The Emerging Knowledge Management Paradigm: Some Organizational and Technical Issues

Jens Pohl
Collaborative Agent Design Research Center
Cal Poly State University (Cal Poly)
San Luis Obispo, California, USA

Abstract

This paper addresses the expectations, organizational implications, and information processing requirements, of the emerging knowledge management paradigm. A brief discussion of the enablement of the individual through the wide-spread availability of computer and communication facilities, is followed by a description of the structural evolution of organizations, and the architecture of a computer-based knowledge management system. The author discusses two trends that are driven by the treatment of information and knowledge as a commodity: increased concern for the management and exploitation of knowledge within organizations; and, the creation of an organizational environment that facilitates the acquisition, sharing and application of knowledge.

Tracing the evolution of the structure of organizations, the author concludes that the web-like features of the Network Model are most conducive to the promotion of knowledge management principles, even though this model does have liabilities that require careful monitoring.

The paper further discusses in some detail the architecture of a knowledge management system that consists of a lower integrated data layer and an upper information layer. Attention is drawn to the need of the data layer to include not only archived summary data as found in Data Warehouses and Data Marts, but also near real-time operational data with convenient access provided by Data Portals. An important distinction is drawn between data-centric and information-centric software environments in terms of software with an internal information model capable of supporting agents with automatic reasoning capabilities. The paper concludes with a brief description of the mechanisms through which a Web-Services environment provides access to distributed data sources, as well as heterogeneous data-centric and information-centric software applications.

Keywords

agents, communication, complex adaptive systems, data, data-centric, Data Mart, Data Portal, Data Warehouse, enabled individual, information, information-centric, information management, knowledge, knowledge management, ontology, organization, organizational structure

Enablement of the individual

One of the more subtle consequences of the rapid advances in information technology over the past several decades has been the increasing focus on the individual. Enabled by powerful communication facilities and computer-based automation tools that vastly increase the capabilities of the user, an individual person can orchestrate and achieve more today than an entire organization was able to successfully undertake a mere decade or two ago. Recognition of the value of the individual is exemplified in multiple ways, ranging from the changing structure of business corporations, the rise of entrepreneurship and self-employment, to apparently exorbitant judicial compensation awards, and the increasing value placed on human life.

Emerging out of this technology driven environment are a new set of personal values and expectations that differ significantly from past social conventions. The enabling nature of this environment, in itself, presents a challenge through the increased opportunities that it offers to the individual. To take advantage of these opportunities, the individual who is proactive and willing to take calculated risks is likely to be more successful than the individual who is reserved and conservative. Similarly, the person who is self-reliant and willing to exercise leadership to reach objectives that are based on future trends, is likely to outperform the person who is subservient and intent on duplicating past successes.

As ideas, initiative and persistent motivation become more useful human qualities, risk taking will become recognized as being increasingly rewarded and conservatism as being increasingly penalized. Under these conditions traditional values such as prudent compliance, measured reactiveness and acceptance of the status quo will gradually fall out of favor. Instead, the more successful individual will have recognized the value of continuously monitoring events, identifying trends, and preparing for taking advantage of opportunities that are largely unpredictable in both their nature and timing.

The enablement and focus on the individual will undoubtedly also increase the level of societal stress and anxiety, as a significant number of persons find it difficult to keep pace with the tempo of technology driven change. Specifically, there is likely to be an increasing demand for freedom without a commensurate willingness to exercise self-constraint. At the same time the rapidly increasing desire for a higher quality of life and the mounting aspirations for personal achievement will for most persons fall short of their expectations.

Knowledge as a commodity

As information technology begins to permeate all aspects of life and the economy turns decidedly information-centric, wealth is increasingly defined in terms of information-related products and the availability of knowledge. Under these conditions employment, whether self-employment or organizational employment, is becoming singularly focused on the skills and capabilities of the individual. In other words knowledge has become a commodity that has value far in excess of the manufactured products that represented the yardstick of wealth during the industrial age.

How this new form of human wealth should be effectively utilized and nurtured in commercial and government organizations has in recent times become a major preoccupation of management. Two parallel and related trends have emerged. The first trend is related to the management and exploitation of knowledge. The question being asked is: How can we capture and utilize the potentially available knowledge for the benefit of the organization? The phrase "…potentially available" is appropriate, because much of the knowledge is hidden in an overwhelming volume of computer-based data. What is not commonly understood is that the overwhelming nature of the stored data is due to current processing methods rather than volume. These processing methods have to rely largely on manual tasks because only the human user can provide the necessary context for interpreting the computer-stored data into information and knowledge. If it were possible to capture information (i.e., data with relationships), rather than data, at the point of entry into the computer then there would be sufficient context for computer software to process the information automatically into knowledge. This is not just a desirable capability, but an absolute requirement for the capture and effective utilization of knowledge within an organization and will therefore be discussed in more detail later in this paper.

The second trend is related to the structure of the organization itself. Efforts in this area are focused on creating an environment that encourages and facilitates the acquisition, sharing, and application of knowledge. Commonly referred to as ***knowledge management***, these efforts have the goal of effectively developing and utilizing the human capital in an organization. More specifically, the objective of knowledge management is to enable all human and organizational capabilities and relationships for the benefit of the individual and the organization. This requires the encouragement of every member of the organization to be a contributor and a potential decision maker. How can this be achieved? Decentralization and concurrency are principal characteristics of knowledge management, aimed at creating an environment that builds relationships for the purpose of maximizing interaction, diversity, responsiveness, and flexibility.

In this respect knowledge management views an organization and its external environment as a complex adaptive system of many component parts acting in parallel. The principal component parts of the organization are the human players, including not only the employees but also the external individuals and groups that the organization interacts with. Holland (1988) characterizes complex adaptive systems as a network of many agents acting in parallel. Each agent is always ready to interact with the system, proactively and reactively responding to whatever the other agents are doing. As a network, a complex adaptive system is by its very nature highly decentralized. In other words, any coherent behavioral patterns of the system are due to the collective competitive and cooperative activities of its parts (i.e., agents or elements). It follows that such a system has many levels of organization, with the agents at any level contributing in a building block manner to the agents at a higher level. For example, a group of individuals will form a team or department, a number of departments will form a division, and so on through an organization. Most importantly complex adaptive systems are constantly changing, revising and rearranging their building blocks through their activities as they adapt to their experiences within the system.

Two essential requirements for the relative success of an organization, within the context of such a dynamically adaptive environment, are anticipation of the future and communication. Neither of these are necessarily akin to human nature. The fundamental (i.e., biological) experience-based nature of the human cognitive system provides us with few tools to deal with situations that are not the same or at least similar to past experiences. Anticipation of the future therefore represents a precarious excursion into unknown territory that is typically accompanied by an elevated level of anxiety due to uncertainty, frustration and fear. The uncertainty stems from the unknown nature of the future, which differs fundamentally from the certainty of the past. Therefore from a human point of view, dealing with the future represents an emotional effort that challenges our confidence to survive and prosper within our environment. We become frustrated as we see many of the methods and tools that have allowed us to survive and prosper in the past, progressively fail as we try to apply them to new conditions and situations. We are forced to stumble along as we learn by trial and error. It is therefore only natural for us human beings to avoid any excursions into the future unless they are forced upon us. With few exceptions we tend to cling to the apparently safe domain of the past, unless we are compelled to face the present and future by developments in our environment that severely threaten the comfort level of our current role. Clearly, the requirement for anticipation in a successful organization is not naturally satisfied by its human players and must therefore be continuously fostered by other stimuli.

Since a complex adaptive system depends greatly on the continuous interaction of its component parts, the maintenance of open communication channels between the human players of an organization is an essential requirement for knowledge management. The more active individuals or groups of players are the more critical the exchange of information and knowledge becomes to the welfare of the organization. Yet, there is a natural tendency for human beings to reduce their external interactions as they become more focused on their activities and, often to an even greater extent, as these activities appear to become successful. Both the concentration of their attention and the selfishness of their ambitions mitigate against the sharing of the knowledge acquired through their efforts. Again, this conflict between inherent human behavioral characteristics and the prerequisites for organizational success requires special attention in a knowledge management environment.

Evolving organizational structures

It is to be expected that organizational structures will evolve over time in direct response to societal changes. The evolution of the role of the individual from a compliant and subservient implementer to a proactive initiator, has had a profound influence not only on the structure of organizations but also on the manner in which they operate. Over the past century and in particular since World War II, the notion that the members of an organization need to be controlled through the application of hierarchical authority has been gradually displaced by the need to survive in an expanding market and under increasing competition. Advances in information technology have not only generated vastly improved ways of accomplishing tasks, but they have also created unprecedented opportunities for persons with few material resources to provide services and products in direct competition with much larger established organizations. The history of the microcomputer abounds with examples of very small groups of individuals who not only created new products but literally forced some of the largest industrial organizations to change their product lines, revise their marketing strategies, and abandon their existing organizational structures, for the sake of survival.

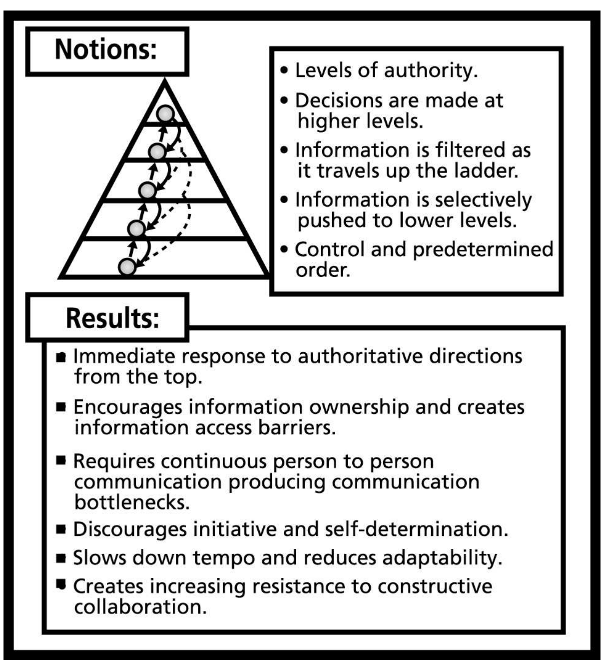

Figure 1: Strictly Hierarchical Model

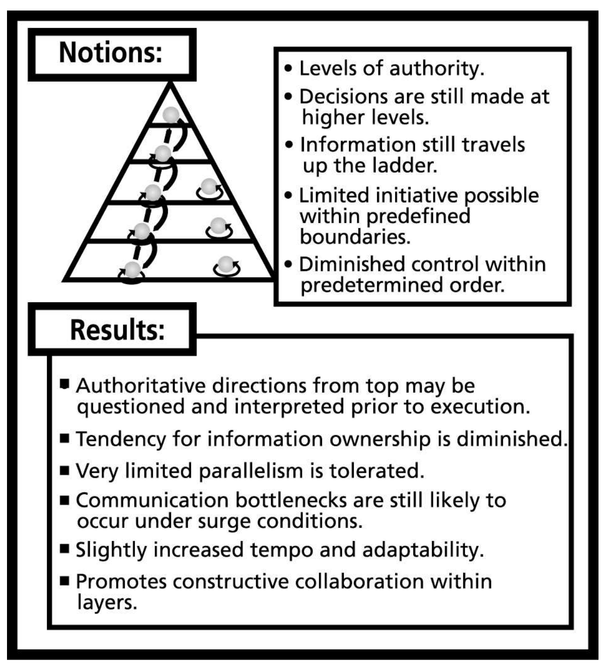

Figure 2: Loosely Hierarchical Model

During the 20th Century the formal structure of organizations has gradually adapted to take advantage of the potential contributions of the individual enabled by a new set of information technology tools and skills. However, even though the transition from the hierarchical authoritarian model to a web-like structure is readily discernable, examples of virtually all intermediate products of this transition can still be found today.

The incompatibility of the traditional Strictly Hierarchical Model with modern knowledge management principles is clearly seen in the notions expressed in Figure 1. This model relies fundamentally on the concept of vertical levels of decision authority. In practice, however, most decisions are made at the highest levels because of the limited delegation of authority to lower levels. Control and predetermined order pervades every operational aspect of the Strictly Hierarchical Model. In particular, the insistence on control inhibits the flow of information both upward and downward. Information is filtered as it travels upward from level to level based on what the lower level believes the upper level would like to receive and hear. The more authoritarian the operational implementation of the hierarchical model the greater the degree of filtering, with the attendant increased isolation of the decision makers from the realities of the operational environment. For entirely different reasons the higher levels of the organization are often reluctant to provide the lower levels with more than the minimum information that they believe is required for the execution and implementation of instructions.

The single advantage of the Strictly Hierarchical Model is that it responds immediately, decisively, and effectively, under predictable conditions that have been anticipated and for which good plans of action exist. However, as soon as the original plan has to be modified due to changing conditions, there is a real danger that the organization will not be able to respond in a timely manner. The more dynamic the operational environment (i.e., driven by external and internal forces) the less effective the Strictly Hierarchical Model becomes. The inability of this organizational model to respond to dynamically changing conditions is exacerbated by information ownership and the propensity for producing communication bottlenecks. The intrinsic limitations placed on the flow of information within this organizational model encourages persons within the organization to consider themselves as custodians of information that is made available to others on a strictly selective basis. This creates serious barriers to the access of information both vertically and horizontally. In addition, the strictly controlled upward and downward flow of information through person-to-person channels tends to produce communication bottlenecks. As a result the operational tempo and adaptability of the organization are greatly reduced, leading to the discouragement of initiative and a general resistance to constructive collaboration.

The Loosely Hierarchical Model (Figure 2) somewhat improves the ability of the organization to respond to a moderately changing operational environment. While it still maintains levels of authority, with all but routine decisions being made at the higher levels, it tends to allow some limited degree of initiative within predefined boundaries. The slightly diminished insistence on control, within the context of the predetermined order of the organization, allows authoritative directions from the upper levels to be questioned and interpreted prior to execution. As a result a limited amount of parallelism is tolerated, leading to the encouragement of a moderate degree of constructive collaboration within the lower levels. However, while the tendency for information ownership is diminished in the Loosely Hierarchical Model communication bottlenecks are still likely to occur under surge conditions.

The need for more timely responsiveness to a dynamically changing environment forced two important recognitions: the need for increased parallelism; and, the need for more direct communication. These requirements led to the elimination of the concept of a pyramid of layers, thereby transforming the hierarchical model into a Star Model (Figure 3) of nodes grouped circumferentially around a central hub. The Star Model assumes that the nodes will function in a parallel mode with a much greater but still limited degree of autonomy and self-determination. However, while information flow from and to the hub is direct for each node, there is little provision for direct interaction among the nodes. One could categorize this model as a form of centrally directed and monitored parallelism that still maintains a significant degree of control. In this respect the Star Model is clearly a transitional compromise that recognizes the restrictive nature of control but at the same time still insists on the guaranteed availability of a dominant control mechanism.

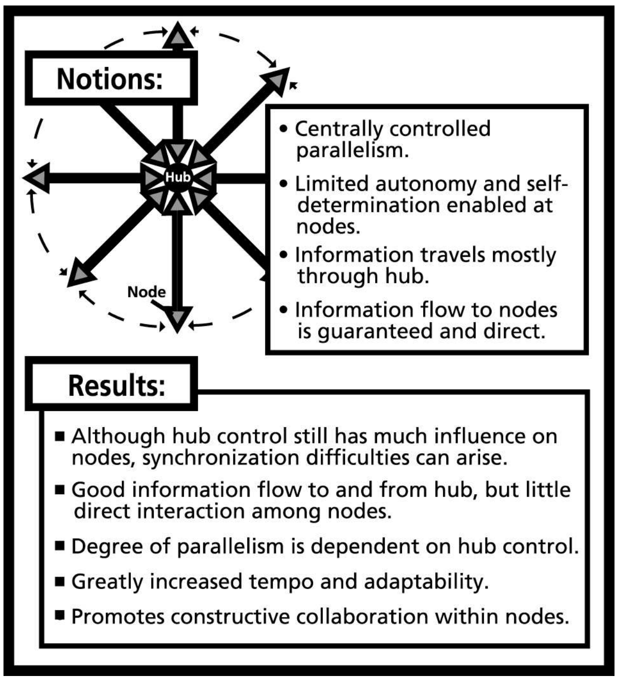

Figure 3: Star Model

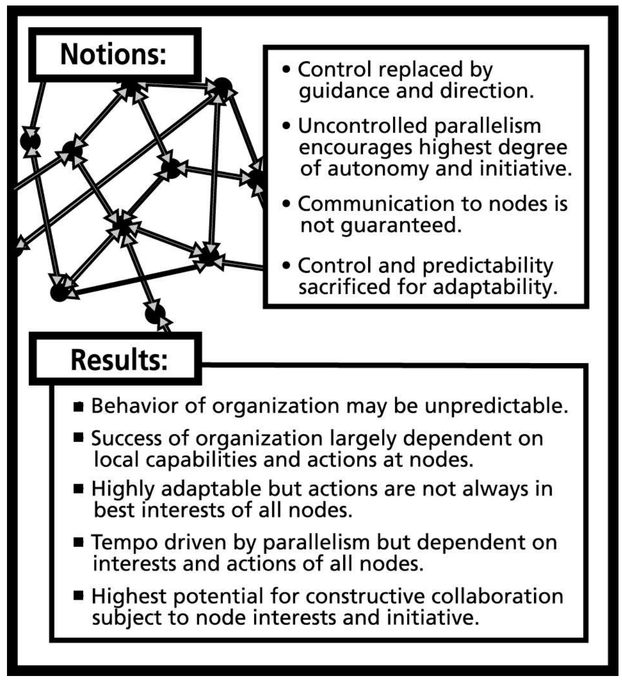

Figure 4: Network Model

Although the degree of parallelism that can be generated in a star-like structure is largely dependent on the degree of control maintained by the central hub, this organizational model provides greatly increased tempo and adaptability in comparison with either version of the hierarchical model. In addition, the Star Model promotes a more or less unrestricted degree of constructive collaboration within nodes even though any node-to-node interaction is constrained by the dominance of the hub.

In very recent times the increased demand for adaptability, self-determination and responsiveness, has progressively transformed the mandates of control and authority to the more acceptable notions of guidance and leadership. Consequently, the hub disappeared and the organizational structure flattened into a web-like Network Model (Figure 4). While there are now no barriers to the interaction of nodes, communication to nodes is by no means guaranteed. The Network Model sacrifices control and predictability for adaptability. It does this by encouraging virtually uncontrolled parallelism potentially leading to the highest degree of autonomy, initiative and self-determination. In this respect, the success of an organization with a web-like structure depends largely on the local capabilities and actions at the nodes. Although this organizational model has the highest potential for constructive collaboration, unrestricted due to the absence of control, the realization of this potential depends almost entirely on the interests and endeavors of the nodes.

Without strong leadership and a clearly articulated vision the Network Model has to struggle with three potentially serious liabilities. Firstly, lack of stimulation and purpose at the nodes can lead to inactivity and isolation. The model assumes that there is a natural tendency for node players to take advantage of their autonomy and exploit their essentially unrestricted freedom to full advantage. In the light of the previous discussion of human nature, this assumption may not be valid under certain circumstances. Secondly, very strong and highly motivated players at one or more nodes may become disruptive as they vigorously compete for resources and force the demise of other nodes. Such activities may not be in the best interests of the organization as a whole. Finally, the Network Model incorporates an innate propensity to be unpredictable. By maximizing its ability to adapt to both internal and external changes the model can adapt at a rate that outpaces the ability of its leaders to recognize the nature of the changes and maintain a relevant organizational vision. Under these circumstances there is a distinct danger that the organization will squander its resources in unproductive areas as the guidance provided by its leaders becomes less and less relevant to the actual activities of the nodes.

Clearly, the Network Model is most compatible with the principles of knowledge management. It provides the necessary freedom for an organizational environment in which leadership serves as a motivator, catalyst and enabler, rather than a taskmaster. However, in the absence of strong and tireless leadership the network model is vulnerable to internal manipulation by overly competitive nodes, to inactive nodes due to lack of stimulation or an unwillingness for node players to exercise initiative and self-determination, and to uneven performance and the formation of isolated groups (i.e., at the nodes) as responsibility assignments and accountability expectations are ignored.

Information-centric computer software

Apart from an organizational structure that encourages initiative and self-determination, and leadership that provides vision and guidance, there is a third prerequisite for a successful knowledge management environment. This prerequisite is related to the capture and exploitation of the information and knowledge that is generated within an organization. What is the nature and form of this information? It includes not only the continuous information streams such as e-mail messages, telephone calls, minutes of business meetings with external parties, and other documents, but also the information and knowledge that is generated within the organization. The latter is typically fragmented throughout the organization and much of it is potentially lost soon after it has been created and used for a particular purpose. It ranges from the minutes of internal meetings, proposals, reports, white papers, technical references, to the cumulative experience and knowledge that resides in the memory of the members of the organization. In most existing organizations attempts to capture this information vary from formal systematic efforts such as maintaining an on-line database of customer service calls and response actions, to some nebulous knowledge of who worked on a particular project and might therefore be able to contribute some key information to the current problem.

With the increasing realization that the information and knowledge generated through the internal and external activities of an organization constitutes a major asset and must therefore be a key component of any knowledge management plan, many organizations are asking themselves the following questions: What are the fundamental elements of this resource?; How can this resource be efficiently captured at the source and stored electronically?; Does this resource have to be processed (e.g., validated, analyzed, and evaluated) in some way to make it useful?; and, How can we provide convenient access and yet keep this valuable resource secure? These questions form the focus of the remainder of this paper.

The fundamental elements: The principal elements or building blocks of a knowledge management system are data, information, knowledge, and wisdom (Figure 5). Data essentially are numbers and words without relationships (Pohl 2001, 2003). We human beings are able to interpret data into information by utilizing the context that we have accumulated in our cognitive system over time (i.e., our experience). Computers do not have a human-like cognitive system and therefore any data stored in a computer will need to be interpreted by the human user (Figure 6). While the computer is able to order, recast, categorize, catalog, and process the data in many different ways, it cannot use it as the basis of any reasoning sequence. However, if we store not only the data but also at least some of the relationships that place the data into context then it is not difficult to develop software modules (i.e., agents) with reasoning capabilities. In this way it is possible to develop computer software with increasing understanding of what it is processing.

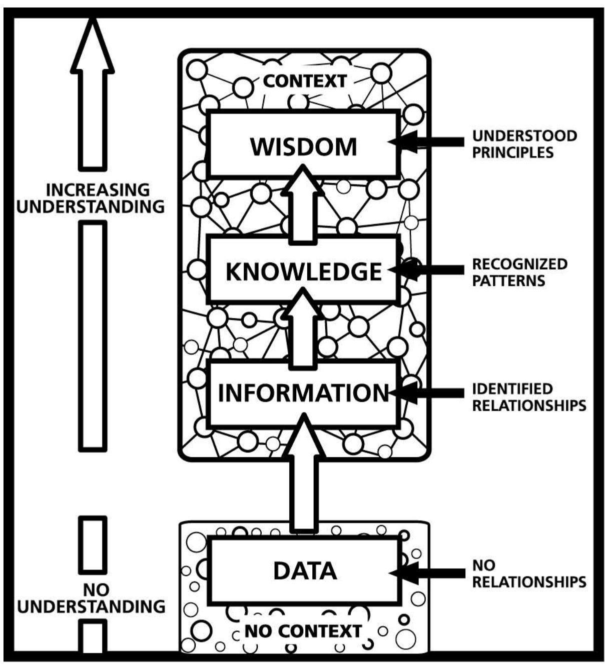

Figure 5: Importance of context

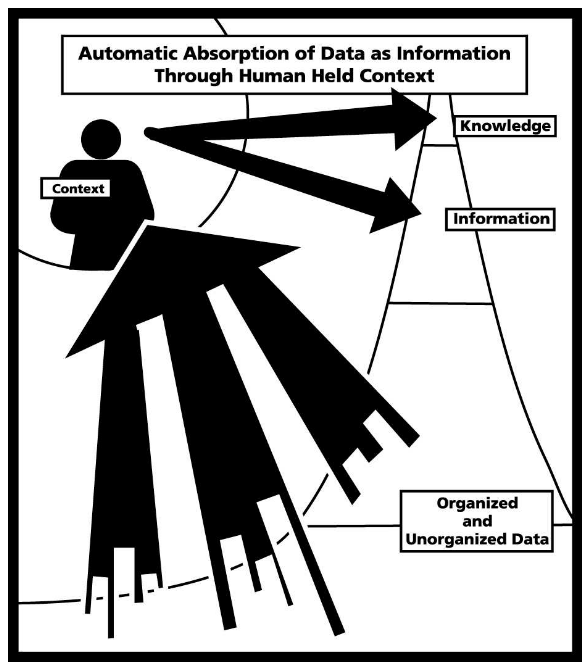

Figure 6: Human interpretation of data

The ability to represent information in computer software has been available for at least the past 30 years (Winston 1970, Biermann and Feldman 1972, Cohen and Sammut 1978). Hampered initially by a lack of hardware power and later by the absence of any compelling need to involve the computer in the direct interpretation of data, these information modeling techniques were not applied in the mainstream of computer software development until very recently. The compelling reasons that have suddenly brought them to the foreground are the increasing volume of computer-based data that is beginning to overwhelm human users, and the homeland security concerns that emerged after the tragic September 11, 2001 terrorist incidents in the United States.

The physical gap that is shown schematically between the realms of the data environment without context and no understanding and the information environment with context and ascending levels of greater understanding in Figure 5, is intended to underscore the fundamental difference between the two realms. The transition from data-processing software to information-centric software requires a paradigm shift in the human perception of the role of computers. Incorporating an internal information model (i.e., ontology) that represents portions of real world context as a virtual environment of objects their characteristics and the associations that relate these objects, information-centric software is capable of performing a useful level of automatic reasoning. A number of software agents with relatively simple reasoning capabilities are able to collaborate and through their collective efforts come to more sophisticated conclusions.

The architecture of a knowledge management system: Since the early 1970s the ability of computers to store large amounts of data has been increasingly exploited by industry and government. The potential bottleneck presented by these electronic data stores did not become apparent until more recent times with the increasing desire and expectation that their contents should be utilized for planning and decision making purposes. The need to integrate and analyze data from multiple sources led to the concept of a Data Warehouse that is updated periodically with summarized data collected from operational data sources (Humphries et al. 1999). Structured into compartments or Data Marts, each focused on a particular functional area, the Data Warehouse serves as a basis for analyzing historical trends with On Line Analytical Processing (OLAP) tools and projecting future conditions with Data Mining tools. However, the usefulness of these tools is greatly constrained by lack of context. Even though the data in Data Warehouses are typically stored in relational databases, they commonly contain few relationships. Therefore, the ability of OLAP and Data Mining tools to answer What?, Why? and What-if? questions is severely constrained by the very limited context provided by the data.

Data Warehouses are one level removed from operational data since they archive summarized data that are periodically updated according to some predefined timeline. While this makes their contents suitable for historical analysis and planning purposes, it does not allow them to be used for near real-time decision-making which is dependent on operational data. Since the operational data involves many data sources, gateways have been implemented in recent times to provide convenient access to disparate data sources. These gateways are referred to as Data Portals and do not in themselves store data. Apart from accessing the data sources the principal functions of the Portal include the presentation of data to the user. Some Data Portals also include data analysis tools aimed at enriching the presentation capabilities.

Data Portals and Data Warehouses represent a structured data level that integrates the multiple, fragmented databases, files, documents, and e-mail messages that constitute the often only moderately organized operational data flow. By providing access to both the operational data (Data Portals) and the archived summary data (Data Warehouses) this structured data level represents the integrating data layer that constitutes the bottom layer of a knowledge management system, serving as a necessary foundation for an upper information layer (Figure 7). The upper layer utilizes an internal information model (i.e., ontology) to provide context for the automatic reasoning capabilities of software agents. Essentially, these agents enabled by their reasoning capabilities constitute a set of intelligent tools that continuously monitor the events (i.e., changes) occurring in the operational environment.

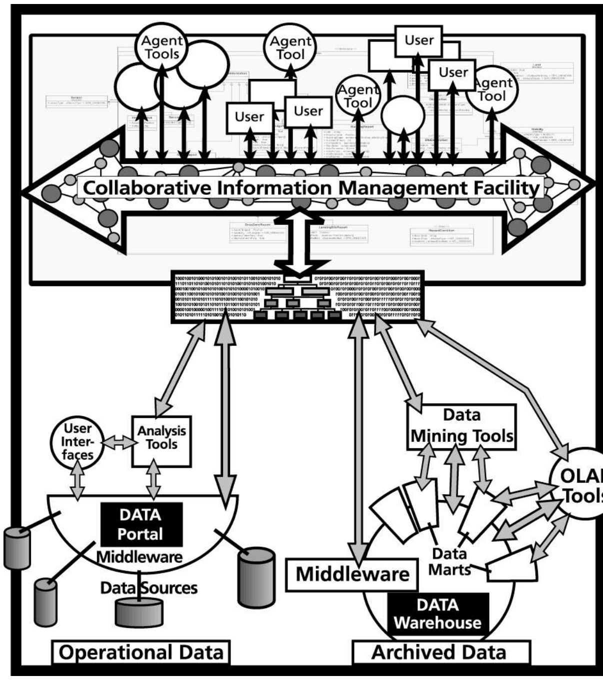

Figure 7: Schematic architecture of a knowledge management system

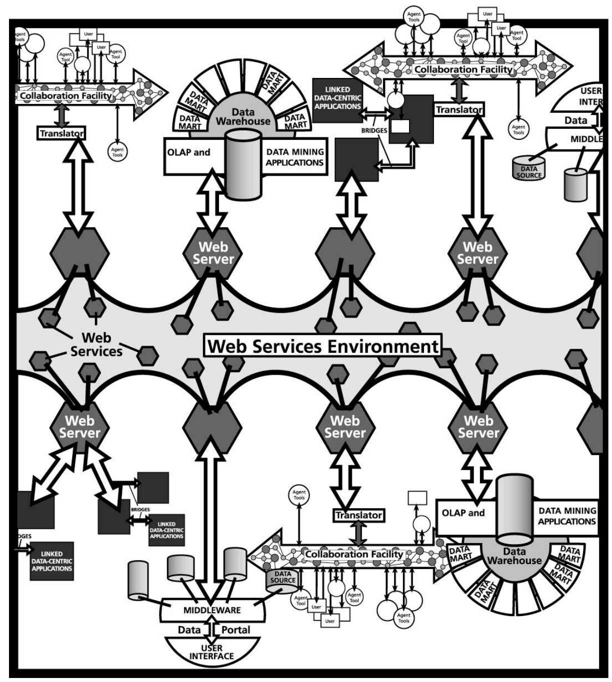

Figure 8: Integration of heterogeneous systems in a Web-Services environment

The interface between the lower data-processing layer and the higher information management layer consists of a translation facility that is capable of mapping the data schema of the lower layer to the information representation (i.e., ontology) of the upper layer (Figure 7). In this manner, the ontology of the information management layer can be populated with near real-time operational data and archived summary data from Data Warehouses. This mapping process should be bidirectional so that the results of agent actions can be readily transmitted to any data-centric applications that reside in the data layer.

Intelligent information management tools: There are many types of software agents, ranging from those that emulate symbolic reasoning by processing rules, to highly mathematical pattern matching neural networks (McClelland and Rumelhart 1988), genetic algorithms (Koza 1992), and particle swarm optimization techniques (Kennedy and Eberhart 2001). In general terms software agents are defined by Wooldridge and Jennings (1995) as "… *computer systems, situated in some environment, that are capable of flexible autonomous actions …*". The three critical words in this definition are situated, flexible, and autonomous. Situated means that the agent receives information from its environment and is capable of performing acts that change this environment. Autonomous refers to the agent's ability to act without the direct intervention of human users. In other words that the agent has some degree of control over its own actions and internal state. And, flexible means that the system is: responsive - by perceiving its environment and being able to respond in a timely fashion to changes that occur in it; proactive - by exhibiting opportunistic, goal-directed behavior and exercising initiative where appropriate; and, social - by interacting, when appropriate, with other agents and human users in order to complete its own problem solving tasks and help others with their activities.

How do these characteristics of software agents translate to the kind of knowledge management system described above (Figure 7)? The agent tools are situated since they receive a continuous flow of operational information generated by the activities of the organization, and perform acts that may change that environment (e.g., creating alerts, making suggestions, and formulating recommendations). The agent tools are autonomous because they act without the direct intervention of human users, even though they allow the latter to interact with them at any time. In respect to flexibility, the agent tools possess the three qualities that define flexibility within the context of the above definition. They are responsive, since they perceive their environment through an internal information model (i.e., ontology) that describes many of the relationships and associations that exist in the real world environment. They are proactive because they can take the initiative in making suggestions or recommendations (e.g., transportation mode selection for a particular shipment, emergency team configurations in crisis management situations, or route selection for moving troops or equipment) and they do that in an opportunistic fashion. For example, when an emergency call is initiated, a Route agent may immediately and without any explicit request from the user, determine the optimum route under current traffic conditions that should be used by the ambulance to reach the injured person.

The ability of software agents to communicate (i.e., socialize) with each other and with human users to work on their own problems or assist others with their problems, is a powerful capability of the information layer in a knowledge management system. It allows several agents to collaborate and concurrently explore different aspects of a problem from multiple points of view, or develop alternative solutions for future negotiation.

Symbolic reasoning agents that are quite common in knowledge management systems incorporate collections of rules that monitor specific conditions and generate alerts when these conditions are satisfied. The general design of such an agent consists of three components: the conditions that trigger the agent (i.e., the functional specification of the agent); the objects and their attributes that are involved in these conditions (i.e., the part of the internal information model (i.e., ontology) that is used by the agent); and, the logic that defines the relationships among these objects and attributes.

One important aspect of autonomy in agent applications is the ability of agents to perform tasks whenever these may be appropriate. This requires agents to be continuously looking for an opportunity to execute. In this context opportunity is typically defined by the existence of sufficient information. For example, to identify a shortage of inventory either some agent has to monitor the consumption of the particular inventory item until there is a shortage and then issue a warning, or one or more agents collaboratively project that based on developing conditions there is likely to be a shortage of the given item at some specific time in the future.

The requirements for rule-based agents are defined in terms of two elements: conditions; and, actions. The conditions are the specifications of the situation that the agent monitors, while the actions are the alerts that should be generated when these conditions are true. Typically, conditions are specified in terms of objects, attributes and the relationships among them. Each condition is formed by a pattern of object, attributes, values, and Boolean tests. Patterns are grouped by logical connectors, such as AND, OR, and NOT. The more patterns and relationships that are specified, the more specific these conditions become. The right hand side of a rule represents the actions to be taken when the conditions are satisfied. The most general type of action is to generate an alert. However, there are many other kinds of actions that rule-based agents can perform (e.g., look for additional information, modify an existing schedule or generate a new schedule, develop a particular solution approach, simulate the likely outcome of a course of action, and so on).

The Web-Services environment: A knowledge management system may be implemented as a set of Web-Services on the Internet or in any intranet environment (Figure 8). Existing Web-Services environments typically comprise a Web Server that utilizes the Hyper-Text Transfer Protocol (HTTP) for communication, the Universal Description Discovery and Integration (UDDI) protocol as part of the standard definition of Web-Services registries, and a Registry that already contains an entry for the accessing application as well as any number of other Web-Services. UDDI is an international standard that defines a set of methods for accessing a Registry that provides certain information to an accessing application. For perhaps historical reasons UDDI is structured to provide information about organizations, such as: who (about the particular organization); what (what services are available); and, where (where are these services available).

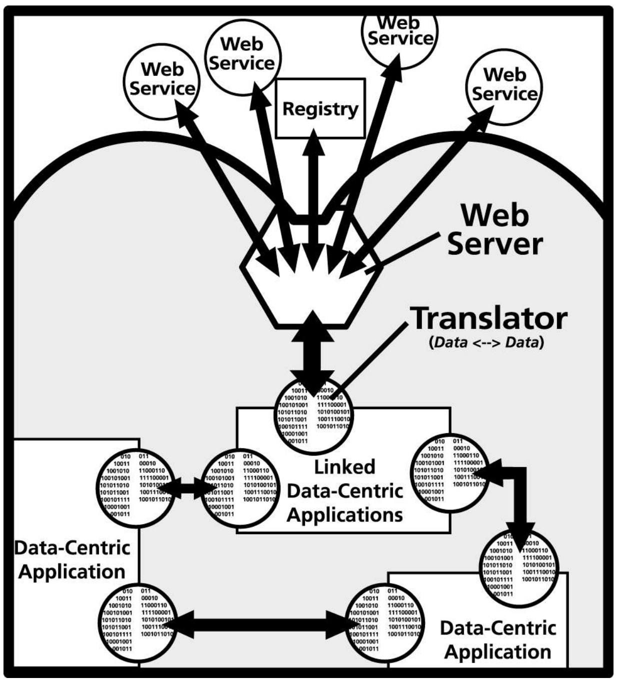

Figure 9: 'Exposing' a data-centric application to a Web Server

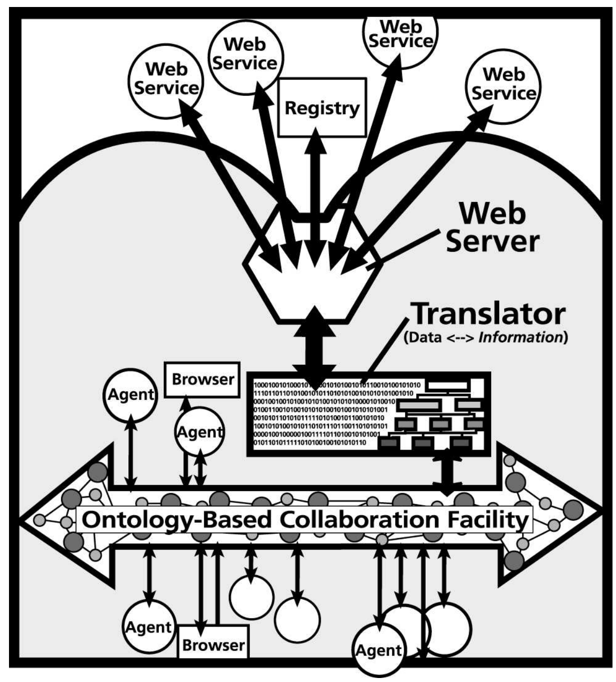

Figure 10: 'Exposing' an information-centric application to a Web Server

The Simple Object Access Protocol (SOAP) defines a protocol for the direct exchange of data objects between software systems in a networked environment (Figures 9 and 10). It provides a means of representing objects at execution time, regardless of the underlying computer language. SOAP defines methods for representing the attributes and associations of an object in the Extensible Markup Language (XML). It is actually a meta-protocol based on XML that can be used to define new protocols within a clearly defined, but flexible framework.

Web-Services are designed to be accessed by software. In the currently prevalent data-centric software environment they are generally clients to the middleware of data sources. The middleware collects the required data and sends it back to the Web-Service, which reformats the data using the SOAP protocol and passes it onto the requester. Depending on its original specifications, the requesting application will have the data downloaded on disk or receive it directly on-line. If the Web-Service is a data-centric application then a data-to-data translation must be performed in much the same way as would be necessary when passing data between two data-centric applications (Figure 9). In the case of an information-centric Web-Service a data-to-information translation is performed when the Web-Service receives data from an external source and an information-to-data translation is performed whenever the Web-Service sends information through the Web Server (Figure 10).

Exposing the data sources within the data layer and the information-centric components of the information management layer of a knowledge management system (Figures 8) to a Web-Services environment provides a means of integrating and conveniently accessing a heterogeneous set of software applications. By treating these applications as Web-Services and advertising these services in a registry enables the implementation of client applications that can utilize functionality from multiple applications (i.e., Web-Services). Clients can discover services based on service type, rather than being restricted to a specific service at a known location. The use of SOAP and other XML-based languages for communication frees both server and clients from dependence on a particular programming language or operating system.

Conclusion

We have entered a period of transformation with several dominant traits that are individually distinct and yet, on deeper examination, appear to be closely related. Separately, they are readily discernable as the enablement and increased value associated with each individual person, the flattening of organizational structures, and the elevation of the computer to the role of an intelligent assistant in an emerging human-computer partnership. However, considered in conjunction they have a common thread.

While the capabilities of the individual are being significantly increased by the availability of more and more powerful computers and faster communication networks, it is the skill that the individual acquires to utilize these enabling facilities that largely determines the value of the individual to the organization. To take advantage of the enabled individual, organizations have had to adapt both in terms of their structural model and management practices. Clearly, persons with powerful tools, expert skills to use these tools, and confidence in their abilities, will demand a high degree of autonomy, a share in the decision making process, and the freedom to exercise their initiative. As the potential value of the contributions made by the individual person increases there is likely to be greater concern by the organization to capture the information and knowledge that is being generated by all of the contributors in the organization.

Soon the volume of information generated by the organization increased to the point where it could no longer be maintained by the human contributors who were, in any case, busy generating more information. It became necessary to utilize the computer to assist in the management of the informational resources of the organization. While initially these management functions could be conveniently divided into the data-processing tasks undertaken by the computer and the interpretation of information into knowledge undertaken by its human users, over time even the information interpretation component became overwhelming.

Closer examination of the data-processing bottleneck has drawn attention to the fundamental difference between data and information, and the need to represent information rather than data in software applications. In this way, by providing context, information-centric software is able to support intelligent tools (i.e., software agents) with reasoning capabilities. The implications are profound and represent a paradigm shift. The role of the computer is being transformed from a visualization and computing device with no 'understanding' of what it is processing, to an intelligent assistant that is able to make intellectually meaningful and useful contributions to its human users. In this respect the new knowledge management paradigm is a natural outcome of the gradual merging of human and computer capabilities into a collaborative partnership.

References

Biermann A. and J. Feldman (1972); 'A Survey of Results in Grammatical Inference'; Academic Press, New York.

Cohen B. L. and C. A. Sammut (1978); 'Pattern Recognition and Learning With a Structural Description Language'; Proceedings Fourth International Joint Conference on Artificial Intelligence, IJCPR, Kyoto, Japan (pp.394).

Holland J. H. (1988); 'The Global Economy as an Adaptive Process'; in Anderson P., K. Arrows and D. Pines, The Economy as an Evolving Complex System, Proceedings Evolutionary Paths of the Global Economy Workshop, September 1987, Santa Fe Institute Workshop, Santa Fe, New Mexico (pp. 117-124).

Humphries M., M. W. Hawkins and M.C. Dy (1999); 'Data Warehousing: Architecture and Implementation'; Prentice Hall, Upper Saddle River, New Jersey.

Kennedy J. and R. C. Eberhart (2001); 'Swarm Intelligence'; Morgan Kaufmann, San Francisco, California.

Koza J. (1992); 'Genetic Programming'; MIT Press, Cambridge, Massachusetts.

McClelland J. L. and D. E. Rumelhart (1988); 'Explorations in Parallel Distributed Processing: A Handbook of Models, Programs and Exercises'; MIT Press, Cambridge, Massachusetts.

Winston P. H. (1970); 'Learning Structural Descriptions from Examples'; Technical Report AI-TR-231, MIT, Cambridge, Massachusetts, September.

Wooldridge M. and N. Jennings (1995); 'Intelligent Agents: Theory and Practice'; The Knowledge Engineering Review, 10(2) (pp.115-152).

Appendix – Glossary of Terms

Data: Numbers and words without relationships. Even though data are often stored in a relational database management system, typically only minimal relationships are stored with the data. Without adequate relationships, data do not contain sufficient context to support automatic reasoning capabilities by software agents.

Data-Centric: Software that incorporates an internal representation of data (i.e., number and words) with few (if any) relationships. Although the data may be represented as objects the lack of relationships, and therefore the absence of context, inhibits the inclusion of meaningful and reliable automatic reasoning capabilities. Data-centric software, therefore, must largely rely on predefined solutions to predetermined problems, and has little (if any) scope for adapting to real world problems in near real-time. Communication between data-centric software applications is typically restricted to the passing of data-string messages from one application to the other. This imposes a larger transmission load than communication between information-centric applications. Since a data-centric application has no 'understanding' of the data that it is processing, a complete set of data must be transmitted so that the receiving application can process the transferred data in the appropriate predefined manner. For example, if the data to be transmitted involves the new location of an automobile then a complete set of data describing the automobile (including its new location) must be transmitted. In the case of information-centric applications only the new location and some object identifier would need to be transmitted, because both the transmitting and receiving applications have some 'understanding' of the general notion of an automobile and the specific instance of that notion representing the particular automobile that has changed its location.

Information: Data with relationships to provide adequate context for the interpretation of the data. The richer the relationships the greater the context, and the more opportunity for automatic reasoning by software agents.

Information-Centric: Software that incorporates an internal information model (i.e., ontology) consisting of objects, their characteristics, and the relationships among those objects. The information model is a virtual representation of the real world domain under consideration and is designed to provide adequate context for software agents (typically rule-based) to reason about the current state of the virtual environment. Since information-centric software has some 'understanding' of what it is processing it normally contains tools rather than predefined solutions to predetermined problems. These tools are commonly software agents that collaborate with each other and the human user(s) to develop solutions to problems in near real-time as they occur. Communication between information-centric applications is greatly facilitated since only the changes in information need to be transmitted. This is made possible by the fact that the object, its characteristics and its relationships are already known by the receiving application.

Context: Meaning conveyed by the combination of data with relationships.

Data Portal: Provides access to operational data, with an emphasis on the presentation of data (usually to human users). Data Portals may also incorporate data analysis tools, and are often accessed in a Web-Services (e.g., Internet) environment. A Data Portal typically does not store data but provides access to data sources that contain stored data.

Data Warehouse: Stores and manages summarized (i.e., archived) data, usually in a relational database management system. The summarized data are periodically updated according to a predefined timeline. Data Warehouses often employ sophisticated data indexing mechanisms (e.g., based on key word indexing schemas) to facilitate the rapid retrieval of data.

Data Mart: A subset of the data stored in a Data Warehouse that is focused on a particular functional area.

OLAP: On Line Analytical Processing (OLAP) tools extract answers to Who?, What?, and Why? queries, constrained by the very limited (if any) context provided in a Data Warehouse (or Data Mart).

Data Mining: Data Mining tools analyze the data in a Data Warehouse (or Data Mart) to establish relationships, identify trends, and predict future trends.

Ontology: An information structure, rich in relationships, that provides a virtual representation of some real world environment (e.g., the context of a problem situation such as the management of a transport corridor, the loading of a cargo ship, the coordination of a military theater, the design of a building, and so on). The elements of an ontology include objects and their characteristics, different kinds of relationships among objects, and the concept of inheritance.

Collaborative Agents: Software modules that are capable of reasoning about events (i.e., changes in data received from external sources or as the result of internal activities) within the context of the information contained in the internal information model (i.e., ontology). The agents collaborate with each other and the human users as they monitor, interpret, analyze, evaluate, and plan alternative courses of action.