An Exploration of Superexpertise

Xenogene Gray

A Thesis submitted in the fulfilment of the requirements for the degree of Master of Philosophy

Department of Computing
Faculty of Science
Macquarie University
Sydney, Australia

June 2011

Abstract

This thesis considers the nature of expertise, focusing on expertise beyond normal human capabilities; such expertise is called, by the author, *superexpertise*. The question asked in this thesis is: how can people manage problems that require superexpertise? A model of superexpertise is constructed through an exploration of aspects of this concept set out in seven published papers on the subject. In the first paper, published in 2007, the concept of superexpertise was identified with the characteristic of automation of large scale combinatorics by a legal expert system shell, eGanges. Further contributions to the model are added in the remaining six papers so that conclusions can be drawn as to the nature of superexpertise. The notion of superexpertise has evolved from the work of expert systems, knowledge based systems, knowledge representations and problem solving methods. It is particularly useful for problems involving large-scale combinatorics and sorting according to a prescribed multi-valued logic.

To manage superexpertise, it must be elicited from expert epistemology. An expert epistemology is a theory about the knowledge bound up, encompassed and contained in the expertise and includes knowledge representation, semantics, specifications, heuristics, reasoning, etc; it is concerned with 'How we know what we know', and deals with the means of producing knowledge, including how knowledge is acquired. Specifically, the aim of this thesis was to show the applicability of two epistemologies, called eGanges and NeGame (a complement to eGanges), for three particular types of problems, namely, Quality Control problems, Adversarial problems and Negotiation problems. Initially a qualitative methodology of case studies of superexpertise is relied on, but quantitative methodologies are also used as mathematical characteristics of superexpertise are revealed and developed. An introductory definition of superexpertise is expanded into a model with two major features: (1) large scale combinatorics that require (2) multi-value logic processing. The NeGame epistemology was introduced in this thesis, but requires further work and refinement, along with a logical verification more detailed than can be provided by these limited case studies.

Statement of Authenticity

Except where otherwise indicated, this thesis is my own original work, and has not been submitted for a higher degree to any other university or institution.

____

Xenogene Gray

29 June 2011

Acknowledgements

First of all I would like to thank my wonderful mother for her many decades of moral, academic and financial support. She is the best mother anyone could ever hope for, and without her I would not even exist.

Next I would like to thank my supervisor Debbie Richards for her academic guidance, and collaboration in the production of academic work.

Similarly, John Zeleznikow has been a helpful academic collaborator, as attested to by our three joint papers covered in Chapter Four.

Finally, as another collaborator, I would like to acknowledge the contribution of Lyn Treanor.

Table of Contents

List of Figures

List of Tables

Chapter 1: Introduction and literature overview

This thesis considers the nature of expertise, focusing on expertise beyond normal human capabilities; such expertise is called superexpertise. The question asked in this thesis is: how can people manage problems that require superexpertise? A model of superexpertise is constructed through an exploration of aspects of this concept set out in seven published papers on the subject. In the first paper, published in 2007, the concept of superexpertise was identified with the characteristic of automation of large scale combinatorics by a legal expert system shell, eGanges. Further contributions to the model are added in the remaining six papers so that conclusions can be drawn as to the nature of superexpertise.

Superexpertise requires the automated, interactive implementation of complicated algorithms, such as the combinatorics of a hierarchical multivalued logic. In expert epistemologies, like eGanges [Gray and Gray, 2003] and NeGame [Gray, Gray and Zeleznikow, 2011], the self-similar hierarchical nature of the expertise is used to break down the complexity into several human manageable interactive maps. Superexpertise has evolved from the work of expert systems, knowledge based systems, knowledge representations and problem solving methods, as the computation of expert epistemology that is beyond the limits of individual human capacity. It is particularly useful for problems involving large-scale combinatorics and sorting according to a prescribed multi-valued logic. To manage superexpertise, it must be elicited from expert epistemology.

An expert epistemology is a theory about the knowledge bound up, encompassed and contained in the expertise and includes knowledge representation, semantics, specifications, heuristics, reasoning, etc; it is concerned with 'How we know what we know', and deals with the means of producing knowledge, including how knowledge is acquired. This thesis investigated two epistemologies: the eGanges epistemology covered in Chapter Two, and the NeGame epistemology covered in Chapter Four. These two epistemologies address different problem types.

1.1 Problem types

When people and institutions are faced with problems, then solutions must be chosen; continuing with the *status quo*, (i.e. ignoring the problem) may be a possible choice. Sometimes these solutions must be chosen without complete knowledge of the problem. The causes of knowledge imperfection may be a lack of sufficient information, or contradictory information. A decision may have to be made with incomplete or inconsistent information, especially if there is a deadline by which a decision must be made. As such, any practical reasoning system must be able to deal with *incomplete* and *inconsistent* knowledge, regardless of the type of problem concerned.

There are many different types of problems, and each problem type has its own unique issues and solutions. In the 1990s, artificial intelligence researchers in the fields of knowledge based systems and knowledge acquisition (KA), such as Breuker [1994], explored the notion of *Problem Solving Methods.* In the same way, but identifying and focussing on different classes of problems, this thesis looks at three particular types of problems: *Quality Control* problems, *Adversarial* problems and *Negotiation* problems. This problem classification is derived from expert epistemology.

Quality Control problems attempt to ensure that knowledge of how to best perform a job can be quickly and easily spread to a workforce, thereby aiding businesses to achieve *agile management*.

Adversarial and Negotiation problems both deal with *dispute resolution*. An adversarial problem is where a dispute between two parties is resolved by a set of rules which determine who 'wins'. Negotiation problems require that people try to resolve a dispute amongst themselves instead of treating the problem as an adversarial problem; either application of dispute resolving rules has been postponed or abandoned, or there are no agreed rules to resolve the dispute. Sometimes there is no rule enforcer to ensure adherence to an adversarial problem resolution; in such cases only negotiation resolutions, or war, can resolve the dispute. These three types of problems can be seen as *Finite Domain Expertise* problems.

1.1.1 Finite Domain Expertise (FDE)

There may be an infinite number of possible solutions to some problems, although it should be possible to group these solutions into a finite number of classes of solutions. For example, it is possible to group the possible responses that deal with human greenhouse gas emissions into two broad categories: 1) humanity as a whole commits to decrease greenhouse gas emissions, or 2) humanity as a whole does not commit to decrease greenhouse gas emissions (i.e. we either stabilise or increase greenhouse gas emissions). These two solution classes can be separated by how humanity answers the *dichotomy*: “Will humanity as a whole commit to decrease greenhouse gas emissions?”

By definition, a dichotomy is a choice between two possible answers. If there is a finite set of non-overlapping possible solutions to a problem, that covers all possibilities, then a series of dichotomy answers can be used to determine which solution is selected. The simplest process for doing this is to iterate through each possible solution and ask if that is the one selected, until an answer of yes is given. As the solution set covers all possibilities, at least one solution will be selected. Since there is no overlap between solutions, only one possibility will be selected. The term *Finite Domain Expertise (FDE)* is defined to represent this class of problem; a field with a finite, non-overlapping solution set requires *Finite Domain Expertise*. Not all fields of expertise can be classified or treated as *Finite Domain Expertise*.

In short, to be a field of Finite Domain Expertise, the field must be:

1) **Finite**: there must be a finite, though potentially huge, number of possible solutions to choose from.
2) **Consistent**: ultimately, only one possible solution can be implemented. That is, the solutions are mutually exclusive.

These two requirements are reasonably common in the computer science literature. If there is not a *finite* number of solution types to pick from, then selecting a solution may not be possible; c.f. Turing's halting problem[1]. The FDE *consistent* requirement equates, for example, to the Singleton Valuation Assumption [Boeva et al, 1998, p6].[2]

Both Quality Control and Adversarial problems rely on rules that have been determined by humans, so both should be problems of Finite Domain Expertise; as humans have *pro tem* (i.e. for the time being) determined all applicable rules, there is a finite set of applicable rules. If the rules are *consistent*, then application of them will not cause any contradictions, so only one possible solution will result. Provided there are a finite number of results from every rule, then a finite rule set ensures a finite set of possible solution types, as there are a finite number of possible results.

Of course, badly designed rules can lead to inconsistencies: for instance, a situation where two solutions that should be mutually exclusive are both possible. However, generally rule makers avoid such inconsistencies, and default priority ordering of rules can prevent such inconsistencies, e.g. assume a defendant is not guilty, unless the defendant is shown to be guilty without a reasonable doubt.

Negotiation problems also generally deal with a finite set of possible solutions; each item of dispute between two parties can either be resolved as a win-win, a win for party A, a win for party B, or a lose-lose (see paper 4.3). While the use of dichotomies should always work, as

1 There is a halting problem in a computer program if it can not be determined whether the program will finish running, or continue to run infinitely [Davis, 1958, pp. 70–71].

2 The Singleton Valuation Assumption states that one and only one proposition is true in each world, all others being false; each such proposition can represent a different possible 'truth' value, as only one of them can apply (be true) in each world.

outlined above, for negotiation between two parties, it is arguable that tetrachotomies (four possible answer questions) are a more natural fit, due to the fact that the dichotomies of 'win for party A' and 'win for party B' are always tied together, and as such form a set of four possible solutions (two solutions from A times two from B, for a total of four). This raises the question of what should be regarded as the basic unit of 'truth' that needs to be represented for an FDE problem.

1.1.2 Representing truth

In this thesis, an ontological truth is a truth value that actually exists. For a single dichotomy there are only two solution sets that actually exist; the solution set where the dichotomy has an answer of yes, and the solution set where it has an answer of no. Each of these dichotomy solution sets may be composed of multiple solutions resulting in further structure to these sets; these multiple solutions can be further divided via answers to further dichotomies. With perfect knowledge, there is no difficulty in determining which solution set of a dichotomy is the true solution set. However, with incomplete or inconsistent knowledge, the 'true' set may not be known with certainty. Therefore any known 'truth' should also have encoded the reliability of its truth, i.e. the truth value should contain an epistemic aspect (i.e. knowledge about a knowledge component; knowledge of how well we know what we know), as well as a truth component. Probably the first, and arguably simplest ways to handle this epistemic aspect of truth is by using multiple truth values [Lukasiewicz, 1920], although the term 'epistemic truths' was first considered by Belnap [1977].

Another important milestone for epistemic truths was the development of bilattices [Ginsberg, 1988] which explicitly use the two dimensional epistemic truth aspect. A bilattice is two linked lattices with one using a truth ordering, the other using a knowledge ordering, where both axes are epistemically important to the overall 'truth'. Bilattices have primarily been further developed by Fitting [1990, 1991, 1992, 2006], but the many-valued epistemic logic based on bilattices by Majkic [2004] is of considerable importance for this thesis, and is further discussed in Chapter Three where the use of multi-valued logics is justified.

Any system that can retract its conclusions when given further information is called non-monotonic, because the advancement of its conclusions is not monotone (i.e. not one-directional). There are many types of non-monotonic logics, each using a different method to represent knowledge (e.g. default logics, multi-valued logics and autoepistemic logic [Marek and Truszczynski, 1991]). However work has been done that shows the intertranslatability between many types of non-monotonic logics (e.g. [Janhunen, 1998]).

Generally, each *knowledge representation* (KR) has advantages and disadvantages, and these can vary between the different types of expertise (e.g. a KR might not be metaphysically adequate for non-finite domain expertise, but can have less disadvantages than other KRs for fields of finite domain expertise). However, to be able to critique these advantages and disadvantages requires an understanding of the type of superexpertise required for the problem, as discussed in Section 1.2. This leads to a series of further specific research questions and approaches set out in Section 1.3.

1.2 Superexpertise and epistemology

In the information age, many fields of expertise require people to deal with super human amounts of complex, changing and competing knowledge. Dealing with this massive amount of ever changing knowledge may cause *information overload*, even for field experts; software tools may be necessary to enable people to handle this *superhuman* level of *expertise*. Such tools would enable people to perform decision-making in fields requiring, what is called in this thesis, *superexpertise*.

Superexpertise requires an understanding of the knowledge associated with how a problem solution can be determined; it requires knowledge about the knowledge of handling the problem. The study of knowledge about knowledge is the study of epistemology. Therefore superexpertise requires appropriate epistemology to be used for the problem. As such, this notion is in line with other Knowledge Level [Newell, 1982] approaches such as Knowledge Acquisition and Documentation Structuring (KADS) [Breuker and van de Velde, 1994], Unified Problem-solving Method Development Language (UPML) [Fensel et al, 1999] and Generic tasks [Chandrasekaran, 1986, 1990].

Electronic computers were first deployed to help humans handle a specialised expertise problem, namely cracking the Enigma code. Not long after that, the field of expert systems was established to use computers to help humans with other specialised expertise problems. This early expert systems research eventually evolved into knowledge based systems research. This thesis poses the next evolutionary step following knowledge based systems, namely superexpertise, which, as such, is introduced and explored here.

In particular, this thesis poses what is required of software tools to enable them to help humans, by providing experts, and others, with appropriate access to the extensive information storage and rapid processing capacity of computers. Such computer based knowledge systems are not a new idea, but some of the fundamental problems have still not been adequately overcome, such as the *knowledge acquisition bottleneck* [Feigenbaum, 1981, p.226; Gillies, 1996, pp.25-31], which is further investigated.

By having a good epistemological understanding of the problem to be solved, we can exploit the fact, “the closer our methodology and programming language model the way we think about the world, the easier it will be to manage complexity” [Liberty, 1998, p11]. So, by assisting the management of complexity, we can ease the knowledge acquisition bottleneck, particularly with the use of knowledge abstraction and knowledge encapsulation.

1.2.1 Knowledge Representation Requirements

Any tool designed to provide superexpertise must also fulfil the knowledge representation (KR) requirements recognised in the literature, starting with McCarthy and Hayes [1969], who described three aspects that any expert system's knowledge representation would need to adequately handle in order to achieve what is required. McCarthy and Hayes also implied a fourth requirement, computational tractability, that was explicitly detailed by Bench-Capon [1990]. These adequacy requirements are as follows:

1. **Metaphysical adequacy**: requires that the representation must not contradict the facts of the world it is to represent. For example, a representation of the real world as non-interacting particles is metaphysically inadequate as particles do interact;
2. **Epistemological adequacy**: requires that the representation must be able to represent the facts of the world it is to represent. For example, a representation of a world must be able to refer to 'objects' in the world and their relationships, and as all knowledge of the world has some degree of uncertainty, this uncertainty must also be able to be represented;
3. **Heuristic adequacy**: requires the representation must be able to 'reason' about the world. For example, representing only the state of a game is heuristically inadequate as the representation can not reason about how to achieve a goal; and
4. **Computational tractability**: a knowledge representation should be computationally tractable, i.e. its problems are solvable in a finite time.

Further extending the adequacy requirements of McCarthy and Hayes, Bench-Capon [1990, pp.15-18] described the desirable expressiveness features of a Knowledge Representation, which are as follows:

5. **Lack of ambiguity**: every valid expression must have exactly one interpretation;
6. **Clarity**: “the representation must be amenable to understanding by people, even those who may not be entirely immersed in the particular representation formalism” [Bench-Capon, 1990, p16];

7. **Uniformity**: knowledge is represented consistently;
8. **Notional convenience**: this is a form of user-friendliness in constructing and reading representations;
9. **Declarativeness**: knowledge should not be stored by destructive assignment. Instead the knowledge should be stored so that all references to a variable declaratively refer to the same instance of the variable; and
10. **Relevance**: it is important to ensure that the level of "expressiveness is required by the task" [Bench-Capon, 1990, p17].

1.2.2 Knowledge Based Systems Requirements

There are six broad tasks that must be dealt with for the production of a knowledge based system to help achieve superexpertise. These are:

1. **Knowledge acquisition**: The problem of knowledge acquisition was first raised by Feigenbaum [1969; 1981, p.226], and proved sufficiently difficult that Shortliffe recognised that it would be sounder to obtain for himself a medical qualification and experience so he could build MYCIN, a medical expert system [Shortliffe, 1976]. Knowledge acquisition requires more than just loading knowledge and rules into a knowledge base; it requires ensuring that the knowledge loaded is validated by a field expert to ensure that any advice given, based on the knowledge base, will be correct.
2. **Knowledge maintenance**: Strongly related to the task of knowledge acquisition, is the task of knowledge management. This involves the process of updating a knowledge base when the rules and knowledge change. The problem of knowledge management can require that the entire knowledge base be revalidated by an expert, as new rules may contradict or cause inconsistencies in any non-monotonic knowledge base update.
3. **Knowledge processing**: Knowledge processing is also strongly related to knowledge acquisition and maintenance. Rule firing order can have significant consequences for the final advice that is provided by an expert system. The concept of Salience Scores [Schalkoff, 2009] was developed to help determine rule priority, for validation by rule experts. Rivers, the knowledge representation used in eGanges, have rule salience implicit in their structure. In some sense, non-monotonic reasoning is all about rule priorities. For example, Prakken and Vreeswijk [2002, p224] point out that for monotonic reasoning "any conclusion that can be drawn from a given set of premises, remains valid if we add new premises to this set." Unfortunately this does not hold for non-monotonic logics; they give an example where several points of information can

lead to different conclusions as to whether or not a cabinet minister's health issues should be published in a newspaper; in essence the argument comes down to which rules are given higher priority.

4. **Verification and Validation (ensuring completeness and consistency):** The task of verification and validation of knowledge bases was recognised quite early in the AI field [Boehm, 1984]; Boehm summarised verification as “Am I building the product right?” and validation as “Am I building the right product?” and also went on to explain what is required to ensure a specification is complete and consistent. In the context of KBS informally, completeness is concerned with domain coverage and consistency requires that only one conclusion is given by the same set of conditions.
5. **Information exchange interface (including the user-interface)**: This is probably the all encompassing problem that has generally been neglected by the AI community; after all, the issues with KBS validation can be eased with an appropriate expert interface. This means the interface provided to the human expert must be cognitively comprehensible to that human expert so they feel comfortable about declaring the system's reasoning as valid. One of the best ways to do this is with knowledge abstraction [Dondossola, 1999]; Richards [1998, p71] says: “The need to abstract knowledge for the purposes of reuse has been espoused by many and has proven to be important”. Stelzner and Williams [1988] also emphasize the importance of the user interface for the acceptance of expert systems.
   Baroff et al [1988] see graphical user interfaces as the most appropriate for the capture of knowledge because they claim people think in pictures. This view of graphical interfacing is also supported by the quote: “a visual language that is both comprehensible and formal offers attractive possibilities not only for the comprehension but also for the editing, and for parts of the elicitation process itself” [Shaw and Gaines 1991, p.9].
6. **Uncertainty handling**: Any practical reasoning aid must deal with uncertainty, as it will undoubtedly be deployed in circumstances with potentially incomplete or inconsistent knowledge. The method for dealing with this imperfect knowledge must be a mathematically reliable one, which is consistent with the other requirements of the system.

1.2.3 Superexpertise Requirements

This thesis acknowledges that businesses need to have *agile management* in order to compete in the modern world, where they need to rapidly spread changes in knowledge related to their business to their employees, and enable their employees to efficiently and accurately use this knowledge to solve business problems and meet business standards. This requires that businesses use a method to encode and represent this knowledge (i.e. encapsulate the knowledge for transmission), so that knowledge transfer amongst employees is quickly, efficiently and accurately achieved. To do this requires superexpertise of Quality Control problems.

The competitiveness of an expert's business depends on how fast and accurately it can decide on a sufficiently optimal solution, given the potentially vast, imperfect, changing information available to it. An expert's time is money for the expert and the expert's organisation; so any tool used must be easy and quick, as well as useful. Computer decision support tools could help reduce decision time, ensure the decision is more accurate (humans can forget things), help advise what additional information is needed, or advise what inconsistencies exist so the expert can attempt to resolve them.

In short, in addition to the 10 requirements of knowledge representation, and 6 requirements of knowledge based systems, superexpertise requires the ability to:

1) handle vast amounts of information,
2) handle rapidly changing information,
    a. accurately alter the knowledge base,
    b. alter as little of the knowledge base as necessary, and
    c. quickly adapt the knowledge base,
3) handle incomplete information:
    a. quickly determine what additional information is necessary for knowledge completeness, and
    b. still be able to make the most optimal solution choice without complete information,
4) handle inconsistent information,
    a. minimise the chance of inconsistencies occurring,
    b. optimise removal of inconsistencies,
    c. ensure the human expert knows about all inconsistent information and its handling, and
    d. still be able to make the most optimal solution choice with the inconsistent information,
5) choose between a large number of possible solutions,

6) accurately decide on the optimal solution,
7) ensure human experts are confident of the solution's accuracy, and
8) quickly decide on the optimal solution.

Any software tool requires the initial knowledge to be acquired and stored in its knowledge base, and this knowledge base needs to be updated when the knowledge changes. The easier, quicker and more accurately a system can acquire and maintain its knowledge base, the more useful the tool.

The acquisition of knowledge can be a very difficult process, and these difficulties result in the *Knowledge Acquisition bottleneck*. This bottleneck refers to the intrinsic difficulties associated with capturing knowledge such as articulating and representing the conceptual model in a person’s head, i.e. their field epistemology. This bottleneck was first recognised in 1965 by Feigenbaum [Feigenbaum, 1981, p.226; Gillies, 1996, pp.25-31]. While it can not be removed, it can be alleviated through appropriate KA methods and KR techniques. Addressing the KA bottleneck is a crucial requirement to achieve superexpertise.

1.3 Research Aims and Approach

The eGanges expert system shell is a useful example of a software tool that can address all the requirements of superexpertise, for certain domains and types of problems. Validating this claim, at least in part, and extending eGanges if necessary, were goals of this thesis. Specifically, the aim of this thesis was to show the applicability of two epistemologies, called eGanges and NeGame (a complement to eGanges), for three particular types of problems of *Finite Domain Expertise* (FDE), namely, *Quality Control* (QC) problems, *Adversarial* problems and *Negotiation* problems. Initially a qualitative methodology of case studies of superexpertise is relied on, but quantitative methodologies are also used as mathematical characteristics of superexpertise are revealed and developed. An introductory definition of superexpertise is expanded into a model with two major features: large scale combinatorics that require multi-value logic processing. The NeGame epistemology was introduced in this thesis, but requires further work and refinement, along with a logical verification more detailed than can be provided by these limited case studies.

As a new conceptual perspective, superexpertise was posed as an aid to understanding the epistemological computation requirements of the various types of FDE problems. This thesis investigates an evolution of 'how we know what we know' (i.e. the epistemological aspects) within the computer science literature, starting from the expert systems literature, which then evolved into the knowledge based systems literature; the thesis then extended this evolution to found superexpertise literature.

An important goal of the thesis was to carry out an exploratory study of the applicability of the eGanges River Logic [Gray and Gray, 2003] epistemology to these three types of FDE problems. The methodology chosen for this was to select representative problems from the three FDE problem types, and conduct applicability case studies contained in the publications. The exploratory case study research then formed a qualitative inductive base for justifying the applicability and use of the eGanges River Logic epistemology, where appropriate, and helped shed light on the epistemological requirements of these three FDE problem types.

An important result of the Negotiation case studies was the discovery that the epistemological requirements for Negotiation problems were better served by a new epistemology; using the ideas and tools developed, the foundations of this new negotiation epistemology, called NeGame, was laid in this thesis. To a certain extent, eGanges can handle negotiation problems, but the NeGame epistemology is better suited to the task of negotiation.

The research question at the heart of this thesis is:

Does the eGanges River logic approach work as an epistemology for agile knowledge management and dispute resolution, and if not, can this epistemology be altered to become workable?

To explore these epistemologies, an investigation framework is required; the framework posed is called superexpertise, as already described in this chapter. Superexpertise details what is required of an epistemology for it to be workable as a solution to a particular type of problem. With the superexpertise epistemological framework now established, the remaining chapters of the thesis apply it to the eGanges and NeGame epistemologies, with respect to the Finite Domain Expertise concerning quality control, adversarial and negotiation problems.

Moreover, the thesis proposes the eGanges River logic epistemology as an elegant, easy to understand way of dealing with incomplete and inconsistent information. It uses four epistemic truth values: unanswered, uncertain, positive (truth value of true with epistemic aspects included) and negative (truth value false with epistemic aspects included).

Unanswered means answering the node question has not begun, and uncertain means only incomplete or inconsistent information is available to answer the node question. In particular, the law and business quality assurance can be appropriately modelled by this epistemic multi-valued logic. To show this several issues must be addressed.

Using multi-valued logics as the basis of knowledge representations for problems that have only two ontological truths has been criticised [Dubois, 2008]. However, the basis of these criticisms have not considered the true semantic meaning of inconsistency, and rely on the standard naive formulation of inconsistency. In Chapter Three, this thesis will investigate whether inconsistency can be redefined in a more semantically meaningful way, and what consequences are derived from this redefinition. It will be shown that River logic [Gray, 1988; Gray and Mann, 2003; Gray and Gray, 2003] preserves the non-naive inconsistency definition as a contradiction, as required of all consistent logics. It will also be shown that this redefinition and the naive formulation are equivalent in two valued logic.

Along with inconsistency issues, potential compositionality problems have also been raised about multi-valued logics. This thesis shows that River logic is not only compositional, but also decidable and consistent due to the three core heuristics used by eGanges.

Finally, the thesis explores the issues associated with extending the eGanges River logic into fields that require negotiation between parties. The thesis investigates whether the hierarchical nature of River logic can be used to help guide negotiations, and what extensions would be required to help prevent or resolve conflicts. The result is a new epistemology, NeGame.

1.3.1 eGanges epistemology exploration methodology

Chapter Two of this thesis investigates the superexpertise aspects of eGanges to see if it can assist businesses to achieve super-agility, by helping their employees to access automated superexpertise. The epistemic multi-valued logic of eGanges River logic is posed as an appropriate knowledge representation model for some fields of expertise to help overcome the KA bottleneck - even when there are only two ontological truths. The method chosen to confirm the superexpertise nature of eGanges for management and law is to create several eGanges applications in management and law. The papers of Chapter Two detail:

- an anti-spam legislation application
- a partnership legislation application
- a requirements for promotion to senior lectureship application
- an Anti-Money Laundering and Counter-Terrorism Financing legislation application
- a Corporate CEO duties application
- an insulation quality control application

These exploratory case studies form a qualitative inductive base for justifying the use the eGanges epistemology for a wide variety of fields of Finite Domain Expertise; this helps support the proposition that the eGanges epistemology is appropriate for certain fields.

The hierarchical structure of the eGanges River logic Knowledge Representation aids in knowledge acquisition, by enabling knowledge encapsulation and abstraction. The knowledge encapsulation and visual coding used in eGanges River logic may help experts to deal with the potentially massive amounts of potentially imperfect knowledge, that can rapidly change. The number of possible solutions available for each problem may also be extensive, so by abstracting much of this complexity into nested River maps, the expert/user does not need to remember everything at once.

River logic can help deal with all the requirements of superexpertise, and thus provide all the requirements to turn someone with or without expertise into someone with superexpertise, at least within a field of finite domain expertise that has a complete and available eGanges application.

In exploring the range of applicable fields for the eGanges epistemology, it was found that an epistemic adaptation should be made for areas of negotiation-based dispute resolution. This is further covered in the papers of Chapter Four. The heuristics of the software package, Family winner, were first used to extend the functionality of eGanges for negotiation. However, given the core importance of reframing to assist in finding win-win options, the candidate developed a new epistemology, more appropriate to negotiation. Reframing is further detailed in the third paper of Chapter Four. In essence, reframing is used to help determine further options that the parties might not be aware of: this is similar to helping rule makers realise the potentialities of new conflict [Gray, 2007b].

Chapter Four again uses the exploratory case study methodology; using a cohabitation and a civilisation application, to show the evolution of the eGanges epistemology to a more appropriate negotiation epistemology (NeGame).

1.4 Achieving superexpertise with eGanges River Logic Epistemology

This thesis builds on and extends eGanges, which will be shown to meet the requirements of KR, KBS and superexpertise. The eGanges core epistemology is described in detail in the PhD thesis of Dr Pamela Gray [Gray, 2007a], which also discusses ideas similar to River logic, namely the work of Fraunce, C. S. Pierce and Ishikawa. For the purposes of this thesis, a brief overview of the eGanges River logic epistemology will now be provided. Further, its ability to address all the above requirements of superexpertise, at least for quality control and adversarial problems, also will be briefly presented.

In essence, the eGanges epistemology recognises the hierarchical nature of rules in certain domains of expertise. Pages 3 and 4 of paper 2.2 in Chapter Two shows how seven related example multi-valued horn clauses can be combined into what is called a River. Each logic literal in the multi-valued horn clauses is called a node. The conclusion node of six of the horn clauses are the same node as an antecedent node of another horn clause. By drawing the linked horn clauses like a River system, the information from assigning a value to an 'upstream' node can be thought of as flowing 'downstream'.

The multi-valued nature of the nodes is the second most important feature of the eGanges epistemology, and will be further addressed in Chapter Three. Each node is more complicated than a binary proposition literal; eGanges uses a four-valued logic where a node can be assigned the values: positive (resolved: known true), negative (resolved: known false), uncertain ('resolved': incomplete knowledge, possibly caused by 'inconsistent' knowledge) or unanswered (unresolved: no knowledge). eGanges uses arguably the simplest method for dealing with incomplete and inconsistent knowledge (i.e. multi-valued logic); it simply ties the epistemic aspects of truth to multiple truth values. Multi-valued logic will be further explained, and its use justified, in Chapter Three.

As Majkic [2004] noted, Belnap's four valued logic could not use the value BOTH without necessarily introducing inconsistency. That is why he redefined BOTH as *possible* for his four valued logic; this was posed one year after the eGanges four-valued logic was described [Gray and Gray, 2003], with the value *uncertain* logically equivalent to Majkic's *possible*.

The third most important feature of the eGanges epistemology is the underlying heuristics. The epistemic truth value of unanswered is an ideal default truth for the logic, and that is indeed what is used; as a default logic, eGanges is necessarily a non-monotonic logic. The way knowledge flows 'downstream' is controlled by two simple multi-valued truth tables, and a truth lattice ordering with positive at one end and negative at the other; i.e. the epistemic truth values are ordered in a lattice chain [Blyth, 2005] such that positive > unanswered > uncertain > negative. Within a single river stream, the value of the conclusion node is equal to that of the most negative antecedent node. Where multiple river streams share the same conclusion, the conclusion's value is the most positive of all the river streams.

1.4.1 Knowledge representation requirements met by eGanges epistemology

eGanges is built from nodes, where each node represents a single trichotomy (the possible answers are: positive, uncertain or negative). This means for fields of Finite Domain Expertise, eGanges fulfils both *metaphysical adequacy* and *epistemological adequacy*. The underlying heuristics ensure eGanges fulfils the requirement of *heuristic adequacy*, and the finite acyclic nature of the domains suited to eGanges means it is guaranteed to be *computationally tractable*. The default logic, simplicity and truth table priority ensures every node is always assigned one and only one epistemic truth value, thereby ensuring a *lack of ambiguity*, *clarity* and *relevance*. *Uniformity* and *notional convenience* are at the heart of the self-similar nature of the River system. The use of overlapping nodes aids in the *declarative* nature of eGanges, though the acyclic nature of the standard eGanges epistemology does cause potential declarative issues; this can be overcome by using the node linking facility to ensure the nodes remain declarative. In short, eGanges fulfils all the knowledge representation requirements.

1.4.2 Knowledge based system requirements met by eGanges epistemology

Knowledge acquisition and maintenance is easy and intuitive in eGanges due to the simplicity of the user-interface and its alignment to the way that a human expert utilises their expertise in problem solving. The hierarchical nature of the River system enables knowledge hiding via the use of nested Rivers [Gray and Gray, 2003]. Together these both make knowledge acquisition relatively straight forward as the human expert can initially lay out the broad rules that need to be dealt with and then create and enter nested River maps to expand on the details; each nested River map forms a submap of its parent River map (see for example figure 5 on page 7 of paper 2.2, which is a submap of the “operation” node also shown in figure 4 on page 6 of paper 2.2). This nesting of River maps is a self-similar knowledge organisation technique (each nested River map is similar to its parent River map) that can be repeated until all knowledge is acquired. The simplicity and self-similarity of the epistemology should make the acquisition of the knowledge as intuitive as possible.

When the knowledge to be represented changes, the hierarchical nested structure of the knowledge representation helps with moving entire River branches, and limits change to only those branches that actually need to be changed. River maps with unchanged knowledge do not need maintenance beyond potentially relocating them within the River system as a whole. Changed River branches can be relatively easily found for alteration, via the node search facilities, and any changes will not have unexpected consequences in other River branches; only Rivers downstream will be affected by any changes.

The four-valued logic and heuristics of eGanges ensure uncertainty is properly handled, and the design of Rivers and River fans (rules in disjunction) enables the expert to build an intuitive knowledge processing system, where what you see is easily understandable and alterable to get what you want; thanks to the inbuilt explicit negation ability of nodes and the use of conjunctions and disjunction which enable the building of any logical multi-valued truth table (further explained in the compositionality section of Chapter Three). eGanges also ensures only one truth value can exist for any node, and therefore ensures consistency; moreover it always ensures decidability (as shown further in Chapter Three).

In short, the eGanges epistemology also fulfils all the requirements of Knowledge Based systems.

1.4.3 Superexpertise requirements met by eGanges epistemology

This River structure is pertinent to many of the requirements of superexpertise. It is suitable for large scale rule systems and manages the hierarchical structure of complex combinatorics. The self-similar nature of a River system makes it simple to code, and provides a guide for easy node location searches (via River map numbering and node labels). This systematisation makes it easy to store huge amounts of knowledge, while still making it possible to handle rapidly changing information, as only directly affected River branches are altered. In other words, the knowledge is not globally changed but locally adapted as needed.

The inbuilt handling of incomplete and inconsistent knowledge as part of the four-valued logic enables eGanges to handle this, with the River structure providing a guide for how best to deal with the incompleteness and/or inconsistencies. The whole system operates as a guide for users to find as quickly as possible the optimal solution.

eGanges is currently designed to handle single classification tasks. This means that an individual River system is designed to answer only one Final dichotomy/trichotomy (such as "is a person guilty of a specific offence"). However, multiple related River systems can be used to answer as many trichotomies as is required to represent any finite number of possible solutions; node sharing between River systems will minimise repetition of answers.

1.5 Thesis Outline

This thesis contains five chapters. The first chapter provides an introduction, overview of the published papers chapters, and discusses generally relevant literature, as well as how the published papers are linked together. It also introduces the concept of superexpertise. Chapter Two consists of four published papers that introduce the concept of superexpertise in relation to adversarial problem resolution. Chapter Three contains further details about the multi-valued logic aspects of eGanges River logic that is necessary to truly understand Chapters

Two and Four. Chapter Four consists of three published papers that further develop superexpertise for negotiation problem resolution. Chapter Five is the conclusion chapter and discusses what further work needs to be done, perhaps as a future doctoral study.

Chapter Two republishes four international conference papers and details the emerging concept of superexpertise, along with the core heuristics of eGanges, which is designed to handle the superexpertise of finite domain expertise. It starts with the second published paper of the thesis [Gray, Gray and Richards, 2007] which introduces the concept of superexpertise. Next Chapter Two republishes the third paper of the thesis [Gray, Gray and Treanor, 2007] which investigates the application of superexpertise to agile management, and points out that River logic uses directional graphs which can be used for teaching and prompting experts. The third republished paper of Chapter Two is the fourth published paper [Gray and Gray, 2009] that describes how eGanges River logic can be used to ensure coherence and co-ordination of large scale intelligence within a workplace. The final paper of Chapter Two [Gray and Gray, 2011] explores the spherical logic epistemology behind eGanges, and an application to provide agile management for government.

Chapter Three puts River logic and the work of the papers in a multi-valued logic syntax more consistent with standard literature of the computing field. River logic is shown to be a type of defeasible multi-valued logic made of a set of multi-valued horn clauses in a priority list, with three heuristics applied to simplify decidability and ensure a more user-friendly interface via the River analogy. Important potential problems (and solutions) with the use of multi-valued logic are addressed in Chapter Three.

Chapter Four republishes the first, fifth and seventh papers of the thesis, which deal with extending the deductive components of River logic to handle negotiation with a complex topic. [Gray, Gray and Zeleznikow, 2007] introduces a plugin module, Negaid, to extend the eGanges River logic to handling negotiations, by adapting the Family Winner [Bellucci and Zeleznikow, 2006] algorithm to help distribute assets. [Gray, Gray and Zeleznikow, 2009] further explores Negaid enhanced eGanges applications to prevent conflict from arising by ensuring parties have eGanges enforceable rules to deal with future conflicts, rules agreed to prior to any conflict arising. Finally [Gray, Gray and Zeleznikow, 2011] completely reworks the underlying eGanges epistemology to more accurately reflect what is required in negotiation; the resulting new epistemology, NeGame, uses many of the same ideas as Family Winner and eGanges, but in a new more synergised way. Chapter Five summarises the work of the thesis and explains what future work might be done.

Chapter 2: Combinatorics in Superexpertise

This chapter consists of four published papers that introduce the concept of superexpertise in relation to adversarial and quality control problem resolution. To do this the papers discuss some of the details of the core epistemologoy and heuristics of eGanges, which is designed to handle the superexpertise of finite domains of knowledge; further details on the eGanges epistemology can be found in paper 2.4 [Gray and Gray, 2011], Chapter Three and [Gray, 2007a]. Each paper also uses a exploratory example eGanges application to investigate the applicability of the eGanges epistemology as a superexpertise tool. A particularly important class of adversarial problems used in the papers are legal problems; rules of law are used to determine whether the defendant is guilty of a criminal or civil offence, hence the law generally deals with adversarial problem resolution. This introduction section discusses the content of the papers within the context of the combinatorics of superexpertise.

2.1 Motivation for this Chapter

This chapter is motivated by the need to improve business value, i.e. the requirements businesses have to get as good a return on their assets and staff as possible, as well as their need to stay competitive in a global market. The business goals of efficiency and productivity must be achieved in the context of quality control.

Quality Control focuses on [Juran and De Feo, 2010]:

- Attaining superior results through quality
- Assuring repeatable and compliant processes
- Creating breakthroughs in performance
- Accurate and reliable measurement systems and advanced tools
- Empowering the workforce to tackle the "useful many" processes
- Business process management, i.e. creating an adaptable organisation
- Software and systems development: using an AGILE, not Waterfall methodology[3]

These requirements lead to the study of agile management; part of agile management is a requirement for agile knowledge management, which is what this chapter investigates for fields of *finite domain expertise* (FDE), particularly in fields that require super human levels of knowledge. The better a business/government is at achieving its goals, e.g. through a more

3 The Waterfall methodology is where software development progress flows steadily downwards through phases such as Conception, Analysis, Design, etc.

agile handling of its knowledge resources, the more competitive it may be in our increasingly globalised world.

This drive for improved business value through agility has been summarised by McDonald [2007, p.1]: “Organizations need to get more done by doing less and delivering business value. An agile inspired business value focus utilizes strategic intent, understands uncertainty and complexity, real options and constant review of business value of projects.” This need to practically deal with the real options available in an uncertain and complex environment is thus recognised by McDonald as being key to a business' success. These same agile knowledge requirements are addressed in the requirements for superexpertise, discussed in Chapter One, which aims to achieve both halves of the four agile software bullet points in the Agile Manifesto [http://agilemanifesto.org]:

- Individuals and interactions over processes and tools
- Working software over comprehensive documentation
- Customer collaboration over contract negotiation
- Responding to change over following a plan

using an appropriate epistemology as the knowledge management design aid.

There may be some confusion about how compatible agility and quality are. Rooney [2007, p.1] says: “Agile methods make quality a top priority.” This statement may seem at odds with the focus of agility on speed; for example, prioritising working software over comprehensive documentation. However, business agility is all about how rapidly a business can adapt to a changing environment – thus, high quality adaptable software tools can greatly enhance business agility.

To be agile, a business must be able to rapidly spread changes in the knowledge that is used by its staff; knowledge of how it should operate, what it should be doing, etc. However, if it does not ensure a reasonable quality level of that change in knowledge, it will not have adapted to the change, and therefore will not have been agile in adapting to a change. Thus agile management seeks a balance between these, often competing, foci.

When there is an unavoidable conflict between the requirements, one must be given priority. For example, if following a certain process or using a certain tool would preclude some individuals (stakeholders) from participating in the project, then that process or tool should not be used. Similarly, if producing comprehensive documentation would result in insufficient time to get the software properly implemented and tested, a subset of important documentation should be identified and produced. The overall aim and outcome of following the agile manifesto is an emphasis on people (individuals team members and customers) and delivering a working product that meets the customer's needs.

If all agile management requirements can be achieved, i.e. all requirement conflicts can be avoided, and the requirements can be achieved at a 'super human' level, then not just agile management would be achieved, super-agile management would be achieved. The papers of Chapter Two are focused on showing how super-agility can be achieved for adversarial and quality control problems with the use of the finite domain superexpertise tool, eGanges.

2.2 Possibilities and Potentialities

In order to understand paper 2.1 fully, it is useful to consider the notions of possibilities and potentialities. The important epistemic differences between possibilities and potentialities was raised in [Gray, 2007b]. In the language of AnsProlog (see Chapter Three), the possibilities are just the elements of the Herbrand Base, i.e. the set of possibilities is just the set of all grounded atoms. Potentialities on the other hand are all the possible new rule changes that could be made to the AnsProlog program; these potentialities clearly effect the possibilities, however, they are epistemically different to the possibilities alone. The different types of behaviour of potentialities has lead to many different fields of study,e.g. non-monotonic logics explicitly recognise that their epistemologies may potentially result in retractions of previously valid conclusions, given an addition, deletion or other change to the rule set.

The most common problem for human experts in a field that requires super-human levels of knowledge is the possibility of *overlooking possibilities and potentialities*. Even if all the experts would agree about the rules that should be implemented, if they knew about them, the problem is there are so many rules to remember they may not know them all; therefore they may come to contradictory conclusions as contradictions are possible for any non-monotonic logic with different applicable rules. As eGanges is a default multi-valued logic (see Chapter Three), it is a non-monotonic logic where such rule differences can lead to contradictory conclusions.

Any source of rule or knowledge difference can result in contradictory conclusions. For example, assume expert 1 knows $\{ (A \rightarrow C), A \}$[4], expert 2 knows $\{(B \rightarrow \neg C), B \}$ and expert 3 knows $\{ (A \rightarrow C), ((B \wedge \neg A) \rightarrow \neg C), B \}$. Expert 1 would determine C is true (by modus ponens), likewise expert 2 would determine $\neg C$ is true, i.e. that C is false; these two experts disagree due to differences in their knowledge. However, expert 3 could not derive any

4 $\{(A \rightarrow C), A\}$ is an example knowledge base; this example represents knowing $(A \rightarrow C)$ is true and A is true. In Chapter Three, the order of the knowledge base is important for determining rule priority order (salience)

conclusion about the value of C; unless he used something like the Closed World Assumption, which would make him favour nonmonotonically concluding ¬C; of course if he knew everything expert 1 knew, he would conclude C instead.

Of course, not all knowledge is necessarily going to be agreed on by experts, nor is all knowledge going to necessarily be knowable. Such disagreements are classified under ambiguity, vagueness and rule salience/defeasibility. These points were explored in Chapter Three, but possibilities and potentialities are core to the five combinatorics identified in the first paper of this chapter.

2.3 Papers

Each of these papers develops the idea that an expert system shell, eGanges, is an agile management and quality control tool that can augment human experts by handling the massive combinatorics required of fields that need superexpertise due to the large and complex knowledge of those fields. The papers detail how the complexity of the knowledge can be broken down by the hierarchical River logic knowledge representation, ensuring that any area of expertise that needs to answer a single yes/no answerable question (i.e. a dichotomy) can use this knowledge representation (epistemology) to make the job of the expert far easier. Indeed, the resulting expert systems can be used to train experts in the field and be used more as a mnemonic aid, or cue, to ensure all relevant questions and issues are dealt with.

The ideas in the four papers of this chapter expanded on the characteristics of superexpertise from a range of perspectives: types of combinatorics, androgogy, large scale implications and total spherical quality control. The four papers are presented next, each following an overview and contribution summary.

2.3.1 Paper 2.1

The first paper of Chapter 2 is:

Gray, P., Gray X. and Richards (2007): “Godel, Escher, Bach and Superexpertise”, in Z. Zhang and J. Siekmann (eds), Proceedings of 2nd International Conference on Knowledge Science, Engineering and Management, 28-30 November, 2007, Melbourne, Australia, Lecture Notes on Artificial Intelligence, 4798, pp.599-604, Springer-Verlag, Berlin.

The paper identified five types of expert combinatorics which indicate a complexity beyond ordinary human intelligence; as such, combinatorics was shown to be a matter of superexpertise. The automation potential of these five types of combinatorics through the superexpert aid, eGanges, was explained with reference to a legal expert application of the shell. The application pertains to the provision of legal advice on an email's compliance with the Australian Spam Act (2003); legislation all Australian businesses that communicate via email must comply with. A distinction was made between the four types of combinatorics which are suitable for automation, and the fifth type which was unsuitable for automation due to Godel's theorem. Thus superexpertise was shown to be limited to predetermined finite domain expertise (FDE); i.e. FDE fields where all applicable rules are known. This analysis helped determine what was required to created an appropriate epistemic representation of the law: the KA bottleneck and the five types of combinatorics.

The knowledge acquisition bottleneck is caused by the complexity of the combinatorics of expertise. Humans generally only work out part of the full combinatorics; this point is further elaborated in Chapter Four of the thesis. The paper also discusses how an expert "with computational intelligence, processes all possible user cases consistently; it has Superexpertise that can process any case much quicker and more expediently than a human expert." [p1]

The core issue of the knowledge acquisition bottleneck problem was laid out in Quinlan [1979, p.168]: "Part of the bottleneck is perhaps due to the fact that the expert is called upon to perform tasks that he does not ordinarily do, such as setting down a comprehensive roadmap of some subject." This is why in the paper it is explained how a "Superexpert system shell addresses the combinatorics issue by guiding the domain expert to articulate and organize their knowledge via a fifth generation communication system which visualizes and constrains the knowledge entered." [p5].

The five sets of combinatorics can be summarised as: positive rivers, negative rivers, uncertain rivers, interactions of the rivers, and rivers not yet within the defined system. This fifth set of not yet defined rivers is potentially infinite, so "Gödel's theorem invalidates the combinatorics of an automated judge as the fifth set is potentially infinite" [p2]. The notions of epistemologies is further discussed in Chapters One and Four.

"eGanges programming epistemology uses McCarthy's concept of circumscription to partition elements of the fifth set of combinatorics from elements of the first four sets of combinatorics." [p5] By doing this partition, the legal potentialities [Gray, 2007b] can be left to human experts to deal with by updating the other four combinatorics when appropriate.

The three core river types (positive, negative and uncertain) are somewhat similar to Defeasible Logic's three types of rules: strict rules that specify that a fact is always a consequence of another (c.f. Positive rivers); defeasible rules that specify that a fact is typically a consequence of another (c.f. Uncertain rivers); and undercutting defeaters that specify exceptions to defeasible rules (c.f. Negative rivers). These three river types lead to a requirement of at least three possible states for a logic literal: in defeasible logic that is justified, rejected, or neither; in river logic these are respectively positive, negative, uncertain/unanswered. These aspects will be further discussed in Chapter three.

Paper 2.1 Contributions

The candidate identified the five types of expert combinatorics, which indicate a complexity beyond ordinary human intelligence. Accordingly, he showed that combinatorics is a matter of superexpertise. Further he distinguished between the four types of combinatorics which are suitable for automation, and the fifth type which is unsuitable for automation due to Godel's theorem. Therefore, superexpertise was limited, prima face, to predetermined finite domain expertise (FDE). The candidate also contributed to the graphical formulation of ideas of the spam application, and their use in the paper. The numerical contribution breakdown is as follows:

	Pamela Gray	**Xenogene Gray**	Deborah Richards
Concept	25.00%	70.00%	5.00%
Design	35.00%	60.00%	5.00%
Analysis	25.00%	70.00%	5.00%
Writing	55.00%	40.00%	5.00%
Average	35.00%	60.00%	5.00%

Due to copyright restrictions pages 24-29 have been omitted from this thesis. Please refer to the following citation for details of the article contained in these pages.

Gray, P., Gray X. and Richards (2007): “Godel, Escher, Bach and Superexpertise”, in Z. Zhang and J. Siekmann (eds), Proceedings of 2nd International Conference on Knowledge Science, Engineering and Management, 28-30 November, 2007, Melbourne, Australia, Lecture Notes on Artificial Intelligence, 4798, pp.599-604, Springer-Verlag, Berlin.

http://doi.org/10.1007/978-3-540-76719-0_65

2.3.2 Paper 2.2

The second paper of Chapter 2 is:

Gray P., Gray X. and Treanor L. M. (2007): "Clues, Cues, Combinatorics and Super-agile Management Androgogy", in M. Z. Hoque (ed), CD Proceedings of the 7th International Business Research Conference, 3-6 December 2007, University of Technology, Sydney, ISBN 9780980455700 (CD-ROM), "D:\Management\Gray,P N & Others.pdf", World Business Institute, Melbourne, Australia.

This paper investigated the application of superexpertise to agile management, and points out that River logic uses directional graphs which can be used for teaching and prompting experts. It was concerned with the difficulty of teaching combinatorics to adults and translates some of the logic elements of superexpertise into ordinary terms as a superexpert pedagogy and androgogy. The antecedents in the rules that are graphically represented as tributaries linked in a River system were renamed cues, and the arrow representing the inference flow in an eGanges River were renamed clues. The inference clues take the learner to the next cue. As there may be many alternative clues, each leading to different further cues, so there is a basis for understanding the combinatorics. Several applications were shown, indicating a variety of matters that require superexpertise. The applications were: basic rule formalisation, partnership issues, and promotion to senior lecturer requirements. The Partnership application can be used to help ensure potential partnership businesses can start with agility. Finally the paper also mentions a brief analysis of the Australian Taxation Act, concluding that the Tax Act's complexity would hinder business agility, unless there were a super-agile Tax aid.

The paper leads with a quote: "Technology has driven the development of agile management" [Joroff, Porter, Feinberg, and Kukla, 2003, p.293], and then explains how the first paper of Chapter 2 leads the way for eGanges to be used as a super-agile management tool.

This paper focuses on the use of the River logic software tool, eGanges, as a tool to prompt the user with clues and cues of how to deal with the massive combinatoric issues of knowledge management in order to teach adults and handle this knowledge in a very agile way. Clues become cues when they are linked by a directional flow from one node to another in a River. The paper goes on to show how a River system can be built up from a list of rules (which are identified as multi-valued horn clauses in Chapter Three), and how such River systems can be used as Clues and Cues to ensure users don't forget important knowledge when determining how to solve the problem.

The paper tries to provide the core requirement of agile management, which according to

Dove [1999, p.18] is that “Agile management requires an ability to manage and apply knowledge effectively”. Dove's initial definition of agile management was as “the ability of an organization to thrive in a continuously changing, unpredictable business environment” [Dove, Nagel, Goldman and Priess, 1991]. The paper shows how eGanges uses the hierarchical structure in knowledge to become a tool for agile management of knowledge, fulfilling Dove's requirement of effective knowledge management to help an organisation to thrive in a continuously changing, unpredictable business environment.

Paper 2.2 Contributions

The candidate supervised the translation of the knowledge terms into ordinary terms for superexpert androgogy as a solution to the problem addressed by this paper of teaching combinatorics to adults. The antecedents in the rules that are graphically represented as tributaries linked in a River system, are renamed, cues, and the arrow representing the inference flow in the River are renamed clues. The androgogy uses the inference clues to take the learner to the next cue. As there may be many alternative clues, each leading to different further cues, so there is a basis for understanding the combinatorics. This androgogy provides a complexity that can be taught according to the terms of ordinary understanding; further, it builds on the commonly understood concept of a River system as a tributary structure. The candidate was coauthor of the applications used indicating a variety of common matters that require superexpertise. The candidate introduced the De Morgan's laws, 5GL and androgogy aspects of the paper, as well as co-creating all the graphics. The numerical contribution breakdown is as follows:

	Pamela Gray	**Xenogene Gray**	Lyn Traenor
Concept	20.00%	50.00%	30.00%
Design	20.00%	70.00%	10.00%
Analysis	20.00%	70.00%	10.00%
Writing	50.00%	30.00%	20.00%
Average	27.50%	55.00%	17.50%

Clues, Cues, Combinatorics and Super-agile Management Androgogy

Pamela N. Gray *, Xenogene Gray ** and Lyn M. Treanor **

Fifth generation language (5GL) knowledge engineering, which provides automated combinatorics of visualisations of extensive complex logic, offers super-agile aids for agile management. The visualisation is a directional graph suited to fundamental teaching and learning, according to a platform for seamless transistion of an androgogy to an agile workplace. Consideration of this technology as a developmental driver of agility, reveals the limits of agility where rogue practices are entrenched as administrative powerpathy, a form of self-defeating hindrance to agile management. Examples of agility and its limits draw on eGanges super-agile aids for teaching partnership law and for university administration.

Field of Research: Knowledge Engineering, Agile Management

1. Introduction

Technology has driven the development of agile management (Joroff, Porter, Feinberg, and Kukla, 2003, p.293) and some new limits of this development are considered in this paper. Knowledge engineering is now able to provide super-agile management aids, through fifth generation language (5GL) applications that process the combinatorics of streamlined extensive complex logic (Gray, Gray,and Richards, 2007). Applications of the user-friendly shell, eGanges, are used to illustrate these super-agile aids. Attributes of this technology are also identified, in order to develop an androgogy for super-agile management; the same applications can be used as teaching aids, and also as vocational aids in work environments. eGanges was also designed for Personal Digital Assistants (PDAs), including suitably equipped mobile phones (Gray, Gray, Tierney and Treanor, 2006), so that its super-agile management applications may be used pervasively.

For the development of super-agile androgogy, firstly, an eGanges application can be seen as a system of clues, constituted by labelled nodes of a logic map. Secondly, clues become cues when they are linked by a directional flow from one to another or others; a cue carries a flow. The relationship between clues is their cue flow; each clue becomes a cue for understanding the associated flow relationships. Thirdly, there are three choices available to the user for each clue: (1) to establish the clue (Positive), (2) to not establish the clue (Negative), or (3) to establish uncertainty about the clue (Uncertain). Each alternative may carry a different consequent. A user's selection of a set of alternatives for each of the clues is only one of the many possible sets of selections; eGanges' automated combinatorics manages any selection, with the speed and efficiency of super-agile management, to produce the Final consequent of any selected set. Agile managers can process one set of selections; the automation of the combinatorics of all possible selections is super-agile management.

*Centre for Research into Complex Systems, Charles Sturt University, Bathurst, NSW 2795, Australia; pgray@csu.edu.au
**Grays Knowledge Engineering, Sydney, Australia; E-mail: {xen,L.Treanor}@grayske.com

After understanding the clues and cues, the user comes to appreciate the combinatorics, that is, the significance of alternative choices and the totality of their effect. The sequence of developing understanding from clues to cues and then to combinatorics, is also the sequence implemented in the development of an application by an agile manager; eGanges is a user-friendly program for easy construction of an application. Clues are identified and linked in such a way that the automation of the combinatorics produces accurate advice for informed decisionmaking or action.

Agile management rests on quality, as noted by Rooney (2007, p.1):

> “Agile methods make quality a top priority.“

eGanges maps are Ishikawa (1985) quality control fishbones, that may be nested as required by the extent and complexity of the management information. In eGanges, they are called Rivers, because they have a tributary structure that is typical or paradigmatic of river systems. The River maps of eGanges also may be shaped in the construction of an application, by a cognitive art that produces mnemonic aids. Like a river system too, an eGanges logic map flows downstream and its streams have confluence, flowing together to a Final result, which may be determined or adopted as a strategic outcome.

On the basis of quality, it is also recognised (McDonald, 2007, p.1) that agile management requires the following:

> “Organizations need to get more done by doing less and delivering business value. An agile inspired business value focus utilizes strategic intent, understands uncertainty and complexity, real options and constant review of business value of projects.”

With the aid of eGanges applications that specify extensive complex management knowledge, for electronic combinatorics, an agile manager has the necessary information readily available and, by doing less in this regard, can get more done expediently. eGanges applications utilize strategic intent, specify and manage uncertainty and complexity, show real options, and permit constant review of business value. The software is a business logic system. In an age of science and technology, agile managers and administrators must know how to deal precisely with extensive complex logic; reliance on common sense and intuitive understanding is no longer adequate.

2. Quality control logic

A fishbone represents extended deductive Major premises that lead to a single Final consequent. Each stream is a premise in the form of a conditional proposition or rule: if (antecedent(s)) then (consequent). For example: if node a, node b and node c then node d. In formal logic this is represented as: $a\ b\ c \rightarrow d$. The inference arrow becomes the directional flow arrow in a River. In a system of rules that can be constructed as a River, overlaps of antecedents and consequents, in the Major deductive premises, which permit extended deduction, take the form illustrated in Figures 1-3.

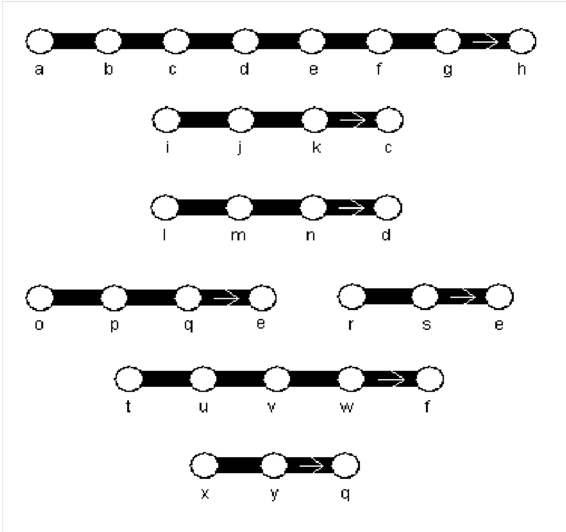


Figure 1: List of wholly formalised rule streams of a River system.
© Pamela N. Gray, 2003

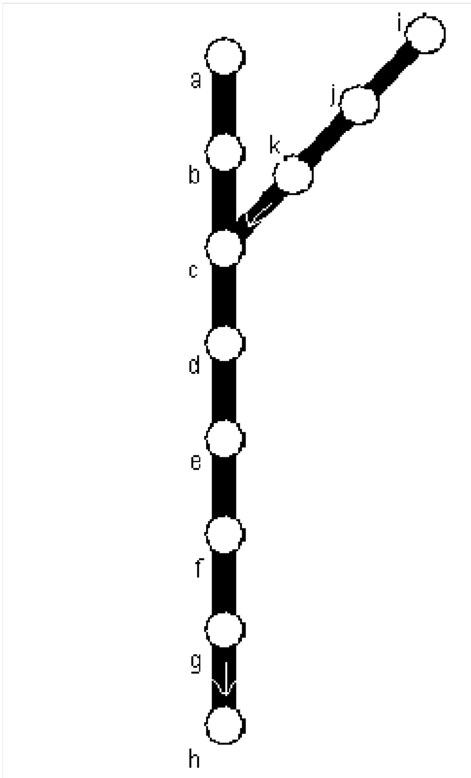


Figure 2: Two formalised rule streams of a River system locked together.
© Pamela N. Gray, 2003

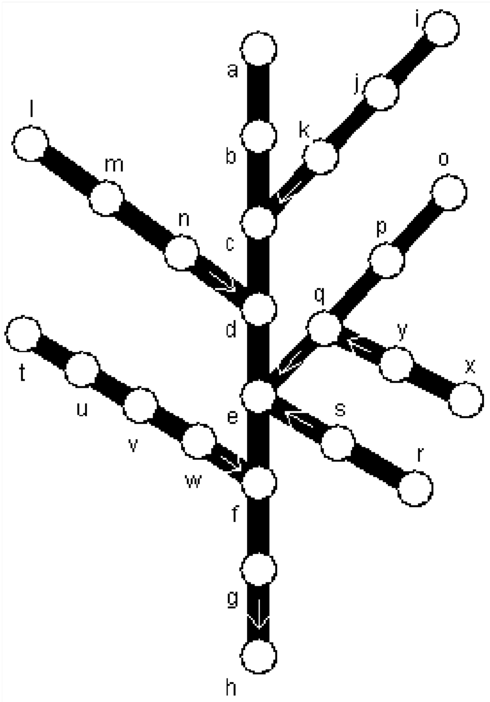


Figure 3: River map of formalised rule streams.
© Pamela N. Gray, 2003

The overlap of antecedents and consequents in Figure 1 can be seen in clues c, d, e, f and q which appear in one stream as an antecedent, and in another stream as a consequent. Clue e also appears as a consequent in more than one stream; overlapping consequents indicate a disjunction, i.e. alternative ways of establishing the same consequent.

Figure 2 illustrates how rule streams may be locked together at overlaps to form a River system. Figure 3 shows the complete locking together as a River system of all the rules in Figure 1; the disjunction at clue e, the only positive disjunction in the system, is shown as a fan structure. Either of the alternative ways of establishing e are consistent with the Positive Final result, h. Positive fans represent another set of choices in the eGanges epistemology, which produce positive combinatorics that are processed in eGanges along with the clue choices of positive, negative and uncertain.

In accordance with de Morgan's law, a positive disjunction or fan corresponds to a negative conjunction or single stream. For example, the negative conjunction of the e fan would be (not o or not p or not q) and (not r or not s) → not e; if o, p, or q fails, then (not o or not p or not q) is established. If either r or s fails, then (not r or not s) is established. A mix of conjunctions and disjunctions in the knowledge complicates the combinatoric processing heuristics; disjunctions that occur in different parts of a River hierarchy add to these complications. Thus q will fail if either y or x fail.

Similarly, a positive fan corresponds to an uncertain conjunction. eGanges combinatorics prioritises negative and uncertain effects on the Final result; In a positive fan, a selected uncertain alternative takes priority over a selected negative alternative, where there is no positive alternative established. Otherwise, in conjunctions, negative selections take priority over uncertain selections. However, sometimes the knowledge contains neutral clues whereby any selection, positive, negative or uncertain, is consistent with a Positive Final result. This is why the three possible answers presented to the user in the eGanges window, may all appear as positive answers.

If any clue in the system is not established, then the Positive Final result, the strategic outcome, fails, except where a fan provides alternatives. eGanges uses combinatoric heuristics, called *pro tem* logic processing, to allow fan alternatives to be exhausted before the Positive Final result fails. These heuristics manage hierarchies of fans, no matter how extensive or complex, to ensure that all choices are processed.

For the purposes of the androgogy, the logic map of an application remains the same aid, from the outset of learning through clues, cues and combinatorics, to its vocational use in agile management, although it may be improved or enhanced with experience.

5GL programs are characterised by visualisation and processing constraints. They are new solutions to the current view of agile management as requiring an ability to manage and apply knowledge effectively (Dove, 1999, p.18). Dove redefined the initial definition of agile management in terms of knowledge management, whereas it was originally defined as (Dove, Nagel, Goldman and Priess, 1991) 'the ability of an organization to thrive in a continuously changing, unpredictable business environment'.

3. Androgogy maps

Business students are introduced to an application as a picture of clues to be learned. It may be that students naturally approach any new subject as initially dealing with information as clues to some richer learning that they will eventually discover. Once familiar with the clues, students begin to explore the hierarchical structure of conjunctions and disjunctions of conditions that are necessary to reach a particular outcome. The hierarchy may make it difficult to see the total alternative flow on effects of available selections. Understanding grows with free navigation of the information, and experimentation with an application that is like a game. This is preferable to laborious detailed demonstration and exhaustive instruction. Students can guess at the combinatoric heuristics of the quality control logic. It is this facility that makes eGanges a super-agile learning and teaching aid.

Figure 4 shows an eGanges map in the Rivers window of the eGanges interface, used to teach the rules of partnership law at Charles Sturt University. These maps further detail a clue in a CEO map known as the TV aerial (Gray, Gray and Treanor, 2006), shown to students to indicate where in business management the partnership knowledge fitted. In answer to an assessment question that required a determination of whether a partnership existed between fictitious people, Callum and Pierre, one student wrote:

> “There is only a possibility of 2 partners so the partnership would not exceed the maximum numbers. The minimum number of partners is 2 this means that the partnership fits this criteria. It is made of two Natural people Callum and Pierre.”

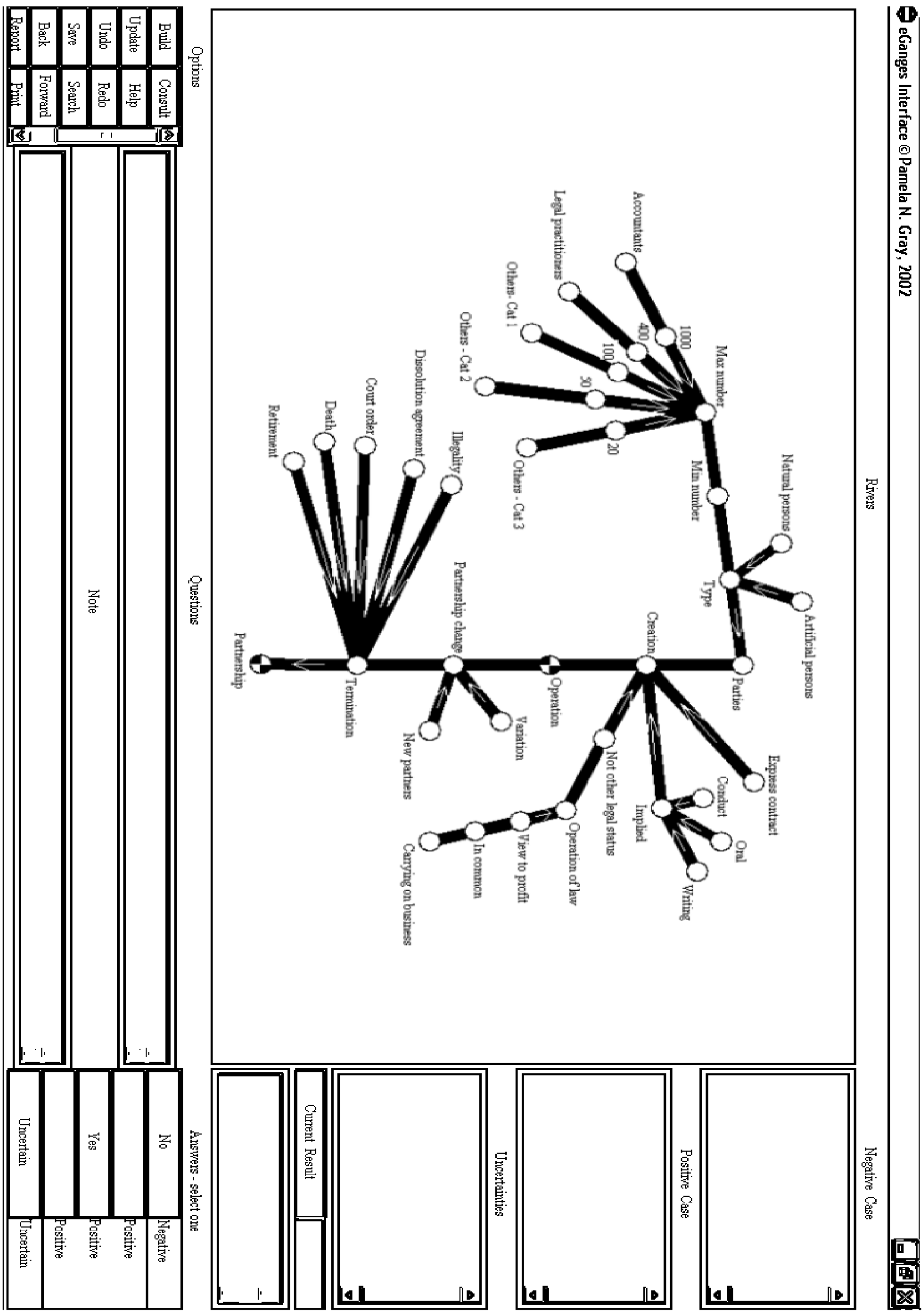


Figure 4: Initial Partnership map in eGanges interface.

The student's failure to punctuate this paragraph may indicate a lack of grammatical skills. The map requires minimal language skills, and the student is able to articulate its complex logic. Figure 5 is the nested map of the soccerball clue, Operation. Soccerballs indicate a further submap. eGanges has no limit to the submapping depths available. This partnership aid can be taken into a real partnership workplace on completion of the course, and continue to be useful. This raises a new androgogy, namely a seamless form of learning from first study to vocational aids.

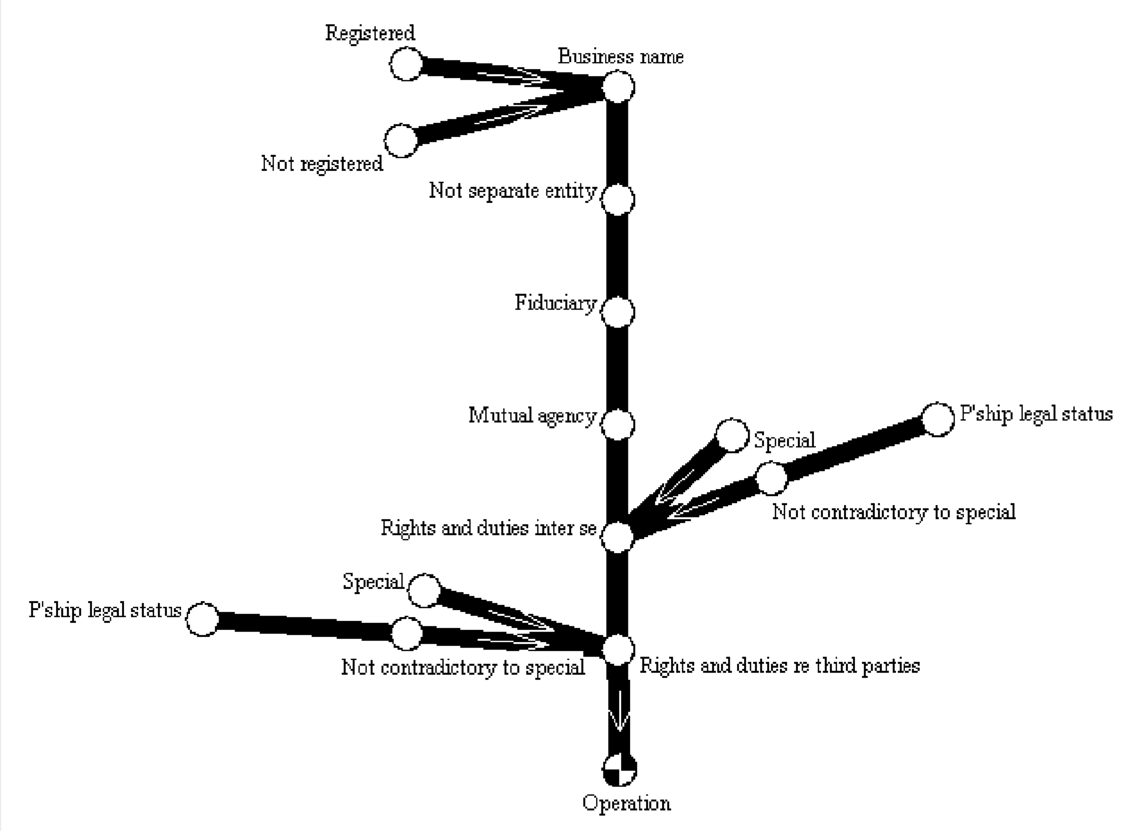


Figure 5: Operation submap

4. Agile Administration

It might be expected that universities, as knowledge businesses, would provide model agile administration systems. The Economist (2007) recently complained that the research output of business schools is irrelevant to the real commercial world. The design of eGanges was produced in a postgraduate research project (Gray, 2007, Gray and Gray, 2003). However, university administration, like democratic government, is not steeped in agile systems. Dove (1999) suggested an agile system as a 'corporate university' model. eGanges makes a different contribution, which tests some limits of agile administration.

The Administrative Manual of one university studied, has promotion criteria for each of four different levels of academic promotion, that involve complex hierarchies of conjunction and disjunctions. One of these levels is Level C. Criteria for promotion from lecturer Level B to senior lecturer Level C are set out in the Manual:

4 Qualifications

4.3 Level C

a) A doctoral qualification relevant to the discipline area; or equivalent accreditation and standing; and
b) A record of significant achievement relevant to the discipline area, and at a national level, in the scholarship of teaching and/or research or professional or creative work.

5 Standards and Expectations

5.3 A Level C academic is expected to:

(a) make a significant contribution to his/her discipline at a national level;
(b) expand knowledge or practice in his/her discipline through original contributions in the scholarship of teaching and/or research or professional or creative work;
(c) provide leadership in his/her discipline in the scholarship of teaching and/or research or professional or creative activity;
(d) promote high quality in course delivery and development; and
(e) have the capacity to:

- coordinate award programs of CSU, or to lead and manage small research teams; and
- teach effectively at all levels, including higher degree supervision.

The natural language of these rules contains a hierarchy of conjunctions and disjunctions, which makes it difficult to see both the options available to promotion applicants, and how the committee should evaluate an application. Figure 6 is an eGanges map of the senior lecturer, Level C criteria, which shows clearly the options available and the necessary requirements; it omits the option of PhD equivalence in 4.3 a, as a discounted choice, for simplicity sake.

One Head of School in the university, whose School had never had a successful promotion application, was of the opinion that it would take an applicant a year to prepare an application; spending this amount of time might considerably reduce recent research output, crucial to satisfy the criteria. If the first application for promotion prepared over a year was unsuccessful, each further application prepared over subsequent years, would add to the Catch 22 problem of diminishing loss of opportunity to satisfy the criteria.

Agile management theory emerged from management systems, including business communication systems and quality control, developed before 1990, to strip away hindrances to speedy informed decisions and actions. Dove (1999) described agility by an analogy to a cat's appropriate leaps. However, he also observed that sometimes a cat gets caught up a tree.

With the streamlining in Figure 6, agile administration may replace the logical confusion created by the wording of dense choices in the criteria statement; dense alternatives may produce a knotting of logic that hinders the application of the administrative provisions. For instance, where alternatives are not clear, they may be treated wrongly as conjunctions, imposing a greater onus on an applicant than the rules require.

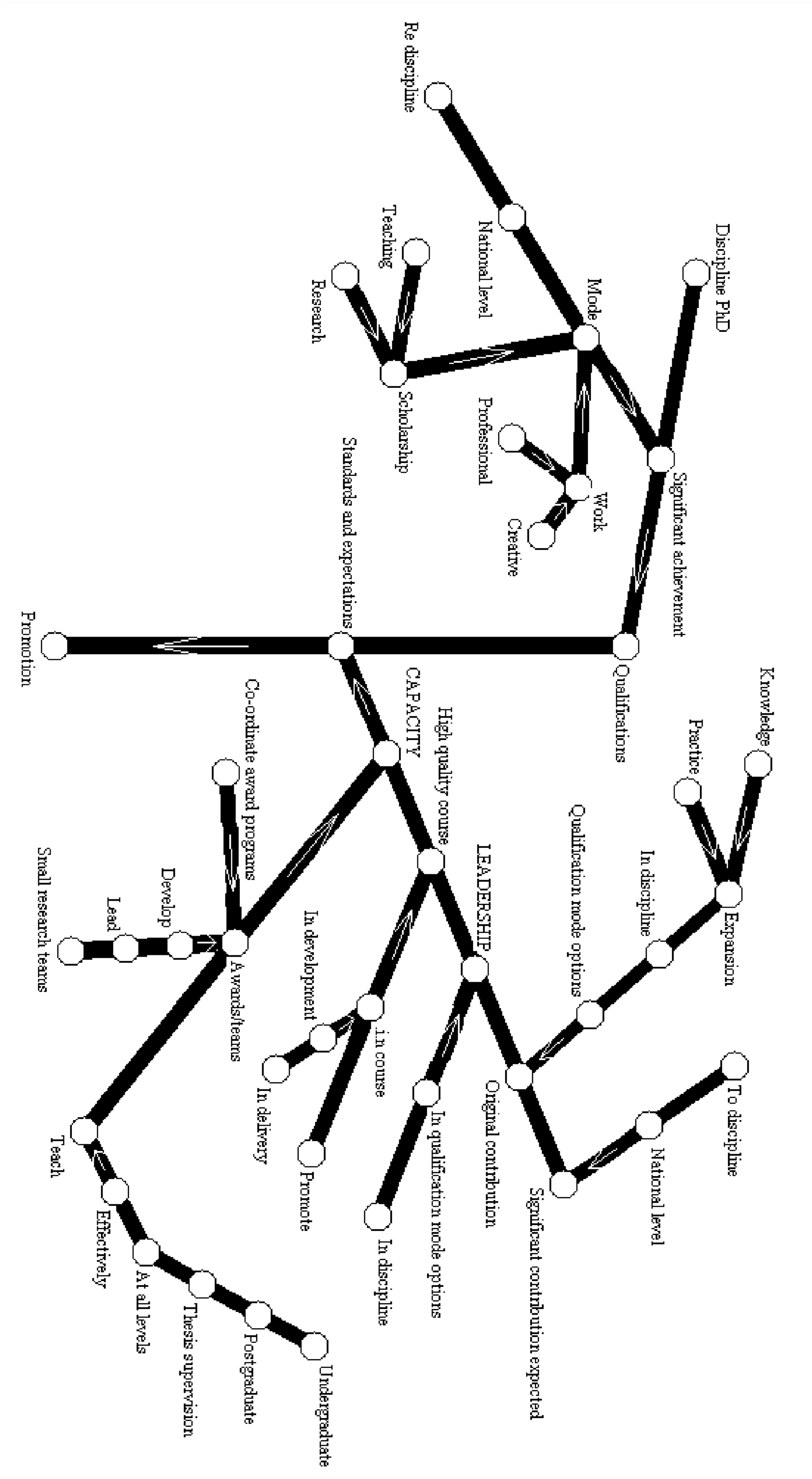


Figure 6: Senior lecturer level C criteria

As an example of a complex hierarchy of disjunctions, Figure 6 may reduce the time required to prepare an application, or free up that time for more detailed preparation. However, it may be that the Head of School is still correct, and the logic map has simply served to show that agile management may only be achieved by reframing the promotion system to remove the powerpathology of a cat up a tree. Thus, the eGanges map illustrates the limits of agile administration.

However, eGanges makes provision for inductive and abductive premises as well as the Major deductive premises of the River maps and the Minor deductive premises of the user's selection input in answer to clue questions. It allows for glosses of the range or spectrum of inductive instances that particularise a deductive antecedent, and also glosses of the abductive reason or justification for a deductive clue, or any of its flow connections or cues. These glosses are stored data that is accessible at the pertinent clue point in a navigation of the River system. A further cue for every clue is the associated inductive and abductive premises.

If administrative information does not contain gloss information as business value data, then there can still be a gloss assessment of the clues and deductive cues. Such assessment might unravel any inagility or powerpathy of the managerial system. For instance, Figure 6 reveals the dual basis for promotion of qualifications and expectations/standards. Qualifications contains certain achievements that are not normally considered qualications in an academic domain. These achievements are difficult to distinguish from the achievements required under expectations/standards. Qualifications seem to be fact-based, whereas expectations/standards seem to require further facts than those established under qualifications; these are facts that justify expectations and future attainment of standards. Qualification facts alone are not enough to justify expectations and attainment of standards; but what justifies expectations and the attainment of specified standards is rather elusive in terms of an inductive base, and the volume of inductive instances required.

Even if expectations and confidence in the attainment of standards could be established, actual realisation of these, following promotion, can only be tentative. Promotion is not a reward or a recognition of achievements and standards already attained; it turns upon an assessment of likely future performance. There is an element of guessing the future in the process, and the applicant carries the burden of this. The criteria pose uncertainties in the expectations/standards, irrespective of qualifications and existing achievements. It is the wide scope of uncertainty to be countered that hampers the application process, and allows covert exercise of wilful discretion by the decisionmakers. Inagility arises from engendering a lack of trust. An agile organisation requires trust beween the people within it. As Owusu (1999, p. 108) observed, one of the manager's primary tasks in a world-class agile management system is embedding trust between managers and workers.

Another aspect that becomes clearer from the Figure 6 map is the interpretation problem in respect of the research criteria located in the teaching area of expectations/standards. Given the maxim of interpretation, *noscitur a sociis*, it might be thought that it is teaching research that is required here; otherwise it appears anomalous to allow a research alternative in the teaching requirements. The uncertainty of the requirement adds to the inagility.

The Australian Income Tax Assessment Act 1936, as amended, which takes up about 5.6MB of Plain Text, exhibits a similar self-defeating hindrance to agile administration, since the provisions for deductions are so dense that any deductions that might be claimed are likely to be less than the cost of working them out. If this frustration of claims for deductions is intentional, then this is rogue legislation, exploiting inagility.

Gloss construction in eGanges, permits business value review, and further development of agility in administration. eGanges is an acronym for electronic Glossed adversarial nested graphical expert system shell. It is the adversarial deduction that introduces combinatoric explosion to the logic; glossing can evaluate substance and complexity.

5. Conclusion - 5GL Androgogy

On a PDA, eGanges brings quality control to any time and place, so time is never wasted and opportunities are never lost; quality control logic is a source of agility. Nor is time wasted on difficult language and complex logic; shared super-agile aids are also a communication system (cf. Owusu, 1999), requiring minimalistic translation into other languages, as a common basis for negotiation of complex understanding.

References

Dove, R., 1999: Knowledge management, response ability, and the agile enterprise, *Journal of Knowledge Management* Vol. 3, Issue 1, pp. 18 – 35.

Dove, R., Nagel, R., Goldman, S. and Priess, K., 1991: *21st Century Manufacturing Enterprise Strategy – An Industry Led View*, Lehigh University, Bethlehem, PA.

Economist 2007, August 28th: Economist.com

Gray, P.N., 2007: Legal Knowledge Engineering Methodology for Large Scale Expert Systems, PhD thesis, University of Western Sydney.

Gray, P. N. and Gray, X., 2003: A Map-Based Expert-Friendly Shell, in D. Bourcier (ed), Legal Knowledge and Information Systems, IOS Press, Amsterdam [www.jurix.nl/pdf/j03-06.pdf]

Gray, P. N., Gray, X. and Richards, D., 2007: Gödel, Escher, Bach and Superexpertise, in *Lecture Notes in Artificial Intelligence (LNAI 4798)*, Springer, Berlin, Germany.

Gray, P.N., Gray, X. and Treanor, L. M., 2006: ‘Handheld Intelligence for contemporary free trade’, in *Proceedings of the International Conference on Contemporary Business*, Leura, Australia.

Gray, P.N., Gray, X., Tierney, R. and Treanor, L. M., 2006: ‘eGanges: Pervasive Peacemaker’, in *Proceedings of the First International Symposium on Pervasive Computing & Applications*, Urumchi, China, August, 2006.

Ishikawa, K., 1985: *What is Total Quality Control? The Japanese way*, translated by David J. Lu, Prentice-Hall Inc., Englewood Cliffs, N.J., USA.

Joroff, M.L., Porter, W.L., Feinberg, B and Kukla, C., 2003: The agile workplace, *Journal of Corporate Real Estate*, Vol. 5, Issue 4, pp. 293 – 311.

McDonald, K. J., 2007: A Business Value Focus for Portfolio Management, *Cutler Consortium, Agile Project Management Executive Summary*, Vol. 8, No. 5.

Owusu, Y.A., 1999: Importance of employee involvement in world-class agile management systems, *International Journal of Agile Management Systems*, Vol. 1, Issue 2, pp. 107 – 115.

Rooney, D., 2007: An Organizational Structure for Agile Projects, *Cutler Consortium, Agile Project Management Executive Summary*, Vol. 8, No. 5.

2.3.3 Paper 2.3

The third paper of Chapter 2 is:

Gray, P. and Gray X. (2009): “The Science of Intelligence for Big, Complex Work”, International Journal of Advanced Corporate Learning, Vol 2, No 2, pp.13-19. This is a revised reprint of the publication, Gray, P. and Gray X. (2008): “The Science of Intelligence for Big, Complex Work” in Proceedings of the First International Conference on E-learning and the Workplace, Columbia University, NY

Paper 2.3 addressed the requirements for superexpetise in the workplace; big complex work warrants superexpert aids. It continues the exploration of superexpertise case studies, this time in relation to superexpertise's potential impact as an aid for society as a whole; both for education, c.f. Paper 2.2, and increasing humanities ability to deal with complex knowledge through appropriate superexpertise tools. The paper describes how eGanges River logic can be used to ensure coherence and co-ordination of large scale intelligence within a society, by using eGanges as an e-learning tool. A very large eGanges' application of the Anti-Money Laundering and Counter-Terrorism Financing Act (AMLCTF) (2006) was used to demonstrate the applicability of the eGanges epistemology to important areas of government administration. The AMLCTF Act pertains to the administrative tasks of an Australian federal agency, namely AUSTRAC, and is again legislation all Australian businesses must be familiar with; legislation that poses big and complex adversarial problems for businesses.

Paper 2.3 shows how the eGanges androgogy aspects introduced in the second paper can be extended to provide an elearning aid for the workplace. It shows how a big complex important piece of legislation relevant to businesses, namely the Anti-Money Laundering and Counter-Terrorism Financing Act (2006), can be made understandable and accessible to the workforce via an eGanges application, as shown in the paper. By demonstrating the applicability of the eGanges epistemology to this complicated adversarial problem, the paper's exploratory methodology helps form an inductive basis for the use of eGanges as an appropriate epistemology.

Paper 2.3 Contributions

The third paper showed a large scale, complex superexpetise required in the workplace of an important area of government administration, namely AUSTRAC. The application illustrates that superexpertise may be very large scale with extensive complex combinatorics that warrant superexpert aids. The ideas in the application were captured from the Anti-Money Laundering and Counter-Terrorism Financing Act (2006). The candidate collaborated extensively in the production of this application and its use in the paper, was co-creator of all graphics and main contributor to section 'IV. Science of Intelligence' and 'V. Use of eGanges androgogy'. He helped construct the short survey, though its optional nature resulted in statistically insignificant quantitative results. The numerical contribution breakdown is as follows:

	Pamela Gray	**Xenogene Gray**
Concept	40.00%	60.00%
Design	35.00%	65.00%
Analysis	45.00%	55.00%
Writing	60.00%	40.00%
Average	45.00%	55.00%

Due to copyright restrictions pages 46-52 have been omitted from this thesis. Please refer to the following citation for details of the article contained in these pages.

> Gray, P. and Gray X. (2009): “The Science of Intelligence for Big, Complex Work”, International Journal of Advanced Corporate Learning, Vol 2, No 2, pp.13-19. This is a revised reprint of the publication, Gray, P. and Gray X. (2008): “The Science of Intelligence for Big, Complex Work” in Proceedings of the First International Conference on E-learning and the Workplace, Columbia University, NY.
>
> http://www.aija.org.au/Law&Tech%2008/Papers/Gray.pdf

2.3.4 Paper 2.4

The fourth paper of Chapter 2 is:

Gray, P. and Gray, X. (2011): “Quality Controlled Government with Spherical Logic” in International Journal of Interdisciplinary Social Sciences, Vol 5, No 10, pp.271-284.

The fourth paper placed emphasis on quality control aspects of superexpertise. It shows an application to illustrate the quality control structure required for administration of the Australian federal home insulation scheme; the lack of quality control of this scheme encouraged unqualified insulation installers, several of whom died on the job. The complete quality control structure in superexpertise was shown in the spherical logic graphics [Gray, 2007a] which were the basis of the object-oriented programming of eGanges. The logic object of the sphere was implicit in superexpertise and provided programming epistemology for superexpert aids. Although combinatorics may be extensive and complex, spherical heuristics keep the processing compact; eGanges is a very small shell, about 600kB that works with Java. Validation of the eGanges specifications required a translation of the spherical graphics into logic terms, via the programming processes.

The paper refers to the work of Harré and Krausz [1996] who discuss what exactly knowledge is: “knowledge is justified true belief” [Harré and Krausz, 1996, p68]. It also touches on epistemic aspects: “Perhaps the difference in epistemic attitude of our two believers can be reduced to a case of differences in … the conceptual systems they employed to formulate the beliefs they took” [Harré and Krausz, 1996, p69].

The paper also states that with globalisation, lawmakers and administrators need to upgrade their precision and effectiveness to manage an increasingly complex and dynamic world. Improved frameworks, concepts and structure for operational effectiveness of government and administration are required. eGanges accommodates information as large and complex as required, and helps achieve these requirements.

This paper show some of the development of quality control, tracing it from the Japanese fishbone [Ishikawa, 1985] to the spherical artificial intelligence of eGanges [Gray, 1990; Gray, 2007a]. This paper summarises the legal epistemology which eGanges was based on, i.e. spherical logic. The paper then goes on to detail a new eGanges application to bring quality control techniques to government policies, in particular the home insulation policy.

The paper shows that the eGanges River is an advance on Ishikawa's fishbone. The fishbone suits a strategic focus on a final objective, with a tributary hierarchy demarcated by goals and targets to achieve this objective. The fishbone is a practical logic based on a cause and effect epistemology of manufacturing; it may be seen as a hierarchy of hypothetical propositions for use in a hierarchy of *modus ponens* syllogisms for deduction of the Final result. eGanges is a multi-valued logic system with dichotomies along the tributaries, and heuristics to process answers to the dichotomies that resolves the answer to the final objective dichotomy; the main reason for eGanges using a multi-valued logic are explained below.

Paper 2.4 Contributions

Although combinatorics may be extensive and complex, the heuristics developed by the candidate, from spherical logic, keep the processing compact; eGanges is a very small shell, about 600kB that works with Java. In this paper, the candidate identified the complete quality control structure in superexpertise by reference to total spherical quality control. The candidate also showed that compact quality control processing supports agile management and collaborated on creating the insulation application.

The paper used a home insulation application to illustrate the quality control structure required for administration of the federal scheme; it is shown in the spherical logic graphics which were the basis of the object-oriented programming of eGanges. The logic object of the sphere is implicit in superexpertise and provides programming epistemology for superexpert aids. The numerical contribution breakdown is as follows:

	Pamela Gray	**Xenogene Gray**
Concept	35.00%	65.00%
Design	40.00%	60.00%
Analysis	45.00%	55.00%
Writing	60.00%	40.00%
Average	45.00%	55.00%

Quality Controlled Government with Spherical Logic

Pamela N. Gray, Charles Sturt University Bathurst, New South Wales, Australia
Xenogene Gray, Macquarie University, New South Wales, Australia

Abstract: In the management of large enterprises, quality control has been implemented since it was successfully introduced to Japanese manufacturing following the Second World War. It began with the Ishikawa fishbone graphic which was a practical advance on the Porphery tree that captured the logic of Aristotelian ontology. The fishbone suited a strategic focus on a final objective, with a tributary hierarchy demarcated by goals and targets; it was like a logic River system that could be contrasted with contradictories, uncertainties and consistencies. Graphical representation of the totality of these contrasts takes the structure of a sphere. Spherical logic provides total fishbone quality control for the quality control of complex governance and administrative matters. In an age of globalisation, science and technology, lawmakers and administrators must upgrade their precision and effectiveness to manage an increasingly complex and dynamic real world. This is a practice-focused paper that poses and demonstrates improved frameworks, concepts and structures for operational effectiveness of government and administration. Now, optimal civilisation can be designed for sustainability with collective intelligence aids such as eGanges, which is a total fishbone quality control system. The capacity of eGanges accommodates information as large and as complex as required, and processes the inherent combinatorial options of pro tem possibilities as well as data retrieval of information about potential and expectations for change or adaptation, at its precise point of relevance. User-friendly eGanges applets can be accessed online by the public. A draft application of eGanges is used to illustrate total fishbone quality control in governance. The application shows the detail of requirements for quality controlled home insulation, to give effect to a recent government-funded home insulation scheme in Australia. The problems which arose from the scheme, common knowledge established by the media, namely installer deaths and home destruction by fire, indicated a lack of quality control in the formulation and implementation of the scheme. The scheme was to give effect to a policy of reducing the use of coal-produced domestic electricity. A sound legal ontology of home insulation was required for the scheme, to optimise its effectiveness, avoid negligence, risks, and unnecessary costs of exploitative entrepreneurs.

Keywords: Quality Control Government, Spherical Logic, Ganges, Fishbone, Sustainable Civilisation Design, Legal Expert Systems

Quality Control

QUALITY CONTROL HAS been implemented effectively in the management of large enterprises, since it was successfully introduced to Japanese manufacturing following the Second World War. The skills of quality control in government turn upon the management of large scale, detailed information that can give rise to massive combinatorics in the application of rules of law to the diverse instances that occur in the real world; the same skills may be required for business rules and other rules of knowledge or expertise.

The International Journal of Interdisciplinary Social Sciences
Volume 5, 2011, http://www.SocialSciences-Journal.com, ISSN 1833-1882

COMMON GROUND

The development of quality control is traced from the Japanese fishbone (Ishikawa, 1985) to the spherical artificial intelligence of eGanges (Gray, 1990, 2007). A contemporary application of eGanges, namely Home insulation, demonstrates the potential use of the technology to aid in quality controlled governance and administration with increased user-friendly public access. A test for the quality of government is the extent of compliance with laws by its subjects. Compliance by the subjects of government requires knowledge by them of what is required. As law becomes increasingly extensive and complex, compliance, as a requirement of quality control, can only be achieved through online technological aids. Otherwise the fundamental principle of the legal system that ignorance of the law is no excuse and therefore no defence in a court action, will produce expanding injustices and, ultimately, a breakdown of civilization through an incoherent society.

Fishbone

After the Second World War, with American assistance in the 1950s, quality control was initiated in Japan with the fishbone graphic of Ishikawa, shown in Figure 1 (Ishikawa, 1985, p.63). Ishikawa was concerned to produce a quality standard in manufacturing and his fishbone set out the matters (causes) that had to be managed to achieve this objective (effect). Five matters are identified in generic terms, and the fishbone graphic showed, by tributaries of details, how there might be further particularisation of each: (1) material, (2) man, (3) machine, (4) method and (5) measurement.

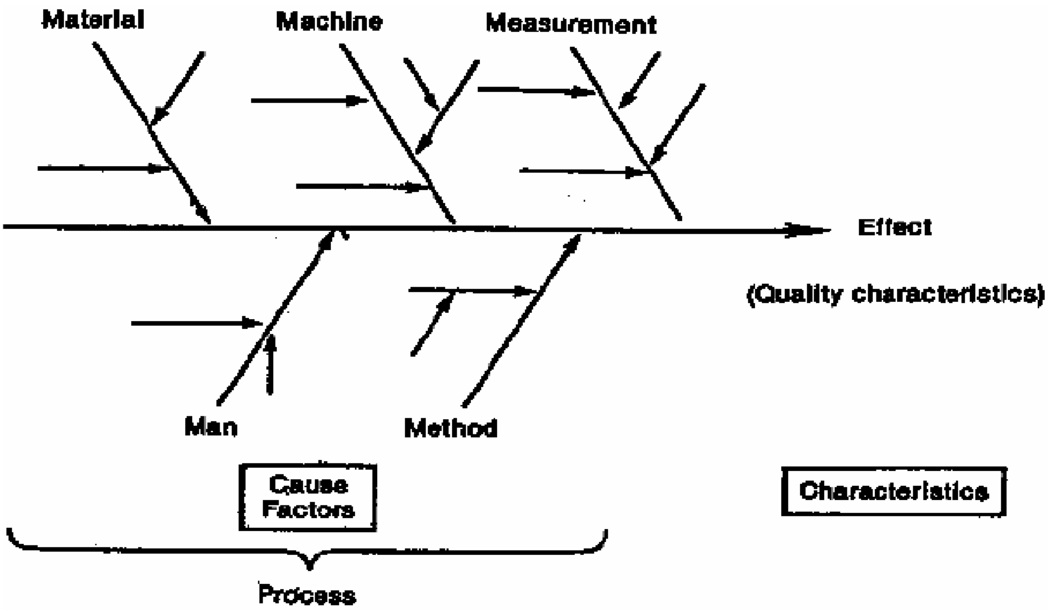


Figure 1: Ishikawa (1985, p.63) Fishbone, Cause and Effect Diagram

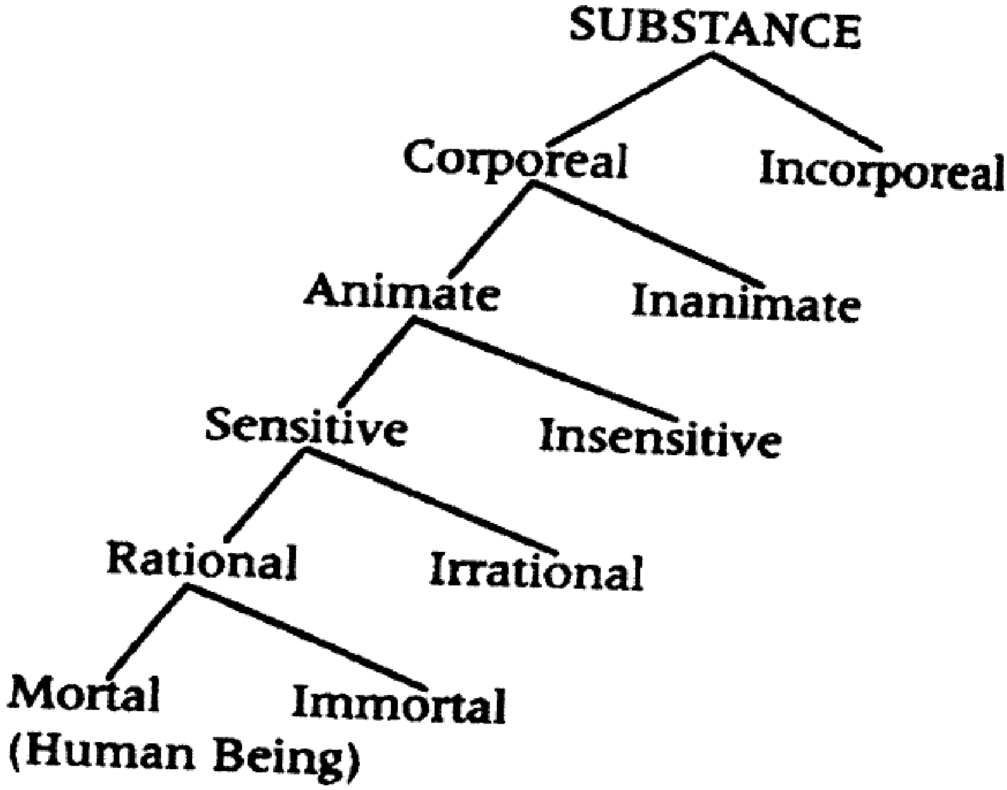


Figure 2: Porphyry's Tree

The fishbone was a practical advance on Porphyry's tree (c.300AD) that captured the logic of the Aristotelian ontology of substance with a hierarchy of genera, species and so on. Figure 2 shows the Porphery Tree and Figure 3 shows Horrock's (2005) annotations of it to assist in developing the semantic web. The generalisations and inductive instances of Aristotles' ontology in a Porphyry's Tree, illustrate the logical validity of his arguments such as:

All men are mortal;
Socrates is a man;
Therefore Socrates is mortal.

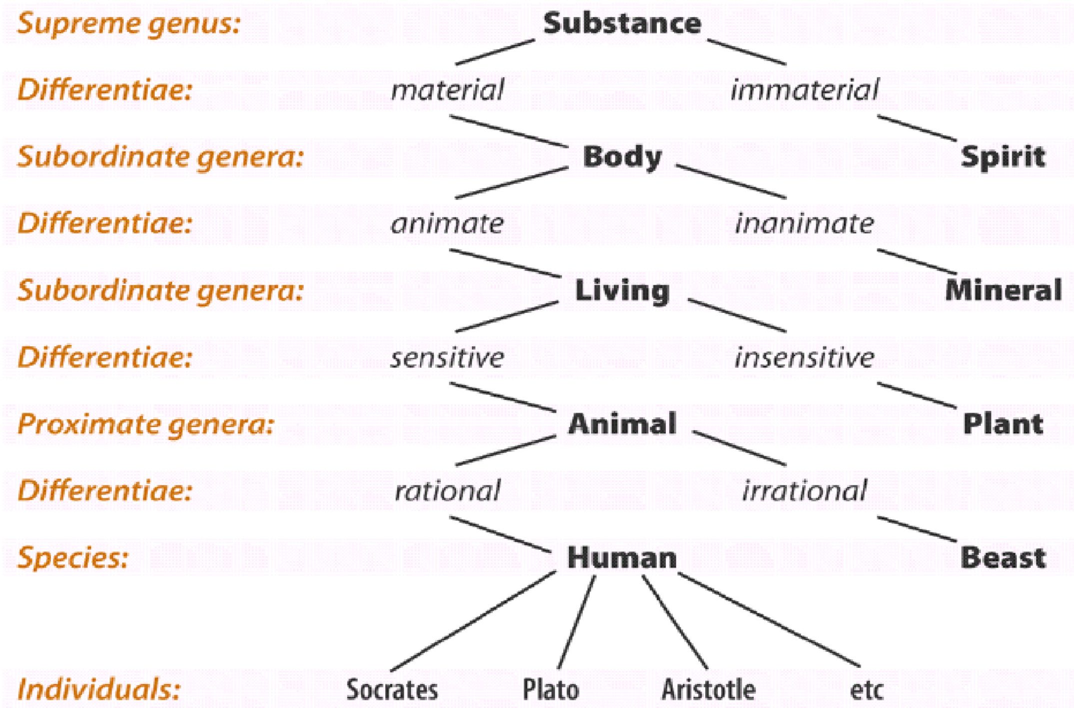
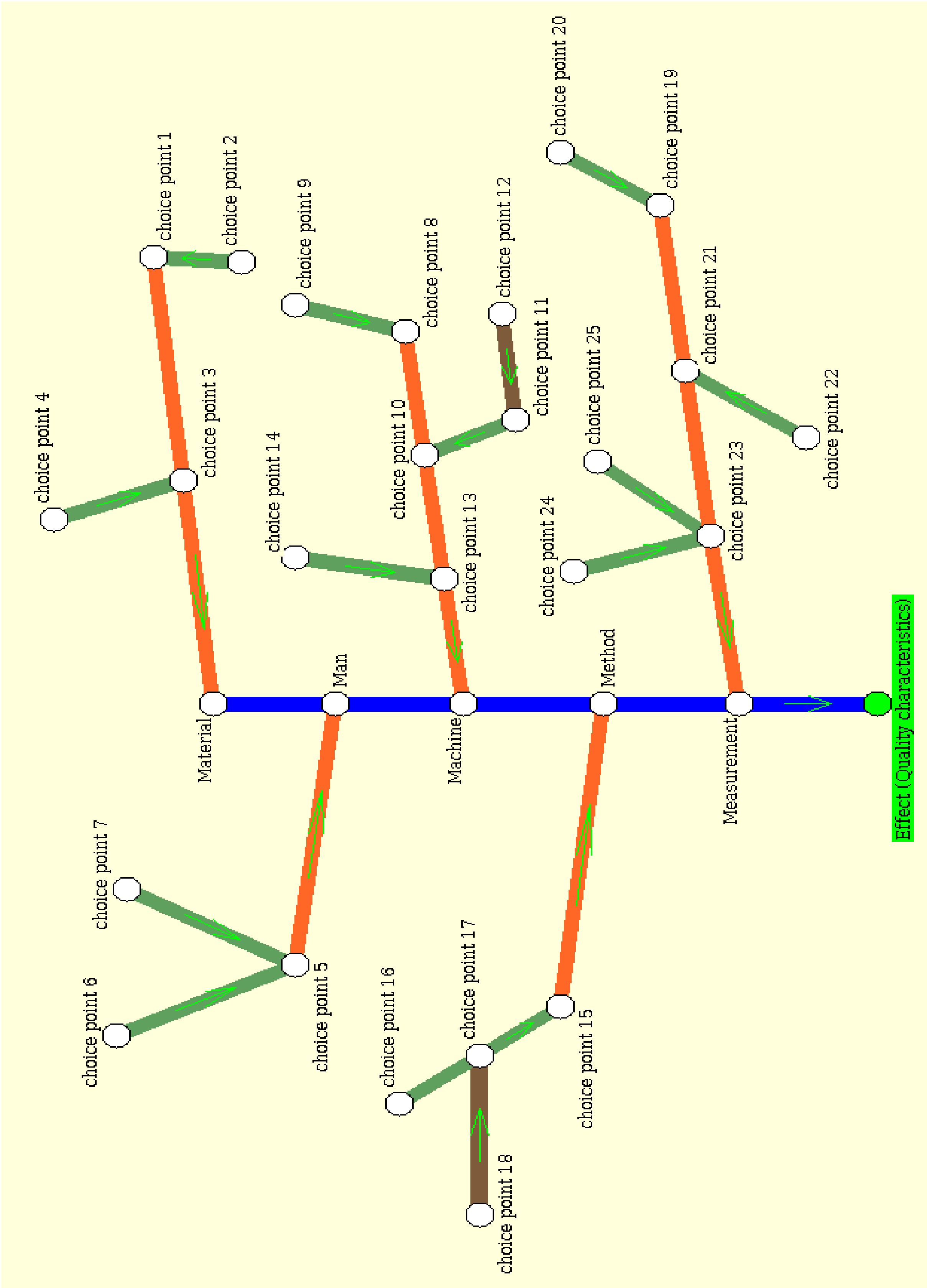


Figure 3: Horrock's Porphyry Tree (2005) - http://www.epsg.org.uk/pub/needham2005/

Unlike the Porphyry tree, the fishbone suited a strategic focus on a final objective, with a tributary hierarchy demarcated by goals and targets to achieve this objective. It was a practical logic based on a cause and effect ontology of manufacturing. The verification of an ontology may be presupposed where experts are relied upon to instruct on it; in the expert realm, experts have the best understanding of what the law of negligence defined as reasonable foreseeability; they are best equipped to determine the risks of negligence and how to manage them. The epistemology of experts in the use of their ontology may better determine the nature of the logical validity of their verification. The practical logic of the fishbone may be seen as a hierarchy of hypothetical propositions for use in a hierarchy of *modus ponens* syllogisms for deduction of the Final result. This will be illustrated further through the Home insulation application. It is also to be noted that Peirce (1883, 1933, Vol 111, p.242) conceived or devised a fishbone graphic as a logic structure.

Although the fishbone runs left to right, it can be turned in the same direction of top down as the Porphyry tree. Figure 4 shows the Ishikawa fishbone turned top down, as an eGanges River, so that it can be read more like the page of a book written in English, from left to right and from top down. In the top down form, the fishbone structure is more readily understood as a tributary River system. The arrows which indicate the direction of flow of the tributaries are in fact the inference arrows of hypothetical premises. Thus, if material, man, machine, method and measurement are controlled, then there is the consequent or effect of quality controlled manufacturing: if a, b, c, d and e → f. Each stream in the tributary hierarchy



Paper 2.4, Figure 4: Ishikawa's Fishbone Turned Top Down
[Gray and Gray, 2011, p275]

is a conditional proposition or a rule. The hierarchy arises because an antecedent in one hypothetical premise or rule, is also the consequent of another hypothetical premise or rule. If a practical objective is determined in a Porphyry tree ontology as the Final Result, a Porphyry tree can be also reconstructed as a fishbone, as shown by Gray (2007) in her Figures 4.18 and 4.20, a consideration of which is outside the scope of this paper. The fishbone and the eGanges River are what Harré and Krausz (1996, Ch.3) recognised as conceptual systems that are used as epistemologies; they act as legal and quality control epistemologies.

Use of the eGanges River

The River representation of the fishbone was used to develop eGanges, software which is a total fishbone quality control tool for use in various stages of quality control, including brainstorming requirements; it uses a River graphic in its interface, which is shown in Figure 7. eGanges is especially suited to government as it was originally designed as a legal expert system shell (Gray and Gray, 2003). An eGanges application can capture the detail required for attainment of an objective, and communicate massive complexity in a user-friendly way; no more than the rural understanding of the tributaries of a river system or the urban understanding of street maps is required. If an eGanges River becomes too large or complex, parts of it may be nested as submaps; the hierarchy of submaps is available in addition to the hierarchy of the River tributaries.

As a logic structure, Ishikawa's fishbone did not locate the possibilities of failure of, or uncertainty about, any of the detailed requirements for quality control. eGanges does, and that is why it is described as a total fishbone quality control system. If a necessary antecedent fails, no matter how deeply nested in the River system, eGanges will report the failure of quality control. The inherent possibilities entailed when each antecedent in a rule may be either satisfied, fail or be uncertain, produces considerable combinatorics, which eGanges can process automatically through its programmed heuristics.

Sphere

The eGanges heuristics were determined from the graphical representation as a spherical structure, of the total possibilities inherent in the fishbone. In order to represent the requirements for quality control, as well as the possibilities of failure and uncertainty, *prima facie*, it is necessary to show three corresponding Rivers for Positive, Negative and Uncertain possibilities. If the eGanges River, like the fishbone, is regarded as the Positive River, and all its requirements for quality control are met, this produces a Positive Final result. However, all the requirements may be Negative, so that there is a wholly Negative Final result. Alternatively, all requirements may be Uncertain and this produces a wholly Uncertain Final result. As illustrated in Figure 5 (Gray, 1990), notional three dimensional logic space is required to represent not just the Positive River, but also the corresponding Negative and Uncertain Rivers. The triad streams represent the spectra of inductive instances that particularise each requirement, with three sectors ranging from the Negative, to the Positive and Uncertain. An informed government subject has justice in decisionmaking.

Where only some requirements are Positive, but also some are Negative or Uncertain, there is a combinatorial explosion of possible alternative situations; results may be Partially Negative or Partially Uncertain. Figure 6 shows, as well as the corresponding Positive, Negative and Uncertain Rivers, the complete logic sphere, with its two poles that represent, respectively, the Partially Negative and Partially Uncertain results.

The representation of the Negative and Uncertain Rivers in Figure 5 is not adequate for the pole heuristics, as it does not account for de Morgan's law of logic which specifies that the contradictory of a conjunction is a disjunction. This is outside the scope of this paper, as is a full exposition of the finer pole heuristics of eGanges; but, more correctly, for pole heuristics, the correspondence in Figures 5 and 6 should show the Negative and Uncertain Rivers in the nature of fans of disjunctive Rivers. An accumulation, in any order, of all failed alternatives in a disjunction may be required to trigger a pole Result, whereas, for a Wholly Negative Result, all requirements must be Negative. *Prima facie*, one contradiction of a necessary condition produces a Partially Negative but conclusive pole Result; similarly, one uncertainty about a necessary condition produces a Partially Uncertain pole Result. Where there is both a Negative and an Uncertain, then finer pole heuristics are required to prioritise each. Generally, for Positive conjunctions with one Negative failure, a Negative Final pole Result prevails; for Positive disjunctions, where all Positive alternatives are exhausted, in any order, an Uncertain prevails over a Negative. The operation of these computational heuristics can be viewed by trialling the online eGanges applet at: www.grayske.com/FinLawTrial/.

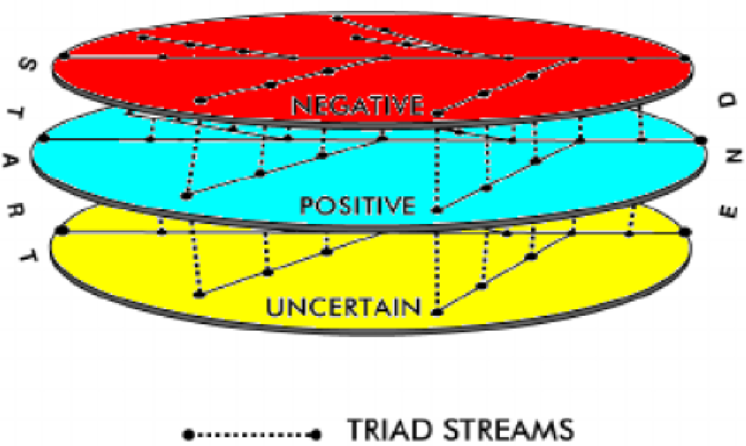


Figure 5: Triad of Rivers with Spectral Links

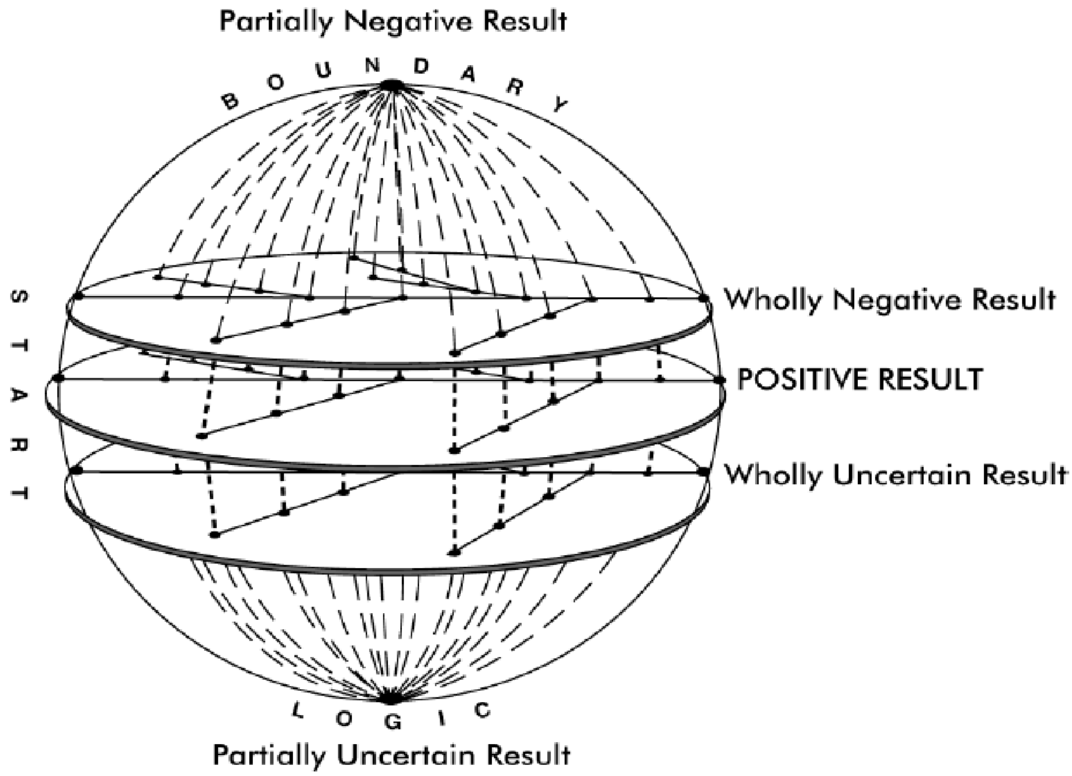


Figure 6: Sphere of Total Fishbone Quality Control Logic

The sphere was used to determine the heuristics of eGanges processing of the application of the River knowledge to a user's case. The sphere is a more complete epistemological system than the fishbone; potentials are outside the deductive structure, but may be retrieved as glosses on the antecedents in the River structure of possibilities at the precise point of relevance. Through processing heuristics, the River system of eGanges is expanded to capture quality control of failure and uncertainties; the heuristics implement the total fishbone quality control of a sphere. The eGanges Positive River can be contrasted with alternatives of contradictories and uncertainties (Gray, 1988, 1990, 2007). Graphical representation of the totality of these contrasts requires the deductive, and linking inductive, structure of the sphere.

3d logic space was discovered or first devised by the Ramist School of logic at the Sorbonne University in Paris in the sixteenth century (Ramus, 1543; Ong, 1958). Adversarial methodology, developed in the common law courts, was adapted for science early in the seventeenth century, by Francis Bacon (1620), who had been head of the English Court of Chancery; he was concerned with lining up the parallel attributes of hot and of cold for scientific determinations. The sphere incorporates the adversarial structure in 3d logic space.

As an epistemological system, the sphere is useful for total fishbone quality control, the application of law, and education involving the academic system of numbering points in a hierarchical way like tributaries of a River system, e.g. 1, 1.1.1, 1.1.2, 1.1.3, 2, 2.1, 2.2, 2.3, 2.4.1, 2.4.2, and so on. An eGanges application can communicate large, complex, educational material.

Application Example: Quality Control Home Insulation

In Australia, in 2009, federal legislation provided funding of home insulation as an environmental measure to reduce the use of electricity for heating and cooling. The insulation industry expanded rapidly with inexperienced insulators. As reported widely in the media, some stapled insulation to electric wiring, causing some deaths; some covered downlights in the ceiling with insulation, causing some house fires; many provided the maximum amount of insulation that could be funded when a lesser amount only was required, and so wasted available funds. These possible negative and uncertain matters could have been reasonably forseen by expert insulators, and thereby provided for in the government scheme as a matter of total fishbone quality control. Commonly experienced quality control practices of business have raised a public expectation of quality control government, both in civilisation design and in administrative operation, no matter how difficult this might be; online technological aids to provide this are available and indispensable as collective intelligence.

Figures 7-8 show a draft eGanges River map for a home insulation application, in the eGanges interface. Figure 8 is a submap of the node, Ceiling inspection, in Figure 7. A node with the appearance of a soccer ball, such as Ceiling inspection, indicates that there is a submap. By submapping in nested graphics, an expansion of an application may cover and manage all the requirements; this also facilitates maintenance of the application when changes are made.

In this real example, it can be seen that a condition on the mainstream, such as Preparations in Figure 7, may be particularised by conditions on the secondary stream which joins the mainstream at the point of Preparations. The hierarchy of finer definition may continue upstream through tertiary, quaternary, quinary etc streams. As conditions are satisfied upstream, there is a logic flow downstream to establish the consequents in lower streams, until eventually the Final objective may be established. If each tributary stream in the River is seen to represent a rule (a hypothetical proposition or a conditional proposition), then the logic of the River expertise can be understood. For example, in Figure 7, the mainstream rule is:

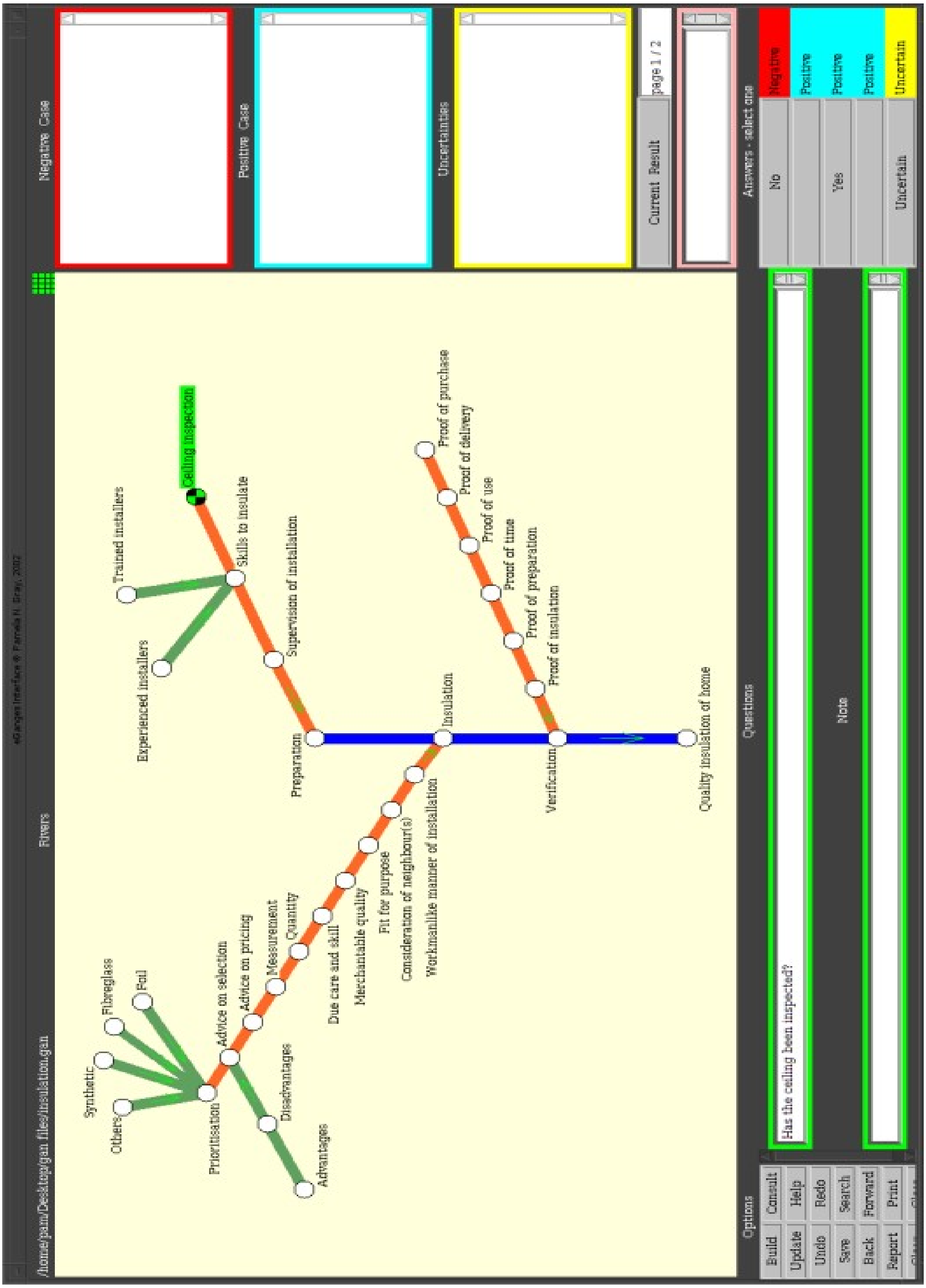
> If there are preparations, insulation and verification, then there is quality insulation of home.
> The secondary stream that establishes preparations is a rule:
> If there is ceiling inspection, skills and supervision, then there are preparations.
> The tertiary stream arising from ceiling inspection, shown in Figure 8, is a rule that:
> If there is inspection of electrical wiring, waterproofing, ownership of ceiling cavity, cleanliness and pests, then there is ceiling inspection

And so on.
Any node can be glossed with retrievable data relevant to that node. Glosses might indicate the spectrum of instances of a condition, or simple text that explains, justifies or warns of some related matter. The availability of a gloss in consultation mode, is indicated by the criss-crossed green square above the top right hand corner of the Rivers window in the eGanges interface in Figure 7.

The interface also shows that a question is asked in the Questions window to establish the existence of the requirement of Ceiling inspection in the user's case, to see if there has been quality control: has the ceiling been inspected? Three alternative answers are also presented to the user for selection: yes, no, uncertain. Beside each answer, its consequent is given. Red identifies a Negative Result, Blue identifies the Positive Result, and Yellow identifies an Uncertain result. If the natural language of questions so requires, the answers may be placed adjacent to the Result required; for example, the answer yes may produce a Negative Result if so positioned. Also, all three answers may be positioned as Positive to manage consistencies of all three alternatives with the Positive Result.

The answers in the interrogation system provide the categorical premise to complete the *modus ponens* deduction which, as necessary reasoning, is then automated by the eGanges processing of the answer input. Feedback verifies the processing; the node label appears as feedback in the appropriate adversarial window which has a colour corresponding to the colour of the consequence attribute of the answer. The Red adversarial window at the right of the interface, identifies the red answer points for the Negative Case, the Blue adversarial window identifies the blue answer points for the Positive Case, and the Yellow adversarial window identifies the yellow answer points for the Uncertain Case; the Current Result button confirms in its pink window the net Current Result, at any point in the consultation. The user is advised of outcomes that should be expected for each possible selection of input, and the interim and cumulative outcome is given as feedback.

The interrogation system implements a question and answer logic hierarchy which was suggested in the epistemology of Collingwood (1940); the hierarchy arises from interlinked sequences of questions and answers, such that each question/answer pair presupposes the answer to another deeper question, up to the watershed of the finest points or down to the outcome mouth of the River that supports the whole hierarchy. Lawyers proceed in this way in conducting a case. In spherical logic there are five River mouths, the possible alternative outcomes of the sphere: Positive, Wholly Negative, Partially Negative, Wholly Uncertain, and Partially Uncertain, for statutory interpretation.



Paper 2.4, Figure 7: Initial map of Insulation Application in eGanges Interface
[Gray and Gray, 2011, p281]

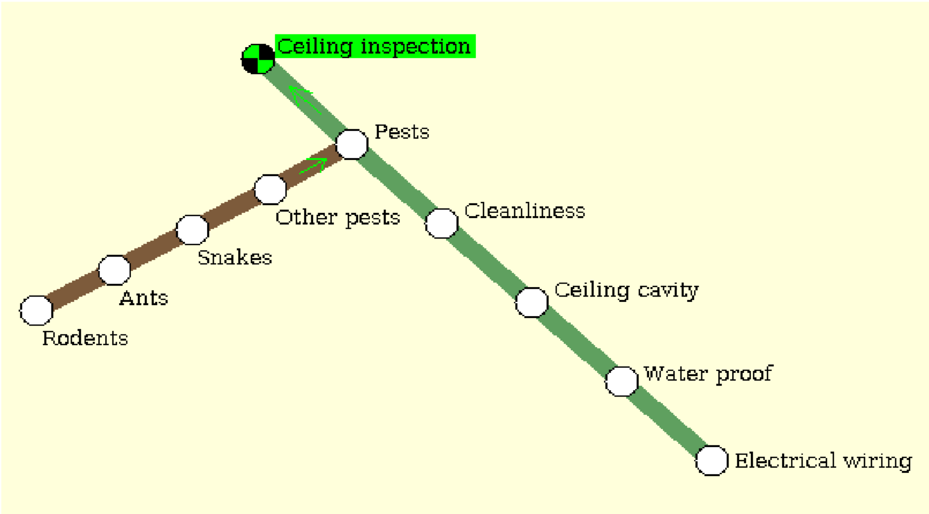


Figure 8: Submap of Ceiling Inspection

Each antecedent in home insulation rules, has a question that is put to establish the condition, and the user is given a choice of three possible answers: yes, no or uncertain. Each answer is associated with either a Positive, Negative or Uncertain Final result i.e. that there is the Positive result of Quality insulation of a home, a Negative result of Failure of quality control, or an Uncertain result of uncertain quality control. The questions provide an opportunity for the user to give answer input that is the categorical premise of a *modus ponens* argument. Thus, if the input answers establish all the conditions for the requirement of Ceiling inspection, then the rule consequent of Ceiling inspection is established. The Final consequent is Quality insulation of home.

Conclusion

The Ishikawa fishbone captures only the limited logic of achieving a single objective in predetermined ways that permit the setting of goals that collectively achieve the objective, and targets that collectively achieve each goal. It does not consider failure as an objective which has failure goals and targets, or uncertain as an objective with uncertain goals and targets. As the world becomes increasingly complex and uncertain, so, too, government is drawn into risk management and failure concerns. With total spherical quality control, complex micro systems of legal choice and legal consequences of selection can be managed, designed, communicated, and maintained, with ready public access. Public coordination is a requirement for quality control government.

References

Bacon, F. (1620, 1952): *Novum Organum*, Great Books of the Western World, Vol 30, Encyclopaedia Britannica Inc, Chicago, USA.

Collingwood, R.G., (1940): Essay on Metaphysics, Clarendon press, Oxford, England.

Gray, P.N. (1988): *The CLIMS (Contract Law Information Management System) Pilot - Automatable Law*, in published proceedings of the 4th International Congress on Computers and Law, Rome, Italy.

Gray, P.N. (1990): *Choice and Jurisprudential Systems*, Vols 1 and 2, LL.M. thesis, University of Sydney.

Gray, P.N. (1997): *Artificial Legal Intelligence*, Dartmouth, Aldershot, England.

Gray, P.N. (2007): *Legal Knowledge Engineering Methodology for Large Scale Expert Systems*, Vols 1 and 2, PhD thesis, University of Western Sydney: http://adt.caul.edu.au/homesearch/find/?recordid=183556&format=main

Gray, P. and Gray, X. (2003): A map-based expert-friendly shell, in D.Bourcier (ed.), *Legal Knowledge and Information Systems*, IOS Press, Amsterdam, Holland.

Harré, R. and Krausz, M. (1996): *Varieties of Relativism*, Blackwell, Oxford, England.

Horrock, I. (2005): *Ontologies and the semantic web*, http://www.epsg.org.uk/pub/needham2005 /Horrocks_needham2005.pdf

Ishikawa, K. (1985): *What is total quality control? The Japanese way*, translated by David J. Lu., Prentice-Hall Inc., Englewood Cliffs, N.J., USA.

Ong, W.J. (1958): *Ramus method and the decay of dialogue*, Harvard University Press, Cambridge, MA, USA.

Peirce, C.S. (1883, 1933): *Exact Logic, Collected Papers*, Vol.3, C. Hartshorne and P. Weiss (eds), Harvard University Press, Cambridge, MA, USA.

Ramus, P., (1543, trans. 1574, 1969): *Dialecticke, C.A.*, Dunn (ed.), San Fernando Valley State College, Northbridge, CA, USA.

About the Authors

Dr. Pamela N. Gray

My doctoral research at the University of Western Sydney developed the design of an expert system shell, eGanges, for the fields of law, management and education. I have constructed many diverse applications using the shell and continue to do so, as my main interest. I am the most senior legal practitioner in the world with a PhD in legal knowledge engineering. I had extensive experience in legal practice in Australia and in London before becoming a law lecturer in business schools in various tertiary education institutions in Australia. While I was a Visiting Fellow in Law at Cambridge University and collaborating in the first ESPRIT-funded legal project which used Prolog, I discovered spherical logic, as a move from hard-coded logic programming and rule-base systems to an easier object-oriented programming. I reported it in 1990 in my LL.M thesis at University of Sydney. The thesis was revised and published in England as a book, Artificial Legal Intelligence (1997) in Dartmouth's Applied Legal Philosophy Series. I have published my research work from time to time in numerous conferences in the field. My son and co-author who programmed eGanges in Java, is currently writing up his thesis on its multi-value logic.

Chapter 3: Multi-valued Logic Superexpertise

This chapter deals with some significant foundational issues of the eGanges and NeGame epistemologies, which are used as the basis of the superexpertise aids discussed in this thesis. These issues stem from the fact that these epistemologies use multi-valued logics; they pertain to the verification and validation of the superexpertise of these aids.

Classically there are only two possible logic values: *true* and *false*. However, in the eGanges epistemology there are four possible epistemic truth values: positive/true, negative/false, uncertain/possible and unanswered/unknown. The distinctions between these four types of epistemic logic values are important within the eGanges epistemology as the conclusions reached vary according to which of the four values holds. The NeG epistemology of chapter four requires six epistemic values as it has six distinct possible conclusions.

Belnap [1977] recognised the important difference between these types of epistemic multi-valued logics, and more standard 'ontological' logic values. However, Dubois [2008] believes that multi-valued logic supporters have become confused between the differences between degrees of belief and degrees of truth. He has correctly identified that by using a multi-valued logic, you must sacrifice either compositionality or classical tautologies. Thus, Dubois believes that binary logic knowledge representations, that use some other symbols in their syntax to represent the epistemic knowledge, are superior to using multi-valued logic, since binary logic is necessary to preserve both compositionailty and classical tautologies.

In reality, alternative binary knowledge representations, such as AnsProlog (Answer Set Programming) do not actually avoid the problems Dubois raises; they merely defer the problems to alternative connectives in order to preserve the classical tautologies for the standard connectives. There are many other alternatives to using multi-valued logics, such as signed logic, AnsProlog, etc. The standard computing alternatives emphasised in this chapter are AnsProlog and generalised horn clauses.

3.1 Representing dichotomies: epistemic and ontological truths

As explained in Chapter One, with perfect knowledge, selecting a single mutually exclusive solution from a finite set of all possible solutions can be achieved by answering a finite number of dichotomies (yes/no answerable questions); the simplest procedure is to iterate through each solution in the solution set and ask if that is the solution to use. As one, and only one, solution must be correct, the procedure will terminate with the correct selection in, on average, a time proportional to the solution set size; the search time is guaranteed to be finite as the solution set size is finite.

Problems occur when knowledge is not perfect, or there is an infinite solution set; the dichotomies may not be able to be answered due to lack of knowledge or lack of time. This raises the possibility that trichotomies may be the simplest answerable question type, the three possible answers being yes, no or uncertain. It is acknowledged that from an ontological perspective, with perfect knowledge, a mutually exclusive solution set requires only yes or no dichotomy answers. Not all problems have mutually exclusive solutions, e.g. quantum systems can have answers that are simultaneously yes and no (the wave-particle nature of physics). A mutually exclusive example would be the fact that someone can, in reality, only be guilty or innocent of a crime, not both. However, as we are not omniscient beings, we must take account of our limited knowledge; hence someone is found to be either guilty or not-guilty of a crime, as a court of law acknowledges these knowledge limits.

3.1.1 Lack of Perfect Knowledge

A mutually exclusive solution system can only be in one of its possible states at any one time. Lack of information limits our reasoning, and our ability to know in which exact state it actually is. For example, if C is not known to be true and can not be proven, and $\neg C$ is not known to be true and can not be proven, then there is no knowledge as to what the value of C is. It might only be physically possible for C to be true or false, but without knowing which it is, if you can not avoid making a decision, you must rely on some form of nonmonotonic reasoning (e.g. intuitionistic, default, or multi-valued logic) to determine the most appropriate default belief about its state..

A monotonic logic is a logic where any new knowledge/rules can never cause previous conclusions and knowledge to be retracted. Any logic where conclusions can be retracted is thus non-monotonic. For example, a retraction is caused by a default assumption being contradicted due to the new knowledge, and as it was only a default assumption it can be retracted to prevent the contradiction. Work has been done to show the translatability between the different types of non-monotonic logics: different default logic semantics [Delgrande and Schaub, 2003], autoepistemic logic [Gottlob, 1995], priority logics [Janhunen, 1998], etc. Multi-valued logics like eGanges are default logics that use a multi-valued semantics; similar examples include a default logic based on the three-valued well-founded semantic.

Semantics is the study of meaning. It focuses on the relation between signifiers, such as symbols, and what they stand for. In AnsProlog there are two main types of semantics: the binary stable model semantic and the three-valued well-founded semantic. These are further discussed below in section 3.5.1AnsProlog syntax and semantics. The eGanges semantic is further discussed in section 3.4eGanges syntax and semantics.

Knowledge and belief are related. Knowledge is belief with high levels of certainty. Perfect knowledge has 100% objective certainty. The inherent uncertainty in non-perfect knowledge means it may be important to take note of the level of certainty about the knowledge. For a binary situation clouded by uncertainty, say the value of the Boolean proposition C, there are actually four possible situations for an agent:

1) C is believed to be true, when it actually is true;
2) C is believed to be true, when it actually is false;
3) C is believed to be false, when it actually is true; or
4) C is believed to be false, when it actually is false.

The problem in real world reasoning is that we don't always have access to perfect knowledge. Even something currently known to be true, could theoretically be shown to be false at some stage in the future. As such, it may be epistemically important to note the confidence level of a conclusion; while a variable may only be binary, its real world consequences may have more than two distinct possiblilities.

3.1.2 Degrees of truth v degrees of belief

Many situations are not binary. For example, a cup of water can be half full; it does not have to be completely full or completely empty, it has degrees of fullness. It is possible to be 100% certain that a cup is half full (within an atom's displacement level of accuracy). Such things have multi-truth options.

In contrast to a multi-truth situation, we might only be 50% sure that a switch has been switched on; the switch itself can only be on or off, but we may not have knowledge of its current state. This leads to degrees of belief as to the state of the switch. Many who oppose the use of multi-value logics, do so because of the general confusion in the literature between the differences between degrees of truth and degrees of belief.

Those who object to multi-value logic do so because they believe degrees of belief about knowledge is a meta-notion; “uncertainty handling is a matter of consequencehood and validity, hence a meta-notion with respect to truth-values” [Dubois, 2008, p.196]. They believe uncertainty should be handled separately to the core logic processing, e.g. as a modal logic. Dubois accepts the validity of multi-truth approaches such as fuzzy logic, but they argue this is quite different to degrees of belief.

Those who support using multi-valued logics to encode belief do so for pragmatic implementation reasons; Belnap says that the systematic use of truth-tables “tells us how the computer should answer questions about complex formulas, based on a set-up representing its epistemic state” [Belnap, 1977. p.41]. They argue that a computer needs some mechanism for processing uncertainties, and using a truth table encodable rule is a simple, elegant way to handle that processing. This pragmatic truth table approach is also adoptedin this thesis.

Regardless of the method used to represent belief and truth, with incomplete or inconsistent knowledge it is impossible to be 100% certain that a proposition is true or false, and this must be representable in the chosen knowledge representation; a real world reasoning system must deal with uncertainty and inconsistencies. For example, AnsProlog can use a standard two valued semantic such as the stable model semantics, in which case the value of unknown for a proposition is represented by neither the proposition nor its explicit negation being in the answer set. Alternatively, AnsProlog can use the three valued well-founded semantics which “is more tractable than computing the entailment with respect to stable models” [Baral, 2003, p5].

Some logics, such as intuitionistic logic and defeasible logic have three options for a proposition to handle this uncertainty; defeasible logic has Justified, Defensible or Overruled conclusions. Even in classical Boolean logic, the truth value of a non-atomic formula (e.g. $A \vee B$) may be either a tautology (always true), a contradiction (always false), or sometimes true/false; i.e. there are at least three possible classifications for non-atomic formulae, even when atomic formulae (e.g. A) can only ever be true or false.

Dubois [2008, p6] acknowledges that belief sets are inherently ternary as a proposition may or may not be in an agents belief set, just as its negation may or may not be. Indeed, this is the method used by AnsProlog to capture a state of unknown (the state where neither a proposition nor its explicit negation are in the answer set).

3.1.3 NeGame: multi-truth epistemology

The NeGame epistemology, discussed in Chapter Four, does not suffer any issues about degrees of belief. The negotiating parties determine which of the six possible states an issue is currently at; these six possibilities represent six distinct possible truths, as there is no uncertainty as to which option the parties are in.

3.1.4 Degrees of belief can create degrees of truth

As Dubois says, “The difficulty lies in the confusion between truth-values and information states” [Dubois, 2008, p1]. However, there is good reason for this confusion, as uncertainty may result in more possible states; real world reasoning can also cause effects that alter the number of possible truths.

Omniscient courts (courts free from uncertainties), would never make a mistake in their judgement. As such there can only ever be two consequences: guilty people are punished and innocent people are never punished. In the real world, courts must often make decisions with uncertainties and lack of knowledge. As such, four possible outcomes may occur:

1) the guilty are punished – a true positive conviction result;
2) the innocent are punished – a false positive, i.e. type 1 error [Neyman, 1928];
3) the guilty go unpunished – a false negative, i.e. type 2 error [Neyman, 1928]; or
4) the innocent go unpunished – a true negative conviction result.

As such, the inability to remove degrees of belief may cause an increase in real world degrees of truth. Thus eGanges epistemology provides for a four-valued logic, where each possible logic result represents a different real world consequence. For example, if the Final result is Positive, the positive case has conclusively been established; in AnsProlog this is equivalent to only the positive literal version of the proposition being shown to be in the answer set. If the final result is Negative, the negative case has conclusively been established; in AnsProlog this is equivalent to only the negative literal version (i.e. the explicit negation) of a proposition being shown to be in the answer set.

If the Final result is Unanswered, then neither the positive nor negative case has been established due to a lack of information; obtaining more information could resolve the Final result. If the Final result is Uncertain, then neither the positive nor negative case has been establisheddue to key atomic fact(s) remaining unresolved, e.g. both positive and negative literal versions of a key proposition appear in the answer set; obtaining more information may not resolve the Final result. The course of action to take may vary depending on which of these four results is obtained, ideally:

1) a Positive result means the positive side wins;
2) a Negative result means the negative side wins;
3) an Uncertain result means one side wins by default - this default win should be recorded in the event that new evidence comes to light that conclusively shows the other side should have won, potentially justifying a retrial.; and

4) an Unanswered result means the adversarial process continues.

In short, there are some problems with using a binary logic when it is inappropriate.

3.1.5 Problems with binary logic representation

Two value (true/1 and false/0) logic is very powerful when it is applicable; however, it is not always applicable. Let us consider three two value variables, and some logical relations as shown in Table 3.1 below.

a	b	c	a xor b	a → c	b → c	(a → c) v (b → c)	(a & b) → c	(a → c) xor (b → c)	Not ((a xor b) → c)
0	0	0	0	1	1	1	1	0	0
0	0	1	0	1	1	1	1	0	0
0	1	0	1	1	0	1	1	1	1
0	1	1	1	1	1	1	1	0	0
1	0	0	1	0	1	1	1	1	1
1	0	1	1	1	1	1	1	0	0
1	1	0	0	0	0	0	0	0	0
1	1	1	0	1	1	1	1	0	0

Table 3.1: truth table of several Boolean formulae

We must be careful in what we try to make the variables a, b and c represent. For example, if we represent having a single apple by a, and having a single banana by b, and having a single piece of fruit by c, then by two value logic: having a single apple implies having a single piece of fruit or having a single banana implies having a single piece of fruit, would be represented by ((a → c) v (b → c)). However, by two value logic, this is identical to requiring having both an apple and a banana in order to imply having a single piece of fruit ((a & b) → c). If we alter the or to ensure that only a single apple or single banana can imply a single fruit at any one time ((a → c) xor (b → c)), this implies that it is not the case that a single apple or single banana (at any one time) can imply having a single piece of fruit (Not((a xor b) → c)). As this example shows, it is important to ensure the logic used is applicable to the situation that the logic is trying to capture, i.e. it is necessary to ensure two value logic truly fulfils the epistemic adequacy for the situation.

3.2 Problems with Multi-Valued Logics

It has been discussed how binary logic can have epistemic problems that make it inappropriate to use. However, there are problems inherent with multi-valued logics that need to be addressed if they are to be appropriately used in an epistmology.

3.2.1 Compositionality, consistency and decidability

In semantics and mathematics in general, the Principle of Compositionality is the principle that the meaning of a complex expression is determined by the meanings of its constituent expressions and the rules used to combine them. This principle is also called Frege's Principle, because Gottlob Frege (1848-1925) is widely credited with the first modern formulation of it [Dummett, 1973].

For logic, compositionality is the property of well formed non-atomic formulae to have their truth value consistently determined by the truth value of the atomic components of the formula. For example, the formula ($P \wedge Q$) follows a truth table, such that the truth value of the formula can be determined by knowing the truth values of P and Q.

In logic, consistency means that there are no contradictions, i.e. if a proposition can only have one truth value at a time (because truth values are mutually exclusive), then it must only have one truth value at any time. In classical Boolean logic, a contradiction is captured by the formula ($P \wedge \neg P$); if this formula is found to be true, then the truth value of P must, by the compositionality of the formula, simultaneously be true and false. Therefore, in classical logic ($P \wedge \neg P$) being true fits the definition of a contradiction. Clearly, consistency is heavily related to compositionality; as Dubois [2008] points out, if you give up compositionality, you do not force ($P \wedge \neg P$) being true to necessarily mean P has two inconsistent values.

Decidability means the truth value of a formula can always be determined. As previously stated, even in classical Boolean logic, it is possible that neither P nor $\neg P$ can be shown to be true; this ternary nature of logic is unavoidable, and without providing a truth value to represent this possible proposition state of unknown, the proposition's truth value can not be determined. If a truth value of unknown does exist, technically a logic can always be decidable.

Default logics will often assume a proposition is false (c.f. Closed World Assumption – see section 3.4.4), in order to proceed with real-world processing, given a lack of sufficient information to resolve the actual truth value of a proposition. Sophisticated default logics, such as AnsProlog, allow program makers to explicitly state when a rule can be used by assuming a false proposition, and when the proposition must be proved false to fire.

3.2.2 Classical tautologies

All formulae that represent contradictions, should always be false to ensure a logic is consistent. Therefore the negation of such contradictory formulae should always be true in a consistent logic. These always true formulae are called tautologies. For example, by application of De Morgan's laws, the contradiction $(P \wedge \neg P)$ becomes the tautology $(P \vee \neg P)$. Clearly, as Dubois [2008] shows, if a multi-valued logic gives up compositionality to preserve classical contradictions, it will preserve classical tautologies. Given the critical importance of compositionality to making a functional logic theory, most such logics choose to instead give up classical tautologies.

Looking at the simplest three valued logic (true, false, unknown):

1) If ¬Unknown = Unknown, then (Unknown ∨ ¬Unknown) = Unknown ≠ true. This forces such a logic to give up the classical tautology $(P \vee \neg P)$, as it is not always true any more.
2) If ¬Unknown = false, then (Unknown ∨ ¬Unknown) = (Unknown ∨ false) = Unknown ≠ true. Again, the classical tautology does not hold.
3) If ¬Unknown = (true ∨ false) = true, then (Unknown ∧ ¬Unknown) = (Unknown ∧ true) = Unknown ≠ false. This means the classical contradictions do not hold, though the classical tautology does.

Moving on to four valued logic does not resolve these problems. For example, {true, unknown, uncertain, false} with (¬unknown = uncertain) and (¬uncertain = unknown) means:

1) (unknown ∨ uncertain) ≠ true; and
2) (unknown ∧ uncertain) ≠ false;

All these formula have assumed conjunction (∧) and disjunction (∨) have the property of compositionality; sacrificing compositionality will enable both the classical contradiction and tautologies to hold. This raises the question: is it worth sacrificing classical tautologies? If so, does a lack of classical tautologies induce paradoxes for reasoning with multi-valued logics?

3.2.3 The true meaning of classical tautologies

As there are only two possibilities for a truly binary proposition P, $(P \vee \neg P)$ must always be true as it covers both possibilities. However, as discussed in section 3.1.4Degrees of belief can create degrees of truth, for a binary proposition there are four possible states, caused by an agent's imperfect knowledge. Each state may have different real world consequences: true positive (tp), false positive (fp), false negative (fn) and true negative (tn). To cover all four possibilities requires something like: $(tp(P) \vee fp(P) \vee fn(P) \vee tn(P))$, where $tp(P)$ is true if

proposition P is known to be a true positive and false otherwise; similarly for fp, fn and tn.
It is important to note that this viewpoint still requires perfect knowledge as the agent does not necessarily know which state they are in (if they are in a false positive or negative and knew it, they would change to the appropriate state). This means the existence of these four states is not directly a result of belief; the fact that there are four states is indirectly caused by the fact that the agent does not have perfect knowledge, and therefore must believe whether a proposition is true or false. However, in reality, these four possibilities are all real-world possibilities, i.e. ontological truths that may have real-world differences that need to be taken into account. Since each state may cause real-world differences, they should be considered multi-truths, not merely multi-beliefs.
The four ontological truth viewpoint above still does not help with practical reasoning as it still requires perfect knowledge to differentiate between the truths. Such perfect knowledge may become available (for example, new evidence could come to light that conclusively proves a convicted man was innocent), or it may never become available. To deal with this imperfect knowledge requires “epistemic multi-truths”, i.e. truths that take an agent's belief into account.
The explicit negation symbol (¬) effectively acts as a falsity testing operator; ¬P = true means P is false. Unknown(P) = true could be another operator that indicates the value of P is not known to be true or false. Given the fact that there is a minimum of 3 epistemic possibilities for the value of a Boolean proposition (true, false, unknown), then such an operator approach might be appropriate. For a multi-valued logic, this operator approach can be generalised as follows:

1) Let the set of possible epistemic truths be $\{ t_1, t_2, \dots , t_n\}$, for some integer n.
2) Any proposition must have a truth value equal to one, and only one, of these possible truths.
3) Therefore, any proposition P must have $(P=t_1)\vee(P=t_2)\vee\ldots\vee(P=t_n)$ as a tautology as exactly one of the (P=t) components of the formula must be true; the others are false.

This approach is used in the Four-Valued Propositional Logic syntax discussed below.

3.2.4 Collapse of four ontological states to one

If the real-world consequence for P and ¬P are identical, then the four ontological truths caused by the imperfect knowledge of the agent collapse down to a single ontological state. There is no difference caused by the truth-value of the proposition, so there is no difference caused by the agent's belief of the truth-value. Without the splitting to more than two possible states, ($P \vee \neg P$) must again be true.

3.3 Four-valued Predicate Logic (4PL)

This section discusses a four-valued predicate logic (4PL) which has a one to one relation with eGanges logic. 4PL is built of an axiom alphabet and formation rules in a similar manner to standard two-valued predicate logic (2PL).

3.3.1 Syntax

Syntax deals with the rules used for constructing, or transforming the symbols of a language. The axiom alphabet of 4PL consists of seven classes of symbols:

1) object constants;
2) variables;
3) function symbols;
4) predicate symbols;
5) connectives;
6) punctuation symbols; and
7) four truth-constants.

The four truth-constants are { f, u, i, t } which respectively represent false, uncertain, incomplete/unanswered, and true. 2PL only has the values { f, t }, and these symbols are almost never used directly in the syntax.

There are two punctuation symbols, namely the open bracket "(" and the close bracket ")".

In 4PL there are three primitive connective symbols: { $\wedge$, $\vee$, = }.

An object constant is a symbol representing one specific real world 'object', e.g. Mr John Smith (of 14 Bond Street Sydney). A variable is a symbol that represents several possible real world objects, e.g. all John Smiths. A function, e.g. address, returns a particular type of real world object(s) when given appropriate real world object(s) as input, e.g. address(Mr John Smith) gives 14 Bond Street Sydney.

Definition 1: Terms are inductively defined as either 1) object constants, 2) variables, or 3) functions of terms. AnsProlog and 4PL define a grounded term as a term without any variables, i.e. only consisting of object constant(s) and functions of the object constant(s).

Definition 2: An Atom is an appropriately satisfied predicate symbol, and if all terms satisfying the predicate are grounded, then the atom is said to be grounded. In this thesis, grounded atoms are treated as identical to propositions.

For simplicity, this thesis will often discuss 4PL from a propositional framework; any uppercase letter, potentially with a subscript, will represent a proposition, e.g. P, Q, R. Propositions act as a place holder symbol for an element of the truth-constants; applying an interpretation to a proposition makes it equivalent to one truth-constant.

Definition 3: Predicate symbols returns an atom when given appropriate terms as input. For example, if “parent_of” is a predicate symbol that returns true if the second term represents a parent of the first term, then parent_of(Jesus,Mary) would return true. A predicate symbol of n arity requires n terms to become a proposition; parent_of is thus a 2 arity predicate.

Definition 4: Interpretation = <Herbrand Universe (*U*), predicate symbol (*P*), mapping (*I*)>. An interpretation makes the predicates symbols equivalent to one particular truth-constant.

Definition 5: A Literal is either an atom (also called a positive literal) or the explicit negation operator applied to an atom (called a negative literal).

Definition 6: The Herbrand Universe is the set of all grounded terms that can be formed.

Definition 7: The Herbrand Base is the set of all ground atoms that can be formed from the available predicates and grounded terms.

Definition 8: Formula/well formed formula (wff)

A formula is inductively created from atoms, punctuation symbols and connectives applied to formulae, using the formation rules:

Definition 9: Formation Rules (FRs)

FR1: an atom by itself is a well formed formula (wff).

FR2: if α is a wff, then (α) is a wff.

FR3: if α and β are wff, then $(\alpha \wedge \beta)$ is a wff.

FR4: if α and β are wff, then $(\alpha \vee \beta)$ is a wff.

FR5: if α and β are wff, then $(\alpha = \beta)$ is a wff.

Definition 10: Primitive Connectives Truth Tables

A∧B	**f**	**u**	**i**	**t**
f	f	f	f	f
u	f	u	u	u
i	f	u	i	i
t	f	u	i	t

Conjunction

A∨B	**f**	**u**	**i**	**t**
f	f	u	i	t
u	u	u	i	t
i	i	i	i	t
t	t	t	t	t

Disjunction

A=B	**f**	**u**	**i**	**t**
f	t	f	f	f
u	f	t	f	f
i	f	f	t	f
t	f	f	f	t

Equals

Table 3.2: Three primitive connectives.

Each non-bold entry in the truth tables is a possible interpretation of the appropriate wff. The bold truth constants in the first column of a truth table represent the possible interpretations for the proposition A, and (where appropriate) the bold truth constants in the first row of a truth table represent the possible interpretations for the proposition B. Therefore, every non-bold cell represents a unique interpretation of A and B.

Complete Lattice Nature

As is seen from the truth tables, four truth values form a complete lattice [Blyth, 2005]:

$$f <_t u <_t i <_t t$$

This means, true has more 'truth' than incomplete, incomplete more 'truth' than uncertain, etc. From this viewpoint, the truth value of $A \wedge B$ equals the lesser truth value of {A,B} and the truth value of $A \vee B$ equals the greater truth value of {A,B}. A=B is true when A and B have the same truth value, and false otherwise.

Definition 11: Equivalence (≡)

Two wff are equivalent if the truth value of every interpretation of their truth tables are identical.

Definition 12: Explicit Negation (¬)

This thesis uses the standard multi-valued definition of negation for a complete lattice, which is to reverse the truth ordering as shown in table 3.3 below.

A	¬A	(A=f) ∨ ((A=u)∧i) ∨ ((A=i)∧u)	¬¬A
f	t	t	f
u	i	i	u
i	u	u	i
t	f	f	t

Table 3.3: Proof $\neg A \equiv ((A=f) \vee [(A=u) \wedge i] \vee [(A=i) \wedge u])$

As shown in the above table, negation can be constructed from the three primitive connectives ∧, ∨ and =. It can also be seen that $\neg\neg A \equiv A$ as these two wff have identical truth values for all four possible interpretations.

Definition 13: Constant wff

If the truth value of a wff is identical for all interpretations, the wff is constant. A tautology is an example of a constant wff, with the constant truth value being true. Likewise, contradictions are constant wff with a truth value of false.

Definition 14: Validity

A wff is valid if it is a tautology.

Definition 15: Unsatisfiable

A wff is Unsatisfiable if it is a contradiction.

Testing for validity:

A finite wff can simply be confirmed to be valid if every interpretation of a truth table is true. Alternatively the *reductio* method can be used to confirm validity; assume the formula has an interpretation that is not true, to see if this is consistent.

Definition 16: Implication (→)

Four-valued implication extends the concept of Boolean implication, and is associated with the truth table of the material conditional shown in Table 3.4 below. Implication is true whenever the antecedent is less true than the consequent, as determined by the complete lattice truth order. In short, A→B means A $\leqslant_t$ B.

A → B	f	u	i	t
f	t	t	t	t
u	f	t	t	t
i	f	f	t	t
t	f	f	f	t

Table 3.4: Truth table for multi-valued implication

A	A→A
f	t
u	t
i	t
t	t

Table 3.5: Truth table showing (A→A) is a tautology

Definition 17: Provability (⊢)

A ⊢ B means there are a series of formation rules and/or theorem steps that can transform A into B.

Definition 18: Entailment (⊨)

Entailment is semantically different to implication or provability. Entailment relates logic terms rather than atoms. A ⊨ B means B is always a consequence of A existing; i.e. every interpretation in which A is true also has B being true.

3.3.2 Consistency and decidability

Definition 19: Inconsistency

In Boolean logic, as shown in Table 3.6 below:

$A \equiv (A=t)$

$\neg A \equiv (A=f) \equiv \neg(A=t)$

$\therefore (A\wedge\neg A) \equiv ((A=t)\wedge\neg(A=t))$

Also, $((A=t)\wedge\neg(A=t)) \equiv ((A=B)\wedge\neg(A=B))$

Therefore, this thesis proposes redefining inconsistency as: $(A=B)\wedge\neg(A=B)$

A	B	A=f	A=t	A=B	(A=t)∧¬(A=t)	(A=B)∧¬(A=B)	A∧¬A	(A=B)∨¬(A=B)
f	f	t	f	t	f	f	f	t
f	t	t	f	f	f	f	f	t
t	f	f	t	f	f	f	f	t
t	t	f	t	t	f	f	f	t

Table 3.6: Boolean logic truth table proof $(A\wedge\neg A) \equiv ((A=B)\wedge\neg(A=B))$ for all B.

This redefined definition of inconsistency is equivalent to the classical definition, $(A\wedge\neg A)$ in Boolean logic. However, $(A=B)\wedge\neg(A=B)$ remains an inconsistency in multi-valued logic as shown by truth table 3.7 below:

(A=B)∧¬(A=B)	f	u	i	t
f	f	f	f	f
u	f	f	f	f
i	f	f	f	f
t	f	f	f	f

Table 3.7: Proof $(A=B)\wedge\neg(A=B)$ is a contradiction in 4PL.

A	A∧¬A	A∨¬A
f	f	t
u	u	i
i	u	i
t	f	t

Table 3.8: Proof $A\wedge\neg A$ is not a contradiction and $A\vee\neg A$ is not a tautology in 4PL.

Table 3.8 shows that $(A\wedge\neg A)$ is not a contradiction in multi-valued logic. However, table 3.7 shows that the Boolean equivalent statement $(A=B)\wedge\neg(A=B)$ is a contradiction in both Boolean and 4PL. The redefinition is semantically justified as it more accurately conveys the inconsistency of saying a proposition is both one particular value, and simultaneously not that value. The redefined concept of inconsistency is thus syntactically and semantically justified.

Definition 20: Consistency

Consistency is the opposite of inconsistency. Semantically it means not having a situation where simultaneously, $(A \leqslant_t B)$ and $(B <_t A)$ holds for any two proposition A and B. This can be expressed as having the tautology $(A=B)\vee\neg(A=B)$ hold. This is why preserving this semantically more accurate version of the tautology is so important for ensuring consistency, as well as completeness.

Definition 21: Decidability

If there are three propositions, {L,C,U}, such that $L \leqslant_t C \leqslant_t U$, then C is decidable if $L =_t U$ for all proposition interpretations, i.e. the truth values of the upper and lower bounds are identical. If $L =_t U$ for all proposition interpretations then it must be the case $L =_t C =_t U$ for all proposition interpretations; therefore the truth value of C can always be determined, and its value is the same as L and U.

This is syntactically expressed by the premise: $\{L\rightarrow C, C\rightarrow U, L\equiv U\}$. Using the contrapositive theorem this is the same as: $\{L\rightarrow C, (\neg U)\rightarrow\neg C, L\equiv U\}$

3.3.3 Generally useful theorems

Theorem 1

All three primitive connectives, and equivalence, are commutative:

$(A\wedge B) \equiv (B\wedge A)$

$(A\vee B) \equiv (B\vee A)$

$(A=B) \equiv (B=A)$

Proofs: this is obvious due to the fact that the functions are symmetric around truth table diagonals.

Theorem 2

$(A \equiv B) \vDash (A \vDash B)$

$(A \equiv B) \vDash (B \vDash A)$

By definition $(A \equiv B)$ means every interpretation's truth value of A is identical to the truth value of B. Therefore, every interpretation of A which is true, must also have B as true, which by definition means $(A \vDash B)$. Therefore, $(A \equiv B) \vDash (A \vDash B)$ holds.

$(A \equiv B) \vDash (B \vDash A)$ similarly holds as equivalence is a commutative relationship.

Theorem 3

$\{(A \vDash B), (B \vDash A)\} \nvDash (A \equiv B)$

Proof by contradictory example:

A	B
f	u
u	f
i	i
t	t

Table 3.9: Contradictory example showing $\{(A \vDash B), (B \vDash A)\} \nvDash (A \equiv B)$

In the above interpretation, $(A \vDash B)$ and $(B \vDash A)$ are Satisfied, but $(A \equiv B)$ is not Satisfied.

Theorem 4: Contrapositive

$(A \rightarrow B) \equiv (\neg B \rightarrow \neg A)$

Proof: Truth table for $\neg B \rightarrow \neg A$ is identical to the truth table of $A \rightarrow B$; therefore the two are equivalent.

$\neg B \rightarrow \neg A$	f	u	i	t
f	t	t	t	t
u	f	t	t	t
i	f	f	t	t
t	f	f	f	t

Table 3.10: proof $A \rightarrow B \equiv \neg B \rightarrow \neg A$

Theorem 5

$(A \rightarrow C) \wedge (B \rightarrow C) \equiv ((A \vee B) \rightarrow C)$

Proof: $A \rightarrow C$ means $A \leqslant_t C$, $B \rightarrow C$ means $B \leqslant_t C$, $(A \vee B)$ means $A \vee B =_t A \leqslant_t A$ or $A \vee B =_t B \leqslant_t B$. Therefore $A \vee B \leqslant_t A \leqslant_t C$ or $A \vee B \leqslant_t B \leqslant_t C$, either way $A \vee B \leqslant_t C$, therefore $(A \vee B) \rightarrow C$ holds.

Theorem 6

$(A \rightarrow B) \wedge (A \rightarrow C) \equiv (A \rightarrow (B \wedge C))$

Proof: $A \rightarrow B$ means $A \leqslant_t B$, $A \rightarrow C$ means $A \leqslant_t C$, $(B \wedge C)$ means $B \wedge C =_t B$ or $B \vee C =_t C$. Therefore $A \leqslant_t B =_t B \wedge C$ or $A \leqslant_t C =_t B \wedge C$, either way $A \leqslant_t B \wedge C$, therefore $A \rightarrow (B \wedge C)$.

Theorem 7

If proposition C is decidable, then C is consistent.

Proof: decidability means $L =_t C =_t U$, where L = greatest lower truth bound of C and U = least upper truth bound of C. Therefore, $(C=B) \vee \neg(C=B)$ must hold as C can only have one possible truth value, to which B will either be equal, or not equal.

Theorem 8: $A \equiv B \vDash \neg A \equiv \neg B$

As negation is applied to every interpretation truth value of A and truth value of B, they remain the same, as negation is a compositional operator.

Theorem 9

De Morgan's laws hold for $\{\wedge,\vee\}$:

$\neg(\neg A \wedge \neg B)$	f	u	i	t
f	f	u	i	t
u	u	u	i	t
i	i	i	i	t
t	t	t	t	t

Table 3.11: De Morgan's laws hold proof

The above truth table for $\neg(\neg A \wedge \neg B)$ is equivalent to the $(A \vee B)$ truth table. Therefore, $(A \vee B) \equiv \neg(\neg A \wedge \neg B)$.

The fact $A\equiv\neg\neg A$, combined with use of substitutions $\neg A$ for A, $\neg B$ for B, means:

$\neg(\neg A \vee \neg B) \equiv \neg\neg(\neg\neg A \wedge \neg\neg B) \equiv \neg\neg(A \wedge B) \equiv (A \wedge B)$.

Therefore, $(A \wedge B) \equiv \neg(\neg A \vee \neg B)$, and therefore, De Morgan's laws hold for $\{\wedge,\vee\}$.

3.3.4 Compositionality via Interactive Truth Tables

Theorem 10

$((C=B)=u) \equiv ((C=B)=i) \equiv f$ for all C, B

Proof: $(C=B) \in \{f,t\}$ for all C, B as the = operator is composed of only f or t interpretations. Therefore $((C=B)=u) \equiv f$ and $((C=B)=i) \equiv f$ as $(C=B)$ can never be u or i.

Theorem 11

It is possible to construct any truth table function from $\{f,u,i,t,=,\wedge,\vee\}$. Each cell in the truth table is one particular interpretation. A formula of the form $(A=a)\wedge(B=b)$ with set constants $a,b \in \{f,u,i,t\}$ will assign false to every cell except the cell where A=a and B=b which is assigned true. This means a formula of the form $(A=a)\wedge(B=b)\wedge v$ with set constants $a,b,v \in \{f,u,i,t\}$ will assign false to every cell except the cell where A=a and B=b, which is assigned the value v.

By combining 4^n $(A=a)\wedge\ldots\wedge(B=b)\wedge v$ style conjunction formulae via the disjunctive ($\vee$) connective, any truth table assignment can be constructed, where n is the number of propositions in the formula. This works as each conjunctive formula determines the truth value of a single cell, assigning that cell the value v, and the entire disjunctive formula therefore assigns the appropriate value to each cell.

As each interpretation is given one particular truth value, which is dependant only on the truth values of the propositions, any compositional formula can be expressed by any equivalent disjunction of conjunctions formula. Therefore a syntax only needs to be able to express the three primitive connectives in order to handle any arbitrary compositional formula. However, this formulation is not as elegant as an ordered list of rules.

3.3.5 Rule salience and defeasibility

Defeasibility is, in essence, the study of situations where rules contradict each other. Let us look at a simple example belief set:

$\{ A \rightarrow C, B \rightarrow \neg C, A{=}t, B{=}t \}$

It is clear, if this is the complete belief set, then an agent must conclude $((C{=}t) \wedge (\neg C{=}t))$ which is logically equivalent to $((C{=}t) \wedge \neg(C{=}t))$, which by definition makes the agent's beliefs inconsistent. The only way an agent can make its beliefs consistent is to alter, or remove, at least one of the elements of its belief sets. However, doing so may make its belief set incomplete, as some elements of its belief set may not be believed to be true any more.

Two such alterations might be a revised belief set of:

1) $\{ A \rightarrow C, (\neg A \wedge B) \rightarrow \neg C, A{=}t, B{=}t \}$, or
2) $\{ (A \wedge \neg B) \rightarrow C, B \rightarrow \neg C, A{=}t, B{=}t \}$

Both these revised belief sets preserve the basic knowledge of the first belief set, but now ensure consistency. Methods for dealing with inconsistent and paraconsistent logic have been surveyed. [see Besnard and Hunter, 1998; Coste-Marquis and Marquis, 2008].

Definition 22: Positive Headed Horn Clause

A positive headed horn clause (phhc) is a rule (formula) of the form:

$A_1 \wedge A_2 \wedge \ldots \wedge A_n \rightarrow C$

where C is the conclusion, consequent, or head of the horn clause, and $A_1, A_2, \ldots, A_n$ are all antecedents, i.e. the body, of the horn clause.

In the literature, a positive headed Horn clause is referred to simply as a Horn clause. However, in this thesis, it is distinguished as a positive headed Horn clause, because this thesis deals with multi-valued Horn clauses.

Definition 23: Negative Headed Horn Clause

A negative headed horn clause (nhhc) is a rule (formula) of the form:

$A_1 \wedge A_2 \wedge \ldots \wedge A_n \rightarrow \neg C$

where C is the conclusion of the horn clause, and $A_1, A_2, \ldots, A_n$ are all antecedents.

In the literature, a negative headed Horn clause is referred to simply as a goal. However, in this thesis it is referred to as a negative headed Horn clause; it is a multi-valued Horn clause.

Several positive horn clauses in a row can be compacted to only one; likewise several negative horn clauses can be compacted to only one. Using theorem 5, { $A_1 \rightarrow C, A_2 \rightarrow C$} becomes { $(A_1 \vee A_2) \rightarrow C$ }

The belief revision example given at the beginning of the section shows a major source of inconsistency in knowledge based systems; negative and positive headed horn clauses with the same head atom can contradict each other. This source of inconsistencies and contradictions can be eliminated immediately; a consistency preserving alteration can be obtained from a human expert when a new horn clause is added to the knowledge base, by using a salience ordering of the rules, where priority is determined by the human expert.

The solution used in eGanges River logic is to recognise the hierarchical nature of horn clauses, which includes an inherent priority (salience) of the horn clauses, and to use an expert heuristic that ensures that a human expert only provides consistent rules to the knowledge base, without removing the ability to establish $\neg C$ as true.

Definition 24: Monad Ordered Rule Set (MORS)

A Monad Ordered Rule Set is an ordered set of Positive Headed Horn Clauses and Negative Headed Horn Clauses, where:

1) Every Horn Clause Head is the same proposition, C, called the conclusion.
2) Every non-conclusion proposition in every Horn clause is independent of every other non-conclusion proposition.
3) The order of the Horn clauses determines their priority; priority is also called salience.

For example, the following is a MORS:

$\{(A \rightarrow C),(B \rightarrow \neg C),(D \rightarrow C),(E \rightarrow \neg C)\}$

In the syntax introduced, $(A \rightarrow C)$ comes first and therefore has the highest salience, $(E \rightarrow \neg C)$ comes last and has the lowest salience.

MORS can easily have the simple heuristic “ensure lower salience Horn clauses are consistent with higher salience conflicting Horn clauses” applied to them. This heuristic means:

1) a positive headed Horn clause must be consistent with all higher salience negative headed horn clauses (it is automatically consistent with higher salience positive headed horn clauses), and
2) a negative headed Horn clause must be consistent with all higher salience positive headed horn clauses (it is automatically consistent with higher salience negative headed horn clauses).

If the negative headed horn clause, $B \rightarrow \neg C$, has lower salience than the positive headed horn clause, $A \rightarrow C$, then it must be altered to: $(\neg A \wedge B) \rightarrow \neg C$ to ensure consistency. Applying the contrapositive theorem results in: $(\neg A \wedge B) \rightarrow \neg C \equiv C \rightarrow \neg(\neg A \wedge B)$. Combined with De Morgan's law, this gives: $(\neg A \wedge B) \rightarrow \neg C \equiv C \rightarrow (A \vee \neg B)$. Thus these two rules form the relation, which is shown in table 3.12 below:

$A \rightarrow C \rightarrow (A \vee \neg B)$

$A \rightarrow C \rightarrow (A \vee \neg B)$	f	u	i	t
f	$f \leqslant_t C \leqslant_t t$	$f \leqslant_t C \leqslant_t i$	$f \leqslant_t C \leqslant_t u$	$C =_t f$
u	$u \leqslant_t C \leqslant_t t$	$u \leqslant_t C \leqslant_t i$	$C =_t u$	$C =_t u$
i	$i \leqslant_t C \leqslant_t t$	$C =_t i$	$C =_t i$	$C =_t i$
t	$C =_t t$	$C =_t t$	$C =_t t$	$C =_t t$

Table 3.12: Truth Analysis Table for C, given $A \rightarrow C \rightarrow (A \vee \neg B)$

If the heuristic that C is the highest possible truth value is used, the value of C is equivalent to $(A \vee \neg B)$, as shown in table 3.13

$(A \vee \neg B)$	f	u	i	t
f	t	i	u	f
u	t	i	u	u
i	t	i	i	i
t	t	t	t	t

Table 3.13: Truth table for $(A \vee \neg B)$

This heuristic of choosing the highest possible truth value is consistent with the fact that the positive headed horn clause dominates the negative headed horn clause; the positive headed horn clause tries to raise the truth value of the head proposition, whereas the negative headed horn clause tries to lower the truth value of the non-negated head.

If instead, the negative headed horn clause, $B \rightarrow \neg C$, has higher salience than the positive headed horn clause, $A \rightarrow C$, then it is the positive horn clause that must be altered to $(A \wedge \neg B) \rightarrow C$ to ensure consistency. Applying the contrapositive theorem results in: $B \rightarrow \neg C \equiv C \rightarrow \neg B$. Therefore these two rules form the relation, which is shown in table 3.14 below:

$(A \wedge \neg B) \rightarrow C \rightarrow \neg B$

$(A \wedge \neg B) \rightarrow C \rightarrow \neg B$	B=f	B=u	B=i	B=t
A=f	$f \leqslant_t C \leqslant_t t$	$f \leqslant_t C \leqslant_t i$	$f \leqslant_t C \leqslant_t u$	$C =_t f$
A=u	$u \leqslant_t C \leqslant_t t$	$u \leqslant_t C \leqslant_t i$	$C =_t u$	$C =_t f$
A=i	$i \leqslant_t C \leqslant_t t$	$C =_t i$	$C =_t u$	$C =_t f$
A=t	$C =_t t$	$C =_t i$	$C =_t u$	$C =_t f$

Table 3.14: Truth Analysis Table for C, given $(A \wedge \neg B) \rightarrow C \rightarrow \neg B$

If the heuristic that C is the lowest possible truth value is used, the value of C is equivalent to (A∧¬B) as shown in truth table 3.15.

(A∧¬B)	f	⊤	⊥	t
f	f	f	f	f
u	u	u	u	f
i	i	i	u	f
t	t	i	u	f

Table 3.15: Truth table for (A∧¬B)

This heuristic of choosing the lowest possible truth value is consistent with the fact that the negative headed horn clause dominates the positive headed horn clause, and a positive headed horn clause tries to raise the truth value of the head proposition, whereas the dominant negative headed horn clause tries to lower the truth value of the non-negated head.

In short, if positive and negative headed horn clauses are used and have the following three heuristics applied to them:

1) lower salience horn clauses must be altered, if necessary, to ensure consistency with higher salience horn clauses,
2) where a negative horn clause dominates, the head's value is the minimum possible, and
3) where a positive horn clause dominates, the head's value is the maximum possible, then the 4PL and eGanges approach of using an equivalence formula is logically equivalent to using horn clause approach.

Converting between the Monad Ordered Rule Set and an eGanges River is easily automatable. First ensure all positive headed horn clauses have the negation of the antecedents of all higher salience negative headed horn clauses added to their conjunction. Then ensure all negative headed horn clauses have the negation of the original antecedents of all higher salience positive headed horn clauses added to their conjunction. This ensures consistency. Next combining all positive horn clauses using theorem 5, results in a single rule of the form, L→C. Then combine the contrapositive of all negative headed horn clauses to obtain a single rule of the form C→U.

L→C and C→U combine to mean $L \leqslant_t C \leqslant_t U$, i.e. L represents the lower bound on the truth value of C and U represents the upper bound on the truth value of C. C is consistent provided $L \leqslant_t U$ for all proposition interpretations; this has been assured by the salience rule alteration procedure.

3.4 eGanges syntax and semantics

The eGanges syntax is predominantly a graphical knowledge representation that is equivalent to that of 4PL. The semantics of eGanges and 4PL rely on a variant of negation as failure: rather than defaulting to false; eGanges and 4PL default to incomplete/unanswered. This default logic aspect makes eGanges and 4PL non-monotonic logics, and ensures they are always decidable.

3.4.1 eGanges nodes

A **normal node** in eGanges represents a predicate symbol. Every node encapsulates one yes/no/uncertain answerable question. This question effectively asks the user to provide the terms and interpretation mapping to convert the node into one of the three non-default truth-constants: false, uncertain or true. Until the user answers the node's question, the eGanges heuristics will default the truth-constant to incomplete/unanswered.

Another possible type of node is the **constant-value node**. Making a node represent a specific truth value may seem redundant, but it can have its uses. **Neutral nodes** for example are almost a constant-value node; they can have only the truth-values of unanswered or positive. Their purpose is to inform an end user of relevant information, not effect the outcome.

The purpose of a neutral node is to represent situations where a proposition is known to obey the classical tautology, i.e. states where the truth-value of a proposition is irrelevant (see section 3.2.4). If the eGanges application builders know that a proposition truth-value has no consequent for the Final result, they can set the node representing that proposition as a neutral node.

The phrasing of the question, and the assignment of answers can enable another type of node, the **embedded-uncertain node**. This node type asks a specific question that embeds any uncertainty within it. For example, “Is there any reasonable doubt about the defendant's guilt?” has the end user's uncertainty embedded within the question, and enables the user to answer yes or no only. The question asked in such a node can enable specific truth values to be extracted. This is equivalent to making a node represent (A=v), where A represents the node's proposition and v represents a particular truth-constant.

3.4.2 eGanges rivers

The eGanges nodes can be linked to form a **river**. A river consists of a single **consequent node** and at least one **antecedent node**. The consequent node is identified by the fact that it is at one end of the river, and has an arrow pointing to it just before that river ends; where the other end of the river has no arrow pointing to the other end node (the initial node). Thus, it is always possible to tell which node is the consequent. The eGanges river heuristic ensures the consequent node's logic value is at least as great as the conjunction of all the antecedent node logic values. As such a single river is equivalent to an implication rule. The nomenclature of eGanges identifies rivers by calling them upstreams of their consequent node.

A river system is a collection of rivers, where node overlaps result in links between the rivers. The consequent of one river may be the antecedent of another river, and if it is, then the river with the node as a consequent is called an upstream river of the river with the node as an antecedent, as well as an upstream of the river linking node. Where two or more rivers share the same consequent node, they form a disjunction **fan**. Due to the eGanges heuristics for a single river, eGanges ensures the shared consequent node's logic value is at least as great as the greatest logic value from all the conjunction logic values.

The eGanges heuristic has unanswered as the default value for a node when it has no upstreams. However, if a node does have upstreams, the eGanges heuristic makes the node's logic value be as low as possible, while ensuring the eGanges river heuristic is still obeyed. This makes the node equivalent to the greatest lower bound of its upstreams. This is also equivalent to making the consequent node equivalent to the disjunction of the logic values of its upstreams. As the logic value of each upstream represents a conjunction of its antecedent nodes, a consequent node's logic value becomes equivalent to a formula representing a disjunction of conjunctions, i.e. a formula in disjunctive normal form[5]. Given the possible node types, the eGanges River system can thus represent any compositional formula, as described in section 3.3.4.

3.4.3 eGange example

In the example below, there are six possible rule priority orders, assuming two rules can not have the same precedence. These are:

1) $(A \rightarrow C), (B \rightarrow C), (F \rightarrow \neg C)$
2) $(A \rightarrow C), (F \rightarrow \neg C), (B \rightarrow C)$
3) $(B \rightarrow C), (A \rightarrow C), (F \rightarrow \neg C)$
4) $(B \rightarrow C), (F \rightarrow \neg C), (A \rightarrow C)$
5) $(F \rightarrow \neg C), (A \rightarrow C), (B \rightarrow C)$

5 A formula is considered to be in disjunctive normal form if and only if it is a disjunction of one or more conjunctions of one or more literals

6) $(F \rightarrow \neg C)$, $(B \rightarrow C)$, $(A \rightarrow C)$

eGanges requires that all possible rules are covered in its rule database. This validated, complete enforceable rule knowledge base does not mean future potential rules can not be added; it merely means that any future rule changes are not enforceable at the moment, i.e. that all currently enforceable rules are covered in the river system.

As the tributary rules fully cover all current possible rules to establish the conclusion, the tributaries act as a logical equivalence to the conclusion, i.e. like a logical parsing of the conclusion. Given this logical equivalence, it is possible to have the truth values of an only positive rule system behave as a logically equivalent system to a combination of positive and negative multi-valued horn clauses.

Figures 3.1 to 3.4 show how the six different rule precedents would be captured by eGanges. In eGanges, as with any knowledge based system, it is the job of the expert to determine which of these rule precedent orders are correct, as part of the expert validation process; the graphical nature of the maps and their modular abstractability helps to ensure the expert does not get lost in the logic.

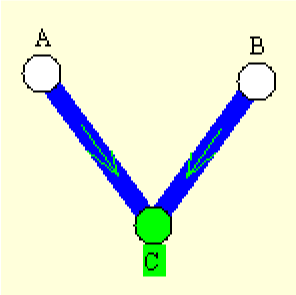


Figure 3.1: eGanges map of $\{(A \rightarrow C),(B \rightarrow C),(F \rightarrow \neg C)\}$ or $\{(B \rightarrow C),(A \rightarrow C),(F \rightarrow \neg C)\}$

The figure above shows that if A or B are the only ways to establish C, then the truth value of F doesn't matter as $F \rightarrow \neg C$ is the rule of least precedence (case 1 and 3); If A and B are false, C will become false.

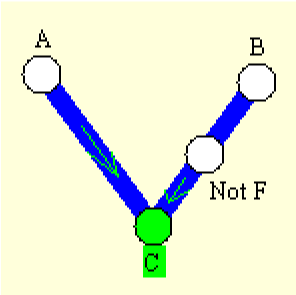


Figure 3.2: eGanges map of $\{(A \rightarrow C),(F \rightarrow \neg C),(B \rightarrow C)\}$

The figure above represents case 2, where $A \rightarrow C$ is dominant, and B can only establish C if F has not been established.

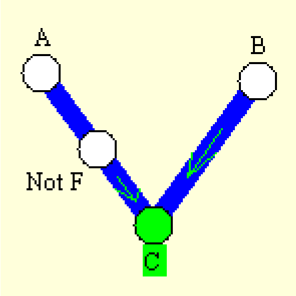


Figure 3.3: eGanges map of {(B → C),(F → ¬C),(A → C)}

The figure above represents case 4; this time B → C is dominant and A → C requires Not(F) in order to establish C, otherwise F can ensure that C is not established.

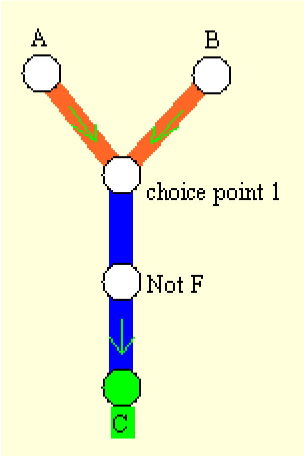


Figure 3.4: eGanges map of {(F → ¬C),(A → C),(B → C)} or {(F → ¬C),(B → C),(A → C)}

Cases 5 and 6 are both represented in figure 3.4 above where F → ¬C is the dominant rule. In this case "Not F" must be established to have any chance of establishing C. Notice how this interpretation, introduces a choice point node into the River.

3.4.4 Unanswered default logic

The eGanges knowledge representation has a default value of unanswered, which represents the situation where there is uncertainty due to a lack of knowledge. All nodes, that are not equivalent to the conclusions of their upstreams, default to unanswered, until additional information is added which changes their truth value to negative, uncertain or positive.
Default logics [Reiter, 1980] use syntax to distinguish between strict facts and default rules, and identify different extensions of the default logic as potential ways of "making sense" of seemingly conflicting or missing information. The Closed World Assumption is one version of default logic; it assumes that any facts that are not known are assumed false.
An advantage of multi-value logics, like eGanges logic, is that they naturally lead to a default which does not require the assumption of false or truth; unknown arguments can be specifically treated as unanswered or unknown.

3.5 AnsProlog language structure

Answer **S**et **Pro**gramming L**og**ic (AnsProlog) is a popular group of epistemologies in the computing field, that are generally based on a stable model semantic or a well-founded model semantic. Regardless of which semantic it is based on, AnsProlog is capable of at least three-valued reasoning. As AnsProlog is a standard, well studied, powerful and expressive language, it is a useful language for comparison with eGanges River logic, but only a basic overview will be given.

3.5.1 AnsProlog syntax and semantics

AnsProlog also has seven primitive types of symbols as its axiom alphabet:

1) object constants;
2) variables;
3) function symbols;
4) predicate symbols;
5) connectives;
6) punctuation symbols; and
7) the special symbol $\perp$.

The first four types are identical to 4PL.

The available connective symbols for AnsProlog come from the set { ¬, **not**, ←, **or**, ',' }, which are called respectively: explicit negation, negation by default, 'implied by', 'or', and 'and'. As will be discussed later, 'implied by' is not the same as material implication; 'or' is not the same as disjunction ($\vee$); and 'and' is not the same as conjunction ($\wedge$).

The available punctuation symbols for AnsProlog come from the set { '(', ')', '.' }

An AnsProlog rule is of the form:

L_0 **or** L_1 **or** … **or** L_k ← L_{k+1}, L_{k+2}, … , L_n, **not**(L_{n+1}), **not**(L_{n+1}), … , **not**(L_m)

where $k \geq 0$, $m \geq n$, the Ls are literals, and **not**(L) means L is false by default (i.e. L is not shown to be true).

Stable semantic and well-founded semantic

A stable semantic [Gelfond and Lifschitz, 1988] allows a predicate to resolve only to a value from {true, false}, whereas a well-founded semantic [Gelder, et al, 1991] allows a predicate to resolve to a value from {true, false, unknown}.

3.5.2 Four-valued nature of AnsProlog

The purpose of AnsProlog is to determine an answer set for an AnsProlog program. An answer set can contain an atom (positive literal), or the explicit negation of the atom (negative literal), or it can contain neither the positive nor negative form of a literal. If the answer set contains only the positive literal form, the atom is true. If it contains only the negative literal form, the atom is false. If it contains neither, the atom is unknown, and if it contains both, the atom is inconsistent. These four possibilities are effectively epistemic truth values equivalent to Belnap's four-valued logic. A practical reasoning system can not avoid dealing with these four possibilities.

3.5.3 AnsPrologs equivalent problems

Like 4PL and eGanges, AnsProlog can not avoid the fact that its (P **or not** P) is not a tautology [Baral, 2003]. It has deferred the problem onto different operators, 'or' and 'not', so the $\vee$ and $\neg$ operators can preserve the tautology ($P \vee \neg P$).

The not operator causes other complications. In classical logic, ($A \rightarrow B$) is equivalent to ($B \vee \neg A$). (B or not(A)) is not a tautology, so trying to make it equivalent to ($B \leftarrow A$) causes potential logical problem has you can not rely on ($A \leftarrow A$) as shown by the truth table below.

$B \vee \neg A$	f	u	i	t
f	t	t	t	t
u	i	i	i	t
i	u	u	i	t
t	f	u	i	t

Table 3.16: Truth table of ($B \vee \neg A$) in four-valued complete lattice logic

In short, the problems complained about by Dubois appear to be unavoidable. At best all that can be done is to create new operators to absorb the problems so the standard operators can preserve their classical properties.

3.6 *Verification and validation of eGanges superexpertise*

Definition 25: Syntactic Theorem

To be a syntactic theorem, a formula must be syntactically derivable from the axioms of a system by use of the derivation and formation rules, i.e. there must be a series of steps (a proof) that shows that the theorem is a wff.

The most important derivation rule is *modus ponens*: $((A \rightarrow B) \wedge A) \vdash B$

Definition 26: Soundness

To be sound, every syntactic theorem of a system must be valid (true for every interpretation)

Definition 27: Completeness

To be complete, every wff that is valid must be a syntactic theorem of the system.

Definition 28: Verification has traditionally been broken up into basically two categories: consistency and completeness [Gonzalez and Dankel 1993].

An expert system application needs to use a system methodology that is verified by thorough analysis. Once the method is appropriately verified, the expert knowledge of the system needs to be captured within the system and that knowledge needs to be validated by a field expert. This validation effectively means the logic needs to be shown to be sound. Traditionally these two aspects have both involved a knowledge engineer and are often treated almost interchangeably in the literature. The design of eGanges and NeGame aims to ensure their KR methods are easy enough to understand so that anyone can be a knowledge engineer, thus allowing field experts to do their own knowledge engineering and validation as they construct applications.

3.6.1 Completeness in the Domain of Discourse

It has been shown in section 3.3.2 that 4PL and eGanges logic do preserve the semantically more accurate redefined classical tautology, while remaining compositional. eGanges also ensures decidability and therefore consistency. As such, any field where eGanges can be shown to be complete will verify eGanges for that field. Clearly an incomplete application can not be complete, so it will be assumed that a field expert has created an application that contains all appropriate rules for the application.

In a field of knowledge, the domain of discourse is the set of entities over which certain variables of interest may range. In essence, it is a subset of the Herbrand Universe that the semantics seek to resolve. For example, in the law, the elements of the domain of discourse include the defendants whose cases must be processed. In the eGanges epistemology, when a defendant is to be categorised as liable or not liable, s/he is processed (as a grounded term) according to the appropriate validated eGanges predicate map (i.e. river system).

The answers given to the node questions resolve each node to an appropriate answered truth-constant; the predicate has been supplied with its required terms and interpretation by answering the node question. Compositionality ensures complex formulae can be reduced to a single truth-constant, provided each predicate has been reduced to a truth-constant; this includes the complex formula to determine if the defendant is liable or not.

As a defendant's liability is entirely determined by the rules captured in the eGanges map, the categorisation of the defendant as either being in the subset “liable” or subset “not liable” is entirely determined by the syntactic application of the rules. Therefore, for fields where the categorisation is actually determined by the rules, the system is necessarily logically sound and complete as the entailments of the domain of discourse are equivalent to what is syntactically provable.

For example, let us assume the predicate map is simply: $\{ A \rightarrow C, (\neg A \wedge F) \rightarrow \neg C, (B \wedge \neg F) \rightarrow C \}$. Defendant X consults eGanges and produces the knowledge base $\{ A_X \rightarrow C_X, (\neg A_X \wedge F_X) \rightarrow \neg C_X, (B_X \wedge \neg F_X) \rightarrow C_X, A_X = t, B_X = f, F_X = t \}$ and therefore eGanges can syntactically prove the conclusion $C_X = t$, which is necessarily how the defendant will be categorised. This predicate map is identical to the predicate map of figure 3.2.

3.6.2 Limits to completeness

Gödel's incompleteness theorem relies on representing theorems about basic number theory as expressions in a formal language, and then representing this language within number theory itself. Consequently, Gödel showed there will always be statements about the natural numbers that are true, but that are unprovable within the system – they are provably true in other systems. Gödel's incompleteness theorem relies on this method's ability for a theorem to refer to itself, either directly or indirectly, causing an infinite resolution loop, ie a paradox. “This sentence is false” is the most famous paradox example. A paradox is the essence of a contradiction: “This sentence is false” = true ↔ “This sentence is false” = false, i.e. the value of the sentence must simultaneously be both true and false.

When discussing this problem, Brachman and Levesque [2004, p.10] conclude:

> “Any procedure that always gives us answers in a reasonable amount of time will occasionally either miss some entailments or return some incorrect answers”

eGanges prevents paradoxes by being acyclic, thus breaking direct and indirect self-referencing. However, provided there is no negation self-referencing, a river system can be cyclic and remain consistent.

3.6.3 Knowledge acquisition and validation

The fields of computing, mathematics and philosophy have provided a rich variety of methods and perspectives on what knowledge is, and how it can be captured. Thus far, it can be argued that the field of AI has two main approaches: the first is to have a programmer encode an expert's knowledge and intelligence into a program (the programmer and expert may be the same person), and the second is to use training sets of data along with algorithms so that a computer can try and learn a desired classification relationship. Yet, we can not necessarily be certain that the machine has learned the correct relationship; if the machine's learned results can be encoded in a way that a human expert can follow, then the learning may be supervised and validated.

Early in the field of AI, Boehm [1984] discussed how important expert validation is. While humans are far from perfect, they are still the most flexible and intelligent double checking option we have. The easier it is for a human expert to double check a machine's knowledge, the more reliable the checking, and thus the machine's knowledge, will be.

This thesis synthesised various approaches to knowledge representation, with the objective of thoroughly investigating the eGanges knowledge representation, to see its virtues and limitations. This synthesis has revealed new ways of looking at knowledge and adapts previous ideas within this new light to help further advance our understanding of knowledge representation, nonmonotonic reasoning and the partial automation of expertise.

By showing the equivalence between two knowledge representations, with appropriate applied heuristics, the thesis also opens the way for results from machine learning techniques to be more readily validated by human experts; the knowledge learned by a machine as horn clauses can be converted to eGanges graphics and validated by a human expert in the user-friendly graphical interface of eGanges. However, development of such machine learning techniques are not considered in this thesis. Also, a human computer interface (HCI) study of the results is not undertaken, although some basis for such a study is set out in this thesis – such a study might test the effectiveness of the River approach versus a salience ordered list of rules.

3.6.4 Failure of ad hoc approaches

If knowledge is not systematically organised, it takes longer to find information within the knowledge base. This delay in knowledge access time slows processing time. More importantly, disorganised knowledge bases will make it hard for a human expert to validate the knowledge base as the knowledge will not be presented in a structured, easy to understand manner for checking.

Just as abstraction, encapsulation and data hiding in object-oriented programming support reuse, thereby providing quicker creation of solutions using validated components [Liberty, 1998]; knowledge abstraction, encapsulation and separation makes knowledge maintenance compartmentalised, thereby requiring less resources of human experts for each section of knowledge. Using an expert's epistemology can help achieve these compartmentalisations This use of the expert's knowledge representation as a specification, means the ideals proposed by Boehm [1984] of verification (“Am I building the product right?”), validation (“Am I building the right product?”) and ensuring the expert's specifications are complete and consistent can be fulfilled for eGanges logic in this thesis.

The eGanges epistemology uses a predominantly graphical syntax that can be viewed as encoding salient multi-valued Horn clauses. This ensures it is decidable and consistent. The simplicity of this method implies that River logic obeys Occam's razor.

The ancient Greeks believed geometry was the purest philosophy; geometry to them was a visual analogical representation of nature. Other visual representations of an analogical nature include the Tree of Porphery which was a visualisation of Aristotle's ontology of substance, and the Ramist graphics, along with other 2d ideographs, representing other epistemologies, such as those of Venn's diagrams [Venn, 1880], Peirce's logic graphics [1982, 1984, 1986], Korzybski's logic objects [1958] and Ishikawa's fishbone.

Ishikawa (1985) developed the concept of the quality control fishbone to represent causation in a manufacturing process, to facilitate quality control in manufacturing in Japan after the second World War. The fishbones of Ishikawa provided a two dimensional model of sequential information which avoided the repetition of factors in the alternative pathways of a tree flowchart. A River is a multi-valued nodalised Ishikawa River.

3.7 Chapter Three Summary

This Chapter extends the work of others, in particular Majkic [2004, 2008], Fitting [2006] and Ginsberg [1988], and shows that eGanges and River logic can be viewed as an adaptation of Belnap's four-valued logic [Belnap, 1977], c.f. [Majkic, 2004]. This was a more standard thesis exposition to show how difficult proving logical completeness and logical soundness is; instead it focused on showing how the unidirectional nature of the River graph (ie its hierarchical nature) ensures that River systems are always decidable, logically consistent and compositional. While at first glance, classical tautologies are sacrificed for this consistency and compositionality, the work of Majkic [2004] shows they can be recovered by using a two value meta-logic that encapsulates the four-valued epistemic logic. This thesis takes a new and different approach by redefining classical tautologies to make a version that remains a tautology for multi-valued logics.

An argument for completeness for a limited set of fields of expertise is given. Those fields where the grouping of the elements of the domain of discourse are fully determined by syntactic provability, must necessarily be both sound and complete as entailment is logically equivalent to provability for those fields of expertise. This appears to contradict the results of Dubois [2008] who showed that multi-valued logics can not be simultaneously compositional and preserve classical tautologies; without preserving classical tautologies, it can be argued a language can not be complete as an entailment exists without a matching syntactic proof.

This chapter showed that the standard formulation of classical tautologies is inappropriate. A new formulation, which is logically equivalent in Boolean logic, was proposed. This formulation more accurately captures the epistemic knowledge entailed in the tautology, and remains a tautology in eGanges logic, thus negating the incompleteness argument based on Dubois' argument.

In all, this chapter shows eGanges River logic is thus consistent and complete for certain fields of expertise, and is thus verified as an appropriate knowledge representation for those fields [Gonzalez and Dankel, 1993]. The simplicity of River logic follows the principle of Occam's razor, and moreover this knowledge representation has inbuilt methods of ensuring decidability, and therefore consistency. Provided the resulting knowledge bases are completed and validated by human experts, eGanges River logic provides a verified knowledge representation for fields where syntactic provability fully determines all entailments.

Practical reasoning systems must be able to make decisions with potentially incomplete and inconsistent knowledge. There are many methods in the literature to deal with this problem of incomplete and inconsistent knowledge, but few start with a thorough epistemic model of the knowledge to be represented.

The main result of the analysis of the relationship between knowledge and truth was to provide an epistemic reason for the use of a four-truth complete lattice, as the basis for a knowledge representation. These four epistemic truth values are the same as those used in eGanges logic [Gray and Gray, 2003] and in the four valued logic of Majkic [2004].

Chapter 4: Negotiation Superexpertise

This Chapter consists of an introduction to negotiation superexpertise and three published papers that further develop superexpertise for negotiation problem resolution. These papers are focused on designing superexpertise tools to help negotiation resolution of disputes. The eGanges epistemology was originally designed with litigation and general adversarial problems in mind [Gray, 2007a], and many works have been written on this, as well as the papers of Chapter Two which show that eGanges is applicable to several quality control and adversarial problems, illustrated by the case studies of the various eGanges applications in the papers.

The overarching principle of conflict resolution by negotiation is that sufficient issues must be resolved by agreement to reach a Final outcome; it is not an adversarial matter where sufficient rules must be applied to determine a Final outcome of the conflict. Thus, negotiation problems have some significant epistemic differences to adversarial and quality control problems.

The first two papers of this chapter develop an extension to the eGanges shell, namely Negaid, which is designed to aid in negotiating resolutions to disputes. These two papers also discuss the idea of negotiating to prevent potential future disputes; both use the hierarchical River system to ensure all issues and their sub-issues are identified and dealt with in detail. The emphasis is on dispute avoidance, by getting parties to agree on the arrangements to prevent and resolve conflicts before any such conflicts can arise. This approach attempts to draw on the adversarial problem-solving strength of the eGanges River epistemology.

The third paper uses a new River logic that is different to that of the tributary system of eGanges. This paper also poses a multi-valued logic more epistemically appropriate to negotiation. The shell resulting from this new epistemology is called Negotiation Game, NeGame, or NeG for short. The full development of NeG's epistemology is left for future work.

4.1 Motivation for this Chapter

In the legal domain, there are two problems for lawyers in resolving and avoiding conflicts for clients, that use different but overlapping epistemologies: adversarial problems and negotiation problems. Chapter two has shown how eGanges applications can provide superexpertise for adversarial problems. The motivation for this chapter is to provide a more appropriate epistemology (NeG) for negotiation problems, thus enabling the provision of a superexpertise aid for negotiation. This further contributes to the model of superexpertise identified and developed in this thesis.

Many problems, particularly international conflict problems, can not be solved by rule enforcement. This makes the need for an appropriate negotiation epistemology all the more significant. Governments, and potential future country governments (e.g. Israel and Palestine), generally must resolve disputes between each other; even when a third party (e.g. the USA) gets involved, to facilitate negotiation.

Following the introduction of the Internet, many businesses now interact on the global market; this can cause problems that may need to be resolved by negotiation. There are some international conventions that businesses can rely on, although these tend to leave open negotiation problems when dealing with contract enforcement issues.

In domestic matters, many cohabiting couples may find they encounter some negotiation problems. Tools such as Negaid and NeG can help couples to foresee and resolve their problems. The provision of such tools may assist relationships to establish a domestic negotiation lifestyle and thereby reduce the high rate of divorce, domestic violence, financial difficulties, conflicts concerning child rearing responsibilities, etc.

4.2 Harvard Model of Principled Negotiation and Reframing

In the third paper it is shown that the Harvard Negotiation Project produced a model called ‘Principled Negotiation' [Fisher and Ertel, 1995; Fisher, Ury and Patton, 1991] which identified seven elements: interests, options, alternatives, legitimacy, communication, relationship, and commitment. An explicit goal of this third paper is to support these seven elements, though the first two papers implicitly have this goal as well.

Fisher & Ury [1981, p6] say that “Negotiation is a basic means of getting what you want from others”. The first two papers of this chapter attempt to use negotiation to create an agreed eGanges application that will resolve potential future disputes before they arise, by converting such negotiation problems into adversarial problems. This adversarial problem resolution is different to standard Best Alternatives To Negotiated Agreements (BATNAs) as the adversarial problem resolution has been made through negotiation by the parties before the disputes arise. The third paper develops the NeGame epistemology to compensate for Negaid's failings if the dispute as already commenced (i.e. there is no pre-agreed eGanges resolution available).

An important aspect of the communication skill element, is the reframing of the dispute to be resolved [Spencer, 2005, pp.23-4]; how a dispute is described or perceived, is the way it is framed. Reframing tries to change the way the negotiating parties perceive the dispute to enhance the probability of resolving it. For example, if party A perceives that party B wants possession of a holy site merely to frustrate access to that site by members of party A's religion, then that will create animosity between party A and party B; this animosity will greatly hinder the negotiation progress by causing a deterioration in the relationship between the two parties. If communication can be used to reframe party A's perception of party B's desire to possess the holy site, then this animosity and relationship deterioration may be avoided. For example, the site where the temple of Solomon once stood is believed by Muslims to be the location where Muhammad's soul ascended to heaven; this is why the Dome of the Rock is such an important religious site to Muslims; this may be a basis for reframing the conflict.

NeG provides a graphical communication tool that can help implement the Harvard Model of Negotiation by preventing negative emotional body language in communication; parties need only interact via computer if they wish. NeG can not prevent the parties feeling negative emotions towards each other, but it can prevent the unconscious body language communication of those negative emotions. Combined with its ability to systematically handle all the options, and produce eGanges based adversarial resolutions both parties can commit to, NeG can help foster the legitimacy of the negotiation, thereby improving the relationship between the negotiating parties. As new alternatives can easily be added to the River structure, or as glosses on the issues, NeG ensures all interests of both parties can be dealt with systematically. It is also possible to construct a negotiation application explicitly to wholly reframe the conflict.

There are three matters to be settled before the construction of a reframed negotiation application: idiosyncracies, exchanges, and reframing. Some parties have idiosyncratic issues that must be included in the application for resolution in their particular dispute. For example, a spouse may insist on keeping in the bedroom, an urn of the ashes of her deceased first husband; this may offend her current husband and become an important idiosyncratic issue to these parties, requiring resolution.

Potential exchanges that are particularly suited as *quid pro quo* between the parties, should be sought in advance of settling the application. For example, a husband might agree not to let the dog in the bedroom in exchange for not having the urn in the bedroom. The main task in reframing is to get each party to understand the mind set of the other party, in order to bring an empathetic response that can prevent negative emotional responses from both parties.

4.3 Compromises and Trade-offs

Negotiation generally involves compromises and trade-offs. In negotiating over a collection of issues, the values of each party for each issue may differ sufficiently so that both parties can obtain more than half of their subjective values: this situation is called a win-win solution. The negotiation may still have an opportunity cost to it, but provided all parties perceive themselves as better off than they were, the outcome is a potential win-win situation (both parties gain).

There are at least five ways to achieve a win-win result:

1) subjective values on a trade-off result in both parties perceiving they have gained;
2) one party giving up something with negative or no value (e.g. a rubbish dump), that is valued by the other party;
3) sharing information that adds value;
4) time sharing; and
5) both agree to the same usage/option

An example of a valuation win-win scenario is where a person values food more than the money a shop wants for the food. By exchanging money for the food, the person can get something more valuable to that person than the money, namely something to eat. By exchanging the food the shop-keeper gets something more valuable, namely more money than it cost to buy the food from the wholesaler. The exchange has left both parties better off, so they have both gained from the exchange.

If one party gives a zero or negative value to something, then that party would not be worse off by letting someone else have it; they would in fact be better off if they gave an item a negative value, i.e. considered it rubbish.

Information itself can have value: information that makes a manufacturer more efficient can result in greater returns, so the manufacturer would still be better off giving up some of the extra profit from the greater efficiency, provided the total net profit is larger than it was without that information. Alternatively, spreading useful information could indirectly benefit a person without a material cost; for example spreading information about cancer genes may lead to a faster cure for cancer, which could save the life of the person who spread the information.

As there is a limited number of things people can do, and places they can be at the same time, they won't necessarily be able to use things they value all the time. This means time sharing can potentially be used as a tool for negotiation.

The easiest win-win situation is where everyone agrees that the same thing should be implemented. For example, if everyone agrees no one should be allowed to kill another person, then it is a win-win situation [Raiffa, 1982].

It is not always clear how to achieve a Win-Win situation. However, such situations might be resolved by introducing new options such as compensation, or possibly by reframing the problem to see what changes in consequences result from a different assignment.

Negotiators generally want to be better off, though of course humans do not always behave rationally, and may be prepared to suffer a loss in order to impose a loss on another, as a form of retribution, or to push for a better final outcome. The fundamental question is: how much is a party prepared to give up to make gains, and is it enough to satisfy the parties that a fair outcome has been achieved.

Some issues have mutually exclusive solutions i.e. only one party can gain from the solution, for example the allotment of particular property that can not be shared. Mutual exclusivity is a negotiation epistemology matter outside this thesis. Subjective values for mutual exclusivity relative to shared solutions, and the implication of differences upstream for each that may vary depending on who gains, will be considered in subsequent work.

4.4 From Negaid Plugin to NeGame Shell

A Negaid eGanges application is used to guide negotiation to ensure no points of discussion are overlooked. The watershed nodes (nodes with no upstreams) of the application can also form a list of negotiation points (the issues) that can be negotiated via the negaid plugin algorithm. If the negotiating parties call on Negaid, then the Negaid list of watershed nodes appears in a survey style form with nine possible valuations for each watershed item: 00 = not negotiable; 0 = resolved; 1 not at all valued; 2 = very low; 3 = low; 4 = neutral; 5 = high; 6 = very high; 7 = absolute must. Both parties provide their respective weightings for every issue, and once this is done, Negaid suggests tradeoffs, which the parties can negotiate.

Negaid first looks for items that have the most difference in valuation. For example, if party A values item X at 1 and item Y at 6, while party B values item X at 6 and item Y at 1, Negaid will suggest giving party A item X and giving party B item Y; if both parties agree to this, item X and item Y are both revalued to 0 as the suggested tradeoff has been agreed. This Negaid algorithm is almost identical to the Family Winner algorithm [Bellucci and Zeleznikow, 2006]. See section four of the paper for further details.

4.4.1 Advantages of NeGame River hierarchy

Like eGanges, NeGame uses a River style graphical representation. However, whereas eGanges uses the analogy of streams flowing down and joining together to enter the sea at a single River mouth (the Final Result node), NeGame uses the analogy of a River delta, where subjective valuations flow down from the source (the all encompassing issue node) towards the numerous River branches.

This representation provides a way to group relevant issues into a single River branch, thus enabling relative subjective weightings to be given by each party, to each issue on the same branch. Then, each issue can have its immediately relevant sub-issues attached as a sub-branch River; each issue becomes the source of the River of its sub-issues, so that subjective valuation flows from the encompassing issue node towards its sub-issue nodes. Thus a naturally hierarchical structure is formed for users' subjective weightings of issues and thereby propagated down this River delta. The advantages of this hierarchically ordering of issues in a River structure are four fold:

1) Valuing all the detailed issues (issue nodes with no sub-branches, i.e. delta-mouth nodes) can be postponed by first only valuing the less detailed encompassing nodes that represent a larger collection of issues; this breaks the complex negotiation issues into simpler sub-issues that can initially be redressed, so minimising the chance of a negotiation becoming bogged down due to its complexity. Regardless of how many detailed issues have been given valuations, value normalisation at some level of the River system hierarchy will be possible. Thus as many conflict points as possible can be resolved despite negotiations being bogged down by some issues that may take much longer to resolve.

2) As the valuations propagate 'downstream' to delta mouths, it is possible for all sub-issues of an issue to be normalised and dealt with, without necessarily affecting the valuations of other issues that are not in the same branch of the River system. This should help with the development of alternatives, as the current delta-mouth nodes might be replaced with agreed further possible sub-branch issues, without disrupting agreements in other branches and the relationship of trust built up between the parties in these previous negotiations;

3) The grouping of issues into more detailed issue organisation via sub-branches will help clarify interests, options in the issues, and agreed packages of options;

4) The use of a quality control representation of issues (a KR similar to the Ishikawa fishbone) should provide legitimacy and foster commitment of the negotiators, as well as provide a means of complex communication and agreement to improve the relationship between the

negotiators.

In short, the use of NeGame should help with all seven aspects of principled negotiation. It monitors progress of a negotiation and the cumulative production of agreement with current assessment of the gains of each party, is available at any point in the negotiation.

4.4.2 Normalisation

The mathematical technique of normalisation is adapted in NeGame to ensure fair assessment of relative gains of the parties, given their respective subjective values. Parties are free to give whatever subjective values they choose. For example, in Figure 1 of paper 4.3, the node 'Human rights' is one of 18 antecedent nodes on the main stream. Each antecedent node may be subjectively valued differently to the others, and differently by each party. Once every node on the primary stream is given a subjective value by both parties, then the sum of the subjective values of a party is used to mathematically divide that party's value of every node on the primary steam. The division ensures that the new value sum equals 1 for each party. This normalised valuation is propagated downstream of encompassing nodes, and is multiplied into the downstream weighting factor of a River branch. This process ensures that the sum of every delta-mouth node's final true valuation is 1 for each party; thus a true comparison of the valuation of the parties for each delta-mouth issue is facilitated, ensuring optimal distribution of true gains.

4.4.3 Analysis of NeGame six-valued logic

As explained in Chapter Three, AnsProlog can either directly handle multi-valued logic by using a multi-valued logic semantic such as the well-founded semantic explained in Chapter Three, or it can indirectly handle three-valued logic by providing a third logic value which is captured when neither p, nor ¬p

are included in an answer set (which indicates that the value of proposition p is unknown). The NeGame epistemology uses a six-valued logic, for two parties, to resolve a dispute. The six epistemic logic values cover all possibilities and are shown in Table 1 below. NeGame gives a choice of five answers; both parties must agree on their single joint answer, which includes the unresolved option. There is a sixth default value representing no answer (incomplete). As can be seen in Table 1, there are a finite number of possible states of an issue, and these states range over more than a single dimension of concern; there is the dimension of whether Party A gains, and a separate dimension of whether Party B gains, etc.

Compacting all the possible states into a one dimensional value assignment eases processing both by the computer (with the use of appropriate multi-valued 'truth tables') and potentially by the user.

Table 4.1: NeG possible value states

	Party A gains, Party B gains	Party A gains, Party B loses	Party A loses, Party B gains	Party A loses, Party B loses
Both parties agree, Discussion complete	Win/Win	Party A gain (Zero-sum)	Party B gain (Zero-sum)	Lose/Lose
No agreement, Discussion complete	Unresolved	Unresolved	Unresolved	Unresolved
Discussion not yet complete	Incomplete	Incomplete	Incomplete	Incomplete

As detailed in the third paper of this chapter, the new six-valued logic introduced for NeGame helps more accurately encapsulate the knowledge that needs to be handled and processed in negotiations. The insights from this multi-valued logic also help to point out five ways that a true win-win scenario can be achieved, as discussed in the “Compromises and Trade-offs” section above. The new combinatorics issues from this new negotiation epistemology are also introduced, but will need further development in future works.

The new epistemology has been applied to an application designed to help with the Israeli-Palestinian conflict. However, this application is not completed, as it raises further issues of research into how the new epistemology should deal with potential disjunctive scenarios. Dealing with these variable scenarios is left as future work.

4.5 Papers

The papers show the advance of a new epistemology from a simple plugin of an adversarial superexpertise shell, eGanges, through to a completely reworked epistemology specifically suited to negotiation superexpertise. Together the eGanges adversarial and NeGame negotiation epistemologies should provide superexpertise aids in managing a range of legal problems. The underlying similarities in these two epistemologies may also assist the use of NeGame applications to guide the creation of eGanges applications of enforceable solutions agreed to by the negotiating parties. Thus a unified global civilisation might be developed in a

manner consistent with the Harvard Model of Negotiation.
The three papers are presented next, each following an overview and contribution summary.

4.5.1 Paper 4.1

The first paper of Chapter 4 is:
Gray, P. N., Gray, X. and Zeleznikow, J. (2007): "Negotiating Logic: For richer or poorer", in R. Winkels (ed), Proceedings of the 11th International Conference on Artificial Intelligence and Law, 4-8 June 2007, Stanford University, pp.247-251, ACM press, New York.
This paper introduces the Negaid plugin, which extends the eGanges River logic to enable it to better handle negotiations. The design of the plugin also extended the adversarial epistemology of eGanges with a limited negotiation epistemology. The paper conducts an explorative study of the usefulness of the Negaid plugin by investigating a cohabitation contract negotiation application of eGanges. The paper shows the use of eGanges to capture the knowledge of cohabitation contract negotiation, as well as how the knowledge is processed both by the eGanges core and the Negaid plugin component. The Negaid algorithm is based on the algorithm used in Family Winner [Belluci and Zeleznikow, 2006] and the main goal of the eGanges cohabitation contract application is to prevent the overlooking of relevant possibilities and potentialities (see Chapter Two). For negotiations that involve large scale knowledge and its management, eGanges, with the Negaid plugin, assists by ensuring no issue is left unconsidered or overlooked. The Negaid plugin ordering of subjective values of each party in respect of possible issues, also helps speed up resolution of issues by advising on likely agreed trade-offs.

Paper 4.1 Contributions

The candidate introduced the idea of a Negaid plugin, along with most of its design features, to help eGanges deal not just with quality control and adversarial problems, but also with negotiation problems. The potential usefulness of eGanges as a means of ordering the list of issues for negotiating a settlement was also determined. The numerical contribution breakdown is as follows:

	Pamela Gray	**Xenogene Gray**	John Zeleznikow
Concept	25.00%	65.00%	10.00%
Design	30.00%	65.00%	5.00%
Analysis	40.00%	50.00%	10.00%
Writing	55.00%	40.00%	5.00%
Average	37.50%	55.00%	7.50%

Due to copyright restrictions pages 109-113 have been omitted from this thesis. Please refer to the following citation for details of the article contained in these pages.

Gray, P. N., Gray, X. and Zeleznikow, J. (2007): “Negotiating Logic: For richer or poorer”, in R. Winkels (ed), Proceedings of the 11th International Conference on Artificial Intelligence and Law, 4-8 June 2007, Stanford University, pp.247-251, ACM press, New York.

http://doi.org/10.1145/1276318.1276366

4.5.2 Paper 4.2

The second paper of Chapter 4 is:

Gray, P. N., Gray, X. and Zeleznikow, J. (2009): “Intelligent Negotiation Technology”, in Legal and Negotiation Decision Support Systems (LDSS 2009): A Post-Conference Workshop at the 12th International Conference on Artificial Intelligence and Law, 4, pp. 38-54.

This paper is much longer and more detailed than the first paper of Chapter Four. It extends the work done in the first paper, with a focus on using negotiation tools, such as eGanges with the Negaid plugin, as a means of foreseeing possible sources of conflict and trying to prevent them from occurring, or at least having the parties pre-agree on how to resolve such conflicts. The paper indicates how the knowledge captured with eGanges can be used to achieve this foresight and prevent possible conflict.

The superexpertise ability of eGanges ensures every captured possible problem is discussed, and helps the parties achieve a higher degree of foresight in dealing with possible future problems. It does this by ensuring the topics discussed in the negotiation are as complete as possible; doing this before an actual conflict situation arises helps ensure such problems can be dealt with, and ideally avoided, when there is a minimal amount of emotional clouding of the judgement and actions of the parties. Thus, potential problems can have rules of resolution agreed to before problems arise, and these rules of resolution also turn future problems into adversarial, not negotiation, problems. This can be of great assistance both for domestic unions (via a cohabitation contract), and business interactions (contractual transactions).

Two eGanges applications are the basis of an explorative study of conflict avoidance in this paper. The applications are a minimax contractual transaction strategy application as well as the cohabitation application of paper 4.1. The usefulness of eGanges and the Negaid plugin for these two applications form an inductive basis for their use to help both business and domestic relationships.

Paper 4.2 Contributions

The numerical contribution breakdown is as follows:

	Pamela Gray	**Xenogene Gray**	John Zeleznikow
Concept	25.00%	60.00%	5.00%
Design	25.00%	65.00%	10.00%
Analysis	35.00%	55.00%	10.00%
Writing	55.00%	40.00%	5.00%
Average	37.50%	55.00%	7.50%

Due to copyright restrictions pages 115-130 have been omitted from this thesis. Please refer to the following citation for details of the article contained in these pages.

> Gray, P. N., Gray, X. and Zeleznikow, J. (2009): “Intelligent Negotiation Technology”, in Legal and Negotiation Decision Support Systems (LDSS 2009): A Post-Conference Workshop at the 12th International Conference on Artificial Intelligence and Law, 4, pp. 38-54.

4.5.3 Paper 4.3

The third paper of Chapter 4 is:

Gray, X., Gray, P. N. and Zeleznikow, J. (2011): "Supporting the Harvard Model of Principled Negotiation with Superexpertise", in Proccedings of 4th Workshop on Legal Informatics and Legal Information Technology: BIS 2011 Workshops, Lecture Notes in Business Information Processing (LNBIP) 97, W. Abramowicz, L. Maciaszek and K. Wcel (Eds.), pp. 301-312, Springer-Verlag, Berlin, 2011.

The final paper of the chapter focuses on analysing what is epistemically required for negotiation, building on the work of the Harvard Negotiation Project, and the suggestion for the use of reframing; it takes the evolution of the Negaid plugin even further to produce the new foundational epistemology for negotiation that is used in a new application shell, Negotiation Game, NeGame, or NeG for short. NeGame exploits the hierarchical nature of the River knowledge representation to abstract away from potentially difficult sub-issues and enable resolution of general encompassing issues before finalising these more complex sub-issues (while still ensuring that some level of weighting prioritisations of these complex issues can be taken into account).

The paper commences by discussing what is a legal epistemology, and the two types of epistemologies required by the legal domain: adversarial and negotiation epistemologies. The use of Ishikawa fishbone knowledge representations for both eGanges and NeGame is also discussed; in eGanges, logic flows downstream and joins up like water in a River tributary system, whereas in NeGame, subjective values flow downstream and branch out into various delta-mouths. In both cases the River structure provides a graphical representation of the hierarchical structure of the knowledge.

The main problem addressed in this paper is how to provide for negotiating parties, throughout a negotiation, knowledge of a quality control evaluation, not just of known possible alternatives in a negotiation, but also of newly realised options devised during negotiation, particularly those options that can be determined more readily thanks to the hierarchical structure of the issues used in NeGame. These options can also be used to help with reframing the various issues in dispute. This complete reworking of the underlying eGanges epistemology to more accurately reflect what is required in negotiation led to the design of NeGame by the candidate. An application was begun as a case study of civilisation negotiation, which is explained in paper 4.3, to reframe for communication the Israel-Palestine conflict.

Paper 4.3 Contributions

Developing the NeGame epistemology was almost entirely the work of the candidate; this significantly extended the negotiation superexpertise beyond the Negaid plugin. The NeGame epistemology is seminal work by the candidate that identified and developed a knowledge structure and processing specifically appropriate to negotiation. The civilisation application was initiated by legal expert, Dr Pamela N. Gray, but reworked by the candidate for the altered NeGame epistemology. The numerical contribution breakdown is as follows:

	Xenogene Gray	Pamela Gray	John Zeleznikow
Concept	75.00%	20.00%	5.00%
Design	90.00%	5.00%	5.00%
Analysis	90.00%	5.00%	5.00%
Writing	75.00%	20.00%	5.00%
Average	82.50%	12.50%	5.00%

Due to copyright restrictions pages 133-144 have been omitted from this thesis. Please refer to the following citation for details of the article contained in these pages.

Gray, X., Gray, P. N. and Zeleznikow, J. (2011): “Supporting the Harvard Model of Principled Negotiation with Superexpertise”, in Proccedings of 4th Workshop on Legal Informatics and Legal Information Technology: BIS 2011 Workshops, Lecture Notes in Business Information Processing (LNBIP) 97, W. Abramowicz, L. Maciaszek and K. Wcel (Eds.), pp. 301-312, Springer-Verlag, Berlin, 2011.

http://doi.org/10.1007/978-3-642-25370-6_29

Chapter 5: Conclusion

The motivation for the work of this thesis has been to assist people and businesses to deal with the super human levels of rapidly changing and expanding expert knowledge that increasingly they are required to master i.e. to help them implement superexpertise. To do this requires methods to be developed that can overcome the Knowledge Acquisition bottleneck, along with the other requirements identified in Chapter One.

Each type of problem faced may require a different approach to achieve superexpertise. The notion of epistemologies is used in this thesis to represent the collection of the knowledge representation, semantics, heuristics, etc of such approaches; all these aspects of an appropriate superexpertise approach collectively constitute an epistemology. Those specifically identified in this thesis amount to a model of superexpertise.

In Chapter One, the thesis set out three broad categories of requirements that an epistemology must satisfy in order to provide superexpertise. These three categories are the requirements of Knowledge Representation, Knowledge Based System and Superexpertise. With the help of the exploratory studies of the papers of Chapter Two, eGanges has been shown to meet all these requirements for Quality Control and Adversarial type problems. Thus, eGanges was shown to provide superexpertise for these two problem types.

This thesis has looked at three types of problems, namely Quality Control (QC) problems, Adversarial problems, and Negotiation problems. These three types of problems can all be classified as problems of Finite Domain Expertise, as outlined in Chapter One. The foundation Quality Control problem, Ishikawa's quality control fishbone, was dealt with in the eGanges application given in Figure 4 of paper 2.4. Several more Quality Control and Adversarial examples, in particular blackletter law eGanges applications, have been given throughout the papers of Chapter Two. Indeed, the range of QC and Adversarial problems satisfactorily addressed by the eGanges River Logic epistemology, as shown in these papers, provides an exploratory study that shows the extensive applicability of the eGanges River Logic epistemology to these two types of Finite Domain Expertise problems.

However, one of the research questions at the heart of this thesis was: does the eGanges River logic approach work as an epistemology for agile management and dispute resolution, and if not, can this epistemology be adapted to become workable? The conclusion is that eGanges has been shown to be an appropriate epistemology for agile management, as agile management is concerned with superexpertise handling of Quality Control problems, and also adversarial dispute resolution. The thesis recognised that an adapted epistemology was required for problems of Negotiation for dispute resolution, leading to the development of the NeGame epistemology.

5.1 Addressing the Research Question

This thesis introduced superexpertise for investigation and modeling. The focus of the research was on two epistemologies designed to provide superexpertise; the eGanges epistemology for Quality Control and Adversarial problems that is covered in Chapter Two, and the newly developed NeGame epistemology for negotiation problems covered in Chapter Four.

To explore these epistemologies, an investigation framework was required; the framework used was the superexpertise framework detailed in Chapter One. Superexpertise details what is required from an epistemology for it to be workable as a solution to a particular type of complex problem. Several exploratory studies of complex Adversarial and Quality Control problems were performed in the papers of Chapter Two, and several exploratory studies of complex Negotiation problems were carried out in Chapter Four.

Moreover, the thesis illustrates that the eGanges River logic epistemology is an elegant, easy to understand way of dealing with incomplete and inconsistent information. It uses four epistemic truth values: unanswered, uncertain, positive (truth value of true, with epistemic aspects included) and negative (truth value of false, with epistemic aspects included). Unanswered means answering a dichotomy has not begun, or can not begin. Uncertain means only incomplete/inconsistent information is available to answer a dichotomy. Chapter Two has shown that the law and business quality assurance can be appropriately modelled by this epistemic four-valued logic.

The use of such multi-valued logics as the basis of knowledge representations for problems that have only two ontological truths has been criticised [Dubois, 2008]. However, the basis of these criticisms have been redressed in the work of Chapter Three, which investigated whether inconsistency can be redefined in a more semantically meaningful way, and what consequences are derived from this redefinition. It also showed that the innate tautology loss criticisms of multi-valued logic can not truly be avoided; they can only be reassigned to alternative functions in order to preserve the old tautologies (see section 3.5.3 of Chapter Three). Introducing additional functions to preserve old tautologies, or using redefined tautologies that more accurately capture their meaning, both ensure compositionality of problems. Indeed Chapter Three showed that eGanges River logic is not only compositional, but also decidable and consistent due to the three core heuristics used by eGanges.

The following subsections draw together and summarise the contributions and conclusions made in the previous chapters including:

- the exploratory studies (section 5.1.1);
- advantages of eGanges for handling superexpertise combinatorics (section 5.1.2);
- the way in which NeGame allows foresight to be applied to negotiation (section 5.1.3);
- the verification of four-valued eGanges logic (section 5.1.4);
- the use of an expert's epistemology to provide validation of eGanges and NeGame (section 5.1.5); and
- the applicability of River logic to any field of Finite Domain Expertise (section 5.1.6).

5.1.1 Exploratory Studies

The papers of Chapter Two detail;

- an anti-spam legislation application
- a partnership legislation application
- a requirements for promotion to senior lectureship application
- an Anti-Money Laundering and Counter-Terrorism Financing legislation application
- a Corporate CEO duties application
- an insulation quality control application

These exploratory case studies form a qualitative inductive base for justifying the use the eGanges epistemology for a wide variety of areas of Finite Domain Expertise; this helps support the proposition that the eGanges epistemology is appropriate for certain problems. Each of Chapter Two's papers develops the idea that the expert system shell, eGanges, is an agile management and quality control tool that can augment human experts by thoroughly automating the massive combinatorics required of fields that need this superexpertise due to the large and complex knowledge of those fields. The papers detail how the complexity of the knowledge can be broken down by the hierarchical River logic knowledge representation, ensuring that any area of expertise that is needed to answer a single yes/no answerable question (ie a dichotomy) can use this knowledge representation (epistemology) to make the job of the expert far more precise but easier. Indeed, the resulting expert systems can be used to train experts in the field and be used more as a mnemonic aid, or as cues, to ensure all relevant questions and issues are dealt with by them.

The exploratory studies of Chapter Two indicate that the epistemology of eGanges is expert-friendly for the domains of law, quality control management, and education. Further, eGanges is a refinement of the Ishikawa quality control fishbones that were so successful after the second world war in producing high quality Japanese manufacturing.

Chapter Four similarly provides a set of exploratory case studies for the NeGame epistemology: negotiation of cohabitation contracts, using an eGanges plug-in, Negaid, to assist, and negotiation of a civilisation using the NeGame shell.

The applications illustrate that the number of River nodes and the number of logic choices for each node are finite and determinable. For NeG, due to normalisation and appropriate rounding, the subjective valuations are also effectively finite and determinable. With determinable numbers, superexpertise can provide measures of complexity for choice, relative freedom, planning, goal attainment, agile management, operations, policy, negotiation, progression of agreement, conflict and social relationships. It is possible to express formulae for the calculation of these complex combinatorics but this is outside the scope of this thesis; the mathematics of superexpertise may underpin new sciences of social complexity and legal choice.

5.1.2 Advantages for Superexpertise Combinatorics

The eGanges superexpert system shell addresses the combinatoric issues of expertise by guiding domain experts to articulate and organise their knowledge via a user interface which visualises and constrains the knowledge entered. The shell facilitates the implicit capture of the positive, negative and uncertain knowledge elements within the domain, and their interaction. Like all real world systems, eGanges is restricted to handling knowledge in a finite domain, such as the Spam Act and other fields of legal compliance and conflict prevention; this is due to the finite limit to the capacities of real systems. However, an application of finite expertise may be as large scale as the knowledge requires. Where the finite boundaries open to further expansion into potentialities, and the accommodation of projections, this knowledge may be glossed on the finite number of nodes for informed decision-making.

Due to the small program size and flexibility of the software, eGanges can operate on a PDA. As such, eGanges can bring Quality Control tools to any time and place, so time is never wasted and opportunities are never lost; this makes quality control logic a source of agility and mobile learning. Nor is time wasted on difficult language and complex logic; shared super-agile aids are also a communication system [cf. Owusu, 1999], requiring minimalistic translation into other languages, as a common basis for negotiation of complex understanding and the extensive resulting combinatorics of that understanding. The graphical interface of eGanges shows that a picture can truly say a thousand words. This saves time and expedites common understanding.

No written coding is required for construction of an eGanges application. The interface is minimalistic, according to the logic structure of the epistemology. It suggests an androgogy in vocational coursework that also permits super-agile job performance in the workplace. The minimalistic language of the eGanges River maps, and communication system, allows for quick translation to foreign languages. Logic graphics that effectively communicate large scale, complex choice, and enable large scale, complex informed agreement or disagreement, introduce a further evolution in human communication and intelligence and make large scale, complex agreements achievable. People could become proficient in managing human relationships in this way; a major leap toward a universal scientific civilisation could result.

The code language of machines and human language is reconciled in eGanges through a visualisation language. While we understand this graphical language, we can enlist machines to process the results. If we use these machine aids, human intelligence is further extended by artificial intelligence.

e-learning in the workplace, with eGanges application aids, is likely to increase available human intelligence. Its use might help people move more fully into the age of science, with greater understanding and safety. Such intelligence aids provide people, who do not have personal accurate knowledge of a complex field, access to the knowledge and reasoning necessary for survival in a world with a more demanding level of higher education and knowledge. The Ishikawa fishbone captured the limited logic of achieving a single objective in predetermined ways that permit the setting of goals to collectively achieve the objective, and targets that collectively achieve each goal. eGanges also calculates risks as possible failures and uncertainties with precise combinatorics beyond the imprecise limits of Ishikawa fishbones. This may enable businesses and government to manage the increasingly complex and uncertain combinatorics they must face.

5.1.3 Foresight Applied To Negotiation

Current research on negotiation systems has focused upon resolving disputes once they have occurred. However, it may be easier to avoid disputes, than satisfactorily resolve them. The papers of Chapter Four started by focusing on designing improved negotiation support processes. On this basis, further measures could be developed for legal fairness in interest based negotiation support systems in family mediation, plea bargaining and housing disputes. The need for intelligent negotiation planning to avoid, rather than resolve, disputes has been discussed in Chapter Four. An eGanges application was used to demonstrate how development of cohabitation agreements might avoid conflicts before, and after, the breakdown of relationships. This approach seeks to convert negotiation problems to adversarial problems by getting parties to acknowledge adversarial rules before problems occur.

The papers of Chapter Four were focused on how to aid negotiations. For example, legal choices, mapped in an eGanges application, may assist negotiation. Alternate pathways through legal possibilities to potentialities are often matters to negotiate. This raises many new design considerations distinct from alternate case pathways through a system of rules. The provision of negotiation support, through heuristics external to the expert system, is a useful technique for selecting conflict prevention measures that extend the resources of conflict resolution from which law is derived and limited.

The exploration of the issues associated with extending the eGanges River logic into fields that require negotiation between parties, eventually led to the development of a completely new epistemology, called NeGame or NeG. The thesis investigated whether the hierarchical nature of River logic could be used to help guide negotiations, and what extensions would be required to help prevent conflicts.

The user-friendliness of eGanges is maintained in the design of NeG, to allow quick construction and alteration of a River system and its glosses. This permits the NeG applications that are to be negotiated, to be fully expressed and particularised, in an ongoing way, as suggested by the Harvard model.

NeG is designed to monitor progress of a negotiation and the cumulative production of agreement, or specific lack of it, with current assessment of the gains of each party, which is available at any point in the negotiation. The parties may record the substance of their negotiations in the Notes window below the Question window. In NeG, the subjective value input of both parties is also available as gloss information, and summarised in the Current result window.

The superexpert model of negotiation posed in the last paper of Chapter Four employs computing capability to support the Harvard model of Principled Negotiation: extended memory, faster retrieval, specification of factors that determine possibilities, and faster processing of consequent combinatorics, multi-valued logic, subjective differences that maximise alternative Win-Win options, and monitoring cumulative agreement and changes.

The use of NeG as a technological aid, is illustrated by the Civilisation application which falls into the Harvard model of Principled Negotiation as communication. The Civilisation River allows the introduction of new options through its provision for further issues that extend the hierarchical tributary structure. NeG provides only for two party conflicts, but the design could be extended for multi-party negotiations.

NeG is designed consistently with common law epistemology and Bologna glossing to assist formulation of an enforceable resolution of the conflict; it enhances all seven aspects of Principled Negotiation. The use of a quality control representation of issues should reinforce legitimacy and foster commitment of the negotiators; as a means of complex communication and processing, it should improve the common understanding and relationship of the negotiators. The communication system of NeG assists the management of clusters of issues in a broader hierarchy as a framework for identifying delta-mouth issues where details terminate, and for expanding options and details at any level of the hierarchy; this assists deconstruction of complex conflicts and may reveal ways to find Win-Win solutions.

The NeG interface obtains instructions according to its six-value sorting requirements, and receives input on the subjective values of the parties which reflect their relative interests. Mathematical normalisation of subjective values, and the propagation of those values lends legitimacy by providing an objective metric of fairness for proposed negotiation solutions. NeG processing of each answer input simulates sorting according to its six-value logic and thereby monitors cumulative agreement in the negotiation with combinatoric processing of subjective values. Feedback throughout NeG's questioning of the negotiators appears in windows that monitor cumulative relative gains of the parties and unresolved issues. This is a basis for fostering a trust relationship between negotiators and their commitment.

5.1.4 Verification of Four-Valued eGanges Logic

Chapter Three laid the foundations for the logical verification of the eGanges epistemology. Verification has traditionally been broken up into basically two categories: consistency and completeness [Gonzalez and Dankel 1993], and Chapter Three outlined how eGanges is not only consistent, but decidable, as well as complete with respect to its domain of discourse. Salle and Hunter [1990] consider that most KBS act as a prosthesis rather than support. Due to the subjective and changing nature of many domains such as medicine, they consider the prosthesis approach to be unsuitable. Users need more than prescriptive systems, they need to assess the answer through better explanation and query facilities. This is an underlying principle of the eGanges and NeG user interface designs, which recognise that users of experts systems do not want to relinquish decision making to a machine [Richards, 2000; Ignizio, 1991; Langlotz and Shortliffe, 1983].

Once the method of knowledge representation is appropriately verified, the expert knowledge of the system needs to be captured within the system, and that knowledge needs to be validated by a field expert. Traditionally these two aspects have both involved a knowledge engineer and are often treated almost interchangeably in the literature. For eGanges and NeGame, the knowledge engineer is made redundant as experts can directly enter and validate their knowledge through an already verified epistemology.

5.1.5 Using an expert's epistemology as a specification

Treating the epistemology of a field, as determined by a field expert, as a specification, means the ideals proposed by Boehm [1984] of verification ("Am I building the product right?"), validation ("Am I building the right product?") and ensuring the expert's specifications are complete and consistent, can be fulfilled for the River logics of this thesis. eGanges River logic has been developed by an experienced legal practitioner, as part of her Ph.D thesis, to be an appropriate knowledge representation, etc, for modelling legal expertise. While the NeGame epistemology was developed by the candidate, it has been preliminarily validated by a negotiation expert, John Zeleznikow.

5.1.6 Applicability of River logic

eGanges River logic is used to answer a simple (yes/no) dichotomy question, such as "Have I complied with the Spam Act?" Due to the binary nature of the question, after all legal processes have been completed, a court will find that either there has or has not been compliance; so there is no liability or some liability. There are only ever two final ontological results; the dichotomy will be resolved one way or the other. However, River logic requires the use of four logic values: Positive, Unanswered, Uncertain, Negative. These four epistemic truths lead to only a binary ontological result, but contain more information than the simple binary ontological results; they assist management of a case by a practitioner prior to adjudication, which may require advice on the likely success of litigation, preparation of evidence, and negotiation of a settlement of the dispute.

Due to the burden of proof rules, a Final result of Uncertain may result in an ontological finding of compliance with the Spam Act. Legally, this is epistemically not the same as obtaining a result of Positive, where there is unquestionably a finding of compliance with the Spam Act. However, from an ontological point of view the same result has occurred.

The reason this epistemic difference is important is because in certain cases, when more evidence is found, the uncertainty may be resolved, and the case might be brought to the courts again, so that a different ontological result may be determined.

Similarly, the epistemic value of unanswered represents that not enough evidence has been gathered for a not unanswered result to be determined, and therefore this round in the courts might not yet be conclusive. This multi-epistemic value approach is used by the eGanges epistemology to deal with the nonmonotonic aspects of reasoning with incomplete information. It also has applicability beyond the legal framework, as any field of Finite Domain Expertise may be able to be represented by River logic.

5.2 Limitations and Future Considerations

There is much work still to be done on further developing the NeGame epistemology; aspects of negotiation disjunctions caused by situational differences caused by specific selected choices might be more fully addressed. For example, if a man is selected as the only doctor in a town, women may not feel comfortable seeing him for a consultation, particularly in Muslim countries. Alternatively, a female doctor presents other potential problems, such as misogynistic attitudes from patients. These disjunctive and other possible limitations, such as NeG's two party limit, might be further considered for future research into negotiation epistemology.

While the eGanges epistemology ensures Rivers flow down to a single Final Result node, more research should be done on possible node repetition among various Rivers. This potential limitation of the eGanges epistemology was considered before the thesis began, and a simple gloss linking method is provided so that node values can be easily compared manually. Direct linking of node values should not cause infinite processing loops, but negation, or any other mapping, will break the acyclic nature of the eGanges epistemology, thus potentially causing infinite loops. These matters require further consideration.

Further work on the potentialities in combinatorics might include explorations of theorem provers, and other methods of finding possible gaps in rule systems. It is possible that a rule base may have many implicit rules, which may not be appropriately used due to the fact they are not explicitly known. For example, if a rule base has the rules ($A \rightarrow B$) and (A), then it implicitly has the rule (B); however, until *modus ponens* is applied to these two rules, the rule base does not explicitly contain (B). Again, the hierarchical structure and single Final Result node should minimise the chance that such derivable rules are left out; upstream rivers have to be added to a River system either by a human expert, or by rule extracting software that processes the legislation text directly. Work on such rule extracting software is also something that could be considered for future work; though ideally for the law, if law makers directly made River systems, such rule extracting software, and its accuracy, would be unnecessary. Courts of law may extend a River system through new cases that are not already included in the law. The simplicity of the eGanges user-interface should allow experts to directly enter their knowledge, without a knowledge engineering intermediary. Investigations into the actual effectiveness of this simple user-interface could also be conducted. Similarly, other usability studies of these epistemology-based shells could be conducted.

The investigation of combinatorics is limited to fields of Finite Domain Expertise (FDE) because of Godel's theorem; if the knowledge is not finite it cannot simultaneously be consistent and complete. This thesis looked at examples of FDE, in particular: Quality Control, Adversarial and Negotiation problems. These examples were covered in eGanges and NeGame applications which all confirmed the epistemology of the a river knowledge structure. Rivers analogues appear to be suited to large scale FDE as they may be as extensive as the FDE requires, while providing an organisational framework for the knowledge that is comprehensible to humans. Further studies on these aspects matters could also be undertaken.

The FDE and superexpertise framework mightbe extended to determine appropriate epistemologies for other types of FDE problems. Such investigations may lead to new epistemologies that are more appropriate for these different types of problems in order to fulfil the requirements of KR, KBS and superexpertise as outlined in Chapter One. This is a future consideration, as the conceptual tools and methods outlined in this thesis indicate an extensive applicability to remove the KA bottleneck for a range of problems.

Chapter Three covered the basics of the logical verification of the eGanges epistemology, but the issues raised, such as redefining classical tautologies to a multi-valued generalisation may require further consideration.

5.3 Significance of Thesis Work

The contributions to the field made by this thesis are that it:

1) constructs a useful classification of types of problems, in particular fields of FDE;
2) constructs a useful model of superexpertise;
3) constructs a specific epistemology for negotiation superexpertise;
4) provides numerous case studies to demonstrate the applicability of two epistemologies to three types of problems;
5) establishes superexpertise as a landmark in AI following on from Expert

Systems/Knowledge Based Systems and Problem Solving Methods;

6) demonstrates deconstruction for clearing the Knowledge Acquisition bottleneck; and
7) accommodates expert epistemology in AI.

5.4 Final thoughts

Many Knowledge Representations are too complex for end users; software developers should be guided by Occam's razor and keep KR simple. Some in the field have recognised this fact [Bench-Capon, 1990], but have not necessarily followed this recommendation. This is probably because creating something elegantly simple is, ironically, often much harder than creating something of great complexity.

The two epistemologies of this thesis rely on simple real world river analogues, that people should be familiar with, to help overcome this complexity problem. Both these river analogues use simple river branches as the basis of a self-similar construction to encode complexity. By requiring end users to only deal with one river at a time, they should not get overwhelmed by the information of a large scale, complex river system. The innate hierarchical structure of both rivers naturally enables knowledge abstraction, encapsulation and hiding; this makes dealing with massive amounts of knowledge possible for ordinary humans.

The innate order, due to the hierarchical nature, of the river systems forces experts and rule makers to think more clearly about their knowledge. This precise specification requirement has caused some problems for knowledge acquisition, but in the long term, should help prevent the chaos that can occur from an *ad hoc* approach by rule makers and experts in general.

As most of the knowledge content of the epistemologies is captured by the structure of the river systems, translating applications into other human languages should be much more straight forward; only the node encapsulated simple questions, possible answers and glosses will need to be in a human language. This should hopefully encourage global trade and interactions as a country's legislation more readily can be made available to all language speakers. In an age of globalisation this is a useful facility.

A resulting clearer universal mapping of legislation, and increase in interactions between different countries might move towards a more international civilisation. Consensus by different countries about which country's legislation works best for each area might bring about a consolidation of global law that facilitates a common understanding for avoidance of war.

Bibliography

Aristotle (1952, originally c.335 BC): Organon, in R. M. Hutchins (ed.) *Great Books of the Western World*, Vol 8, Encyclopaedia Britannica Inc., Chicago, USA.

Bacon, F. (1952, originally 1620,): Novum Organum, *Great Books of the Western World*, Vol 30, Encyclopaedia Britannica Inc, Chicago, USA.

Baral, C. (2003): *Knowledge Representation, Reasoning and Declarative Problem Solving*, Cambridge University Press, Cambridge, England.

Baroff, J., Simon, R., Gilman F. and Schneiderman, B. (1988): Direct Manipulation User Interfaces for Expert Systems in J. A. Hendler (ed.), *Expert Systems: The User Interface*, Ablex Publishing Corp, New York.

Bellucci, E and Zeleznikow, J. (2006): Developing Negotiation Decision Support Systems that support mediators: a case study of the Family Winner system, *Journal of Artificial Intelligence and Law, 13(2): 233-271*.

Belnap, N. (1977): How a computer should think, in G. Ryle (ed.), *Contemporary aspects of philosophy*, pp. 30-55, Oriel Press, Stocksfield, Northumberland.

Bench-Capon, T.J.M. (1990): *Knowledge Representation: An Approach to Artificial Intelligence*, Academic Press, London.

Besnard, P. and Hunter, A. (1998): Introduction to actual and potential contradictions, in D. Gabbay and P. Smets (eds), *Handbook of Defeasible Reasoning and Uncertain Information*, volume 2, Kluwer Academic, Berlin.

Bibas, S. (2004): Plea bargaining outside the shadow of the trial, *Harvard Law Review, 117:2464-2547*.

Blum, G. (2007): *Islands of Agreement: Managing Enduring Armed Rivalries*, Harvard University Press, Cambridge MA.

Blyth, T. S. (2005): *Lattices and Ordered Algebraic Structures*, Springer, San Francisco.

Boehm, B. W. (1984): Verifying and Validating Software Requirements and Design Specifications, *IEEE Software, 1(1):75-88*.

Boeva, V., Tsiporkova, E. and De Baets, B. (1998): Modelling Uncertainty with Kripke's Semantics, in *Proceedings of AIMSA 1998*, pp.129-140.

Brachman, R. J. and Levesque, H. J. (2004): *Knowledge representation and reasoning*, Morgan Kaufmann, Palo Alto, CA.

Brams, S. J. and Taylor, A. D. (1996): *Fair Division, from cake cutting to dispute resolution,* Cambridge University Press, Cambridge, England.

Breuker, J. (1994): Components of Problem Solving and Types of Problems, in L. Steels, G. Schreiber and W. Van de Velde (eds), A Future for Knowledge Acquisition, Proceedings of the 8th European Knowledge Acquisition Workshop, EKAW'94, *Lecture Notes in Artificial Intelligence* 867, Springer Verlag, 118-136.

Breuker, J. and Van de Velde, W. E. (1994): *The Common KADS Library, Technical report*, University of Amsterdam and Free University of Brussels.

Chandrasekaran, B. (1986): Generic Tasks, in Knowledge-Based Reasoning: High Level Building Blocks for Expert System Design, *IEEE Expert Fall 1986, 23-30*.

Chandrasekaran, B. (1990): Design Problem Solving: A Task Analysis. *AI Magazine, Special Issue on Design*, J. S. Gero & M. L. Maher (eds), AAAI, 11(4):59-71.

Collingwood, R. G. (1940): *Essay on Metaphysics*, Clarendon press, Oxford, England.

Compton, P. and Jansen, R. (1988): Knowledge in context: a strategy for expert system maintenance, in *AI 1988: Proceedings of the Second Australian Conference in Artificial Intelligence*, Adelaide, pp. 283-297.

Condliffe, P. (2008): *Conflict Management: A Practical Guide*, 3rd ed, Lexis Nexis, Sydney.

Coste-Marquis, S. and Marquis, P. (2008): Recovering consistency by forgetting inconsistency, in *Proceedings of the 11th European Conference on Logics in Artificial Intelligence (JELIA'08)*, LNCS, 5293:113-125, Dresden. Germany.

Davis, M. (1958): *Computability and Unsolvability*, McGraw-Hill, New York.

Delgrande, J. and Schaub, T. (2003): On the relation between Reiter's default logic and its (major) variants, in *Proceedings of the Seventh European Conference on Symbolic and Quantitative Approaches to Reasoning with Uncertainty (ECSQARU 2003)*, pp. 452-463.

Dondossola, G. (1999): Formal Methods for the Engineering and Certification of Safety-critical Knowledge-based Systems, in *Proceedings of European Symposium on Validation and Verification of Knowledge Based Systems (EUROVAV 1999)*, pp. 113-129.

Dove, R. (1999): Knowledge management, response ability, and the agile enterprise, *Journal of Knowledge Management, 3(1):18 – 35*.

Dove, R., Nagel, R., Goldman, S. and Priess, K. (1991): *21st Century Manufacturing Enterprise Strategy - An Industry Led View*, Diane Publishing Co, Darby, PA.

Dubois, D. (2008): On Ignorance and Contradiction Considered as Truth-Values, Logic *Journal of the IGPL, 16(2):195-216.*

Dummett, M. (1973): *Frege: Philosophy of Language*, Harvard University Press, Boston.

Feigenbaum, E. A. (1981): Expert systems in the 1980s, in A.H. Bond (ed.), *Machine*

Intelligence, Infotech State of the Art Report Series 9, no.3, Pergamon Infotech, Maidenhead, Berkshire, England.

Buchanan, B.G., Sutherland, G. and Feigenbaum, E. A. (1969): Heuristic DENDRAL: a program for generating explanatory hypotheses in organic chemistry, in B. Meltzer and D. Michie (eds), *Machine Intelligence 4*, Edinburgh University Press, Edinburgh, Scotland.

Fensel, D., Richard, V., Enrico, B. and Bobwielinga, M. (1999): UPML: A Framework For Knowledge System Reuse, in *Proceedings of the International Joint Conference on AI (IJCAI-99*), pp.16-23, Morgan Kaufmann, Palo Alto, CA.

Fisher, R. and Ertel, D. (1995): *Getting Ready to Negotiate*, Penguin, New York.

Fisher, R. and Ury, W. (1981): *Getting to YES: Negotiating Agreement Without Giving In*, Haughton Mifflin, Boston.

Fisher, R., Ury, W., and Patton, B. (1991): *Getting to Yes*, Random Century, London.

Fitting, M. (1990): Bilattices in logic programming, in G. Epstein (ed), *Proceedings of the Twentieth International Symposium on Multiple-Valued Logic*, pp. 238--246. IEEE, 1990.

Fitting, M. (1991): Bilattices and the semantics of logic programming, *Journal of Logic Programming, 11:91-116*.

Fitting, M. (1992): Many-valued non-monotonic modal logics, in A. Nerode and M. Taitslin (eds), *Proceedings of 2nd Symposium on Logical Foundations of Computer Science*, Tver, Russia, Springer, Berlin, Germany.

Fitting, M. (2006): Bilattices are nice things, in T. Bolander, V. Hendricks, and S. A. Pedersen (eds) *Self-Reference*, pp. 53--77, Center for the Study of Language and Information, Palo Alto, CA.

Fraunce, A. (1969, originally published 1588): *The lawyer's logic*, Reproduced from the original manuscript in the British Museum, *Lawiers Logike*, first published by William How, London, The Scolar Press Limited, Menston, Yorkshire, England.

Galanter, M. (2004): The Vanishing Trial: An Examination of Trials and Related Matters in State and Federal Courts, *Journal of Empirical Legal Studies, 1(3): 459-570*.

Van Gelder, A., Ross, K. A. and Schlipf, J.S. (1991): The Well-Founded Semantics for General Logic Programs, *Journal of the ACM, 38(3):620-650*.

Gelfond, M. and Lifschitz, V. (1988): The stable model semantics for logic programming, in *Proceedings of the Fifth International Conference on Logic Programming (ICLP)*, pp. 1070-1080.

Gillies, D. (1996): *Artificial intelligence and scientific method*, Oxford University Press, Oxford, England.

Ginsberg, M. L. (1988): Multivalued logics: a uniform approach to reasoning in artificial intelligence, *Computational Intelligence, 4:265-316*.

Ginsberg, M. L. (1990): Bilattices and Modal Operators, *Journal of Logic Computation, 1(1):41-69.*

Ginsberg, M. L. (1992): Multivalued logics: A uniform approach to inference in artificial intelligence, in *Computational Intelligence, 4(3):256-316*.

Gonzalez, A. J. and Dankel, D. D. (1993): *The Engineering of Knowledge-Based Systems*, Prentice Hall, New York.

Gottlob, G. (1995): Translating default logic into standard autoepistemic logic, *Journal of the ACM, 42:711-740*.

Gray, P. N. (1973): A New Lease of Life. Precedent for a Cohabitation Contract, *New Law Journal*, vol 123, pp. 591, 596.

Gray, P. N. (1988): The CLIMS (Contract Law Information Management System) Pilot - Automatable Law, in *Proceedings of the 4th International Congress on Computers and Law*, Rome, Italy.

Gray, P. N. (1990): *Choice and Jurisprudential Systems*, Vols 1 and 2, LL.M. thesis, University of Sydney, Sydney, Australia.

Gray, P. N. (1997): *Artificial Legal Intelligence*, Dartmouth Publishing Co., Aldershot, England.

Gray, P. N. (2005): eGanges: Epistemology and Case Reasoning, in P.E. Dunne and T. Bench-Capon (eds), *Argumentation in Artificial Intelligence and Law*, Wolf Legal Publishers, Nijmegen, Netherlands.

Gray, P. N. (2007a): *Legal Knowledge Engineering Methodology for Large Scale Expert Systems*, Volumes 1 and 2, PhD thesis, University of Western Sydney (2007): http://trove.nla.gov.au/work/3901700 [accessed 27/6/11]

Gray, P. N. (2007b): Ontology of Legal Possibilities and Legal Potentialities in *Proceedings of the Workshop on Legal Ontologies and Artificial Intelligence Techniques, International Conference on Artificial Intelligence and Law*, Stanford University, ACM: http://ftp.informatik.rwth-aachen.de/Publications/CEUR-WS/Vol-321/paper1.pdf [accessed 27/6/11]

Gray, P. N. and Gray, X., (2003): A Map-Based Expert-Friendly Shell, in D. Bourcier (ed), *Legal Knowledge and Information Systems*, IOS Press, Amsterdam, Holland: http://www.jurix.nl/pdf/j03-06.pdf [accessed 27/6/11]

Gray, P. N. and Gray, X. (2008): The Chair of Artificial Legal Intelligence, in *Proceedings of* *the 4th AIJA (Australasian Institute of Judicial Administration) Law & Technology Conference*, CD.

Gray, P. N. and Gray, X. (2009): The Science of Intelligence for Big, Complex Work, *International Journal of Advanced Corporate Learning, 2(2):13-19*. This is a revised reprint of the publication, Gray, P. and Gray X. (2008): The Science of Intelligence for Big, Complex Work in *Proceedings of the First International Conference on E-learning and the Workplace*, Columbia University, NY.

Gray, P. N. and Gray, X. (2011): Quality Controlled Government with Spherical Logic, *Int. J. Interdisciplinary Soc. Sci., 5(10):271-284*: http://iji.cgpublisher.com/product/pub.88/prod.1305 [accessed 27/6/11]

Gray, P. N., Gray, X. and Richards, D. (2007): Gödel, Escher, Bach and Superexpertise, in *Lecture Notes in Artificial Intelligence (LNAI 4798)*, Springer, Berlin, Germany.

Gray, P. N., Gray, X. and Zeleznikow, J. (2007): Negotiating Logic: For richer or poorer, in R. Winkels (ed), *Proceedings of Eleventh International Conference on Artificial Intelligence and Law*, Palo Alto, Ca., USA, ACM Press, New York.

Gray, P. N., Gray X. and Zeleznikow, J. (2009): Intelligent Negotiation Technology, in *Proceedings of Legal and Negotiation Decision Support Systems (LDSS 2009): A Post-Conference Workshop at the 12th International Conference on Artificial Intelligence and Law*, 4, pp. 38-54.

Gray, P. N. and Mann, S. (2003): The Fraunce (1588) model of case-based reasoning, in *Proceedings of 9th International Conference on Artificial Intelligence and Law*, ACM, New York.

Gray, P. N., Tierney, R., Gray, X. and Treanor, L. M. (2006): eGanges: Pervasive Peacemaker, in *Proceedings of the First International Symposium on Pervasive Computing and Applications*, Urumchi, China, August 2006, IEEE.

Gray, P. N., Gray, X. and Treanor, L.M. (2006): Handheld intelligence for contemporary free trade, in *Proceedings of third International Conference on Contemporary Business*, Leura, N.S.W. (CD), Charles Sturt University, Australia.

Gray, P. N., Gray, X. and Treanor, L.M. (2007): Clues, Cues, Combinatorics and Super-agile Management Androgogy, in *Proceedings of the International Business Research Conference*, University of Technology, Sydney, CD.

Gray, X., Gray, P. N. and Zeleznikow, J. (2011): “Supporting the Harvard Model of Principled Negotiation with Superexpertise”, in Proccedings of 4th Workshop on Legal Informatics and Legal Information Technology: BIS 2011 Workshops, Lecture Notes in Business

Information Processing (LNBIP) 97, W. Abramowicz, L. Maciaszek and K. Wcel (Eds.), pp. 301-312, Springer-Verlag, Berlin, Germany.

Harré, R. and Krausz, M. (1996): *Varieties of Relativism*, Blackwell, Oxford, England.

Hofstadter, D. R. (1979): *Gödel, Escher, Bach*, Penguin Books, Harmondsworth, England.

Horrock, I. (2005): *Ontologies and the semantic web*: http://www.epsg.org.uk/pub/needham2005/Horrocks_needham2005.pdf [accessed 27/6/11]

Ignizio, J. P. (1991): *Introduction to Expert Systems: The Development and Implementation of Rule-Based Expert Systems*, McGraw-Hill Inc, New York.

Ishikawa, K. (1985): *What is total quality control? The Japanese way*, translated by D.J. Lu., Prentice-Hall Inc., Englewood Cliffs, N.J., USA.

Janhunen, T. (1998): On the intertranslatability of autoepistemic, default and priority logics, and parallel circumscription, in *Proceedings of the Sixth European Workshop on Logics in Artificial Intelligence (JELIA'98)*, pp. 216-232.

Joroff, M. L., Porter, W. L., Feinberg, B. and Kukla, C. (2003): The agile workplace, *Journal of Corporate Real Estate, 5(4):293 - 311*.

Juran, J. M. and De Feo, J. (2010): *Juran's Quality Handbook: The Complete Guide to Performance Excellence*, 6th edition, McGraw-Hill Professional, New York.

Kleene, S. C. (1952): *Introduction to Metamathematics*, North Holland, Amsterdam.

Korzybski, A., (1958, originally published 1933, second edition, 1941): *Science and sanity: an introduction to non-Aristotelian systems and general semantics*, The International Non-Aristotelian Library Publishing Company, Lakeville, Conn., USA.

Langlotz, C. P. and Shortliffe, E. E. (1983): Adapting a Consultation System to Critique User Plans, *International Journal of Man-Machine Studies, 19:479-496*.

Liberty, J. (1998): *Beginning object-oriented analysis and design : with C++*, Wrox Press, Birmingham, UK.

Lodder, A. and Zeleznikow, J. (2005): Developing an Online Dispute Resolution Environment: Dialogue Tools and Negotiation Systems in a Three Step Model, *Harvard Negotiation Law Review, 10:287-338*.

Lukasiewicz, J. (1920): O logice trojwartosciowej (On three-valued logic), *Ruch filozoficzny, 5, pp.170-171*.

Majkic, Z. (2004): Ontological encapsulation of many-valued logic, in *Proceedings of Convegno Italiano di Logica Computazionale*: http://www.cs.unipr.it/CILC04/DownloadArea/Majkic-CILC04.pdf [accessed 27/6/11]

Majkic, Z. (2008): Bilattices, Intuitionism and Duality: Concepts and Foundations, *Journal of*

Multiple-valued Logic & Soft Computing, 14(6):525-564.

Marek, W. and Truszczynski, M. (1991): Autoepistemic logic, *Journal of the ACM, 38(3):588-619*.

Mayer, B. (2000): *The Dynamics of Conflict Resolution*, Jossey-Bass, San Francisco.

McCarthy, J. and Hayes, P.J. (1969): Some philosophical problems from the viewpoint of artificial intelligence, in B. Meltzer and D. Mitchie (eds), *Machine intelligence* 4, Edinburgh University Press, Edinburgh, Scotland.

McCarthy, J. (1980): Circumscription: a form of non-monotonic reasoning, *Artificial Intelligence, 13:27-39*.

McDonald, K. J., (2007): *A Business Value Focus for Portfolio Management*, Cutter Consortium, Agile Project Management Executive Summary, Vol. 8, No. 5. http://www.johnbryce.co.il/owpdf/STKI_A%20Business%20Value%20Focus%20for%20Portfolio%20Management-summary.pdf [accessed 27/6/11]

Menkel-Meadow, C. (1984): Toward another view of legal negotiation: the structure of problem solving, *University of California Law Review, 31, 755*.

Mnookin, R. and Kornhauser, L. (1979): Bargaining in the shadow of the law: The case of divorce, *Yale Law Journal, 88:950-997*.

Mnookin, R., Peppet, S. R. and Tulumello, A. S. (2000): *Beyond Winning: Negotiating to Create Value in Deals and Disputes*, The Belnap Press of Harvard University Press, Cambridge, MA, USA.

Nash, J. (1953): Two Person Cooperative Games, *21 Econometrica 128, 128-40*.

Newell, A. (1982): The Knowledge Level, *Artificial Intelligence, 18:87-127*.

Neyman, J. and Pearson, E.S. (1928): On the Use and Interpretation of Certain Test Criteria for Purposes of Statistical Inference, *Biometrika, vol 20A, no 1/2, pp.175-240*.

Ong, W.J. (1958): *Ramus Method and the Decay of Dialogue*, Harvard University Press, Cambridge, MA, U.S.A.

Orna, E. (2005): *Making Knowledge Visible*, Gower, Aldershot, England.

Owusu, Y.A., (1999): Importance of employee involvement in world-class agile management systems, *International Journal of Agile Management Systems, 1(2):107-115*.

Peirce, C.S. (1933, originally published 1883): Exact Logic, C. Hartshorne and P. Weiss (eds), *Collected Papers*, Vol.3, Harvard University Press, Cambridge, MA, USA.

Peirce, C.S. (1982): E.C. Moore (ed.),*Writings of Charles S. Peirce*, A chronological edition, Vol. 1 (1857-1866), Indiana University Press, Bloomington and Indianopolis, USA.

Peirce, C.S. (1984): E.C. Moore (ed.),*Writings of Charles S. Peirce*, A chronological edition, Vol. 2 (1867-1871), Indiana University Press, Bloomington and Indianapolis, USA.

Peirce, C.S. (1986): C.J.W. Kloesel (ed.), *Writings of Charles S. Peirce,* A Chronological Edition, Vol. 3 (1872-1878), and 4 (1879-1884), Indiana University Press, Bloomington and Indianapolis,.USA.

Popple (1995): *A Pragmatic Legal Expert System*, Dartmouth Publishing Company Limited, Aldershot, England.

Prakken, H. and Vreeswijk, G. A. W. (2002): Logics for defeasible argumentation, in D.M. Gabbay & F. Guenthner (eds.), *Handbook of Philosophical Logic*, 2nd edition, Vol 4, pp. 219-318, Kluwer Academic Publishers, Dordrecht/Boston/London.

Quinlan, J.R. (1979): Discovering rules by induction from large collections of examples, in Mitchie, D.E. (ed.) *Expert systems in the micro-electronic age*, Edinburgh University Press, Edinburgh, Scotland.

Raiffa, H. (1982): *Art and Science of Negotiation*, Harvard University Press, Boston.

Ramus, P., (1969, originally published 1543, trans. 1574): *Dialecticke*, C.A., Dunn (ed.), San Fernando Valley State College, Northbridge, CA, USA.

Reiter, R. (1980): A logic for default reasoning, *Artificial Intelligence, 13:81-132*.

Richards, D. C. (1998): *The Reuse of Knowledge in Ripple Down Rule Knowledge Based Systems*, PhD thesis, The University of New South Wales, Australia.

Richards, D. C. (2000): Giving Experts Ownership and Control over their Knowledge, in *Proceedings of OzCHI2000: Interfacing Reality in the New Millenium*, December 4-8, University of Technology, Sydney.

Rooney, D. (2007): An Organizational Structure for Agile Projects, *Agile Project Management Executive Summary, Vol. 8, No. 5*, Cutter Consortium, Arlington, MA: http://www.johnbryce.co.il/owpdf/STKI_A%20Business%20Value%20Focus%20for%20Portfolio%20Management-summary.pdf [accessed 27/6/11]

Salle, J.M. and Hunter, J. (1990): Computer/User Cooperation Issues for Knowledge-Based Systems: A Review, *Technical Report AUCS/TR9003*, Aberdeen University. Scotland.

Schalkoff, R. J. (2009): *Intelligent Systems: Principles, Paradigms and Pragmatics*, Jones & Bartlett Learning, London.

Shaw, M.L.G. and Gaines, B.R. (1991): A Methodology for Analyzing Terminological and Conceptual Differences in Language use Across Communities, in R. Kohler & B.B. Rieger (eds), *Contributions to Quantitative Linguistics, Proceedings of the First Quantitative Linguistics Conference, QUALICO)*, Trier, 1991, Kluwer, Dordrecht, pp. 91-138.

Shortliffe, E. (1976): *Computer-based medical consultation: MYCIN*, American Elsevier, New

York, USA.
Spencer, D. (2005): *Essential Dispute Resolution*, Cavendish Australia, Sydney.
Stelzner, M. and Williams, M.D. (1988): The Evolution of Interface Requirements for Expert Systems, in J.A. Hendler (ed.), *Expert Systems: The User Interface*, pp.219-247.
Stolarski, P., Tomaszewski, T., Zeleznikow, J. and Abramowicz, W. (2008): A Description of Legal Interpretations in Risk Management with the Use of Ontology Alignment Formalisms, Expanding the Horizons of ODR, in M. Poblet (ed.), Proceedings of the 5th International Workshop on Online Dispute Resolution, Florence, Italy, December 13, *CEUR Workshop Proceedings Series* (CEUR-WS.org), Vol. 430, pp. 73-83.
Stranieri, A., Zeleznikow, J., Gawler, M. and Lewis, B. (1999): A hybrid-neural approach to the automation of legal reasoning in the discretionary domain of family law in Australia, *Artificial Intelligence and Law 7(2-3):153-183*.
Tamma, V., Phelpsa, S., Dickinson, I. and Wooldridge, M. (2005): Ontologies for supporting negotiation in e-commerce*, Engineering Applications of Artificial Intelligence, 18(2):223-236*.
Valente, A. (1995): Legal Knowledge Engineering: A Modelling Approach, *Frontiers in Artificial Intelligence and Application, Vol 30*, IOS Press, Amsterdam, Netherlands.
Venn, J. (1880): On the Diagrammatic and Mechanical Representation of Propositions and Reasonings, *Philosophical Magazine and Journal of Science, Series 5, vol. 10, No. 59, July*.
Walton, R. E. and Mckersie, R. B (1965): *A Behavioral Theory of Labor Negotiations*, McGraw-Hill, New York, USA.
Zeleznikow, J and Vincent, A. (2007): Providing Decision Support for Negotiation: The Need for Adding Notions of Fairness to Those of Interests, *University of Toledo Law Review, 38:101-143*.
Zeleznikow, J. (2004): The Split-Up project: Induction, context and knowledge discovery in law, in *Journal of Law, Probability and Risk, 3:147-168*.
Zeleznikow, J. and Bellucci, E. (2011): Classifying Online Dispute Resolution by comparing family mediation and the Israel - Palestinian conflict, in *Proceedings of 13th International Conference on Artificial Intelligence and Law*, ACM, New York.