting in setting the case attribute address? to "Yes." Next, EZ Reader will perform a search against the case-base, ranking CASE001 with a score higher than CASE002 because of the match on address?. The email will be referred because the sign-up kit must be sent out via postal mail, and the sender will receive an electronic acknowledgment that their request has been received and that it is being processed. The detected action simply aids the ChaseDirect staff in quickly determining important contents in the email. Next suppose another person requests the kit but does not include his postal address in the email, in which case the request cannot be fulfilled. The case-base search will result in CASE002 scoring higher and being selected over CASE001. The sender will then receive an automatic standard ChaseDirect response with instructions on how to receive the sign-up kit.

One of the main benefits of case-based retrieval is that the cases retrieved from the case-base do not have to match the criteria exactly (as in the message text attribute), but the desired precision of a match can be easily specified. This quality is important to the success of email interpretation. Because EZ Reader processes free-format text, it cannot simply rely on an exact match between the incoming message body and the message text attribute. As a consequence, some superfluous literals, such as the actual address of the message sender, were removed from message text attribute values during the case creation process. Since the case-based retrieval algorithm performs trigram matching on the message text feature, literals such as addresses can unintentionally affect a case-base match score.

In addition to the attribute-setting rules described above, EZ Reader's rule-base consists of several "action-setting" rules. The rules can detect information that a human reader may overlook. Some aspects of the customer's email reveal valuable information for ChaseDirect but do not necessarily contribute to the reply. For example, ChaseDirect keeps track of prospects' and customers' phone numbers. This information is important to ChaseDirect, but ChaseDirect does not necessarily want to respond to the customer in a different manner. EZ Reader can tag this email with a message about detecting a phone number, which will consequently be easily seen by staff members when manually reviewing the email.

The case-base currently contains over 300 cases; the introduction of more sample emails over time will enable EZ Reader to interpret a wider variety of messages, and increase precision through further case feature refinements.

Exception Message Handling. There are some messages that EZ Reader is not able to interpret, e.g., exception messages. The content of a exception message is ambiguous even for a knowledgeable person to interpret, for example, a message that contains only the word "Test". The EZ Reader application contains ease examples of previously received exception messages, so it is able to forward them and all other uninterpretable messages to the Lotus Notes database for manual evaluation.

Error Processing. EZ Reader checks for errors both in its knowledge-base processing and in the API connection to Lotus Notes. Errors generated by the Lotus Notes API do not cause EZ Reader to terminate; instead, EZ Reader will continue trying to access Lotus Notes until a connection is made. For instance, if the Lotus Notes server is temporarily down, the API will send the relevant error message back to EZ Reader. EZ Reader then waits a few seconds before trying the connection again. In this manner, EZ Reader is self-monitoring and maintains maximum up-time.

Application Benefits

EZ Reader played a critical role in establishing ChaseDirect's ability to provide and maintain a responsive online marketing and service channel. The implementation of automated reasoning enabled process simplification, speed and consistency of responses, as described below.

EZ Reader increased the speed of response to the customer. EZ Reader eliminated manual intervention for a percentage of messages[3] and more than halved the time to process messages requiring manual intervention. The reduction in manual intervention allowed ChaseDirect to turn around email responses faster to customers. With a knowledge base processing speed of 1 message per second plus 2-5 seconds for Lotus Notes communications, EZ Reader reduced overall processing time by 6-8 minutes per message.

Another factor contributing to faster customer response was that the application was available on a 24-hour basis, allowing continuous processing of automatic response types of messages over a weekend when many customers tended to be active online.

---

[2] around 5 percent of all messages

[3] see *Automatic Throughput Percentage* for a discussion of the effect of message content on average speed of response

EZ Reader enforced consistency of responses to customers. Before deployment of EZ Reader, ChaseDirect workers alternated the daily responsibility of reviewing emails. Each day, a different worker typically spent the whole work day manually reading and responding to email. Consistency of response was an important business consideration for ChaseDirect, and since quality of response was a function of knowledge and experience, responses frequently had to be checked by a supervisor. EZ Reader assured consistency of response because it automatically assigned prepared text depending on its singular interpretation of a message.

EZ Reader simplified the business process substantially. EZ Reader enabled ChaseDirect to reduce the number of manual steps and the effort needed to process its incoming email. A comparison of processing steps eliminated and modified by EZ Reader is outlined in Table 2 below.

Task	Before EZ Reader	After EZ Reader
Dial in to email system	one min. per access	eliminated
Read and analyze	one minute or more	eliminated
Print and annotate	up to one minute	eliminated
Select/Adjust response	five minutes or more	eliminated/ modified
Send email	up to one minute	modified
Archive original	up to one minute	modified
Delete from folder	up to one minute	modified
Enter into CMS[4]	avg: five minutes	no change

Table 2. Process Changes Enabled by EZ Reader

Application Maintenance

Currently, knowledge-base maintenance requires editing of cases, rules, and actions, all of which are in ART*Enterprise syntax. One proposed technical enhancement to EZ Reader is to build a framework for maintenance in which business users could add and modify case objects, action objects, and even rules, through a GUI interface where the underlying ART*Enterprise syntax of the objects is invisible to the maintainer.

Although specialized knowledge and skill sets are required to maintain EZ Reader today, the application is highly modular and object-oriented. The case-base and rule-base are independently structured. In addition, the

---

[4] a contact management system used to track contacts with customers and prospects

design is general enough to be easily adapted to other domains, and code-level maintenance procedures are straightforward.

EZ Reader maintenance is required whenever certain business or technical environment changes occur. The following types of business changes usually require EZ Reader maintenance:

- A new or revised association between a type of message and its prepared response is required. For instance, if emails concerning a former marketing promotion need to be answered differently than they were originally, EZ Reader maintenance is performed. (Modifications to prepared response wording are performed through a Lotus Notes edit view.)

- A new type of message needs to be associated with a prepared response. For instance, if emails concerning a new marketing promotion need to be processed automatically, EZ Reader maintenance is performed.

- A new informational message needs to be generated by EZ Reader when it detects a particular type of incoming message, regardless of the response. For instance, if emails from current customers need to be flagged, EZ Reader maintenance is performed.

In addition to these knowledge-base maintenance scenarios, whenever a change in the technical environment around EZ Reader is planned, the application is assessed for required modifications and retested in the target environment. For example, because EZ Reader interfaces with Lotus Notes, if a new version of Lotus Notes is planned for installation, the EZ Reader / Lotus Notes API is tested and assessed for any relevant maintenance. In this case, because the connection to Lotus Notes is independent of the knowledge-base, only the API portion of EZ Reader requires retesting.

One person from the resident KB team at Chase is responsible for maintaining EZ Reader, among other duties. The team is trained in ART*Enterprise as well as C language and Lotus Notes development. They also understand the business requirements of EZ Reader and are able to translate change requests into EZ Reader knowledge-base modifications.

No manual intervention is necessary for the day-to-day operation of EZ Reader. A local replica of the Lotus Notes databases resides on the development machine; these databases are used for testing and maintenance of the system. In addition, EZ Reader can be run in strictly manual mode in which it bypasses the API connection. Input data in this mode is retrieved from an external text file and allows for quick testing of the knowledge-base independent of the API.

Chase Manhattan Bank, Regional Banking, launched an initiative in 1995 to explore how artificial intelligence (AI) could help meet its business challenges. At that time, 12 applications were identified to leverage AI at Chase and EZ Reader is one of those implementations.

EZ Reader started with a concept paper written by members of Chase's Knowledge Base (KB) technology team, including Brightware consultants, in May 1995. Further development of functional specifications, cost analysis and presentations to management served to get approval for prototype development, which began in July 1995 and was finished in August. Development of a production system was granted approval in September 1995.

Beginning September 1995, EZ Reader production application development proceeded with one full-time technical developer and several part-time staff including four business analysts, three testing specialists, and numerous technical support personnel who specialized in Lotus Notes and the Chase email network.

The knowledge-base, consisting of the rule-base and case-base, were completed first, enabling user testing while the Lotus Notes API was being developed. Initial knowledge-base testing was performed in November, resulting in accuracy very close to our target level with throughput of one message per second. Further refinements increased the accuracy to our goal in December. Although the knowledge-base could have been deployed at this point in a semi-manual mode, management decided for a single integrated implementation after completion of the Lotus Notes API. The Lotus Notes API was completed, consolidated and tested with the knowledge-base a few weeks later.

EZ Reader was developed using ART*Enterprise version 2.0h with the included Microsoft Win32s library, Visual C/C++ 1.5 and Lotus Notes version 3.0C all running on Windows™ 3.1. The application runs on a PC with an Intel-based 486/66MHz processor, 500MB of hard disk space and 32MB of RAM. The Lotus Notes server is an IBM 9595 running OS/2 version 2.1.1, with an Intel-based Pentium 122MHz processor, 2 hard disk drives of 500MB and 1GB, and 64MB of RAM.

Measuring EZ Reader Performance

EZ Reader was evaluated before implementation to ensure that ChaseDirect production approval criteria were met. To evaluate EZ Reader's performance, the team analyzed three measures: speed, accuracy and automatic throughput percentage. EZ Reader's performance on each measure is described below.

Speed. One essential success criterion for EZ Reader was that it reduce the total amount of time spent processing emails. The earliest tests of EZ Reader knowledge base demonstrated speed of one second or less for interpretation functions that manually took one minute or more. After Lotus Notes integration, an additional 3-5 seconds was required for each message. EZ Reader's Lotus Notes integration reduced the amount of time needed to attach and send responses where manual intervention was still required.

Accuracy. ChaseDirect was concerned that any software would be able to interpret messages with the required level of accuracy. The estimated accuracy level for manual processing was 98%. The team set EZ Reader's accuracy goal at a rate equal to that of 95% of the accuracy of manual processing.

The case-based reasoning logic in the EZ Reader knowledge-base was able to deliver a high level of accuracy. Its capability to rank the degree of similarity between incoming messages and previous messages in its case-base, combined with rules, results in a high level of accuracy.

Automatic Throughput Percentage. *Automatic throughput percentage* was a measure established by the team to evaluate the equilibrium between incoming message content and EZ Reader's knowledge base. It refers to the percentage of messages that can be processed without manual intervention. As more cases and rules are added to EZ Reader, the team expects to achieve an automatic throughput percentage of 80% or more.

During the early months of initial testing, the automatic throughput percentage varied from 20% to 80%. It fell when high-volumes of messages with new, unexpected content were received. It also dropped when business requirements were implemented to refer additional types of messages. Based on a comparison of manual processing time with EZ Reader processing time, the automatic throughput percentage translates to significant productivity gains for each percentage point gained in automatic throughput.

Measures within Lotus Notes calculate and track the volume of messages by category (automatic or referral), date and sender email address. When a decreasing percentage of automatic throughput is detected, it alerts the business to the need to add new rules or examples to EZ Reader to enable it to recognize new types of messages sent by customers and prospects.

Knowledge Base Testing Method

To test EZ Reader's knowledge base, the team performed parallel testing to compare EZ Reader with manual processing, using the same messages both manually and

through EZ Reader. The results from each were compared to assess the readiness of the EZ Reader knowledge-base for production.

Business analysts collected electronic copies of the actual incoming messages received during a two-week period as test bed data for the parallel test. EZ Reader developers used actual messages received by ChaseDirect as input data for testing EZ Reader. The testers fed the test bed messages electronically into EZ Reader in its native ART*Enterprise* development mode and printed the results. The printouts contained the incoming messages, the EZ Reader output classifications and the amount of time it took EZ Reader to process the message. The printouts also contained a blank formatted area that was used to record evaluation remarks.

Two reviewers from ChaseDirect who were not involved in the original manual processing stage analyzed the printouts. Where EZ Reader produced an incorrect category or response, the reviewer noted the expected response, and the error was reviewed by the EZ Reader team. Refinements to the case-base and rules were made and verified in subsequent abbreviated tests before the application was approved for production.

The Lotus Notes API and end-to-end network communications were thoroughly tested over a period of weeks using conventional systems testing techniques.

Summary

EZ Reader is an AI application that provides many tangible and intangible benefits to Chase Manhattan Bank as it seeks to maximize opportunities in the Internet market.

EZ Reader enabled ChaseDirect to eliminate the cost of overtime for email processing, helped meet customer expectations for service standards (such as response timeliness), provided for smoother implementation of ChaseDirect marketing programs and enabled unattended processing of email on weekends.

—Since EZ Reader's knowledge base was developed for ChaseDirect, its initial utility was limited to email processing for that department. The following enhancements for EZ Reader are currently being considered:

- Add business knowledge for other business areas who choose to take advantage of World Wide Web communication with customers.

- Enable EZ Reader output to be addressed automatically to recipients throughout Chase via Lotus Notes.

- Automate the business knowledge maintenance functions of the application, i.e., enable ChaseDirect

business users to interact with EZ Reader to change how its knowledge base interprets messages and links them to responses.

- Provide the ability to automatically process incoming messages in Spanish or other languages, providing a potential global marketing advantage.

- Enable EZ Reader use of historical email and profile data to personalize EZ Reader processing.

- Link the contact management system to a process that adds customer knowledge from emails processed by EZ Reader.

Acknowledgments

References

Compton, M. and Wolfe, S. 1993. Intelligent Validation and Routing of Electronic Forms in a Distributed Work Flow Environment. Technical Report, FIA-93-31, NASA Ames Research Center, Artificial Intelligence Research Branch.

Goodman, M. Prism. 1991. A Case-Based Telex Classifier. In Innovative Applications of Artificial Intelligence, Vol. 2: AAAI Press.

Sahin, K., and Sawyer, K. 1989. The Intelligent Bauking System: Natural Language Processing for Financial Communications. In Innovative Applications of Artificial Intelligence: AAAI Press.

Taylor, D.; Mehta, B.; and Wurster, T. 1996. Online Delivery & the Information Superhighway; Searching for Retail Strategies. Bank Management 72(1): 22-29.

BR 001262