Management Challenge

October 1997

Whitemarsh Information Systems Corporation
2008 Althea Lane
Bowie, Maryland 20716
Tele: 301-249-1142
Email: mmgorman@wiscorp.com
Web: www.wiscorp.com
1
Will the Real Database ...

1.1 Database

1.2 Where are we today?

1.3 Database benefits

1.4 Database management

1.5 Reality revisited
1.1 Database

A consistent set of semantics across a large area of corporate knowledge

! To promote research of the past,

! To plan for current activities, and

! To project requirements for the future
Database means ........

! Making private knowledge public

! Leveraging single data collections for multiple uses

! Maintaining order, uniformity, and commonness of understanding in data representation
Some very key points....

! Database is an operating philosophy

! It can be implemented either manually, or by computer hardware.

! Hardware or software is merely an implementation facilitator

! Database has been around since there was a highly organized, non-redundant filing system.
1.2 Where are we today ??

! Organizations with 100 to 500 databases, thinking they have database

! Organizations with multiple DBMS thinking they have database

! Organizations with multiple application packages that use DBMS as access methods, thinking they have database

! Organizations with long standing applications that contain many hundreds of programs, each containing data semantics, thinking they have database
**But do they have database?**

! Yes, if there is a single set of semantics for all programs and data.

! No, otherwise !!!

! Most are in the otherwise column
If you are in the otherwise column,

You probably have

- Multiple versions of the truth: impossible to determine which is the most accurate data
- Re-invention of the wheel: each user/application group creates his own data & semantics
- High cost applications: each application has to invent its own data storage and access system
- Low quality data: decentralized, redundant private files that are impossible to control
- Information in disarray: nearly impossible to assemble complete picture
Measuring database success

False measures

! 5 billion character databases

! 65 transactions per wall clock second

! 7 levels of hierarchy without redundancy

! 1 record type for 8 different uses

Real measures

! Organizational quality

! Coherent policy

! Efficient decision making

! Multipurpose, high-integrity data

! Valid projections and forecasts
## Comparing traditional and database approaches

<table>
<thead>
<tr>
<th>Characteristic</th>
<th>Traditional Data Processing</th>
<th>Database Approach</th>
</tr>
</thead>
<tbody>
<tr>
<td>Ownership</td>
<td>Programmer</td>
<td>Corporation</td>
</tr>
<tr>
<td>File</td>
<td>Stand-alone, transient storage and high performance structure design</td>
<td>Permanent storage; small part of a naturally organized structure</td>
</tr>
<tr>
<td>Input</td>
<td>Only critical data to serve report needs; one data collection per report</td>
<td>Capture data and contexts; preserve environment and history; one data collection per database</td>
</tr>
<tr>
<td>Output</td>
<td>Multiple reports via single pass of data file</td>
<td>Transient report requirements that change often</td>
</tr>
<tr>
<td>Program</td>
<td>File division and procedure division with program based semantics to process multi-meaning data fields</td>
<td>Cobol only for updates and complex reports; ad hoc language; all semantics within the database</td>
</tr>
<tr>
<td>Data Field</td>
<td>Small as possible; pack as many values and meanings as possible</td>
<td>Single meaning; well defined with strict rules</td>
</tr>
</tbody>
</table>
Traditional Paradigm: User-System-File
Database Paradigm: Policy-based, Consistent & Centralized Semantics
But that requires

! Public rules/policies

! Public programs

! Natural data contexts

! Single purpose data
For that, we need a change of focus:

From:

PROGRAM (TRADITIONAL VIEWPOINT)

To:

DATABASE (DATABASE VIEWPOINT)

and that enables...
The promise of database

! Database centered analysis & design

! Database designs built on natural reflections of business function fundamentals

! Long term stability in data emphasis n "canned" tools to accomplish transient work

! Corporate ownership

! Common processing
1.3 Database benefits

If database is achieved within an organization, it brings:

- Short term benefits
- Long term benefits
- Financial benefits
- Flexibility benefits
- System development benefits
Short-term benefits

- Single set of facts
- Efficient change capability
- Centralized semantics
- Single foundation for building applications
Long-term benefits

- Define once and use many times
- Collect and store once, use many times
- Past, present and future data collected and stored on same basis
- Increased value
- Able to be used for long range trends
- Reduced data volumes mean reduced cost

To have database is to have effective organization, and that is a prerequisite for quality system development

*If management had enforced quality data semantics there wouldn’t be a Year 2000 problem! The solution was available and in place within DBMSs by the middle 1970s.*
Financial benefits

Database can be viewed as a way to cut down on the data processing backlog

<table>
<thead>
<tr>
<th>Database Dollars</th>
<th>Data Processing Dollars</th>
</tr>
</thead>
<tbody>
<tr>
<td>Serves many applications</td>
<td>Serves only 1 application</td>
</tr>
<tr>
<td>Single database design for many</td>
<td>Each application/program has own file design</td>
</tr>
<tr>
<td>applications</td>
<td></td>
</tr>
<tr>
<td>Standardized database access across</td>
<td>Each application/program creates its own</td>
</tr>
<tr>
<td>many applications</td>
<td>access strategy</td>
</tr>
<tr>
<td>Multiple choice of interrogation</td>
<td>Only 1 choice, COBOL</td>
</tr>
<tr>
<td>languages for each database</td>
<td></td>
</tr>
<tr>
<td>Maximum dollar use spread across</td>
<td>Maximum dollar cost for each application</td>
</tr>
<tr>
<td>many applications</td>
<td></td>
</tr>
</tbody>
</table>
How much does it cost not to have database?

! An organization has 100 billion characters of space for only 30 billion characters of real data that’s 300% more per year in storage costs alone

! An organization has 1200 programs, only 300 are needed. At $25,000 life cycle cost per program, that is $22,500,000 in wasted programs

! An organization has 30 different financial systems for: general ledger property management accounts payable, etc. Estimated cost for redundant maintenance: $150 million/year

! An organization spent over $250 million attempting database through the wrong techniques, result: lots of fire starter!

! The U.S. DoD attempted to build data standardization on top of a flawed model–result, >$55 million in wasted funds. No viable, useful result. Worse than that, almost 10 years wasted!
**Flexibility benefits from database object paradigm......**

<table>
<thead>
<tr>
<th>Questions regarding database object distribution effects</th>
<th>Semantic Control</th>
<th>Development Control</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Centralized</td>
<td>Decentralized</td>
</tr>
<tr>
<td>Are database objects able to be shared among sites?</td>
<td>yes</td>
<td>yes</td>
</tr>
<tr>
<td>Is concurrent processing of the same database object instance possible?</td>
<td>yes</td>
<td>maybe</td>
</tr>
<tr>
<td>Are common or corporate reports possible?</td>
<td>yes</td>
<td>yes</td>
</tr>
<tr>
<td>Can there be an overbearing &quot;big brother&quot; feeling?</td>
<td>yes</td>
<td>maybe</td>
</tr>
<tr>
<td>Is there local control and ownership?</td>
<td>no</td>
<td>maybe</td>
</tr>
<tr>
<td>Does there need to be common data standards &amp; policies?</td>
<td>yes</td>
<td>yes</td>
</tr>
<tr>
<td>Can local data requirements be satisfied?</td>
<td>maybe</td>
<td>yes</td>
</tr>
</tbody>
</table>
And, once we have database, we’ll have:

- High-quality information
- Greater control
- Faster decision-making

In short, database leads to data flexibility.
Data flexibility means . . . . .

! The ability to use and reuse data without distortion

! Ability to draw data from consistent source for varied specialized processors (data warehouses, data marts, OLAP processors, spread sheets, graphics, etc.)
Finally, we have: System development benefits

! Database enables new systems to be developed more quickly as 50-70% of every program is eliminated.

! Database causes data to be collected & stored such that the natural context is retained, enabling it to be used over and over.

! Code generators enable quick development of high-quality, high-performance applications

! Transient applications can be developed with 4th generation languages and report-writers that may be inefficient, but can be developed quickly and easily.
1.4 Database management basic terms

<table>
<thead>
<tr>
<th>Term</th>
<th>Definition</th>
</tr>
</thead>
<tbody>
<tr>
<td>Database</td>
<td>Coherent collection of data</td>
</tr>
<tr>
<td>Database management</td>
<td>An implementation of database</td>
</tr>
<tr>
<td>Database management system</td>
<td>A generalized soft/hard/IRM ware solution for implementing database management</td>
</tr>
</tbody>
</table>
Database management is not the same as database

Database is business philosophy

Only when database has been implemented. Will database management, a technique to implement database, actually work.
What does it take to succeed at database

<table>
<thead>
<tr>
<th>Viewpoint</th>
<th>Database Component</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Logical</td>
</tr>
<tr>
<td>Technology</td>
<td>Data model</td>
</tr>
<tr>
<td>Staffing</td>
<td>Database specialist</td>
</tr>
<tr>
<td>Project</td>
<td>Conceptual specification phase</td>
</tr>
<tr>
<td></td>
<td>Implementation phase</td>
</tr>
<tr>
<td></td>
<td>Production and administration phase</td>
</tr>
<tr>
<td>DBMS</td>
<td>Schema and sub-schema</td>
</tr>
</tbody>
</table>
What's this thing called a DBMS?!

Never, never, never confuse DBMS with Database. If they were the same then every Word Processor user would only produce Pulitzer Prize winning books!
### How do I pick a good DBMS package?

<table>
<thead>
<tr>
<th>Facility</th>
<th>Quality characteristics</th>
</tr>
</thead>
<tbody>
<tr>
<td>Metadata repository system</td>
<td>Stores metadata, database, database applications environments&lt;br&gt;DBMS must enforce all these definitions</td>
</tr>
<tr>
<td>Logical database (ANSI/SQL)</td>
<td>Variety of logical structure definitions. Must have strong editing, validation, and automatic procedures</td>
</tr>
<tr>
<td>Physical database</td>
<td>Should enable a variety of physical structures, indexes, tuning capabilities, etc.</td>
</tr>
<tr>
<td>Interrogation (ANSI/SQL)</td>
<td>Should have a variety of languages for Cobol access, technical user, and end user. Full suite of SQL query language facilities. Should have a variety of packages for mainstream corporate applications</td>
</tr>
<tr>
<td>System Control</td>
<td>Good backup and recovery, security and backup, multiple database processing, concurrent operations, etc.</td>
</tr>
</tbody>
</table>
1.5 Reality revisited

! Database is an operating philosophy, not something you can buy.

! Database represents a fundamental change.

! Database requires significant management commitment, manpower, and technological resources.
Database successes can be within 18 months, but the full implementation will take much longer

Requirements summary

- Management understanding & perception
- Involvement
- Manpower
- Involvement
- Analytical methods
- Involvement
- Learning time, i.e., wisdom thru mistakes
- Involvement
- Development time
- Software and hardware
- Involvement
- Commitment and dedication