ON THE TABLE

Table of Contents

Introduction

In the historic changes that happened since 1980, what have I witnessed and what have I done?

What I have witnessed

- Personal computers change office work.
- Transition from print to digital within the publishing industry
- The emergence of the Web
- Social media. Connecting people for the better and for the worse.
- Cloud and Big Data
- Artificial intelligence and the transformation of intellectual work

What I have done

- Mix text and database
- Markup standards
- Knowledge Base Architecture (Topic Maps)
- Software Development: From scripts to full-stack
- Bridge between IT and end users

From Print to Digital

Procedural Markup and Semantic Markup

In the early 1980s there were still a few books and newspapers that were composed with lead characters. Typographers were getting each character from cases (there were lower cases and upper cases).

Phototypesetting, though, was a technique that was used in most cases. A photograph of every page was created by typing the text on a computer. There were special keys for font changes, such as font size, italic, bold, etc. Each typesetting system used their own proprietary system and dedicated keys.

Each key used to indicate a font change appeared as a special code. The set of these codes is called "markup". Each document was made of text content surrounded by markup. This markup is called "procedural markup" because it is there to trigger a new procedure.

In 1969, a language was invented at IBM by Charles Goldfarb, Edward Mosher and Raymond Lorie. The language was called GML, using the initials of their surnames, and became known as the Generalized Markup Language. The difference between GML and the proprietary typesetting language was that GML was using descriptive tags to indicate to the machine that it needed to perform a series of procedures. For example, list items were marked as :li and were triggering the creation of an indentation and a bullet.

Goldfarb went one step further and created a meta-language, i.e. a language to create languages that allow any variation of the document structure to be described. Each document structure is encoded into a formal "document type definition". The formating system needs to be configured to interpret the specific tags contained in a specific document type. Each document could be parsed to determine whether it was valid or not according to the definition it was declaring. This new meta-language is called SGML, and was finalized as an ISO standard in 1986. It clearly separates the content from the presentation, as the markup can be interpreted differently for each specific formatting system. This allowed the same source document to produce a printed version and a digital version (at that time aimed mainly at Cd-Rom).

Furthermore, the infinite flexibility of SGML allowed markup to represent not simply procedural operations (such as changing the font into italics), but to represent semantic information. For example, a book title, which is traditionally printed in italics, could be marked up as a book title, and the fact that a book title should appear in italics was left to the instructions to the formatting system.

It also allowed processes to be performed on each document to enable the extraction of lists of data belonging to a common type, including the automatic creation of tables of contents, or indexes. In other words, a structured content similar to a database could be included within a document, opening many new possibilities for documents. This method was quite successful, far superior to the proprietary tagging systems used by typesetting machines.

SGML was completed with a standard language for formatting specifications, named DSSSL, The Document Style Semantics and Specification Language, which was a complex standard describing style sheets and procedures to apply to a document during the formatting process.

SGML was adopted as a required language by the US Federal Government which required its vendors to provide document in that format. That requirement was enforced by the Department of Defense. This constraint played a big role in the adoption of SGML.

At the same time, personal computers invaded the office. As they were replacing typewriters, they were first considered machines for secretaries and accounting personnel. As machines got smaller and as their capabilities grow, managers and even executives starting to use computers.

Every word processing package had its own format. The first major product was named WordStar, soon to be replaced by WordPerfect and Microsoft Word. Each software stored documents in its own format. There was an attempt to list all the features ever found in word processors, and make the list available under a standard format called the Office Document Architecture. This standard was presented as a reasonable, down-to-earth, realistic approach to SGML, which was considered abstract and esoteric, and too flexible to be considered by professionals. Instead, users had at their disposal conversion tools that helped them transform a document format into another when needed, and that was enough to ensure interoperability when needed. Microsoft Word became the de facto industry standard as its market share grew.

SGML, instead was promised to a bright future. It served as an inspiration for HTML, the language that made the Web happen, and was later simplified as XML, a language that became prevalent during the 2000 decade and served not only for documents, including Microsoft Word, but also to exchange all kinds of data from a variety of structured sources.

The structured markup approach was applied to musical notation. An integrated model including hypermedia, addressing, scheduling, rendition was released in 1992 in a standard called HyTime, an abbreviated form of

Hypermedia/Time-based Structuring Language. HyTime was not widely adopted, but its pioneering breakthroughs and advanced vision have served as starting points to develop new architectures and languages, such as XPath and Topic Maps.

XML as the Lingua Franca

In 1998, a simplified version of SGML, was adopted as a World Wide Web specification. XML was widely adopted, not just by the publishing industry to create books and technical documents, but also as a way to interchange structured data from configuration files, or output from databases.

Soon after, satellite standards were created to describe how information can be located within structured documents (XPath), how information can be formatted and more generally transformed (XSL and XSLT) and how information can be queried (XQuery).

Microsoft adopted XML as a format for their office documents. The "x" within the docx, xlsx, or pptx file extensions meant that these files are in XML.

The constraints of validity imposed by XML were both a blessing and a curse. The requirement that any document should be parsed by an XML-aware system was overwhelming in many cases, and added a constraint that was considered un-necessary. Furthermore, the web browsers natively understood another notation, Javascript. An alternative to XML was developed as the "Javascript Object Notation" (JSON) and became prevalent among web developers and cloud-based applications.

XML is still used in heavily structured documents such as those used for technical documentation. But its use within the publishing industry overall has not met the expectations of its initiators. Today, less people are using XML and the generation of programmers who started

their career circa 2010 would rather avoid using it, as much as possible.

A Web of Links

Hypertext and the World Wide Web

The Arpanet project founded by the United States Department of Defense was launched in 1969. It was a way for the military and academics to exchange information, email, bulletin boards across communities. It evolved and took several names, and in 1974 became known as "Internet", an abbreviation for "Internetworking". NASA developed further protocols to enable scientists to access information across that network all over the world. Its decentralized nature was the major innovation, and helped establish this network as a sustainable, and highly scalable system.

In 1989, a physicist working at CERN, the particle accelerator located near Geneva, Tim Berners-Lee, proposed a universal linked information system, that added to the Internet the ability to link together the information content available on the servers. The language he used for linking, HTML (Hypertext Markup Language), was extremely easy to use, and the first browsers able to activate the links were deployed around 1993. The World Wide Web was born, and it changed everything.

The World Wide Web, now simply called the Web, is a network of interconnected resources connected through links. The technical developments didn't stop there. Soon, the need to organize linked information started to emerge. Two main representation of knowledge networks were developed during the 1990s.

The Resource Description Framework (RDF), which became the foundation of what became the Semantic Web, is used to deploy rule-based ontologies, and provides powerful querying techniques to gather information all over the

Internet based on shared properties. Libraries have adopted the Semantic Web as a way to make their catalogs available online, and biological research information is widely available using this representation.

The Topic Maps architecture was developed in parallel, to emulate the traditional navigational aids provided within printed books, such as indexes, tables of contents, cross-references, glossaries. Whereas RDF was focused on automating information retrieval, Topic Maps was focused on providing the freedom for humans to create and maintain access to subjects with no limits on the complexity or the nuances that were needed to describe knowledge networks.

The Semantic Web

The Semantic Web's foundation is a graph of statements, formalized in the specification called the "Resource Description Framework" (RDF). A statement asserts that a subject is related to an object through a predicate. In the statement "An elephant is a mammal", the subject is elephant, the object is mammal and the predicate is "is a". As a subject can be related to more than one object, and an object can be related to more than one subject, the set of objects and subjects comprises a graph. A node used as an object in one predicate can serve as a subject in another predicate. That would be the case when we assert that "A mammal is an animal".

The goal of RDF is to provide a semantic layer to the Web that helps distinguish the metadata from the data. It also aims at facilitating automation by enabling semantic queries to be performed on the data, using predicates. RDF is used as a basis to provide rules that are used to automate processing of data present on the graph.

One of the major constraints - and benefits - of RDF is the fact that every resource, whether it's an object, a

subject, or a predicate is an entity that is located on the Web, and therefore is addressable through a uniform resource identifier, previously called uniform resource locator or URL.

RDF has been used as a foundation to create taxonomies, i.e. hierarchical sets of terms. The "Dublin Core" is used in many libraries and is used for online catalogs.

Rules to process RDF graphs have been formalized into ontologies, using a language called the Web Ontology Language (acronym: OWL).

Semantic Web applications have been used in many academic projects, within government entities, especially within the European Union. It is being used extensively in bio-medical research.

Topic Maps

I happen to have been instrumental in creating the Topic Maps architecture. For that reason, the part on Topic Maps is more developed in this book.

What are topic maps?

Our brain works by associating thoughts. A thought can be expressed as a set of subjects. We communicate with others by talking about subjects in conversations. Our mental representation of what these subjects mean does not uniquely relate on the name we use to designate them. The distinction between the "signified" - the meaning of a concept - and the "signifier" - the symbols used to express it - is at the foundation of modern structural linguistics and semiotics. The name is purely a label that we use as a shortcut but names are different depending on the language we speak, a name can be used to mean things that have nothing to do with each other

(homonyms, context-dependent names), and a thing can be labeled with different, equivalent names (synonyms).

The way we understand what a subject means is complex and is a major challenge to overcome when using computers to manage knowledge-based systems. It happens during conversations that the most important things we communicate is not something that we say, but something that is implied to be understood either by the tone we use to say it, or some gesture that communicates something important not conveyed by the words themselves. Computers are not well equipped to deal with ambiguity, misunderstandings, context-dependent information, or non-verbal communication. They can handle correctly the low-hanging fruits, which represent a significant part of information, but they do not cover the entire spectrum of human communication.

Knowledge systems, including computerized taxonomies and ontologies, usually require an agreement on a name used to identify a concept. These names are supplemented by metadata that describe the inherent characteristics of the object under consideration. These representations work as long as the concepts, and the categories to they belong, are well-identified, but they may fail to grasp the concepts described, as time goes by, as when new subjects are introduced and the distinction between previously well-described objects becomes blurry.

Topic maps provide independence between the data that are being described and the subjects that qualify them. For example in a book index, there is no requirement that the term in the index be identical to a word used in the page. What matters is the fact that the abstract concept used in the index corresponds to a subject that can be derived from reading the content located at the corresponding page. A topic map is an overlay which provides a perspective on the knowledge that is described. The Topic maps paradigm consists of the ability to point to any kind of data from outside, without the constraint of having to rely on what is already there. In fact, several perspectives can be created on top

of the same data set, that can provide filtered information aimed at different audiences.

The Topic Maps paradigm aims at providing a model that enables these ideas to be processable by computers. The abstract subjects are represented by proxies, which are compound objects with properties. These proxies are called *topics*, to indicate that they represent a subject as a location (from the greek word *topos*) in a semantic space called a topic map. They are nodes —vertices—, in a graph that can be navigated following the connections —edges— between them or to the documents serving as sources.

The Topic Maps international standard provides a way to interchange topic maps between applications. By design, it does not prescribe any way to specify topics nor any particular semantic for their relations.

The Topic Maps Model at a glance

The Topic Maps model is based on establishing one location per subject. A subject is any subject of conversation, which can represent an actual thing, or an abstraction, a relation, a point of view, etc. The subject exists independently of the names used to designate it.

A topic is a computer representation for a subject. A subject is defined for human consumption by a "subject indicator" which contains an unambiguous definition. It is identified by a computer as a unique object by a "subject identifier".

When an information resource is considered a subject, it is referenced by a "subject locator", indicating where it can be found. For example, if the resource is available on the Web, its subject locator is a Uniform Resource Identifier, or more accurately an IRI (Internationalized Resource Identifier), using Unicode characters instead of just the ASCII character set.

A topic has inherent properties, including its names, occurrences and associations.

There can be multiple names for a topic, and, since the name is not uniquely identifying a topic, one given name can be used in different topics. The "scope" attribute is used to delimit the domain in which certain properties are applicable. For example, a scope can be used to specify a scientific domain in which a name is valid, the language for that name, or to disambiguate homonyms. The scope can also be used to define contexts for relationships and occurrences.

When a topic applies to an information resource, the relation between the topic and the resource is called an occurrence. In other words, the subject for this topic occurs in this particular resource. An occurrence can be typed, and scoped, if needed.

Topics can be interconnected, and the relations between topics are called "associations". Each association belongs to a type, and the role that every topic plays in this association is described by a role property. The role property has a player (the topic) and a role type (the nature of the role played).

Since the purpose of topic maps is to provide a single location per subject, regardless of the name(s) used to describe it, the procedure for merging topics is precisely defined by the data model. When two topics merge, their properties are added, except when they are redundant.

In order to improve interoperability between Topic Maps-enabled applications, some relationships are defined, but are not mandatory. These relationships include the type-instance relationship, the supertype-subtype relationship. In addition, a special variant of name called "sort name" is used to indicate strings that are used when sorting the names.

Subjects

The Topic Maps international standard does not prescribe how a subject should be defined. This empowers the creators of a topic map with the ability to choose the conceptual basis on which they establish subjects.

> A subject can be anything whatsoever, regardless of whether it exists or has any other specific characteristics, about which anything whatsoever may be asserted by any means whatsoever. In particular, it is anything about which the creator of a topic map chooses to discourse."
> http://www.isotopicmaps.org/sam/sam-model/#d0e746

The rationale for this definition, which in practice amounts to the absence of a definition, is to make topic maps the ultimate merging space for information coming from a variety of sources. And the reference to the process of creation of a topic map also indicates that topic maps can be used to capture information that was created by humans as well as by machines.

Traditional navigational aids

The Topic Maps model is a knowledge representation aimed at describing units of meaning and their connections. It has been designed to capture the traditional navigational aids, such as indexes, thesaurii, glossaries, catalogs, dictionaries, tables of contents, cross-references, bibliographical references. Indexes are a list of terms-topic names-presented in alphabetic order together with the locations where they are relevant-occurrences: in printed books, occurrence indicators of topics are page numbers. Thesaurii are topics connected by a predefined set of relationships: generic (related), or hierarchical (broader/narrower). Glossaries are a list of topics followed by an explanatory, which is an occurrence of the topic which plays the role of definition. Tables of contents are list of topics whose occurrences are the chapters, or sections, in which they

occur. References are links to other occurrences of the same topic: they are called cross-references if they point to another location in the same text, or bibliographic references if they point to another text which needs to contain a way to locate it, i.e. its author, title, date of publication, or URL.

Taxonomies and Topic Maps

Taxonomies are classification systems, based on assigning categories for each item. Categories are often organized into a hierarchy. For example, a taxonomy of living creatures could consider animals as living creatures, mammals as animals, cats as mammals, tigers as cats, etc. A hierarchical relation is a relation that can be expressed with a statement in which the middle term is "is a": an animal is a living creature, a mammal is an animal, a cat is a mammal, etc. This relationship is sometimes considered to be a "class instance relationship".

The description of the ways categories are organized is called a schema. A database schema, or an XML schema, does not only describe the containment rules, but also the data types, and additional constraints. For example, a valid date could be defined as a positive integer not higher than the current year. Schemas complemented with rules and constraints are referred to as ontologies.

Hierarchical relations can be expressed as associations between topics representing items and topics representing categories. Therefore topic maps encompass taxonomies and ontologies, with the addition of rules and constraints in a separate layer. The Topic Maps model is itself an ontology, in the sense that it is made of constructs, such as "topic", "name", "association", "occurrence" that have rules attached. But it is an ontology capable of representing other ontologies. The exact nature of the constraints in the Topic Maps model, and the levels to which the rules applied has given rise to extensive discussions, and there are several possible

interpretations of how to use the Topic Maps ontology. For example, there are contexts where topic names should be considered topics, associations should be considered occurrences. An extensive practice of topic map implementations shows that the power of the Topic Maps ontology is its flexibility, whereas other ontologies are adopted because of the constraints that they provide.

Reification

A relation type can itself be considered a "topic". The ability to consider relation types as topics enables integration between datasets coming from heterogeneous sources in which the schemas use different representations for certain types of relationships, but are all considered equivalent in an integrated environment. For example, relations expressed as follows: "is a", "instance of", "belongs to", "narrower than", "component", "child of" could all be seen as a generic hierarchical relation type. But this can be fine-tuned depending whether it makes sense in a given application environment.

Furthermore, an instance of a given relationship can be reified, i.e. it can be considered a topic per se: the assertion "London is a city" which contains relations between three topics ("London", "is a" and "city"), is itself a topic. It can therefore serve as a node in an association with other topics. The fact that London is a city has already been attested during the first century before AD is an association between the topics "London is a city" and "First Century before AD" whose semantic is represented by the role "attested in" (itself a topic).

Data vs. Metadata

The Topic Maps paradigm obliterates the traditional distinction between data and metadata. The classic distinction is based on the view that data represents the content of information, while metadata adds contextual information about the information. In a taxonomy,

metadata is the category to which data belongs. Metadata also serves to add information outside of the domain of the content. A book on the history of painting is about art, but it also has an author, a publisher, a date of publication, all considered metadata, because they do not refer to the subjects in the content of the book. The distinction between data and metadata has become even more acute when information is stored on computers using relational databases, where tables have a header (or a field name) that establishes the field and cells containing the data.

The principle of separation between structure and content used by markup languages such as XML or HTML, separates the data content from the metadata encoded as tags. In a library, a catalog is metadata, while the books and other source materials are considered data. But this distinction is not as clear as it seems. For example, a library catalog notice which is technically part of metadata includes the name of the author, considered data content inside the book. Inside the book, though, the author name is considered to be part of the data, since it is part of the title page which is part of the content. This illustrates the fact that someone's data is someone's metadata. The same information seen in the perspective of Topic maps is described as a graph of interconnected topics. The fact that there are no constrinats on what the semantic of the relationship represents, results in an unlimited number of possible connections between information items. Flattening is of paramount importance for data integration. It can be used not only to collect information from a variety of resources, but also to keep track of the provenance information, by creating topics and connections to the original repositories characteristics from which integrated information comes from.

A topic map can be construed as an added information layer, which, instead of being aimed at replacing a data structure, points at it in order to shed a new light on its components for usages that were unforeseen at the time

the data models were created. It can be used also for forensic analysis, containing not only the provenance information, but also the processes through which the information came into existence. Once the information exists as a topic map graph, it can be displayed in a variety of ways to provide custom perspectives, and it can be recalculated in a variety of ways to provide new visualizations including new combinations of existing elements, as in a kaleidoscope.

The essential distinction can even be pushed one step further by describing processes as relationships, using the same underlying structure. Robert Glushko writes:

> The vanishing difference between data and metadata, and the equivalence between a semantic relationship and a process, opens a whole range of applications for integration of information coming from very diverse sources, as well as the ability to conduct forensic analysis by describing processes that explain the provenance of potentially any information item.[Robert J. Glushko, The Discipline of Organizing: Professional Edition, O'Reilly, The MIT Press, 2013, section 1.3: The Concept of "Resource"].

The equivalence between a semantic relationship and a process is not something has been explicitly part of the Topic Maps paradigm, but it is a consequence of it that has been developed further in a model called the "Data Projection Model" . In this model, the arithmetic expression "2 + 3 = 5" is seen as a relationship between two topics, "2" and "3", with the relationship being expressed as an addition, abbreviated with the "+" sign. The equal sign can be expressed in two different ways in a topic map context, either as an association between the reified relation "2 + 3" considered a topic and the topic 5, associated through a relationship whose semantic is the equality. It can also be interpreted by saying that the topic "2 + 3" has an alternate name, i.e. "5".

Addressing complexity

Information content is inherently complex and ambiguous. The assertions "London is a city" and "the City is in London" have a completely different meaning, which is expressed by the capitalization of the word "city" that represents respectively a generic term to describe a group of people living together in a dense area, and a unique term coined to designate the Financial District in London. In addition to the fact that humans speak different languages, the polysemic structure of a human language, results in differences in understanding depending on where people come from, their education level, and even the mood in which we are or the tone in which we speak, that can alter the meaning of what we say. In writing, ambiguities still subsist, and misunderstandings happen. The use of computers to handle information leads to denying that ambiguity exists. As computers are not inherently able to cope with these kinds of complex or ambiguous situations, the way information is stored tends to deny these multiple levels of complexity, forcing to a simplification of interpretation. Although most of times, especially in a professional context, there is an implicit agreement on what things mean, there is always a chance that the meaning is missed, especially when a considerable amount of information is collected.

By going Semantic, the World Wide Web Consortium has moved in a direction of trying to connect knowledge based on common understandings of concepts, best exemplified with the Linked Data architecture, where every imaginable topic gets an addressable web address (URL), and therefore can be referenced by others. The main problem with this approach is that by setting the application span so large, it diminishes the value of the information retrieved, because it's often taken out of the context in which it is queried.

The potential inherent to the paradigm represents a long-term, enduring, applicable functionality that may

well persist long into the future, regardless of the formats used to interchange information.

Database systems and Topic Maps

A database is a storage system in which data are structured in order to facilitate queries based on common criteria. A relational database model has a structure similar to rows in a table, with well-defined column headers. Each column header represents the type to which a specific data (cell) belongs. A database schema contains the definitions of the fields allowed as well as the data type to which they are required to belong, in order to provide validation. In addition, relational databases provide the ability to join tables by fields. For example, an invoice table can be related to a customer name, which is an entry in another table.

Object-oriented databases consider that data are stored as objects with properties, for example the data type: a "person object" may be assigned a "date of birth" property that is constrained to be an integer with four digits. The advantage of object-oriented databases is that objects can be reused across several higher level objects.

Graph databases store data as nodes and edges (links). The edges have their own properties, for example bidirectionality. The main difference between these three kinds of databases — relational, object, and graph — is the interface they provide to their users. In a relational database, the records are data corresponding to predefined fields, optionally joined to other records through foreign keys. In object databases, a related object could be described as a property of the source object, and in a graph database, the relationships themselves are considered first class objects and can be addressed as such. Performance issues, query languages, reusability and familiarity with the paradigm are factors that are taken into account while deciding which data storage is best fitted for a given environment.

Topic maps can be implemented using either of these database technologies. A topic can be defined as a record in a relational database object, as an object in an object-oriented database, and as a node in a graph-oriented database. The way a topic maps system is implemented does not necessarily impact the user experience, except for constraints that are imposed by an implementation's specific requirements. When a topic maps-based system contains export functionalities, it is posssible to use the data coming from another system regardless of the internal database on which it is built.

Humans vs Machines

Managing knowledge has been performed by human beings, long before computers were used. Therefore, a subject matter experts have accumulated lore to do this, and the way they operate is somewhat different from what computer systems require. Recently, subject matter experts have been learning how to use technology to do their work. And technologists still have to learn about the specific requirements needed by knowledge experts.

Since a long time, subjects are used to describe contents, either in classification systems such as library catalogs, or at a finer granularity level, in book indexes: a book record in a library catalog is about one subject, while an index entry refers to a subject in a given location in the book, usually indicated by the page number. The way subjects are created may also greatly differ, on a conceptual level. Isolating a concept by naming it and establishing its subject are not easy tasks. This is at the core of philosophical and epistemological theoretical world views. The library and information science curricula consider this at the core of their teachings. Practically speaking, building agreements within a community that has vested interests in sharing certain well-defined concepts or products is a complex task. And it not guaranteed to be stable over an extended time period, as new information may break the previous consensus with the introduction of new,

unhearded concepts. A comprehensive review of these complex matters is presented by Birger Hjørland in the article "Subject" of the encyclopedia http://www.isko.org/cyclo/subject, who introduces a distinction between the conceptual analysis and the translation stages, i.e. the assignment of the applicability of subjects to indexing and classification.

The transition from print-based technologies to digital-based technologies has modified the scale of knowledge management, and has triggered profound changes in the way knowledge management is handled.

Several attempts have been done, and continue to be done, to circumvent the inherent difficulty of qualifying the semantic of information items by using algorithms to replace human determination of subjects by automatic processes. The availability of the Internet as a global network and the World Wide Web as a universally accessible knowledge repository offers an opportunity for connecting concepts at a scale that changes the qualitative nature of the landscape. At the scale of one book, recording all words into an index and pretending that the value of such an index is equivalent to a humanly crafted index has been ridiculed. Just getting a concordance table of all words used is interesting for statistical purposes, and is useful in comparative literary studies, to establish the frequence of usage of certain words by certain authors. However it doesn't provide a way for most readers to view and access the major concepts in this particular corpus.

It is a completely different story when the amount of information is so massive that no human being will ever be able to analyze and process the trove of available data. Because of the availability of massive data, it is possible to create algorithms that can scan all of it, and be refined with multiple criteria to yield to results that are much more meaningful, and therefore much more usable than algorithms that were applied on a much smaller amount of data. Artificial intelligence techniques, including the fact that machines contain

"learning" abilities that take into account new data to dynamically reconfigure themselves are changing the landscape. The search algorithms used to browse the World Wide Web are based on such techniques, although they are also supplemented by human work to solve some of the issues that were bypassed by the algorithms.

The philosophical and epistemological preconceptions used to determine how knowledge "presents itself" still exist, but they are now hidden into algorithms that are not accessible directly by human users. As knowledge consumers, we are left with using the results of processing, with no ways to understand why and how the information we see is what it is. But it is not because this information is not visible that it doesn't exist. It only means that if we want to understand what we see, we need to dig deeper into how why this information has become what it is. In a book index, we rely on one individual's choices, and we may disagree with the choices that were made, and we may think we would have done it differently. Nevertheless, we give credit to the author of the index. On a massive scale, unless we can see the algorithms used to produce the information, we lose our ability to offer a different perspective. Worse, we will never know what we do not know. There may be some important information available that was missed by the search algorithm, but we won't know it. However, these are cases where we do know, if either we are experts in a domain or if we are in charge of managing a well-known data repository.

The Topic maps paradigm can be used as the conceptual basis on which forensic analyses can be conducted and provide methods for auditing content.

History of Topic Maps

Origins

Topic Maps originate from the community involved in promoting generic markup in the publishing industry. When computers started to be used to publish books and other printed materials, the idea of describing the semantic nature of a textual fragment rather than the way it should be formatted became the founding principle of generic markup. Furthermore, instead of providing a fixed set of predefined tags, the idea of letting users define themselves the elements they needed to use in a given application context was considered an important advantage. Naturally, communities of users could agree on a set of shared elements, so that each individual member or company in a domain would not have to reinvent the wheel. The precedent of handling a set of data (database) using a model with pre-defined fields and characteristics was extended to handing textual content (document) that complies with a pre-defined structure.

The Standard Generalized Markup Language (SGML) originated from the Generalized Markup Language (GML) used by IBM for its technical documents, and was published in 1986 by the International Organization for Standardization (ISO). It defines document structuring, and validation mechanisms to parse document instances against a document structure. Element names appear in a document surrounded by angle brackets. Although this notation can be changed, the angle bracket notation is the visual symbol for SGML documents.

As the World Wide Web was in its inception, hyperlinks became more important, with the emergence of the Hypertext Markup Language (HTML), a library of tags using angle brackets like SGML.

The work continued to extend the methodology emanating from standard markup to time-dependent information and hyperlinks. The HyTime standard (Hypermedia/Time-based

Structuring Language) was published in 1992 and was considered at the time as a potential successor to SGML.

The rapid rise of HTML and the associated browser technologies showed that, more than the strict compliance with generic markup standards, success came from the ability to create a gigantic hyperlink-based network that connected documents accessible through the Internet via a simple HyperText Transfer Protocol, known as "http". As the World Wide Web started to expand and change everything, one major feature was missing, i.e. the ability to qualify the nature of the relations. This requirement turned out to become the basis for what would a few years later emerge as the "Semantic Web". Two different approaches were taken that would eventually become the Resource Description Framework and Topic Maps.

The quest for interoperable indexes

In contrast with the web hyperlinks — unidirectional links pointing to a specific location — the publishing industry was looking for ways to describe the structure of richer connections. A consortium of Unix vendors was under pressure by its customers to harmonize the vocabulary used by the various tools providers, and initiated a study group looking for ways to design interoperable indexes in the technical documentation of their products. This group, called Davenport, was trying to match the requirements of the publishers with the new features available in the newly established standards.

A traditional index, generally located at the back of a book, contains a list of entries pointing to locations in the content of the book relevant to the concepts expressed by the entry terms. The Davenport Group was concerned by the applicability of its findings. A description of an index, as part of a document, should be able to describe entries as structured paragraphs, made of a term, and a locator (usually a page number). In addition, internal links inside the index, known as "see"

and "see also" could also be described by regular link structures.

Animated discussions started in the group, as it was joined by members of the standards community who were promoting the use of complex hyperlink models which were part of the HyTime standard. HyTime has a hyperlink module made of several models. One, called "independent link", was of particular interest, because, rather than being embedded in a document which is the origin of the link as in HTML, this link has multiple targets pointing to various locations. Therefore, it is created and maintained independently of its anchors. The idea that it was possible to manage links independently served as a starting point to define a topic as an object independent from its source, and more broadly to define an architecture in which information can be qualified as a superimposed layer. It opened the possibility to create an architecture that was independent from the sources. In this perspective, an index could be seen as a set of terms, created independently of what the text contains, that qualified the subject of text fragments (for example, pages), and this approach could be generalized to other situations where the semantic qualification of information needs a certain degree of freedom towards the sources to which it applies.

These two approaches on how to handle indexes had their own merits, but were technically very different. Acknowledging this situation, members of the Davenport decided to split the work into two groups. One group created a structured markup schema specifically adjusted for handling technical documentation, which included a whole model for documents, not only indexes, and they created what became the Docbook document type definition, which has been widely used since in the world of technical documentation. The other group worked on what would become known as topic maps.

Towards an ISO standard

The concept of independent linking was evaluated as a possible abstract model to capture the essence of indexing. After a few years of intense brainstorming, we arrive to the idea of a topic, which was an abstract concept, that could be represented by multiple names (even by no name at all), pointing to various locations in external resources that were relevant to it. Whether the string of characters that represented the term was present or not was irrelevant. A portion of a document could be about a given subject without mentioning the name under which the subject was qualified. This characteristic is also present in the book indexes. The indexer is creating a qualifying name for a concept that applies to a particular section of the book without requiring that this name would be present.

In 1995, after having tried several possible models, we came to realize that a topic could be a computerized instantiation of a subject, represented by a HyTime-based independent link construct. In addition, we wanted to be able to navigate between topics that could be related together, and created the notion of "associated topics" that collectively comprise a network graph.

As this model was applicable far beyond interoperable indexes for technical documentation, we proposed to the ISO technical group that was responsible for SGML and related standards to make it a generic standard.

ISO accepted the project, and we started the standardization process under the name "Topic Navigation Maps" (TNM). The name was later simplified and the standard became simply known as "Topic Maps". Its first edition was published in 2000.

As we were working on developing the Topic Maps standard as an application of HyTime, the markup community was working on simplifying SGML and removing many of the features that were used only rarely, and were particularly challenging to implementers. A new

standard, the eXtended Markup Language (XML) was published in 1996 by the World Wide Web consortium and became widely adopted. Many SGML applications were converted to XML applications. Soon after the publication of the first version of the Topic Maps standard, we worked on an XML version, which was published in 2001 under the name XML Topic Maps (XTM). We dropped the reference to HyTime, and used the XLink specification instead, which came out as part of the XML family of new standards. The way it was used was not significantly different from the simple hypertext links in HTML.

Topic Maps and the Semantic Web

Concurrently to the work done on Topic Maps, the World Wide Web consortium was working on a graph representation of hyperlinks between web resources. The Resource Description Framework (RDF) was published practically simultaneously as the first version of Topic Maps and become the core of the Semantic Web. The data represented with RDF was stored as "triples" (subject-predicate-object), with the distinct feature that the resources were represented by a unique Uniform Resource Identifier (URI). RDF is used to represent metadata in several reference initiatives, including the Dublin Core, which has been adopted by the community of librarians to represent online resources. Topic Maps and RDF have in common the fact that data is represented as a graph. But the focus of the RDF team was on ontology processes, and the ability to automate the creation of knowledge representations, while the Topic Maps team was focused on providing ways to create human-created index-like graphs.

While the standards were in their inception, there were contacts between the teams and a willingness to cooperate, but they were not frequent enough to have resulted in a common, or unified approach.

An analysis of guidelines for RDF and Topic Maps interoperability shows that there are several approaches being considered: semantic mapping, object mapping, and

hybrid. Subjects in Topic Maps are considered equivalent to RDF resources. Topic Maps allow for n-ary associations whereas RDF triples amount to binary relationships. But it is possible to envision a set of mapping rules to translate automatically one notation to the other (Presutti V., Garshol L.M., Vitali F., Pepper S., and Gessa, N. 2005).

At the same time, the Semantic Web project, based on RDF, and Linked Data, was also showing signs of decrease in interest. The agreement on a universally valid URL-based concepts is still very much alive, and serves as the foundation for libraries and open data exchange in general. But the learning curve has become a deterrent, as more and more technical layers were added.

In the industry, the willingness to share information was not a priority. Most of the times, the purpose is to share information internally. When only one topic map application is present, there is no need for an interchange format. As companies look at organizing their internal information repositories in the most efficient way, Since most of the topic map tools were designed to facilitate the XTM format, they fell out of grace, and many built-in topic maps applications were created.

The origin of topic maps was an attempt to bring the connected information, then called "hypermedia" to the world of publishing, by leveraging the traditional navigation systems, such as book indexes, library catalogs, thesaurii, dictionaries, etc. The analysis of these interconnections as a graph structure was the major breakthrough accomplished by the topic maps designers. Then came the Web and the Semantic Web, which aimed at relating online resources by their semantic properties, in order to facilitate access and automated processes. That was the origin of RDF, the Resource Description Framework, and the Dublin Core, a metadata set aimed at library cataloging, and later the Linked Data project. Every topic is represented by a unique Web address (its URI) and is accessible by many applications.

The "Linked Data" project, that is part of the Semantic Web project by the World Wide Web Consortium, aims at connecting data available on the Web through relationships. The Open Linked Data format is used by many libraries in an effort to create cross-library accesses to materials. It is based on the Dublin Core, which is a metadata format itself based on RDF that was created in 1995 in a meeting in Dublin, Ohio which was attended by members of the RDF team as well as the Topic Maps team.

The need for a more flexible organization of data has persisted, and even increased. The gradual rise of graph databases and the relatively new interest around knowledge graphs has revived the need for concepts similar to those of the Topic Maps model, bypassing the need for a common XML-based syntax. Other standards for exchanging graph-based information exist or have emerged. One of them is RDF [1], also on the decline, due to its verbosity and complexity. The standard that is looking most promising today to exchange graph data is called "Property Graphs" and it is based on key/value pairs for nodes and edges in the graph. Because of its generic nature, it can serve to transfer data from an environment to another. It doesn't contain the distinctions made in the Topic Maps about multiple names, nor does it consider the relation semantics as topics, but it can be used for interchange purposes between graph databases.

The Knowledge Graph, which appears on many Google search pages, comes from Freebase, a knowledge base created by a company called Metaweb, that was bought by Google in 2010, and was inspired by Topic Maps [2].

In 2010, Google coined the term "knowledge graph" to describe a knowledge base they acquired, Freebase, from a company called Metaweb, which has designed this base

[1] Resource Description Framework, a W3C Recommendation (1999)

[2] Private conversation with Veda Hlubinka-Cook, co-founder of Metaweb

using the concepts from topic maps, to semantically connect topics together.

The Topic Maps Standards Family

ISO/IEC 13250:2003 — SGML applications — Topic maps

The initial version of ISO/IEC 13250, Topic Maps, was published in 2000. The standard is defined as a set of architectural forms, a set of templates from which a document type definition can be written. Architectural forms are a feature of HyTime, the Hypermedia/Time-based Structuring Language (ISO/IEC 10744:1997) containing far-reaching hyperlinking and addressing facilities. HyTime has been used as a source for other standards or concepts, such as the Document Object Model used in HTML and XML.

A simplified version of the interchange syntax, written for XML and called XTM — for XML Topic Maps —, was published in 2001 by an organization called topicmaps.org. It was submitted to ISO for integration into the standard.

The second edition of the Topic Maps standard was published in 2003. It reproduces the content of the first edition, and adds the XML representation of the topic maps architecture, in a document type called "XML DTD for Web-oriented Topic Maps", which is identical to the previously mentioned XTM (XML Topic Maps) and was already started to gain significant momentum. The HyTime-based notation for Topic Maps, now called HyTM, is still present. As Topic Maps were applied and implemented, the HyTM model was practically never used.

The usual process of transforming an SGML document type definition into XML is straightforward and doesn't require any major change. However, in this case, HyTM was not a document type definition but a set of architectural forms, i.e. a set of templates that would serve as a basis to create document type definitions. Consequently, significant changes between the two versions are worth noting: XTM introduces a distinction between two kinds of

subjects: those which are addressable online and those which are not addressable. The links used in XTM are based on "simple xlink" instead of Hytime "varlink". These are links that are very similar to the universally used "href" attribute in HTML, and therefore facilitate implementation of topic maps on the Web. Finally, the notion of facets, which were designed to qualify properties for a subject, have been discarded, because they could be easily represented by using topic associations.

The new version of the standard facilitated the adoption of Topic Maps through the XTM notation.It is more concise, straightforward, easier to understand by implementers. In 2013, a new version of XTM, 2.0, was introduced (see below).

After the publication of the second edition of the Topic Maps standard, work continued. The standard was supplemented with new parts, that were intended to clarify how to interpret the existing standard (data model) and provided new languages to either express or process topic maps.

The definitions of the concepts were later reorganized into part 2 of the standard, under the name "Data Model" and the XTM representation of Topic Maps was later reorganized into part 3 of the standard, under the name "XML Syntax".

ISO/IEC 13250-2: 2006 — Topic maps — Part 2: Data model

The data model describes the properties of the constructs in the XML version of the Topic Maps standard. It specifies data types for the constructs present in the XTM document type definition, and documents them with UML diagrams. The merging operation procedure is described in detail. Core subject identifiers are defined. These subjects identifiers are given URLs and are available as "public subject identifiers".

ISO/IEC 13250-3: 2013 — Topic Maps — Part 3: XML Syntax

The XTM structure for topic maps was revised in 2013 and is now known as XTM 2.0. The main motivations for this revision were to remove features that were practically never used and added complexity, mainly without loss of functionalities. For example, the reference to XML Base was removed, and support for XLink was removed. [3] Other differences are changes in the names of the elements and attributes, removal of wrappers. Several new functionalities were added, such as the support for typed data, the ability to declare types for topic names, and a change in the way reification is indicated.

ISO/IEC 13250-4: 2009 — Topic Maps — Part 4: Canonicalization

The purpose of the Canonical XTM format, or CXTM, is to ensure that two instances of topic maps which conform to the Data Model are serialized identically byte-by-byte, in order to allow the creation of test suites to test topic maps products. Its features include the order in which the various elements should appear, and the requirement that empty lists are used, wherever applicable, even when they contain no item. In addition, the encoding of the documents should conform to Unicode Normalization Form C.

ISO/IEC 13250-5: 2015 — Topic Maps — Part 5: Reference Model

The Topic Maps Reference Model adds one level of abstraction above the data model. It considers the Topic Maps Data Model as just one instance of a more general model of a "subject map". A subject is any abstract item of conversation that is represented by a computer proxy, which consists in a declared set of key/value properties. A subject map is a finite set of proxies. The key/value properties are more general than the defined properties for a topic in the Topic Maps Data Model, which is one instance of what can be represented with the reference model.

However, the Reference Model defines two types of relationship, "is a" (instance of) and "sub" (subclass of)

[3] The differences are documented in http://www.garshol.priv.no/blog/85.html.

that can be used for inferencing. A number of primitive navigation operators are defined, that return a set of keys relative to a given proxy. Maps are defined by sets of constraints. Subject merging and clues for interpreting a subject are also defined.

ISO/IEC 13250-6: 2010 — Topic Maps — Part 6: Compact syntax

The Topic Maps Compact Syntax is a text-based notation that provides a light-weight alternative to the XML Topic Maps (XTM) notation. Its purpose is to be useful for manually authoring topic maps, provide human-readable examples in documents, and service as a syntactic basis for the Topic Maps Constraint Language.

ISO/IEC 19756: 2011. Topic Maps Constraint Language

As the topic maps defines a structure of data which is generic, there is a need to further constrain specific implementations. This opens the ability to validate a topic map according to specific semantic constraints valid within a given environment. For example, `works-for isa tmcl:association-type` expresses the fact that "works-for" is an association type. The Topic Maps Constraint Language contains features that enable the ability to follow an association, optionally with roles of a certain type, establish whether a topic is a subtype or an instance of another topic, and locate occurrence values. There are two types of validation rules, those that apply to an individual topic, and those which apply globally to the whole topic map.

Uncompleted projects

Query language

Several implementations of topic maps software contain notations for querying topic maps. Queries play an important role in what users expect from their topic maps-based system. A substantive amount of work went on to harmonize these languages into a standard. But the work has not been completed yet.

The proposed query language contains a notation to query topic maps and defines how processors should behave. The latest draft available at the time of publication of this article lists the navigation axes that are allowed in a topic map environment (types, players, roles, etc.) as well as traversal results (the result of a forward traversal), reification (which expresses the topic emerging from a construct such as an association between topics), and atomification (which describes a way to convert a complex construct in integer or strings). Furthermore, the Query Language defines comparison and ordering of tuples. It defines a notation for querying a topic map. For example:

```
select \$p / name where \$p isa person & lives-in-city (being : \$p
, city : \$\_)
```

finds all person names living in any city.

Graphical notation

An attempt was made to define a graphical notation, inspired by UML (Unified Modeling Language), for Topic Maps.

Where are Topic Maps being used?

Early Adoption of Topic Maps

Topic maps have been adopted in commercial applications as well as in the academia. In the initial phase, topic maps applications have been mostly developed in Europe and in North America, before taking off in Asia, most notably in Japan, Korea and in the recent years, in China.

The interest for topic maps took off in 2000, the year when the standard was initially published. It gave rise to an increasing number of works in the academic world, where it continued to grow until 2008. The number of publications on topic maps decreased since then, but the visibility of topic maps has decreased since then and in recent years a renewed interest for topic maps has

emerged, most notably in Asia. However, beyond the publicly available sources, several projects were developed, and still are, that refer to topic maps. Some of them are confidential, and others have simply not publicized enough to be visible.

One of the biggest knowledge bases using concepts inspired by the Topic Maps paradigm was Freebase, developed within a company called Metaweb, which was acquired by Google in 2010, and now appears on the search pages as the "Google Knowledge Graph". The US government made uses of Topic Maps, including by the Department of Energy, and the Internal Revenue Service. In Europe, city portals were created using Topic Maps. Norway has been an active center of development for various topic maps applications. Topic maps applications continue to be developed in East Asia, including Japan, South Korea and China.

In the academic world, many research projects have been conducted using Topic Maps, included those funded within the European Community supporting the Semantic Web initiative. Topic maps have been a recurring subject during markup conferences, especially between 2005 and 2010. A research laboratory dedicated to topic maps was created in Leipzig, Germany, and the Topic Maps Lab organized annual conferences dedicated to research and applications in the domain.

When topic maps were created, the web was still in its infancy, and the amount of information available in search was far smaller than it is now. Search technologies were for a large part based on string recognition, and information owners had an idea of the extent of the data they were dealing with. In that period, the need to organize information was considered a high priority. As the amount of information available started to grow, and the concept of "big data" started to emerge, the science of data analytics became prominent, and the use of techniques based on automated processes, labeled under the term "artificial intelligence" took the lead on more traditional information management techniques.

Refinements in the algorithms, machine learning features, and improvements in the quality of search results have provided an alternative to a fully human-driven classification of information.

Graph-based Ontologies and Taxonomies

Many applications are not strictly based on topic maps, but address a similar goal, which is to organize information into a network of topics.

Wikidata, which is the knowledge base that feeds Wikipedia, has a structure which resembles a topic map, although it doesn't directly reference it as such.

Many applications have been created by direct reference to Topic Maps at Columbia University (New York), New York University, the American Geophysical Union, RILM (Répertoire international de littérature musicale), the US Department of Energy, the US Internal Revenue Service, the European Community, the Norwegian government, publishing companies in Netherlands and the United States, by industry manufacturers in Germany, and in many other countries.

From Search to Generative Artificial Intelligence

As most human activities are now available online, getting information is critical for most human activities. Google has succeeded in dominating the search industry by accumulating knowledge available and making it available through search engines. The way information is presented by a Google search is absolutely essential to the success of many companies.

The accumulation of data available has also proven to be a huge opportunity for designing products able to quickly collect and analyze the trove of data. The

ability to access this information has been a key factor to the development of large language model, and has made possible the emergence of generative artificial intelligence. Instead of simply searching for information available using keywords, like in a traditional search query, users can now formulate long questions, asking computers to analyze information and return a full written report on a question that is of interest to them. There is enough information available to make it possible to create answers that can replace hours or days of compiling information and writing reports. The products are not entirely reliable, and sometimes appear right while being somewhat, sometimes entirely, wrong. Educated users can use these reports as drafts or starting points. People who are not entirely fluent in one language can write texts that appear to have been written by a native of that language.

Taking Information Into Accounts

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]

perspector.

another

itself

serve

an

in

as

Tables Everywhere

Preliminary

Table is a word in English that means a lot of different things. The verb "table" means the exact contrary whether you are American or any other English speaker. In America, when someone wants to "table" a discussion in a meeting, they mean they want to postpone it, sometimes undefinitely. In all other places, it means starting right away to discuss the subject in question. This has caused numerous, sometimes comic, misunderstandings in international meetings.

It is however not the only misunderstanding that comes with that word. In the world of data, a table is way to display information in rows and columns. But that is just the appearance. A table may be a sheet in a spreadsheet, a view of records in a database. It can be filled with numbers, or non-numeric data, such as text, or images; depending on the context, a table may be either all what there is or the tip of an iceberg.

The goal of this book is to show that using tables to represent information has benefits but also weaknesses. The benefits are the powerful representational value of tabular data, based on centuries of accumulated knowledge and tradition. The weaknesses are centered on the fact once tables are used, it's much more difficult to think outside of the box.

I intend to show the power of tabular representation and tables and will distinguish between tables within text, within spreadsheets and within databases. I will uncover the hidden

relationships between data items in tables, that are revealed by understanding how databases are structured: flat databases, relational databases, object databases, and graph databases have different models to express the relationships between data points.

When tabular views become predominant, it creates an incentive to put everything within a box. This practice, which is widely spread, makes it practically impossible to think outside of the box. It is even worse than that: many people are incapable, within a professional context, to even envision that it is possible to think outside of a box. This pernicious side effect translates into missed opportunities, loss of productivity. It appears counter-intuitive, as many "modern" applications are extensively using a box model, and it naturally seems like the natural thing to do.

Then, I will show that connectivity is key, and that the human brain is capable of things that no computer based application can do: freely associate things with each other, with no other constraint than the imagination and the creativity. This capability is only partially represented in a table. What is interesting is what is missing.

1. Tables
    1. What is under the hood with tabular presentation of data.
2. Spreadsheets
    1. What they were intended for
    2. How they are used
3. Databases
    1. Relational, Object and Graph
4. Metadata and Schemas.
    1. Classes and Instances
    2. Metadata is data

5. Human Brain
    1. The Power of Connections
6. Brain Damage

    Thinking in a box.

History of tables

History of spreadsheets

History of databases

History of Structured documents: Tables as the Achille's heels in SGML/XML

Tabular Model: Structure and Formats, Thomas Bressoud & David White, https://link.springer.com/chapter/10.1007/978-3-030-54371-6_6

- Where presentation and structure are mixed. The Achille's heal of XML. Separating structure and presentation works with everything, except with tables.

- Dirk Schlimm, Tables as Powerful Representational TOols. International conference on Theory and Applications of Diagrams, Springer

- Pandas

Numbers and beyond

The name "computer" implies that these devices are originally calculators that deal primarily with numbers. The way computers are designed is based on switching on or off bits of information, that can be 0s or 1s. Most of the information we are dealing with can be digitized, in

other words, transformed into numbers. Thanks to the increasing processing power developed along the years, digitized information is not just available for textual information, but also serves to encode images, video and audio. In that sense, the computer has lost its original unmediated connection with numbers. But it still heavily relies on numbers.

Before the 1980s, huge computers were used for calculations and for storing and crunching data. Computers were able to yield scientific results, or statistical results. When IBM entered the market introducing the notion of "personal computer", the computer started to hit the economy in a big manner. Companies could create and manage their own data using computerized databases, they could improve their accounting and financial capacities with spreadsheets, and they could improve the productivity of their employees by shifting from typewriters to word processors.

A decade later, the Web emerged as a way for anybody to connect to the Internet. The Web rapidly evolved into a commercial platform, with the emergence of e-commerce giants such as Amazon, a knowledge distribution platform with search giants such as Google, and a few years later with the emergence of social networking platforms, such as Facebook, Twitter, Instagram and the like.

A decade later, software shifted to being provisioned on "the cloud", a dematerialized platform available through Internet connections, and independent of any specific operating system.

The computers came back to what they were at the beginning, i.e. simple terminals connected to a network. This time, the network span was worldwide instead of being owned and maintained by giant companies. Practically all software available today is provided through the cloud. The main advantage for individual users is the fact that they can use more than one device to access their data. If one of the devices breaks and

needs to be replaced, it is an easy step, as any individual device contains very few data which is not at the same time on the cloud.

Companies have replaced their internal networks, called Intranets, by networks that use the same technologies as the Internet, and are therefore much widely accessible to any kind of companies, regardless of its size.

Everybody who works uses computers. Farmers use computers, truck drivers use computers, office workers use computers, sciendigital economy in which we live now relies on numtists use computers, teachers use computers, executives use computers. There is not one sector of activity that could function today without computers. Smartphones are nothing else than miniaturized computers, that have, in addition to all other features, the ability to give and receive phone calls. Computers therefore are used not just for work, but are an essential and indispensable piece of equipment that is used by every one to manage their daily life, for the better or the worst.

The purpose of this book is to try to assess what changes have occurred by the omnipresence of computers in our lives. A lot has been written on the impact of computers on society, freedom, privacy, healthcare, productivity,

The Power of Tabular Representation

We all use tables, since a long time. Tables are a condensed a powerful way to see information in a way that is very condensed, powerful. If an image is worth a thousand words, a table provides value that is critical for businesses.

Tables are used to collect numbers, and display operations on those numbers.

Tables of Numbers

Used for accounting, spreading tables (spreadsheets)

Dynamic Tables

Using calculated values instead of fixed values Operations. Formulas. The most popular application of personal computers.

Tables of Data

Lists of items sharing a common characteristic.

When items have multiple characteristics, each of the property can be represented as a column in a table. Each row in the table contains all the characteristics of a given item.

Creating/Editing Tables within a word processor.

The user interface often allows to select an empty table with a number of columns and rows. It is possible to add, or delete, rows and columns. Any value can be entered in a cell.

Creating /Editing Tables with a spreadsheet

Spreadsheet software have been designed to facilitate numeric data entry, and to define operations that can be performed on these data.

These software products open with an empty table. There are no constraints on what kind of data should be entered. The data doesn't have to be numbers. It can be plain text data, as well. As the table is already there, the spreadsheet software are easier to use than the word processors for creating and editing tables. Furthermore, each column can be sorted or filtered according to user-defined criteria. Office suite software enables tables created with spreadsheet software to be pasted inside a text created with a word processor.

Databases

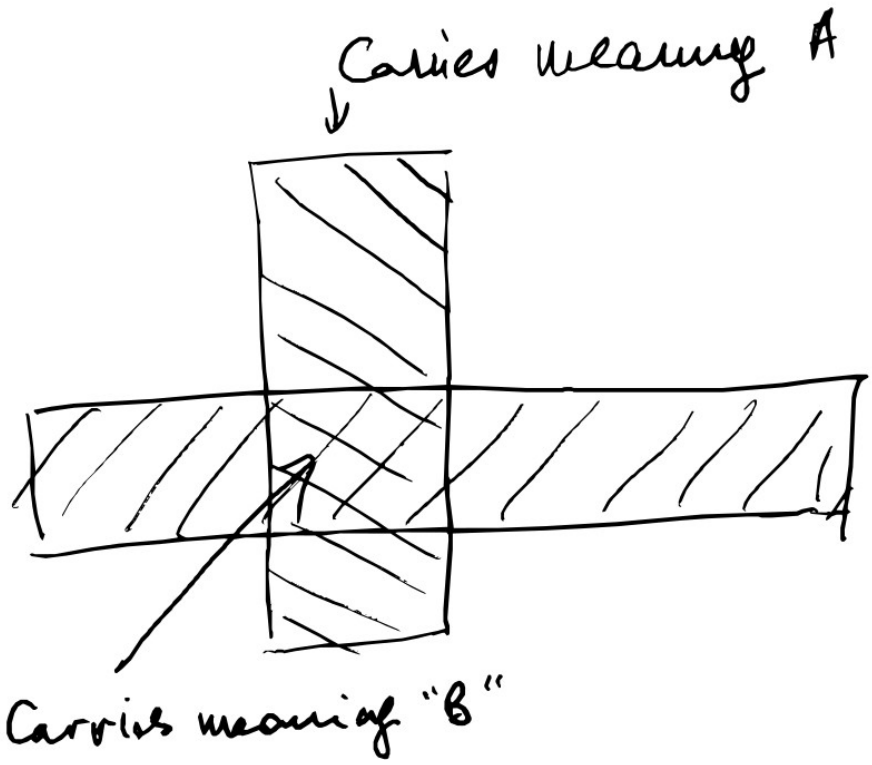
Tables can also be the representation of data within databases.

Textual Tables

Contact Table2

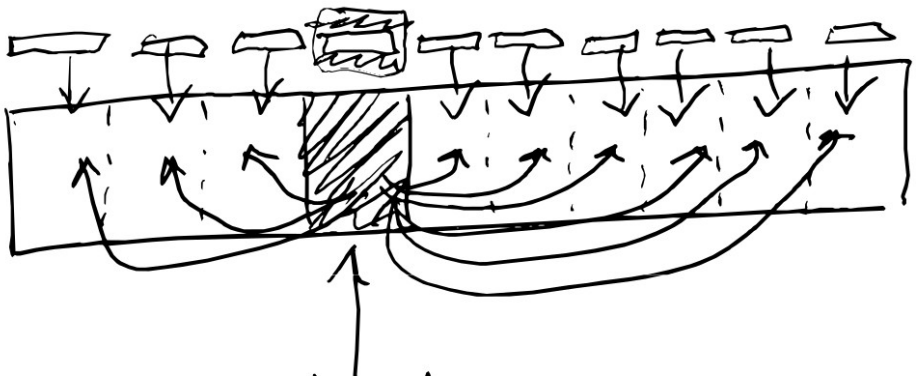
John	Doe	25 65th Street	New York	NY
Mary	Poppins	525 Broadway	New York	NY
…	…	…	…	…

Hytime Scheduling Module



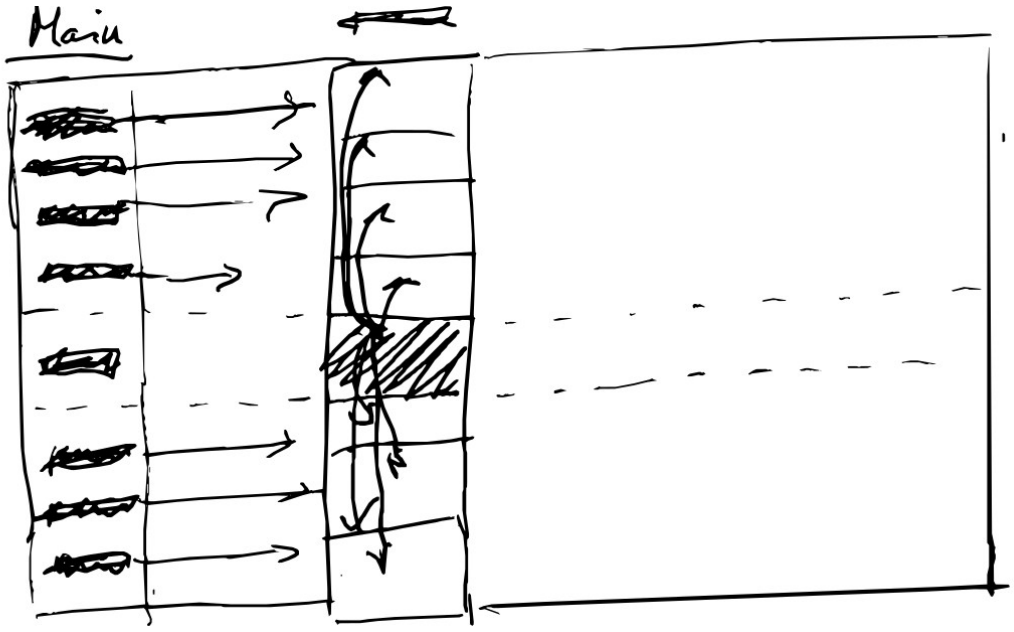
Intersection of meaning A and meaning B.

Horizontal Links



Content of cell: Connect to adjacent cells using the column headers to define the semantic of the relationship.

Vertical links



Vertical Links

Vertical Links

Tables are a very powerful way of representing information. They allow items to be listed and structured by properties. Each property is defined by a column header and each item has a value for that propery in the cell located in that column.

An equivalent notation in a key, value format for a table would be:

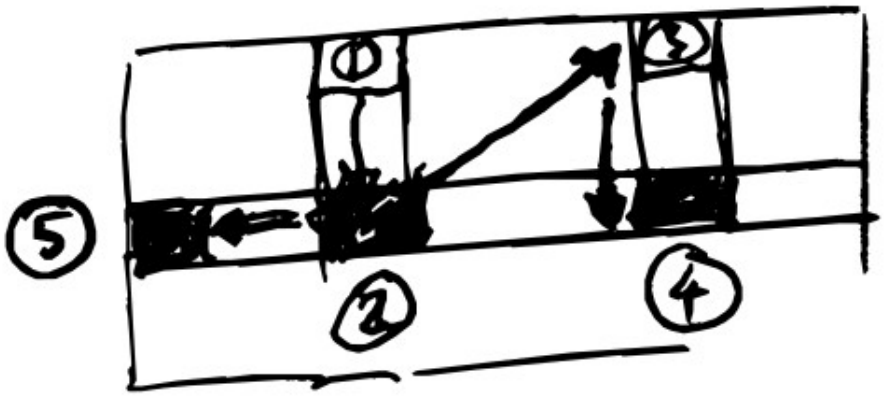
```
[
  ((prop1, value11), (prop2, value12), ...),
  ((prop1, value21), (prop2, value22), ...),
  ((prop1, value31), (prop2, value32), ...),
]
```

If a table represents a database, the column headers express the schema, and the cells contain the data. Each column has a defined data type, and cells must contain data that complies with that data type.

The ability to represent data in a table depends on the fact that data must be clearly structured.

Museums

Names	Neighborhood	Borough	Type	Focus	Summary
Aperture Foundation	Lower Manhattan	Manhattan	Media	Photography, Film, New media	Exhibitions dedicated to contemporary and classic photography
Brooklyn Museum	Crown Heights	Brooklyn	Art	Encyclopedic	Collections include Egyptian, Classic and Ancient Near Eastern art, Feminist, European and the art of the Pacific islands.



In the borough (1) of Manhattan (2), in Summary (3), there are exhibitions dedicated to contemporay and classic photography (4). They are at the Aperture Foundation (4).

Reading a line without the column headers:

|Aperture Foundation|Lower Manhattan|Manhattan|Media| Photography, Film, New media|Exhibitions dedicated to contemporary and classic photography|

The table provides significant information, with implicit knowledge that most humans would be able to find out. We know that a foundation is sometimes a place that is open to visitors. Lower Manhattan is unambiguously an area of Manhattan. Manhattan itself is usually understood as a borough of New York City. Etc., Without accessing the column headers, we can more or less figure out all the information available for the Aperture Foundation.

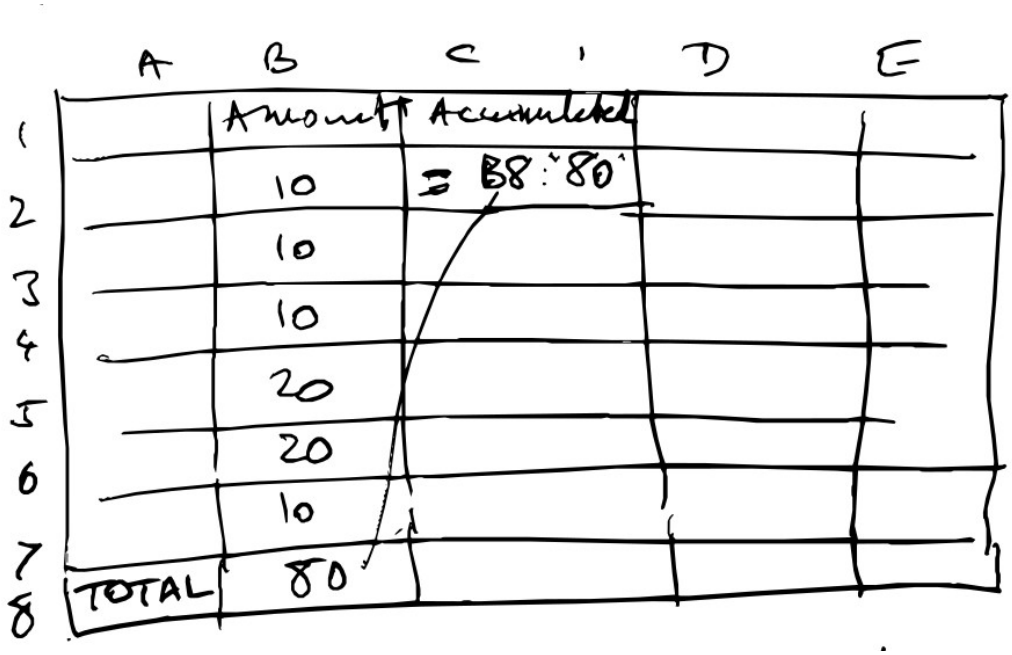
Notes for Tables

- Tabular data. Mostly numeric. It's a concept used in statistics. The "opposite" of tabular data is "visual data". [statology] https://www.statology.org/tabular-data/
- Tabular presentation of data [byjus.com] https://byjus.com/commerce/tabular-presentation-of-data/. Most

significant benefit: it coordinates data for additional statistical treatment and decision making (?)

- Tabular presentation of data. Geeks for Geeks. https://www.geeksforgeeks.org/tabular-presentation-of-data-meaning-objectives-features-and-merits/. Meaning, objectives, features and merits.

Spreadsheets



	A	B	C	D	E
1		Amounts	Accumulated		
2		10	=B8		
3		10			
4		10			
5		20			
6		20			
7		10			
8	Total	=SUM(B2:B7)			

Used for Arithmetic operations, formulas, simulations

Designed for accounting, planning, simulation scenarios, etc.

Abused for storing just textual data

	A	B	C	D	E
1	John	Doe	25 W 65th Street	New York	NY
2	Mary	Poppins	525 Broadway	New York	NY

Why?

- Because it's easy to use. Learning curve is close to zero.
- Enables filtering and sorting by columns.

Problems:

- Duplication of data. Example : New York - NY
- Disconnected. Spreadsheets are not related with each other, even if they contain the same data.
- If the same cell content is duplicated in one spreadsheet, it's not connected.
- Hard to use if the number of columns is wider than the monitor's width. Need to constantly scroll to the right or to the left to enter or visualize data.
- Hard to print, for the same reason.
- Can be enhanced with custom plug-ins, which can be built by experts, often external consultants, for a price, plus: It has been updated in case anything changes in the structure of the table.

Databases

Databases are made of tables. They provide a much better control of the content of the data. Data integrity is checked, and databases allow data to be queried, reorganized and mangaged in ways that are more powerful than spreadsheets.

However, in some cases, data stored in databases may look very much as if it were in a spreadsheet or just a table.

Relational databases allow for tables to be connected (joined) to each other using common values belonging to predefined fields. The value used in an external table is sometimes called a foreign key.

Object databases are very similar to relational databases, but their user interface is different. Each item is an object with predefined properties. An object is a set of key-value pairs, where the key is the name of the property and value is the specific local value for that property. For example `(city, New York)` is a key-value pair.

The object databases are, internally, similar to relational databases. But, for a programmer, the experience is different. Instead of having to design a set of tables joined by a common field, the key-value pair mechanism automatically produces joined tables, as many as necessary.

Graph databases push the paradigm one step further. Components are nodes in a graph, and those nodes can be interconnected to other nodes. Very frequently, nodes are defined as objects with properties, making the graph databases look and operate in a way which is very similar to object databases. Here, links become first class objects. Compared to relational databases, where joined tables appeared as an afterthought, graph databases put the emphasis on the connections, and describe the nodes independently. Again, the way graph databases are used

make them look different to a programmer, but is still based as the same underlying technology.

Tabular Databases

Databases are a way to store data in a structured way, and make that data available for all kinds of operations. Data can be queried according to multiple criteria. Lists, and reports can be produced. The most frequently used representation of data stored in a database is through tables. It is so frequent that "table" has practically become a synonym for "database". Columns represent fields, and rows represent records. The column hea ander is the field name. Each field is declared to comply with a given data type: text, number, date, etc. The list of the fields and their data types is called a database schema.

The fact that the data is declared to comply with a predefined structure imposes constraints on what kind of data can be entered within a particular column. A user will not be able to enter a data within a cell if the data type does not correspond to the one that is expected. For example, trying to insert alphabetic letters within a text field will create an error.

Other Databases: Relational, Object, Graph

Just having tables in databases is not sufficient. A table can be related to another one, in multiple ways. In a relational database system, two tables can be joined if they have a common column. In an object database system, a record is an object with properties, expressed by key-value pairs.

- Viewing tables: rows and columns
- Tweaking tables: spreadsheets
- Organizing tables: databases
- Connecting tables: relational databases, object databases, graph databases.

Tabular views are a very old representation of data. Originally conceived to align numbers, often with a sum on the last line - the bottom line - tables have been used on stones, then on paper.

As needs grow, the single sheets of paper were not sufficient. As columns of numbers kept augmenting, sheets were spreading. There was a need for longer sheets, and sheets that were also expanding horizontally. Sheets spread into large foldable paper, extending horizontally and vertically. They were called spreadsheets.

In parallel, as data started to accumulate, the need arose to classify them into categories, which was the origin of taxonomies, such as the taxonomy of living species. Mammals, invertebrates, and all kinds of animals were described within the categories to which they belonged.

When computers started to be used, data types were invented. An item could be constrained to belong to a predefined type: a string of characters, a number, a date, etc. Using these types was adding a constraint, but it helped checking if the data was corrected, and also helped doing some calculationss, such as the number of days between two dates, etc.

When computers started to be introduced, they "excelled" at crunching numbers, and, were a welcome tool to handle databases producing tables. When "personal" computers were introduced, many companies bought tons of them first to replace the paper-based spreadsheets, which were very cumbersome.

When computers started to get networked, more powerful were introduced to link data between individual tables and spreadsheets. Database technologgie evolved into relational databases, with multiple user interfaces: purely relational, object, graph, and schemaless.

Tables are a powerful, condensed, visual representation of data. A table can be read horizontally or vertically. Column headers define the kind of information that we

expect to find in the corresponding column. Row headers, which do not always exist, provide the semantics shared by every cell in a given row.

There are several tools available to create tables. Most word processors have functions to allow users to create tables, add or delete columns or rows, and format the tables.

The tool that is used most often to create tables is called spreadsheets. The tables are ready to be used. The users can fill the data immediately. It is possible to create formulas to perform arithmetic operations on the data. Because these tools are easy to use, they can be, and are, used massively, to enter any kind of data, numeric and non-numeric. There is no constraint on the type of data that can be entered in any given cell of the table.

Connections

Metadata

What databases, tables and spreadsheets have in common is that a header defines the type of information. Every item belonging to a type, or a class, is an instance of that type. This classification schema corresponds to taxonomies. A human is a mammal, a mammal is an animal, an animal is a living creature. Etc.

Designing the appropriate database schema takes time and effort. Especially when several tables need to be joined together. The process is very similar when designing the set of properties for each object. Once the design is finalized, applications can be built, with multiple queries returning the results that end users expect.

This process is quite efficient, but somewhat rigid. Once decisions have been made, it is difficult to change something, as many add-ons that have been built depend on the existing structure of data.

Designing a spreadsheet follows a similar process, but much lighter. Columns can be added or removed at any time, especially if the table has no interdepencies with other tables. Naturally, its utility is much more limited.

Best and Worst Practices

Computers: are they made to enslave us or to free us?

Should we adjust to what computers are able to do, or should be ask them to do what we need?

The big misunderstanding

Mess is part of our life.

This book is focused on the impact of computers on our intellectual activity.

What has changed, if anything, on our way of thinking? We can access instantaneously all of the available knowledge, wisdom, art, music, etc. ever produced. Companies can access to lots of data that were not available to them before. In which ways has this change our way of seeing things, of working, of solving problems, finding solutions, etc.

The Uber Apps

In other words, Apps Uber Alles. The domination of apps over our activities, our well-being, our communications with even our closest relatives, and over our daily work activities has created a world which smells different. The apps are fighting for the "attention market", they need to get us reacting to them in real time, all the time. We are living in a permanent flow of notifications. Before, we were living under permanent threats of interruptions with phone calls. When the phone rang, we had to interrupt what we were doing, and focus our attention to the conversation with the caller. Some people were exclusively working answering phone calls. Phone calls were both a blessing and a curve.

Isolation happened when we didn't receive any phone call during a whole day, or during evening hours.

Besides, the television programs were constantly interrupted by commercial break trying to get our attention on something that clearly disrupted our attention while listening to a program. The online economy has made this distraction-based economy more pernicious and more subtle. Instead of being interrupted blatantly, we are submitted to a constant flux of notifications, which are either work-related (please, do something right now), either giving us information we more or less care about from people we care about, or by proposing us products that the apps "think" we might be interested in, as they listen to our conversations and read our messages, and therefore know about us much more than at the time of TV adds which were the same for everyone.

We are kept in a constant alarm mode, by receiving alerts about what is going on. Instead of us actively looking for information, for example by purchasing a newspaper, we are notified that something has happened that we should know about. The accumulation is at the same time exhausting and addictive. It is comparable, albeit worse, than waiting while dreading for phone calls. We feel both solitary and isolated when nothing happens, and overwhelmed when too many things happen in a short amount of time. We don't have time to breathe any more.

What does this do to our capacity of thinking? Can we think when being submitted to this constant bombardment of information? How do we adjust to the new information we are receiving all the time? Are we able to adjust our thinking to integrate it? Or do we decide to pause thinking and instead accommodate whatever is coming to us? Because we get information tailored to what we are supposed to think or believe, we receive only information that

we want to hear, and neglect or ignore the ones that does not please us. It can have the adverse effect of reinforcing our prejudices and our system by preventing us to absorb new things. As it has become increasingly difficult to sort out valid information from gossip, propagated conspirational theories and rumors, attempts to manipulate public opinion, etc., it has become harder to rely on information we're receiving to forge an opinion.

With that context in mind, let's refocus on the main point of this book, which is the specific constraints imposed by computerized activities. One of the main observations is that many activities are being described using spreadsheets or more generally tables. Tables enable us to immediately grab a condensed view of information... (To be continued)

The Tabular Illusion

- Viewing tables: rows and columns
- Tweaking tables: spreadsheets
- Organizing tables: databases
- Connecting tables: relational databases, object databases, graph databases.

Tabular views are a very old representation of data. Originally conceived to align numbers, often with a sum on the last line - the bottom line - tables have been used on stones, then on paper.

As needs grow, the single sheets of paper were not sufficient. As columns of numbers kept augmenting, sheets were spreading. There was a need for longer sheets, and sheets that were also expanding horizontally. Sheets spread into large foldable paper, extending horizontally and vertically. They were called spreadsheets.

In parallel, as data started to accumulate, the need arose to classify them into categories, which was the origin of taxonomies, such as the taxonomy of living species. Mammals, invertebrates, and all kinds of animals were described within the categories to which they belonged.

When computers started to be used, data types were invented. An item could be constrained to belong to a predefined type: a string of characters, a number, a date, etc. Using these types was adding a constraint, but it helped checking if the data was corrected, and also helped doing some calculationss, such as the number of days between two dates, etc.

When computers started to be introduced, they "excelled" at crunching numbers, and, were a welcome tool to handle databases producing tables. When "personal" computers were introduced, many companies bought tons of them first to replace the paper-based spreadsheets, which were very cumbersome.

When computers started to get networked, more powerful were introduced to link data between individual tables and spreadsheets. Database technologgie evolved into relational databases, with multiple user interfaces: purely relational, object, graph, and schemaless.

Tables are a powerful, condensed, visual representation of data. A table can be read horizontally or vertically. Column headers define the kind of information that we expect to find in the corresponding column. Row headers, which do not always exist, provide the semantics shared by every cell in a given row.

There are several tools available to create tables. Most word processors have functions to allow users to create tables, add or delete columns or rows, and format the tables.

The tool that is used most often to create tables is called spreadsheets. The tables are ready to be used. The users can fill the data immediately. It is possible to create formulas to perform arithmetic operations on the

data. Because these tools are easy to use, they can be, and are, used massively, to enter any kind of data, numeric and non-numeric. There is no constraint on the type of data that can be entered in any given cell of the table.

Brain damages.

What digital economy is doing to us

1. Attention spans

- TV with commercial breaks
- Notifications
- Pop ups
- Apps ## 2. Long-term consequences
- Personal: stress level, fulfillment and happiness
- Professional:
    - Reacting to notifications is like working on a production line. You can still go to the bathroom, but you need to carry your phone with you.
    - If it's an app able to show metrics and do data analysis, whatever the data is, you are covered.
    - Working in constant surveillance
    - Working under threat. Clicking on the wrong link can have bad consequences. ## 3. What has been lost
- Deep thinking
- Innovation. Thinking out of the box.
- Creativity
- Ability to take distance, refocus on the essential ## 4. What do tabular thinking have to do with this?
- Put everything in a box
- Disconnect everything ## 5. Management methods
- Tabular thinking implies micro-management
- Tabular thinking creates the illusion that work is being done. It may accomplish nothing ## 6. Available products ### Dangerous products
- SharePoint: user is not in control
- SalesForce: employee surveillance
- Jira: micromanagement at its worst

- Monday: hybrid between spreadsheet and database. Murky.
- Excel: too loose, applications enforce rigidity ### States to be classified
- Tableau
- AirTable ### Other paradigms
- Notion ?

Many people spend most of their professional life filling data into table cells. People who work in finance use tables of numbers to calculate results, and use these results to take decisions. The accumulation of data has made this approach relevant to other sectors as well. Numbers have become an essential data representation for people working in sales and marketing, for the advertisement industry, etc. That usage of tables is outside the scope for this book. I am going to focus on people using tables to represent all kinds of non-numeric data.

Project planning, workflow management, are made using tables. This has grown to a point where, in these activities, people do not even think it is possible to perform any task other than putting it within a table. Management loves tables, because they are a condensed way to present the activities.

The Missing Link: the design of the Networker

Graph databases combined with Generative Artificial Intelligence is the new holy graal of the Information Industry. But it fails to address the main expectation that people have when working on graphs, i.e. the flexibility to link anything with anything else.