Taking Information into Accounts

Michel Biezunski (June 27, 2017)

Notes

Who is this book for? This book is for people who are interested by a big picture on how information systems work. Lawmakers, academics,

Contents

Accountability of Information

Information is, generally speaking, not accountable. It is hard to figure out where information is coming from, by whom it is being accessed and where it ends up going. This was not a problem in the infancy of the Internet and the World Wide Web, when online information existed to fill the needs of scientists and technologists who were voluntarily and enthusiastically sharing information with each other, in order to build an interconnected world. But since online information has become the principal medium for all businesses and governments alike, new problems have arisen and the lack of accountability has become a major issue opening new threats including cyberattacks, identity theft, spreading of fake news and propaganda, dismantling of critical computer-powered power plants. If we want to overcome these hurdles, reach the point where the information society is reaching to its full potential, we need to find better ways to account for the information we deal with. This article is about understanding what is at stake and trying to uncover the various conceptual layers that are needed to grasp to ultimately reach a fully operational information society, where accountability is the guarantee for building and maintaining trust.

Accountability is well implemented for handling money. Businesses, and also individuals, account for money exchanges, a mandatory requirement for tax purposes. Accounting can be somewhat complicated, but it is based on a simple idea, which is that any amount of money comes from somewhere and go somewhere else. It relies on recording transactions between money accounts. Even cash out of pocket is considered an account. An account statement is a record of everything that has ever happened regarding one account. A payroll and a bank statement are examples of account statements.

In order to extend this practice to information exchanges, we should first acknowledge the premise that no information exists in the vacuum, i.e., that it is always related to at least another information. Therefore, every piece of information can be seen as an account, related to other pieces of information. And keeping a log of all the connections going to and from any piece of information is the equivalent of a money account statement. Being able to fully record what any piece of information is connected to, be it another piece of information or a process, makes it fully accountable.

Information all the way down

A piece of information has many facets. It is created at a certain time, by an author. Its name itself is a string of characters, encoded in a given character set. The location where it is stored, the number of times it is accessed, are also related to that information item. Between the information itself and its computer representation as a sequence of bits (0 and 1), many intermediary steps exist, and they are are usually taken for granted, and ignored.

When information items get decomposed into more elementary components, the amount of overhead created can be enormous. Most of it is way beyond the reach of the capability of current computer systems. In most situations, they are useless, but if we are looking for accountability, things matter more. For example, it is enough on a tax return to declare the total income earned for a year, but in case of an audit, it becomes necessary to document the figure by giving evidence of every quantity involved in the total. Similarly, it is generally useless to use such a microscopic view on information and its components, except for cases where accountability is required. As computer technology evolves, new possibilities are looming at the horizon with quantum computing, which would multiply the capabilities offered by current systems. This is a domain which is not yet ready for prime time, and it is likely that fully accountable information systems are a future prospect rather than an immediate endeavor. This is not a reason for it to be neglected. The need for improved accountability can also play as an argument in favor of developing the increased power enabled by quantum computing. Therefore, we are interested here to pave the way for the future.

In the meantime, it is possible to filter the analysis so that it is reduced to what is technically possible. For example, we could be interested by looking at the number of names used for the city of New York, but we may discard analyzing each name as a sequence of letters, or care about the character set or the encoding. As information is digitized, the internal representation seen from a computer perspective amounts to sequences of bits (0 or 1), that are represented in a way that makes sense to a human user thanks to a number of software layers, including operating systems, character encodings, computer languages, and software applications with sophisticated user interfaces. The same information therefore is presented differently depending who or what is looking at it. "New York" can be immediately understood by humans as a city, while it is also a sequence of letters ("N" followed by "e" followed by "w" etc.) in a specific character set with a specific encoding. Or, an operating system can see it just as a sequence of bits in a specific memory location. A full accountable information system must take into account the various layers. That can be used for example to explain why 自由編輯个維基 百科 which is the Chinese representation of New York can't always be displayed correctly depending on the configuration of the computer. If we are in an environment where we need to account for character sets, every single piece that plays a role in the transformation processes needs to be present, including the various encodings.

Processes vs. relationships

Processing a piece of information is similar to relating it with another piece of information. Saying "2 + 3 = 5" expresses an equality relation between the operation "2 + 3" and the value "5". This expression describes a process called addition, with the left part

playing the role of "before" and the right part playing the role of "after". Now, saying just "2 + 3" expresses the fact that "2" and "3" are related by the operator "+". This arithmetic operation is made of two operands (2, 3) and one operator (+). In this latter example, we are not describing the result of the process of adding the numbers, but simply asserting the fact the these two numbers are associated through a process of addition. This expression is purely descriptive. From a purely informational point of view, a process is simply a kind of relation. Recording which processes are allowed on information items is important for accountability purposes, before they are not actually activated, or even if they are not activated. The notion of relations naturally extends beyond just processes. Descriptions can be used as well to describe other types of relationships. Saying that "New York City is in United States" is a semantic relation. It expresses the relation that the city of New York entertains with the country United States. It can be decomposed into three parts: "New York City", "is in", "United States".

Binary relationships

There can be a variety of relationships between information items. Organizing the relationships into logical blocks, and creating an architecture of relationships can be complex. Taxonomies are hierarchical relationships used to describe how concepts are related to others. Well-known examples include the taxonomy of living species, library catalogs classifying sciences and disciplines into domains and subdomains. Taxonomies used in library science usually feature two main types of relationships: "broader term" and "narrower terms". Other information systems are organized according to more complex schemas. For example, family trees are strictly hierarchical (parent - child relationship), but the relationships between spouses is not hierarchical. Spouses come from other trees. Also, the evolution of society has created many situations in which the classical family tree representation doesn't hold.

A graph, or networked representation of the relations between information items, is widely open, because it doesn't constrain the relationships between information items to be hierarchical. Hierarchical taxonomies are just one possible form of graph. An information item can be related to multiple others, by the same kind of relationship or by others. For example, a woman can have many children. It is equivalent to say on the one hand that A is the mother of B, C, D, or to say on the other hand that A is the mother of B, and A is the mother of C, and A is the mother of D. The first expression is called a n-ary relation, whereas the second expression is a functionally equivalent set of binary relations. Mathematicians have established that there is a strict equivalence between n-ary relations and binary relations. Under the hood, it is possible to converge to a representation uniquely relying on binary relations.

As a result, we can assert that the whole world of information can be represented ultimately as a set of binary relations. The transformation that takes as input a set of n-ary relations and outputs them as binary relations can be described as a flattening operation.

Perspective in a flattened world

Perspective comes into play when representing a three dimensional scene on a two dimensional flat surface. The scene is seen from a viewer's point of view, whose eye is in a certain location. Objects are scaled depending on their distance. Remote objects will appear smaller than closer objects. Parallel lines are represented as converging in a point called a "vanishing point". The laws of perspective have been studied by mathematicians and artists. In information land, the number of layers to uncover, while opening where a piece of information leads us, depends on what we want to see. Auditing the information is like seeing it with a microscope. It helps us focus on some aspects of it, while explicitly ignoring others. We may be in an environment where the interesting matter is located at a high level, and where there is no need to explore what is beneath the surface. Or we may be in an environment where some people have access to more layers of information than others (typically where some information is "classified"). The level of information made visible becomes a matter of perspective. In information modeling terms, a perspective is a view that contains filters. Making information accountable is done by defining the perspectives in which we want to look at it. Furthermore, several perspectives can be defined on the same information repository. There is no universal perspective that makes the information accountable, once for all.

Metadata is also data

Information technology traditionally distinguishes between data and structure. The structure of data, in a database, is defined by a schema. The schema is a framework containing types of information allowing us to identify the nature of the data we are dealing with. For example, a contact database schema would contain fields for the last name, first name, telephone number, email, etc. Another commonly used distinction is the one between data and metadata. For example, in a document, data is considered to be the content, while metadata contains fields such as the author name, the creation date, etc. Sometimes, metadata are added automatically, sometimes they can be created by the user.

In order to enable full accountability, the first step is to consider that all information is equivalent. Data, metadata, field name, an xml tag name -- also known as a generic identifier --, a character, a byte, etc., should all be treated as information units. They are all related to at least another one. Saying that "New York is a city" is not different than saying that "New York is in the United States". However, the notion of "city", when considered a type, is privileged over the notion of "United States". This vision of information united belonging to types is a shortcut enabling to filter better information according to types and distinguish between types and instances. This vision is the basis for many computer systems dealing with the way information is organized. It enables for example to retrieve all things that are cities. City is metadata whereas New York is considered data. In the second phrase (New York is in United States), the relation is considered as a semantic relation between two instances (New York as a city, United States as a country). But it is possible to consider New York being part of "all things in United States", and retrieve all of them the same way we list all cities. Should United States therefore considered a type to which New York belong? Not necessarily. This example shows that the traditional distinction between data and metadata is somewhat artificial, and only applies in a context where a "schema" containing predefined types is very rigidly defined. Many relations between information items can't be described using this simple relationship. The difference between data and metadata doesn't hold when trying to analyze what information is at a deeper level. There can be any kind of relations between two pieces of information. For example, the fact that the string "New York" starts with "the letter N" is a relation between two pieces of information. The list of strings starting with "N" is typically what gets collected in a dictionary. Therefore, "New York" is to be found in the account statement for "Strings starting with N". This relation is useful, although it's so "obvious" that it usually doesn't need to be expressed explicitly. It is taken for granted by those of us who use an alphabetic character system.

Furthermore, it is interesting to introduce a distinction between the things and the names by which they are designated. That distinction corresponds to the difference defined in linguistics between "signified" and "signifier". New York may well be the name of a city, but it is also the name of a state and the name of a county. It is not the only name for the city, also referred to as "New York City", "Big Apple", etc., and it is not the only name for the state, also referred to as "New York State", "NY", or "Empire State". The New York county is also called "Manhattan". An information unit therefore can not be reduced to its name, even if its unique. An information is an unnamed object, which has a mental representation, to which names can be assigned. The name itself is an information unit, related to the information unit that it describes. And the relation is itself an information unit. For example, the fact that "New York" is called "New York" has a historical background (1667). When that city changed its name from "New Amsterdam", it was still the same city.

Consider now the sentence: "Nueva York is the Spanish name for New York". Actually, this proposition is misleading. It would be more accurate to say that Nueva York is one possible name for this thing that some call New York. This name happens to be in Spanish. But Spanish is the English way to designate the language designated by its own speakers as Español. In other words, Spanish is English, meaning that "Spanish" is an English word. Even that proposition is ambiguous. English can have several variants: "organise" is English and "organize" is English. More precisely, "organise" is the British variant of English spelling and "organize" is the American variant of English spelling. Therefore even a proposition as straightforward as "this word is in English" doesn't pass the smell test for accountability.

Computer systems are universally based on unique names and are not immune from ambiguity, despite their claim to the contrary. Unique identifiers are assigned to objects, therefore representing each object unambiguously. But some systems consider that it's acceptable to reuse the unique identifier of a deleted object for a new object, because, once the object is deleted, its unique

identifier becomes available for reuse. But that may turn out to become a problem when tracking each object individually. Some information may have been lost.

Luca Pacioli

Luca Pacioli was an Italian mathematician and Franciscan friar, who was born around 1447 and died in 1517. He also is known as the "father of accounting", after having published on book describing the bookkeeping method used by Venetian merchants during the Renaissance. He recommends, in Particularis de Computis et Scripturis ("The Rules of Double-Entry Bookkeeping"), "to arrange all the transactions in such a systematic way that one may understand each one of them at a glance, i.e. by the debit (debito—owed to) and credit (credito—owed by) method [p. 16]. This is very essential to merchants, because, without making the entries systematically it would be impossible to conduct their business, for they would have no rest and their minds would always be troubled." This method, known as "double-entry accounting", describes monetary transactions between accounts. He was a mentor to Leonardo da Vinci and wrote several books which were syntheses of knowledge of mathematics at the time. He was interested in the aesthetics of geometry, wrote the "Divine proportion" and discussed how painters used perspective.

The Data Projection Model

The Data Projection Model is a description of any transaction between two information items. Two information items are similar to operands in an arithmetic expression, and the transaction is indicated by the operator that indicates the process or the relationship between them. Each combination of "operand-operator-operand" is called a perspector. Every perspector is unique and can : operand [operand | operator | operand] . For example, the mathematical expression `2 + 3` can be noted as the perspector `[2 | + | 3 ]` . Perspectors can be nested, i.e. one perspector can be used as an operand in another perspector. `[ [2|+|3] | = | 5 ]`