Autonomous Agents Design for Digital Network Maximization in Joint C4I Systems

Michael W. DaBose, Luqi

Abstract—The advent of the computer age has brought about an ever increasing demand to transfer exponentially increasing amounts of information, and the associated problems of information sharing. The focus of this paper is to best utilize available digital communications assets in a constrained radio frequency (RF) spectrum to allow sufficient transfer of information providing Department of Defense (DOD) assets flexible, rapid, and in-flight reprogramming, re-planning of strike and cruise missile assets, to engage a high value, emergent target, in the shortest possible time. The postulated methods of utilizing autonomous agent type methods to manage information flow across network nodes has applicability to all digital networks.

Inspired by the pioneering work of Pattie Maes, at Massachusetts Institute of Technology (MIT), and previous examination of communications node management, the implementation of independent processes, working on behalf of a host system, to optimize the effective meaningful throughput on a communications channel is not only desirable, but necessary. The evolution of semi intelligent software, whether called Artificial Intelligence, Intelligent Agents, or Autonomous Agents, has reached a level of sophistication allowing the insertion of meaningful articulated processes within existing, and future systems to maximize the network efficiency systematically. Recent work by Michael Cohen on Sodabots, and the evolution of user interactive TinyMUDS of the Maas-Neotek family, a virtual type personality environment, has demonstrated the ability of software to deal with dynamic and changing conditions. The exponential increase in micro-processor power has, for the first time, made available the hardware for such agent implementations as compact, self contained, embedded systems, in direct support of larger existing systems.

Index Terms—digital network maximization, compression methods, embedded maximization, bandwidth optimization, agents

I. Introduction

An important problem arising from the increased sharing of information across networks is bandwidth constraint. Then limitations of communications channels in the transmission of voluminous information is the singular bottleneck dictating processing capability and robustness of current and future distributed systems. Bandwidth utilization with the goal of optimizing the actual information transmitted, has to date, been largely ignored. Many of the current network strategies, both commercial and DOD specific, rely on the repeated broadcast of a standardized message. As a result, much available bandwidth is wasted. The specific approach taken to maximize specific network node throughput on a digital network is a three-layer paradigm, managed by an embedded autonomous semi-"intelligent" software "agent" located at each network node. For the purpose of this research effort, agent is defined as semi-intelligent embedded software, acting on behalf of an attached system. The first layer consists of a network specific strategy for reducing the message content. The second layer is a frame by frame analysis of the reduced message content, to determine the best compression method to be applied to the information itself (MPEG, Lempel-Ziv, etc.). Finally a packaging strategy is utilized to maximize the compressed content for each specific network packet. The first phase, consisting of a proof of concept prototype has been implemented. Initial results, via a network simulation, have demonstrated a quantative 300% plus increase in effective information throughput capability, on the data link, utilizing the same bandwidth. Since this approach is an embedded technique, existing network hardware, software, and standards remain unaffected. An observed positive effect has been greater network responsiveness, over time, as measured by increased critical information transfer.

This research is not to develop compression technology. There are numerous techniques available for a multitude of media types. Instead this research employs methodology to bring to bear the best capability, within a real time scenario, in this case a responsiveness granularity of $1/10^{th}$ to $1/100^{th}$ of a millisecond, to dynamically maximize throughput potential. For the purpose of this paper, these collective processes will be referred to as Link Maximizer.

II. Network Specific Strategy for Reducing Message Content

Many digital networks have message standards as part of the interprocess communication protocol. In the specific instance of the network maximization effort, the DOD standard Link16 tactical digital network was utilized as the

M.W. DaBose is with the Space and Naval Warfare Systems Center, San Diego, Ca 92152 dabose@spawar.navy.mil

Luqi is with the Computer Science Department, Naval Postgraduate School, Monterey, Ca. 93943 luqi@cs.nps.navy.mil

reference. Link 16 is based upon a modification of the Slotted Aloha broadcast method. Specifically, individual network nodes have assigned transmit and listen slots, which when combined define an epoch, which repeats over a designated period. The message standard of Link16 is called TADIL J.

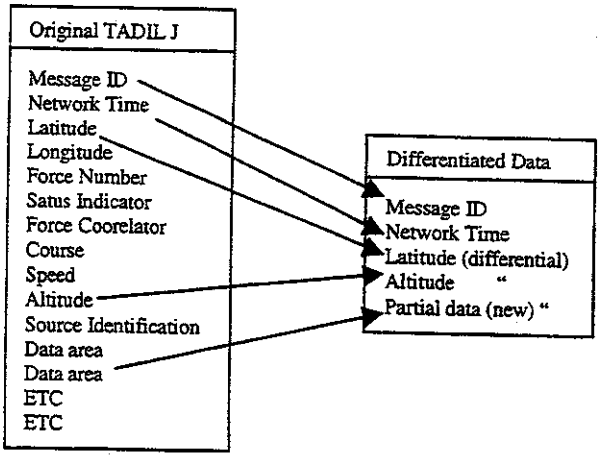

Fig 1.Complete v.s. changed information

Given the periodic transmission of information, the following assumptions can be made:

1. Receiving stations on the network will have the historical or latest information available relating to this instance of a message type.
2. Specific instances of information, relating to the message type, can be differentiated from the historical record.
3. It is only necessary to transmit the new information. The complete "new" record can be reconstructed through "differentiating", then "re-integrating" the old historical record to the new information.
4. It will be possible to generate an embedded method for keeping track of differentiated messages. This will insure proper ordering, and complete information.

Message differentiation or "delta" messaging can, and has been implemented by two distinct techniques within the constraints of Link 16. The first method involves a field by field comparison of specific values within a message instance. The second method is a binary differentiation of the message instance against the historical record. The difference of either method is then transmitted to the receiving network nodes, where either the field by field comparison or the binary differentiation can be made to generate an exact copy of the new message of a specific type.

III. Frame by Frame Analysis to Reduce Message Content

Compression techniques fall into three broad instances of utilization, dependent upon their optimization strategy. Text, general binary, and multimedia compression comprise the domains upon which the analysis of information will be based. Given the assumption the available techniques are useful, the focus is upon the ability to distinguish the best method to be employed on a dynamic basis. This method is specific to Link 16, however, as will be noted later, other method(s) to further reduce message size can be invoked.

A. Link Maximization Methods for Link 16

For this project, several techniques are under development for reducing both the length of frequently transmitted messages and the frequency of transmissions. These techniques are lossless in that there is no concomitant reduction in content or user situation understanding. The benchmarks will reveal, for each technique, the best possible gain in virtual bandwidth.

Simultaneously, Link Maximizer will be designed to further reduce bandwidth by adding intelligent, distributed link control, as well as enhance the message length-reduction strategies with such techniques as transmission frequency reduction. Software agents have been shown to be especially useful for network management in other contexts. The rules of the link will be instantiated in the Link Maximizer, which, along with an automated transmission planning capability, will provide substantial gains in effective bandwidth.

The concept of Link Maximizer is not in violation of the ISO/OSI Network Model, rather this process is embedded within a layer of the model. The primary advantage of embedding is the seamless support of current architectural standards, resulting in no current system modifications.

ISO / OSI Model (7 Layer)
7) Application
6) Presentation
5) Session
4) Transport
> *Link Maximizer* <
3) Network
2) Data Link
1) Physical

B. Atomic Data Element Transmission ("Delta Messages")

Redundant elements of messages (e.g., track update messages) will be reduced to a minimum. In general, only the elements of a message that have changed since the last update will be transmitted, together with identifying information. The goal is to provide a decrease of one-third to two-thirds bandwidth requirements for track messages, in addition to only transmitting such Delta Messages when new information deviates outside expected parameters as determined by relative navigation models, see Extrapolation-Driven Updates, below. The effective reductions would make needed bandwidth available for other purposes, and increase the links overall track handling

capabilities through being able to identify and update a greater number of tracks.

Message objects with strong real-time requirements will be cached in RAM, or other strategies developed, to enable the system to keep up with the link requirements.

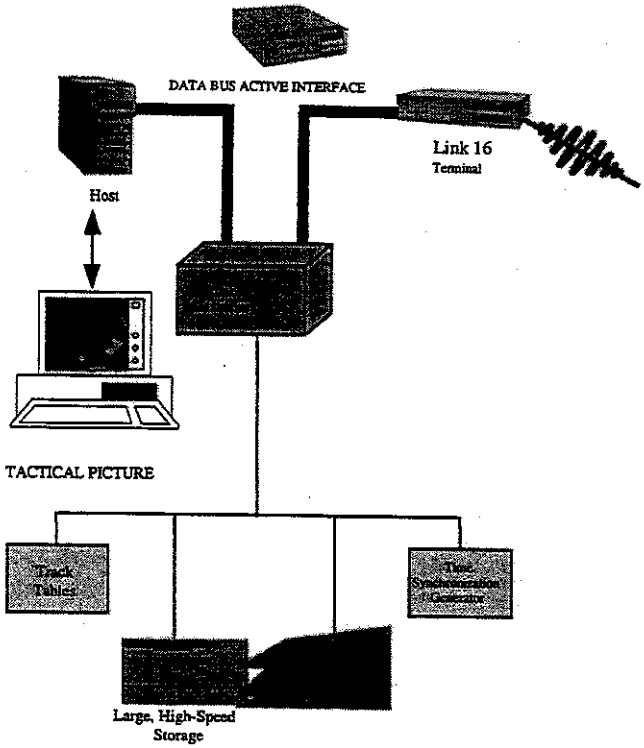

Figure 2, Physical Implementation

Figure 2 provides a conceptual view of delta message reduction. Although the TADIL message type is "Free Text," a message type local to the Link Maximizer will be embedded within the message, for each type of message it transmits. Following the message type is an indicator of the number of fields present for that particular message type. After that is a variable number of fields with field types (i.e. integer, character, binary, etc.) associated with the message type. Although most fields are fixed-length, some will be variable. For example, with respect to latitude, although a full update message has new data, probably only a very few of the least significant bits from the previous update have changed. The Link Maximizer will identify and insert the changed bits, and the field length will be the number of them. The receiving Link Maximizer will know to replace the lower order bits from the previous message with the new ones. Of course, if the whole field changes, then the whole field will be sent. It is understood that if there is significant change in the original TADIL message, the delta message could be significantly longer than the original message. The implication is that the delta message is checked against the original message size. Only the shorter of the two is sent.

This kind of severance system can be introduced and has already been proven with the Link 16 virtual gateway, which today provides both a passive tap, for relaying the tactical picture to other systems, and the ability to insert messages directly.

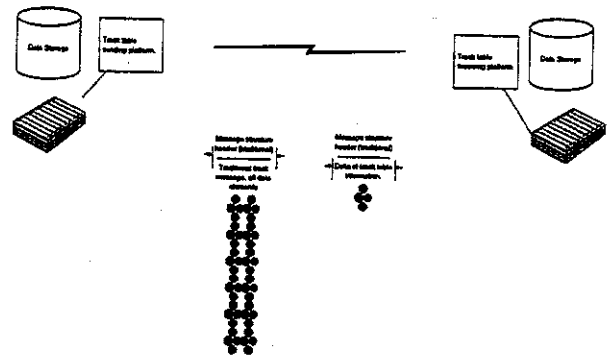

Figure 3, Conceptual Implementation

One challenge, at present, is that track update messages belong to the surveillance Network Participation Group (NPG), which is connected to one of only three existing JTIDS buffers. The buffer is there to ensure that all messages of the surveillance type will be transmitted (in overload conditions, a message not buffered can be tacitly dropped from the system).The command host processor manages itself based upon the available buffer size. At present, free-text messages do not belong to the surveillance NPG. This poses no significant problem for the benchmark demonstrations because there is an operator-assignable buffer that can be used.

C. *Update Bundling*

Not all frequently transmitted messages are of high priority. The priority of a given type of message may vary with the situation. Delta messages, or other types of messages, can be bundled into a single transmission using the free-text message format. This would make, for example, the transmission of historical track data feasible (by encapsulating a full message and all of the subsequent updates into a single free-text message). Because all of the information is in a single message, rather than in many messages, the system may be able to process the information more efficiently (that is, the total package of information is received more quickly, and less administrative load on the system).

To minimize this overhead, another function might be to "bundle" a historical message with all its subsequent delta messages, so that a platform entering the network could be updated on all of the data concerning a particular track in a single, free-form transmission of minimal size.

In today's system, it is a goal that all track updates be sent within a specified predetermined period of time. This is true for slowly-moving objects, such as ships, and faster-moving objects, such as airplanes. Resources permitting, delta messages from slower-moving objects would be "packed" as in figure 3 above, for transmission in a single message

after a fixed interval (TBD). At the receiving end, all of the updates would be applied by Link Maximizer, with the resulting full update message being sent on to the attached host processor.

It is important to validate bundling as a concept. The total network load factor of sending multiple small messages must be compared to the load factor from sending one, larger message, within the context of a valid simulation. Network performance-measuring tools currently being developed by others will be used to measure the expected gain. Since it is estimated that network bookkeeping amounts to some twenty percent of net loading, the gain promises to be significant.

D. Extrapolation-Driven Updates

In this technique, each type of vehicle being tracked is modeled on every platform, utilizing a flat Earth 3-D model (dead reckoning). Each platform makes a projection of the tracked-vehicle's position at its next-scheduled update cycle, based on observed behavior. If the predicted position is close enough to the actual position, then either no transmission is made and the individual local Link Maximizer generate the update, or a very small transmission is made confirming that the prediction is accurate. If too great a deviation has occurred, then a full update message or a delta message is actually transmitted. Since most tracked vehicles behave predictably over short intervals, this technique promises to greatly reduce network loading—with no loss of information at the receiving platforms.

At any given time, one platform has the reporting responsibility for a given track (based on track quality). The concept of extrapolation-driven updates is to have a set of vehicle simulations at each platform that predict the position of the tracked vehicle at the next scheduled update. If the prediction of the update matches the reported "true" position, within an appropriate tolerance (calculation based upon range and sensor resolution), no update would be sent. Instead, Link Maximizer would internally generate an update message based on its prediction and send that update to the attached host processor. The concept is illustrated in Figure 4.

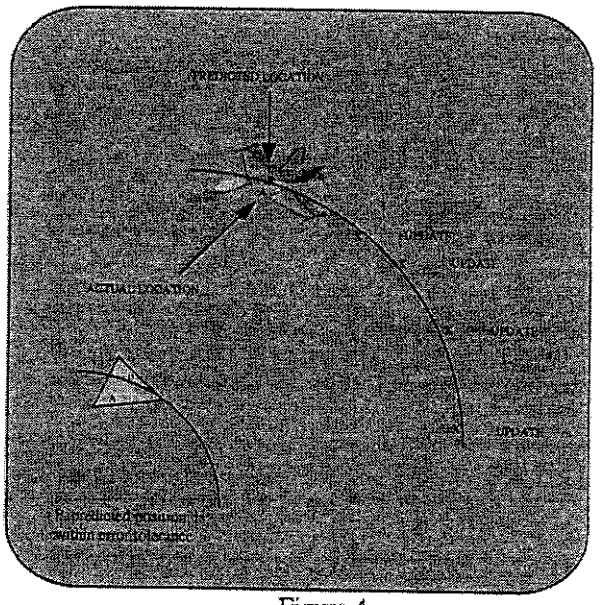

Figure 4

The shape and size of the error tolerance is determined by the vehicle type, elapsed time, and the accuracy of the sighting information. For example, if the gaussian circular-error-probable (CEP) of the sighting is known, the circular coverage function can be used to determine if the sighting is within a particular level of assurance of a circular tolerance shape. In the process, the total probability distribution within the circle is integrated. If, say, a ninety percent assurance is required, then ninety percent of the distribution must lie within the circle. If so, no message is sent. An automatic curve fitting mechanism then adjusts to the new position, recalculates the error shape, and continues. If not, the actual position is sent and the curve fitting adjusted.

IV. Message Packaging Strategy

A significant objective of this effort is to provide compressed packaging, such that compressed / differentiated information could be assembled in such a manner as to allow transmission via established methods, i.e. Link 16 hardware and software. Two message types exist within the message standard which support this effort. Both the Free Text Message, and Variable Format message provide a structure suitable to wrap binary, compressed information within a standardized message header.

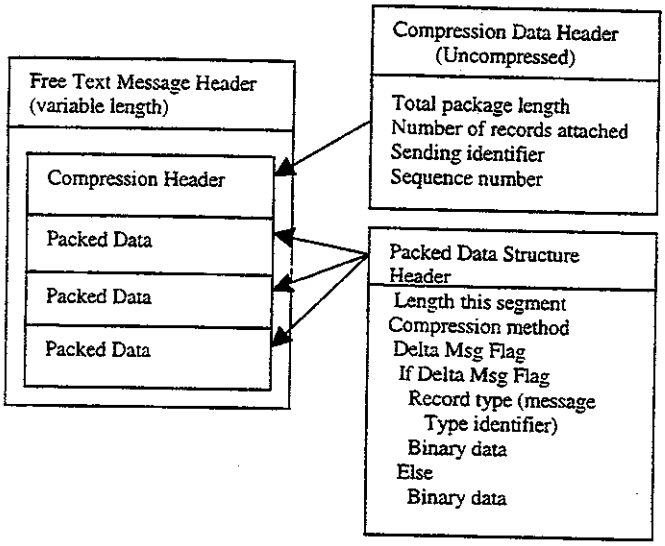

Fig 5. Compressed package schema

The individual message type packages (text, binary, MPEG / Wavelet, audio) are bundled into logical lengths consistent with the target network. For example, if the least packet size for the target network is 1500 bytes/packet, then the optimum compressed package size is:

1500 bytes >= packet size = (packet header + Free Text Message)

The specific field identifiers have combined goals of providing a variable length record structure, so no space is wasted, the second goal is to minimize the header information size by utilizing bits field delimitations.

V. FITTING THE PIECES TOGETHER

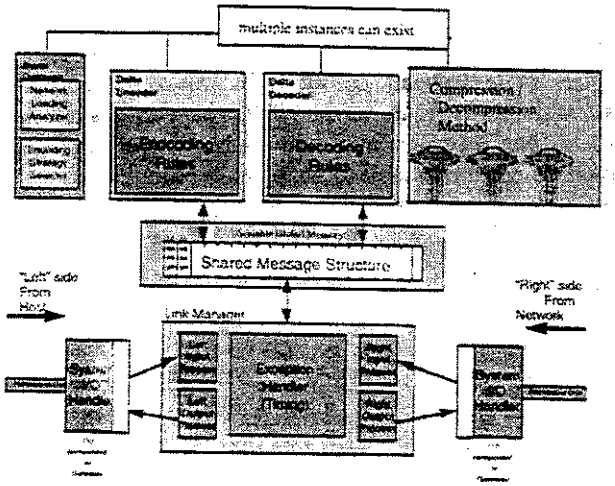

Figure 6. Network maximization implementation

The Link Maximizer enforces the time rules of the network, as well as performing housekeeping functions for itself. If only the Shared Message Structure and the Link Maximizer existed, the network into which Link Maximizer was inserted would function normally. Messages would be intercepted from the left (host) or right (terminal), be placed into the structure, age, and then be asserted on the output network or bus in the correct direction (that is, to the host or to the terminal).

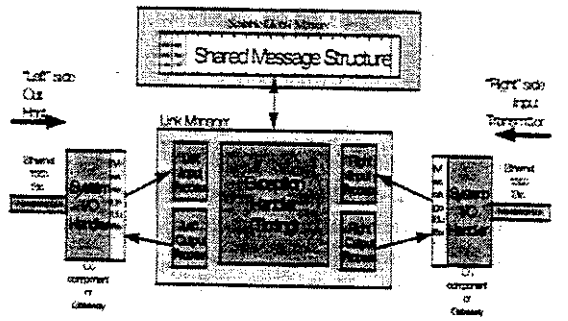

Figure 7, Digital Link Maximizer

At the physical level, this is a severance implementation. That is, the network or bus between the host and the terminal are physically severed, and the Link Maximizer is inserted at the severance point. The software functions by interception. Messages coming in either direction are intercepted by the Link Maximizer, processed, and then asserted back on the network or bus. The messages may be asserted in their original form or in Link Maximizer format. If repetition rate reduction is in effect, outgoing messages (that is, messages from the host to the terminal) can be deleted by the Link Maximizer.

Each of the components of the Link Maximizer is a separate process. They may very well be implemented on separate processors, in order to meet run time requirements. They all operate asynchronously, communicating indirectly through the Shared Message Structure.

It is assumed that there is a fixed-length message header in all outgoing messages. It is assumed that the type of message is in this header as well as in the message itself.

The time the message was originated is presumed to be in the header. There is also a field in the header that indicates that an outgoing message must be acknowledged. However, this field is ignored at this time because none of the types processed require acknowledgment. If a type is at some point handled that does require acknowledgment, it can simply be passed through as is (it is presumed that such messages are a rare occurrence).

For simplification, each J-Series message is presumed to be the same length. Each contains the following fixed-length fields:

- •Message Type
- •Track Number
- •Update Time
- •Latitude
- •Longitude

Of course, real messages have considerably more fields than this. However, they would be handled in precisely the same way as the identified fields. It is presumed that the field positions within the three update types of interest are identical.

The "Update Time" field refers to the time of the observation or update, not the time the message was originated or the current time. It is also presumed that the time in a series of these updates is predictable plus or minus a small delta. Therefore a Link Maximizer, looking at the last few update messages, can determine if it has a complete recent sequence from these times.

VI. FUTURE WORK

Specifics regarding the implementation of Link Maximizer with respect to Link 16, are easily circumvented, or modified to accommodate new network architectures. The generic ideas expressed represent a software approach to maximize the efficiency of any digital network, within a predefined bandwidth. Future work envisions utilizing the architecture of the Link Maximizer, as a network node "attachment". Through the utilization of the "hooks" provided by this method, the next level of network(s) control, maximization, and redundancy is possible. The existing bonds between host systems, and their attached communications assets can be severed. The next step architecture is to provide a communications management architecture, capable of directing information, independent of traditional channels, based upon urgency, security, and available communications assets. The end result of this process, is an information processing infra-structure, independent of the means of communications, providing a high degree of redundancy, survivability, flexibility, legacy system support, while opening avenues of connectivity to new network, and processing system assets.

VII. REFERENCES

[1] Virdhagriswaran, Sankar, *Heterogeneous Information Systems Integration*. Crystaliz Inc. paper September 29, 1994

[2] Foner, Leonard N, *Clustering and Information Sharing in an Ecology of Cooperating Agents* Agents Group, MIT Media Lab. E15-305, 20 Ames St., Cambridge, MA 02139. paper 1994-5.

[3] Coriat, Michel, *Formal Specification Using Agents Conceptualization* Laboratoire MASI, Institut Blaise Pascal / CNRS-UA 818, Universite de PARIS VI, 4 place Jussieu 75252 Paris cedex 05, France. paper 1995.

[4] Foner, Leonard N., *What's an Agent Anyway? A sociological Case Study*. Agents Group, MIT Media Lab, paper Agent Memo 93-01, 1993.

[5] Maes, Pattie, *Modeling Adaptive Autonomous Agents*. MIT Media-Labatory, 20 Ames Street, Rm 305, Cambridge, MA. paper 1994.

[6] Coen, Michael H., *SodaBot: Agent Environment and Construction System*. MIT Artificial Intelligence Labatory, Cambridge, MA. paper September 14, 1994.

[7] Tambe, Milind, et al. *Building Believable Agents for Simulation Environments: Extended Abstract*. Information Sciences Institute, University of Southern California and Artificial Intelligence Laboratory, University of Michigan, paper 1994.

[8] Kotay, Keith D., et al. *Transportable Agents*. Department of Computer Science, Dartmouth College, paper November 10, 1994.

[9] Finin, Tim, et al. *DRAFT Specification of the KQML Agent-Communication Language*. The DARPA Knowledge Sharing Initiative External Interfaces Working Group, paper June 15, 1993.

[10] Moukas, Alexandros, *Amalthaea: Information Discovery and Filtering using a Multiagent Evolving Ecosystem*. MIT Media Laboratory, Cambridge, MA, paper 1995.

[11] Thirunavukkarasu, Chelliah, et al. *Secret Agents - A Security Architecture for the KQML Agent Communication Language*. Enterprise Integration Technologies, Computer Science and Electrical Engineering, University of Maryland, paper December 1995.

[12] Genesereth, Michael R, et al. *A Distributed and Anonymous Knowledge Sharing Approach to Software Interoperation*. Computer Science Department, Stanford University, paper November 15, 1994.

[13] Maes, Pattie, et al. *Kasbah: An Agent Marketplace for Buying and Selling Goods*. MIT Media Lab, paper 1996.

[14] Pitt, Jeremy, et al. *Autonomous Agents in Inter-Organization Project Management*. Department of Computing, Imperial College of Science, Technology, and Medicine, UK, paper 1995.

[15] Maes, Pattie, *Intelligent Software; Programs that can act independently will ease the burdens that computers put on people*. paper 1995. Massachusetts Institute of Technology

[16] Franklin Stan and Graesser Art, *Is it an Agent, or just a Program? A taxonomy for Autonomous Agents*. 1996. Proceedings of the Third International Workshop on Agent Theories Architectures, and Languages, Springer-Verlag, 1996

[17] Maes Pattie, *Artificial Life Entertainment: Lifelike Autonomous Agents*. paper 1996. Massachusetts Institute of Technology

[18] Hayes-Roth Barbara, et. al. *A Domain-Specific Software Architecture for Adaptive Intelligent Systems*. April 1995. IEEE Transactions on Software Engineering, Vol. 21, No. 4, pp288-301

[19] DaBose, Michael W., *Tactical Link Object Oriented Technology Insertion*. paper 1995. Naval Research Development Test and Evaluation NRaD

[20] DaBose, Michael W., *Tactical Link Object Oriented Technology Insertion*, paper 1996.

[21] Chavez, Anthony, et al. *Challenger: A Multi-agent System for Distributed Resource Allocation*. Autonomous Agents Group, MIT Lab, paper 1996.

[22] Durham, Jayson and Torrez, William, *A Neuron Model Using Cumulative Distribution Functions*, Proc. Of The World Congress on Neural Networks, Portland, OR.

[23] Durham, Jayson, Gillcrist, Brenda, and Heckman, Paul, *A Testbed Processor for Embedded Multi-Computing*, Proc. Of The 6th International Symposium on Unmanned Untethered Submersible Technology, Washington, DC.

[24] Durham, Jayson, *Engineering Intelligent Undersea Vehicles*, Proc. of OCEANS'89, September 18-21.

[25] Maes, Pattie, *Intelligent Software*, Scientific American, 1995

[26] Maes, Pattie, *Artificial Life Meets Entertainment: Lifelike Autonomous Agents*, CACM, 1995

[27] Shannon, Claude, *Mathematical Theory of Communication*, 1948