

Getting Data to Think Like Us

We possess the technology to capture and process vast amounts of information. The problem now becomes one of effectively using what we have. There are plenty of tools and products currently on the market to address this issue of “business intelligence”- SQL, OLAP, metadata repositories, report writers, ETL (extract, transform, load) utilities, ad hoc queries, experts systems, data mining, ... The list goes on and on. One can be immersed in a sea of data with all the latest tools yet not be able to get what one needs. Or to paraphrase from the Ancient Mariner, “Data, data everywhere yet none of it fit to think”. Why is this?

There are two major reasons. The first is that our underlying data models are inappropriate for the task of business intelligence. This has led to a proliferation of specialty products, each trying to address some analysis problem. I liken the current situation to that of astronomy 500 years ago. Back then the underlying “data model” was that the earth was the center of the universe. Advancing technology permitted astronomers to compile more accurate data on the movements of the planets and stars. The simple model of the earth at the center of ten perfect spheres was no longer tenable. Rather than question the basic assumptions, models were “enhanced” with additional spheres, off-centered spheres, and spheres with retrograde motions. All this to justify the belief that the earth was the center of the universe. In time it became generally accepted that the earth and planets revolved around the sun. This change in perspective enabled Kepler and Newton to formulate their simple and elegant laws of motion and gravity. Current database dogma likewise locks us in a similar position. Instead of rethinking the problem, we try to “fix” it with more and more specialized products.

The second reason arises from the differences between the data stored in computerized databases and the data stored in our heads. Man and machine are not well coupled. Computers are fast, accurate, and inflexible. People are slow, intuitive, and adaptable. The engineering phrase for this poor coupling is “impedance mismatch”. This makes it difficult for us to express our needs to the machine, and for the machine to communicate results back to us.

Our current data models are incorrect; our man-machine-data interfaces are poor. How can we best improve this state of affairs? We can start by stepping back and looking at the big data picture; how we use data, not only in databases, but also in our lives. We need to understand that the center of our data universe is not the relational database, but something much broader. Only then can we begin to formulate new laws of data.

Virtually all databases store *atomic* data. By this, I mean that each data element stands on its own. Individual data points may be aggregated for implementation efficiencies but, at least in theory, could be stored anywhere in a database. For example, the sales of a product in 1999 could be stored before or after the sales of that product in 2000. Or it could be stored in with data about customers. It does not matter where or how, as long as the database can access it as needed.

A *sequence* is an ordered collection of atomic data. A sequence can be something as simple as the spelling of a word (W O R D) or the letters of the alphabet (A B C ... Z). In a sequence, the ordering is important. In some sense, the ordering is the data! Sequences are becoming more prevalent as online voice, music, and video grow in popularity. Another form of sequence is a procedure— a sequence of steps or instructions to accomplish a defined task. Computer programs are sequences of statements which are compiled into more detailed sequences of machine instructions.

It is one thing to be able to *represent* atomic and sequential data. Decision making (the key to business intelligence) depends, in part, on timely data. More importantly, decision making depends on the ability to *recognize* patterns, recognize possible consequences, to draw conclusions. The data necessary to represent a thing or concept is different from the data required to recognize it. There are probably many things you recognize yet are unable to communicate precisely how you are able to do so.

Below is a simple two-by-two matrix of these data concepts- atomic and sequential versus representation and recognition. The matrix entries contain examples of data that correspond to the row-column properties. Traditional databases form the basis of atomic representation. Recognition of atomic data takes on many forms including data mining (finding patterns in data) and expert systems (recognizing a solution given a set of pre-conditions). Computer programs are representations of sequential data. Video and sound are good examples of time-varying sequences. Recognition of sequences is still in its infancy. A simple recognizer would be the compiler for a programming language. Voice recognition and time varying scene recognition are currently hot research topics.

Getting Data to Think Like Us

	Atomic	Sequential
Representation	Traditional Databases	Programs Multi-media
Recognition	Data Mining Expert Systems	Compilers Voice Recognition

We think of data as hard facts. Data can also be procedural. What is “ 48.3×10 ”? You recognize this as an arithmetic expression, one involving multiplying by ten. You know the answer, not from memorization of a fact but from memorization of a procedure that states “multiplying by 10 is done by shifting the decimal place to the right”.

Data is not always explicitly defined. Data can be given as general rules. If I were to show you a horse and ask you if it could fly, your answer would be no. The same question asked of another horse would also be answered with a negative. You have no specific facts about these horses. You have general rules, one of which is that all horses cannot fly. My next question is “What is Pegasus?” The correct answer is a flying horse! So not only do we have general rules but often have exceptions to the rules.

What does “7 8 9” mean to you? Just a number or possibly a short sequence of numbers? If I preface the question with, “Why is 6 afraid of 7?” the answer becomes “because 7 8 9!” This is an example of context influencing our recognition of data. Everything we see, hear, say, and do is influenced by context. We use context to maintain contradictions (horses fly in mythology). We use context to handle exceptions to rules (“i” before “e” except after “c”). We use context to build pretend worlds (let’s assume that our costs decrease 5% next year). We use context when giving information (“sales” when talking to Jim means sales of his product line, “sales” when talking to the CFO means sales for the entire corporation). The notion of context has far reaching consequences. Research in code reusability (how can this code be used in different contexts?), modeling (how does the model react to different contexts?), optimizations (what is the optimal context?), expert systems (what is the problem/solution given the parameters of the current context?), and artificial intelligence (how should an autonomous agent react in this context?) would benefit from a formal understanding of context.

MKS Inc. has taken all of the above mentioned ideas and developed them into an integrated theory- the context sensitive multidimensional space (CSMD). It uses a single representation for atomic, sequential, and recognition data. All information within the multidimensional space is both defined and accessed using a single construct. Rules and exceptions to those rules are supported. The declaration of a rule is independent of the declaration of its exceptions (e.g. an exception may be declared before the rule). A database entry may be defined as a literal value or as a sequence of procedural steps. And most importantly, *all interpretation is contextual*. V4™ is an implementation of the CSMD model. V4 is a real product and has been successfully used in a wide range of applications.

There are many obvious benefits to be gained by the CSMD model. Not surprisingly, there have been several unexpected benefits. Conciseness is the first. Analyses performed with V4 are more concise than corresponding SQL/4GL solutions (see “The V4 Challenge” for examples). The ability to relate data from many different sources is another unexpected benefit. Legacy, ERP, third party data, and incremental data (e.g. daily sales data) can be easily integrated into a single CSMD space. The third benefit is the ability to merge descriptive data (metadata) into a CSMD space allowing simple interfaces to complex data.

Providing effective business intelligence to today’s management is a real problem in need of a real solution. MKS has addressed this issue from both a theoretical and engineering perspective. The development of a new, expanded theory of data has led to an elegant theory and the implementation of a powerful product.

V4™ - Data that thinks like us

Victor E. Hansen, President
MKS Inc.
610 989 9905
veh@mksinc.com