Submitted Dec. 2002: IIIS Agent Based Computing, Orlando, July 27-30, 2003.

VIPAR: Advanced Information Agents discovering knowledge in an open and changing environment

Thomas E. Potok, Mark Elmore, Joel Reed, and Frederick T. Sheldon[1]
Oak Ridge National Laboratory[2], Computational Science and Engineering
Oak Ridge, Tennessee 37831-6414

Abstract

Given the rapid evolution of information technology, most people on a daily basis are confronted by more information than they can reasonably process. The challenge to organize/classify and comprehend immense amounts of information is vitally important to the scientific, business, and defense/security communities (particularly when projecting the future evolution of information technology. For example, the defense/security community is faced with the daunting challenge of gathering and summarizing information so that military/political leaders can make informed decisions and recommendations. One such group, the Virtual Information Center (VIC) at US Pacific Command, gathers, analyzes, and summarizes information from Internet-based newspapers on a daily basis (largely a manual, time and resource intensive process).

This paper discusses the VIPAR project[3], which has addressed this need. Intelligent agent technology was chosen 1) to utilize the ability for broadcast as well as peer-to-peer communication among agents, 2) to follow rules outlined in an ontology, and 3) because of the ability for agents to suspend processing on one machine, move to another, and resume processing (persistence). These strengths are well suited to addressing the challenges of automatically gathering Internet-based information.

The VIPAR system is a multi-agent system that demonstrates the ability to self-organize newspaper articles in a manner comparable to humans. The VIPAR system demonstrates the important ability where agents use a flexible RDF ontology to monitor/manage Internet-based newspaper information. Moreover, VIPAR extends this capability by dynamically adding/clustering new information entering the system. The VIPAR system includes thirteen information agents that manage thirteen different newspaper sites. Results from the project show that VIPAR can organize information in a way comparable to human organized information and validates the agent approach taken.

1 Introduction

One main focus of information technology is to provide more information to people faster. Clearly, this challenge has been met in quite remarkable ways. Vast information sources are readily available via the Internet. However, this success raises the challenge of how to quickly/effectively organize and classify this information, a significant hurdle for industry and government organizations (including the Virtual Information Center (VIC) at US Pacific Command). VIC must produce daily Pacific region news summaries that may influence governmental policy decisions. Performed daily, analysts scan Internet newspapers, and create a summary of relevant news articles. The appropriate governmental/civilian decision makers consequently can better use the distilled information in their decision-making processes. Ironically, advances in information technology provide significant access portals, but unfortunately a great deal of human effort is required to effectively organize and classify the discovered information.

This paper describes the Virtual Information Processing Agent Research (VIPAR) project that has developed a to organize and classify large amounts of information [11]. The two main issues include the following: 1) How to efficiently gather heterogeneous and frequently changing information from the Internet, 2) How to organize, classify, and quickly present this information to an analyst. Reviewing the literature, it appears that the only viable technology to address these complex issues are information agents. Information agents can provide heterogeneous access and translation of Internet information [1], likewise, they can be effectively used to organize complex information [2].

The results include the following: 1) information is gathered automatically from some number of Internet newspapers, and (2) the information is classified in a way that rivals human classification. Particularly significant is the ability for the agents to use a flexible RDF ontology to transform heterogeneous HTML documents to XML tagged documents, and their ability to rapidly cluster newspaper articles that arrive in an asynchronous manner.

[1] Potok (potokte@ornl.gov, 574-0834) is director of the Applied Software Engineering Research group at Oak Ridge National Laboratory (ORNL). Sheldon, the contact author (sheldon@acm.org, 865-576-1339, 865-574-6275 fax), is a member of the research staff at ORNL and director of the SEDS (Software Engineering for Dependable Systems) Lab.

[2] This manuscript has been authored by a contractor of the U.S. Government (USG) under contract No. DE-AC05-96OR22464. The USG retains a non-exclusive, royalty-free license to publish or reproduce the published form of this contribution, or allow others to do so, for USG Purposes.

[3] The Virtual Information Processing Agent Research (VIPAR) is a software research and development project funded by the Office of Naval Research (ONR) via Interagency Agreement 2302-Q326-A1 with the Department of Energy's ORNL.

2 Background

The classification of large amounts of electronic information is not a new problem. Yet, with the advent of several new technologies, a new solution is emerging, which allows for detailed analysis of large collections of heterogeneous unstructured information. This solution can be viewed in two parts, first how to gather and structure information, and second how to organize and classify information.

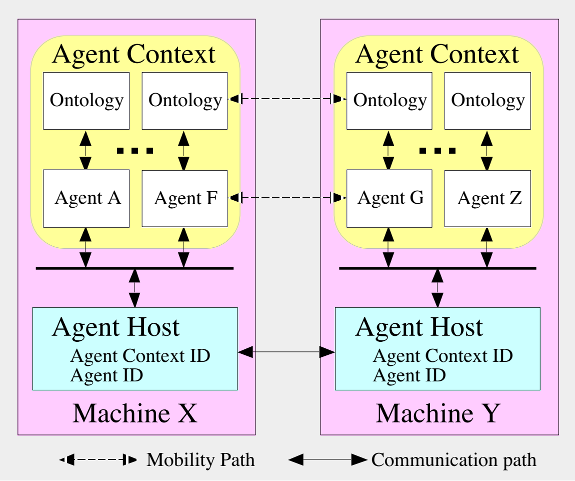


Figure 1 ORMAC mobility-communication architecture.

There are two broad approaches to addressing the problem of how to efficiently gather and structure frequently changing heterogeneous information over the Internet. The most obvious is the use of any number of Internet search engines. These engines typically use programs that recursively traverse Internet links, capturing non-trivial terms on each page. These traversed pages are then organized based on the terms encountered in each document enabling a wide variety/number of documents to be traversed and made available for keyword searches. The weaknesses include 1) existing pages in the system are infrequently re-traversed tending to make the information stale, 2) the Internet pages have no consistent format, and therefore, the content of a page cannot be easily discerned, 3) the documents are organized based solely on the presence of a keyword in a document (regardless of other attributes like timeliness).

Alternatively, our approach gathers and structures Internet information using agents. The agents provide various ways to retrieve/organize information, including agents that are capable to access multiple sources, and to filter based on relevance to the user [3,4,5]. The most basic of these systems utilize non-cooperating agents that perform the information retrieval task. The next step uses cooperating agents, and finally, adaptive agents that deal with uncertain, incomplete, or vague information [6]. These agents provide the capability to efficiently gather heterogeneous and frequently changing information. While the concept is appealing, much of the literature describes various agent characteristics/attributes with little detail on the specific advantages. Another challenge is whether the inherent and somewhat chaotic structure of newspaper articles can be transformed into a common schema.

A number of technologies are available to structure/organize the retrieved information (i.e., marked up in XML). The most basic uses a set of keywords as a means of document classification. This rather trivial approach yields mixed results since documents that contain the same words may have no semantic relationship. Yet another organizes information into a vector space model (VSM) to represents each unique word within a document collection as a dimension in space, while each document represents a vector within that multidimensional space. Vectors that are close together in this multidimensional space form document clusters (i.e., groups of that are similar). Through local and global weighing schemes this approach can be tuned to provide a way to compare the similarity of one document to another. Furthermore, clustering techniques can be easily applied using the VSM approach. One of the limitations of this approach is that the entire document set must be available at the time of the analysis, and the clustering algorithms are computationally expensive (i.e., O(n^3) in complexity for n documents). A third approach to organizing information is to use neural networks as a means of determining patterns within documents, the concept being that documents with similar word patterns are similar in content. These models are built on the premise that historic patterns will hold in the future. This is not the case in articles where topics, people, and events change at frequent intervals [7]. On this basis, the VSM approach is best suited for the problem at hand; however, the challenge of dynamically creating VSM and clustering similar documents was an issue.

3 Approach

To organize and classify Internet newspaper information, we developed cooperative and adaptive information agents. These agents cooperate to gather and organize information. We created a number of different agent types, and implemented a communication protocol enabling them to interact. For example, one team of agents gathers information from individual newspapers, another agent team analyzes the articles, and organizes the information. To deploy such agents we used the Oak Ridge Mobile Agent Community (ORMAC) framework. This framework has been under development over the course of several agent-based research projects. ORMAC is a generic agent framework providing transparent agent communication and mobility across any Internet connected host (see Fig. 1).

ORMAC enables an agent community to be quickly created using a set of machines with each machine executing the ORMAC agent host software. The ORMAC agent host software allows agents to migrate among machines. Moreover, agents to be truly mobile by physically moving from machine to machine as needed. This capability helps facilitate communication among agents within an ORMAC community. ORMAC agents can also interact with systems and agents that are not part of the community. Agent mobility through the Internet is very limited based on the enforced Internet security limitations. The ORMAC framework uses the Foundation for Intelligent Physical Agent (FIPA) compliant agent communication language (ACL) messages. This allows any FIPA compliant agent, such as SPAWAR's TIIERA system, to be able to interact with an ORMAC agent [8,9]. Within the ORMAC community, each agent host has a name server responsible for tracking where agents are currently being hosted. In addition, the name server is responsible for answering queries from agents trying to locate other agents in the community. For example, an agent may want to broadcast information to all of the agents within the community. The name server for each agent host is used to locate all of the agents so that the message can be delivered.

Agents move from one machine to another by changing *agent hosts*. The *ontologies* move with the agents. When an agent is received at an agent host, the agent host provides it with an agent context. This agent context is the agent's only point of contact with the machine it is running on and provides machine specific environments for the agent to work. The agent is not allowed to directly communicate with the agent host or other agents. This provides an architectural layer for security in the ORMAC system (written in JAVA, ORMAC uses Remote Method Invocation (RMI) to communicate among agents).

3.1 Why Agents?

This agent-based architecture provides several advantages over existing object-oriented technologies. In object-oriented systems, objects communicate through messages. The sender object must know the address of the receiver object, and the public methods of the receiver object. This is not the case with the ORMAC framework. Agents conform to a communication protocol that allows an agent to send a message to agent(s) without needing to know the address of the agent(s) or the specific methods available to the agent. This allows agents to move, yet still be in contact with other agents, and allows for agents to broadcast requests to some or all other agents. Another advantage with ORMAC is the ability to use ontology to direct agents through a task. An ontology can act as a script, or rule base for an agent to follow. This difference is perhaps more conceptual than practical because there currently is very little ontology standardization. We use the ontology to describe the characteristics of each newspaper within the system while agents use the ontology to correctly retrieve information from the newspapers.

There is also an advantage in the ability for agents to suspend processing on one machine, move to another, and resume processing. This provides the ability to prioritize tasks or agents by sending high priority agents to faster resources. Along the same lines, this capability can be used to load balance a system depending on the workload of each agent. Rather than having a global scheduler determine this priority or allocation of agents, the agents can determine this cooperatively. For example, within VIPAR, if the GUI agent requests processing, a message is sent to all other agents to stop retrieving messages until the GUI agent has completed its task.

3.2 ORMAC Agents for VIPAR

VIPAR is implemented as a set of ORMAC agents. The VIPAR system is broken into two main components: 1) a server, which performs most of the information retrieval and processing, and 2) a series of clients, which perform most of the user interface functions. Although these have certain conceptual parallels to a typical client/server system, in VIPAR using the ORMAC framework, these are peer processes where any peer may initiate communication. The VIPAR server is implemented using a set of information retrieval agents, whose task is to gather news related, non-redundant information from Internet newspapers, and to format the information using XML (see Fig. 2). A whiteboard agent acts as an information-clearing house. Agents submit their articles to the whiteboard agent, who manages the information ensuring no duplicate articles, archiving stale articles that are beyond a given number of days old, and providing articles to agents that have "subscribed" to the whiteboard. There is a team of cluster agents that organize articles into a vector space model (VSM), then into a cluster of articles.

3.3 Information Agents

The information agents must initially gather and organize heterogeneous Internet information. This is accomplished through the transformation of HTML formatted information into XML formatted information. The conversion from HTML to XML is a two-step process. First, we define an ontology to enable a common semantic representation and structuring of heterogeneous information. This ontology embodies the transformation of HTML formatted information to XML formatted information. This ontology is expressed in an XML variant called the Resource Description Framework (RDF, see http://www.w3.org/RDF/). The RDF syntax allows directed graphs to be expressed in an XML-like format. An Internet site is a collection of linked Internet pages. A site can be viewed as a directed graph, from which, RDF provides a solid way of modeling the linked pages. Furthermore, these RDF instructions can be understood and followed by a software agent. A series of RDF ontologies have been developed for the newspapers accessed by the VIPAR system. Each ontology describes a single newspaper's site, provides meta-information about the newspaper, and describes additional site-specific actions an agent is to take. Based on the ontological description of a newspaper site, a newspaper agent monitors and manages the information at the site

Second, an HTML to XML transformation is completed based on the defined ontology. Once an agent can understand an RDF file that describes the layout of an Internet newspaper site and its semantics, then this agent can autonomously go to the site, retrieve articles of interest, and convert the unstructured heterogeneous information into a structured XML formatted document. Each converted article will contain a rich set of XML tags ranging from the time and date the article was discovered, to the URL location of the information and to the XML tags that format the article itself. Each of these information agents monitors the newspaper site looking for new articles. Fresh articles are subsequently formatted and posted to the whiteboard agent.

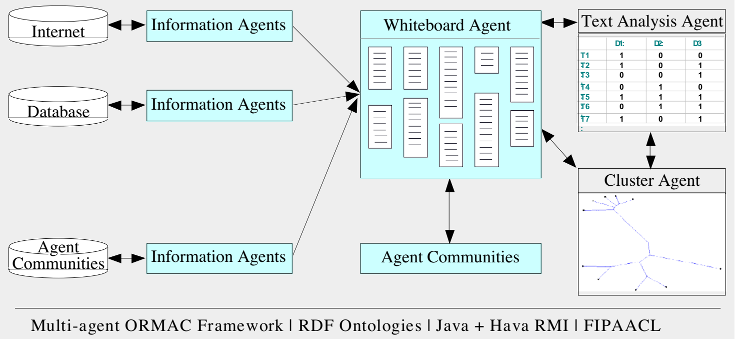


Figure 2. Architectural view of agents within the VIPAR system.

The ontological description (OD) of the site includes a root URL from which the agent is to begin its traversal of the site and from which the agent is to resolve relative URLs found at the site. The OD also includes a series of one or more regular expressions used to describe the table-of-contents pages for the newspaper site. Finally, the site description includes a series of one or more regular expressions that describe article pages of interest along with information used by the agent to discern the text of an article versus the myriad of other information on the page (boilerplate, banners, advertisements, etc). Meta-information is maintained which includes the newspaper's name and the name of the collection under which VIPAR classifies the newspaper, as well as site-specific actions taken by the agents (e.g., includes the search depth limit [number of hops from the root URL, number of minutes to wait between rechecking the site for new articles, etc.).

Based on the RDF ontology, the information agent monitors and manages information at an Internet newspaper site. The agent checks each link found at a site against the ontological criteria to determine table-of-contents pages and article pages. If an article page of interest is found, the agent checks with the whiteboard agent to verify that the article was not already incorporated into the VIPAR system. If the article is indeed new, the agent reads the page, discerns clean article text (i.e., filters the raw text from the news article from the other nonessential/extraneous information on the page. The agent then marks up the clean text using XML, tagging the parts of the article (title, author, date, location, paragraphs, etc) depending on the site, and then posts the information to the VIPAR whiteboard agent. The agent continues to monitor the site, posting new information of interest as it becomes available. The VIPAR client is also an ORMAC agent that contains a graphical user interface. The client agent communicates with both the whiteboard agent and cluster agent to perform searches and clustering.

The whiteboard agent maintains all of the current articles, making sure there are no duplicates, and removes any articles that are beyond a given time period. The cluster agent subscribes to the whiteboard agent and thus is notified any time an article is added or removed from the whiteboard. When the cluster agent is notified of a new article (as discussed below), it examines the contents of the article and adjusts its search and clustering tables appropriately. Likewise, the tables are adjusted when the whiteboard removes an article.

3.4 Dynamic Information Clustering

There are two basic steps taken to organize the newspaper articles into clusters. The first step is to create a VSM from the articles. The basic premise of the VSM is that the newspaper articles and the significant words within each article are modeled as elements of a multi-dimensional vector space. Within this space, each significant word is represented by a new dimension, and a document is represented as a vector within this multidimensional space [2]. The value of each vector coordinate is an entropy-based function of "local" and "global" frequencies of the word corresponding to this dimension. The cluster agent maintains information containing the frequency of occurrence of terms within a document, called local term frequency, and over the entire set of documents, called global term frequency. These term frequency counts are then used to calculate a weight for each term in each document, which is called the document term weighting.

The second step is to create a similarity matrix that provides a pair wise comparison of each document in the system. We use the dot product (i.e., cosine of the angle between the vector pair) as the measure of similarity between two document vectors. This generates a global similarity matrix of size *n* x *n*, where n is the number of documents contained in the document collection. Only the upper triangular portion of this matrix is needed because it is a symmetric matrix. Note, when a document is added or removed the VSM *must* be updated. This is due to the changes in the global frequency of words that are contained in this document. The brute-force approach is to re-compute all the document vectors in the document collection (i.e., document term weights of each document vector) as well as a global similarity matrix. However, the time and space complexity of this task is $O(n \cdot d) + O(n^2)$, where *d* is the dimensionality of the document vector space. This process becomes prohibitively expensive when the collection size grows. A more intelligent approach is needed to effectively and efficiently update the similarity matrix. We use a sliding-window-based approach to perform the update of the similarity matrix.

The whole similarity matrix is modeled as a circular array of size $\frac{n(n-1)}{2}$ with a pointer initially pointing to the first element of the array. Whenever a new document is added or removed from the document collection the p percentage of the similarity matrix is updated and the pointer is forwarded $p \cdot \frac{n(n-1)}{2}$ steps from its current position, thus pointing to the next stale entry of the array.

We conducted a series of experiments to determine how changes in global term frequency would affect the similarity values. From these experiments, we determined that updating 5% of the global similarity matrix every time a single document is added or deleted preserves high accuracy. To compare similarity matrices, we used several measures based on the values of their determinants, traces, and x^2 distribution. In other words, it takes 20 document additions or removals to fully update the similarity matrix. This method resulted in acceptable dynamic similarity update performance.

Finally, the global similarity matrix is used to perform on-demand clustering of the documents of interest (e.g., the documents retrieved in response to a user query). For a set of documents to be clustered, the local similarity matrix is constructed by including the cells of the global similarity matrix that in turn corresponds to the documents of interest. This local similarity matrix is used to analyze the documents of interest based on their closeness in the document vector space. The documents are merged into clusters using an agglomerative hierarchical clustering algorithm, such as SLINK, CLINK, Ward's [10]. When all of the documents are combined, a Phylips Tree is generated to illustrate the hierarchical tree structure of the clustered documents. This cluster diagram is a type of dendrogram called a Phylips Tree. The nodes of the tree represent each article while the links between the nodes represents relationships. In general, the closer two nodes are, the more similar the articles. If links from two nodes share a vertex, then these articles are the closest in the set of articles. The longer the links between article nodes, the greater the dissimilarity is between the articles.

4 Results

Results obtained from a demonstration characterize the operational environment, and organizational capability.

4.1 Nuclear threat example

This one-year demonstration project was completed March 2002 and delivered four iterations of software, each requiring roughly 3 months to develop. The final system included thirteen information agents that manage thirteen newspaper sites including: (1) Asahi Shimbun, (2) Asia Times, (3) BBC, (4) Japan Times Online, (5) Japan Update, (6) Korea Times, (7) Manila Times, (8) Pacific Islands Report, (9) Sydney Morning Herald, (10) Taipei Times, (11) The Hindu, (12) The Star, and (13) Times of India. *We verified that only current news articles are gathered from each news source, and that news articles are not lost or mischaracterized by the information agents.*

4.1.1 Organization capability

The organization of the information using the VIPAR system was very successful. In an experiment comparing the organization of news articles done manually, versus organized by VIPAR, results favor VIPAR as the preferred method. The experiment involved searching a collection of newspapers for key terms that produced a number of relevant news articles. This collection of articles was then manually organized based on the contents of the articles. Following this manual process, VIPAR was then used to organize the same set of articles, and the results from both methods were compared. For example, one experiment was performed on September 21, 2001, searching on the phrase "nuclear weapons." At the time, five newspapers were in the VIPAR system, the Japan Times, the Pacific Islands Report, Inside China Today, Russia Today, and the Sydney Morning Herald. The results of this search produced 10 articles, with various titles (see text box).

1. Wen Ho Lee Spends First Day Savoring Home Delights
2. Lee Case Points up Scientists' Attitude on Security
3. India and Pakistan: Troubled relations
4. U.S. China Trade Vote Milestone on Rocky Road
5. Troops die in Kashmir clashes
6. Clinton Concerned Over Lee Case - Reno On Defensive
7. IAEA Supports Putin Nuclear Power Initiative
8. China Rejects Moves to Tighten Regulation of Nuclear Materials
9. Clinton Calls For Review Of Lee Secrets Case

These results are what you would typically expect from an average search engine, except that the information within VIPAR is targeted just to newspapers, is more up to date, and covers only a few days time span. In manually organizing these articles, the objective is to put similar articles into the same category. The articles in this collection cover four broad areas, 1) the Los Alamos Nuclear Scientist Wen Ho Lee, 2) the India and Pakistan conflict spurring nuclear weapons development, 3) an International Atomic Energy Agency meeting dealing with nuclear material, and 4) U.S. China Trade Policy dealing with nuclear material. To manually organize a small number of articles like these can be done fairly quickly by a knowledgeable person. However, as the number of articles increases so does the time required to manually organize the articles.

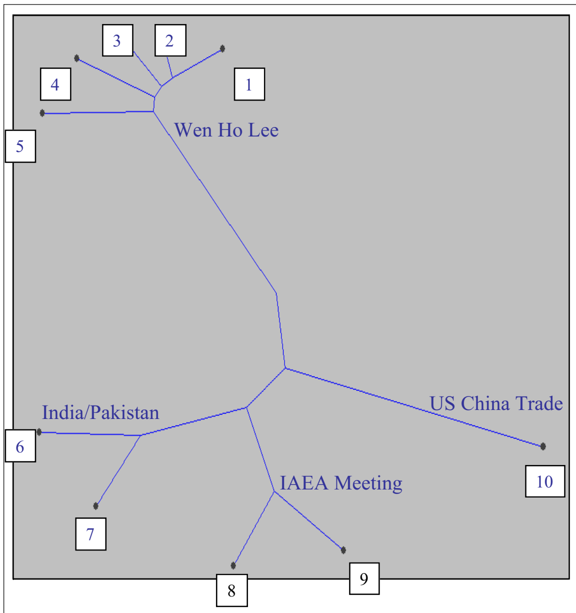
VIPAR was used to cluster the articles within only a few seconds, and produced 4 distinct groupings. Fig. 3 shows a comparison of the VIPAR cluster to the manual clustering. The four groups determined by VIPAR match extremely well to the four groups of articles manually organized. The VIPAR clusters provide an intuitive (i.e., natural, quick and effective) way to organize/visualize this information.

4.1.2 Discussion

The VIPAR tool has demonstrated that it can self-organize a very large number of articles in a comparable manner to humans (this example uses only a few articles for the purpose of brevity/clarity). This is a very significant accomplishment. It is very useful in small cases like the experiment described above, but is absolutely essential in dealing with larger volumes of information where VIPAR's advantage as compared to manual approaches are commanding. Moreover, VIPAR's ability to organize a large amounts of quickly is complemented by the graphical results as seen in the Phylips graph. When comparing this approach to traditional search engines the most significant value is easily discernable. Search engines are capable of producing large volumes of documents in a list. A great deal of time and energy is often spent traversing through such a list. The VIPAR approach organizes the list, and presents it in graphical form to the user. This allows a user to look at one picture from a query and understand the results as opposed to cycling through a large list of documents.

VIPAR is based on a number of significant research contributions. The three most noteworthy include the: 1) ability for information agents to use a flexible RDF ontology for managing newspaper information. 2) ability to dynamically add and cluster new information entering the system. 3) ability to graphically represent the organization of information so that it is easily understandable (intuitive). The use of these contributions has provided the ability to greatly enhance the way that Internet information is searched and analyzed.

A fundamental aspect of this project is the value of using software agents as the basis for the solution. Clearly any number of technologies could be used to solve this problem, however, software agents provide three key architectural advantages that we believe improved the overall solution. Agents have the ability to communicate at a higher level, and in a more general manner then do traditional objects. This provides generality among the information agents and the whiteboard agents. Agents can move, fail, or ignore messages, yet the system easily continues. Agents are typically designed to derive information from ontologies. This concept allow for rules and information to be easily changed, without having to rewrite software. This was very important for the information agents. During this project, a few of the newspaper formats we were using changed without warning. These changes were quickly fixed by simply adjusting the RDF ontologies. Another benefit we observed with agents was the ability to move from one system to another. This concept allows us to rapidly adapt to changes in hardware configurations, load balance work, and adjust the priority of agents. The drawback with this mobility is that the agents must work within a trusted environment, or be restricted to being little more than an applet. We found the agent concept very well suited to this problem. There are several limitations of agent technology however, the first being the temptation to build a simple object system and call the objects agents. At this stage in the evolution of agent technology, this appears to be a common problem. The second limitation is a fairly steep learning curve about what agents are, and how they are developed. There are a number of



- Wen Ho Lee
  1. Wen Ho Lee Spends First Day Savoring Home Delights
  2. Clinton Calls For Review Of Lee Secrets Case
  3. Clinton Concerned Over Lee Case - Reno On Defensive
  4. Asian-Americans Demand White House Action on Lee
  5. Lee Case Points up Scientists' Attitude on Security
- India and Pakistan
  6. India and Pakistan: Troubled relations
  7. Troops die in Kashmir clashes
- IAEA Meeting
  8. IAEA Supports Putin Nuclear Power Initiative
  9. China Rejects Moves to Tighten Regulation of Nuclear Materials
- U.S. China Trade
  10. U.S. China Trade Vote Milestone on Rocky Road

Figure 3 Comparing VIPAR article organization versus manual organization.

frameworks; however, which frameworks are suitable for production use is often unclear. Lastly, there are some modest efforts in standardization, however, without broad agreement on agent protocol, ontologies, and trust in the network, the full potential of this technology may never be reached.

The VIPAR project exceeded our own expectations with respect to maintainability. The RDF ontologies are a natural means of representing Internet sites. Surprisingly, they appear to be quite easy for non-programmers to extend and modify. This allows great flexibility and extensibility in the VIPAR system. In a matter of days, users can learn to add or modify newspaper sites that may be of interest and to be monitored by the system.

We were also surprised by the memory limits of the system. The system maintains the VSM resident in memory. As the number of articles increases, it appears that the VSM is causing the paging system to thrash. If this is indeed the case, then the obvious solution is to maintain the VSM in disk storage, which will cause a performance penalty for small clusters, but will provide the ability to cluster far larger collections of information. We have also developed several approaches to reducing the complexity of VSM and clustering algorithms. These approaches are beyond the scope of this paper.

5 Conclusion

The VIC problem involved gathering/analyzing more information than could be reasonably accomplished given the available manpower (common dilemma that promises to worsen). To address this challenge, the multi-agent VIPAR system was developed which uses software agents to retrieve, organize, and graphically present Internet-based newspaper information to analysts. The system organizes articles in a fashion comparable to that of intelligence analysts. VIPAR extends the field of agent technology through the use of a flexible RDF ontology for managing information including the capability to dynamically add and cluster new information entering the system.

Agent technology is well suited to this type of problem for three main reasons. First, the communication mechanism allows for broadcast *and* peer-to-peer message passing. Second, using an external ontology allows for an easily maintainable and/or replaceable mechanism for adapting to an open/changing information environment and rules (intelligence needs). Consequently, agents can be redirected without the need to modify code. Finally, agents are mobile, a natural solution to the needs of intelligence gathering. The ability for agents to suspend operations, move to another computer, and resume operations on command provides for various design/implementation options for rapid deployment. There are some drawbacks with agent technology, including a steep learning curve and little standardization, however, such limitations are to be expected with an emerging technology.

The delivered system included thirteen information agents that manage thirteen different newspaper sites. Since delivery of the system, the VIC staff has added additional newspapers to the system by creating new RDFs.

6 References

1. Potok T.E., Mark T. Elmore, and Ivezic N.A. "Collaborative Management Environment" Proc. InForum'99, http://www.doe.gov/inforum99/proceed.html
2. Samatova N.F., Potok T.E., and Leuze M., "A VECTOR PERTURBATION APPROACH TO THE GENERALIZED AIRCRAFT SPARE PARTS GROUPING PROBLEM" accepted for publication in Int'l Jr. of Flexible Automation and Integrated Manufacturing.
3. Sycara K., Decker K., Pannu A., Williamson M., and Zeng D., "Distributed Intelligent Agents", *IEEE Expert* 11(6), 36-46 (1996).
4. Mladenic D. "Text-Learning and Related Intelligent Agents: A Survey", *IEEE Intelligent Systems*, 14(4), pp. 44-54 (1999).
5. Kilander F. "A brief comparison of news filtering software", http://www.dsv.su.se/¸fk., (1996).
6. Klusch M. "Information agent technology for the Internet: A survey", Data & Knowledge Engrng 36, pp. 337-372 (2001).
7. Jain A.K., Murty M.N., and Flynn P.J., "Data Clustering: A Review", ACM Comp. Surveys, 31(3), pp. 264- 323 (1999).
8. Potok, T.E., Ivezic N.A, and Samatova, N.F., "Agent-based architecture for flexible lean cell design, analysis, and evaluation," Proc. 4th Design of Information Infrastructure Systems for Manufacturing Conf., (2000).
9. Ivezic N.A., Potok T.E., Pouchard L., Manufacturing Multiagent Framework for Transitional Operations. *IEEE Internet Computing*, 3 (5), 58-59, (1999).
10. Anderberg, M.R., *Cluster analysis and applications,* Academic Press, (1973).
11. Potok, T.E., Elmore, M.T., Reed, J. W. and Samatova, N. F. "An Ontology-based HTML to XML Conversion Using Intelligent Agents", Proc. 36th Hawaii Int'l Conf. On System Sciences, pp.120b-130b, January (2002).