WILLIAM KENT

DATA and REALITY

basic assumptions in data processing reconsidered

NORTH-HOLLAND

DATA AND REALITY

Basic Assumptions in Data Processing Reconsidered

William KENT
IBM
San Jose, California

1978

NORTH-HOLLAND PUBLISHING COMPANY – AMSTERDAM • NEW YORK • OXFORD

ISBN: 0 444 85187 9

Published by:
NORTH-HOLLAND PUBLISHING COMPANY – AMSTERDAM • NEW YORK • OXFORD

Sole distributors for the U.S.A. and Canada:
ELSEVIER NORTH-HOLLAND INC.
52 Vanderbilt Avenue
New York, N.Y. 10017

Library of Congress Cataloging in Publication Data

Kent, William, 1936-
Data and reality.

Bibliography: p.
1. Data base management. 2. Data structures (Computer science) I. Title.
QA76.9.D3K46 001.6'4 78-19130
ISBN 0-444-85187-9

PRINTED IN THE NETHERLANDS

DATA AND REALITY

PREFACE

> A message to mapmakers: highways are not painted red, rivers don't have county lines running down the middle, and you can't see contour lines on a mountain.

For some time now my work has concerned the representation of information in computers. The work has involved such things as file organizations, indexes, hierarchical structures, network structures, relational models, and so on. After a while it dawned on me that these are all just maps, being poor artificial approximations of some real underlying terrain.

These structures give us useful ways to deal with information, but they don't always fit naturally, and sometimes not at all. Like different kinds of maps, each kind of structure has its strengths and weaknesses, serving different purposes, and appealing to different people in different situations. Data structures are artificial formalisms. They differ from information in the same sense that grammars don't describe the language we really use, and formal logical systems don't describe the way we think. "The map is not the territory" [Hayakawa].

What is the territory really like? How can I describe it to you? Any description I give you is just another map. But we do need some language (and I mean natural language) in order to discuss this subject, and to articulate concepts. Such constructs as "entities", "categories", "names", "relationships", and "attributes" seem to be useful. They give us at least one way to organize our perceptions and discussions of information. In a sense, such terms represent the basis of my "data structure", or "model", for perceiving real information. Later chapters discuss these constructs and their central characteristics -- especially the difficulties involved in trying to define or apply them precisely.

Along the way, we implicitly suggest a hypothesis (by sheer weight of examples, rather than any kind of proof -- such a hypothesis is beyond proof): there is probably no adequate formal modelling system. Information in its "real" essence is probably too amorphous, too ambiguous, too subjective, too slippery and elusive, to ever be pinned down precisely by the objective and deterministic processes embodied in a computer. (At least in the conventional uses of computers as we see them today; future developments in artificial intelligence may endow these machines with more of our capacity to cope.) This follows a path pointed out by Zemanek, connecting data processing with certain philosophical observations about the real world, especially the aspects of human judgement on which semantics ultimately depend ([Zemanek 72]).

In spite of such difficulties (and because I see no alternative), we also begin to explore the extent and manner in which such constructs can and have been incorporated into various data models. We are looking at real information, as it occurs in the interactions among people, but always with a view toward modelling that information in a computer based system. The questions are these: What is a useful way to perceive information for that purpose? What constructs are useful for organizing the way we think about information? Might those same constructs be employed in a computer based model of the information? How successfully are they reflected in current modelling systems? How badly oversimplified is the view of information in currently used data models? Are there limits to the effectiveness of any system of constructs for modelling information?

In spite of my conjecture about the inherent limits of formal modelling, we do need models in order to go about our business of processing information. So, undaunted, I have assimilated some of my own ideas about a "good" modelling system, and these appear toward the end.

Keep in mind that I am not talking about "information" in a very broad sense. I am not talking about very ambitious information systems. We are not in the domain of artificial intelligence, where the effort is to match the intellectual capabilities of the human mind (reasoning, inference, value judgements, etc.). We are not even trying to process prose text; we are not attempting to understand natural language, analyze grammar, or retrieve information from documents. We are primarily concerned with that kind of information which is managed in most current files and data bases. We are looking at information that occurs in large quantities, is permanently maintained, and has some simplistic structure and format to it. Examples include personnel files, bank records, and inventory records.

Even this modest bit of territory offers ample opportunity for misunderstanding the semantics of the information being represented.

Within these bounds, we focus on describing the information content of some system. The system involved might be one or more files, a data base, a system catalog, a data dictionary, or perhaps something else. We are limiting ourselves to the information content of such systems, excluding such concerns as:

* Real implementations, representation techniques, performance.

* Manipulation and use of the data.

* Work flow, transactions, scheduling, message handling.

* Integrity, recovery, security.

A caution to the lay reader in search of a tutorial: this book is not about data processing as it is. As obvious as these concepts may seem, they are not reflected in, or are just dimly understood in, the current state of data processing systems. "We do not, it seems, have a very clear and commonly agreed upon set of notions about data -- either what they are, how they should be fed and cared for, or their relation to the design of programming languages and operating systems. This paper sketches a theory of data which may serve to clarify these questions. It is based on a number of old ideas and may, as a result, seem obvious. Be that as it may, some of these old ideas are not common currency in our field, either separately or in combination; it is hoped that rehashing them in a somewhat new form may prove to be at least suggestive" [Mealy]. That opening paragraph of a now classic paper, some ten years old, is still distressingly apt today.

There is a wonderful irony at work here. I may be trying to overcome misconceptions which people outside the computer business don't have in the first place. Many readers will find little new in what I say about the nature of our perceptions of reality. Such readers may well react with "So what's new?" To them, my point is that the computing community has largely lost sight of such truisms. Their relevance to the computing disciplines needs to be re-established.

People in the data processing community have gotten used to viewing things in a highly simplistic way, dictated by the kind of tools they have at their disposal. And this may suggest another wonderful irony. People are awed by the sophistication and complexity of computers, and tend to assume that such things are beyond their comprehension. But that view is entirely backwards! The thing that makes computers so hard to deal with is not their complexity, but their utter simplicity. The first thing that ought to be explained to the general public is that a computer possesses incredibly little ordinary intelligence. The real mystique behind computers is how anybody can manage to get such elaborate behavior out of such a limited set of basic capabilities. The art of computer programming is somewhat like the art of getting an imbecile to play bridge or to fill out his tax returns by himself. It can be done, provided you know how to exploit the imbecile's limited talents, and are willing to have enormous patience with his inability to make the most trivial common sense decisions on his own. Imagine, for example, that he only understood grammatically perfect sentences, and couldn't make the slightest allowance for colloquialisms, or for the normal way people restart sentences in mid-speech, or for the trivial typographical errors which we correct so automatically that we don't even see them. The first step toward understanding computers is an appreciation of their simplicity, not their complexity.

Another thought, though: I may be going off in the wrong direction by focussing so much concern on computers and computer thinking. Many of the concerns about the semantics of data seem relevant to any record keeping facility, whether computerized or not. I wonder why the problems appear to be aggravated in the environment of a computerized data base. Is it sheer magnitude? Perhaps there is just a larger mass of people than before who need to achieve a common understanding of what the data means. Or is it the lost human element? Maybe all those conversations with secretaries and clerks, about where things are and what they mean, are more essential to the system than we've realized. Or is there some other explanation?

The flow of the book generally alternates between two domains, the real world and computers. Chapter 1 is in the world of real information, exploring some enigmas in our concepts of "entities". Chapter 2 briefly visits the realm of computers, dealing with some general characteristics of formally structured information systems. This gives us a general idea of the impact the two domains have on each other. Chapters 3 through 6 then address other aspects of real information. Chapters 7 through 11, dealing with data processing models, bring us back to the computer. We top it all off with a smattering of philosophical observations in Chapter 12.

This has been an approximate characterization -- one view -- of what the rest of the book contains. Please read on to discover what you might think the book is about.

* * * *

I want to thank the people who took the time to comment on (and often contribute to) earlier versions of this material, including Marilyn Bohl, Ted Codd, Chris Date, Bob Engles, Bob Griffith, Roger Holliday, Lucy Lee, Len Levy, Bill McGee, Paula Newman, and Rich Seidner. George Kent, of the Political Science Dept. at the University of Hawaii, provided a valuable perspective from a vantage point outside of the computing profession. Karen Takle Quinn, our head librarian, was immensely helpful in tracking down many references. I thank Willem Dijkhuis of North Holland for his substantial encouragement in the publication of this book.

And very special thanks go to my wife, Barbara, who helped make the book more readable, and who coped and sacrificed more than anyone else for this book.

CONTENTS

1.0 ENTITIES

> "Entities are a state of mind. No two people agree on what the real world view is." [Metaxides]

An information system (e.g., data base) is a model of a small, finite subset of the real world. (More or less -- we'll come back to that later.) We expect certain correspondences between constructs inside the information system and in the real world. We expect to have one record in the employee file for each person employed by the company. If an employee works in a certain department, we expect to find that department's number in that employee's record.

So, one of the first concepts we have is a correspondence between things inside the information system and things in the real world. Ideally, this would be a one-to-one correspondence, i.e., we could identify a single construct in the information system which represented a single thing in the real world.

Even these simple expectations run into trouble. In the first place, it's not so easy to pin down what construct in the information system will do the representing. It might be a record (whatever that means), or a part of one, or several of them, or a catalog entry, or a subject in a data dictionary, or .... For now let's just call that thing a representative, and come back to that topic later. Let's explore instead how well we really understand what it is that we want represented.

As a schoolteacher might say, before we start writing data descriptions let's pause a minute and get our thoughts in order. Before we go charging off to design or use a data structure, let's think about the information we want to represent. Do we have a very clear idea of what that information is like? Do we have a good grasp of the semantic problems involved?

Becoming an expert in data structures is like becoming an expert in sentence structure and grammar. It's not of much value if the thoughts you want to express are all muddled.

The information in the system is part of a communication process among people. There is a flow of ideas from mind to mind; there are translations along the way, from concept to natural languages to formal languages (constructs in the machine system) and back again. An observer of, or participant in, a certain process recognizes that a certain person has become employed by a certain department. The observer causes that fact to be recorded, perhaps in a data base, where someone else can later interrogate that recorded fact to get certain ideas out of it. The resemblance between the extracted ideas and the ideas in the original observer's mind does not depend only on the accuracy with which the messages are recorded and transmitted. It also depends heavily on the participants' common understanding of the elementary references to "a certain person", "a certain department", and "is employed by".

1.1 ONE THING

What is "one thing"?

That appears at first to be a trivial, irrelevant, irreverent, absurd question. It's not. The question illustrates how deeply ambiguity and misunderstanding are ingrained in the way we think and talk.

Consider those good old workhorse data base examples, parts and warehouses. We normally assume a context in which each part has a part number and occurs in various quantities at various warehouses. Notice that: various quantities of one thing. Is it one or many? Obviously, the assumption here is that "part" means one kind of part, of which there may be many physical instances. (The same ambiguity shows up very often in natural usage, when we refer to two physical things as "the same thing" when we mean "the same kind".) It is a perfectly valid and useful point of view in the context of, e.g., an inventory file: we have one representative (record) for each kind of thing, and speak loosely of all occurrences of the thing as collectively being one thing. (We could also approach this by saying that the representative is not meant to correspond to any physical object, but to the abstracted _idea_ of one kind of object. Nonetheless, we do use the term "part", and not "kind of part".)

Now consider another application, a quality control application, also dealing with parts. In this context, "part" means one physical object; each part is subjected to certain tests, and the test data is maintained in a data base separately for each part. There is now one representative in the information system for each physical object, many of which may have the same part number.

In order to integrate the data bases for the inventory and quality control applications, the people involved need to recognize that there are two different notions of "thing" associated with the concept of "part", and the two views must be reconciled. They will have to work out a convention wherein the information system can deal with two kinds of representatives: one standing for a kind of part, another standing for one physical object.

I hope you're convinced now that we have to go to some depth to deal with the basic semantic problems of data description.

We are dealing with a natural ambiguity of words, which we as human beings resolve in a largely automatic and unconscious way, because we understand the context in which the words are being used. When a data file exists to serve just one application, there is in effect just one context, and users implicitly understand that context; they automatically resolve ambiguities by interpreting words as appropriate for that context. But when files get integrated into a data base serving multiple applications, that ambiguity-resolving mechanism is lost. The assumptions appropriate to the context of one application may not fit the contexts of other applications.

There are a few basic concepts we have to deal with here:

* Oneness.

* Sameness. When do we say two things are the same, or the same thing? How does change affect identity?

* What is it? In what categories do we perceive the thing to be? What categories do we acknowledge? How well defined are they?

These concepts and questions are tightly intertwined with one another.

Consider "book". If an author has written two books, a bibliographic data base will have two representatives. (You may temporarily think of a representative as being a record.) If a lending library has five circulating copies of each, it will have ten representatives in its files. After we recognize the ambiguity we try to carefully adopt a convention using the words "book" and "copy". But it is not natural usage. Would you understand the question "How many copies are there in the library?" when I really want to know how many physical books the library has altogether?