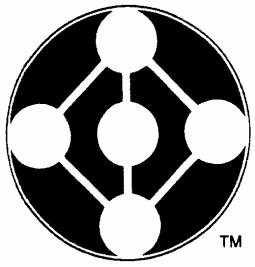

Alternative Technologies

Inventing Integrity

Imagine being the inventor of the technical foundations for a software and hardware industry with yearly revenues in the tens of billions of dollars year after year, and with no end in sight. Imagine the fame, fortune, respect, and sense of accomplishment accrued. Reflections on just such a possibility triggered by recent events force me to depart from my scheduled activities. I hope my reader's will stay with and forgive this brief but important intrusion while I address the value of a disruptive invention and its inventor.

As a disruptive invention, our hypothetical technology would spawn new markets with a wide range of products and services. The obvious products and customer support would deliver the invention. Revenue from software (DDD – design, development, and deployment) and hardware (servers and peripherals) products, educational services, conferences, books, and journals, DDD consulting services, and of course industry and financial analyses would emerge. A host of derivative and enabled technologies, along with business infrastructures and extended relationships, would arise. Finally, there would be business consulting as managers struggled to develop business strategies that would anticipate the technology trends and capabilities. All would contribute to an economy of considerable value. Then too there is the ROI obtained by user's over the life of the new technology and all its derivatives (not just over the life of products purchased).

One such invention was the relational model from which all SQL products and technologies are derived. Few realize how far-reaching this invention's implications have been, let alone their continued future impact. I won't provide a complete analysis, but I hope I can open your eyes and encourage a bit of appreciative reflection. Consider the following points, organized as "Without" and "With" the invention of the relational model:

- ***Without:*** Applications were much more costly and more difficult to design, development, and maintain. The data for every application (accounting, MRP, shipping, etc.) were captive to custom data structures, even when managed in a non-relational DBMS. This greatly limited the number, variety, and scope (department vs. enterprise) of applications. ***With:*** Departmental and enterprise software systems (custom and packaged) proliferated. Imagine SAP, Seibel, Peoplesoft, etc., or even e-commerce and Internet sites without databases like Oracle, DB2, Sybase ASE, SQL Sever, or MySQL!

- ***Without:*** Applications and application modules only rarely, and with great effort, took advantage of a common database schema, let alone the possibility of data integration. ***With:*** Applications designed on a shared database schema are common place. Application integration technologies for data integration were as necessary to EAI as messaging. Think integration brokers, data transformation hubs, and ODBC/JDBC adapters.

- ***Without:*** Data administration was more costly, and was difficult to learn due to the tight cohesion between applications and stored data structures. Physical data schema modification, expansion, and performance tuning were extremely limited and required excessive project planning. This fact alone limited typical database sizes to mere megabytes and their scope to departments. ***With:*** We now readily accept plans for terabyte databases and consider those in the petabyte range because it is *administratively* feasible and query optimizers eliminate hand coding – due primarily to relational technology (even with hardware cost and performance affects acknowledged). Imagine data marts and data warehouses without relational!

- ***Without:*** The use of reporting, querying, and analysis tools was generally limited to computer professionals trained in a data *programming* language or the use of costly ad-hoc tools with cryptic interfaces and simplistic functionality. We called them "end-users" because their business agenda was primary. This limited management, monitoring, and analysis of business data to batch processes. ***With:*** Online, *ad-hoc* decision support tools abound that can be used by unsophisticated end-users (often a computer phobic non-programmer). We now consider reducing the delay between business events and analysis of associated data to real-time. Think BI, BAM, BPMS, closed loop DSS, and (relational) OLAP – all thriving!

Invented before software patents, the relational model earned no royalties or license fees. Neither SQL DBMS vendors nor any other beneficiaries (software, server, and storage vendors, and big consulting firms) shared their wealth with the man who made it possible, Dr. E. F. Codd. The debt this industry owes is enormous, a debt it's now too late to repay: Dr. Codd died April 18, 2003 at age 79. I offer this farewell salute to a professional acquaintance who occasionally offered criticism or praise of my own insignificant contributions, the man who invented databases capable of preserving *enterprise integrity.*