The Meaning of an Information-Centric Computer Environment

Jens Pohl
Collaborative Agent Design Research Center
Cal Poly, San Luis Obispo, CA, USA

It is often lamented that we human beings are suffering from an ***information overload***. This is a myth, as shown in Fig.1 there is no information overload. Instead we are suffering from a data overload. The confusion between data and information is not readily apparent and requires further explanation. Unorganized data are voluminous but of very little value. Over the past 15 years, industry and commerce have made significant efforts to rearrange this unorganized data into purposeful data, utilizing various kinds of database management systems. However, even in this organized form, we are still dealing with data and not information.

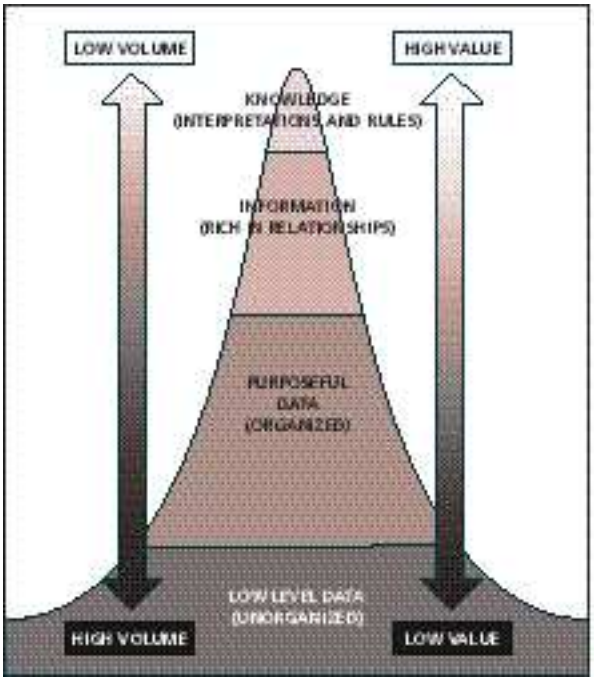

Fig.1: The ***information overload*** myth

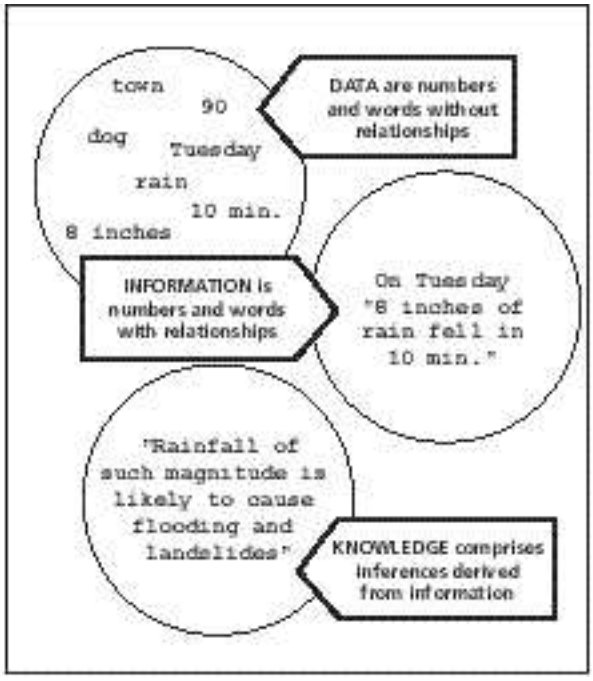

Fig.2: Data, information and knowledge

Data are defined as numbers and words without relationships. In reference to Fig.2, the words “town”, “dog”, “Tuesday”, “rain”, “inches”, and “min”, have little if any meaning without relationships. However, linked together in the sentence "On Tuesday, 8 inches of rain fell in 10 min." they become information. If we then add the context of a particular geographical region and historical climatic records, we could perhaps infer that "Rainfall of such magnitude is likely to cause flooding and landslides." This becomes knowledge.

Context is normally associated solely with human cognitive capabilities. Prior to the advent of computers, it was entirely up to the human agent to convert data into information and to infer knowledge through the addition of context. However, the human cognitive system performs this

function subconsciously (i.e., automatically); therefore, prior to the advent of computers, the difference between data and information was an academic question that had little practical significance in the real world of day-to-day activities. As shown in Fig.3, the intersection of the data, human agent and context realms provides a segment of immediately relevant knowledge.

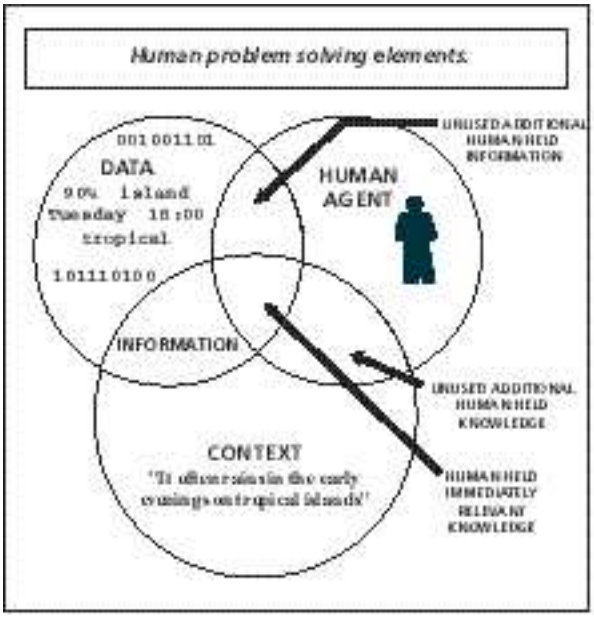

Fig.3: Unassisted problem solving

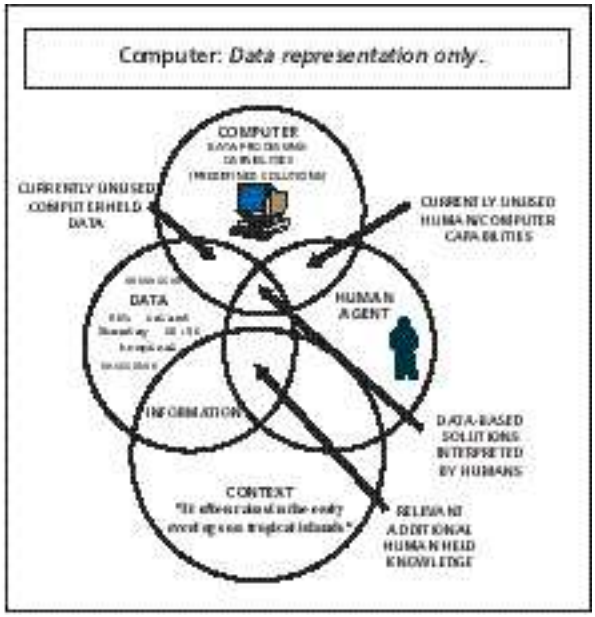

Fig.4: Limited data-processing assistance

When computers entered on the scene, they were first used exclusively for processing data. In fact, even in the 1980s computer centers were commonly referred to as data-processing centers. It can be seen in Fig.4 that the context realm remained outside the computer realm. Therefore, the availability of computers did not change the need for the human agent to interpret data into information and infer knowledge through the application of context. The relegation of computers to data-processing tasks is the underlying reason why even today, as we enter the 21st Century, computers are still utilized in only a very limited decision-support role. As shown in Fig.5, in this limited computer-assistance environment human decision makers typically collaborate with each other utilizing all available communication modes (e.g., telephone, FAX, e-mail, letters, face-to-face meetings). Virtually every human agent utilizes a personal computer to assist in various computational tasks. While these computers have some data sharing capabilities in a networked environment, they cannot directly collaborate with each other to assist the human decision makers in the performance of decision-making tasks. Each computer is typically limited to providing relatively low-level data-processing assistance to its owner. The interpretation of data, the inferencing of knowledge, and the collaborative teamwork that is required in complex decision-making situations remains the exclusive province of the human agents. In other words, without access to information and at least some limited context, the computer cannot participate in a distributed collaborative problem-solving arena.

In this context, it is of interest to briefly trace the historical influence of evolving computer capabilities on business processes and organizational structures. When the computer first became more widely available as an affordable computational device in the late 1960s, it was applied immediately to specialized numerical calculation tasks such as interest rate tables and

depreciation tables (Fig.6). During the early 1970s, these computational tasks broadened to encompass bookkeeping, record storage, and report generation. Tedious business management functions were taken over by computer-based accounting and payroll applications. By the late 1970s, the focus turned to improving productivity using the computer as an improved automation tool to increase and monitor operational efficiency.

In the early 1980s (Fig.7), the business world had gained sufficient confidence in the reliability, persistence, and continued development of computer technology to consider computers to be a permanent and powerful data-processing tool. Accordingly, businesses were willing to reorganize their work flow as a consequence of the functional integration of the computer. More comprehensive office management applications led to the restructuring of the work flow.

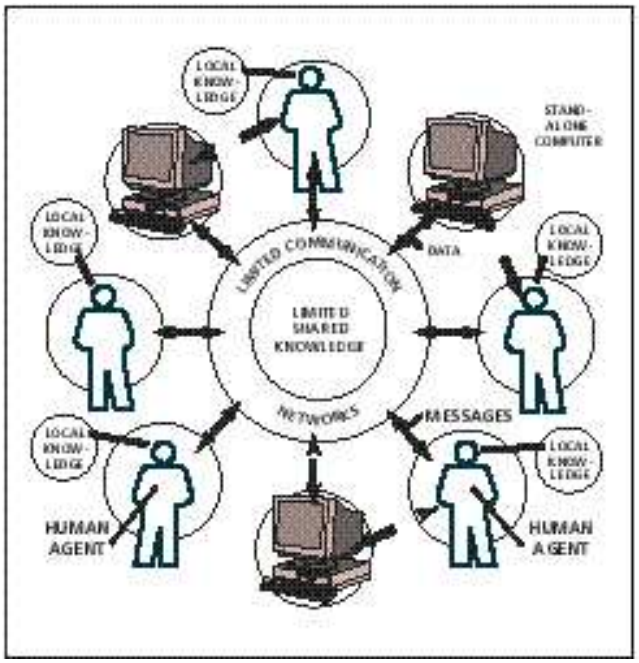

Fig.5: Limited computer assistance

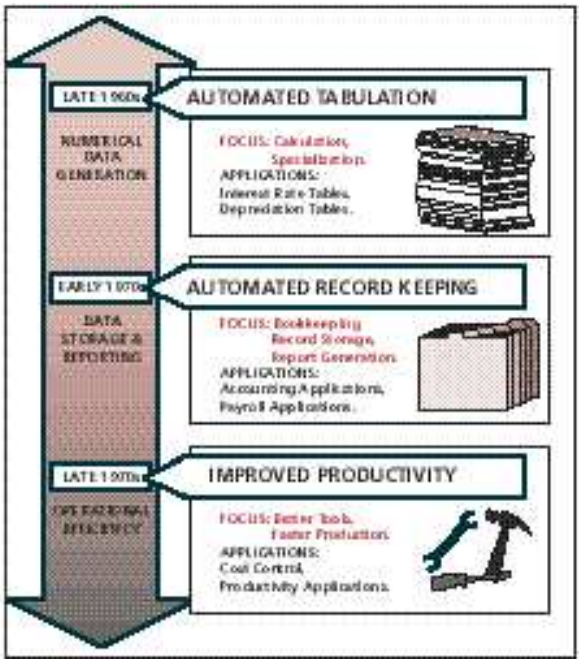

Fig.6: Evolution of business intelligence (A)

By the late 1980s, this had led to a wholesale re-engineering of the organizational structure of many businesses with the objective of simplifying, streamlining, and downsizing. It became clear that many functional positions and some entire departments could be eliminated and replaced by integrated office automation systems. During the early 1990s, the problems associated with massive unorganized data storage became apparent, and with the availability of much improved database management systems, data were organized into mostly relational databases. This marked the beginning of ordered-data archiving and held out the promise of access to any past or current data and reporting capabilities in whatever form management desired.

However, by the mid 1990s (Fig.8), the quickening pace of business in the light of greater competition increased the need for a higher level of data analysis, faster response, and more accurate pattern detection capabilities. During this period, the concepts of data-warehouses, data-marts, and On-Line Analytical Processing (OLAP) were conceived and rapidly

implemented (Humphries et al. 1999). Since then, the term 'business intelligence' has been freely used to describe a need for the continuous monitoring of business trends, market share, and customer preferences.

In the late 1990s, the survival pressure on business increased with the need for real-time responsiveness in an Internet-based global e-commerce environment. By the end of the 20th Century, business began to seriously suffer from the limitations of a data-processing environment. The e-commerce environment presented attractive opportunities for collecting customer profiles for the implementation of on-line marketing strategies with enormous revenue potential. However, the expectations for automatically extracting useful information from low-level data could not be satisfied by the methods available. These methods ranged from relatively simple keyword and thematic indexing procedures to more complex language-processing tools utilizing statistical and heuristic approaches (Denis 2000, Verity 1997).

The major obstacle confronted by all of these information-extraction approaches is the unavailability of adequate context (Pedersen and Bruce 1998). As shown previously in Fig.4, a computer-based data-processing environment does not allow for the representation of context. Therefore, in such an environment, it is left largely to the human user to interpret the data elements that are processed by the computer.

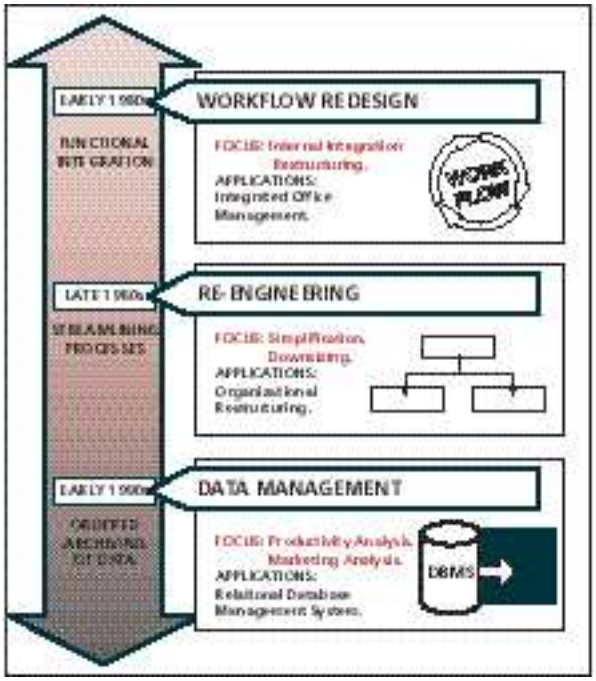

Fig.7: Evolution of business intelligence (B)

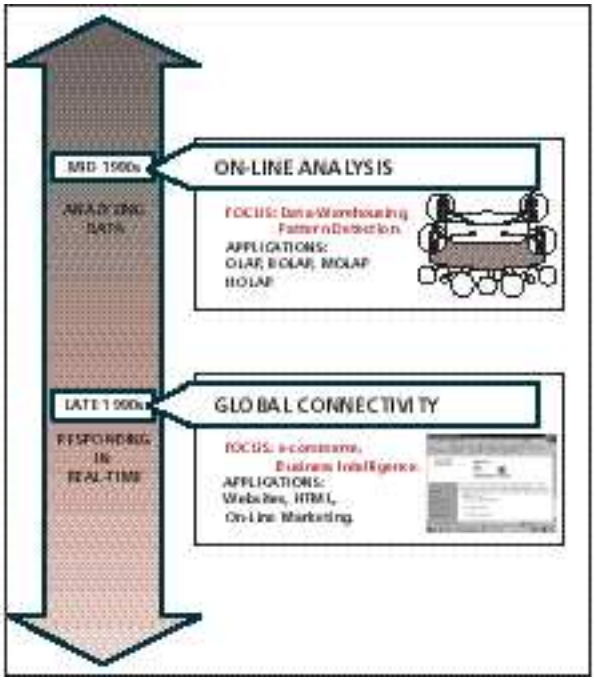

Fig.8: Evolution of business intelligence (C)

Methods for representing information and knowledge in a computer have been a subject of research for the past 40 years, particularly in the field of 'artificial intelligence' (Ginsberg 1993). However, these studies were mostly focussed on narrow application domains and did not generate wide-spread interest even in computer science circles. For example even today, in the year 2000, it is difficult to find an undergraduate computer science degree program in the USA

that offers a core curriculum class dealing predominantly with the representation of information in a computer.

The Representation of Information in a Computer

Conceptually, to represent information in a computer, it is necessary to move the context circle in Fig.4 upward into the realm of the computer (Fig.9). This allows data to enter the computer in a contextual framework, as information. The intersection of the data, context, and human agent circles provide areas in which information and knowledge are held in the computer. The prevailing approach for the practical implementation of the conceptual diagram shown in Fig.9 is briefly outlined below. As discussed earlier (Fig.2), the principal elements of information are data and relationships. We know how data can be represented in the computer but how can the relationships be represented? The most useful approach available today is to define an ontology of the particular application domain in the form of an object model. This requires the identification of the objects (i.e., elements) that play a role in the domain and the relationships among these objects (Fig.10). Each object, whether physical (e.g., car, person, building, etc.) or conceptual (e.g., event, privacy, security, etc.) is first described in terms of its behavioral characteristics. For example, a ***car*** is a kind of ***land conveyance***. As a child object of the ***land conveyance*** object, it automatically inherits all of the characteristics of the former and adds some more specialized characteristics of its own (Fig.11). Similarly, a ***land conveyance*** is a kind of ***conveyance*** and therefore inherits all of the characteristics of the latter. This powerful notion of inheritance is well supported by object-oriented computer languages such as C++ (Stroustrup 1987) and Java (Horstmann and Cornell 1999).

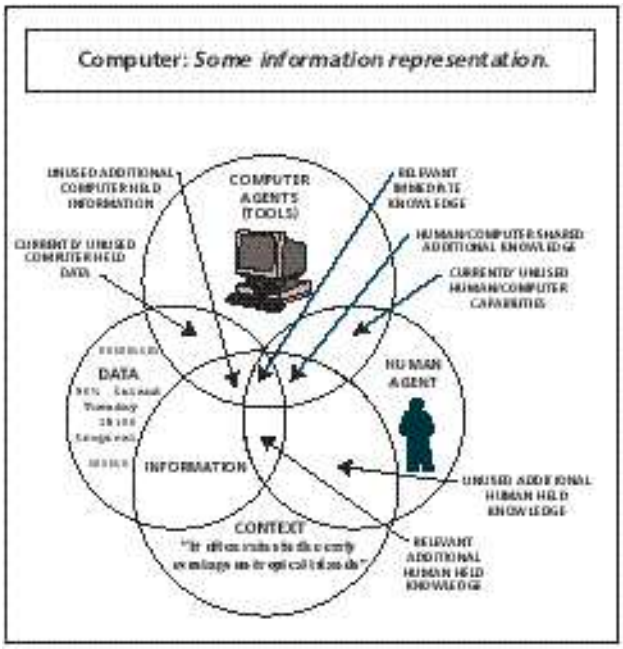

Fig.9: Early human-computer partnership

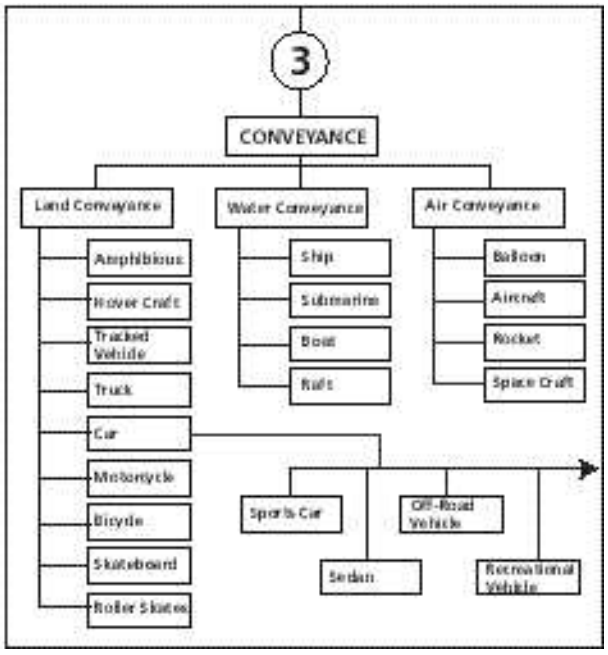

Fig.10: Branch of a typical object model

However, even more important than the characteristics of objects and the notion of inheritance are the relationships that exist between objects. As shown in Fig.12, a car incorporates many components that are in themselves objects. For example, cars typically have engines, steering

systems, electric power units, and brake systems. They utilize fuel and often have an air-conditioning system.

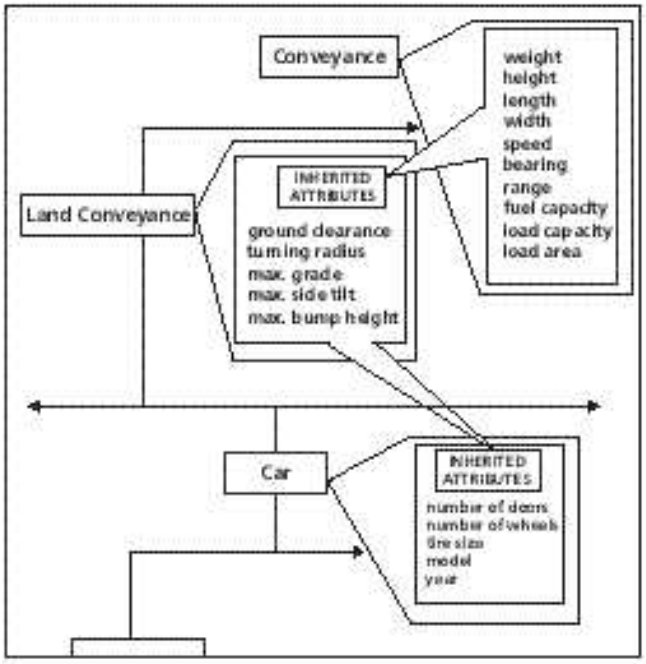

Fig.11: Object model - ***inheritance***

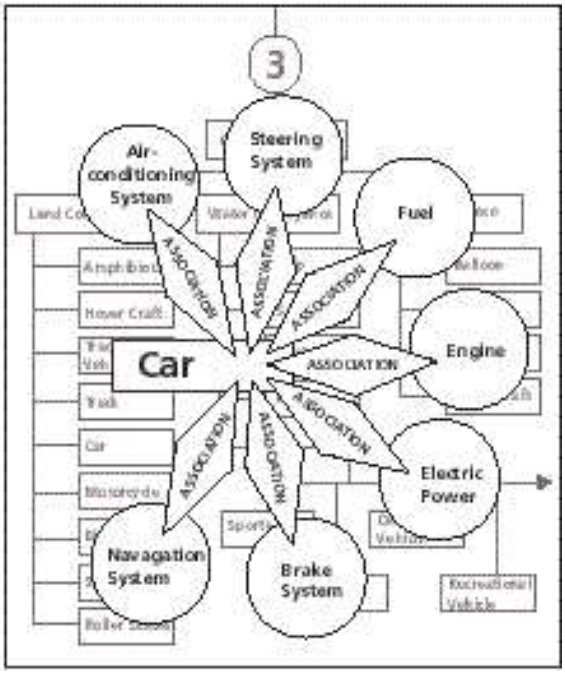

Fig.12: Object model - ***associations***

For several reasons, it is advantageous to treat these components as objects in their own right rather than as attributes of the car object. First, they may warrant further subdivision into parent and child objects. For example, there are several kinds of air-conditioning systems, just as there are several kinds of cars. Second, an air-conditioning system may have associations of its own to other component systems such as a temperature control unit, a refrigeration unit, an air distribution system, and so on. Third, by treating these components as separate objects we are able to describe them in much greater detail than if they were simply attributes of another object. Finally, any changes in these objects are automatically reflected in any other objects that are associated to them. For example, during its lifetime, a car may have its air-conditioning system replaced with another kind of air handling unit. Instead of having to change the attributes of the car, we simply delete the association to the old unit and add an association to the new unit. This procedure is particularly convenient when we are dealing with the association of one object to many objects, such as the wholesale replacement of a cassette tape player with a new compact disk player model in many cars, and so on.

The way in which the construction of such an ontology leads to the representation of information (rather than data) in a digital computer is described in Fig.13, as follows. By international agreement, the American Standard Code for Information Interchange (ASCII) provides a simple binary (i.e., digital) code for representing numbers, alphabetic characters, and many other symbols (e.g., +, -, =, ( ), etc.) as a set of ***0*** and ***1*** digits. This allows us to represent sets of characters such as the sentence ***"Police car crossing bridge at Grand Junction."*** in the computer. However, in the absence of an ontology, the computer stores this set of characters as a meaningless text string (i.e., data). In other words, the computer has no understanding at all of

the meaning of this sentence. As discussed previously, this is unfortunately the state of e-mail today. While e-mail has become a very convenient, inexpensive, and valuable form of global communication, it depends entirely on the human interpretation of each e-mail message by both the sender and the receiver.

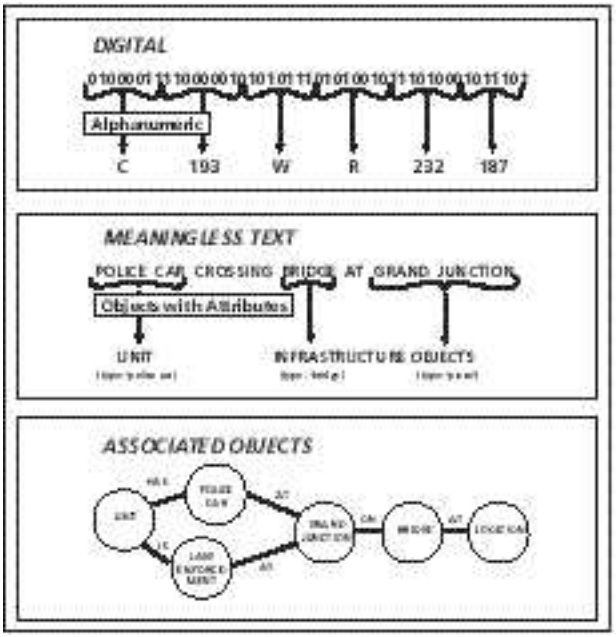

Fig.13: From ***digital*** to ***information***

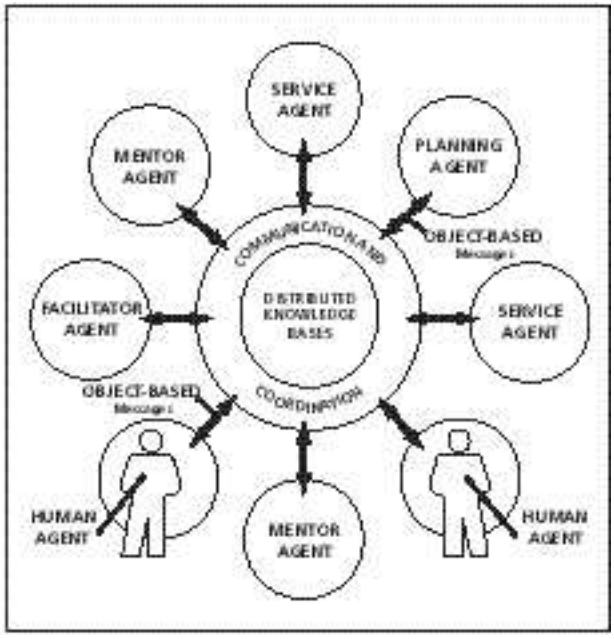

Fig.14: Types of agents

Now, if the ***"Police car crossing bridge at Grand Junction."*** message had been sent to us as a set of related objects, as shown at the bottom of Fig.13, then it would be a relatively simple matter to program computer-based agents to reason about the content of this message and perform actions on the basis of even this limited level of understanding. How was this understanding achieved? In reference to Fig.13, the police car is interpreted by the computer as an instance of a ***car*** object which is associated with a ***civilian organization*** object of kind ***police***. The ***car*** object automatically inherits all of the attributes of its parent object, ***land conveyance***, which in turn inherits all of the attributes of its own parent object, ***conveyance***. The *car* object is also associated with an instance of the infrastructure object, ***bridge***, which in turn is associated with a ***place*** object, ***Grand Junction***, giving it a geographical location. Even though this interpretational structure may appear primitive to us human beings, it is adequate to serve as the basis of useful reasoning and task performance by computer-based agents.

Such agents may be programmed in many ways to serve different purposes (Fig.14). Mentor agents may be designed to serve as guardian angels to look after the welfare and represent the interests of particular objects in the underlying ontology. For example, a mentor agent may simply monitor the fuel consumption of a car or perform more complex tasks such as helping a tourist driver to find a particular hotel in an unfamiliar city, or alerting a platoon of soldiers to a hostile intrusion within a specified radius of their current position in the battlefield (Pohl et al. 1999). Service agents may perform expert advisory tasks on the request of human users or other agents. For example, a computer-based daylighting consultant can assist an architect during the design of a building (Pohl et al. 1989) or a Trim and Stability agent may continuously monitor

the trim of a cargo ship while the human cargo specialist develops the load plan of the ship (Pohl et al. 1997). At the same time, Planning agents can utilize the results of tasks performed by Service and Mentor agents to devise alternative courses of action or project the likely outcome of particular strategies. Facilitator agents can monitor the information exchanged among agents and detect apparent conflicts (Pohl 1996). Once such a Facilitator agent has detected a potential non-convergence condition involving two or more agents, it can apply one of several relatively straightforward procedures for promoting consensus, or it may simply notify the user of the conflict situation and explain the nature of the disagreement.

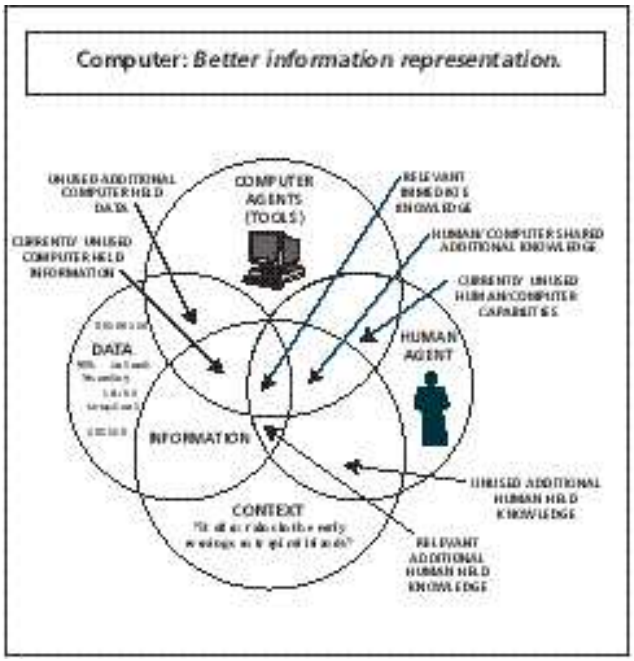

Fig.15: Evolving human-computer partnership

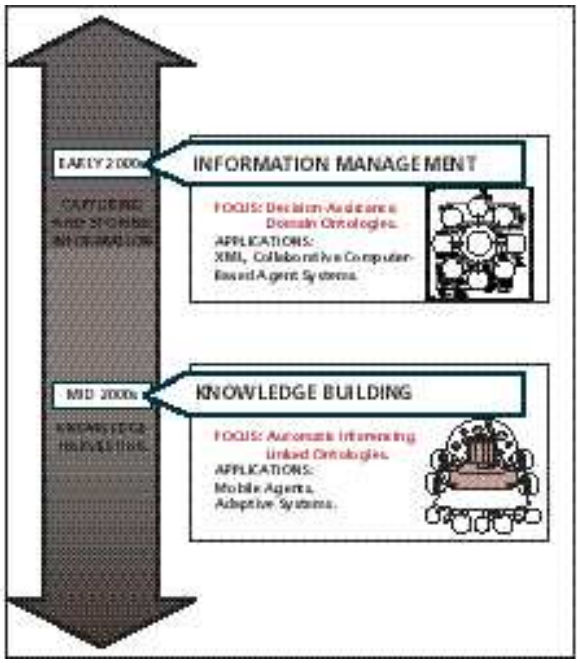

Fig.16: Evolution of business intelligence (D)

Conclusion

While the capabilities of present day computer-based agent systems are certainly a major advancement over data-processing systems, we are only at the threshold of a paradigm shift of major proportions. Over the next several decades, the context circle shown in Fig.15 will progressively move upward into the computer domain, increasing the sector of "relevant immediate knowledge" shared at the intersection of the human, computer, data, and context domains. Returning to the historical evolution of business intelligence described previously in reference to Figs. 6, 7 and 8, the focus in the early 2000s will be on information management as opposed to data-processing (Fig.16). Increasingly, businesses will insist on capturing data as information through the development of business enterprise ontologies and leverage scarce human resources with multi-agent software capable of performing useful analysis and pattern-detection tasks. Toward the mid 2000s, we can expect some success in the linking of these ontologies to provide a virtually boundless knowledge harvesting environment for mobile agents with many kinds of capabilities. Eventually, it may be possible to achieve virtual equality between the information representation capabilities of the computer and the human user. This virtual equality is likely to be achieved not by the emulation of human cognitive capabilities, but

rather, through the skillful combination of the greatly inferior artificial cognitive capabilities of the computer with its vastly superior computational, pattern-matching and storage facilities.

References

Denis S. (2000); 'Numbers'; Intelligent Enterprise, April 10 (pp. 37-44).

Ginsberg M. (1993); 'Essentials of Artificial Intelligence'; Morgan Kaufmann, San Mateo, California.

Horstmann C. and G. Cornell (1999); 'Core Java'; Vol. 1 and 2, Sun Microsystems Press, Prentice Hall, Upper Saddle River, New Jersey.

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

Pedersen T. and R. Bruce (1998); 'Knowledge Lean Word-Sense Disambiguitization'; Proceedings 5th National Conference on Artificial Intelligence, July, Madison WI.

Pohl J., M. Porczak, K.J. Pohl, R. Leighton, H. Assal, A. Davis, L. Vempati and A. Wood, and T. McVittie (1999); 'IMMACCS: A Multi-Agent Decision-Support System'; Technical Report, CADRU-12-99, CAD Research Center, College of Architecture and Environmental Design, Cal Poly, San Luis Obispo, California, August.

Pohl J., A. Chapman, K. Pohl, J. Primrose and A. Wozniak (1997); 'Decision-Support Systems: Notions, Prototypes, and In-Use Applications'; Technical Report, CADRU-11-97, CAD Research Center, Design Institute, College of Architecture and Environmental Design, Cal Poly, San Luis Obispo, California, January.

Pohl J., L. Myers, A. Chapman and J. Cotton (1989); 'ICADS: Working Model Version 1'; Technical Report, CADRU-03-89, CAD Research Center, Design Institute, College of Architecture and Environmental Design, Cal Poly, San Luis Obispo, California.

Pohl K.(1996); 'KOALA: An Object-Agent Design System'; in Pohl J. (ed.) Advances in Cooperative Environmental Design Systems, Focus Symposium: International Conference on Systems Research, Informatics and Cybernetics, Baden-Baden, Germany, Aug.14-18 (pp.81-92).

Stroustrup B. (1987); 'The C++ Programming Language'; Addison-Wesley, Reading, Massachusetts.

Verity J. (1997); 'Coaxing Meaning Out of Raw Data'; Business Week, June 15.