

# What's Wrong with Knowledge Management? And the Emergence of Ontology

Mark Burgess  
*CTO CFEngine  
Norway*

## 1. Introduction

Knowledge Management (KM) is undoubtedly the challenge of this decade, and it is destined to shape the way we go about a wide range of fields of human activity for decades to come. Yet, while technologies claiming to enable KM abound, there is little sign that any wide reaching principles have been clearly understood or articulated, or that current research approaches have any positive benefit beyond brute force searching for answers, see Hicks et al. (2006). Nor is there any realistic alternative to the two major approaches to information organization: random search and retrieval (indexing), versus catalogue classification (the directory or table of contents).

In this essay, I should like to discuss some of the principles of knowledge organization, as I see them, from a perspective that has yielded some success in the related area of configuration or pattern management, see Bergstra & Burgess (1994-now); Burgess (2005). In order to keep things concise and focused, I will concentrate on spelling out a few specific criticisms of current approaches to KM, and then go on to propose adjustments to these approaches that could lead to large improvements in the current state of the art. Finally I'll set some challenges for future investigation.

## 2. Background

Knowledge management, or knowledge engineering (KE) today conjours up associations like: database, catalogue, ontology, semantic reasoning, etc. Yet, before information technology (IT) arrived on the scene, thousands of years of human development came up with quite different answers to the problem of passing on knowledge:

- Exploring and Discovering.
- Storytelling.
- Teaching and apprenticeship.

Only with the arrival written word, see Wolf (2007), did libraries begin to consider ways of managing large amounts of information, to accumulate knowledge and set about the task of organizing it. Today, however, those who work with knowledge (knowledge engineers, if we may call them that) feel that there is no mileage in these simple matters and are now only concerned with stockpiling and organizing information, then retrieving it, assuming that its simple existence as some form of documentation is enough to guarantee its usefulness.

Concepts like ‘taxonomy’ and ‘ontology’ abound, both of which we must return to in this essay, and methods are borrowed from computer logic to try to reason about categories of information, see et al. (2003); Strassner (2007).

The state of the art in knowledge management today, may be summarized by these things, in my view:

- Databases.
- The Web, and its search engines.
- Wikis for collaborative writing.
- Ontology or semantic modelling as a dream of something better.

The bulk of information today is available on ‘the Web’. Although the web is only a publishing mechanism, it has played a genuine role in encouraging the documentation and dissemination of knowledge, by making it easy (cheap) for people to participate in a shared process. Wikis have helped to make this happen (with flagship examples like Wikipedia), but the ways in which Wikis are used, in a wider context, suggests that they have a less than effective track-record as long-term repositories of knowledge (see below for some speculations on this). Today, we aspire to greater sophistication. Semantic technologies attempt to go beyond the pure text-search approach to using the web, by attaching actual intended meaning to words. Tools like RDF, see W3C (n.d.), Topic Maps, see Moore (2001); Pepper (2009), OIL, OWL, Protegé, see Strassner (2007), etc., attempt to use intended meaning to provide better hints and suggestions to knowledge-seekers by wrapping information in ‘meta-data’ – or information for computers to use in cataloguing and reasoning to capture domain expertise.

Semantic webs or networks (often called ontologies today) begin with the desire to classify topics ‘meaningfully’. One reason is to be able to disambiguate different patterns of usage as we talk about things. An ontology is a model of someone’s particular ‘world view’. Technically, an ontology is defined as ‘a set of specialized concepts within a domain’ (from the Greek logos (talk about) and ontos (that which exists)). It is the term used for describing domain knowledge.

Judging both from my own personal experience and from existing research, ontologies are notoriously hard to create for a number of reasons, see Dicheva & Dichev (2005); Moench et al. (2003). They typically represent domain expertise, they need to be populated with knowledge by domain experts; however, the technologies for doing so are not user-friendly and so these trained experts need other experts (trained knowledge engineers) to make the models on their behalf – experts, requiring more experts. All this places a high cost on creating useful systems, and it has essentially killed most efforts to use semantic technologies, by trying to be too clever. This is one of the issues I would like to address here. The other main problem with technologies today is that they have little capacity to teach users about new things they didn’t already know of. So we would like to find a way to offer new insights to users, as they interact with a knowledge base.

### **3. The shortcomings of databases and search**

Stockpiling information does not increase knowledge in any real sense, but this is what we do in libraries and databases. According to the IT Infrastructure Library’s knowledge management guide, see of Government Commerce (2000), the knowledge ladder goes through four phases: DIKW, or from mere Data (numbers, texts, assertions, etc), to Information that has some context around it, which in turn becomes knowledge when we are able to apply it and use it as a tool. Then finally there is Wisdom, down the line as something we aspire to.

The ability to search within such a repository can throw up all kinds of possibilities, but usually either too few or too many to make real sense of. With the advent of Google's PageRank, we have come to trust statistical 'mass-voting' as a guide to relevance. Popular voting a normative process in which common habits converge and encourage the spread of the habit to others. What is interesting about the enormous success of Google is that it does not try to be too sophisticated. It basically offers up some sorted data to users, who must then use their brains and exploring skills to rummage through the results. I believe we should learn something from this.

Databases store data row upon row, like a warehouse. They form the bread and butter technologies for storing data in a retrievable way, by associating names and addresses with information, so that a catalogue of entries can be made and searched by users. The limitation of most databases is that they have fixed tabular structures or schemas into which all information must fit. They are just 'data' - void of interpretation, and inaccessible to non-specialists. This makes sense from the viewpoint of machine retrieval, but it puts a burden of creativity on the user to standardize something that it perhaps only loosely understood. As a result, tidy database models tend to bloat into garbage dumps for all kinds of semi-intended annotation.

I shall argue that this is part of a common flaw in approaches to Knowledge Management: the first flaw of KM is that the user is expected to subsidize the technology in order to make it work. The tools of knowledge management, far from helping, often put up barriers and costs for the user. It is interesting to see the great leaps and bounds that have been made in user interface design, and compare this to the poverty of progress in knowledge management.

A second problem with database search is the attempt to use formal logic to whittle away at a large number of possible matches, see Strassner (2007). Even today's experimental derivative technologies for semantic search try to use assumed facts to reason in first order logic, hoping to find enlightenment. The problem with logic is that it needs strictly defined, inflexible categories to work with – and this is the second flaw of knowledge modelling. As one whittles away results, the result tends to be either utterly obvious, or empty due to over constraint.

#### 4. Knowledge representations

Knowledge is what you get when you combine context and experience with information so that its consumers can form their own mental model, believe it and apply it for themselves.

The way we represent this mixture can vary enormously. Examples range from the obvious to the subtle, and as scientists we should also have the humility to recognize art as a form of semantic commentary with a definite role to play in communicating things with cultural context.

**Definition 1** (Knowledge Representation). *A piece of work that attempts to communicate observations about something, e.g. experiences, opinions, or some other form of human understanding.*

Examples of Knowledge Representation include the obvious books, articles, stories. We also have keywords, titles, tables of contents, index entries, pointers, references and relationships to other works. Paintings, songs, music, and all other forms of communicable works may be thought of as different representations for knowledge.

How do we know when something is knowledge? This brings us to the third flaw with Knowledge Management. It is generally assumed in the field of Knowledge Management that facts and knowledge are authoritative, i.e. that which is knowledge is defined for the

masses by schools and universities or experts, or other figure-heads. This however is patently false. Accepting something as knowledge is an entirely voluntary choice that every one of us makes at our discretion. No one can force us to believe or accept what is represented. We must therefore adopt a model of knowledge based on voluntary adoption if we hope to understand something significant.

**Definition 2** (The three flaws of knowledge engineering). *Current approaches to knowledge make three errors:*

1. *Users are expected to work too hard to interact with knowledge.*
2. *Knowledge is treated as a logical framework.*
3. *Knowledge categories are defined authoritatively, but users only accept them voluntarily, if the context of their own experience.*

#### 4.1 Topic maps and RDF

One of the representations of knowledge index that has been developed over the years is the Topic Map, see Pepper (2009). Topic Maps are a form of subject index with detailed annotations that explain the relevance of associations. Topic maps have been made into a standard for the representation and interchange of knowledge. The ISO standard is formally known as ISO/IEC 13250:2003.

Topic Maps have several competitors in this space, the Resource Description Framework (RDF) being the best known, see W3C (n.d.). What makes Topic Maps interesting is that they were designed for human consumption, where the emphasis for RDF is on artificial machine reasoning.

A topic map represents information using an index model called TAO:

- Topics, or subject fragments (atoms).
- Associations that explain relevance and meaning.
- Occurrences of independent information about topics.

Every knowledge item we want to talk about is a topic which has a name and a type-category. Relationships to related issues are made by association (e.g. see also ...). Finally, occurrences are pointers to specific documents or other representations of knowledge that are asserted to be relevant to the named topic.

As I studied Knowledge Management in the beginning, I was drawn to topic maps over RDF and OWL, as it seemed to avoid wallowing in syntax and logic. Even Topic Maps have sought validation through this kind of formal computer science approach however, and I believe that the biggest flaw in Topic Maps was in choosing to model in the classic database approach. This led to unnecessary constraints, such as non-overlapping type categories, see Dietz (2006); Kipp (2003); W3C (n.d.).

The reason this approach has failed is a basic limitation on human willingness to get involved in highly technical reasoning. Modelling knowledge with logic is very hard, and not very convincing as knowledge is based on human faculties which are seldom fully rational and are never uniquely structured. On a scale of posting a note on the refrigerator to publishing a scientific paper in *Nature*, Topic Maps and RDF are much closer to the latter. That is simply too hard for most users, and hence these technologies curse Knowledge Management to be the domain of wizards and kings, unable to capture the knowledge of everyman.

## 5. Promises as a model for knowledge

To motivate a solution to the three problems above, I want to take a simple point of view based on a modelling framework known as *promise theory*<sup>1</sup>, which has yielded some success in understanding issues related to knowledge management: organization and cooperation. Promise theory is a way of describing how individual 'agents' advertise their behaviour so that they can form expectations of one another. They do this by communicating promises. An agent can be a human, group of humans, or some other entity being steered by humans. Like all models, it is intentionally simplistic, but I shall claim (in the interests of knowledge) that this approximation can be a good anchor point from which to begin a more sophisticated understanding.

### 5.1 The promise model

Promises are about trying to understand and govern expectations, given that we (i.e. people or agents) have at best incomplete information about the world. If we knew everything, there would be no use for promises. Promises made by agents to each other act as a kind of signalling mechanism to raise or lower expectations. Promise theory is uniquely suited to studying knowledge management because it captures something about the human condition: subjective individuals working together in a partially cooperative environment.

**Definition 3** (Promise). *The public expression of something that is, has been, or might be intended, made by an individual agent (the promiser), to a limited audience called the scope of the promise, which generally includes the promisee(s), i.e. the intended recipient(s) of the promise.*

A *promise*, in this technical meaning, is a statement of intent, by an agent, that is meant to reinforce another's expectation that the intention will turn out to be true. For instance:

1. I promise to feed your cat – expectations about what might happen in the future and cannot yet be confirmed.
2. I promise I have fed your cat – expectations about what might have happened in the past but has not yet been verified.
3. Feed cats seems a promising strategy – we speak of the expectation that feeding cats will bring positive outcomes in the future.

Clearly each of these statements can be considered a matter of knowledge. In each case, promises are about expectations<sup>2</sup>. Consider next the following promises about someone's understanding of certain terms:

- I promise that a 'chapter' means a part of a book, in the context of literature.
- I promise that a 'chapter' means a sub-division of a religious order, in the context of churches.
- I promise that chapters consist of words, in the context of literature.

<sup>1</sup> Actually this phrase was coined rather loosely, and its authors have tried to find a more specific name for it such as 'micro-promises' that is less omnipotent in its claims, but simplicity often reigns above reason, as we shall see in this essay.

<sup>2</sup> In philosophy, promises are usually thought of as a moral issue, but here we shall discard any moral connotations and deal exclusively with expectations. The promise of good weather, for instance, means that there is some expectation on the part of listeners that the weather will turn out well. This is not a promise made by morally good or evil clouds, but rather an imagined intention (embodying a harmless anthropomorphism) made by holiday-makers and wedding planners, etc.

- I promise that chapters consist of persons, in the context of religious orders.

These are promises about what an agent thinks he/she/it knows. Does anyone actually know these things for certain, i.e. are they facts? I think not. When, as individuals, we say 'a chapter is a part of a book' this is a confident statement, steeped in self-assurance. Anyone haunted by a modest amount of scientific humility would be less unilateral in making this kind of assertion and would probably try to qualify it with all kinds of uncertainty. Of course, what we mean is: to the best of my knowledge and belief, this knowledge is correct – please trust me, see Bergstra & Burgess (2006). We must not call these statements facts, but rather subjective expressions that might or might not be confirmed by others' viewpoints, thus essentially a promise.

Promises themselves constitute what we think we know, but they also discuss another level of knowledge. By thinking of knowledge as something that agents (i.e. you and I) have to promise to know, we will be able to reverse the simple design error in modern information systems that assumes authoritativeness, and turn logical ontology databases (designed by committees) back into simple language and hearsay that ordinary people own.

### 5.2 Basic principles of promises

There are two ways in which we are going to apply promises in this discussion: as a model of the people interacting with knowledge, and as a model of the knowledge itself. I will use the term 'topic' as a shorthand for knowledge item. Moreover, there are two kinds of promises that we should mention:

- Promises to offer or give something, e.g. I claim to be an expert in Foo.
- Promises to use or accept something, e.g. I believe your claim.

Both underline the individuality or autonomy of an agent, and we can use the term voluntary cooperation to express this freedom to disbelieve or not accept.

In the promise theory view, a promisee is not obliged to accept something promised by another agent. He or she must promise to accept a given promise in order for communication to be complete.

The basic principles of promises may be applied as follows, see Bergstra & Burgess (1994-now):

1. Anything that can be independently expressed or can change in some way can be an agent (promiser), entitled to its own viewpoint, and can make promises about its condition. We use this to say that any topic we can think of is assumed to exist and can promise to be related to any other, in the mind of an agent.
2. An agent (promiser) can only make promises about itself, not about other agents. Thus, in our case, a topic is only responsible for what it claims to know about itself and how it claims to relate to other topics.

Each topic is therefore self-contained, independent and can be unique in an individual's context, while still allowing for multiple interpretations. This allows multiple, private viewpoints.

### 5.3 Applying promises to knowledge indexing

The modern approach to category design is grounded in the goal of making sophisticated indices to make information accessible and to codify experience. To discuss how humans interact with knowledge, we shall make people (i.e. the users of knowledge) into the role of

promise-agents, in the first case. They will make certain promises to one another, which will involve agreeing about meanings, responsibility to learn and use concepts, etc. This is fairly unambiguous.

To apply promises to the representation of knowledge itself, we have to think more abstractly. Each topic, or item of knowledge, will be an agent that can make promises. The promises a topic can make are like the following:

- A topic promises a brief explanation of itself, in a given context.
- A topic promises to be *associated* with another topic in a particular way.
- A topic promises references *occurrences* of information that are about itself.

According to the rules above, no one else can make these promises. An example of the latter might be written like this:

```
topics:
  literature::
    "Odyssey"      # The topic/promiser
      comment => "A book from classic Greek literature",
      association => a("was written by", "Homer");
  authors::
    "Homer" comment => "A Greek writer from around 850 BC",
      association => a("wrote", "Odyssey");
  television::
    "Homer" comment => "Usually refers to Homer Simpson";
```

Promises therefore have two relationships to knowledge:

- Every time we propose to know something, the inevitable uncertainty on the part of the involved parties (promiser and the promisee), means that the assertion at best has the status of a promise, not a fact.
- Such a promise, once made, becomes a form of meta-information that describes the structure of knowledge, and is thus useful for analysis and reasoning.

We might write the above in a formal promise language, just to drive the point home:

```
topics:
  knowledge::
    "uncertainty"
      comment => "Incomplete information about something",
      association => a("is a basic assumption of", "promises");

    "promises"
      association => a("may be viewed as", "meta-information");

    "meta-information"
      comment => "Information about some other information";
```

Promises themselves constitute information and hence may be perceived as a kind of knowledge; moreover promises may be about knowledge itself. Conversely, knowledge about a promise can influence the behaviour of agents who are not the intended promisee, so knowledge about knowledge can be discussed with promises. From an engineering viewpoint, the scope can have an important influence on behaviour.



Why introduce these notions? The most compelling answer is that this model has several useful properties: it defines atomic 'topics' with a minimum of assumed structure, but embodies some principles that preserve things we know to be true about knowledge: that knowledge is subjective in both content and structure, and is defined by a collaborative process by individuals.

#### 5.4 Graphical representations - knowledge maps

Whenever things are interrelated, they form networks. In mathematics, the technical term for networks is *graphs*. The World Wide Web is one such network. There is a science of networks and their properties that has been studied at length in the literature, see Albert & Barabási (2002); Newman (2003). Since networks play such a large role in knowledge transfer, any theory of knowledge management must take their properties into account.

The agent making the promise (called the promiser) often directs it at a particular agent (one or more promisees), but others may also know about the promise. We call the set of agents who know about a promise the *scope* of the promise. There is thus communication from

$$\text{promiser} \rightarrow (\text{promisee} + \text{scope}) \quad (1)$$

We define a knowledge map as follows:

**Definition 4** (Knowledge Map). *A directed graph  $\Gamma(T, A)$ , where  $T$  is a set of nodes representing topics, and  $A$  is a set of edges or links representing associations between topics.*

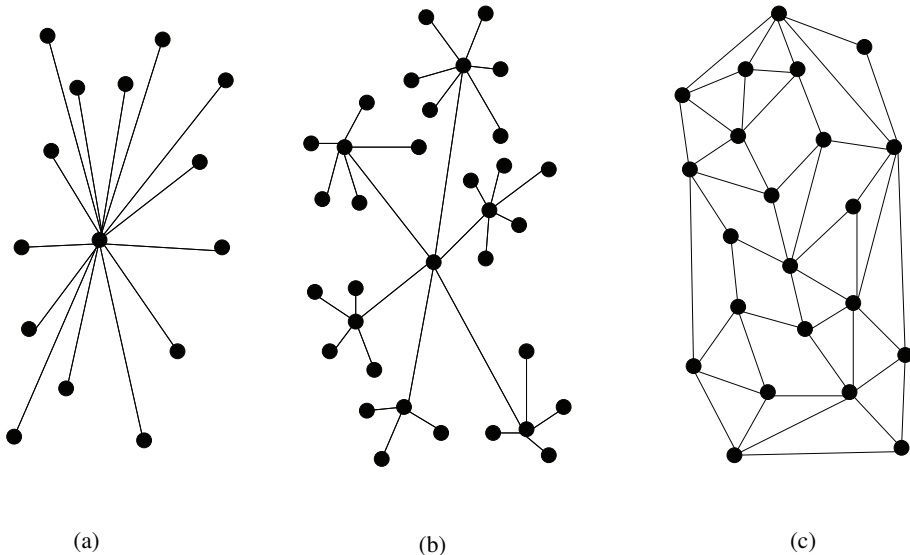


Fig. 1. Networks with different shapes: (a) A star network, (b) a tree, (c) a mesh.

Graphs or networks have many different shapes. If we begin to associate ideas freely, we end up with a 'mesh' (see fig 1 (c)). On the other hand, if we try to subdivide topics into categories in a 'branching process', we get a tree or hierarchy (see fig. 2 (b)). A tree is also called an acyclic graph (or DAG for Directed Acyclic Graph) because it contains no loops. A single category looks like the figure (a).



When we subdivide subject categories in the manner of a hierarchy, we are laying an artificial tree on top of the mesh network and pretending that the new shape is a good representation of the old shape (see fig. 2). It is clear to see that we can lay many different trees across

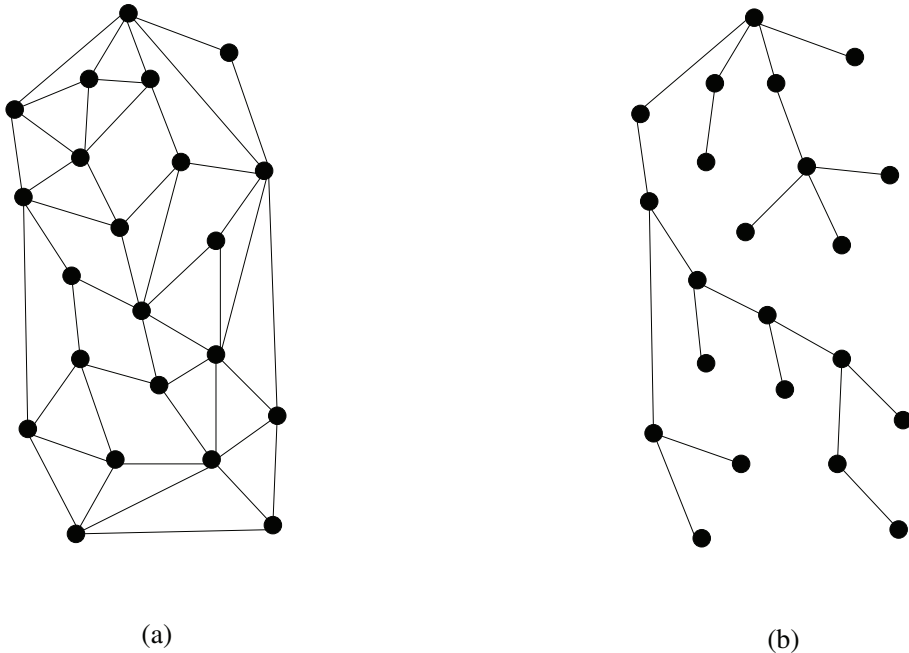


Fig. 2. Illustration of a spanning tree (b) for a general network (a).

a generalized set of topics, and so there is no unique tree that represents a given set of knowledge items.

**Definition 5** (Spanning tree). *Any DAG (tree) that starts from an arbitrary root node in a graph and covers all the nodes once only.*

Taxonomies have the property of a spanning tree (see sections below and fig. 3).

**Definition 6** (Singleton). *An isolated node in a graph, unconnected to any other. In our case, this is a topic with no associations.*

For the evaluation of graphs, let us define one more thing.

**Definition 7** (The degree of a node). *Denoted  $k$ . The number of associations a node has.*

If we plot a frequency diagram of  $n(k)$ , the number of nodes of degree  $k$ , this degree-distribution can be used to characterize the processes behind the structure.

### 5.5 Agreement and consensus

In discussing what individuals know as a group, we will have use for the notion of agreement. The term agreement is often confused with 'contract' in economics and social sciences, but that is a derived meaning. We are only concerned with what agents claim to agree about. This is also a matter of promises, it turns out, because agents cannot know if they actually have the

same understanding as any other. At best they can promise to agree to something given their best understanding.

Agreement is what happens when two or more agents seem to arrive at a common understanding. For example, two parties can agree that  $2 + 2 = 4$ . Two agents may or may not know about their common state of agreement. By promising to agree, they can make this public.

**Definition 8 (Proposal).** *A proposal  $P$  is a prototype promise, i.e. the statement of an intention that has not yet been made public, or acted upon.*

**Definition 9 (Agreeing).** *Agents are said to agree about a proposal  $P$ , if they independently promise to adopt  $P$  i.e. if they formally promise to use the proposal and make the promise themselves.*

Agreement is the basis of most of cooperation, and it is the way in which agents arrive at a common understanding. It is therefore central to knowledge management.

Agreement, about some body of information, can thus be viewed as a number of promises. In a contract, for example, one writes down a number of proposals for each side to follow (which are themselves prototype promises), and then the parties promise to subscribe to these by signing. If all parties promise that a set of proposals will be honoured, then an agreement may be expressed as a promise to keep some specification or promise proposals. This may be called the *body* of the agreement. The term contract is also used here.

**Definition 10 (Agreement).** *A promise agreement is a pair of use-promises between two parties to acknowledge and adopt the body of a proposal.*

Now, armed with this briefest introduction to the promise model, we can get back to the main story: knowledge.

## 6. Knowledge maps

### 6.1 Some definitions

To speak of a technology for knowledge maps, we need to formulate a more precise and technical definition of categories, using a promise model. Let's begin with some core concepts.

**Definition 11 (Taxonomy).** *A hierarchy of topics organized into categories in the manner of a tree with parent (container) concepts and children belonging to parent-categories.*

Taxonomy has been one of the main tools for classifying 'things' in the world, especially in biology. It is a way of putting things, concepts and ideas into one and only one box. Because every concept can only belong to a unique part of the tree hierarchy, the structure of knowledge is fragile to mistakes in choosing the wrong categories for something. Because everything that follows depends on the decisions made, making a change to a tree can be an expensive operation that requires unpicking and redesigning. Moreover, trees are branching processes, they tend to lead to too many different locations for information to reside, and trade complexity of information for complexity of categorization<sup>3</sup>.

<sup>3</sup> I often refer to this as the depth versus breadth problem. If we try to hide complexity inside category containers, and sub-containers we simply turn a one dimensional list into a two-dimensional structure, but the total number of things one has to deal with can still be the same, depending on the effectiveness of the categories.

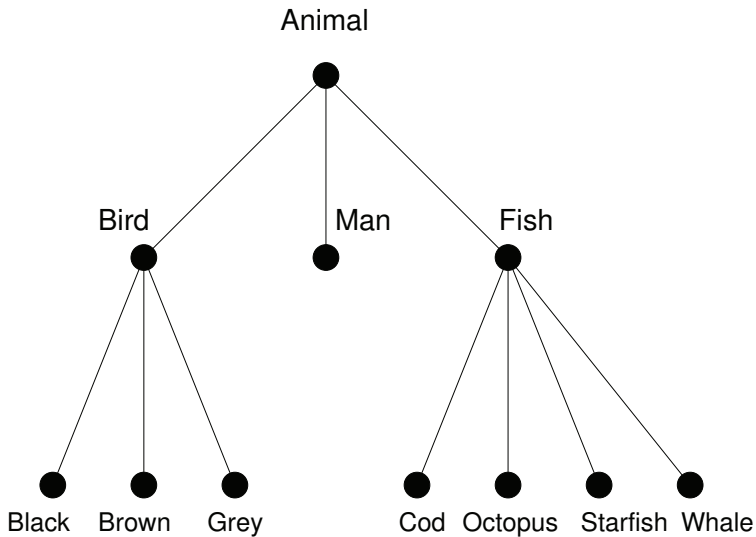


Fig. 3. A taxonomy usually has the structure of a tree - which has the technical name of Directed Acyclic Graph (DAG).

Since taxonomies are non-unique (they are merely arbitrary spanning trees, viewed from a particular starting point), they cannot be considered fundamental. They are design structures, often decided by committees or standardizing bodies, in order to achieve a consensus.

Taxonomies are about putting things in one special box with a special name. A less contrived structure approach to classification is to use a network representation where names are less important than qualities. Semantic webs, or the ontologies they represent, are designed to allow this by letting anything associate itself with anything else, without due regard for boundaries, by using flexible associations.

**Definition 12** (Ontology). *A mesh of topics and categories, supplemented by promises of associations to other topics.*

The state of IT ontologies today borrows a lot from taxonomy, in that it is usual for concepts or topics still to be arranged into hierarchical boxes. In this respect, ontology has been unable to divorce itself from hierarchical taxonomy. This seems to be more a case of habit than actual judgement – a hangover from computer science’s database modelling and object orientation doctrines. However, and I would like to propose that this causes more problems than it solves.

## 6.2 Survivability of an ontology

If taxonomy and ontology pretend authority from arbitrariness, what might then be fundamental? Promise theory emphasizes that ‘fundamental’ is a subjective issue: it belongs to a specific agent’s point of view, which in turn is limited by what information is available to it. If we are to build something fundamental, we must therefore base it on subjective viewpoints. Indeed, it suggests that everyone has to build their own viewpoint. This is what ontologies try to achieve and fail at because they retain a notion of authority.

A set of associations can just as effectively capture worthwhile aspects of hierarchy, by documenting which concepts are generalizations of others. For example ‘animal’ is sometimes

the right generalization for 'bird' or 'fish', but not always. It is also true that information classified as applying in any context generalizes information true only in a special context. These are set relationships, not hierarchical ones. With a 'generalization promise', any topic can bear allegiance to a number of possible generalizations in different contexts, without pre-designed limitations.

The survival of an ontology depends on its being used, which in turn depends on the agreement of all agents: i.e., keeping the promises to use the terms correctly, by all those who participate in it.

How does one know if a taxonomy (or technical ontology) has been a success? In our promise approach, we can simple measure this as the extent to which users keep the promises to support the ontology.

## 7. The economics of knowledge

The giving and receiving of promises quickly turns into a question of economics. There are motivations, incentives to act and costs involved in acting to keep promises. The economics of knowledge will be central to understanding both the weaknesses of current approaches to knowledge management, and the way towards a more 'natural' or less contrived approach. By taking an economic approach to the subject we join the ranks of scientific models in which rational modelling (or bounded rationality) is the explanation for motivations.

Knowledge management involves activities like the following:

- Acquiring knowledge, with associated gain and cost of effort.
- Retaining knowledge, a maintenance cost.
- Disseminating knowledge, with cost of effort. This might be written off against the maintenance above, but it is normally a cost. Disseminating knowledge can also be considered a long term investment however – investing in others' educations, means that we might save on work down the line.

The extent to which knowledge continues and grows in the minds of agents is therefore a question of the individual economic considerations of those agents. It costs little and brings rewards, knowledge will flourish. If the cost is too high, knowledge will not be maintained or spread.

The expected payoffs can be short term or long-term, and any human interpreting the value of a knowledge promise will form their own value-judgement based on how long they are willing to wait for a payoff. Through the promise model, there is a natural connection with economic game theory here, see Burgess & Fagernes (n.d.).

To model what we may expect of knowledge and its usefulness, we start by writing down the promises that are relevant, and what the benefits and costs of having these promises kept might be. Due to the subjective nature of promises, we can only suggest these things in broad terms, and it will remain for specific contexts to determine values for these things.

Agents (humans, machines or service centres, etc) can promise

- To reveal information.
- To use information revealed to them.
- To document or write down information.
- To train or teach others to interpret information.
- To search for answers.

### 7.1 The value of promised knowledge

It is not easy to come up with a universal measure of value for knowledge, of course, but in the spirit of modelling that is what we are going to pretend. Since there is no standard for this, we are free to invent one based on the promise model.

**Definition 13** (The value of a promise). *Let  $V(\pi)$  denote a function, which when applied to a promise  $\pi$  returns a real number with the interpretation of a payoff/return. A negative value represents a net cost associated with keeping the promise.*

Let us suppress, for now, the issue of when the payoff occurs, and assume that keeping the promise leads to rewards that we are accounting for far off in the future, at some final reckoning. We may add up values over the time each time the status of a promise is assessed, with values for:

1. A promise found to be kept, positive if good or negative if bad.
2. Costs associated with keeping the promise (always negative).
3. The promise was not kept leads to a possible loss for the promisee.

However we choose to account for these values, they are perceived very clearly in our minds when we interact with knowledge.

The hypothesis I propose is then that it must be possible to increase the net value of knowledge by adopting some simple strategies.

**Hypothesis 1** (Increase value and reduce cost). *Some principles:*

- *Knowledge is made more accessible by reducing the cost of lookup.*
- *The value of information rises when accompanied by bonus associations to related topics.*
- *The cost of knowledge assimilation can be reduced by avoiding knowledge management overheads.*

The cost of knowledge includes the work done acquiring, using, transmitting it, etc. If we use accounting terminology, it is a simple matter of time and materials. The cost of information retrieval or lookup by either a person or a computer is related to the algorithmic complexity of the search required to find it. This translates most likely into the time a user has to invest in accessing information.

The costs are probably quite different depending on the representation of the knowledge items one uncovers while searching. For instance a book is likely easier to interpret than a painting, As a general question we would like to know how such costs that inhibit the learning and dissemination of knowledge can be reduced.

### 7.2 The cost of categories

Classification into categories plays a central role in this essay because it has been such an important activity in knowledge engineering for centuries. Why is it so popular? It was introduced in order to cluster together books about similar subjects. The contention is that this reduces tangibly the cost of finding relevant information, assuming that you understand the classification in the first place<sup>4</sup>. Let us test this idea.

Suppose there are  $T$  topics divided into  $C$  categories. In a linear search (starting at the beginning and running through until we find the right one). If there is only one category, then the cost of finding a topic on average is about half the length of the search list:

$$\chi_1 = \frac{1}{2}T \quad (2)$$

<sup>4</sup> Librarians are trained knowledge engineers who learn classifications and became smart assistants.

If there are multiple categories, this generalizes to:

$$\chi_C = \frac{1}{2}T/C + \frac{1}{2}(C - 1) \quad (3)$$

So, if categories are to have any economic value,

$$\frac{1}{2}T/C + \frac{1}{2}(C - 1) < \frac{1}{2}T. \quad (4)$$

This gives us a quadratic constraint for  $\chi$ , which is easily solved to give  $T > C$ . In other words, introducing categories is always likely to reduce search times a little for any number of categories, assuming that one knows which category the topic belongs to. The saving  $\Delta\chi = \chi_C - \chi_1$  actually shrinks with the number of categories however, since, if we write the total number of topics  $T = t + C$ , as there cannot be more categories than topics, then the saving is

$$\Delta\chi = \frac{1}{2}(t - t/C), \quad (5)$$

where  $t$  is the number of 'pure topics', i.e. those that are not themselves subject categories.  $\Delta\chi$  gives a constant saving for large  $C$  at around half the number of non-categories. Thus the saving is not large if we go berserk with sub-categories. The cost does not depend on whether the topics are arranged in a flat partitioning or in a tree either<sup>5</sup>, so hierarchy does not help here either.

### 7.3 Cost of adding a category: missing freedoms

Categories form lists that we call directories, that we are still familiar with today (e.g. yellow pages, and Yahoo). These stem from a time when libraries were the most important search engine. Classification was conceived of as a cost saving mechanism, and for a few privileged scholars with more expert knowledge, it also worked to identify larger patterns that enhanced the meaning of the information. This worked because scholarly subjects were handled by a few well-trained people, and subjects were placed into relatively few categories or fields of study. There was little focus on overlap, more on the 'nobility' of knowledge.

For most users, however, the cost of categorization gets counted twice: both when using knowledge and contributing knowledge. The cost  $\chi_C$ , above, is the cost of lookup, but also the minimum cost of adding something to the category structure, since placement requires us to parse the model to look up the right place to add something. There can be additional overheads:

- A search for the correct category is required.
- If a new category is needed, it must be localized and agreed upon by the authorized designer of the taxonomy.
- If no category can be added, topics become orphaned.

The cost of agreeing on a change to the taxonomy depends on its governance, for instance:

- All agents must agree.

<sup>5</sup> The approach of introducing categories is similar to the use of hashing algorithms to locate values by replacing a linear search with a cheap constant-cost function that finds the approximate location of a value, but the difference is that there is no semantic hashing function to reduce the cost of finding subject categories in a list.

- An elected body must agree.
- Anyone can decide to change the categories.

A significant cost of working with knowledge thus comes from the need to anchor the topic in a particular place. If we could simply dump knowledge in a known location (e.g. treat every instance of the word as simply a word) then this can be done with essentially zero cost<sup>6</sup>.

What this shows is that, from the perspective of someone wanting to contribute to knowledge, the presence of categorization adds a large cost deterrent to the activity. This is a plausible reason for the failure of ontology and semantic modelling so far today.

This need to choose a unique category is major hindrance to creating the model, and getting data into the model. What if I make a mistake? Every programmer knows that putting data into the wrong class or structure causes huge problems down the line, so you'd better get it right! But wouldn't it be nice if you didn't have to be so careful? Isn't the computer supposed to help us, not the other way around?

#### 7.4 The cost of knowledge maintenance: memory and repetition

Repetition is a key tactic in learning and training. Repetition serves two goals: to gather experience about the consistency of information, as well as confirming and reinforcing a fixed message, thus increasing its value. The scientific method, for example, is based on the idea that verification of knowledge increases its value.

It will come as no surprise to anyone that merely documenting something cannot be viewed as an automatic strategy for increasing anyone's knowledge about it. Promising to write something does not imply an obligation on the part of a reader to read it. There must be a corresponding promise to read what has been written in order for it to be effective. Once read, knowledge must be remembered in order to be used. This suggests that there must be an approach for memory.

There are three kinds of memory:

- Rehearsed memory (e.g. muscle reflex memory).
- Short term memory (brute force, short-lived cramming).
- Long term cognitive memory (associative pathways).

In economics one uses dilemma games, to estimate the trade off between short and long term strategies, see Burgess & Fagernes (n.d.). I have no comments to make on that here. What this suggests is the following: if we are to improve the actual utility of knowledge over time, then we have

**Hypothesis 2** (Knowledge requires practice). *Any knowledge management scheme must encourage user to interact with it regularly.*

Positive reinforcement is needed to turn information into knowledge, and this repetition incurs a cost.

#### 7.5 The Dunbar numbers

It is worth mentioning, as an addendum to the economic question, some limitations we should expect to bump into in knowledge engineering. Science always throws up certain scales that need to be attended to in understanding organized phenomena. The Dunbar numbers could be such scales for Knowledge Management.

<sup>6</sup> This is the logic behind hash tables in computing.



Studies by anthropologists, interested in the origin of human intelligence, have demonstrated statistical evidence for the idea that our capacity to perceive and know things and people well is limited by our brain size, see Dunbar (1996); Zhou et al. (2004). The evidence for this assertion comes from studies of inter-human relationships, but from there it is not a huge stretch to imagine that similar limitations apply to any kind of learned acquaintance, such as acquaintance with knowledge in its various forms.

The so-called Dunbar hierarchy identifies some key numbers that suggest economics limits on level of intimacy we can have to knowledge, because more intimate knowledge has a greater cognitive cost. The precise implications for knowledge management are, as far as I know, not fully understood, but the following numbers can be expected to appear if the hypothesis is correct (parentheses describe the original inference).

- 5 Detailed intimate interactions (close friends).
- 15 Team-level or frequent interaction (teams).
- 30 Daily working group size (tribe, workgroup, extended family, etc).
- 100 Things we recognize and understand (acquaintances).

Table 1. Key numbers from the Dunbar hierarchy, relating cognitive cost.

These numbers might appear anywhere cognitive cost plays a role. When presenting knowledge, for instance, we can expect people to have difficulty in relating to large numbers of choices and interrelationships, e.g. returned search results, number ingredients in an idea, number of steps in a recipe, etc. It would not be right to speculate unduly on how the Dunbar numbers might apply to knowledge management, but it is worth flagging this subject as worthy of further study.

## 8. From type hierarchy to overlapping contexts - a cheaper solution to encourage participation

Experience seems to show that users rarely contribute their own expertise back to projects that attempt to build taxonomies or strongly typed ontologies. It costs them too much. The same applies to Wikis that are organized hierarchically, because users either cannot find the right place to put something or they put it in the 'wrong' place, creating little value. The problem lies in quickly knowing how things should be organized in relation to one another.

Why is it so hard to know what topics should be related and how, to see what information is going to be needed and in which context? The answer is simply that this decision involves creative design. It is not a matter of pre-determined fact, but an arbitrary choice – but we don't like arbitrariness, so we look for agreement within a group or permission from an authority, etc. What started out as a simply desire to share, becomes an exercise in multi-party logistics. There is thus a significant 'mental computation' involved in this.

Suppose we could add a topic wherever we pleased, with some context to explain our usage. Then the cost of adding would be reduced by the entire cost of searching for the right place. Instead of an  $O(N)$  search, we have an  $O(1)$  insertion, costing the user little effort. Let's examine this idea further.

### 8.1 The meaning of domain and context

Users embellish facts with contextual information and want to emphasize certain aspects of knowledge. This freedom must be allowed in any account of KM. Terminology might begin as a set of unique terms, but quickly becomes distorted by ordinary linguistic creativity into slang

and jargon in special contexts. Context is therefore both highly important, and somewhat neglected in knowledge research, see Deutsche (2006).

Users further seek generalizations of things, because we all look for common patterns as a way of compressing a large number of special instances to a single representative category<sup>7</sup>. The danger is that we become obsessed with dividing and subdividing knowledge into precise categories, perhaps spurred on by a feeling that information will get lost if we don't put everything into exactly the right box. The problem with packing things into boxes, as many will have experienced, is how to find the right box again later as the number of boxes grows large. Moreover, we often change our minds about how to classify information so what was classified last year is not findable tomorrow unless we refresh our understanding of context. The first step in traditional classification schemes is to break things into a taxonomy. But a majority perhaps of concepts do not merely fall into just one category and so this artificial notion works against us once things go beyond the trivial. One has to either be an expert on a particular model of categorization or perform a brute-force search through the model to find the appropriate category.

If we look more critically at the way humans think outside of what we've been taught in school, this is not what we do. Our minds tend to do the opposite: we generalize from little evidence into much broader categories, implying that we are not that consumed with an obsessive compulsion for semantic correctness.

## 8.2 Topic type, a redundant concept

Topics play a dual role in the topic map standard, as:

- Subject identifiers that point to explanatory documents and semantic relationships (*t*).
- Context 'containers' or categories for other subjects (*C*).

In my own testing of knowledge modelling, attempts to use typing of topics have proven excessively difficult, and have thrown up many conflicts and singletons from unnecessary repetition during modelling. The fixation on 'type', apparently shoehorned into ontology languages from the historical origins of database modelling, seems to have been both a red herring and an expensive hindrance to locating useful information. To see what we can replace type with, we need to define some terms.

If knowledge is to be used and interacted with regularly, it will become the domain of a particular social group. Let's assume that such a group converges on a basic set of ideas. A knowledge domain is then any set of topics claimed by a group of individuals.

**Definition 14** (Knowledge domain). *An arbitrary set of topics used commonly by a group of users, forming a cultural body of knowledge.*

Unlike the case of type annotation, there is no limitation on overlapping between different knowledge domains. The contention in topic map modelling is essentially that data-types model different cases. Again, this seems to flow more from a classic computer science dogma, based on some kind of entity relation model, rather than on a clear philosophy of the problem. In addition to a domain of knowledge, there is also the idea of usage in circumstances that are governed by factors far outside of the domain itself, such as time and environment.

**Definition 15** (Context). *Any set of topics, either associated with one another or not, that describe the current situation of an agent when using or searching for information.*

---

<sup>7</sup> This is the basic approach behind all data compression.

Contexts are words and phrases we attach to a topic to disambiguate its usage. A context need not be either semantically or categorically related to the topic it describes. For example, the knowledge about 'tea' can be in the context of 'flavours', 'botany', 'drinking', 'suppliers', or even 'afternoon' none of which has any particular affinity to tea other than by association.

As mentioned above, it seems that, more by habit than reason, one has used a hierarchy of non-overlapping category types to disambiguate usage in different contexts. This seems to be a simple error of modelling. The problem is that it mixes up two different models unnecessarily: a description of generalizations or 'type' (which is forced unnaturally to be unique) and a model of brainstorming relationships to related things. We do not need a type to disambiguate usage, just another existing topic within the model, with no special status.

### 8.3 Contextualization is not strongly ordered

To illustrate why context is not at all hierarchical, consider this example of a geographically distributed organization, with finance, engineering and legal departments in three countries. Let us suppose that the organization has headquarters in 'usa', 'uk' and 'norway', and each branch has departments for 'finance', 'engineering' and 'legal' matters.

We have now two choices when making a hierarchy for the organization, depending on what we happen to think is of primary importance. In the first version, we treat geography as the primary distinction, and can express the full hierarchy like this:

```
usa.finance
usa.engineering
usa.legal
uk.finance
uk.engineering
uk.legal
norway.finance
norway.engineering
norway.legal
```

In this notation, the dot has the apparent interpretation of 'member' because the departments are smaller than the countries and are contained within them.

A different agent might feel that this model is upside down and that one should consider the finance department to be a unified global entity, with branches in three different countries. In that case, you would write

```
finance.uk
finance.usa
finance.norway
```

and so on. This example highlights the fact that we often want to slice and dice complex concepts in different ways, and attending too closely to a single hierarchical model prevents that.

If we think technically for a moment, the key observation is to notice that the '.' (dot) operator is really an intersection of sets (AND)<sup>8</sup>, and that this is a much more flexible notion than hierarchy.

<sup>8</sup> It is a commutative operator, which is why it makes sense to write both `usa.finance` and `finance.usa`.

Underlying hierarchies and networks is the concept of *sets*. A set or *collection* of something is just a number of instances that satisfy some property. For example, the *set of all vending machines*, or the *set of times between 2 and 3 o'clock*. Sets can be thought of as networks in which the elements are all joined to each other by a common relationship 'in the same set as'.

We often write subset membership using a membership '.' character, e.g. if `linux` is the set of hosts with property 'linux', then a subset (or sub-class) of these hosts is 'debian' (see figure). The class *64 bit hosts* is not a subset of `linux`, as part of it lies outside. It is a subset of *hosts*.

```
usa.finance
usa AND finance
usa ∩ finance
```

Context sets have the property that

```
usa.finance = finance.usa
```

i.e. the commutativity of membership ordering. Hierarchies do not have this property.

Sets can be made hierarchical when every subset is contained entirely by one and only one parent set, and in turn contains zero or more whole subsets which it does not share with any other. The problem with hierarchical sets is that they are too restrictive. If you design them incorrectly in the first place, you shut parts of the organization inside a box that prevents other parts from accessing them.

With sets, we can perform filtering based on logical reasoning, just as with search languages – but in a very efficient way. We can promise to association meanings by set-computation:

```
classes:
```

```
"english_speaking"
  expression => "(usa|uk)!.!legal";
```

Thus the English speakers promise to identify themselves as those entities belonging to the USA 'OR' to the UK, excepting the legal department (! means NOT).

Henceforth, I will use the notation 'context::topic', e.g. 'X::Y' to mean a mention of a topic Y in context X.

#### 8.4 Topics in multiple contexts

Consider a slightly different case of 'homonyms', i.e. words that have multiple meanings in different contexts. As an example, I shall borrow from the Topic Map literature a fascination with opera as a knowledge domain, see Pepper (2009), by examining the topic "Peter Grimes" which is a character from a poem made famous through Benjamin Britten's acclaimed opera. What type or types should this topic have? We might interpret a mention of the name in multiple ways:

- As a name, e.g. 'names::Peter Grimes'.
- One or more persons with the same name, e.g. 'names.friends::Peter Grimes', 'names.tv\_show::Peter Grimes'.
- An opera, e.g. 'opera::Peter Grimes'.
- A libretto from an opera, e.g. 'opera.libretto::Peter Grimes'.
- A character from an opera, e.g. 'names.opera::Peter Grimes'.

- A character in a poem (The Borough) on which the opera was based, e.g. 'names.poem::Peter Grimes'.

And so on. The list is potentially infinite.

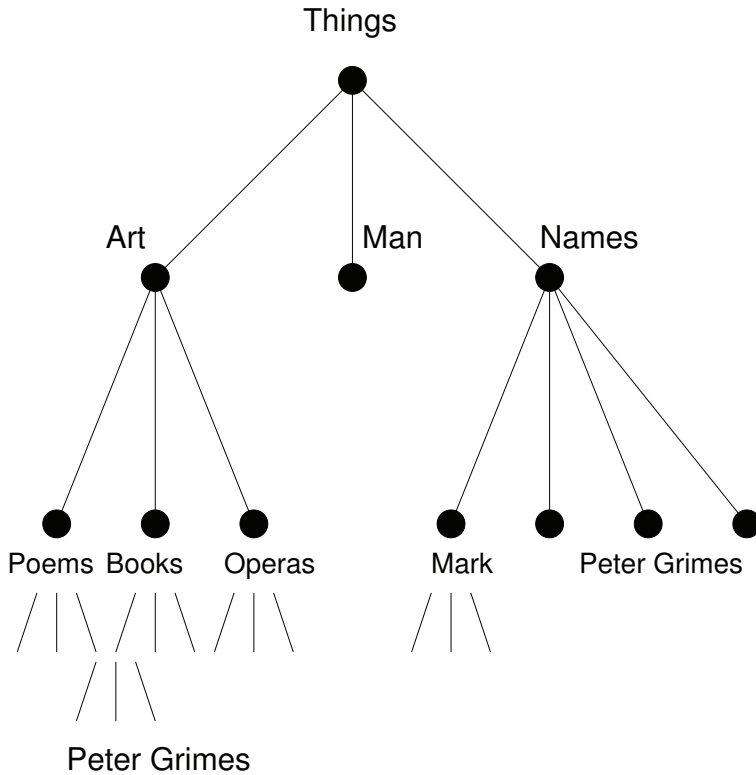


Fig. 4. A single topic, like 'Peter Grimes' is really a linguistic element, that has usage in many different contexts. Arranging topics in a tree confuses context with parent-node, which is wrong.

The desire for simplicity and parsimony encourages people to think about topics as falling into neat, mutually exclusive categories, but what we see is really something with a much more linguistic freedom: a single phrase 'Peter Grimes' used in a wide variety of overlapping contexts, with slightly different meanings.

At this point, most people feel an uncomfortable need to anchor the righteous place of each usage in their model by making an exclusive choice. Suppose this is to place this name within the category of opera, along with Aida and The Ring cycle, etc, but what criterion does one have for deciding on types? In fact, a type is just a topic itself, and the entire type notion could be eliminated in favour of an association: Aida "is an" opera, as one does in an object oriented model approach.

### 8.5 Reasoning about categories in searches

Consider occurrence of text for different interpretations of Peter Grimes. There is a book of the libretto

occurrences:

```

peter_grimes.opera::

  ``http://onlinebooks.com''
    represents => { ``Book about the opera'' };

  ``http://onlinescores.com''
    represents => { ``The complete score of the opera'' };

peter_grimes.name::

  ``http://names.com''
    represents => { ``The origin of the name'' };

```

In our new set interpretation, the sub-set operation ‘.’ is commutative and reflexive. So it doesn't matter if we consider `peter_grimes` to be the superset or `opera` to be the super-set. Suppose however that the first book contains the complete text of the libretto and the etymology of the name, an explanation of the poem, etc. Then it is relevant to all these contexts, or indeed in a generic context ‘any’ and should appear in the results of any search.

If we interpreted `opera`, `libretto`, `poem`, `name` as type categories that do not overlap (e.g. as a conventional topic map) then it would be necessary to register this book as an occurrence in every single category – i.e. with multiple registrations; that is because the combination type+topic is a unique entity, and thus multiple types significantly increases the cost of documenting this information for users. The overlapping set model collapses all these registrations into one, but is not broken by multiple references, so we achieve two things: a reduction in cost of inserting data, and robustness to inserting multiple times.

Using context sets, we have many more possible ways to give useful information. Suppose we search for `peter_grimes.opera`, returning results for `peter grimes` in any category is generally more helpful than unhelpful to a human being. The issue then becomes for whom are the results intended? If we admit that humans play a role in the process (because they are far superior reasoning agents than software) then a freer interpretation is the correct one. This requires less expertise to set up and leads to better results.

Conversely if machines are to do all the work, users must have access to a complete and mathematically correct technical ontology, with type-correct documentation to yield precise search results. The only way to address this is the Topic Maps standard is the introduction of a search language, which pushes the complexity back onto the user, violating the first principle. The user is forced to fight the logic of the system rather than using it for inspiration. In practice it is rare that we want to restrict information so stringently as through a logic of types – and when we do so, we often end up finding nothing because the data are so over-constrained that the intersection of all constraints is the empty set. We need to simplify all this structure drastically.

## 9. Emergent norms and common knowledge – the recipient's view

An interesting and highly relevant question is thus the following: given a free approach to ontology, based on context rather than a pre-arranged taxonomy of types, would a free

user process of adding topics and associations converge to a graph that auto-selects a set of attractors we might call popular 'well known concepts'?

It is tempting to answer: yes, this must be so, since our natural language evolves in basically this way, and seems to have achieved just that. However, we also know from natural language that, when a sufficient number of individuals is involved, languages fragment and sub-cultures emerge. All of these things are natural from a network point of view however.

Let's sketch out how one might go about discovering whether this is possible.

### 9.1 Norms, swarms and attractors

We would like to know if a group of agents, making no promises to obey a predetermined ontology, would effectively promise to follow an emergent ontology after a sufficient amount of time. Understanding this hypothesis fully goes beyond the scope of this essay, but we can sketch out some of the issues.

The concept of emergent behaviour is tied to so-called swarm intelligence, see Bonabeau et al. (1999) and has enjoyed a fashionable period over the past 20 years. It has brought both insight and a lot of hype to modelling. Let us focus on a simple promise model of swarming that attempts to bring a simple but clear meaning to how swarming and emergent norm-formation (normation) takes place, see Burgess & Fagernes (2007a,b).

A swarm is simply a flock of agents (birds, insects, etc) that seem to exhibit collectively organized behaviour, even though each of the agents is a free entity with only weak links to its nearest neighbours. Swarms often come together to minimize costs of some kind, e.g. the cost of protecting each agent against predators. We call such behaviour 'emergent' because it is not explicitly designed, but is perceived as a side effect of something else, by a particular user's perspective.

In promise theory, emergent behaviour is explained by noting the indistinguishability of certain collections promises from others. Without getting into details, we say that a system has emergent properties if it seems to promise something, from the viewpoint of an external observer who in scope, that in fact it does not explicitly promise, see Burgess & Fagernes (2007a) In the same way, it is possible for a knowledge model to make no explicit design promises about category and yet still form a structure that appears to cluster around certain 'attractor topics' in the manner of a hierarchy.

The spontaneous formation of hierarchies is a relatively well-known phenomenon in network science, see Newman et al. (2001); Watts (1999) and is related to the 'small worlds' phenomenon. This could provide an explanation for the preponderance of attention given to hierarchical organization. Put simply, what happens is that certain early-defined topics acquire an economic advantage to being used. Topics that have the most associations and usage tend to attract even more attention, and therefore acquire the status of an anchor point or emergent category for knowledge. This phenomenon is called 'preferential attachment'. For a simple review, see , see Barabási (2002).

What is exciting about this model is that it can be tested by looking at the statistics of the graphs that result from such a free collaboration. Preferential attachment leads to long-tailed or power-law distributions in the node degree  $k$  of the association graph, of the form  $N(k) \sim 1/k^n$ , for some  $n$ , whereas a designed hierarchy would likely show a much sharper distribution of node degrees, see Barabási (2002); Newman et al. (2001).

**Hypothesis 3** (Convergence of knowledge graph). *A knowledge map will converge to a graph with a power-law degree distribution if a type-free context model is used.*



We are not able to say what result a graph will converge to as users add associations and topics.

- An arbitrary choice by policy of a desired outcome for the meaning of a norm.
- An attractor or potential surface with a unique minimum, e.g. based on popularity.

We identify semantic 'votes' for discrete subjects, although many of the concepts might view things less precisely, living only in the suburbs of these concept's centra. Any suitable model must account for this uncertainty, and multiplicity of viewpoints (a town can have many districts).

## 10. Emergence friendly rules for ontology

Let's summarize what minor changes to, say Topic Maps, are needed to encourage spontaneous ontology, and lower the cost of knowledge development. The data model for topic maps contains no major errors or omissions, but it contains one unnecessary constraint that makes topic maps hard to build models with. That is the constraint that topic types should be non-overlapping categories, see Kipp (2003).

**Hypothesis 4** (Correction of Topic Maps). *We replace non-overlapping types with overlapping contexts so that a topic can belong to more than one contexts. Topic types become contexts, and topic names are registered only once, with associations and occurrences belonging to contexts, and topics existing universally as pure syntax.*

The beauty of this reinterpretation is that it does very little violence to existing technology, but extends the possible interpretation of the data in potentially valuable ways.

Under this new regime, we can assume that:

- All topics exist, whether defined or not.
- Only topic associations need be explicitly promised, including in which context they are relevant, i.e. in which context a topic promises certain properties.
- The context of a topic (i.e. the usage of the term or phrase) explains its semantics, not a classification of its type within a separate ontological spanning tree.

The 'current context' in a topic search, for instance, can be assumed from the path taken by the user through the history of topics, etc. This also motivates the idea of stories below.

The transition from selection by taxonomic classification to selecting topics by usage is subtle but wide ranging. It's not what the topic is, but how the term is used that is important. In other words, topics themselves are reinterpreted from labelled semantic concepts to being simple syntactic fragments.

## 11. Roles and collective promises – user-perceived black boxes

The Dunbar numbers, mentioned previously, suggest that cognitive complexity is related to the number of things we need to (promise to) know at different levels of intimacy. So a relevant economic question is: how can we reduce this number of items, and thereby reduce the cognitive cost for end users of information?

Categories are clearly an attempt to do reduce the numbers by providing umbrella concepts, but their introduction is often overwhelmed by hierarchical design issues. I believe that too great an emphasis is placed on the hierarchical aspect of the taxonomic category trees. Promise theory's basic tenets lead to a suggestion for this reduction that is simpler: spanning sets – or

what I shall call *roles*, see Bergstra & Burgess (1994-now). A role is simply a group of things that identifies some pattern amongst knowledge items (agents). If these agents or topics make similar kinds of knowledge promises, then they must play a similar role in the scheme of things too. Put simply, a role is a bundle of topics that make similar knowledge promises.

Some examples help to make this clear. For instance, at a low level, different categories of words play certain roles in the construction of knowledge; e.g. some words promise to be verbs, some promise to be nouns or 'things', and so on. These roles are functional and therefore have a practical value. Next, one might have other roles, like 'colour' or 'animal', which are less clearly functional, but more an attempt to put a name on a perceived phenomenon.

We can define such roles simply in terms of the promises they make:

**Definition 16** (Promise role). *A collection of agents or things that is assessed by a user as making the same kind of promise (or collection of promises).*

Patterns like this tie in with the use of repetition to emphasize learning, as mentioned before, and thus patterns are related to our notion of learning by rote.

Now, consider a different type of knowledge promise that is not about describing an intrinsic property of a single item, but is rather about interpreting the result of a collaboration between different promises that individuals might classify in very different ways. Take, for example, the concept of a radio. Someone might call a particular group of functional elements (e.g. electronic components) a radio. Each of the components promises certain properties like 'will act as a switch' or 'which store electric charge', if this collective set of components keeps the promise to play music from various radio stations, we might indeed call it a radio.

The attachment of a concept like 'radio' to a set of collaborating relationships is nothing like the naming that happens in a standard taxonomy: it is an interpretation, based on probably an incomplete understanding of the structure of the internal properties, based on a superficial evaluation of its behaviour. In a hierarchical decomposition one would separate the components into rigid categories like 'resistor', 'capacitor', 'transistor', or 'plastic' and 'metal', none of which say anything about what these parts contribute to.

A radio is thus an emergent property of a collaborative network of properties that has no place in a taxonomic categorization related to its parts. A radio is not more than the sum of its parts, as we sometimes like to say, but rather it forms a collaboration which comes alive and takes on a new interpretation at a different level. Typical taxonomic decompositions are reductionistic, leaving no room for the understanding of this as a collective phenomenon. This defect can really only be repaired if we understand that it is the *observer* or recipient, not the designer, that ultimately makes the decision whether to accept the assessment of a set of component promises is a radio or not.

**Definition 17** (Collective role). *A collection of agents that is assessed to form a collaborative role, if the agents work together to keep a promise that requires the participation of all the agents collectively.*

The concept of a radio is clearly much cheaper to maintain as a new and separate entity than a detailed understanding of how the components collaborate to produce the effect. We frequently use this kind of information hiding to reduce the cost of knowledge, but clearly knowledge gets lost in this process.

**Definition 18** (Black box). *The purposeful forgetting or discarding of knowledge in order to reduce the cost of accepting a collective role.*

The ability to replace a lot of complexity with a simple label brings great economic efficiency for end-users of knowledge, which one could measure in concepts per role. However, I believe that it is not the size of a group or role that is the best indicator for providing a reduction in perceived complexity, but rather the affinity that a receiver who promises to *use* this role's defining pattern feels for the concept. In other words, how well does a user identify with the pattern?

When things combine to play a collaborative role, a single recipient can use the promise as an entity, i.e. a 'black box'— and experience a cognitive simplification. The cost of understanding this functional collaboration is greatly reduced.

The important point here, as we see repeatedly in this essay, is that it is the way that these terms are perceived by the user, i.e. the *usage* (not the definition) of these terms that is the crucial element here. If I talk about 'Mr Green' and understand this usage as a description of the man's colour, and you interpret it to be a name, the intended implication will not be passed on. We therefore require a binding of terms or ontologies between different agents who engage in knowledge interactions.

**Definition 19** (Knowledge bindings). *A binding occurs between the author of knowledge and the recipient when there is something in common between their promised interpretations. If the recipient's promised understanding of concepts about the original intention does not overlap at all with the meaning promised by the author, then nothing will be communicated.*

The interpretation of knowledge made by a recipient clearly depends formally on both the original promised usage of terms by the author/promiser of the knowledge and whatever terms the recipient has promised to accept.

It follows straightforwardly from promise theory that what is offered is only a pre-requisite. It is what is *accepted* or used by agents that is important. Clearly, it is true that knowledge that no one accepts or uses is valueless.

## 12. Making usage the knowledge enabler

How might we use this cost reduction approach by grouping things effectively for users? One approach might be to simply write down items, as suggested before, and wait for natural social processes to normalize usage into common knowledge, but this can take significant time and can lead to an explosion of new items. One would likely need to go back and re-organize the stored information later to eliminate redundancy. All this accomplishes, however, is to delay the inevitable cost of organizing the information in the first place.

While there might be optimum ways of approaching categorizations that reduce the cost of knowing everything, this suggests that there is an intrinsic cost to knowing something that is associated with the what the receiver of the knowledge has decided is an acceptable set of category bundles. This is such an important observation, it is worth turning it into a hypothesis

**Hypothesis 5** (Knowledge has a minimum cost). *There is an intrinsic minimum cost to comprehending information that depends on the complexity of the model used to interpret information by the recipient, i.e. the complexity of the recipient's use-promise.*

This hypothesis harks of Shannon's entropy theorem for intrinsic information, and is almost certainly related through the definition of modelling 'alphabets', see Burgess (2004); Shannon & Weaver (1949). As Einstein remarked: everything should be made as simple as possible but

no simpler. The subjectivity of this intrinsic cost might also explain why some people find it harder than others to learn or accept knowledge from certain sources than others. The economic perspective we are pursuing here suggests a simple strategy for reducing personal cost by an end user that has to do with short-circuiting others' predefined or authoritative categories by recategorizing everything in his or her own set.

**Hypothesis 6** (Personal simplification strategy). *Each individual student or recipient of knowledge begins by remapping apparent categories of information used by the source into a personal reduced set of trusted categories, according to their own world view and experience. In this way the cost of lookup, mistrust and unfamiliarity is reduced.*

In modelling terms, we can imagine forming usage-categories called, say, 'virtual bundles of knowledge promises', i.e. virtual roles for the things a user promises to accept, which any knowledge agent is free to edit and manipulate as it sees fit.

More work will be needed to identify what the optimum approach might be in certain circumstances, and this could depend on a number of factors, so I shall leave the subject dangling on this point, as an opportunity for future work.

### 13. How can we be certain about meaning?

If an ontology is not determined by a standardizing authority, how can we be certain that anyone will end up understanding each other? The story of the Tower of Babel comes to mind, as one advocates tearing down the standard ontologies. In fact, I believe that the underpinning of knowledge by these spanning trees is entirely unnecessary. It is rather up to each and every user to apply such a tree as a filter if they so desire.

What the promise model underlines is that every agent individually promises only its own intended meaning, and in fact no two agents can truly know if they mean the same thing. Rather than seeing this as a problem to be forced into submission, it is better to accept this as the nature of reality and deal with the uncertainty. Only an independent third party can determine whether or not they *seem* to agree for all intents and purposes. The frequency of use will determine how stable word usage is. Note that the irregular verbs are those that are most frequently used. Less well used words tend to be normalized into common patterns quickly to reduce the cost of recall.

The main difference in the emergent approach is the distribution of cost. For the authoritative ontology, the up-front cost of contribution and usage is high, and it assumes expert knowledge. For the emergent context approach, there is no initial cost, but rather one must promise to practice over time to retain meaning. The advantage of a purely linguistic classification is that it is not a separate rehearsal from daily usage. We have little choice but to practice language, so in some ways the overhead is gratis, or at least can be 'charged to a different account'.

### 14. Curiosity, inspiration, learning and understanding – narratives

Turning to the future now, the promise model has some other nice features. Often, we claim to have understood something when we are happy to stop looking for further explanation. This usually happens when we are able to construct a satisfactory story or explanation about it. These stories or explanations stop in often arbitrary places – they are more about satiating our curiosity to some level of satisfaction than total revelation. Humans evolved language

through story telling. It is entirely possible that our brains are wired to support this form of narrative.

Consider what might happen if we looked up Einstein's famous equation  $E = mc^2$  in a knowledge base. This simple looking equation is associated with a huge amount of popular culture about Einstein, and most people would immediately think of him, but there is nothing unique or 'copyrighted' about it. A nutrition expert who had never read any physics might use this formula for something quite different, such as "Eating is munching and chewing twice".

Still, one context dominates above all others and that is physics. Under this context, there are many associated ideas. When we think of  $E = mc^2$ , we might associate it with the atomic bomb, or nuclear power, or mass-energy conversion, or with a funny photograph of Einstein pulling tongues at the camera.

In a book one could integrate all of these apparently unrelated meanings from cover to cover, weaving them into a story, with a progressing storyline that explains, organizes and blazes a repeatable trail for all these ideas, or one could go directly to look up keywords in an index. The table of contents in a book spans the highlights of the story, whereas the index is a quite different covering of the subjects that pays little attention to the ordering or developments in narrative, or the totality of the theme in the book.

The concept of a story is large in human culture, but as far as I can tell very little attention has been given to them in Knowledge Management research. In work with Alva Couch, I have tried to remedy this by exploring a simple notion of stories, including the notion of automated storytelling, by identifying causal associations between topics, see Burgess (2009); Couch & Burgess (2009; 2010). A table of contents, in a book, is a rough outline of the story told by the book at a very high level. It gives a different perspective on a book's content than the index does (yet the index is what current KM technology is almost exclusively focused on). Knowledge technology needs to support the idea of storylines, in which ideas and information build upon the context of earlier information, because this is how humans communicate, see Wolf (2007).

**Definition 20 (Story).** *A collection of topics connected together by associations in a causative thread.*

Causality (i.e. cause and effect) can be embodied in associative relationships such as 'affects' or 'always happens before', 'is a part of' etc. These relationships have a transitivity that most promised associations do not have, and this property allows a kind of automated reasoning that is not possible with arbitrary associations.

Automated story generation has been discussed by myself and Alva Couch in , see Couch & Burgess (2009; 2010), so I will not repeat the detailed arguments here. Today, there are no semantic knowledge models that are able to model creative narratives by association, or even ordered tables of contents in books for that matter! This is an extraordinary omission and a key capability in integrating random access knowledge with documents.

It is worth studying this possibility to derive new and 'unknown' stories from emergent repositories of knowledge promises. In this way, one could imagine discovering a new story about  $E = mc^2$  that has never been told before, derived perhaps from the contributions of a swarm of twenty different individuals who were not even thinking about this matter.

Understanding more about the principles of story detection could also have more far-reaching consequences for knowledge than just automated reasoning. In school, not all students find it easy to make their own stories from bare facts, and this could be why some students do better than others in their understanding. We tend to feel we understand something when we can

tell a convincing story about it. With more formal principles behind an effort to understand stories, technology could help struggling students to grasp connections better, and one could imagine a training program to help basic literacy skills.

We don't have time to explore all these details in this essay, except to point to this as a promising (pun intended) area of research for the future. I think that storylines combined with a linguistic approach to ontology can go a long way to addressing the deficiencies of today's web ontology methods.

## 15. Conclusions and challenges

There are many questions pending after this short essay, too many to provide many plausible answers to even a few; this is both regrettable and very exciting. The potential for future work is large, and the aim here has rather been to highlight the challenges ahead. I believe that current research has to some extent lost its way in trying to impose the intricacies of formal ontologies, and that instead of designing new standards of representation for descriptive logics, we need to re-examine some basic ideas about the aims of Knowledge Management and its relationship to pedagogy.

With a few modifications, we abandon the idea of searching for the 'right place' to put topic information, and put it anywhere with some context labels. The promises will then self-organize into an emergent pattern. Users could also use the idea of storytelling to organize their knowledge:

- Write down topic atoms and try to combine them into a network of associations.
- Write down complete stories and break them down into simplified narratives that can be represented as milestone topics linked by association.
- Start by defining a hierarchy of categories and sub-categories. This is still possible, but not recommended.

For the future we still have to demonstrate the validity of this approach:

1. Prove the independent compatibility of multiple ontologies, using promise theory.
2. Show how different ontological viewpoints can emerge from groups and norms/swarming.
3. What is the process for refactoring and establishing norms?

Teaching skills are bound to become a more valuable resource as society enters a truly knowledge oriented era. Without these, we might have massive data storage, but all of it will be largely wasted. An important aspect of learning is selective forgetting, and one can question the wisdom of stockpiling information forever. But as humans we tend to hold on to the the past somewhat irrationally.

In this essay, I have proposed using a model for knowledge engineering based on autonomous cooperation, as a way of working around the modelling errors of hierarchical categorization most commonly used today. By stripping away unnecessary structure, a promise approach to knowledge grants knowledge the freedom for it to evolve in a direction dictated by common collaborative culture, see Ostrom (1990). I believe that organized ontology has a limited usefulness, in much too specialized circumstances to be generally useful, and that we must return to a simpler linguistic approach to documenting ontologies.

Three important focus areas come to mind to explore further:

- The neglected role of narratives and storylines in knowledge representation.



- The role of repeated interaction in cementing normative language.
- The economics in the dynamics of agreement.

The tension between the desire to hierarchically divide and conquer subjects and the freedom to develop storylines unconstrained is likely to haunt knowledge management for many years to come. In the writing of this essay, for instance, I have striven to seek a balance between serving two masters: to organize things into a simple hierarchy of sections (for later 'dipping into the story' i.e. for reference), while at the same time recognizing that the whole narrative much be readable from start to finish. It is through the storyline that the illusion of understanding is most likely to emerge, because there we control the context of information from moment to moment. A novel never has to satisfy the former constraint, and is therefore a purer form of writing.

The likelihood that we will ever unify meaning into a single, standard, crystalline tree of concepts is about the same as the likelihood of unifying all the world's cultures into one. The evidence from social networking, see Newman (2003); Watts (1999), suggests that the human desire for social interaction evens out and normalizes: like swarming behaviour, we follow involuntarily the influences of others, and this leads to a condensation that has manageable proportions.

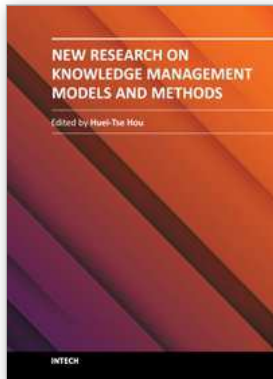
The final answers about knowledge management lie probably with social anthropology. It will be a challenge for more empirical studies to come up with evidence for the success or failure of the suggestions contained here. In the mean time, there seems to be little to lose by trying a promise approach, so I leave it to readers to explore these simple guidelines in practice.

## 16. References

- Albert, R. & Barabási, A. (2002). Statistical mechanics of complex networks, *Reviews of Modern Physics* 74: 47.
- Barabási, A. (2002). *Linked*, (Perseus, Cambridge, Massachusetts).
- Bergstra, J. & Burgess, M. (1994-now). *An engineering approach to promises*, work in progress.
- Bergstra, J. & Burgess, M. (2006). Local and global trust based on the concept of promises, <http://arxiv.org/abs/0912.4637>.
- Bonabeau, E., Dorigo, M. & Theraulaz, G. (1999). *Swarm Intelligence: From Natural to Artificial Systems*, Oxford University Press, Oxford.
- Burgess, M. (2004). *Analytical Network and System Administration — Managing Human-Computer Systems*, J. Wiley & Sons, Chichester.
- Burgess, M. (2005). An approach to understanding policy based on autonomy and voluntary cooperation, *IFIP/IEEE 16th international workshop on distributed systems operations and management (DSOM)*, in LNCS 3775, pp. 97–108.
- Burgess, M. (2009). Knowledge management and promises, *Lecture Notes on Computer Science* 5637: 95–107.
- Burgess, M. & Fagernes, S. (2007a). Laws of systemic organization and collective behaviour in ensembles, *Proceedings of MACE 2007*, Vol. 6 of *Multicon Lecture Notes*, Multicon Verlag.
- Burgess, M. & Fagernes, S. (2007b). Norms and swarms, *Lecture Notes on Computer Science* 4543 (Proceedings of the first International Conference on Autonomous Infrastructure and Security (AIMS)): 107–118.
- Burgess, M. & Fagernes, S. (n.d.). Voluntary economic cooperation in policy based management, *IEEE Transactions on Network and Service Management* p. (submitted).



- Couch, A. & Burgess, M. (2009). Compass and direction in topic maps, (*Oslo University College preprint*).
- Couch, A. & Burgess, M. (2010). Troubleshooting with human-readable automated reasoning, *Proceedings of the Twenty-Fourth Systems Administration Conference (LISA XXIV) (USENIX Association: Berkeley, CA)*, p. (unnumbered pages).
- Deutsche, G. (2006). *The unfolding of language*, Arrow.
- Dicheva, D. & Dichev, C. (2005). Authoring educational topic maps: Can we make it easier?, *Proceedings of Fifth IEEE International Conference on Advanced Learning Technologies (ICALT'05)*, ISBN: 0-7695-2338-2.
- Dietz, J. (2006). *Enterprise ontology*, Springer.
- Dunbar, R. (1996). *Grooming, Gossip and the Evolution of Language*, Faber and Faber, London.
- et al., F. B. (ed.) (2003). *The Description Logic Handbook*, Cambridge.
- Hicks, R., Dattero, R. & Galup, S. D. (2006). The five-tier knowledge management hierarchy, *Journal of Knowledge Management* 10: 19–31.
- Kipp, N. (2003). A mathematical formalism for the topic maps reference model, *Technical report*.
- Moench, E., Ullrich, M., Schnurr, H. & Angele, J. (2003). Semanticminer – ontology-based knowledge retrieval, *Journal of Universal Computer Science* 9(7): 682–696.
- Moore, G. (2001). Rdf and topicmaps: An exercise in convergence, *Technical report*, XML Europe.
- Newman, M. (2003). The structure and function of complex networks, *SIAM Review* 45: 167–256.
- Newman, M. E. J., Strogatz, S. & Watts, D. (2001). Random graphs with arbitrary degree distributions and their applications, *Physical Review E* 64: 026118.
- of Government Commerce, O. (ed.) (2000). *Best Practice for Service Support*, ITIL: The Key to Managing IT Services, The Stationary Office, London.
- Ostrom, E. (1990). *Governing the Commons*, Cambridge.
- Pepper, S. (2009). *Encyclopedia of Library and Information Sciences*, CRC Press, ISBN 9780849397127, chapter Topic Maps.
- Shannon, C. & Weaver, W. (1949). *The mathematical theory of communication*, University of Illinois Press, Urbana.
- Strassner, J. (2007). *Handbook of Network and System Administration*, Elsevier Handbook, chapter Knowledge Engineering Using Ontologies.
- W3C (n.d.). Defining n-ary relations on the semantic web: use with individuals. <http://www.w3.org/TR/2004/WD-swbp-n-aryRelations-20040721/>.
- Watts, D. (1999). *Small Worlds*, (Princeton University Press, Princeton).
- Wolf, M. (2007). *Proust and the Squid*, Harper Collins.
- Zhou, W., Sornette, S., Hill, R. & Dunbar, R. (2004). Discrete hierarchical organization of social group sizes, *Proc. Royal Soc.* 272: 439–444.



## **New Research on Knowledge Management Models and Methods**

Edited by Prof. Huei Tse Hou

ISBN 978-953-51-0190-1

Hard cover, 426 pages

**Publisher** InTech

**Published online** 23, March, 2012

**Published in print edition** March, 2012

Due to the development of mobile and Web 2.0 technology, knowledge transfer, storage and retrieval have become much more rapid. In recent years, there have been more and more new and interesting findings in the research field of knowledge management. This book aims to introduce readers to the recent research topics, it is titled "New Research on Knowledge Management Models and Methods" and includes 19 chapters. Its focus is on the exploration of methods and models, covering the innovations of all knowledge management models and methods as well as deeper discussion. It is expected that this book provides relevant information about new research trends in comprehensive and novel knowledge management studies, and that it serves as an important resource for researchers, teachers and students, and for the development of practices in the knowledge management field.

### **How to reference**

In order to correctly reference this scholarly work, feel free to copy and paste the following:

Mark Burgess (2012). What's Wrong with Knowledge Management? And the Emergence of Ontology, New Research on Knowledge Management Models and Methods, Prof. Huei Tse Hou (Ed.), ISBN: 978-953-51-0190-1, InTech, Available from: <http://www.intechopen.com/books/new-research-on-knowledge-management-models-and-methods/principles-of-knowledge-management-emergent-ontology-and-spanning-trees-in-knowledge-representation>

**INTECH**  
open science | open minds

### **InTech Europe**

University Campus STeP Ri  
Slavka Krautzeka 83/A  
51000 Rijeka, Croatia  
Phone: +385 (51) 770 447  
Fax: +385 (51) 686 166  
[www.intechopen.com](http://www.intechopen.com)

### **InTech China**

Unit 405, Office Block, Hotel Equatorial Shanghai  
No.65, Yan An Road (West), Shanghai, 200040, China  
中国上海市延安西路65号上海国际贵都大饭店办公楼405单元  
Phone: +86-21-62489820  
Fax: +86-21-62489821