The Functional-Entity Relationship Diagram: Conceptual Modelling For Complex Knowledge Systems

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
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. Modelling such 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 KM systems, and illustrate its use with a case study. We also show how the establishment of standard types of functional entity motivates practical guidelines for the conversion of the design to the implementation level.

Keywords
Functional entity, functional-entity relationship diagram, question-answering system, entity-relationship diagram, conceptual knowledge modelling.

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.

**Question-Answering Systems**

The question-answering (QA) 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.

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.

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. This paper describes such a 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 Entity-Relationship Diagram Revisited**

In the field of system design, there are few conceptual design tools as elegant and powerful as the ERD. In crystallising the interrelations of relational databases (Codd 1970), it serves that twin goal of any notation formalism, of sketching designs and documenting a completed system. The ERD is a form of digraph, with entities as nodes and relationships as edges, with the edges usually qualified to represent cardinality and participation of the relationships. With these simple tools, it is possible to sketch most databases and to analyse and verify the sketch: the ERD provides a snapshot of both the internal logic and the existential import of a database, and does it in a way that is implementation-independent and platform neutral. The extreme simplicity of the ERD renders it immediately graspable by both designer and implementer, suitable for its role in an iterative development process.

All existing notations, while useful in their domain, have limitations when considering a generalised solution. While there are other more generalised diagramming standards such as UML object notation (OMG 1997), or Higraphs (Harel 1988), the ERD has the advantage of grounding the model in a set of associations that match on to the erotetic (question-answering) process we described above. Moreover Chen himself pointed out the noun/verb mapping of entities and relationships in Chen (1983; 1997). The continued life and widespread usage of the ER diagramming convention testifies to its continued utility (Chen 2002), and justifies the choice of the tool for extension; the generalisations of system-wide erotetic qualification of associations would, however, be mappable onto these other systems. A similar reuse of the ERD is found in Green and Benyon (1996), who argued succinctly that the ERD formalism is a useful tool for mapping interactive information systems, both in terms of abstraction and simplification on the one hand, and the ease of comparing multiple accounts of the same system from participants.
There is also a tradition established soon after Chen for modifying the ERD: these include the widespread qualification of the relationship arc to indication cardinality and participation (the CACI method as outlined by Barker 1990 among many) or to indicate aggregation, composition, generalisation, dependency and realisation by the NIST (1993) following the work on abstraction and aggregation by Smith and Smith (1977a; 1977b). Typed entities and relations were investigated to solve complex multimodal or multidimensional data modelling problems, including papers at the 1985 conference devoted to extending the ERD for Knowledge Representation (ER85 1985), including temporality (Ferg 1985) and first order logic (Cazin et al. 1985). Chen himself proposed temporal extensions (Chen 1986) to his own formalism, as did Klopprogge (1981). Other extensions include fuzzy logic (Vert et al. 2002), extensibility (Liu and Sunderraman 1987) and multidimensionality (Hay 1999). The Extended ERD (Elmasri et al. 1985; Gogolla 1994; Hadzilacos and Tryfona 1997) and the Hierarchical ERD (Thalheim 2007) added further categorical, geographical information system (GIS) and hierarchical qualifications to the model, while at the same time acknowledging that the extensions were similarly couchable in terms of English verbs and nouns (Hartmann and Link 2007). An historical survey of the range of extensions to the ERD can be found in Patig (2006).

**FUNCTIONAL ENTITIES**

What we require is a way to model and diagram these kinds of QA systems, in a way that lets us detail both requirements and expectations.

A QA system acts in a declarative manner – it describes the goal that is needed. This is in strong contrast to (say) SQL or ARC which describe the solution in a procedural manner. We need a declarative layer of explanation to permit modelling of complex systems. The question that is asked gives a set of criteria that determines the possible matching values in the dataset. In a way, this is like traditional formal logic, wherein the conclusion may be considered implicit in the premises when there is a syllogism. This core criteria set is indexical to that tuple it retrieves: not only is the tuple implicit in the criteria, the style of answer is implicit. The question will imply what the answer will bring and the mechanism whereby it will be brought.

The internal mechanism of the answer-provider is immaterial to the model, apart from requirements of reliability and truthfulness. It is a “black box” from the vantage point of the questioner: we know that there is an internal state within the black box that effects the answering, which takes the declarative mechanism and converts it into a procedural process, something that the questioner does not need to know.

In systems theory such a black-box mechanism is known as a reliable, relatively isolated system (RIS). RISs are systems where all that is (or can be) known externally are the responses to any input by the system, the repertoire of states and the trajectories of paths within it. The answering system has very few characteristics in the abstract - a name, a single pattern for a repertoire and a small set of responses. The trajectories are paths that describe the creation of tuples derived from any input key set. By using such encapsulations, the mechanisms to articulate the answers become *referentially transparent* (Søndergaard and Sestoft 1990) to the questioner, that is to say they react to the questions in the same manner as the underlying system.

In Codd’s 12 principles, (1974) we find the idea that an entity can be a subsystem for which basic considerations of responsiveness to data can be met. In Codd’s RMT paper (1999) the semantic aspect of such subsystems is shown to be the optimal level for description and manipulation. Although not strictly entities in the sense of Chen (1976a) they behave in a systems-theoretical manner as if they were. We can therefore make the case that any logically-defined subsystem that provides referential transparency, and which provides a schema and a tuple for a given key, and which has a consistent degree and cardinality behaves as an entity, is functionally the equivalent (within the context of the ERD) of an entity, and we may call such a logical subsystem a *Functional Entity*.

The system of notation we propose is based on depicting a formalised relationship between encapsulated data resources, having a typed behaviour, which responds according to an agreed norm. These encapsulated data resources, termed Functional Entities (FEs), provide a mechanism for designing a system with all forms of data, relations, internal consistencies and logics. By placing the complexity of the relationship inside the Functional Entity, the simple nature of the digraph formalism is preserved.

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 present part of the FERD formalism in the next sections, and will refer briefly to some additional types extending the mechanism in the discussion.
FUNCTIONAL ENTITY TYPES

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) F(A,x) - which translates as “which of A has the value x?”
2) F(x,B) - which translates as “what x exists such that its function F is B?”
3) x (A,B) - which translates as “is there a connection 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) Connection-dominant entailment, wherein the connections entail the answer. This can be either where we know two things exists and we are trying to find what connects them, or where we know what connections to look for, but don’t know what the connected instances are. We model such QA systems as Type 3 functional entities.

In summary, a functional entity is a question-answering system for a particular type of question, as defined by the nature of the entailment. We will now consider the three types defined above in greater detail. A complete set of FE symbols is shown in the Appendix.

Type 1 functional entities: Instance-dominant

There is already an established Type 1 question in the existing ERD – it is the standard relationship. It presents a set of entailed values as an answer to a query. We begin with the standard representation as this is the simplest case for our system of generalisation. When we ask “what records match this criterion?” we are effectively asking for the tuple that is a standard subset of the table. We call this type of question the standard question, and represent it using the conventional line or crow’s foot mechanism.

However, there is no standard way with the ERD to indicate that the entailment can be recursive. (We can indicate the recursive relationship through means of a secondary reflexive relationship edge: this is not the same thing as representing a functional entity that is essentially recursive.) A great deal of knowledge is recursive in form. This is because of the relative ease conceptually of representing the world in terms of meronym/holonym (part/whole) relationships. We call these questions standard recursive questions. (Note that we are not discussing hierarchies and ontologies here, only those cases where there is an entailed instance is indexical to other entailed instances.)

There is a third form of recursion of instance dominant sources of knowledge, where there is a rule determining the links between relational versions. In a true recursive relationship the link between the parent and child instance is identical no matter how many times the data relation is called, and only involves one entity. However, it is possible to have a different subtype of membership at each level, and in such cases we need extra details at every parent-child component of the hierarchy to tell us what values exist. This means that in getting an answer we have to pay heed to the subtype from which the information is derived. One important fact here is that the individual records may not be true peers in their attributes, as subtypes may differ in these regards. We call these kinds of questions constitutive recursive questions.

Figure 1 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 car, and a service organisation that can fit the part.
The first relationship is straightforward: each car model is made by a manufacturer, and a manufacturer makes many models. This is modelled with a standard functional entity.

Each car can have parts replaced, but the spare parts themselves are made up of subassemblies, each of which quite frequently can be purchased separately, and if only a larger part is available, sometimes that will be purchased to acquire a subcomponent. The parts or subassemblies themselves can often fit many different models and even brands of cars. This second relationship is a standard recursive relationship, but the question it is answering here is “is this part available?” - the answer is that either it is available, or else something exists that will contain the needed part. The standard recursive functional entity supplies this answer.

A similar question involving recursion would be “which company could fit the part?” Here the answer sought pertains to expertise, which would reside in a technician, employed by a service centre which may be part of a chain or franchise, or a division of a company. The qualification of the particular service group would depend on whether or not it had such a technician, but the nature of the recursion will depend on the different structures the organisations have in place. This kind of recursion involves different relationships between the parent-child relationships in the hierarchy – we term it a constitutive recursion. While it can still be implemented with a relational database, it may require some kind of stored procedures to operate.

**Type 2 functional entities: Value-dominant**

Type 2 questions are where the values are determined and queried, and where the records entailed do not necessarily partake of the same entity: these are questions that ask “what things have the consistent attribution of value B applicable to them?”. The application can be general (“what exists there?” or “what happened then?”), or indeed “what happened there and then?”); or it can be particular (“what is green there?”; “what was the value of the Yen yesterday?” or “what was 34°C there and then?”). This information draws its value from the interpolation of information – the values either in between known values, or the values between known points.

When we explore material with Type 2 questions, we could be looking 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.

We can identify three kinds of Type 2 data, responding to three kinds of questions. Firstly, where values are presented in absolute terms, within a universal static framework (such as latitude-longitude pairs, Gregorian calendars), which we can term absolute aggregative questions. In such cases, the results are absolute irrespective of position or occasion of enquirer. Typical questions would be “what flora exists in this location” or “are there recorded break-ins in the area”.

Contrasted to this, we have such situations where the meaning derives from being instantiated in a moment or a place, of both observer and of defined point/time. Just as there will always be the tallest mountain, or the longest river, so there will always be an answer (even if a tied answer) for the question. But it will depend on an immediate analysis of the population. These are intensional aggregative questions and are tethered to the instance.

These two types of questions have an unmediated nature, but a significant kind of question will involve the use of the fuzzy logic paradigm (Kosko 1993; Yen and Langari 1999; Zadeh 1965) 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. We call these fuzzy aggregative questions where values are deliberately rounded according to a series a quanta/steps to force fractal aggregation at the point.
Figure 2. Type 2 functional entities. From left: absolute aggregative; fuzzy aggregative; intensional aggregative.

Figure 2 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. The first question we might ask is “where is the phenomenon distributed?”, a question of the sort frequently employing a GIS. There are many possible answers to this question in purely physical terms: location can be a square centimetre, a hectare, or a square kilometre – in fact all three can be correct at the same time – but a footprint is also a location, as is a block, a paddock or a suburb. An action of location at the higher level doesn’t necessarily have meaning for greater precision, and one locating at a lower level has no implication for spread of phenomenon. It does however have an absolute framework: it is uniquely identifiable in terms of a word coordinate system, and is the same regardless of enquirer. We thus model distribution as an absolute aggregative functional entity.

The second aggregative question being asked in this scenario is “where is the nearest response team?” Here, the answer is based on the situation, and is relative to the problem encountered. The question is really – “which is the response team that is equipped to deal with this situation, and which is closest in terms of travel rather than actual location?”. Here the answer is a relative one depending on the rest of the set, and possibly involving tradeoffs (a closer, less-equipped team may be sufficient). We model closest response team as an intensional aggregative functional entity.

A third kind of question we can ask is “what should we do about it?”, the answer to 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. The answer which subrange of a pre-established value range it falls into, in terms or either or both of quantitative and evaluative (subjective) analysis. Here we could be asking a question like “How serious is this crisis?”. Such a question will involve will involve fitting the values into a matrix of rules and thresholds. We thus model response strategy as a fuzzy aggregative functional entity.

Unlike the Type 1 example, the Type 2 scenario shown in Figure 2 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 sources are where the either the values or the instances are determined (or both) 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?” Here there is no useful general application, as everything is linked to everything else in innumerable ways like the cat in Dirk Gently Holistic Detective Agency (Adams 1987). In the particular form it is found in all systems of knowledge that can be represented by a graph (i.e. networks, stars and trees. These are manifold, ranging 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 predictable) numbers very quickly indeed.

When we consider how data can be connected, we can identify three different forms of Type 3 questions (and hence Type 3 functional entities). A very common form is 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, and we term these questions *ontological connective* questions.

The second form of Type 3 question is the one born of propinquity, 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. We call such questions *networked connective* questions. An important thing to remember here is that 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.

A third form is where values and instances are associated by chains of reasoning, inductive or deductive (or indeed of any other form of chainable non-standard logic) and can be forward- or backward-chaining. The answers here can be goals set, or they can be implicated instances, or they can be likely values: the most significant thing is that unlike the other two there is a process of reasoning before the network can be created. We call such questions *ruleset connective* questions. Expert systems (either inferential or production) can be modelled as such question systems. A large number of pre-established facts are rules are recorded so that when the expert system encounters a new set of facts, the reason can be carried out automatically.

![Figure 3. Type 3 functional entities. From left: networked connective; ruleset connective; ontological connective.](image)

Figure 3 explores the knowledge relations involved in examining the advance of an infection in a population, a scenario where links between instances are the most significant knowledge relation. *Epidemics* manifest themselves as *cases*, representable simply as a one-to-many relationship. More complicated is 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 contacts, further contacts still. Epidemiological knowledge bases rely very heavily on such questions and their answers. When tracking an infection back to Patient Zero, or finding the source of food poisoning in a supply chain, this kind of connectivity provides the infrastructure for recording facts. We represent *contacts* as a *networked connective* functional entity.

A second kind of question would be “which classification is the infection under?” with respect to WHO infection classification. These represent an ontological discussion of the world, so such a question involves asking where in a pre-established ontological scheme the infection fits. Knowledge workers have to deal with classificatory systems frequently, and the ability to indicate this kind of consult is vital. We represent *classification* as an *ontological connective* functional entity.

The third kind of question in this scenario involves suggested *treatments*, either through prompted queries of the old fashioned expert system kind, or of a rule production system. A question like “what is the best treatment for this epidemic?” will have an answer based on stored medical and clinical knowledge, and the answer will draw on the conjunction of the existing values and this stored knowledge. We represent treatments as *ruleset connective* functional entities.

### A CASE STUDY IN KNOWLEDGE MODELLING: THE BOX-IRONBARK ECOLOGICAL THINNING TRIAL

We shall now discuss the use of FERD in modelling an existing wide-area knowledge management system, the informatic representation of the Box-Ironbark Ecological Thinning Trial coordinated by Parks Victoria. We have described this project in previous work (Pigott et al. 2009).

This project involved coordinating a number of different organisations and systems and the problem of representing the knowledge resources of the entire project at a conceptual level lends itself well to the declarative nature of the FERD approach, particularly given the specialist nature of the expertise, the scale of the undertaking, and the distribution of the responsibility for data and reporting amongst organisations.
In the Box-Ironbark thinning trial, treatments were applied to select plots for different sites, which were monitored using repeated measurements such as species cover and growth. Each site was located in Parks Victoria Reserves containing Box-Ironbark forests that were logistically feasible (Pigott et al. 2008). Observations were made from these measurements, and then evaluated by expert groups, whose constitution changed over time, and which were drawn from either organisational offices or individual experts. The participation of the organisations changed as individuals moved between them, according to their internal structures (Pigott 2009). The FERD of the Box-Ironbark ecological thinning trial is shown in Figure 4.

![Figure 4. FERD of the Box-Ironbark ecological thinning trial.](image)

We shall now discuss each of the functional entities in the model. We represent the Classification of the Vegetation as a Type 3 ontological connective FE. By linking the selected instances of vegetation into the classification FE, we are requesting of an ontological framework the situating of the specimen in the standard Linnean taxonomic scheme, represented in the HISCOM system, the Australia-wide authority.

The Vegetation is represented by a conventional entity, and is recorded as point data in a GIS system, which also contains details of the plots that were chosen for the trial. Plot is represented by an absolute aggregative FE, because while the areas are identified by the known tree specimens, the plots are the level at which the results are to be presented. This points to a particular feature of Type 2 FEs, the risk of inapplicability of measurement—something that is true of the whole may not be true of the part—tree density, for instance, will not be true of the roads running through the area plot.

The details of the treatment trials are recorded externally (in a series of spreadsheets, by the foresters on the ground) and are tabulated in a series of tables in a Microsoft Access™ database. This is represented by the standard entity Treatment in the Figure 4.

Because of the number of recordings involved, it is not possible to do real-time querying of Treatment to get derived statistical information. In consequence of that, dedicated processing time is given over to processing the data every time a new set comes in, and the processed data is stored as Observations in a separate set of tables in another database, represented as an intensional aggregative functional entity. This means that rapid response to querying is possible, but a direct link to the source data is not, with no guarantee of applicability of the result to any particular measurement.

The summaries in Observation have been used by various authorities and individuals to write reports over time, and the reports have fed back in an adaptive scientific management life cycle, as discussed in Pigott (2009). Because of the varying nature of the scientific establishment in the State of Victoria and the movement of scientific professionals between organisations, as well as in and out of the workforce, the origin of the expertise and of the reports generated varies through time. This is represented by the constitutive recursive FE Expert.

We can see from this example how a wide area knowledge catchment and reporting system, involving several informatic systems and multiple professional jurisdictions, can be modelled clearly in one diagram, showing the flow of observation and understanding from planned measurement to final report. As part of the process of migrating these knowledge protocols to a new wide area conservation project by the principals involved, the authors are collaborating in developing a series of FERDs to explore alternative paths to effect that transfer.

**DISCUSSION**

In this paper we have introduced the formalism of the functional entity (FE), an encapsulated data resource that acts as a question-answering system. A FE 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.

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.

We defined three main types of functional entity: Type 1, or instance-dominant, Type 2, or value-dominant, and Type 3, or connection-dominant, each with three subtypes. We note that other types of functional entity can also be identified. The nine functional entities we have described are part of a set of 16 intended to cover the spectrum of knowledge sources. Space does not permit us to elaborate them here, but we note that the additional functional entities include non-Aristotelian (where the Aristotelian unities of space and time implicit in Type 1 functional entities break down, resulting in intermittent or emergent entities), ordered data, and functional entities that simplify the modelling of large scale or external systems.

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.

Implementation and maintenance of a system that has been modelled with FERD requires paying attention to events that mitigate the reliability of the referential transparency. These fall into two sorts: those circumstances that alter the context (and therefore call into question the assumptions about consistent interoperability of values made in the design) and those circumstances which cause the repeatability of the erotetic constraints (and therefore call into question the expectations of continuity in the design).

As an example, we can consider the implementation, maintenance and quality issues associated with the networked connective functional entity. (Similar considerations could be made for all of the functional entities described here.)

To realise a networked connective functional entity, the nature of the connection would have to be analysed in the design phase, and the useful instances established. The design would have to employ a constrained termset to prevent over-population of the discourse, and some agreement about measurement units would have to be established. Where there was a matching relational table in another system some degree of key-deference (either the relational table or the network would have to accept key surrogacy) to ensure alignment: validation and verification would check that this was the case, and refuse new records that didn’t adhere.

For implementation, a suitable network database (such as Berkeley DB XML; Oracle, 2009 ) capable of holding rich networks would be employed. In developing the material for input, accuracy and consistency of the source system is critical at each node – that would determine the reliability of responses, and hence of values for use. There would also be runtime issues (checking for network artefacts (small world networks mean erroneous results at 4+ jumps) whereby everything might seem related and timeouts or over-retrievals, explicitly curtailing searches in data requests as search times are massively non-polynomial for searches in the entire domain.) Finally, when the system involves resources that span two hegemonies, and for which different goals are stipulated, it would be necessary to watch for change at the teleological level with respect to the real world phenomena they are describing.

Future research in this aspect of FERD involves formalising these guidelines and developing methodologies for using the FERD system, and continuing to test its usefulness in real world knowledge systems.

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. We also believe the FERD has great potential as a teaching tool and have trialled its use in an undergraduate Knowledge Management course, to enable the conceptual design of knowledge bases in Protégé. The use of the FERD notation not only provided an insight into the ways that knowledge assets could be put together, but also allowed the students to compare each others’ designs easily, and to appreciate the problems of multiple perspectives in knowledge systems design.

In conclusion, by introducing the formalism of erotetics from philosophical 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


NIST. 1993. "Integration Definition for Information Modeling (Idefix)."


ACKNOWLEDGEMENT

We thank J.P. Pigott for helpful discussions on the Box-Ironbark case study discussed in this paper.

APPENDIX: FERD SYMBOLS

<table>
<thead>
<tr>
<th>Type 1</th>
<th>Standard</th>
<th>Standard recursive</th>
<th>Constitutive recursive</th>
</tr>
</thead>
<tbody>
<tr>
<td>Type 2</td>
<td>Absolute aggregative</td>
<td>Intensional aggregative</td>
<td>Fuzzy aggregative</td>
</tr>
<tr>
<td>Type 3</td>
<td>Ontological connective</td>
<td>Networked connective</td>
<td>Ruleset connective</td>
</tr>
</tbody>
</table>

COPYRIGHT

Diarmuid J Pigott & Valerie J Hobbs © 2009. The authors assign to ACIS and educational and non-profit institutions a non-exclusive licence to use this document for personal use and in courses of instruction provided that the article is used in full and this copyright statement is reproduced. The authors also grant a non-exclusive licence to ACIS to publish this document in full in the Conference Papers and Proceedings. Those documents may be published on the World Wide Web, CD-ROM, in printed form, and on mirror sites on the World Wide Web. Any other usage is prohibited without the express permission of the authors.