

A Neural Network-Based Agent Framework for Mail Server Management

Charles C. Willow, Monmouth University, USA

ABSTRACT

Amidst the era of e-economy, one of the difficulties from the standpoint of the information systems manager is, among others, the forecast of memory needs for the organization. In particular, the manager is often confronted with maintaining a certain threshold amount of memory for a prolonged period of time. However, this constraint requires more than technical and managerial resolutions, encompassing knowledge management for the group, eliciting tacit knowledge from the end users, and pattern and time series analyses of utilization for various applications. This paper proposes a framework for building an automated intelligent agent for memory management under the client-server architecture. The emphasis is on collecting the needs of the organization and acquiring the application usage patterns for each client involved in real time. Due to the dynamic nature of the tasks, incorporation of a neural network architecture with tacit knowledge base is suggested. Considerations for future work associated with technical matters comprising platform independence, portability, and modularity are discussed.

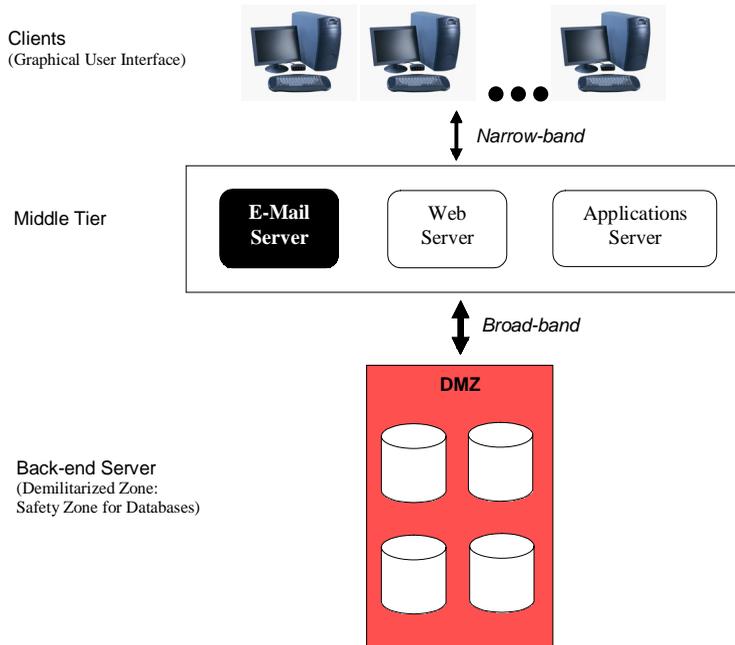
Keywords: automata; automatic intelligent agent; computer-supported collaboration work; human-computer interaction; information resource management; knowledge base; knowledge management; memory management; neural networks; tacit knowledge

INTRODUCTION

Integrated information systems for distributed organizations comprising the information technology (IT) infrastructure

and generic business applications, such as enterprise systems (ES), supply chain management (SCM), customer relationship management (CRM), and knowledge management system (KMS), are by far

Figure 1. General information systems architecture



one of the vital assets to sustain the competitive edge in e-economy. At the same time, however, administrators of these information systems often are confounded with a vast array of management problems, which, in large, may be classified into the following:

- Selection and upgrades of hardware platform, middleware, and applications combination;
- Information resource management;
- Integration of applications; and
- Security management.

This paper discusses problems associated with information resource management (IRM) and suggests a development framework for reconfiguring the server as well as clients for moderately large-scale information systems.

Among others, one of the difficulties concerning IRM from the standpoint of the information systems manager is the correct forecast of overall memory needs for the organization. In particular, the manager often is confronted with maintaining a certain threshold amount of memory for a prolonged period of time. One may argue that the cost of memory is declining rapidly, and its management may not be a factor affecting IRM. Contrary to this common misbelief, however, a number of authors suggest that there should be a certain threshold for memory management (Appen, 2002; Kanawati & Malek, 2002; Kankanhalli et al., 2003; Lansdale, 1988; Mathe & Chen, 1998; Pinelle & Gutwin, 2002; Pinelle et al., 2003; Roos et al., 2003) within a prescribed time window, analogous to budgetary considerations. In essence, memory management affects

overall performance for both client-server and peer-to-peer architectures of the information system.

Under multi-tiered server configuration, which seems to be the norm for most organizations, access to the very back-end server(s) requires high bandwidth networks (Willow, 2005a). As a consequence, the impact of effective memory management is amplified further. Figure 1 represents a general three-tier architecture of information system.

The focus of this paper is on building a conceptual framework for developing intelligent memory management system for the mail server. To tackle the IRM problem more comprehensively, a modular approach is employed. As e-mails become one of the preferred methods of communications for most organizations, mail server management indeed requires attention with top priority.

The major contribution of this paper lies in the development of a framework for building multiple agents for the mail server, with an emphasis on memory management. Agent-based autonomous systems (i.e., automata) in recent years have been adopted as one of the better methods for managing virtual organizations in various applications (Flenner et al., 2002; Kanawati & Malek, 2002; Mathe & Chen, 1998; Murch, 2004; Taylor, 2004). They range from consumer applications, such as online travel arrangements, to system diagnostics, including online data quality audits and remote troubleshooting.

The organization of this paper follows. In the second section, knowledge management for eliciting, building, and managing end-user preference and e-mail

usage patterns is discussed. The third section follows to illustrate the core system—multiple agents. Suggestions for (hands-on) construction of the proposed framework are made in the fourth section, followed by a section devoted to conclusions.

KNOWLEDGE MANAGEMENT

The key to maintaining accuracy of the proposed multi-agent system lies in managing highly subjective knowledge for sharing and customizing or personalizing end-user memory usage patterns across the organization. Information technology (IT) may support knowledge management (KM) in two classes: codification and personalization (Kankanhalli et al., 2003). In essence, the codification approach manages structured knowledge, whereas personalization manages unstructured, tacit knowledge. Because e-mail usage patterns for end users may entail both types of knowledge, a separate knowledge base or repository is suggested for the framework of this research. That is, there may be, on the one hand, common patterns of e-mail management among end users, such as removing messages which are more than 36 months old, organizing e-mail folders every 30 days, and so forth. On the other hand, each user may have highly subjective patterns that may not be consistent with those codified knowledge. Lansdale (1988) emphasized in his early research the need for Cognitive Interface Tools (CITs) to collect, organize, build, and share both types of knowledge for the office system. However, not many literatures to date have been dedicated to solving this problem.

Table 1. Attributes of knowledge management for e-mail usage

Attributes	Type of Knowledge
Time Stamp	Structured
Size (KB)	Structured
Recipient/Sender	Tacit/Unstructured
Event/Subject	Tacit/Unstructured
Attachment	Tacit/Unstructured
Significance	Tacit/Unstructured

Markus (2001) describes the general stages of the KM process: knowledge elicitation, knowledge packaging, distributing or disseminating knowledge, and reusing knowledge. To this end, building knowledge bases in conjunction with automatic agents is considered one of the better methods for managing user knowledge associated with e-mail usage in real time.

Attributes for Knowledge Management

This section illustrates the necessary set of attributes to be incorporated into KM. Note that the values of these attributes will be collected in real time from end users for generating patterns of e-mail usage.

As noted in Lansdale (1988), the process of information retrieval in the human mind is fundamentally different from a filing or library system, in which items are accessed by location rather than by their meaning. The first notion is that people recall chronological information about information (i.e., what else was happening at roughly the same time). Consequently,

time stamp of e-mails may be a good source of structured information or knowledge. Association is another means by which humans retrieve information. Each e-mail message is associated with four pieces of tacit information: recipient or sender, event or subject, attachment(s), and significance of the message. Table 1 summarizes the attributes for KM concerning e-mail usage.

A set of six generic attributes associated with e-mails, as described in Table 1, is to be employed in the suggested framework of this paper. Notice that the first two attributes — time stamp and size — are structured information, which may be available for both the server and clients. By contrast, each client may manage his or her e-mail messages, based on one or more of the four tacit attributes: recipient/sender, event/subject, attachment(s), and significance. Given a certain restriction of memory size, say 100MB per e-mail account holder, one client may choose either to remove or to archive (on local memory store) e-mails based on recipient/sender, event/subject, attached file(s), significance, or any com-

bination of the four, so far as tacit knowledge management is concerned. Alternatively, he or she simply may choose to archive or to remove e-mails with regard to structured information, such as time stamp and/or size. It is precisely this knowledge associated with each e-mail client that is expected to be elicited by the automated multiple agents proposed in this paper, preferably in real time.

INTELLIGENT AUTOMATED AGENT

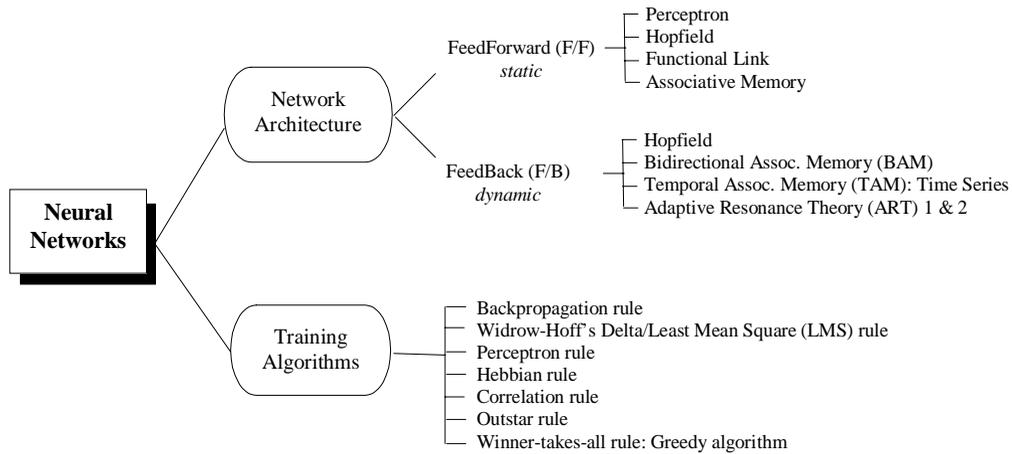
Under a certain memory constraint for each e-mail client, the administrator of the information system (i.e., mail server) may choose to adopt a brute-force approach, based on structured information, such as time stamp or size of the message. As a consequence, clients (without their consent) often may realize that their e-mails are unavailable at times, once they have reached the memory quota set by the system. However, this aggressive method is not effective, due to its user service and, perhaps, legal implications. Thus, an automated system that may advise clients in real time about their e-mail usage patterns is considered an attractive alternative for information resource management. Once the client is logged on to the system, the automatic intelligent agent generates a list of e-mail messages that are to be removed as well as those that are highly likely to be candidates for local archives. In addition, another agent system is suggested in order for the server to advise the administrator(s) of potential preventative measures. In essence, a conceptual framework for a multi-agent system is proposed in this paper. Similar ideas are being incorporated at

present into the Web and applications servers in Willow (2005a, 2005b).

Neural networks (NN) are employed as the inference engine for the proposed multi-agent system. An NN typically processes large-scale problems in terms of dimensionality, amount of data handled, and the volume of simulation or neural hardware processing (Willow, 2002). It emerged as an area of artificial intelligence (AI) to mimic human neurons in both perception and learning. It is interesting to note, however, that a conceivably disparate area within information science classified as knowledge representation brought the attention of researchers to pursue classes of computing and processing, such as neural networks. An object-oriented paradigm emerged as one of the better models for knowledge representation. In fact, the motivation for NN research was to seek an improved methodology in machine learning and, more specifically, in the area of planning algorithm, thereby augmenting the techniques available at the time. However, as the research progressed, more obstacles to emulating human neurons were realized. Toward this end, the jargon NN at present would be more appropriate if it were replaced with parallel, distributed simulation. Figure 2 illustrates taxonomic views of NN (Willow, 2002). Notice that it is not comprised of an exhaustive list of available NN models to date.

The concept of feedback plays a central role in learning for NN. As illustrated in Figure 3, two different types of learning are to be distinguished: learning with supervision (i.e., training) vs. learning without supervision (Willow, 2002).

Figure 2. Classification of neural network models



In supervised learning (see Figure 3(a)), the desired response \mathbf{d} of the system is provided by the teacher at each instant of time. The distance $\rho[\mathbf{d}, \mathbf{o}]$ between the actual and the desired response serves as an error measure and is used to correct network parameters externally. Since adjustable weights are assumed, the teacher or supervisor may implement a reward-or-punishment scheme to adapt the network's weight matrix, \mathbf{W} . This mode of learning is pervasive and is used in many situations of natural learning. A set of input and output patterns, called a training set, is required for this learning mode. Often, the inputs, outputs, and computed gradient are deterministic; however, the minimization of error proceeds over all its random realizations. As a result, most supervised learning algorithms reduce to stochastic minimization of error in multi-dimensional weight space.

In learning without supervision (Figure 3(b)), the desired response (\mathbf{d}) is not known; thus, explicit error information cannot be used to improve network be-

havior. Since no information is available as to correctness or incorrectness of response, learning somehow must be accomplished, based on observations of responses to inputs of marginal or, at times, no knowledge. Unsupervised learning algorithms use patterns that typically are redundant raw data having no labels regarding their class membership or association. In this mode of learning, the network must discover for itself any possible existing patterns, regularities, separating properties, and so forth. While discovering these, the network undergoes a change of its parameters, which is called *self-organization*. Adaptive Resonance Theory (ART) is a good example of such a class.

Adaptive Resonance Theory

Adaptive Resonance Theory (ART), as illustrated in Zurada (1992), is a unique unsupervised class of neural network algorithm. It has the novel property of controlled discovery of clusters. Further, the ART network may accom-

Figure 3. Learning modes for neural networks: (a) supervised (left diagram) vs. (b) unsupervised (right diagram)

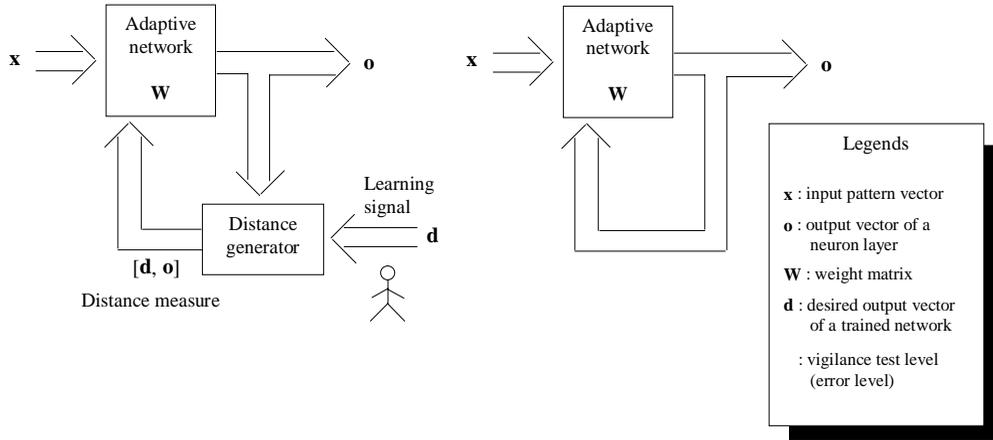
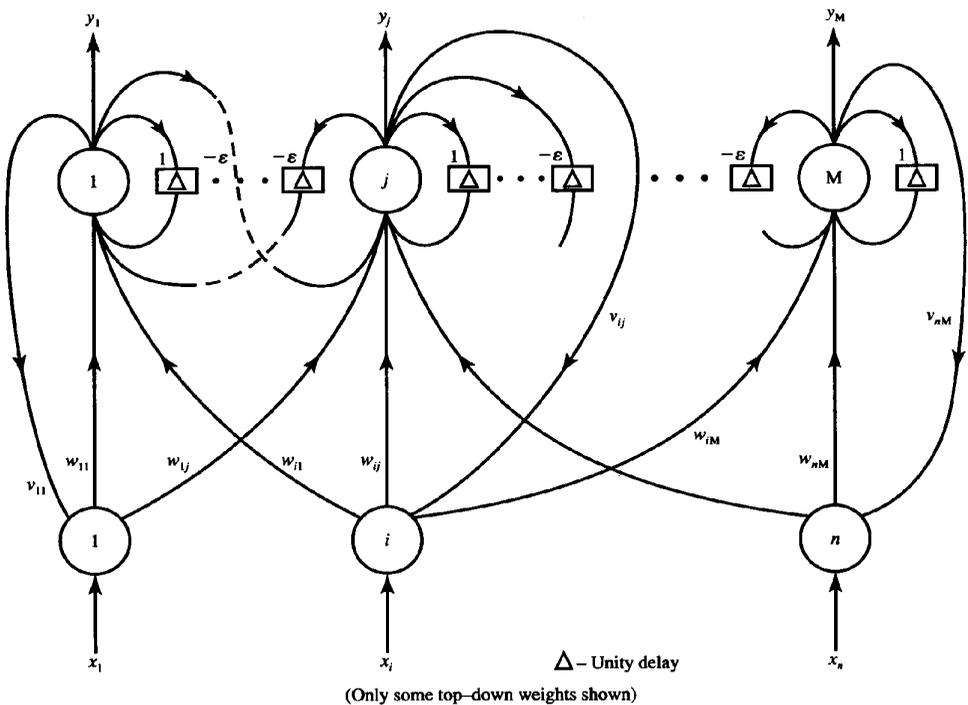


Figure 4. Neural network architecture for Adaptive Resonance Theory



moderate new clusters without affecting the storage or recall capabilities for clusters that were already learned, fit for the

scope of the problem of this paper. Figure 4 illustrates the ART architecture (Zurada, 1992).

Nomenclature of the model follows:

Subscripts and Superscripts

- i* Subscript for input variable, x ; $i = 1, \dots, n$.
j Subscript for output clusters, $j = 1, \dots, M$.
m Subscript for output neuron, y or neuron of hidden layer; $m = 1, \dots, j, \dots, M$.
k Superscript for neuron y at layer k ; $k \geq 0$.

Parameters

- M* Total number of clusters set by the decision maker.
n Total number of variables for input vector/tuple, $\mathbf{x} = [x_1, \dots, x_n] = \langle x_1, \dots, x_n \rangle$.

Variables

- \mathbf{x} Input vector; $\mathbf{x} = [x_1, \dots, x_n]$.
 \mathbf{w} Weight of the input vector; $\mathbf{w} = [w_1, \dots, w_n]$.
 \mathbf{y} Output vector; $\mathbf{y}^k = [y_1, \dots, y_M]$.
r Controlled vigilance factor indicating closeness of input to a stored cluster prototype to provide a desirable match; $0 < \rho < 1$. The ART net will seek a perfect match for $\rho = 1$ and loosely coupled matches for lower values of ρ .
 \mathbf{v} Weight vector for verifying cluster exemplar proximity; $\mathbf{v} = [v_1, \dots, v_n]$.
t Update index for weights, \mathbf{w} and \mathbf{v} .

Algorithm for ART is summarized as follows (Zurada, 1992):

Step 1: Initialization

The vigilance threshold, ρ , is set.

Weights are initialized for n -tuple input vectors and M top-layer neurons.

$(M \times n)$ matrices \mathbf{W} and \mathbf{V} each are initialized with identical

$$\mathbf{W} = \left[\frac{1}{1+n} \right] \quad (1)$$

$$\mathbf{V} = [1] \quad (2)$$

$$0 < \rho < 1 \quad (3)$$

Step 2: Input Neuron Processing

Binary unipolar input vector \mathbf{x} is presented at input nodes, $x_i = 0, 1$ for $i = 1, 2, \dots, n$.

Step 3: Matching Score Computation

All matching scores are computed as follows:

$$y_m^o = \sum_{i=1}^n w_{im} x_i \text{ for } m = 1, \dots, M. \quad (4)$$

In this step, selection of the best matching existing cluster, j , is performed according to the maximum criterion, as follows:

$$y_j^o = \max_{j=1, \dots, M} (y_m^o) \quad (5)$$

Step 4: Resonance

The similarity test for the winning neuron j is performed as follows:

$$\frac{1}{\|\mathbf{x}\|} \sum_{i=1}^n v_{ij} x_i > \rho \quad (6)$$

where the norm is defined as:

$$\|x\| \equiv \sum_{i=1}^n |x_i| \quad (7)$$

If the test as illustrated in equation (6) is passed, the control is passed on to Step 5. Upon failing the test, Step 6 is followed only if the top layer has more than a single active node left. Otherwise, Step 5 is followed.

Step 5: Vigilance Test

Entries of the weight matrices are updated for index j passing the test of Step 4. The updates are only for entries (i, j) , where $i = 1, 2, \dots, M$, and are computed as follows:

$$w_{ij}(t+1) = \frac{v_{ij}(t)x_i}{0.5 + \sum_{i=1}^n v_{ij}(t)x_i} \quad (8)$$

$$v_{ij}(t+1) = x_i v_{ij}(t) \quad (9)$$

This updates the weights of the j -th cluster, newly generated or existing. The algorithm returns to Step 2.

Step 6: Cluster Generation

The node j is deactivated by setting y_j to 0. Thus, this node does not participate in the current cluster search. The algorithm goes back to Step 3 and will attempt to establish a new cluster different from j for the pattern under test.

Clusters are generated by the network itself, if such clusters are identified in input data, and store the clustering information about patterns or features in the absence of *a priori* information about the

possible number and type of clusters. In essence, ART computes the input-pattern-to-cluster matching score (y), which represents the degree of similarity of the present input to the previously encoded clusters. The vigilance threshold, ρ , where $0 < \rho < 1$, determines the degree of required similarity, or match, between a cluster or pattern already stored in the ART network and the current input in order for this new pattern to resonate with the encoded one. If no match is found, then a new class or cluster is created.

Applications of ART to the proposed multi-agent system follow in the next two subsections.

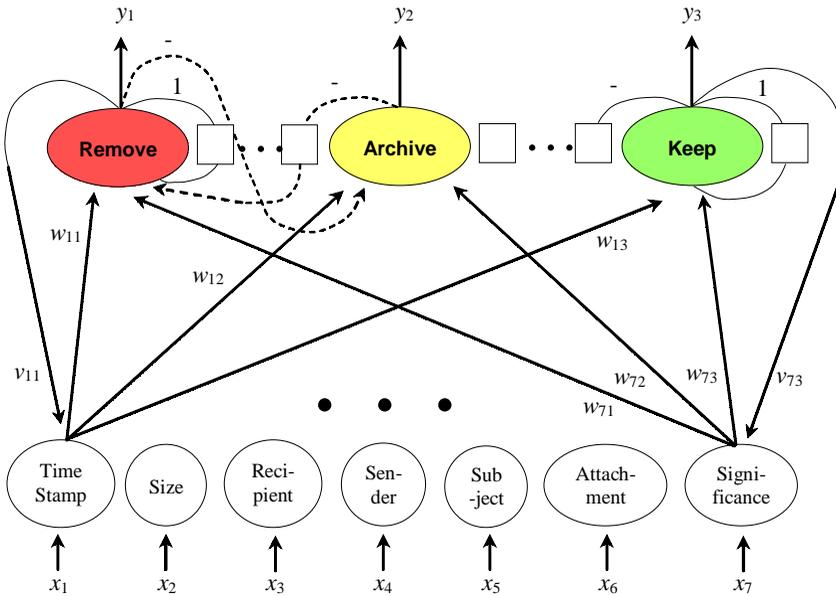
Client Agent Architecture with ART

This section describes the ART architecture for the suggested client agent. The object of the client agent is to provide real-time knowledge regarding e-mail management for each client end user. Figure 5 follows to illustrate.

For each e-mail message, an input vector comprised of the following seven-tuple attribute is produced; $\mathbf{x} = \langle x_1, \dots, x_7 \rangle$:

- x_1 Age of the message. It is automatically computed as system-generated time (TNOW) – (Time_Stamp).
- x_2 Size of the e-mail, generally measured in kilobytes (KB).
- x_3 Recipient information in SMTP address format (To: johndoe@xyz.com).
- x_4 Sender information in SMTP address format (From: janedoe@xyz.com). Note that x_3 and x_4 are mutually exclusive, and may have null values associated. That is, each message is either

Figure 5. ART for the client agent



received from or strictly sent to an SMTP address.

x_5 Subject of the message (character strings).

x_6 Attachment to the message. A unique four-digit code encompassing the number of attachments between 0 and 99 (first two digits) and their file types is assigned for x_6 . To simplify the data structure, file types are restricted to the three most common on the Internet; ASCII.txt (1), Microsoft.doc (2), and Adobe.pdf (3). Examples of x_6 values are:

- 0000 No attachments.
- 0103 One attachment in .pdf format.
- 9923 Ninety-nine attachments in mixtures of .doc and .pdf files.

x_7 Significance of the e-mail message, set by the end user. It ranges from 1 to 5, 5 being the most significant and 1 being

the least. Note that this value is applicable exclusively for the body of the e-mail. For instance, a message with $x_7 = 1$ does not warrant automatic removal from the mailbox. Instead, the user may choose to archive it due to the importance of its attachment(s), x_6 , for example.

A simplified numerical example follows to illustrate. Consider the following three messages for a client:

- Message #1 = $\langle x_1, \dots, x_7 \rangle$
 = $\langle 60, 2000, jmis@xyz.edu, \text{null}, \text{"publication consideration"}, 0203, 5 \rangle$
- Message #2 = $\langle 02, 20, \text{null}, jdoe@usa.com, \text{"Greetings"}, 0000, 1 \rangle$
- Message #3 = $\langle 14, 250, family@home.org, \text{null}, \text{"get together"}, 0000, 5 \rangle$

Thus, each message forms an input pattern vector, \mathbf{x} . Tacit input values are converted into a utility scale of 1 to 5 for neural processing, based on interactions with the knowledge base. This pertains to the attributes, $x_3, x_4,$ and x_5 . As a consequence, the input vectors are:

$$\begin{aligned} \mathbf{x}_1 &= \langle 60, 2000, 4, \text{null}, 5, 02, 5 \rangle \\ \mathbf{x}_2 &= \langle 02, 20, \text{null}, 5, 1, 00, 1 \rangle \\ \mathbf{x}_3 &= \langle 14, 250, 5, \text{null}, 1, 00, 5 \rangle \end{aligned}$$

When \mathbf{x}_1 is presented, the steps of the ART algorithm are:

$$M = 3; n = 7$$

$$w_{ij} = \frac{1}{n+1} = 1/8 = 0.125;$$

$$v_{ij} = 1, \quad i = 1, \dots, 7, j = 1, 2, 3 \text{ for Remove, Archive, Keep.}$$

Given a standard vigilance value of $\rho = 0.5$, the left term in inequality (6) is of unity in the first pass, allowing the similarity test to be passed. This results in unconditional definition of the first cluster, the default being the "Archive." Equations (8) and (9) of Step 5 produce:

$$w_{32}(2) = \frac{1 \times 4}{0.5 + [4 + 0 + 5 + 2 + 5]} = 0.2424$$

$$w_{52}(2) = \frac{1 \times 5}{0.5 + [4 + 0 + 5 + 2 + 5]} =$$

$$w_{72}$$

for the tacit variable set only. Notice $x_3, x_5,$ and x_7 had significant values of 4 or above. The remaining weights $w_{i2} = 0.125$, as initialized in Step 1. In addition,

$$v_{32} = v_{52} = v_{72} = 1,$$

as initialized, while the remaining weights are recomputed as $v_{i2} = 0$.

For the second input pattern vector \mathbf{x}_2 , there are no significance values, and the similarity test of equation (6) yields

$$\frac{1}{\|\mathbf{x}\|} \sum_{i=1}^7 v_{ij} x_i > \rho = 0 < 0.5$$

Due to the failure of the vigilance test and the absence of other nodes for further evaluation and for potential disabling, pattern \mathbf{x}_2 is treated as another new cluster. Further, a null value for the left-hand side of equation (6) is classified as the "Remove" cluster.

In essence, the ART-based neural network processing is expected to advise the e-mail clients of possible action(s) for each (e-mail) message.

Server Agent Architecture with ART

The Adaptive Resonance Theory (ART) model also may be employed for another set of agent systems dedicated to assisting the e-mail server administrator(s). The two systems of agents, which target individual clients as well as administrators, then may communicate in real time by accessing the integrated knowledge base. However, the server agent architecture is relatively more complicated due to differences in network protocols. Two major methods employed are: **Internet Message Access Protocol (IMAP)** and **Post Office Protocol (POP)**. IMAP integrates messages on the shared mail server and permits client e-mail programs to access remote message stores as if they were local. Thus, the memory burden on the

Table 2. Classes of information availability based on e-mail application protocols

E-Mail Protocols Information Process	IMAP (central)	POP (distributed)
Server-based	Server only	Mixed
User-based	Mixed	Client only

server is far greater for IMAP than for POP. However, complete monitoring of client e-mails is possible under the IMAP scheme. That is, a single server-side agent system may suffice for the IMAP, whereas a single client-agent may fit systems with POP being implemented. See Table 2 for an illustration of this.

A multi-agent system, therefore, is expected to be highly useful for managing client knowledge under the IMAP (fourth quadrant in Table 2) and for managing server knowledge under the POP structure (second quadrant).

IMPLEMENTATION AGENDA

Architecture of the proposed multi-agent system, followed by description of its functions and the challenges encountered during systems analysis, are discussed in this section.

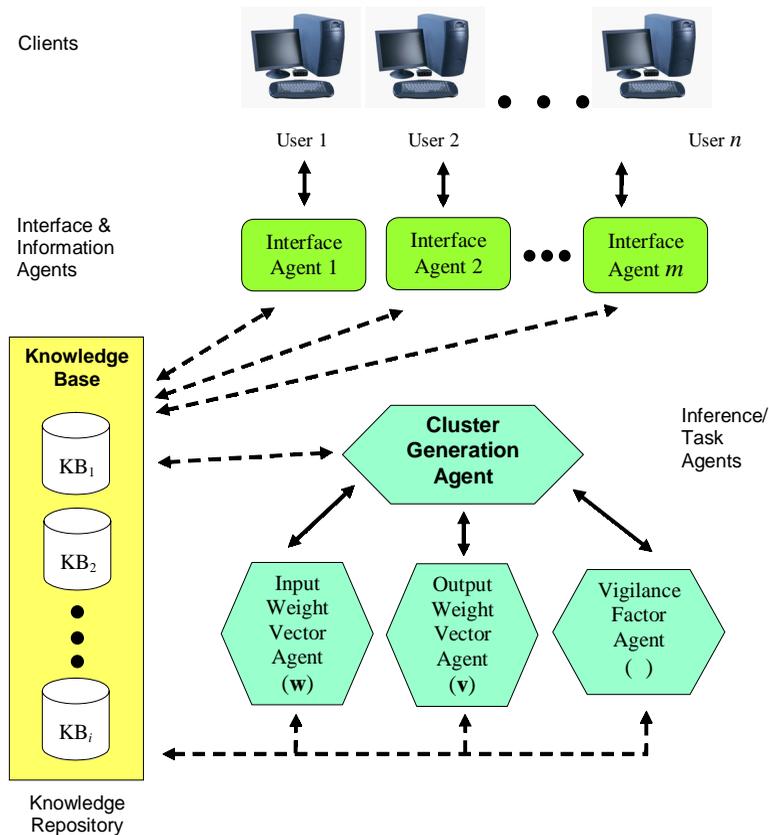
A generalized Multi-Agent System (MAS) architecture called RETSINA has been presented by Sycara et al. (1996). Three classes of agents are proposed in Sycara et al. (1996): interface, task, and information. The major objective of the interface agents is to interact with the clients/users in order to receive user specifications and to deliver the results. They

acquire, model, and utilize user preferences to guide system coordination in support of the user's tasks. Task agents perform the majority of autonomous problem solving and, thus, are regarded as the inference schema of RETSINA. It exhibits higher levels of sophistication and complexity than either an interface or an information agent. Information agents provide intelligent access to a heterogeneous collection of information sources depicted at the bottom of Figure 6.

Having gathered an intuitive understanding of a generalized Multi-Agent System (MAS) architecture in Sycara et al. (1996), the architecture of the proposed MAS for mail server management is built similar to the RETSINA architecture. Figure 6 shows our proposed architecture, and its components are briefly described in subsequent paragraphs.

In the application domain of mail server memory management, it is interesting to note that the users or clients act as information source. That is, the input pattern vector \mathbf{x} , to be incorporated into the Adaptive Resonance Theory (ART) algorithm used by the four task agents and represented by shaded hexagons, is collected by each corresponding interface agent for a user. In essence, the interface agents

Figure 6. Multi-agent system architecture for mail server management



function as both interacting and information agents. There may be as many as m number of interface agents for the n number of clients, where $n \geq m$, since some users may choose to decline the automated service that tracks and manages their e-mail messages. Instead, they will be liable for managing their memory quota manually. Dashed arrows indicate access to the knowledge repository, in which patterns of e-mail usage for each client are retained in the form of rules.

Three task agents — Input Weight Vector, Output Weight Vector, and Vigilance Factor — interact dynamically with knowledge bases in order to adjust asynchronously in real time. In effect, the neu-

ral network based on ART learns without supervision, and a unique cluster is generated for each e-mail message. Possible clusters were illustrated in Figure 5.

Implementation of the proposed architecture has been initiated with Dell PowerEdge™ 1850 server with Intel Xeon processor at clock speed up to 3.0GHz and 1.0GB RAM. At present, a closed proprietary network with two clients is being tested for building the prototype multi-agent system. Network Operating System (NOS) of choice is Linux, with Visual C++ as the major development platform.

In building a technical infrastructure, the following obstacles are expected, among others:

- **Difficulty of Data Mining.** Execution of a spyware is inevitable on the client machine, which may develop legal implications. At present, cookies are being considered as the quick implementation vehicle.
- **Network Utilization.** Running a multi-agent system in real time may decrease the network performance in terms of bandwidth utilization and reliability to a critical level.
- **Portability/Scalability.** There is a constant portability problem of this proposed agent system with respect to operating system and/or hardware platform. Platforms running operating systems other than Linux have to be simulated and tested for, once this prototype completes its pilot run.

CONCLUSION

A conceptual framework for a real-time multi-agent system built with neural network and knowledge base has been presented in this paper, with an emphasis on information resource management (IRM). Managing client as well as server knowledge concerning e-mails was selected as the scope of this research due to its significance as a major communication vehicle in the e-economy.

Adaptive Resonance Theory (ART) was the primary algorithm of choice for the neural-network engine due to its capability to achieve unsupervised learning. A simplified numerical example was provided to illustrate the effectiveness of ART applied to the problem domain.

Marked differences were discovered for the two major e-mail protocols for the server: IMAP and POP. As a conse-

quence, the suggested multi-agents are expected to be most effective for managing client knowledge under the IMAP and for managing server knowledge under the POP structure.

Challenges of implementing the proposed framework include but are not restricted to data mining, network utilization, portability, and security.

REFERENCES

- Ahuja, M.K. (1998). Network structure in virtual organizations. *Journal of Computer-Mediated Communication*, 3(4).
- Applen, J.D. (2002). Technical communication, knowledge management, and XML. *Technical Communication*, 49(3), 301-313.
- Cohen, F. (2004). *JAVA testing and design: From unit testing to automated Web tests*. New York: Prentice-Hall.
- Curran, T.A., & Ladd, A. (2000). *SAP R/3 business blueprint: Understanding enterprise supply chain management, 2/E*. New York: Prentice-Hall.
- Davenport, T.H., & Prusak, L. (1998). *Working knowledge: How organizations manage what they know*. Boston: Harvard Business School Press.
- Dutta, S. (1997). Strategies for implementing knowledge-based systems. *IEEE Transactions on Engineering Management*, 22, 79-90.
- Eales, R.T.J. (2004). A knowledge management approach to user support. In *Proceedings of the Fifth Conference on Australasian User Interface Conference (AUIC2004)*, Dunedin (pp. 33-38).
- Flenner, R. et al. (2002). *Java P2P unleashed: With JXTA, Web Services, XML, Jini, JavaSpaces, and J2EE*.

- New York: Prentice-Hall.
- Gallivan, M.J., Eynon, J., & Rai, A. (2003). The challenge of knowledge management systems: Analyzing the dynamic processes underlying performance improvement initiatives. *Information Technology and People*, 16(3), 326-352.
- Gold, A.H., Malhotra, A., & Segars, A.H. (2001). Knowledge management: An organizational capabilities perspective. *Journal of Management Information Systems*, 18(1), 185-214.
- Grudin, J. (1988). Why CSCW applications fail: Problems in the design and evaluation of organizational interfaces. In *Proceedings of the 1988 ACM Conference on Computer Supported Cooperative Work*, New York (pp. 85-93).
- Kanawati, R., & Malek, M. (2002). A multi-agent system for collaborative bookmarking. In *Proceedings of the First International Joint Conference on Autonomous Agents and Multi-agent Systems: Part 3*, Bologna, Italy (pp. 1137-1138).
- Kankanhalli, A., Tanudidjaja, F., Sutanto, J., & Tan, C. Y. (2003). The role of IT in successful knowledge management initiatives. *Communications of the ACM*, 46(9), 69-73.
- Lansdale, M. (1988). The psychology of personal information management. *Applied Ergonomics*, 19(1), 55-66.
- Laurillau, Y., & Nigay, L. (2002). Clover architecture for groupware. In *Proceedings of the 2002 ACM Conference on Computer Supported Cooperative Work*, (pp. 236-245).
- Majchrzak, A., Rice, R.E., Malhotra, A., & King, N. (2000). Technology adaptation: The case of a computer-supported inter-organizational virtual team. *MIS Quarterly*, 24(4), 569-600.
- Malek, M. (2000). Hybrid approaches integrating neural networks and case based reasoning: From loosely coupled to tightly coupled models. In K.P. Sankar, S.D. Tharam, & S. Y. Daniel (Eds.), *Soft computing in case-based reasoning* (pp. 73-94). New York: Springer.
- Markus, M.L. (2001). Toward a theory of knowledge reuse: Types of knowledge reuse situations and factors in reuse success. *Journal of Management Information Systems*, 18(1), 57-93.
- Markus, M.L., & Connolly, T. (1990). Why computer supported collaborative work applications fail: Problems in the adoption of interdependent work tools. In *Proceedings of the International Conference on Computer Supported Collaborative Work (CSCW '90)*, New York (pp. 371-380).
- Mathe, N., & Chen, J.R. (1998). Organizing and sharing information on the World-Wide Web using a multi-agent systems. In *Proceedings of ED-MEDIA '98 Conference on Educational Multimedia and Hypermedia*, Freiburg, Germany.
- Murch, R. (2004). *Autonomic computing*. New York: Prentice-Hall.
- Nonaka, I., & Takeuchi, H. (1995). *The knowledge-creating company: How Japanese companies create dynamics of innovation*. New York: Oxford University Press.
- Pinelle, D., & Gutwin, C. (2002). Groupware walkthrough: Adding con-

- text to groupware usability evaluation. In *Proceedings of the ACM SIGCHI Conference on Human Factors in Computing Systems: Changing our World, Changing Ourselves*.
- Pinelle, D., Gutwin, C., & Greenberg, S. (2003). Task analysis for groupware usability evaluation: Modeling shared-workspace tasks with the mechanics of collaboration. *ACM Transactions on Computer-Human Interaction*, 10(4), 281-311.
- Roos, L.L., Soodeen, R-A., Bond, R., & Burchill, C. (2003). Working more productively: Tools for administrative data. *Health Services Research*, 38(5), 1339-1357.
- Roseman, M., & Greenberg, S. (1996). Building real-time groupware with GroupKit, a groupware toolkit. *ACM Transactions on Computer-Human Interaction*, 3(1), 66-106.
- Sycara, K., Pannu, A., Williamson, M., Zeng, D., & Decker, K. (1996). Distributed intelligent Agents. *IEEE Intelligent Systems*, 11(6), 36-46.
- Taylor, W.A. (2004). Computer-mediated knowledge sharing and individual user differences: An exploratory study. *European Journal of Information Systems*, 13, 52-64.
- Wiesenfeld, B.M., Raghuram, S., & Garud, R. (1998). Communication patterns as determinants of organizational identification in a virtual organization. *Journal of Computer-Mediated Communication*, 3(4).
- Willow, C.C. (2002). A feedforward multi-layer neural network for machine cell formation in computer integrated manufacturing. *Journal of Intelligent Manufacturing*, 13(2), 75-87.
- Willow, C.C. (2005a). *Neural network-based multiple agent system for Web server management*. Monmouth, NJ: Monmouth University.
- Willow, C.C. (2005b). *Neural network-based multiple agent system for application server management*. Monmouth, NJ: Monmouth University.
- Ye, Y., Fischer, G., & Reeves, B. (2000). Integrating active information delivery and reuse repository systems. In *Proceedings of the Eighth ACM SIGSOFT International Symposium on Foundations of Software Engineering: 21 Century Applications*, San Diego, California (pp. 60-68).
- Zurada, J.M. (1992). *Introduction to artificial neural systems*. St. Paul, MN: West Press.

ACKNOWLEDGMENTS

My deepest gratitude is extended to Drs. Edward Christensen, Bob Smith, and Diana Sharpe of Monmouth University for their time and invaluable suggestions while acquiring ideas and intuition for this paper. Gratification is also expressed to the members of the Computing Society for *INFORMS* and Dr. Ramesh Sharda of Oklahoma State University in particular for thoughtful insights.

This paper was supported in part by the 2004 Summer Research Grant from the Business Council of the Monmouth University, West Long Branch, New Jersey.

Charles Willow, PhD, is an assistant professor of management information systems (MIS) and management of technology (MOT) in the School of Business Administration at the Monmouth University, West Long Branch, New Jersey, USA. As an engineer-and-computer-scientist-turned-management-faculty, Dr. Willow's research agenda has been extensive, ranging from information systems development to case-driven strategic management issues, systems engineering, and operations research. His current research interests include information systems and network security, neural network applications, intelligent software agents, computer-generated graphics, cost-model analysis of Internet-based business, and strategic management of technology. Dr. Willow's papers have appeared in journals such as ACM Transactions on Information Systems, Journal of Intelligent Manufacturing, and IEEE Transactions on Systems, Man, and Cybernetics, among others. He is a member of the IEEE Computer Society, ACM, INFORMS Computing Society, AIS, and the National Society of Professional Engineers (NSPE). At present, Dr. Willow is on the editorial board for the Journal of Management Information Systems (JMIS), MIS Quarterly (MISQ), and Information Systems Research (ISR). Outside academia, he has been an active member of the International Simultaneous Interpreters Society (ISIS) and the International Judo Federation (IJF).