# A Conceptual Framework for Agent-based Information Resource Management

Charles C. Willow

Management Information Systems, Monmouth University, West Long Branch, NJ 07764-1898, U. S. A.
cwillow@monmouth.edu; www.monmouth.edu/~cwillow

**Abstract.** The information systems manager is often constrained by maintaining a certain threshold amount of memory for an organization. However, this 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 dynamic nature of the tasks, incorporation of a neural network architecture with tacit knowledge base is suggested.

**Keywords**: Information resource management, automatic intelligent agent, automata, knowledge management, neural networks.

## 1 Introduction

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 administrator is the correct forecast of overall 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. One may argue that the cost of memory is declining rapidly, and its management may not be a factor effecting IRM. Contrary to this common misbelief, however, a number of authors suggest there be a certain threshold for memory management [2, 7, 8, 9, 15, 18, 19] 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.

Major contribution of this paper lies in the development of framework for building multiple agents for the mail server, with emphasis on memory management. By far, agent-based autonomous systems (*i.e.* automata) have been adopted as one of the better methods for managing virtual organizations in various applications [7, 15, 16,

22]. They range from e-commerce such as on-line travel arrangements to system diagnostics including on-line data quality audits and remote trouble-shooting.

Organization of this paper follows. In section 2, knowledge management for eliciting, building, and managing end-user preference and email usage patterns is discussed. Section 3 follows to illustrate the core system, 'multiple agents'. Suggestions for construction of the proposed framework are made in section 4, subsequently followed by conclusions in section 5.

## 2 Knowledge Management

The key to maintaining accuracy of the proposed multi-agent system lies in managing highly subjective knowledge for sharing and customizing/personalizing end-user memory usage patterns across the organization. Information Technology (IT) may support Knowledge Management (KM) in two classes: codification and personalization [8]. In essence, the codification approach manages structured knowledge, whereas personalization manages unstructured, tacit knowledge. Because email usage patterns for end-users may entail both types of knowledge, separate knowledge base or repository is suggested for the framework of this research. That is, there may be common patterns of email management among end users such as removing messages which are more than 36 months old or organizing their mail folders in every 30 days, and so forth on the one hand. On the other hand, each user may have highly subjective patterns which may not be consistent with those *codified* knowledge. Lansdale [9] emphasized the need for Cognitive Interface Tools (CITs) to collect, organize, build, and share both types of knowledge for the office system in his early research. However, not many literatures have been dedicated to solving this problem to date.

### 2.1 Knowledge Management Attributes

As noted in Lansdale [9], the process of information retrieval in the human mind is fundamentally different from filing or library system, in which items are accessed by location rather than their meaning. The first notion is that people recall chronological information about information: what else was happening at roughly the same time. Consequently, 'time stamp' of emails may be a good source of structured information or knowledge.

Association is another means by which humans retrieve information. Each email 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 email usage.

A set of six generic attributes associated with emails, 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 her/his email 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 email account holder, one client may choose to either remove or archive (on local memory store) emails based on recipient/sender, event/subject, attached file(s), significance, or any combination of the four, so far as tacit knowledge management is concerned. Alternatively, s/he may simply choose to archive or remove emails with regard to structured information such as time stamp and/or size. It is precisely this knowledge associated with each email client, which is expected to be elicited by the automated multiple agents proposed in this paper, preferably in real time.

**Table 1.** Attributes of Knowledge Management for Email 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 |

## 3 Intelligent Automated Agents

Under a certain memory constraint for each email 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 – may often realize their emails 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 if not legal implications. Thus, an automated system which may *advise* the clients in real time about their email usage patterns is considered as an attractive alternative for information resource management. Once the client is logged onto the system, the automatic intelligent agent generates a list of email messages which are to be removed, as well as those which are highly likely to be candidates for local archives. In addition, another agent system is suggested 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 into the web and applications servers in Willow [25, 26] at present.

Neural networks (NN) are employed as the inference engine for the proposed multi-agent system. A Neural Network typically processes large-scale problems in terms of dimensionality, amount of data handled, and the volume of simulation or neural hardware processing [24]. It emerged as an area of Artificial Intelligence (AI) to mimic the 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' had brought the attention of researchers to pursue classes of 'computing and processing', such as neural networks. 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 in emulating the human neurons were realized. To this end, the jargon NN at present, is more appropriate if it were to be replaced with *parallel, distributed simulation*. An excellent taxonomic view of NN models is provided by Willow [24].

### 3.1 Adaptive Resonance Theory

Adaptive Resonance Theory (ART), as illustrated in Zurada [28], is a unique *unsupervised* class of neural network algorithm. It has the novel property of controlled discovery of clusters. Further, the ART network may accommodate new clusters without affecting the storage or recall capabilities for clusters which were already learned, fit for the scope of the problem of this paper.

Nomenclature of the model follows:

Subscripts and Superscripts

$i$ Subscript for input variable, $x$; $i = 1, \ldots, n$.
$j$ Subscript for output clusters, $j = 1, \ldots, M$.
$m$ Subscript for output neuron, $y$ or neuron of hidden layer; $m = 1, \ldots j, \ldots, M$.
$k$ 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, $\mathbf{x} = [x_1, \ldots, x_n] = < x_1, \ldots, x_n >$.

Variables

$\mathbf{x}$ Input vector; $\mathbf{x} = [x_1, \ldots, x_n]$.
$\mathbf{w}$ Weight of the input vector; $\mathbf{w} = [w_1, \ldots, w_n]$.
$\mathbf{y}$ Output vector; $\mathbf{y}^k = [y_1, \ldots, y_M]$.
$\rho$ 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 $\rho$.
$\mathbf{v}$ Weight vector for verifying cluster exemplar proximity; $\mathbf{v} = [v_1, \ldots, v_n]$.
$t$ Update index for weights, $\mathbf{w}$ and $\mathbf{v}$.

Algorithm for ART is referenced to Zurada [28], and is summarized as follows:

Step 1: Initialization

The vigilance threshold, $\rho$, 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

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

$$\mathbf{V} = [1] \tag{2}$$

$$0 < \rho < 1 \tag{3}$$

Step 2: Input Neuron Processing

Binary unipolar input vector **x** is presented at input nodes, $x_i = 0, 1$ for $i = 1, 2, \ldots, n$.

Step 3: Matching Score Computation

All matching scores are computed as follows:

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

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

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

Step 4: Resonance

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

$$\frac{1}{\|x\|} \sum_{i=1}^{n} v_{ij} x_i > \rho \tag{6}$$

where, the norm is defined as

$$\|x\| \equiv \sum_{i=1}^{n} |x_i| \tag{7}$$

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 $i = 1, 2, \ldots, M$, and are computed as follows:

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

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

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 mode 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, $\rho$, 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 sections 3.2 and 3.3.

### 3.2 Client Agent Architecture with ART

The ART architecture for the suggested client agent follows. The major purpose of the client agent is to provide real-time knowledge regarding email management for each client end-user. Fig. 1 follows to illustrate.

For each email message, an input vector comprised of the following 7-tuple attribute is produced; $\mathbf{x} = \langle x_1, \ldots, x_7 \rangle$:

$x_1$ Age of the message. It is automatically computed as system-generated time (TNOW) – (Time_Stamp).

$x_2$ Size of the email, 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 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 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.

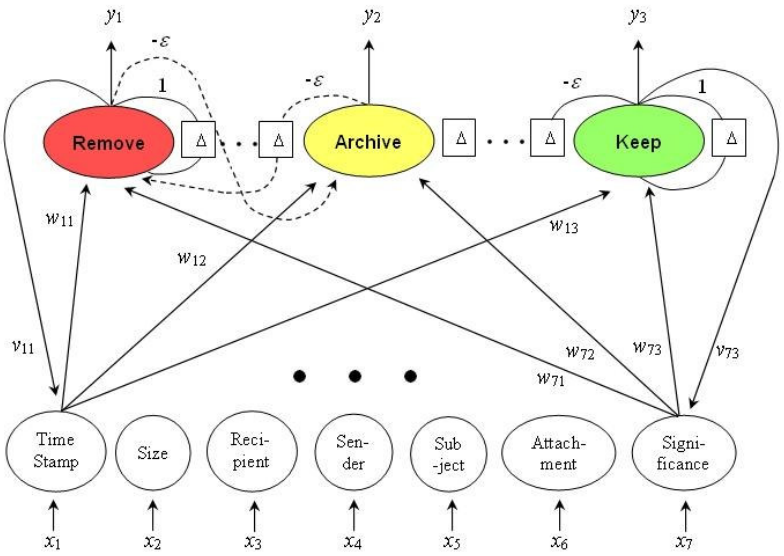


**Fig. 1.** ART for the Client Agent

$x_7$ Significance of the email 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 email. 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 = <$x_1$, …, $x_7$>
= <60, 2000, *jmis@xyz.edu*, null, "publication consideration", 0203, 5>
Message #2
= <02, 20, null, *jdoe@usa.com*, "Greetings", 0000, 1>
Message #3
= <14, 250, *family@home.org*, null, "get together", 0000, 5>

Thus, each message forms an input pattern vector, **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:

$\mathbf{x_1}$ = <60, 2000, 4, null, 5, 02, 5>
$\mathbf{x_2}$ = <02, 20, null, 5, 1, 00, 1>
$\mathbf{x_3}$ = <14, 250, 5, null, 1, 00, 5>

When $\mathbf{x_1}$ is presented, the steps of the ART algorithm are:

$$M = 3; \quad n = 7; \quad w_{ij} = \frac{1}{n+1} = 1/8 = 0.125;$$

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

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:

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

$$w_{52}(2) = \frac{1 \times 5}{0.5 + [4 + 0 + 5 + 2 + 5]} = 0.3030 = w_{72}$$

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

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

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 (6) is classified as the 'Remove' cluster.

In essence, the ART-based neural network processing is expected to advise the email clients of possible action(s) for each (email) message, while self-organizing the (neural) network dynamically.

### 3.3 Server Agent Architecture with ART

The Adaptive Resonance Theory (ART) model may also be employed for another set of agent system dedicated to assisting the email server administrator(s). The two systems of agents, targeting individual clients as well as administrators, may then 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: **I**nternet **M**essage **A**ccess **P**rotocol (IMAP) and **P**ost **O**ffice **P**rotocol (POP). IMAP integrates messages on the shared mail server, and permits client email programs to access remote message stores as if they were local. Thus, the memory burden on the server is far greater for IMAP than for POP. However, complete monitoring of client emails 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. Table 2 follows to illustrate.

**Table 2.** Classes of Information Availability based on Email Application Protocols

| Email Protocols / Information Process | IMAP *(central)* | POP *(distributed)* |
|---|---|---|
| Server-based | Server only | **Mixed* |
| User-based | **Mixed* | Client only |

## 4 Implementation

A *generalized* Multi-Agent System (MAS) architecture entitled RETSINA has been presented by Sycara et al. [21], in which three classes of agents are proposed: *interface*, *task*, and *information* agents.

The architecture of the proposed MAS for mail server management is built similar to the RETSINA architecture. Fig. 2 shows the proposed architecture, and its components are briefly described below.

In the application domain of mail server memory management, it is interesting to note that the users or clients themselves act as information source. That is, the input pattern vector **x**, to be incorporated into the Adaptive Resonance Theory (ART) algorithm used by the four task agents, represented by shaded hexagons, is collected by each corresponding interface agent for a user. In essence, the interface agents 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 email messages. Instead, they will be liable for managing their memory quota manually. Dashed arrows indicate access to the knowledge repository, in which patterns of email usage for each client are retained in the form of rules.

Three task agents, 'Input Weight Vector', 'Output Weight Vector', and 'Vigilance Factor' dynamically interact with knowledge bases to adjust themselves asynchronously in real time. In effect, the neural network based on ART learns without supervision, and a unique cluster is generated for each email message. Possible clusters were illustrated in Fig. 1.

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.

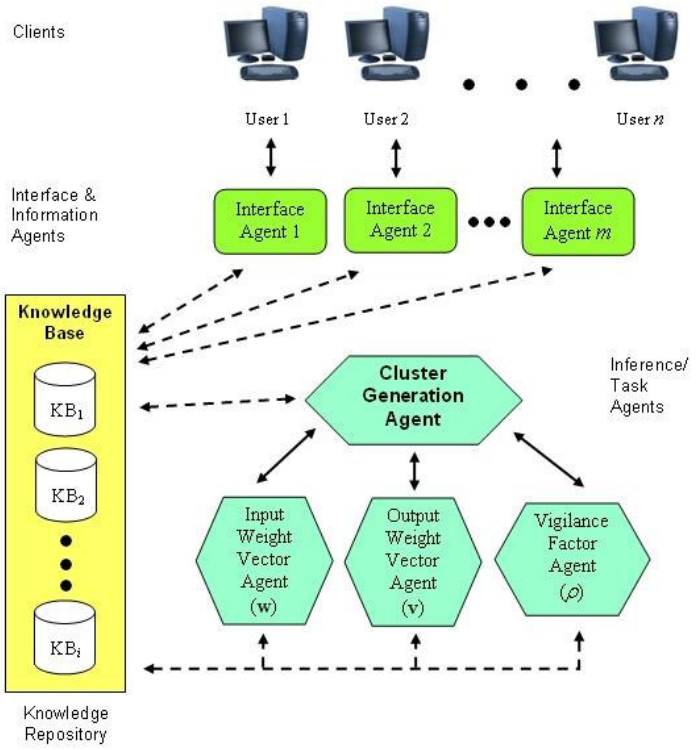


**Fig. 2.** Multi-Agent System Architecture for Mail Server Management

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.

## 5 Conclusions

A conceptual framework for real-time multi-agent system built with neural network and knowledge base has been presented in this paper, with emphasis on Information Resource Management (IRM). Managing client as well as server knowledge concerning emails 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 are discovered for the two major email protocols for the server: IMAP and POP. 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 not restricted to data mining, network utilization, portability, and security.

**Acknowledgments.** This paper was in part supported by the 2005-2006 Research Grant from the Business Council and in most part by the 2005-2006 Faculty Grants-in-Aid for Creativity funds of the Monmouth University, West Long Branch, New Jersey, U. S. A.

## References

1. Ahuja, M. K.: Network Structure in Virtual Organizations. Journal of Computer-Mediated Communication. 3 (1998) [online journal]
2. Applen, J. D.: Technical Communication, knowledge management, and XML. Technical Communication. 49 (2002) 301-313
3. Davenport, T. H., Prusak, L.: Working Knowledge: How organizations manage what they know. Harvard Business School Press, Boston (1998)
4. Dutta, S.: Strategies for implementing knowledge-based systems. IEEE Transactions on Engineering Management. 22 (1997) 79-90
5. Gold, A. H., Malhotra, A., Segars, A. H.: Knowledge management: an organizational capabilities perspective. Journal of Management Information Systems. 18 (2001) 185-214
6. Grudin, J.: Why CSCW applications fail: problems in the design and evaluation of organizational interfaces. Proceedings of the 1988 ACM Conference on Computer Supported Cooperative Work. New York (1988) 85-93
7. Kanawati, R., Malek, M.: A Multi-agent system for collaborative bookmarking. Proceedings of the First ACM International Joint Conference on Autonomous Agents and Multi-agent Systems: Part 3. Bologna, Italy (2002) 1137-1138
8. Kankanhalli, A., Tanudidjaja, F., Sutanto, J., Tan, C. Y.: The role of IT in successful knowledge management initiatives. Communications of the ACM. 46 (2003) 69-73
9. Lansdale, M.: The psychology of personal information management. Applied Ergonomics. 19 (1988) 55-66
10. Laurillau, Y., Nigay, L.: Clover Architecture for Groupware. Proceedings of the 2002 ACM Conference on Computer Supported Cooperative Work. (2002) 236-245

11. Majchrzak, A., Rice, R. E., Malhotra, A., King, N.: Technology adaptation – the case of a computer-supported inter-organizational virtual team. MIS Quarterly. 24 (2000) 569-600
12. Malek, M.: Hybrid approaches integrating neural networks and case based reasoning – from loosely coupled to tightly coupled models. In: Sankar, K. P., Tharam, S. D., Daniel, S. Y. (eds): Soft Computing in Case-based Reasoning. Springer, New York (2000) 73-94
13. Markus, M. L., and Connolly, T.: Why Computer Supported Collaborative Work applications fail – problems in the adoption of interdependent work tools. Proceedings of the International Conference on Computer Supported Collaborative Work (CSCW '90). Association for Computing Machinery, New York (1990) 371-380
14. Markus, M. L.: Toward a theory of knowledge reuse: types of knowledge reuse situations and factors in reuse success. Journal of Management Information Systems. 18 (2001) 57-93
15. Mathe, N., Chen, J. R.: Organizing and sharing information on the World-Wide Web using a multi-agent systems. Proceedings of ED-MEDIA '98 Conference on Educational Multimedia and Hypermedia. Freiburg, Germany (1998)
16. Murch, R.: Autonomic Computing. Prentice-Hall, New York (2004)
17. Nonaka, I., Takeuchi, H.: The Knowledge-creating company – How Japanese companies create Dynamics of Innovation. Oxford University Press, New York (1995)
18. Pinelle, D., Gutwin, C., Greenberg, S.: Task analysis for groupware usability evaluation – modeling shared-workspace tasks with the mechanics of collaboration. ACM Transactions on Computer-Human Interaction. 10 (2003) 281-311
19. Roos, L. L., Soodeen, R-A., Bond, R., Burchill, C.: Working more productively – tools for administrative Data. Health Services Research. 38 (2003) 1339-1357
20. Roseman, M., Greenberg, S.: Building real-time groupware with GroupKit, a groupware toolkit. ACM Transactions on Computer-Human Interaction. 3 (2001) 66-106
21. Sycara, K., Pannu, A., Williamson, M., Zeng, D., Decker, K.: Distributed Intelligent Agents. IEEE Intelligent Systems. 11 (1996) 36-46
22. Taylor, W. A.: Computer-mediated knowledge sharing and individual user differences – an exploratory study. European Journal of Information Systems. 13 (2004) 52-64
23. Wiesenfeld, B. M., Raghuram, S., Garud, R.: Communication Patterns as Determinants of Organizational Identification in a Virtual Organization. Journal of Computer-Mediated Communication. 3 (1998) [online journal]
24. Willow, C. C.: A Feedforward Multi-layer Neural Network for Machine Cell Formation in Computer Integrated Manufacturing. Journal of Intelligent Manufacturing. 13 (2002) 75-87
25. Willow, C. C.: Neural Network-based Multiple Agent System for Web Server Management. MU WP No. W05-02, Management Information Systems, School of Business Administration, Monmouth University, NJ (2005a)
26. Willow, C. C.: Neural Network-based Multiple Agent System for Application Server Management. MU WP No. W05-03, Management Information Systems, School of Business Administration, Monmouth University, NJ (2005b)
27. Ye, Y., Fischer, G., Reeves, B.: Integrating Active Information Delivery and Reuse Repository Systems. Proceedings of the Eighth ACM SIGSOFT International Symposium on Foundations of Software Engineering – 21 Century Applications. San Diego (2000) 60-68
28. Zurada, J. M.: Introduction to Artificial Neural Systems. West Press, St. Paul (1992)