Data Quality and Systems Theory

Ken Orr

The Ken Orr Institute

1. Introduction

There is a critical scene in the movie "War Games," set in the Strategic Air Command Center under a mountain near Colorado Springs, Colorado. On a huge map of the globe, there are arcs indicating large numbers of ICBMs coming in from the Soviet Union, the general in charge is trying to decide if he should ask the President for permission to launch a retaliatory attack, and the technicians are telling him that the threat is real. At this point, the scientist who designed the system originally and who knows that something is wrong with the system says, "General, what you see on that board is not reality, it is *a computer generated hallucination!*"

Recently, it was revealed by a former Director of the CIA that a real life version of this fictional scenario was actually played out for real when a test tape was inadvertently installed and the screen at a similar center warned of a similar nuclear attack. As computers play a more and more important role in the real-world— a world in which the computer outputs often presents a picture of the real-world for critical activities—it is increasingly vital that that picture be correct!

2. A Systems Model for Data

In the mid-1970s, I and a number of colleagues developed a model for information systems that predicted, among other things: (1) major problems with making the transition to the Year 2000, (2) data quality difficulties in many operational systems of the time and (3) fundamental issues involved in the accuracy of confidential/secret data. All of these predictions were based on some very simple systems models that have been born out over the past two decades.

The genesis of the theory that allowed us to formulate our predictions involved viewing of information systems as being imbedded in a larger framework of a real-word feedback-control system (FCS) (Figure 1). Two observations caused us to look at information systems this way: (1) all of the information systems we developed operated in a larger, goal-seeking, enterprise environment, and (2) those systems that failed to take into account that larger FCS context were often difficult to operate and their outputs difficult to reconcile with the real-world. Clearly, we began to see that the data within our information systems did not exist in a vacuum. As a result, we began to explore the implications of a true systems model on information systems.

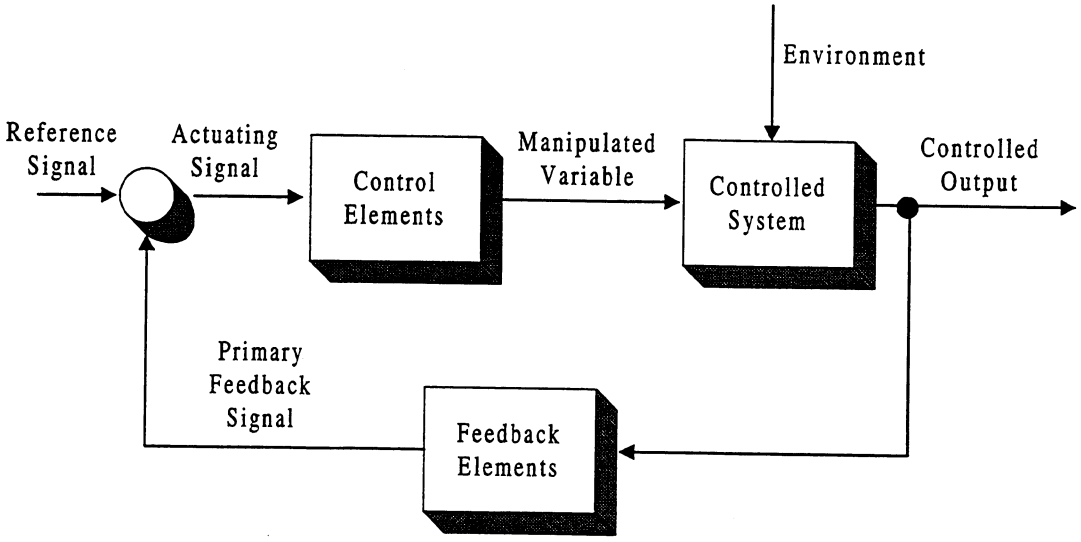

Figure 1 - Feedback-Control Systems Model

The principal role of most information systems is to present views of the real-world so that the people in the organization can create products or make decisions. If those views do not agree with the real-world for any extended period of time, then the system is a poor one, and, ultimately, like a delusional psychotic, the organization will begin to act irrationally.

3. Defining Data Quality

From a FCS standpoint, data quality is actually an easy thing to define. *Data quality is the measure of the agreement between the data views presented by an information systems and that same data in the real-world.* A data quality of 100% would indicate, for example, that our data views are in perfect agreement with the real-world, whereas a data quality of 0% would indicate no agreement at all.

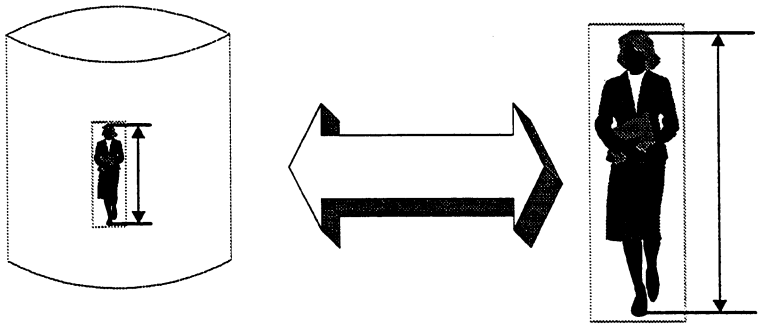

Figure 2 - Data Quality

Now, no serious information system has a data quality of 100%. The real concern with data quality is to insure not that the data quality is perfect, but that the quality of the data in our information systems is *accurate enough, timely enough* and *consistent enough* for the organization to survive and make reasonable decisions.

The real difficulty with data quality is change. Data on our databases is static, but the real-world keeps changing. Even if our system has a database that is 100% in agreement of the real-world at time t_0, at time t_1 it will be slightly off, and at time t_2 it will be even further off. FCS theory states that if you want a system to track the real-world, then you must have some mechanism to do so—you must have feedback!

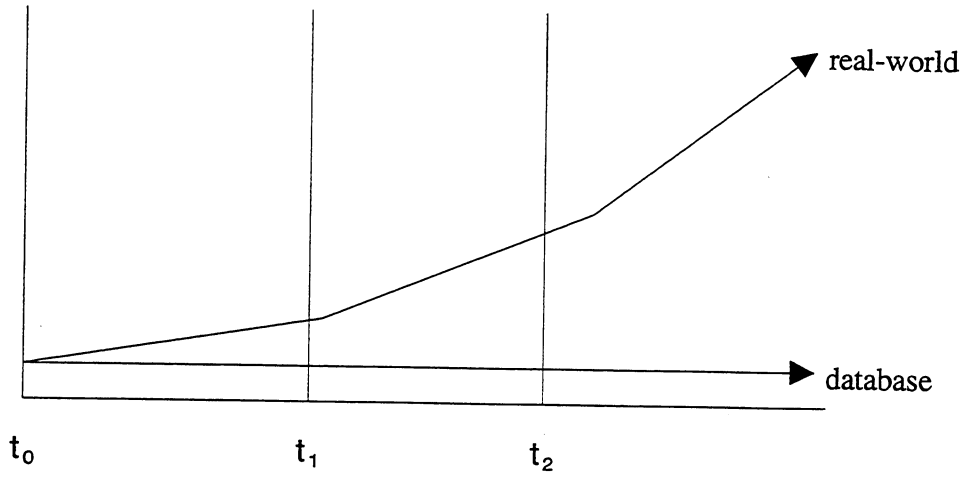

Figure 3 - Data and the Real-world

But where does this feedback come from? The classic answer from information systems developers is that feedback is solely the responsibility of the business user. "Our job is not to understand what our systems are being used for or even their context!" they maintain, "We simply build systems that meet the requirements of our users—it is the job of the users to insure that the data on our data bases is maintained in an accurate and timely manner. The best we can do is to insure that the database is internally consistent and that the user's business rules are enforced."

Users, on the other hand, have historically felt that they were held responsible for the quality of data in information systems that they often did not understand, where it was often difficult to make appropriate corrections and where the results of certain kinds of changes were unpredictable.

Unfortunately, as it turns out, the problem of data quality is *fundamentally* tied up in how our system fits into the real-world, in other words, with how users actually use the data in the system. In fact, two things have to happen for data on any database to track the real-world: (1) someone or something, e.g. a automatic sensor, has to compare the data views from the system with data

from the real-world and (2) any deviations from the real-world have to be corrected and re-entered.

Too often, systems developers have an overly simplistic view of how systems are organized; they think of systems in a simplistic Input-Process-Output (IPO) "transform" model (Figure 4).

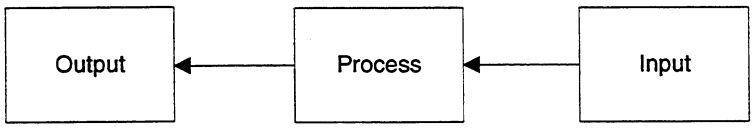

Figure 4 - Input-Process-Output Model

But, this IPO model fails to account for the role that the database plays in a broader context (Figure 5).

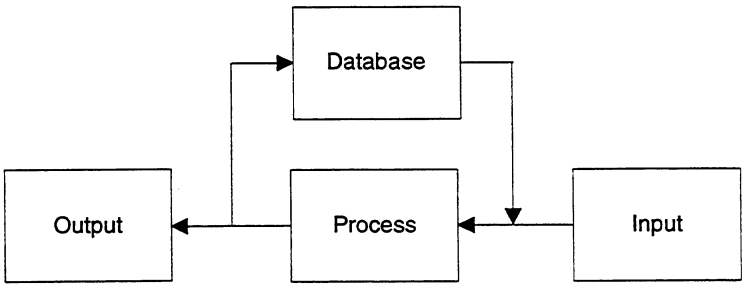

Figure 5 - Input-Process-Database-Output Model

In real information systems, the database acts to mediate between the input and the output, where the input and output: (1) occur at different times, and/or (2) represent different views of the real-world. This broader view of a system then makes it possible to fully understand the FCS model, in which the information system fits within actions taken in the real-world (Figure 6)

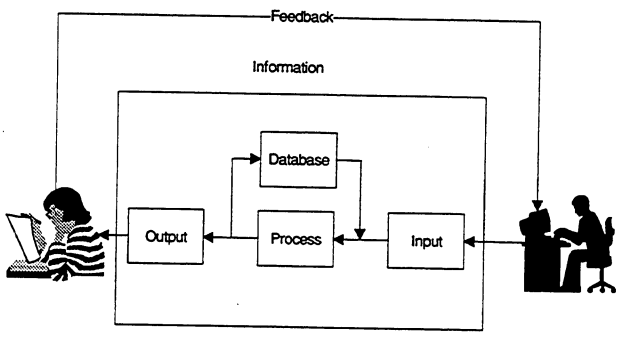

Figure 6 - Information Systems in a Real-world Context

In this model, data is entered in the system based on external inputs, it undergoes processing and gets stored in a database, which in turn is processed to produce outputs that are used in (compared with) the real world. Finally, new inputs are produced (and fed back) so that the database can be kept correct. Without this final loop, the system will fail to maintain its database, and therefore its outputs correctly. This final FCS model (Figure 4) allows us to understand more fully the true problem of data quality—the better our information system fits within the real world, the better will be the quality of our data, the worse the fit, the worse the data.[1]

[1] It is possible to formulate other ways in which data in a database can deviate from the real-world: poor data definitions, failure to correctly enter data, rounding and/or compounding errors in calculations, faulty calculations, etc. However, while these other errors can create data quality problems, those created by lack of consistent user feedback dwarfs all the other kinds of errors.

4. Data Quality Rules

There are a number of general Data Quality Rules that one can deduce from a FCS view of information systems:

DQ1. Data which is not used cannot be correct for very long;

DQ2. Data Quality in an information system is a function of its use not its collection;

DQ3. Data Quality will, ultimately, be no better than its most stringent use;

DQ4. Data Quality problems tend to become worse with the age of the system;

DQ5. The less likely some data attribute (element) is to change, the more traumatic it will be when it finally does change; and

DQ6. Laws of data quality apply equally to data and meta-data (the data about the data

4.1. Data Quality and Use

Unfortunately, many, if not most, of these observations fly in the face of systems practice. It is common practice, for example, to collect large numbers of unused data elements on the premise that someday someone might want to use them and it is cheaper to put them in the system now than to do so when the need actually arises. However, if you follow the FCS model, you understand that if the organization is not using data, then, over time, errors in capturing real-world changes will be ignored (because there is no mechanism for comparing the data that the system has and the actual value).

In a biological systems, scientists refer to this as atrophy. If you don't use a part of the body, for example, then it atrophies—"use it or lose it!" is the common way of describing atrophy. In a practical sense, something similar to atrophy happens with unused data—if no one uses the data, then the system become insensitive to that data. Like an animal that is blind or deaf, changes in the real-world are not accounted for.

However, in most large systems, it is difficult to tell if a particular pieces of data are truly being used. In many cases, for example, data elements actually appear on outputs somewhere but no one actually uses those reports or screens, at least any more. In other cases, the data elements

are used, but not seriously. Here DQ3 comes into play, namely, that the quality of a specific piece of data will be no better than its most stringent use. Data which is not used very seriously tend to be not very good. In general, its data quality will be better than data that isn't used at all, but not much better. For example, names and addresses that are only used for mailing and are not corrected tend not to be very accurate.

The nature of data quality, then, hinges then upon the connections of that system to the outside world. The stronger those connections, the better the system and the better the data quality.

4.2. Data Warehousing and Data Quality

It was also clear to us, nearly twenty years ago, that many operational systems being developed then would have data quality problems because much of their data was not routinely used. This practice has continued down through the years. Now, as large organizations have begun to create integrated Data Warehouses for decision support, the problem has become painfully clear. They have discovered that the quality of the data in their legacy databases is their single biggest problem in building reliable, trustworthy Data Warehouses or Data Marts. One data manager for a large company reported that fully 60% of the data that was transferred to their data warehouse failed to pass the business rules that the systems operators had said were in force, something that could have perhaps been predicted based on poor data usage.

The good news here is that developing Data Warehouses represents a quantum leap forward in terms of end user usage of data. As more advanced Data Warehouses and Data Marts are created, more and more people will be using data in more and more important ways. The need for quality data has already begun to focus more management attention on just how poor our data quality is in some cases.

In the 1970s, when we first began to understand the implications of the FCS model, it became clear why so many of the systems we had worked on failed to meet their data quality objectives. In many cases, we had tried to develop systems that created data that no one used. We recognized that these systems were difficult to define, difficult to program, and difficult to

operate, what we didn't understand was why. After we began to understand the implications of FCS, we used a new development approach which, to insure that all data collected and stored would actually be used, involved designing systems by working backward from uses to outputs to database to inputs. In one case, developed in this manner, we found that the legacy system we were replacing had three times more data elements than it actually needed. Imagine the waste, imagine the problems with data quality. Attempting to build quality systems without understanding FCS theory is much like attempt to building an airplane without understanding the implications of aerodynamics.

4.3. Data Quality and The Year 2000

It was also clear early on that the Year 2000 would be a serious problem because there would be very little use of the millennium and century fields until the Year 2000 actually arrived.[2] Unfortunately, we failed to see just how massive the problem would actually be when we got to it. Estimates now range in the hundreds of billions of dollars[3]. The actual problem involved with finding, fixing and testing the changes to the Year 2000 problem is its ubiquity—it represents a simple problem repeated a billion times—just changing all of the software and hardware involved will take the information technology industry better part of the next five years.

Could it have been avoided? Possibly, but only if "use-based" data quality programs had been in place. Millennium and century fields have not been tested on a large-scale because of the "time-horizons" of our systems do not use those fields. As the Year 2000 approaches, more and more systems will fail because their systems practice will be forced to deal with dates in the 21st Century.

---

[2] While it is true that some organizations have had to deal for decades. Savings and Loan Companies financing 30 year mortgages have had to deal with the year 2000 since the late 1960s, for the most part, organizations are only now reaching their "century time horizons" where they need accurate dates that reach into the 21st Century.

[3] The most recent estimate of the worldwide impact is actually 1.5 trillion dollars! (Jones, 96)

4.4. Data Quality, Systems Age and Meta-data

Predictable, also, is that fact that as systems get older their data quality problems get worse. In the early days of data processing, it was widely thought that the lifespan of the average information system would only be a few years; therefore, it didn't make sense to try to put in place costly data quality programs, since any problems or shortcomings in the current system would correct itself in subsequent versions. In fact, major information systems have turned out to be much longer lived than anyone would have anticipated. There are large numbers of legacy systems in operation today that date back 20 or 25 years. As a consequence it is necessary to view data quality over time.

What we have found is that not only does data quality suffer as a system ages, so too does its meta-data. Clearly, what happens is that people who are responsible for entering the data discover which data fields that are not used, either they make little effort to enter the correct data, or they begin to use the data for other purposes. The consequence is that the both the data and the definitions of the data (the meta-data) no longer agree with the real-world.

Another predictable problem occurs where the structure of the real-world differs significantly from the real-world. Often times, systems designers do not actually look at the structure (patterns) of data that occurs in various fields, but rather arbitrarily assign data to fixed fields based on technology limits or constraints. Because the developers are not looking at the data, the structure is not changed. As a result, lots of round data is forced into square holes.

4.5. Data Quality and Secrecy

One of the most troubling implications of the FCS model to data quality has to do with confidentially and secrecy. If the quality of data is truly wrapped up in its use, then there seems to be serious limitations to the quality of confidential/secret data. The converse of the data quality rules seems to be that confidential/secret data will always have limited quality. This may account for the fact that while dictatorships seem to be the most efficient way to run a society, democracies, for all their inherent inefficiencies work better. A Free Press and an open political process, though bothersome, provide feedback.

It is not clear just yet what the impact of information overload will have on data quality. In our modern technologically-based society, there may be such a thing as too much data. Because there is so much, a smaller and smaller part of that data may actually be used.

5. Use-based Data Quality Programs

If data quality is a function of its use, there is only one sure fire way to improve data quality, improve its use! We call this program *use-based data quality*. Use-based data quality programs are built around finding innovative, systematic ways to insure that critical data is used. Such programs involve:

- Use-based Data Quality Audits
- Use-based Data Quality Redesign
- Use-based Data Quality Training
- Use-based Data Quality Continuous Measurement

5.1. Use-based Data Quality Audits

In order improve our data quality, it is necessary to get a good handle on just how good the data in our databases is today. Use-based Data Quality Audits involved answering a number of key questions:

- What data are we interested in?
- What is the data design?
    - What is the data model?
    - What is the meta-data
- How is the data used today?
- Who uses it?
- For what purposes is the data used?
- How often is the data used?
- What is the Data Quality
    - What is on the database?
    - How does it compare with the current data in the real-world?
    - How current is the data?

For the most part, data quality audits are best done using statistical sampling. It is rarely a good idea to try to verify all of the data on a real data base, what is necessary that a sufficient sample be created to be able to draw meaningful conclusions.

5.2. Use-based Data Quality Redesign

In order to improve data quality, it is mandatory to improve the linkage between the usage of data throughout the system. One of the problems that many people face is where to begin. In point of fact, while most legacy environments contain hundreds of records (tables) and thousands of data elements, all the data is not equal. In most systems, there a few critical sets of data that make all the difference. Often, the "customer," "product," "order" and "organizational structure" data is what is most important. The first step in a serious data quality redesign program then is to identify the critical data areas.

The first element of redesign involves a careful reexamination of the how the critical pieces of data are used. Normally this is most manifest in two areas, the basic business processes (order entry through fulfillment, etc.) and in decision support. Use-based design means focusing on exactly how the data will be used, and in trying to identify inventive ways to insure that the data is used more strenuously. In a great many cases, this means becoming more creative in getting the people most knowledgeable about the data to take responsibility for that data.

A good example is the "frequent flyer" programs offered by the airlines. In addition to creating customer loyalty, such programs also go a long way to improving the quality of the data. In the normal case where the same flyer may have more than one Frequent Flyer Identification No. assigned, it is in the best interest of the customer to make sure that the records are consolidated and vital information such as name, address, family relationships and preferences are kept up to date. Developing a use-based data quality is to expend much more effort on the actual process of completing the feedback use of data. In general, that often means the reduction of the number of data elements collected. If data cannot be maintained correctly, then it is questionable whether that data provides any value to the enterprise.

Another major component of use-based design is to understand the content of the existing critical data bases. A number of tools have emerged in recent years that aim at analyzing and combining data from multiple databases to create a common view of "customers", "products", "vendors", etc.

The normal result of these programs is to dramatically reduce the size (consolidate) major data bases. Consolidations of 5:1 or even 10:1 are not uncommon. A second byproduct that is derived from this process is the development of much more sophisticated set of meta-data based on data content.

Other techniques for improving data quality is to promote (demand) "sharing" of data through the use of "common databases". With the advent of the Internet, more and more people are able to access data more easily. Providing easy data access to as broad an audience as possible has the long-term effect of dramatically improving our data quality.

5.3. Use-based Data Quality Training

One of the major problems with data quality problems is getting both users and managers to understand the fundamentals of data quality. In order for any data quality program to work long term requires devoting a significant amount of time to training and education in the nature of use-based data quality. It is hard to convince users and managers who have been used to requiring all sorts of data arbitrarily that unless they can guarantee that the data is used it won't be any good. Fortunately, this kind of thinking becomes natural after a relatively short time, and begins to be reinforced in practice.

5.4. Use-based Data Quality Continuous Measurement

As in most areas of quality, data quality requires constant measurement to insure that use-based practices are followed through. As Deming noted, most quality problems are systems problems, not worker problems. However, individual errors contribute to poor quality data as well. Measurement and quality programs must go hand-in-hand. Periodically, all of the same questions that were raised in the Data Quality Audit need to be redone for the redesigned system as well.

A final note on measurement is not to be persuaded by internal measures without external verification. All that internal measurement can ultimately insure is that our data is internally consistent. No large organization can rely on its inventory records without periodic "physical

inventories". History has shown that having records that show that we should have 23 computers in Warehouse X does not mean that there are actually 23 computers on the shelves. If we want our data on our databases to agree with the real world, we must periodically verify that it really does, and we must take actions to reconcile the differences. Data that is truly vital must be audited.

6. Conclusion

Too often, the primary focus of data quality projects is to increase the internal controls involved in entering and editing data. As laudable as these efforts are, they are ultimately doomed to failure. The only way to truly improve data quality is to increase the use of data. If an organization truly wants to improve data quality, it needs to insure that there is stringent use of each data element.

Because of the problems created by the Year 2000, every organization that uses computers in the world will have to step up to the problems of data quality in the next couple of years. This, coupled with the increase need for quality data for marketing and planning purposes will make data quality a high priority item in every enterprise. Use-based Data Quality provides a theoretically sound and practically achievable means to dramatically improving our data quality.

Bibliography

Austin, Robert, *Measuring and Managing Performance in Organizations*, Dorset House, New York, NY 1996

Burrill, C. and Ellsworth, L. Quality Data Processing, Burrill-Ellsworth Associates, Tenafly, NJ, 1982

Jones, T. C., *The Global Economic Impact of the Year 2000 Software Problem*, SPR, Lexington, MA 1996

Orr, K., *Structured Requirements Definition*, The Ken Orr Institute, Topeka, KS 1981

About the Author

Ken Orr is a Principal Researcher at The Ken Orr Institute. He has been involved in systems and software development/management since the early 1960s. Mr. Orr's work involves consulting and education in a variety of area including Knowledge and Data Management, Data Warehousing, Business Process Reengineering and Application Development.

Mr. Orr can be reached at (913)357-0003/email 70540.2465@compuserve.com.