Pigott, D.J. and Hobbs, V.J. (2011) Complex knowledge modelling with functional entity relationship diagrams. Vine, 31 (2). pp. 192-211.

Complex Knowledge Modelling with Functional Entity Relationship Diagrams

Diarmuid J Pigott
Murdoch University
Murdoch, Western Australia
Telephone: 9360 2521
Email: d.pigott@murdoch.edu.au

Valerie J Hobbs
Murdoch University
Murdoch, Western Australia
Telephone: 9360 2817
Email: v.hobbs@murdoch.edu.au

Abstract

Modelling complex knowledge resources can be problematical as there is currently no formalism that can represent the nature of the data-seeking process at a conceptual level. We introduce the functional entity (FE), an encapsulated data resource that acts as a question-answering system, and identify nine different functional entities based on three main types of question-answer entailment: instance-dominant, value-dominant, and connection-dominant. We use functional entities to develop a generalisation of the Entity-Relationship Diagram (ERD), the Functional-Entity Relationship Diagram (FERD), which can be used for high level conceptual modelling of heterogeneous knowledge systems. We further describe extensions to the FERD that permit modelling of resources that cannot be represented by traditional propositional form, and extensions relevant to large-scale documentation.

Keywords

Functional entity, functional-entity relationship diagram, question-answering system, sketch logic, conceptual knowledge modelling, non-Aristotelian

1. Introduction

Any knowledge management system relies ultimately on the timely and accurate retrieval of appropriate facts, and self-evidently facts come in many different forms. They have different structures; they vary in terms of certainty, reliability, applicability, and accessibility; they may be located within the enterprise's own data and information management systems, in external systems and libraries, or embedded in human expertise. Designing and building a knowledge management system involves ensuring that the right facts can be called upon to answer the question at hand, and coordinating a number of disparate resources.

The problem facing the designer is that the same material will be required to provide different functions, yield different facts, and be subjected to different methodologies. On the other hand a single knowledge seeking mechanism may draw on material owned by different groups, updated with different frequencies, and funded in different manners. As illustration, Brilliant (1988) and Bearman (1988) separately showed how the same information in an art historical information resource would show value to insurers, range to a curator, examples to an artist, size and shape to removalists, and the opinions of rivals to an art historian. O'Sullivan and Unwin (2003) discussed the situation in which the same details stored by different owners – the geographical information for a rural district, maintained by a council and a bus company – would provide information on surfaces and potential conflicts with other agencies (telecoms and gas) to the council, while it would provide information on routes and demographics for timetabling to a bus company.

We can see from both of these examples that one single source of material lends itself to multiple use and interpretation, and one system of use and interpretation can rely on multiple sources and ownerships (hegemonies). Every new observer or questioner of a system will compound the problem, and there is no guarantee of stability. When we view a knowledge system as a communicative process (Walsham, 2005) (embedded in both the understanding and expectations of the practitioners and the mechanisms being created to meet those expectations) we can model the system at the teleological level, aggregating the needs that can be anticipated, and modelling the entirety as a series of *questions* that are going to be asked of such a system when complete. In other words, when we model a system for knowledge retrieval we have to model the flow of questions and answers that exist within that system. By modelling the questions and answers a system needs to provide, we can plan allocation of question-answering resources – we can delegate to different infrastructures the questions that are best suited to it, including outsourcing complex queries or work out what is best suited to a reference librarian or a consultative expert.

2. Question-Answering Systems

Question-Answering (QA) has existed as a rival to the signal-processing tradition of knowledge pioneered by Shannon and Wiener from the origins of modern computing (McCulloch, 1974). At the 1951 Macy Conference (Pias, 2003) Mackay, representing an English school of cybernetics stemming from the work of Mackay, Gabor and Cherry (in the tradition of Fisher, Wittgenstein and Pearson), proposed a operational view of information: "information is that which logically enables the receiver to make or add to a representation of that which is the case, or which is believed or alleged to be the case"(MacKay, 1951). In "What makes a question?" MacKay (MacKay, 1960) proposed that in addition to Shannon's intentionally context-free conception of information (which he termed selective information), there was a contextualised version that was conceived of as a response to the need to acquire information (structural information). Moreover, there was a third role of validating information that can only be conceived of as a response to a question.

The question-answering paradigm was influential at the simple database system level at the start of semantic information retrieval: see Green (1961), Marill (1962 cited by Minsky, 1968 and especially Black, 1968) and Robinson (1965). This research was based on the logic of Quine (1959) and considered what "amounted to" a satisfactory answer to a given question. A more mature version of the logical paradigm, erotetic logic (developed by Harrah, 1961) gives us a richer picture of this question-satisfying. In this paper we use Harrah's erotetic logic to develop an abstraction of the information-seeking processes in complex distributed knowledge systems, whereby the client-server process is envisaged as a series of questions and answers.

QA systems are useful as models because they permit partial and incomplete answers, as well as the modelling of nonsensical answers, when the question is insufficient or when the answer is vague. They also help modelling of questions that aren't possible with a current KM system (or even with the current state of the art), but which could be provided by an enquiry of human resources or generalised expertise (e.g. in a library). A QA system also permits us to reserve a portion of the role for the enquirer in interpretation – we can't assume that the details that are delivered by the tuple returned are going to necessarily provide the final answer – it may require reprocessing by another system, or combination with other answers to make up an answer in a hypersystem. What we are modelling is the entirety of the knowledge resources of an enterprise, not just the portion of it that is computerised, let alone encoded and stored in a database.

What this means is that the distinction between the *symbol level* and the *knowledge level* established by Newell (Newell, 1993, Newell, 1981) has to be reflected in both the design process, and the tools used to represent the model. The dynamic entities that make up the question and answering entities can not be bound by the symbol level of the knowledge system, but will be invoked through either ad hoc or established knowledge representation mechanisms. We have already explored this emergence mechanism in a previous paper describing *just-below-the-surface* systems (Pigott et al., 2004), which belong in the symbol level, yet are designed to permit the emergence of epiphenomenal views of the knowledge, typically through frames, databases, or spreadsheets. The same stored values and rules can give rise to different knowledge epiphenomena, and typically such an epiphenomenon will call on more than one set of values within the symbol level. Just as the abstraction of the database conceived of by Codd was a series of *relations* epiphenomenal to but inherent in the values (Codd, 1969), so we must look for a way of describing *knowledge relations*, dwelling in the knowledge level while being epiphenomenal to but inherent in the symbol level[1].

Iverson's Turing lecture ''Notation as a tool of thought '' (Iverson, 1980) stresses the explorative nature of conceptual-level problem-solving, and the role that notation plays in that iterative process. Mathematical, geometric and algebraic systems are perhaps the most abstract of notation systems, but the notational symbologies of chemists, physicists or meteorologists play just as important a role in the stages of their thought development. To properly plan and monitor our question-answering system, therefore, we need a formalism that enables us to manipulate the system at the highest possible level.

In addition to a claim of increased facility given by notation, there is a parallel development in arguing for a logic of graphical signs. In additional to the standard proof from written symbol ("string logic") Wells (Wells 1984) amongst others has drawn on category theory to propose a logic of graphical representation ("sketch logic") to create proofs and demonstration in purely graphical form. This sketch logic has been used to justify many of the standard formalisms of IS including Entity-Relationship Diagrams (ERDs) (Dampney and Johnson, 1995, Diskin and Kadish, 1997, Diskin et al., 2000, Johnson et al., 2002) and UML (Dingel et al., 2008, Diskin, 2005), while Ruqian Lu (2004, 2005) and Colomb et al (Colomb and Dampney, 2005, Colomb et al., 2001) have shown how *typed* categorical frameworks are a unifying explicative framework for the disparate elements involved in knowledge modelling. Being able to explore question-and-answering systems in a categorically justified diagramming system provides an additional mode of proof to the designer.

We require a generalisation of *types* of the questions to be asked, with a matching generalisation of the type of answer available. What is needed is a conceptual modelling tool that permits the types of fact retrieval operations, the entities that can be seen (in set terms) as representing the replies, and the existential and quantitative qualities they have. This is not modelling at a software or product level, but modelling in terms of how the system as a whole responds to requests made of it.

In Pigott and Hobbs (2009) we introduced such a knowledge representation formalism, the Functional-Entity Relationship Diagram (FERD), which comprises a set of extensions to the industry standard Entity-Relationship Diagram (ERD) established by Chen (1976a, 1976b). The diagrams (Functional-Entity Relationship Diagrams, or FERDs) consist of nodes representing these functional entities, edges representing the relationships between them, and modified heads and tails of these edges to indicate the type of functional entity. The formalism is compatible with a standard ERD. Where the additional features are not required, standard ERD representation is used. We review and extend the FERD formalism in the next sections.

3. The Functional Entity Formalism

Robinson (1965) stated that "the central problem of fact retrieval is: Given an interrogative sentence, how does one recognize a matching sentence that supplies an answer?" Her suggestion is a degree of commonality between them: a common term in their formation.

Robinson's simplest form of the question "What is B?" has the answer, "A is B" (where A is the subject and B is object), which can be generalised as the functional relationship F(A,B) where the function F is either the simple copula "is" or a more complex attribution. The questions we ask contain constraints that *automatically entail* a series of instances based on these commonalities. We can see that there are three possible variations on this simple statement:

[1] Fox (1987) identifies a higher level that the knowledge level in enterprise-wide KM: a need for agent-oriented coordination requires an understanding at an *organisation level*, situated above the individual/group level proposed by Newell. This identification of a potential higher level is in accord with Walsham's position, and with the FERD approach identified in this paper. (Fox, M. 1987. Beyond the knowledge level. *In:* KERSCHBERG, L. (ed.) *Proceedings of the First International Conference on Expert Database Systems*. Menlo Park, CA: Benjamin/Cummings Pub. Co.)

1) F(A,*x*) - which translates as "what values *x* do we find as F-values for A?" (which is generally met as the derived forms "Is there an A with an F-value of *x*′ ?" or "What A′ exist such that their F-value is *x*′ ?")
2) F(*x*,B) - which translates as "what *x* exists such that it has an F-value of B?"
3) *x* (A,B) - which translates as "is there a function *x* such that it gives B for A?" (generally met with as the derived form "Is there a link *x*′ between A and B?")

Establishing a typology of the commonalities, we can distinguish:

1) Instance-dominant entailment, wherein the *instances* entail the answer. Here there is a direct and unambiguous entailment of instances based on a specification of their attributes. This is the data modelling system present in the conventional ERD. We model such QA systems with Type 1 functional entities.
2) Value-dominant entailment, wherein the *values* entail the answer. This is where we are seeking values that match the question, but do not necessarily have a clear indication of what entities may be entailed. We model such QA systems as Type 2 functional entities.
3) Linkage-dominant entailment[2], wherein the *linkages* entail the answer. This can be either where we know two things exists and we are trying to find what links them, or where we know what connections to look for, but don't know what the linked instances are. We model such QA systems as Type 3 functional entities.

It is important to note that there can inherently be only three types of entailments. While analysis will present entailments that appear to be more complex, the algebra of argument will show that they reduce to two or more instances of these three types, either recursively the through the substitution of a new constraint set for the entailment, or else combined through the set-based operations on union, intersection or disjunction on the individual entailments. Given what is known of the knowledge level (Rosenbloom et al., 1989), such a recursive web of entailments is what would be expected.

We can see how this typification satisfies the needs of a typed category-theoretical system. A category comprises a domain, a codomain and a mapping function (Fig X). For our typed entailments, we can see a generalised category for Newell's symbol layer: Domain ⇨instance, Codomain ⇨value, Function ⇨linkage; and for the entailments acting as constraints on each of these components of the symbol layer, we can have a typed categorical representation.

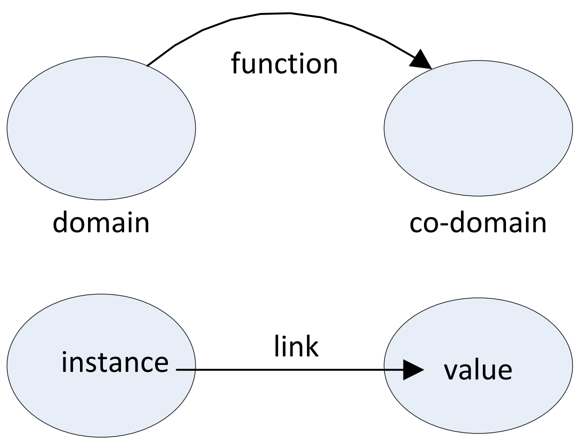

Figure 1. Instance, link and value as a typed category-theoretical system. Above: the generalized category; below, the three categorical components of symbol layer propositions.

[2] In an earlier paper we used the term *connection*. We now use the term linkage to avoid the prior associations in KM and cognitive science of connection and similar words. We thank John Gammack for the suggestion.

This typification is also what is to be expected given our earlier work on how the stuff of knowledge – *noetica* – is organised in both thought and automation of thought (Pigott et al., 2002, Pigott and Hobbs, 2001). The standardised answers proposed by Robinson above ("A is B") are what are termed *noetic simples*, which are combined with each other to make richer and more complex representations of the world. The noetic simples can be organised according to three distinct principles:

1) *Shape*: Alignment resulting from commonality of structure and domain, leading to regularisation of the noetica
2) *Granularity*: Clustering resulting from commonality of values and value applicability, leading to aggregation
3) *Scope*: Interrelation resulting from commonality via interconnected networks, leading to contextualisation

These three principles form the vertices of a 4 dimensional conceptual space termed the *noetic prism*.

Again it is important to note that there can inherently be only three kinds of organising principle, with any apparently more complex organisation being reducible to a vector sum of two of more of these. Higher order organising structures within the noetica are created through an ad hoc process of interaction, or prepared in advance to facilitate interaction, as found in the archetypal typed knowledge resources of databases, spreadsheets and frames which embody greater *scope, granularity* or *shape* respectively (Pigott et al., 2004).

We can therefore see that the three forms of entailment match up with three organising principles and three categorical components:

- Instance-dominant entailment is informed by shape, found in constraining the category domain
- Value-dominant entailment is informed by granularity, found in constraining the category codomain
- Link-dominant entailment is informed by scope, found in constraining the category mapping function

The answer entailed by any knowledge-seeking question is a *knowledge relation*, in a form similar to Codd's relations. Like Codd's relations, knowledge relations are sets of stored attribute-value pairs; however, knowledge relations are unlike Codd's abstractions in that they are polysemous: they will always participate in multiple roles within the knowledge base, so cannot have a single formalised existence. Knowledge relations are always automatically entailed (for the universe of discourse) in a logical sense at the moment of formulation. As such knowledge relations are always implicit in the data set, even if only as an empty relation.

Moreover, while relations are array-form, knowledge relations ADT representation will vary, inasmuch as they are representing the found instances, which have differing representational requirements. The difficulty comes in planning the mechanism for providing the populated or empty ADT that satisfices the answer.

Formally, with knowledge-seeking questions, what is passed from the inquiring system to the responding system is a *question mode* (detailing which question type is being given) and a *key* (which is a pair-tuple of the two givens for the question). What is returned (the answer) is a knowledge relation, which is always delivered in the context of the mode + key combination.

There will always be a knowledge relation present for any given key, contextualised by that key. However, there will also always a generic class of keys of similar form. The abstraction that is the mode, generalised key, and generalised responsive ADT is what is termed the *Functional Entity*.

3.1 Deriving the Functional Entity typology

We can now examine how question-answering systems can be considered according to these types of entailment and the organising principles present in the noetica. Since answers are a collection of "A is B" statements, with a declarative nature only to be found in the knowledge level, the onus is on the responding system to create such rich collections, and we can see that posing a question to the noetica, and gaining a meaningful answer, will increase the extent and significance of the vertex that most matches the question type, enriching the knowledge resources correspondingly.

In answering a question within a knowledge system, this process of complexification takes place reactively when the erotetics process commences. This is enabled either through transient structures or making use of systems that have been prepared. Since such systems have been found to have one vertex dominant in practice, with the other two playing a supportive role, then it is to be expected that the answering component of a question–answering system will be similarly comprised. This also applies to further stages in the question and answer processes such as subsequent confirmation (Bromberger, 1966), teachback/entailment (Pask et al., 1973) , or cascading questions (Graesser et al., 1992, Lauer, 2001).

Thus, there is a question-answering process which is informed significantly for each one of these ordering principles for each one of these question types, which results in a 3 × 3 categorisation scheme (Figure 2).

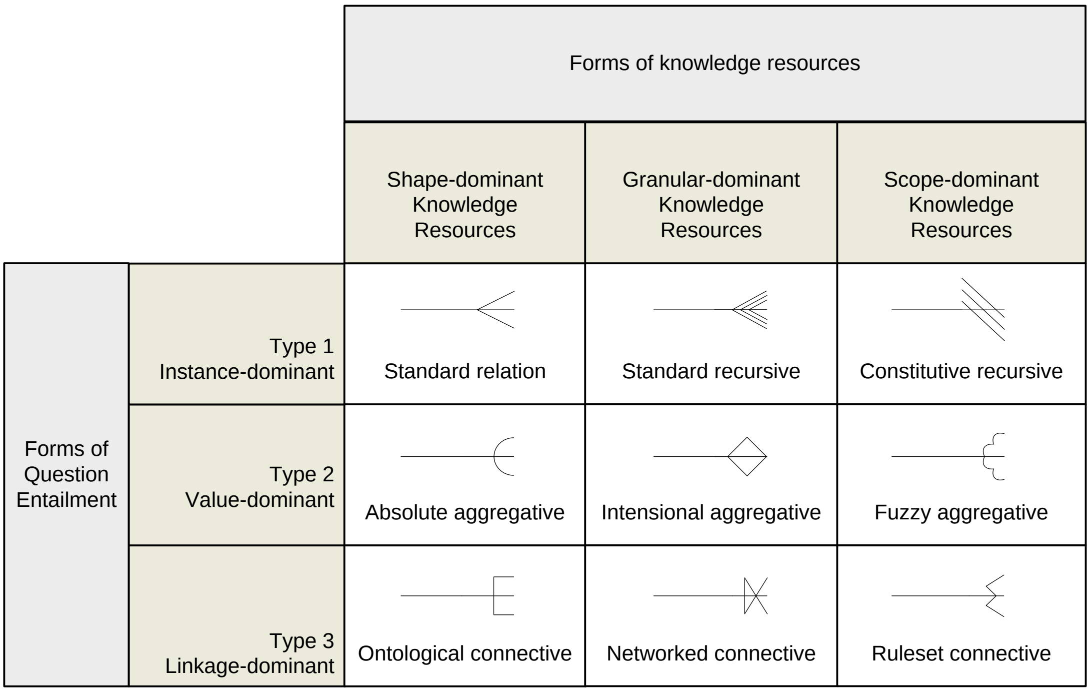

Figure 2. A typology of knowledge relations formed by the interaction of 3 types of question entailment and 3 types of knowledge resource, and the associated Functional Entity symbol set.

This matrix permits a categorisation of the knowledge relations that we find in knowledge repositories. The 9 resulting knowledge relations, and the functional entities they are represented by, will be explored below, but for the moment, we can consider some generalisations as to how the two axes determine the knowledge relation types.

The *rows* in Figure 2 determine how the enquiring systems come to ask the questions. This is a concomitant of the principle of substitutability in the basic question-making format. Each of the knowledge relations in the row has a commonality of entailment based on their being type 1 (instance-dominant) type 2 (value dominant) or type 3 (linkage-dominant), described next.

- All Type 1 questions (instance-dominant) seek potential set-membership within acknowledged sets via directly-matching values in prepared knowledge structures. All present as a key an attribute for all members of the set, and seek either a confirmation of existence or a tuple representing identities and attributes, for a value or range of values.

- All Type 2 questions (value-dominant) potentially seek anything with co-extensive recorded values in the universe of discourse, regardless of set membership. (Set membership can be added as an additional constraint). The results are mediated back to instances from the discovered values before being returned. Type 2 questions present as a key any recorded attribute, but as a pair – a measurement and a frame of reference, and seek either confirmation of existence of any recorded value at that point, or potential instances of interest.

- All Type 3 questions (linkage-dominant) seek to contextualise instances or values through the discovery of interconnections to any recorded instances in the universe of discourse, regardless of connection type. (Limitations as to which links are of interest can be added as an additional constraint.) Discovered instances can be returned directly, while discovered values must be mediated to instances. Type 3 questions present as a key comprising a pair of a designator of which one of instance or value has been presented, and either an instance or a value as appropriate, and seek either confirmation of existence of any connection to key, or a graph structure containing a map of the network of discovered values.

The *columns* in Figure 2 determine how the responding systems come to present the answers. This will be through the mechanism of ad hoc higher order noetic structures, or of prepared structures that have been optimised for such questions. Each of the knowledge relations in the columns will have a commonality of structural principle because of their being based in type 1 (shape-dominant) type 2 (granularity-dominant) or type 3 (scope-dominant) structures, as described next. (Note that all noetic structures will always from first principles have all three vertices significant to some extent, it is the dominant vertex that informs the answering process.)

- All Type 1 (shape-dominant) knowledge resources have *direct* entailment from the key. All get us an answer straight away because at some stage a value has been recorded against an instance.

- All Type 2 (granular-dominant) knowledge resources have *articulated* entailment from key. All give us an answer at one remove because it is co-extensive values that are returned, and the responding system must then retrieve or construct matching instances to make those values intelligible.

- All Type 3 (scope-dominant) knowledge resources have *mediated* entailment from the key. All give us an answer by starting with values or instances and locate connections of significance to the enquirer, applying reasons to determine membership of the answer set.

3.2 The Functional Entity Types

We now briefly describe and illustrate the nine generated functional entities in Figure 2. For a more extensive discussion of the examples in this section, and a worked case study, see Pigott and Hobbs (2009).

Type 1 Functional Entities: Instance-dominant

In all Type 1 questions there is a *direct and unambiguous entailment* of instances based on a specification of their attributes. This entailment may be immediate, or via recursion. Figure 3 illustrates the three Type 1 functional entities within a single system. This scenario shows the relationships among a car manufacture, a model of a car, a spare part for that model, and a service organisation that can fit the part.

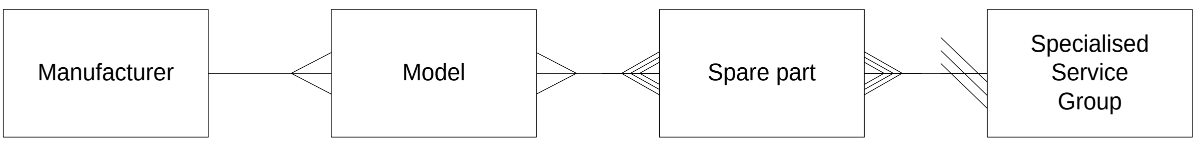

Figure 3. A FERD Modelling an automotive parts supply system using Type 1 functional entities: standard relation (manufacturer, model); standard recursive (spare part); constitutive recursive (specialised service group).

The *standard relation functional* entity uses the conventional relationship of the ERD. It presents a set of entailed instances as an answer to a query: when we ask "what records match this criterion?" we are effectively entailing the tuple that is a standard subset of the table or view. Figure 3 uses the standard relation functional entity to show the one-to-many relationship between *manufacturer* and *model*.

The *standard recursive* functional entity is used where questions are recursive in form, involving knowledge that is represented in terms of part/whole relationships. In Figure 3 the question "is this spare part available for this model?" is represented by the *Spare part* standard recursive functional entity, which shows that either the part is available, or else a subassembly exists that will contain the needed part.

The *constitutive recursive* occurs where there is a rule determining the links between parent and child instances, and the nature of the relationships among parent and child in the hierarchy differs for each. In Figure 3 the constitutive recursive functional entity represents the question "which service company can fit the part?" Here the answer refers to expertise, which would reside in a technician, who is employed in turn by a service centre (which in turn can part of a chain or franchise, or a division of a company). The constitutive recursive functional entity *Specialised Service Group* would tell if a particular service group would be certified to fit the part depending on whether or not a branch somewhere had a qualified technician.

Generally, all of the Type 1 questions can be implemented with a relational database, although the standard recursive and constitutional recursive questions may require some kind of stored procedures to operate.

Type 2 Functional Entities: Value-dominant

All Type 2 questions (value-dominant) potentially seek anything with co-extensive recorded values in the universe of discourse, regardless of set membership. They ask: "what things have the consistent attribution of value B applicable to them?". When we explore material with Type 2 questions, we could be trying to determine causal relationships, or searching for clusters or outliers in a population, or looking for trends in a series, or even requiring prediction or extension beyond known material. This is in general an enquiry as to the import of the field at a given designation. Typically such material is investigated using spreadsheets, statistical or epidemiological databases, or GIS.

Figure 4 illustrates the three kinds of Type 2 functional entities in one scenario, which models an environmental phenomenon (such as a chemical spill) requiring an immediate response.

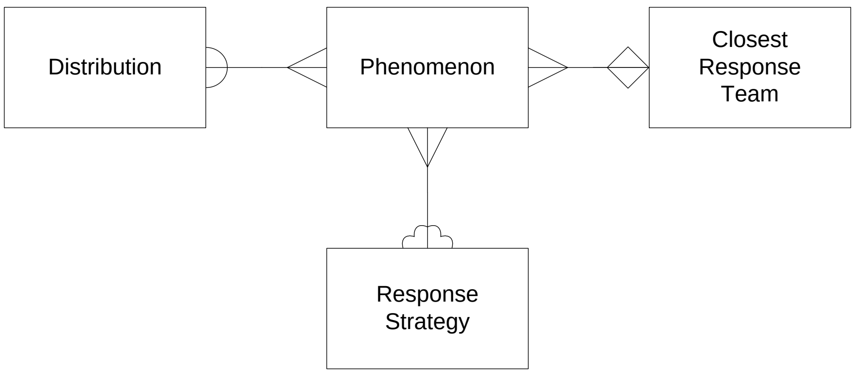

Figure 4. Modelling a chemical spill response system using Type 2 functional entities: absolute aggregative (distribution); intensional aggregative (closest response team); fuzzy aggregative (response strategy.)

The results of an *absolute aggregative* question are irrespective of position or occasion of enquirer: here, the question "where is the phenomenon distributed?" can be answered within an absolute framework such as a coordinate system, and consequently distribution is modelled as absolute aggregative functional entity.

In contrast, the result of an *intensional aggregative* question will depend on an immediate analysis of the population, relative to the problem encountered. Here "which is the closest response team?" is modelled as an intensional aggregative functional entity.

The final type 2 question, fuzzy *aggregative,* involves the use of the fuzzy logic paradigm (Zadeh, 1965, Kosko, 1993, Yen and Langari, 1999). This is where results determine if values are members of fuzzy sets, and therefore invoke a kind of rule mediation to determine what kind of phenomenon the value amounts to. In this example the question ‘'what is the appropriate response strategy?” is answered by the functional entity *response strategy* which will involve preset ranges for rapidity of spreading, age of phenomenon, area or distribution of phenomenon, commonality of phenomenon, resilience of environment to phenomenon, and so forth; fitting the values into a matrix of rules and thresholds.

Unlike the Type 1 example, the Type 2 scenario shown in Figure 4 is not straightforward to implement in a standard relational model, but will require services to be farmed out to dedicated systems. Statistical or GIS packages are frequently the only way to achieve this goal.

Type 3 Functional Entities: Link-dominant

Type 3 questions are where the either the values or the instances (or both) are determined and either or both are queried; what is sought are the other instances of the same set to which the denoted instances are connected. These are questions that ask “what things can this initial link be chain-linked to?” Such questions are found in all systems of knowledge that can be represented by a graph (i.e. networks, stars and trees). Uses range from classification schemes to family trees, from classification rules to epidemiological contact charts.

Such material is notoriously difficult to corral and control: graphs are by their very nature one of the n-P difficult problems of computer science. The rules of entailment and consistency across graphs are likewise difficult to ascertain: some systems (like an old-fashioned tree of life) can have a clear terminus by definition. Others have a practical limit of knowledge (ancestor charts for instance) since only so much is known and can be known. Others still such as contact networks are limitless, since they propagate out to unmanageable (if predicable) numbers very quickly indeed.

Figure 5 explores the three kinds of Type 3 functional entities involved in examining the advance of an infection in a population, a scenario where links between instances are the most significant knowledge relation.

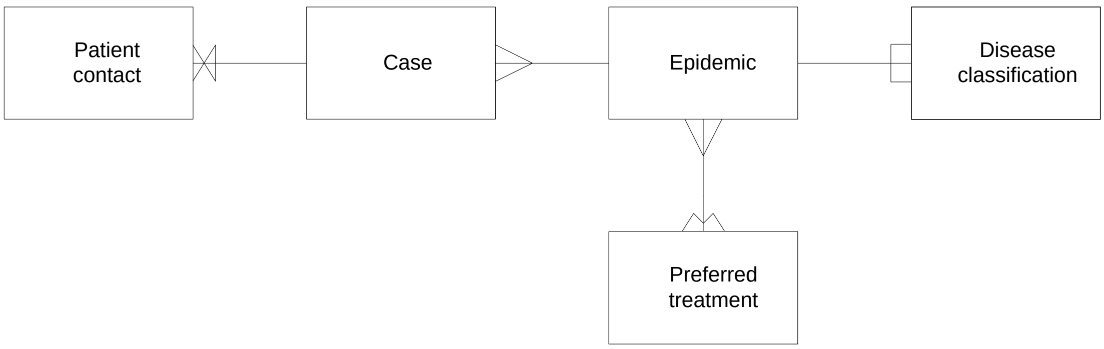

Figure 5. A FERD modelling an epidemic response system using Type 3 functional entities: networked connective (patient contact); ruleset connective (preferred treatment); ontological connective (disease classification).

Ontological connective questions are where there is a hierarchical relationship between instances based on attributes that are pre-established as significant, with predetermined methods of establishing set membership. Ontologies, subject classifications and naturalistic taxonomies are found here. In Figure 5 *classification* is represented as an ontological connective functional entity, answering the question “which WHO classification is the infection under?”

Networked connective questions are where there is an association between entities within a dataset, and the number of instances entailed by the key connection can vary to an indeterminate degree. There may be no reason for the network over and above the shared value: they are linked by a momentary shared time and space (or topological space) and that is sufficient membership for a set of answers. In Figure 5 *contact* is represented as a networked connective functional entity, addressing the question “whom has this (infected) person contacted, and whom might they have been infected by?” which implies a network of contacts, and through those, further contacts still, back to Patient Zero.

Ruleset connective questions are where values and instances are associated by chains of logical reasoning. The answers here can be set goals, or implicated instances, or likely values: the most significant thing is that unlike the other two forms of question there is a process of reasoning before the network can be created. Expert systems (either inferential or production) can be modelled as such question systems. In the scenario of Figure 5, *preferred treatment* will have an answer based on stored medical and clinical knowledge.

4. FERD Extensions

The 9 functional entities discussed in the FERD typology above are all based on the assumption of traditional propositional form, following the ideas of standard traditional logic that explicitly underpin the relational model as formulated by Codd (Codd and Strehlo, 1990). Two factors in describing knowledge relations make this assumption problematic: the representation of unknown information, and the description of the modality (likelihood, trustworthiness, conjecture) of the knowledge relation both defy traditional propositional form.

Codd himself found that in some situations it was in fact partial or missing information that required representation, and worked around the issues that arose when he wanted to extend these capabilities to empty sets or missing values in a series of papers (Codd and Date, 1993, Codd, 1987, Codd, 1986) working within the bounds of the traditional forms (e.g. portraying NULLs as operations on values of unknown statuses).

However, these factors are crucial to defining and representing knowledge. The FERD system has to be able to represent just such details over and above the cases already outlined, and we use two extensions for this purpose, termed the *non-Aristotelian* functional entity and the *knowledge mixin*.

4.1 Non-Aristotelian Functional Entities

The traditional form of logic (Aristotelian) rested on three principles – of identity, non-contradiction and excluded middle – called the "Laws of Thought" (Boole, 1958), and we generally need to ensure that these three principles hold when we are constructing higher order noetic artefacts from knowledge repositories. But situations arise where we have to store noetic simples that seem to contradict these time-honoured principles. In a conventional situation, the process of reasoning with the existing knowledge would break down here. But knowledge workers can, and regularly do, make the best of the data to proceed to build higher order noetic structures. What is needed is a mechanism for showing how this process can be modelled and anticipated, and how it can fit into a greater FERD with essential differences showing.

Fortunately there is a tradition in logic, called non-Aristotelian logic, of examining the consequences of contradicting these laws, after the fashion of non-Euclidian geometry or non-Newtonian physics (Bradford Smith, 1919). We represent the knowledge relations based on these principles with *non-Aristotelian* (Ā) functional entities.[3]

As there are three laws of thought, so there are three functional entities concerning them. And as the laws can be mapped to the parts of the proposition (Instance, Value and Link) we can identify three Ā knowledge relations for each of the knowledge-seeking question forms. We know that the Ā knowledge relation is entailed as a set and valid for involvement in high level declarative form, but the set contains what are effectively instances of which relations can be predicated.

The following discussion of Ā functional entities draws on a case (Pigott and Mitchell, 2003) where the placement of a gas pipeline through bushland is being assessed for impact on remnant vegetation, especially with regard to native bird nesting habitats (Figure 6).

---

[3] It should be note that this approach to such propositional conflict follows in the path of both the Cybernetics and General Systems theory movements in the 1950s. Korzybski, A. 1994. *Science and sanity: An introduction to non-Aristotelian systems and general semantics*, Pias, C. (ed.) 2003. *Cybernetics - Kybernetik. The Macy-Conferences 1946-1953*, Zürich/Berlin: diaphanes, Jutoran, S. 2005. The Process from Observed Systems to Observing Systems. *School of Humanities and Social Sciences*, *Nova University*, Holl, H. 2007. Second thoughts on Gregory Bateson and Alfred Korzybski. *Kybernetes*, 36**,** 1047-1054.

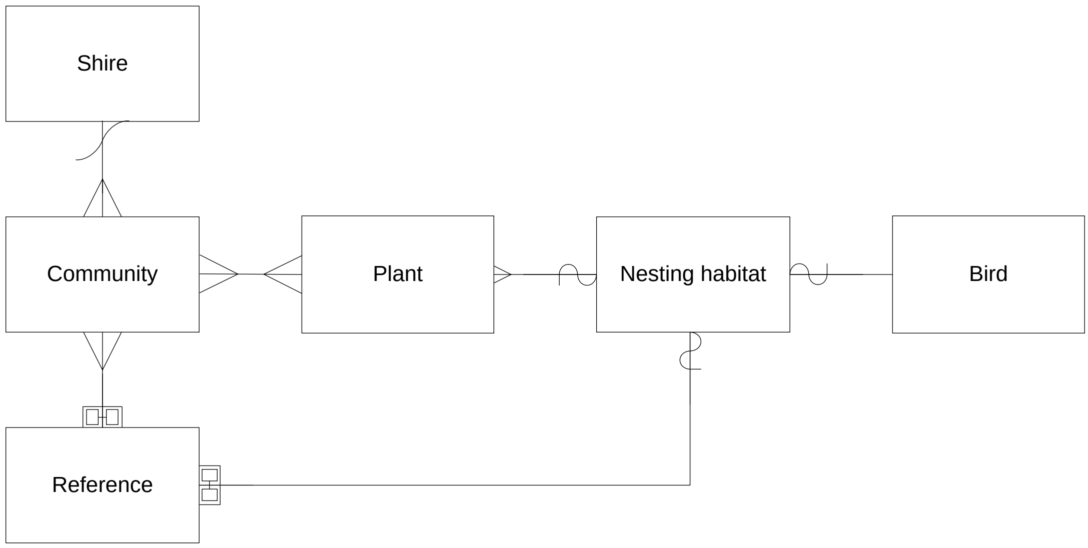

Figure 6. A FERD modelling an ecological impact study using Non-Aristotelian functional entities: contiguous Ā (shire); emergent Ā (reference), abductive Ā (nesting habitat).

Type 1 non-Aristotelian functional entities: Contiguous non-Aristotelian

The law of identity, A ≡ A, is germane to type 1 non-Aristotelian questions. It is violated when the nature of an unchangeable (an identity) is fractured or fragmented. We call these *contiguous non-Aristotelian* functional entities.

Figure 6 shows a contiguous non-Aristotelian functional entity representing *shires* for location of woodland communities suitable for bird nesting. If records are kept of observations at a shire level, and the boundaries (being political entities) change, then the fact of a hatching flock being observed in a region would have a value depending on where the boundaries were for that period of time.

Historical and legal databases have to be able to contain knowledge statements made with regard to such assumed-contiguous identities.

Type 2 non-Aristotelian functional entities: Emergent non-Aristotelian

The law of non-contradiction, ¬(P∧¬P), is problematic when we change the purpose of a recorded value, which means that such considerations as domain and range (including type, storage, keying, null-permission, choice of lookup) are lost, while new significance (comparison with other values, new null values, new key-dependence) is granted. We call these *emergent non-Aristotelian* functional entities.

Figure 6 represents the references to plant community observations and bird nesting habitats as an emergent non-Aristotelian functional entity, in the form of a *reference library* for recording accounts of observations in the literature, including scholarly articles, reports, surveys, interviews and newspaper clippings. With a bibliographic system, a repository of templates for assigning late-binding labels values to make a set of attributes. Up to the point of combination, the values have a standardized slot address but no final semantic significance. Changing the type of reference changes the meaning of the value.

Another, more general, common usage is data-mining or text-mining. It is designed to repurpose values gathered for a different context but needed in a new context. With text mining, it is more obvious that the original semantic context has gone, but with the numerical form of data, there is an equally complex knowledge context framing it all.

Type 3 non-Aristotelian functional entities: Abductive non-Aristotelian

The law of excluded middle, P ∨ ¬P, creates problems for a knowledge representation system if we need to represent the different items within the knowledge system that we are assuming (in the absence of proof) are linked in some way. Examples of such candidate links might be Graesser knowledge arcs such as *causes, implies, enables, is a member of* (Otero and Graesser, 2001). Systems designed to call on sets of candidate links are termed *abductive non-Aristotelian* functional entities.

Figure 6 uses an abductive non-Aristotelian functional entity to show the assertion links between potential nesting sites and plants that provide hollows. We have an assumed (unproven) set of links between plant communities and nesting birds, indicating nominal *habitats* which sit alongside the established links.

A broader general usage of the abductive non-Aristotelian functional entity would be to represent a call on a link-exploration system such as a Bayesian database.

4.2 Mixins

Knowledge modelling has a requirement for representing modal qualities of relations. Conventional propositional forms are either true or false, and so are insufficient for the complexities of knowledge representation. The FERD system uses *knowledge mixins* to indicate different modalities of knowing. The term mixin is used in computer science to describe a class of items whose functionality can be applied to all other objects within the domain of discourse to enhance their functionality additively (Moon, 1986).

At present we can identify five FERD mixins:

- *Order* – representing the natural sequence of recording information
- *Accuracy* – representing the scale, accuracy and precision of a relation
- *Likelihood* – representing a probability-valued relation
- *Evidentiality* – representing issues of trust, experience, consistency for a relation
- *Conjecture* – representing whether the relation is known, conjecture, or one of a set of alternate conjectures.

Mixins are represented using symbols overlaid on the box for the functional entity, which indicate the need for the mixin. The documentation of such mixins being complex, they are represented with a simple label on the diagram alone, rather than a full explanation. All mixins can be applied to all functional entities, including the standard relation of the ERD, to express complex knowledge.

4.3 Cartographic Functional Entities

Over and above the needs already described for the functional entities, there are challenges that must be met with the operation of the FERD as a system of diagramming itself.

A diagram of any complex system (a wiring diagram, a tube map, or a flowchart) has an upper limit on what can be shown. This requirement has already proved problematic with UML, and has led to the inclusion of recursion as a formal structural component of the ever-expanding UML 2.0 (Störrle, 2006, Störrle, 2001, Erickson and Siau, 2004, Siau and Tian, 2005, Zhao and Siau, 2002, Erickson and Siau, 2003).

The requirement here is for a mechanism to represent occlusion, to hide components, or to represent sections that are hidden by virtue of being outside the diagram's bounds. To that end, the FERD system has three *cartographic functional entities* whose purpose is to represent occlusion: Since there are three ways in which this can happen, there are three functional entities: the *external cartographic* functional entity for showing external systems which are not under the control of the designer, the *folded cartographic* functional entity to represent subsystems which are to be represented as a black box, and the *exofolded cartographic* functional entity to represent the supersystem which is the context of the system under design.

The example shown in Figure 7 is of a library circulation system, featuring an offsite *administration* system as an external cartographic functional entity, a *loans* subsystem as a folded cartographic functional entity, and the general *library system* of which it is a part as an exofolded functional entity.

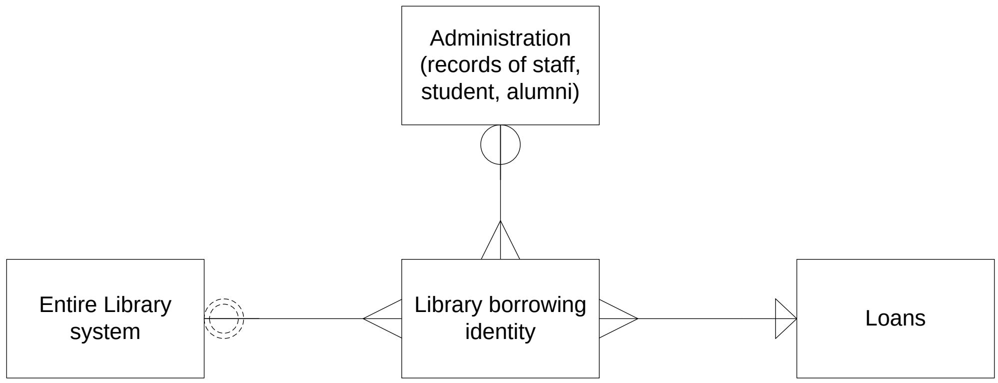

Figure 7. A FERD modelling a library circulation system using cartographic functional entities: external cartographic (administration); folded cartographic (loans); exofolded cartographic (entire library system).

5. Discussion

In this paper we have proposed an extension to ER conceptual modelling that fully models the space of all answerable questions, which is demonstrably an extension of the relations of standard data modelling theory, and which is underpinned by formal category theory.

We have defined the formalism of the functional entity, an encapsulated data resource that acts as a question-answering system. A functional entity is a generalisation of the Entity for sources of knowledge that are non-relational, or for which the standard processes of single entity modelling are difficult to achieve. A functional entity permits the modelling of any source in response to a request for information by returning a tuple of a consistent nature, while black-boxing the inner working in both design and use.

We enumerated a typology of 9 functional entities generated from two established principles to produce a modelling framework that can depict all existentially quantifiable relations, and demonstrated a satisficing extension to that framework (non-Aristotelian functional entities) for those situations that do not permit such relations. We discussed the system of knowledge mixins for qualifying the relations, and finally, we presented an adjunct to the framework, cartographic functional entities, to permit formally verifiable recursive and allorecursive (Pigott and Hobbs, 2001) documentation.

The encapsulation and occlusion of the functional entity permits us to show the logical relations that exist between parts of a distributed knowledge management system. This enables the physical design to be deferred or resources to be replaced with others that return the same answer at a functional level. This is very useful in high level planning, as knowledge management systems require that there be no destruction of the material recorded for a system as it is built. When the individual components of a wide area system are placed under the hegemony of different organisations, or even different professions, a high level map is necessary in order that some form of mutual understanding underwrite the progress of the KMS development. While this paper has emphasised the use of the FERD in conceptual analysis of organisational knowledge networks, the same tools can equally be used to model personal knowledge systems and knowledge systems for recording general cultural collections involving physical artefacts, documents, media and online web-resources.

The establishment of standard types of functional entities can provide a framework for the methodical conversion of the declarative design level to the imperative implementation level. We can identify consistent paths to follow (including design, documentation and verification strategies), common traps to avoid, and a way of ensuring a cross-system quality assurance that is currently not available with heterogeneous KM systems. We have discussed some of these in Pigott & Hobbs (2009) and will defer a detailed treatment of them to a later paper. Future research will also address the pragmatic/sociological dimensions of FERD modelling. Part of the framework's methodology are core requirements that must be met before a system can be modelled as a functional entity. A functional entity can only be modelled if it is timely (both in the sense of being up-to-date and being extractable in a sufficiently fast enough time to be useful), reliable, repeatable, and for externally called systems, substitutable.

In conclusion, by introducing the formalisms of erotetic logic and sketch logic to the process of modelling KMS, we can establish a theoretical underpinning for the conceptual modelling of knowledge systems that possesses a simplicity and rigour equivalent to that of modelling for traditional information systems. This new conceptualisation then incorporates traditional IS modelling as one aspect of a richer modelling system, and thereby includes all of traditional IS repositories as first class, unmediated sources of knowledge.

References

Bearman, D. 1988. Considerations in the Design of Art Scholarly Databases. *Library Trends,* 37**,** 206-219.

Boole, G. 1958. *An Investigation of the Laws of Thought on Which are Founded the Mathematical Theories of Logic and Probabilities,* New York, NY, Dover Publications.

Bradford Smith, H. 1919. Non-Aristotelian logic. 40.

Brilliant, R. 1988. How an art historian connects art objects and information. *Library Trends,* 37**,** 120-129.

Bromberger, S. 1966. Questions. *The Journal of Philosophy,* 63**,** 597-606.

Chen, P. P.-S. 1976a. The entity-relationship model—toward a unified view of data. *ACM Transactions on Database Systems (TODS),* 1**,** 9-36.

Chen, P. P.-S. 1976b. The entity-relationship model: a basis for the enterprise view of data. *ACM Transactions on Database Systems,* 1**,** 9-36.

Codd, E. 1969. Derivability, Redundancy, and Consistency of Relations Stored in Large Data Banks. *IBM Research Report RJ,* 599.

Codd, E. 1986. Missing information (applicable and inapplicable) in relational databases. *ACM SIGMOD Record,* 15**,** 53.

Codd, E. 1987. More commentary on missing information in relational databases (applicable and inapplicable information). *ACM SIGMOD Record,* 16**,** 42-50.

Codd, E. & Date, C. 1993. Much ado about nothing. (a debate over the issue of missing values in relational databases). *DBMS Magazine,* 1993**,** 46-50.

Codd, E. & Strehlo, K. 1990. Relational philosopher: the creator of the relational model talks about his never-ending crusade. *DBMS Magazine,* 1990**,** 34-37.

Colomb, R. & Dampney, C. N. G. 2005. An approach to ontology for institutional facts in the semantic web. *Information and Software Technology,* 47**,** 775-783.

Colomb, R., Dampney, C. N. G. & Johnson, M. 2001. Category-theoretic fibration as an abstraction mechanism in information systems. *Acta Informatica,* 38**,** 1-44.

Dampney, C. N. G. & Johnson, M. 1995. Application of "consistent dependency" to corporate and project information models. *OOER'95: Object-Oriented and Entity-Relationship Modeling***,** 445-446.

Dingel, J., Diskin, Z. & Zito, A. 2008. Understanding and improving UML package merge. *Software and Systems Modeling,* 7**,** 443-467.

Diskin, Z. 2005. Mappings, maps, atlases and tables: a formal semantics for associations in UML 2. *[Technical Report] Department of Computer Science University of Toronto Technical Report CSRG-566***,** 1-48.

Diskin, Z. & Kadish, B. 1997. A graphical yet formalized framework for specifying view systems. *Advances in databases and information systems, 1st East-European Symposium,. St. Petersburg,Russia,1997*.

Diskin, Z., Kadish, B., Piessens, F. & Johnson, M. 2000. Universal arrow foundations for visual modeling. *Theory and Application of Diagrams***,** 323-334.

Erickson, J. & Siau, K. 2003. Unified Modeling Language - Theoretical and Practical Complexity. *AMCIS 2003 Proceedings***,** 1-6.

Erickson, J. & Siau, K. 2004. Theoretical and Practical Complexity of UML. *AMCIS 2004 Proceedings***,** 1-7.

Fox, M. 1987. Beyond the knowledge level. *In:* KERSCHBERG, L. (ed.) *Proceedings of the First International Conference on Expert Database Systems.* Menlo Park, CA: Benjamin/Cummings Pub. Co.

Graesser, A. C., Person, N. & Huber, J. 1992. Mechanisms that Generate Questions. *Questions and information systems edited by Thomas W. Lauer, Eileen Peacock, Arthur C. Graesser; New Jersey: Lawrence Erlbaum Associates 1992***,** 1-27.

Green, B. F., Jr., Wolf, A. K., Chomsky, C. & Laughery, K. 1961. Baseball: an automatic question-answerer. *Proceedings of the Western Joint Computer Conference.*

Holl, H. 2007. Second thoughts on Gregory Bateson and Alfred Korzybski. *Kybernetes,* 36**,** 1047-1054.

Iverson, K. 1980. Notation as a tool of thought. *Journal of the ACM,* 23**,** 444-465.

Johnson, M., Rosebrugh, R. & Wood, R. 2002. Entity-Relationship-Attribute Designs And Sketches. *Theory and Applications of Categories,* 10**,** 94-112.

Jutoran, S. 2005. The Process from Observed Systems to Observing Systems. *School of Humanities and Social Sciences, Nova University*.

Kiwelekar, A. & Joshi, R. 2007. An object oriented metamodel for Bunge–Wand–Weber ontology. *Workshop on Semantic Web for Collaborative Knowledge Acquisition, IJCAI***,** 1–8.

Korzybski, A. 1994. *Science and sanity: An introduction to non-Aristotelian systems and general semantics*.

Kosko, B. 1993. *Fuzzy thinking: the new science of fuzzy logic,* New York, Hyperion.

Lauer, T. 2001. Questions and information: contrasting metaphors. *Information Systems Frontiers,* 3**,** 41-48.

MacKay, D. M. 1951. In Search of Basic Symbols. *Circular Causal and Feedback Mechanisms in Biological and Social Systems.* New York.

MacKay, D. M. 1960. What Makes a Question? *The Listener.*

Mandell, D. & McIlraith, S. 2003. Adapting BPEL4WS for the semantic web: The bottom-up approach to web service interoperation. *The Semantic Web - ISWC 2003***,** 227-241.

March, S. T. 2008. Ontology in the Design of IT Artifacts. *Presentation DESRIST 2008***,** 1-21.

McCulloch, W. S. 1974. Recollections of the Many Sources of Cybernetics. *ASC Forum,* VI.

Moon, D. A. 1986. Object-oriented programming with Flavors. *Proc. First Annual Conference on Object-Oriented Programming Systems, Languages, and Applications.* ACM.

Newell, A. 1981. The Knowledge level: presidential address. *AI Magazine,* 2**,** 1-1.

Newell, A. 1993. Reflections on the knowledge level. *Artificial intelligence in perspective,* 59**,** 31-38.

O'Sullivan, D. & Unwin, D. J. 2003. *Geographic Information Analysis,* Hoboken, NJ, John Wiley and Sons.

Otero, J. & Graesser, A. C. 2001. PREG: Elements of a Model of Question Asking. *Cognition and Instruction,* 19**,** 143-175.

Pask, G., Scott, B. & Kallikourdis, D. 1973. A theory of conversations and individuals (exemplified by the learning process on CASTE). *International Journal of Man-Machine Studies,* 5**,** 443-566.

Pias, C. (ed.) 2003. *Cybernetics - Kybernetik. The Macy-Conferences 1946-1953,* Zürich/Berlin: diaphanes.

Pigott, D. & Hobbs, V. 2001. The Noetic Prism: A New Perspective on the Information, Data, Knowledge Complex. *Western Australia Workshop on Information Systems Research, University of Western Australia, November 2001.*

Pigott, D. & Hobbs, V. 2009. The functional-entity relationship diagram: conceptual modelling for complex knowledge systems. *20th Australasian Conference on Information Systems, 2 - 4 December.* Melbourne.

Pigott, D., Hobbs, V. & Gammack, J. 2002. The Noetic Prism. *Computing and Information Systems,* 9**,** 78.

Pigott, D., Hobbs, V., Gammack, J. 2004. Just below the surface: developing knowledge management systems using the paradigm of the noetic prism. *Australian Conference on Knowledge Management and Intelligent Decision Support. Melbourne, December 11-12 2003.*

Pigott, J. P. & Mitchell, D. M. 2003. Significant flora and remnant vegetation on stream and roadside crossings of the onshore pipeline in Gippsland for BassGas. *Report to Clough Engineering.* Melbourne: HLA Envirosciences P/L

Psoinos, A. & Smithson, S. 1996. Exoloring the relationship between empowerment and information systems. *Proceedings of the 4th European Conference on Information Systems Lisbon Portugal July 2-4 1996.*

Quine, W. v. O. 1959. *Methods of logic,* New York, Holt.

Robinson, J. J. Year. The Transformation of Sentences for Information Retrieval. *In:* Congress of International Federation for Documentation (FID), Washington, D.C., October 1965, 1965. 1-14.

Rosenbloom, P. S., Newell, A. & Laird, J. E. 1989. Toward the knowledge level in Soar: The role of the architecture in the use of knowledge. *AIP 65*

Ruqian Lu 2004. Looking for a Mathematical Theory of Knowledge. *KEST04 Keynote presentation***,** 1-64.

Ruqian Lu 2005. Towards a Mathematical Theory of Knowledge. *Journal of Computer Science and Technology,* 20**,** 751-757.

Siau, K. & Tian, Y. 2005. A Semiotics View of Modeling Method Complexity-The Case of UML. *AMCIS 2005 Proceedings***,** 318.

Störrle, H. 2001. Describing Fractal Processes with UML. *Product Focused Software Process Improvement.*

Störrle, H. 2006. On different notions of model size. *Proceedings of the Model Size Metrics workshop at Models'2006.*

Walsham, G. 2005. Knowledge Management Systems: Representation and Communication in Context. *Systems, Signs & Actions,* 1**,** 6-18.

Yen, J. & Langari, R. 1999. *Fuzzy Logic: Intelligence, Control, and Information,* Englewood Cliffs, NJ, Prentice Hall.

Zadeh, L. A. 1965. Fuzzy sets. *Information and Control,* 8**,** 338-353.

Zhao, L. & Siau, K. 2002. Component-based development using UML. *Communications of the AIS,* 9**,** 12.

Appendix: FERD symbols

		Forms of knowledge resources			
		Shape-dominant Knowledge Resources	Granular-dominant Knowledge Resources	Scope-dominant Knowledge Resources	Non-Aristotelian Knowledge Resources
Forms of Question Entailment	Type 1 Instance-dominant	Standard relation	Standard recursive	Constitutive recursive	Contiguous non-A
	Type 2 Value-dominant	Absolute aggregative	Intensional aggregative	Fuzzy aggregative	Emergent non-A
	Type 3 Linkage-dominant	Ontological connective	Networked connective	Ruleset connective	Abductive non-A

Cartographic Functional Entities	External cartographic	Folded cartographic	Exofolded cartographic