Towards a Theory for a Sociable Software Architecture

Emmett Davis
(651) 699-4367
EmmettDa@aol.com
University of Saint Thomas
Graduate Programs in Software Mail # OSS 301
2115 Summit Avenue
Saint Paul, MN 55105

Bonnie Holte Bennett
(651) 962-5509
bhbennett@stthomas.edu
University of Saint Thomas
Graduate Programs in Software Mail # OSS 301
2115 Summit Avenue
Saint Paul, MN 55105

Abstract

The complex relationship of networks of information that exist on the web and in modern computer systems requires advanced link-analysis capabilities in order to make sense of the deluge of data. Applications are emerging which use "intelligent agents" to gather data, however, they are rarely more than specially tuned search engines with the goal of gathering individual's purchasing and chat proclivities. We propose a link-analysis architecture focused on the individual with attention on the individual's "discourse communities." We believe this focus, along with the development and refinement of a discourse community thesaurus can uniquely begin to model the social and cultural component's of a user's profile. This profile can help to begin to create more truly intelligent agents performing richer link analysis.

Introduction

In order for advanced information systems to be truly intelligent, they will need a social and cultural context. We will need to understand and articulate the social and cultural environments of each information system. In the past information systems were isolated databases and the extent of our architectural theory was normalization. Then we were only plumbers linking pipes without regard to content or on the opposite extreme working in only one domain.

Future systems will need to interact with a variety of foreign systems. While object-oriented is a step forward in dealing with these challenges, we need to be clear about our philosophy relating to knowledge in a variety of types of systems ranging from databases to artificial intelligence and information retrieval.

Beyond object oriented programming, information and knowledge require an explicit understanding of their cultural and social environment from which existing data and information comes. We must articulate how information is used as knowledge and how it is increased or destroyed. For each information system we must understand: 1. its scientific and academic discipline, industry, or discourse community; 2. the roles of actors in discourse communities; and 3. meaning from its intellectual dynamics over time. The representation of these provides the basis for advanced, intelligent link analysis.

The next section describes the technological and commercial pressures on advanced information systems. Section III describes the theoretical tenants underlying our work. Section IV explores the benefits we see in pursuing this approach. Section V describes an approach to implementing this model. Section VI describes some mechanisms in such an implementation.

Motivation

This section describes a number of technological and commercial factors which are putting extreme demands on advanced information systems.

Business tempo is accelerating. Exponential information growth inundates us daily. We expect to interact with smart systems that learn our preferences and meet our specific needs. We hope to "manage by wire" where we communicate our intentions and our agents "make it so." The "value chain" is identified as an abbreviated business cycle in which highly customized goods arrive immediately as the consumer feels the desire for the product (with, ideally no time in the wholesaler's or the retailer's hands, and thus, no inventory costs). As organizations grow, they seek the optimal balance between distributed operations and economies of scale. Agents abound on the web, but most seem to exist to invade our privacy, not maintain is as all the predictions of the electronic "Jeeves" have predicted. All of these changes are dramatic and challenging for the information systems they use, and they provide the motivation for sociable software architecture.

Exponential Information Growth Inundates Us. Tractability, the combinatoric impact of large numbers of interacting entities, increases the computational resources needed to complete a task. "The challenge posed to businesses by the soon-to-be obscene amount of information is staggering. Yet the promise is fabulous as well, fueling hopes that knowledge management will finally come of age." (Bovet and Sheffi 1998). While Bovet and Sheffi are focusing on logistics, other examples, many web and agent related, easily come to mind. Our desires to create space stations, anticipate the weather, and predict geological cataclysms are instances of our need to understand "large numbers of interacting entities."

Smart Systems Become an Exponential Force Multiplier. (Davis and Botkin 1994) describe smart objects,"Products are smart because they filter and interpret information to enable the use to act more effectively. ... They are interactive, they become smarter the more you use them, and they can be customized." (Davis and Botkin 1994). "The more you use knowledge-based offerings, the smarter *they* get." A knowledge-based system will learn more and put the new information to productive use. The Ritz-Carlton hotel uses a customer preference tracking system. All the hotels learn if a customer expresses a preference for six hypoallergenic pillows at one of the hotels. "The more you use knowledge-based offerings, the smarter *you* get." An example would be tutor systems that become group enhanced decision support systems. "Knowledge-based products and services adjust to changing circumstances." Not only do smart products tell you something is wrong, they also "tell you what to do." (Davis and Botkin 1994). David and Botkin recognize that, "knowledge-based businesses can customize their offerings." Smart systems will allow large organizations and alliances of organizations to ultra customize their services and products. Smart systems act like compound interest, an exponential force multiplier with growth.

Management by Wire: Carry Out my Intentions. In aviation "flying by wire" is a term that indicates that computer systems are used to augment a pilot's ability to assimilate and react to rapidly changing environmental information. "By-wire" systems present selected abstractions of a few crucial environmental factors. Instrumentation and communication technologies aid in evaluating alternative responses. Computer system intercepts the pilot's command and translates it into thousands of detailed orders that orchestrate the system in real time. When pilot's fly by wire, they're flying information representations of airplanes. The key is that the instructions are intentions, not orders. In a similar way "managing by wire" is the capacity to run a business by managing its informational representation. "Information systems have reduced decision information costs by allowing decision makers cost-effective access to information and powerful tools (e.g., simulation and econometric modeling) for analyzing the retrieved information. The improvement in decision quality in turn increases operational efficiency. For example, accurate forecasting of future demands, coupled with efficient handling of material flows and production scheduling, can achieve a significant reduction of inventory costs. Indeed the impact of this information revolution has been felt at all levels of organizations, industry, and society as a whole." (Gurbaxani and Whang 1991). The effect is a beginning of management by wire of complex industries and value chains. Professionals will increasingly manage by instructing systems of their intention, without giving detailed orders.

Value Chain: an Abbreviated Business Cycle. The value chain is a metaphor for all the processes and information involved from acquiring raw material, creating products, and distributing them to customers." Only interacting organizations, which deploy information systems across supply chains, may survive. Increased integration within the supply chain will "create better information, which in turn should support lower inventories and improved financial performance. But the evolution of IT and diminishing transaction costs will also lead to a fundamental restructuring of industry practices for distributing and supporting products. ... The biggest impact of this restructuring will be on the intermediaries, such as distributors and warehouses, that handle physical goods and the information associated with them between the producer and final consumer." (Lewis and Talalayesky 1997).

Organizational Behavior is Becoming Interorganizational. "The contemporary competitive environment requires interorganizational exchange and coordination of logistics information to achieve common goals. ... The managerial and cultural aspects of logistics are as critical as the impact of IT. While the emergence of electronic data interchange (EDI) and electronic markets would seem to permit an increase in the numbers of suppliers firms can do business with, the opposite trend seems more evident: firms are developing close relationships with a small number of major suppliers. For example, Wal-mart's in-store inventories of items such as Tide and Crest are managed by Procter & Gamble, who have direct access to point-of-sale data." (Lewis and Talalayesky 1997) Suppliers manage inventories sometimes without any intervention by the retailer. In the retail grocery industry, the movement toward "efficient consumer response" seeks to more precisely match inventory levels at the manufacturer, distributor and retailer to actual demand. "Risks (such as transaction risks and agency costs) remain a real issue: how much information should be shared with suppliers? Should suppliers provide their customers with details of their manufacturing processes? What will happen if suppliers are allowed to see information on their competitors? A final issue is the development of performance indicators and reward mechanisms for managing partnerships between producers, distributors, retailers and transportation firms. For example, buyers and suppliers may need to decide how they will share reductions in transportation and inventory costs. " (Lewis and Talalayesky 1997).

Agency Theory. The reshaping of players in markets, including the demise of some firms, means that people's behavior will be changed. Information systems are crucial to this process. The alliances, which arise from value or supply chain management, must be supported by interorganizational information linkages. These alliances, as had electronic data interchange linkages, represent a relationship commitment that moves transactions away from the open market. These transactions require that organizations decide to implement interorganizational information technology. One example of the impact of information technology is in the size of organizations. According to Gurbaxani and Whang "firm size and the allocation of decision-making authority among the various actors in a firm are, to a considerable degree, determined by the costs associated with acquiring, storing, processing and disseminating information." (Gurbaxani and Whang 1991). Gurbaxani and Whang "argue that as decision-making rights are pushed downward in the organizational pyramid, the costs of communicating information upward decrease while agency costs resulting from goal divergence increase. Therefore, decision rights in an organizational hierarchy should be located where the sum of these costs is minimized. Modern IT can reduce the costs of communicating information by improving the quality and speed of information processing and management's decision making, leading to more centralized management. At the same time, IT can also provide management with the ability to reduce agency costs through improved monitoring capabilities and performance evaluation schemes, inducing decentralization of decision making." (Gurbaxani and Whang 1991).

Theoretoical Tennants for Sociable Software Architecture

This section describes the theoretical tenants motivating our work.

The issues discussed in the previous section indicate an important change in demands on technology. As information grows exponentially, as expectations grow to manage by wire, as the value chain tightens, as organizational behavior and information sharing increases, and as agency theory exerts its pressure in human as well as algorithmic manifestations, there will be a need for software with an ability to interact in a richer social context. As we explore "sociable" software architecture our core concern is our concept and treatment of representation. What do the symbols we manipulate represent? How does one determine meaning? Is meaning stable over time?

Representation: Internal and External – Internalize the Social, Externalize the Web

A rich debate about "representation" has swirled for decades. In the most recent chapter, Rodney Brooks has asserted that, "Representation is the wrong unit of abstraction in building the bulkiest parts of intelligent systems." (Brooks 1991). In another work, he states that in his robot creatures, "There are representations, or accumulations of state, but these only refer to the internal workings of the system; they are meaningless without interaction with the outside world." (Brooks 1998). The only external representation or models of the world are those determined to be useful. And these representations are stored in the world, not in the agent. For sociable software architectures, accumulations of compressed data and evaluations on its conclusions are internal representations of the system's history. Meaning about the "real" world is stored in the world, in the history and projected interactions with humans. Meaning is derived only from interactions with real world. A side effect is that we are well prepared to support knowledge systems. Knowledge is potential action based on information, data with meaning.

An insight we wish to contribute is that words (strings of characters and their vocal counterparts) are of the real world. We treat words as real objects, but not just as external representations of meaning. Words are data and not information. We must build meaning, create information, from local interactive processes with this data. While Brooks appropriately rejects internal representations (which would seem to include words) for robots, we treat them as sensor data, and the social interactions they indicate as the "real world". Is our culture and society any less real than gravity and physical objects? Artificial intelligence can "move around" our culture as easily as it does when Brooks' creatures move around the halls of his labs. For both of us actions are the crucible. Only words are part of the "outside world" just as sonar readings.

Discourse Communities Model the Social Realities of Interactions

This observation has interesting implications when viewed in light of Hjorland's observations in information science. Specifically, that "discourse communities" provide a tool for analysis, comprehension, and processing of social interactions.

Following pragmatic objectivity, we hold that there is no intrinsic meaning for knowledge or information in isolation from action. Information's meaning is neither subjective nor relative. Pragmatic objectivity is based on action theory/utility and a collective methodology.

Objective idealism will search subject matter in permanent, inherent characteristics of knowledge, or in permanent, inherent semantic relationships and try to establish standardized, permanent, fixed ways of analyzing relationships (disregarding their potential use). Subjective idealism, on the other hand, will search for the subject matter of a document in either the author's or the users' subjective perception of the document and try to develop a theory of subject analysis based on either the author's psychological world (as in classical hermeneutics) or the users' psychological world (as is done in parts of modern cognitive viewpoint).

From the position of activity theory, we do not look on subject as either inherent characteristics or as something subjective in an individual way. Individual subjects' behavior in relation to use and to representation of information should be interpreted in the light of a (sub)disciplinary context. (Hjorland 1997).

Meaning is social and collective, that is, we use a methodological collectivism methodology. We need

> … to see knowledge as a developed historical-cultural-social product. The alternative to studying individual subjects and individual information-seeking behavior is to study knowledge domains. That is to study their informational structures, their terminology, knowledge representation, communication patterns, all in connection with their theories of knowledge and theories of science. (Hjorland 1997)

The discourse community for Hjorland's information systems is an academic discipline. This generalizes to business and culture for systems that are more than information retrieval systems. The discourse community becomes central. It is the sole source of meaning. Based on observing a discourse community's action over time and by interacting with it, sociable software architectures can use and create information. Given a discourse community as our unit of work, and not just the individual, we become aware of the roles individuals have within the discourse community. Our approach must therefore be collectivist. The challenge is to implement a recording of cultural and social environment and behavior. If we can do this, we have a basis for intelligence. Again the key to intelligent meaning is action, not representation of a permanent world.

For Brooks' creatures, there is intelligence without external representation. They operate in a real, sensory world with humans. We assert that the Internet and its databases are also a real, sensory world with humans. It is a world filled with words and figures. Agents operating on perceptions of these objects can learn to create useful behavior. "The key idea … is to be using the world as its own model and continuously match the preconditions of each goal against the real world." (Brooks 1991). Semantics arise from the success or failure of actions based on outcomes of processes performed on the raw data. There is no intrinsic meaning or semantic content before this. There is no external representation except for "maps" and "behavior" in sociable software architecture.

Social Interaction: Mentorship for Learning

Because we need meaning for information and knowledge and meaning comes from interaction with humans, social interaction is vital to sociable software architecture. Social interaction can be structured in a variety of ways, but the most basic is that of mentorship. For elemental interactions with one human, systems should allow for a mentor. Sociable software architecture will need a mentor the way Star Trek's Data needed Dr. Su. The mentor provides artificially structured or meaningful interactions. Brooks recognized this need when he used scaffolding and feedback for his creatures. Scaffolding and feedback allow humans and other creatures to learn from humans. The learning process allows us to share attention with the care giver. The care giver in return narrows our focus to what is relevant and most helpful in learning about the matter at hand.

While sociable software architecture' architecture will change its clustering or conclusions depending on human evaluation of the proposals, sociable software architecture has two other methodologies for mentors to use. Since sociable software architecture consists only of rules and data, a mentor could directly intervene in one or the other. The use of Lenat's Cyc, for example, is an addition that would improve sociable software architecture' discernment of words. (Lenat and Guha 1990).

For sociable software architecture, there are many individuals in many discourse communities and there are roles for these individuals in their discourse communities. In either case, learning from humans is basic.

Roles in the Discourse Community: Innovators, Trend Setters, Maintainers. Because meaning is not just psychological, but is also sociological, we need a social unit of meaning of work. The discourse community is that unit of meaning. While there are agents for individuals in sociable software architecture and the individual contributes to meaning (especially trendsetters), the individual is not the determiner of meaning. The meaning we seek to understand is not subjective at either the author's or the consumer's end.
Discourse communities though are composed of many individuals. Each individual may have one or more roles and these will change over time. The challenge will be to first place an individual in a discourse community and

then determine what roles they play. Some individuals are innovators and early change agents. Other individuals are successful trend setters and while not the first to innovate, when they do so, their new direction will usually be successful. Other individuals are maintainers of the discourse community. These individuals provide the instruction needed by new members to adapt to the existing discourse community. That an individual operates in several domains or discourse communities is a source of richness, appropriately gray, however, knowledge of an individual's membership in a discourse community and of their role in it is useful to the individual's agent in predicting interest in new sites, for example, if I am a trendsetter in a Star Trek community and an increasing number of innovators have been visiting and returning to "www.foo.com" then my agent is safe in directing me to it.

Changes Over Time are Important. Current web agents are either one time use, or require registration and its subsequent loss of privacy. Ideally, users would like persistent web agents under their control. Time is significant for discourse communities.

There are several forms of time: online and offline; evolution of individual, community, individual in community; and world of words and of sensory data of the world of words. Two forms of time that are particularly important are the evolution of discourse community's meaning and the interaction time with members of a discourse community.

The evolution of the discourse community's meaning must be tracked. Paradigm shifts can sometimes rapidly occur. Changes in legislation or regulation, for example, can shift the meaning of what had been a stable representation of the world. Because a sociable software architecture stores its representation of the world's model in the world, sociable software architecture is both vulnerable to shifts and should be accurate in tracking them.

The interaction time of the individual is entirely the business of the individual's agent and is entirely personal. Understanding this sense of time will allow sociable software architectures to respond to queries in a "get-back-to-you" mode when queries require a bit of time. Also, sociable software architectures will need to understand when to present innovations or changes, that is, when to anticipate a query.

Added to all these processing times, we still have the time frames inherent in the raw source data. Therefore there is a need for data fusion and knowledge representation structures such as blackboards. While not significant in all situations, sociable software architectures will need to understand whether a new datum is a confirmation of a known fact or a new fact.

Improving Problem Solving Capacity

This section explores the benefits we see in pursuing this technical approach.

An analysis of the potential improvements to problem solving, consider the problems trying to find something on the web. Current web searches return thousands of hits to a query. Even meta searches are inadequate since the best they can do is show the intersection of individual searches. My search engine needs to know about me, what I do, what I'm interested in, and I'd like it to work overtime finding interesting sites, calling my attention to them, and not just selling me new things. I'd like it to learn about me, but to keep that information private, and not share it with vendors so they can bombard me with email, and slot me into the appropriate banner add categories. I'd like my web search agent to be proactive, have good recall and find related work, and to guard my privacy. If it can learn by sharing information with other agents, fine, I'd like it to learn to become more customized, but all the while, it must safeguard my privacy.

Anticipating Needs with Proactive Models. Many current information-seeking systems rely on a query before processing and returning solutions. We propose proactive solutions either triggered by a specific query or more often searching based on both past queries, probable queries, and queries of similar users. If the system designer considers the solution space as a social epistemic space, this becomes feasible. Information retrieval system, rather than searching for a subject string indexed to a solution, should consider Hjorland's statement that: "The subject of a document is the epistemological potential of that document." (Hjorland 1997).

Improving Recall and Relevance. Doing so will improve both the recall and relevance and the timeliness of the solutions offered by systems. If we evaluate solutions using ecological behavior, we have a more robust evaluation system. "The understanding of an individual's optimal choice in information seeking is based on some conditions about the ecosystem in which that individual forms a part." (Hjorland 1997).. If, for example, we evaluate the content of a representation on the basis of whether it can contribute to the solution of a concrete problem or satisfy a need (such as one or more of Maslow's hierarchy of needs), then we can search a larger space continually, even before a specific query is asked.

This would improve retrieval time of very large and distributed stores and processes. This allows for spontaneous pre-processing and continuous agent clustering of potential solution spaces. It permits use of local syntax and representation of potential solution spaces. A sociable architecture also provides the robustness to handle diverse time frames and sources through data fusion.

Maintaining Privacy. Unlike current systems, it is both possible and advantageous to implement privacy of discourse community individual members. Solution paths are through clusters' of members' behavior separate from their identity. Perhaps if the current systems had been originally built by consumers instead of vendors, there would be fewer added process and data store baggage of collecting individual identities in order to actively approach individuals to buy more services and products.

Improving Efficiency with Distributed Processing. Among the pre-conditions of such social cognitive systems are a memory of similar others' use and the ability to identify similar others. Advantageous in today's networked systems would be distributed stores, pre-processing and processing behaviors. Agents need to be able to evaluate user, vendor, and system data and behavior. Agents need to be able to seek information with or without a query.

Customized to Individual Users' Needs. Two post-conditions are the capability to contact an individual in an appropriate style as chosen by the individual and to accept evaluation of the by the individual, both directly and by inference. Appropriate access includes language pre-requisites and formats.

Because of the need for sociable systems, many types of information systems must link with each other: databases and data warehousing systems; information retrieval applications; and artificial intelligence and decision support systems. Link analysis agents with intent controls own execution, and object oriented programming systems especially with polymorphism at trans-component level are indications of this need. Also important will be the treatment of time.

Implementation Model

This section describes an approach to implementing a sociable software architecture.

Sociable systems must interact with the primary users they serve. Often the primary users are humans. Humans learn and change the rules with which systems must stay abreast. Humans also define the meaning of the reality or world sensible to systems. Often this sensible reality is data. The systems then have processes to use on the data, often transforming data into information and at times knowledge.

We use an n-tier agent model, consisting of at least source, discourse community, and individual user agents. In order to handle the complexity and ultra-customization demanded by the people providing our motivation, we needed an implementation that was intrinsically distributed.

Within this n-tiered system, universal structures and operations occur. Data compression, distributed processing, background processing, and data stores are resulting benefits.

Source Agents

Source agents work with the raw source data whatever it may be. In interactive systems this includes tracking the bulk data about use of source data by users. These source agents make the first approximation of sorting with demographics and other structural clues the source and use data into discourse communities.

The source agents have discourse community maps and trees of their characteristics and overlap of these clusters. The source agents use a variety of tools including statistical, and node and link analysis. This analysis includes citation analysis where appropriate against a tree of citations. "Citations" is used here in a very general sense. In addition to the usual bibliographic citations, we include hot links in html sources and even borrowed vocabulary or techniques, if they and their source can be identified.

This identification and clustering essentially becomes input to one or more discourse community agents via blackboards and other structures.

Discourse Community Agents

Through blackboards and other structures the agents use for a discourse community are available to source agents. Both sets of agents share many of the same resources, which are distributed so that discourse community agents use the portions of the resources usually needed by them without drawing down the whole resource pool.

An example of this is the thesaurus. Certainly tools such as Lenat's CYC will provide a foundation, but the thesauri will grow through interaction between source and usage data. A more permanent thesaurus, or at least one with a longer historical memory, will be the meta-thesaurus where higher level patterns are stored. The thesauri will be trees within trees, a terminology semi-hypercube. The discourse community agent thesauri are the keys to the compressed data files created by source and other agents.

The discourse community agents and their data stores and thesauri are distributed by community. Some communities are themselves distributed, others are centralized.

Individual's Agents

Each individual served has an agent that understands that the individual is a member of several discourse communities and has different roles in each community. It is the partnership of the discourse community agent and the individual's that generates the anticipatory responses from the individual's agent.

It is the individual's agent that receives the queries and the evaluations to the responses to queries. The individual's agent maintains the data identifiable to the individual. This agent is the one that creates and protects the individual's privacy. This agent is also the maintainer of the individual's personal thesaurus as well as the sense of appropriate time and formats for interacting with the individual. In an extreme example, this agent may communicate with the individual solely through another person. Instances might include the parents of infants.

Models of the Discourse Communities

This section describes some of the mechanisms used to implement a sociable software architecture. We depend on similarity measures or classification using vectors in an n-dimensional space. As source agents conjure a new discourse community or discourse community agents decide that a discourse community is splitting into separate communities, we create new agents for the new discourse community.

The models of the discourse are multi-dimensional arrays. Vector analysis of the array dimensions provides us with a discourse community's centroid or fingerprint characteristics. This includes the features of clusters of individuals performing a role within the discourse community.

Lately we have been experimenting with compressing the data store by transforming similar data into layers of a tree. These will be shared across agents as a source of the "meta knowledge" about the individual's membership in and roles related to discourse communities. Three benefits can be derived from this: data compression, facilitation of distributed processing, and background processing.

The compression highlights similarities as well as outlier patterns. We then look for the rare patterns that over time become mainstream. The individuals attached to the once rare patterns are the trendsetters. If their role persists, then we can view probable developments for the community.

Data Compression Derives Naturally from the Use of Discourse Community Thesauri

All the agents regardless of level, create and use compressed data. We can compress data in such a way that it does not need to be uncompressed for operations to occur. Inspired by wavelets, we may be able to compress data so that the differences are highlighted, rather than obscured.

A classic discourse community example is the collectors of Eighteenth Century Hong Kong stamps. When a discourse community is well defined, then the compression will be tighter. Essentially we create an array string of what is most similar. We then remove these dimensions from the arrays of all members and create a pointer to what is similar, or note differences with the similarity standard. The identifier array for Hong Kong stamps will overlap with those of stamp collectors and those interested in the history of Hong Kong.

Thesauri terms and links most attached to the fingerprint array becomes the key to the compression of source data. We use the fingerprint thesauri to store the similarities. This allows us to also restore the raw data to a specified level. This creating, dissembling, and re-assembling initially will be common as the agents create hypotheses that they later abandon.

Because the thesauri and the compressed data are actually the internal representation of the world, other agents and entities may use them as sociable software architecture evolves.

Compartmentalization of Thesauri Facilitates Distributed and Background Processing

The distributed architecture of sociable software architecture means that we must use techniques such as information fusion. Raw data will arrive from a variety of sources in a variety of modes. Blackboards and other such mechanisms will be used by sociable software architecture agents to communicate clearly on the details of the raw data.

This will be important for all the agents at every level, but especially true for the source agents who are mining data streams.

Conclusions

The complex relationship of networks of information that exist on the web and in modern computer systems requires advanced link-analysis capabilities in order to make sense of the deluge of data. A number of factors contribute to this including exponential information growth demand for smart systems, desire to "manage by wire," the abbreviated value, and dynamic business organizations. The agents currently available on the web seem to have no memory and exist only to exploit privacy. We propose a link-analysis architecture focused on the individual with attention on the individual's "discourse communities." We believe this focus, along with the development and refinement of a discourse community thesaurus can uniquely begin to model the social and cultural component's of a user's profile. This profile can help to begin to create more truly intelligent agents.

The goals of this paper have been to begin to delineate the pressures towards more advanced intelligent agents which will provide a richer framework for link analysis, to survey some related work from information theory, and to suggest an architecture which might be useful. We suspect that we've opened more questions than we've answered, and (an admirable thing to do in some respects), and we hope that our further research will continue to address these pertinent issues.

References

Bovet, D. and Sheffi, Y. "The Brave New World of Supply Chain Management." *Supply Chain Management Review.* Vol 2. No. 1 Spring 1998. 14-22.

Brooks, A. Rodney, et al.. "Alternative Essences of Intelligence" in *Proceedings: Fifteenth National Conference on Artificial Intelligence (AAAI-98) and Tenth Conference on Innovative Applications of Artificial Intelligence (IAAI-98)* 961-968. Menlo Park, CA: AAAI Press/MIT Press, 1998.

Brooks, R., ``Intelligence without representation," *Artificial Intelligence*, vol. 47, 139--160, Jan. 1991.

Byrne, S. M. and Javad, S. "Integrated Logistics Information Systems (ILIS): Competitive Advantage or Increased Cost?" *Annual Conference Proceedings* (Oak Brook, IL: Council of Logistics Management, 1992). 55-73.

Davis, S. and Botkin, J. "The Coming of Knowledge-Based Business." *Harvard Business Review.* Sept-Oct 1994. 165-170.

Gurbaxani, V. and Whang, S. "The Impact of Information Systems on Organizations and Markets." *Communications of the ACM.* Vol. 34. No. 1. January 1991. 59-73.

Hjorland, B. *Information Seeking and Subject Representation: An Activity-Theoretical Approach to Information Science.* Westport, CT: Greenwood Press, 1997. ISBN: 0-313-29893-9

Lenat, D.B. and Guha, R.V. "Building Large Knowledge-Based Systems", Addison-Wesley, Reading, MA, 1990. ISBN 0-201-51752-3.

Lewis, I. and Talayevsky, A.. "Logistics and Information Technology: A Coordination Perspective." *Journal of Business Logistics.* Vol. 18. No. 1. 1997. 141-157.