Chapter 2

ASPECTS OF KNOWLEDGE REPRESENTATION

Andrew Basden

1. INTRODUCTION

Programmers represent knowledge about the world in a computer language to produce an information system (IS) that may be used to help human beings in various tasks. For example, to make the process of writing contracts in the construction industry more open, the team led by Hibberd and Basden (1995) created the INCA knowledge based system (KBS) to write contracts that more accurately reflect the intentions of the parties than is possible by adapting standard forms of contract. On asking its user around three dozen questions and making inferences based on the answers obtained, the KBS selects appropriate clauses and combines them intelligently together to form the draft contract. An important feature of the KBS was that it encouraged the parties to explore the reasons for each clause, to question and change clauses. INCA 'contains' sophisticated knowledge about legal principles of contract and how to use it.

All this knowledge had to be represented in a programming language, or 'knowledge representation language', KRL. But what form of KRL is appropriate to such wide-ranging types of complex knowledge? Many of the issues in knowledge representation (KR) were explored in the 1980s but have now re-emerged in the field of knowledge management (KM), which seeks to represent wide-ranging knowledge relevant to an organisation, as a general aid to that organisation's prcesses.

There is one issue that had not been resolved by the end of the 1980s and still deserves our consideration. Reflecting on the experience of the 1980s, Brachman (1990) called for 'KR to the people'. KR had become a specialist field from which lay people were excluded. In KM, 'the people' are employees immersed in the knowledge that requires representation, and they should be the ones to represent it.

'KR to the people' is the issue we address in this chapter. It is part of a wider story, alongside the issues of knowledge elicitation and analysis that are discussed in chapters 3 to 5, and that of whether IS are effective and useful in aiding human tasks that is discussed in chapters 9 to 11. In this chapter, we are concerned solely with KR, and specifically with the basic facilities that enable 'the people' to represent knowledge easily.

1.1 Knowledge representation

In the process of representing knowledge, depicted in Fig. 1, a programmer (or 'knowledge engineer', KE) expresses knowledge in symbols offered by the chosen KRL (which can be textual or graphical). The knowledge so represented is that which is relevant to a domain of the world (such as contract authoring).

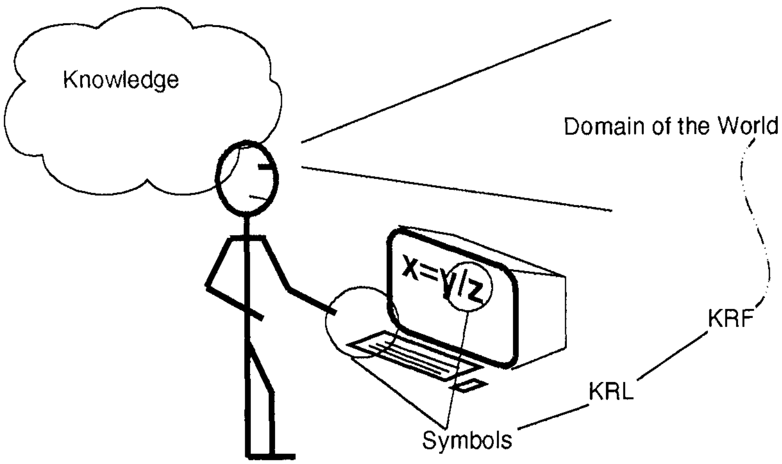

Figure 1. Knowledge representation as human activity

Traditional KRLs tend to be of certain basic types, which we will call knowledge representation formalisms (KRFs), developed out of pioneering work in the 1970s in artificial intelligence (AI) and computer science:

- ♦ production rules, in which knowledge is expressed as IF-THEN productions,
- ♦ logic programming (e.g. PROLOG), in which knowledge is expressed as predicate logic statements,

- functional programming (e.g. LISP), in which knowledge is expressed as nested function calls,
- frames and object oriented (OO) approaches, in which knowledge is expressed in terms of objects, classes, attributes and methods, and
- procedural languages like C, in which knowledge is expressed in terms of variables and instructions that manipulate them.

To design KRF software (for the KE to use) requires a number of decisions to be made about: types of 'thing' the KE is to be faced with (e.g. rules), ways these may relate (e.g. antecedent-consequent), types of property (e.g. truth values), types of activity (e.g. rule-firing), what valid inferences may be made (e.g. those of propositional logic), and types of constraint (e.g. that rules should not form an infinite loop). Such things are made available to the KE as atomic facilities but, at the bottom level, are implemented in machine code and memory structures.

But - as Brachman discussed - KR has proved more difficult than originally expected. Each KRF is appropriate for some types of knowledge and less appropriate for others; for example, it can be difficult to represent nuanced business strategy in any of the above KRLs. Moreover, if the domain involves several types of knowledge, the KRF used will be appropriate at some points, inappropriate at others. KR becomes esoteric expertise, complex and error-prone, and not suited to 'the people'. Expert KEs may be called in but they often fail to understanding the important nuances of the domain, so the ISs they build often fail to deliver the expected benefits when in use.

Thus the ideal KRF would be one that is appropriate to the whole range of types of knowledge that could conceivably be encountered.

This chapter, which is based on Basden (1993) but takes it further, discusses whether it might be possible to find such an ideal KRF. Using a concrete example of knowledge we might wish to represent, we briefly review the characteristics of various types of knowledge, which are discussed in Basden (1993). We outline the problems that arise in representing these using inappropriate KRFs and note that philosophy is needed to address them. After outlining some portions of a pluralistic philosophy, we show how they might be applied to develop more appropriate KRFs that enable diverse knowledge to be represented.

2. KNOWLEDGE REPRESENTATION PROBLEMS

The difficulties we are considering here are not those that arise from deficiencies in the KR software (which could be remedied by redesign) but those which seem inherent in the KR formalism. Consider the following scenario, illustrated in Fig. 2, in which we want an IS to help us keep track of birds' nests:

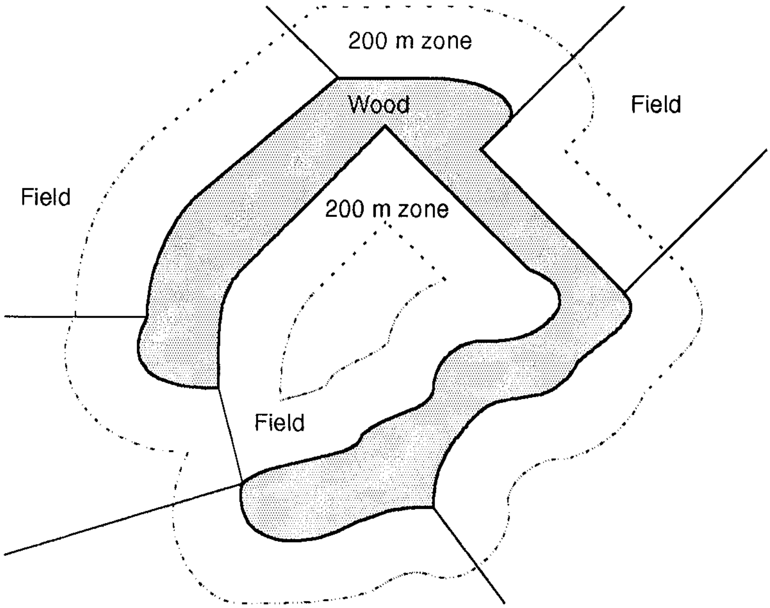

Figure 2. Nests, fields and woodland

- We have a terrain of rough pasture with fragments of woodland.
- The rough pasture is divided into fields.
- In the fields a species of bird nests.
- Eggs are laid in the nests, which (hopefully) will hatch into chicks.
- We want to keep track of, among other things, eggs in each nest.
- A species of fox lives in the fragments of woodland.
- At night the foxes emerge from the woodland to raid the nests (thus reducing the number of eggs in nests).
- The foxes tend to travel up to 200 m from the edge of woodland.
- We want to know which nests are in danger from foxes (so that we can, for example, watch and protect them).
- To do so requires obtaining permission from whoever owns the fields in which the nests are situated.
- So we need to know which fields each nest is in.
- But our knowledge of the dangers is limited (e.g. the behaviour patterns of the foxes) so we want also to search literature (on the Internet) to find relevant knowledge and link it to decisions we make.
- Finally, our usage of such a system might expand; for example, we might

need to start collecting and storing new information about nests that we do not currently anticipate.

All this knowledge must be represented in some KRF in order to build the IS we need. The reader will most likely agree that most of it can be understood intuitvely - the concepts of nests, fields, foxes, fragments of woodland, the concept of number of eggs per nest, that things change (eggs are laid and stolen), that certain nests are in certain fields which, conversely, contain those nests, that of ownership, the notion of fields and woodland as habitat, of the edge of woodland and of a zone up to 200 m from that edge, and so on.

Most knowledge that the KE must represent is explicit in the statement above, but some is tacit cultural knowledge. Our concern here, however, is not with whether knowledge is tacit or explicit, but with the diversity of concepts involved that seem intuitive or 'natural' to the lay person.

2.1 Representing knowledge to be used

Most ISs are designed to be used, and the best ones become part of the lifeworld of their users. After learning the features of the software itself, the user becomes familiar with the knowledge it holds, progressing from obvious, ordinary knowledge to the less obvious, exceptional knowledge, from seeing the knowledge as individual pieces to engaging with a nuanced, interwoven whole, from scepticism to trust.

What this implies is that not only should the knowledge of the domain encapsulated in the IS be accurate and understandable, but it should also be rich, nuanced and faithful to the recondite parts of the domain, so the user can rely on it without fearing lest some small but important piece of knowledge is missing. Much of the responsibility for ensuring this lies with the knowledge analysis and acquisition process (see chapters 3-5), but as Basden and Hibberd (1996) have argued, knowledge acquisition and representation cannot be separated, the latter often either hindering or stimulating the former. This means that any KRF the KE employs should facilitate faithful representation of a wide diversity of knowledge.

2.2 Appropriate KRFs

But which KRF should we use? As discussed more fully in Basden (1993), if we consider some of the 'aspects of knowledge' in our story above - items, relationships, values, changes, spatial and textual knowledge - we find that no KRF is appropriate to all of them.

An item (such as nest, field, fox) is a complex thing that persists over a period of time and has attributes that might change during this period. OO is well-developed for handling items of this kind, and PROLOG is very flexible in the way it breaks items down into statements about them. But procedural languages, in implementing items as structured collections of variables, are too rigid to allow new attributes to be added. Functional languages, though more flexible, hide the organic distinctness of items.

Relationships link items (such as "Field contains nest") and can be quite complex. They often have attributes (e.g. their start dates or strengths) and may even relate to each other (e.g. "The friendship between John and Jim was damaged by Jim's promotion"). Most real-life relationships have automatic inverses (e.g. if Field2 contains Nest5 then Nest5 is in Field2). Most are binary but not all are (an order line may be seen as a relationship between customer, supplier and product). Some types of relationships imply constraints, such as of hierarchy or prevention of loops ('directed acyclic graphs'). None of the KRFs above handles relationships well, most of them relying on implementing relationships as special attributes (known as pointers or keys) that hold identifiers to other items. This not only makes it tricky to implement genuine richness of relationships but it holds dangers (e.g. referential disintegrity, when the value in a key happens not to be a true identifier, or infinite loops when searching). As a result, for example, Blaha and Premerlani (1998:22) actively discourage consideration of inverses - but this inherently hinders the KE's ability to represent richness.

Values may be quantitative (such as 7, 'around 6') or qualitative (such as North, True, Loving), and are usually used as contents of attributes (e.g. 7 is NumberOfEggs attribute of a certain nest). Quantitative values may be incremented and imply notions like more, less and 'approximately near'. Procedural and OO KRFs offer a facility for simple values of both types, but logic programming has no inherent notion of quantity and must treat values as mere tokens, so it is cumbersome to increment or test for quantitative relations. (Practical logic programming KRFs handle quantity by means of hidden side-effects; see later.) The full richness of values extends beyond these simple notions to include, for example, degrees of precision (e.g. how near North?) and accuracy (how much to trust them). But few KRFs offer facilities to handle these.

Change is endemic in most applications - new nests being created, number of eggs in a nest changing, foxes raiding, and so on. Change can be either events or processes, and what triggers a change may be a particular time (e.g. spring time), or some sequence (egg laid then brooded then hatched), or some event (such as fox raiding) or on receipt of a message (such as alarm cry). Procedural and OO KRFs recognise the reality of change, with OO doing so in a slightly more sophisticated way, but logic programming has no inner notion of change and functional programming reduces change to the process of evaluating a function.

Spatial phenomena like shape, boundary, and a zone extending 200m out from the boundary, are important in many applications. Though the lay person finds such phenomena relatively simple to understand intuitively, none of the above KRFs offer spatial facilities. So none of them allows us to represent a spatially rich knowledge domain easily. (Software exists that encapsulates genuine spatial knowledge, including geographic information systems and Funts (1980) innovative 'geometric reasoner', but their facilities are not found in KRLs.)

Our final example is of information embodied in text or other symbolic form (diagrams). Our everyday experience of text - which is what we are concerned with in such an application - is of a complex structure that appears to be linear and hierarchical but is in fact interwoven in a more complex way (e.g. by both explicit cross references and subtle allusions), of interpreting and understanding it, of moving around it, of searching for approximately relevant material, etc. While most of the above KRFs offer simple text strings, none offer interwoven textual structure, approximate matching, aids to interpretation and the like that we encounter in everyday life. Software like word processors, Internet search engines and browsers offer such facilities, but none of this has found its way into KRFs yet.

2.3 Problems of inappropriateness

It is clear that none of the above KRFs make it easy to represent the everyday experience of all such aspects of knowledge, which is what may be necessary in a real-life application. Nor, indeed, does it seem that any KRF is able to approach the full richness of any single aspect. This leads to a number of problems in knowledge representation, many of them arising from pressure imposed by the restrictions in the KRFs.

a) **Richness** in knowledge may be reduced. For example, Blaha and Premerlani (1998) discourage using inverse relationships.

b) Knowledge can become **distorted** and so inaccurate or misleading. For example, if we can use only exact numbers, imprecision is masked.

c) Things for which the KRF is inappropriate, but which are acknowledged to be needed, tend to be implemented as **side-effects** - for example, PROLOG implements iterative change by means of the infamous 'cut-fail' technique. Side effects feel 'dirty', are difficult to learn and esoteric to use. 'The people' become confused and misled.

d) The knowledge base of the IS becomes more **complex**. For example, implementing inverse relationships by means of a pair of attributes (forward and back) introduces the need to keep them in synchronization even in the unforeseen circumstances. For example, the shape of the woodland in Fig. 2. and of the 200 m zone could each be represented as a list of (x,y) coordinates, which must then be managed. 'The people' should not have to bear such responsibility.

e) As a result, what seems 'natural' to our intuition requires extensive **KR effort** to represent. For example, to derive the buffered shape of woodland requires a complex trigonometrical algorithm that 'the people' are unlikely to know.

f) 'The people', under pressure to 'get something working', might inadvertently implement an **over-simplified** version of the aspect of knowledge, which later proves unable to support the full richness encountered in the domain. For example, a simple list of (x,y) coordinates cannot support a shape with holes in it (Fig. 2) - a very common mistake in spatial KR.

g) The **inner workings** of the implementation of an aspect become exposed and open to interference. For example, the 'key' by which a relationship is implemented might be given an illegal value.

h) Such interference can be **dangerous**. For example, an illegal value in a pointer might cause the whole computer to crash.

i) Finally, when a KRF covers several aspects of knowledge, the KE must decide on the **interface** between them.

Because of (a, b) the resultant IS can be inaccurate. Because of (b, c, e, f, g) KR takes on the nature of esoteric expertise. Because of (b, d, g, h) KR becomes more error-prone. Because of (c, d, e, i) the IS becomes more difficult to understand, which is problematic when, some years later, a new development team needs to upgrade it. Because of (a, b, f) some necessary upgrades are made impossible. Inappropriate KRFs, therefore, make Brachman's vision of 'KR to the people' difficult to achieve.

The conventionally recognised criteria for a good KRF, sufficiency, efficiency and expressive power (Minsky, 1981; Levesque and Brachman, 1985), do not address such problems. All they do is make (d, e) easier to achieve by expert programmers, and a focus on sufficiency and expressive power in particular often proves counter-productive because it suggests side-effects (c) are no problem.

2.4 Need for ontology

The issue of appropriateness presupposes a delineation of distinct aspects of knowledge and an understanding of the richness of those aspects. This in turn presupposes an ontological commitment to a vision of the nature of the diversity we encounter in the world. Tacit recognition of this may be seen in the proliferation, in the 1990s, of attempts to create 'ontologies' for KR and KM. But, with a few notable exceptions such as Wand and Weber (1995) who created a KR ontology based on Bunge (1977, 1979), most of these have only a tenuous link with philosophical ontology. Even Wand and Weber's ontology may be criticised because Bunge did not start from the notion of everyday lifeworld but from an assumption of an hierarchical system of levels informed mainly by the natural sciences (discussion of this must wait until another time).

We will explore an ontological approach to designing KRFs that derives from the philosophy of the Dutch thinker, the late Herman Dooyeweerd (1894-1977), which is based firmly in the everyday lifeworld.

3. DOOYEWEERD'S ASPECTS AS BASIS FOR APPROPRIATE KNOWLEDGE REPRESENTATION

3.1 Modal aspects

We have spoken loosely about distinct 'aspects of knowledge'. Dooyeweerd also spoke of aspects, and indeed we shall find some correspondence between the two. On the first page of his *magnum opus*, *A New Critique of Theoretical Thought* (1955), he listed fifteen:

> "A[n] indissoluble inner coherence binds the numerical to the spatial aspect, the latter to the aspect of mathematical movement, the aspect of movement to that of physical energy, which itself is the necessary basis of the aspect of organic life. The aspect of organic life has an inner connection with that of psychical feeling, the latter refers in its logical anticipation (the feeling of logical correctness or incorrectness) to the analytical-logical aspect. This in turn is connected with the historical, the linguistic, the aspect of social intercourse, the economic, the aesthetic, the jural, the moral aspects and that of faith. In this inter-modal cosmic coherence no single aspect stands by itself; every-one refers within and beyond itself to all the others."

Unlike Bunge, however, Dooyeweerd related aspects to everyday experience. Immediately before the above text, we find:

> "If I consider reality as it is given in the naïve pre-theoretical experience, and then confront it with a theoretical analysis, through which reality appears to split up into various modal aspects then the first thing that strikes me, is the original *indissoluble interrelation* among these aspects ..."

In a footnote he explained:

> "[By aspects] are meant the fundamental universal modalities of temporal being which do not refer to the concrete 'what' of things or events, but are only the different modes of the universal 'how' ... For instance, the historical aspect of temporal reality is not at all identical with what actually happened in the past. Rather it is the particular mode of being which determines the historical view of the actual events in human society. These events have of course many more modal aspects than the historical."

Though Dooyeweerd began by setting out these aspects, he did not expect his readers to accept them uncritically but, at this stage, was merely setting the scene for a rigorous exploration not only of the structure of reality but also of the very nature of philosophical thought itself, by which we might address the structure of reality.

During his lengthy work, he uncovered the presuppositions underlying Western thinking since its root in Greek thinking 2,500 years ago, showed how they are responsible for the deep recurrent problems in Western thought, and then, starting from different presuppositions, developed a positive philosophy (in which aspects play an important part) to account for the nature of being, meaning, process, knowledge, normativity and time, which is breathtaking in its scope and novelty. This provides an underpinning for the aspects he introduced on his first page.

Many of the chapters of this book discuss parts of Dooyeweerd's philosophy; so here we will outline only those parts that help us understand appropriateness of KR and aspects of knowledge.

3.2 Roles of aspects

Many thinkers make reference to aspects, maybe under a different name, largely as irreducible categories that should be separately taken into account in analysis (such as Checkland's E's that are discussed in chapter 5). It was one of Dooyeweerd's insights however (Henderson, 1994:37-8) that aspects are not just categories of meaning but, being grounded in a transcendental notion of Origin, possess a modal character which enables creation to Be and Occur. Each aspect provides or enables:

- distinct ways in which things may be meaningful
- distinct ways in which things make sense (rationality)
- distinct modes of being

- distinct ways of functioning
- basic kinds of properties
- ways in which things can relate
- a distinct sphere of normativity.

Each aspect is a whole constellation of meaning, which accounts for the richness of meaning we experience. But at the centre of each is a kernel meaning can be grasped with the intuition, but never fully grasped by theoretical thought. In approximate terms the kernel meanings of the aspects (reversed in sequence, and with some names different from above partly because Dooyeweerd varied the names he used):

- Pistic aspect: faith, commitment and vision
- Ethical aspect: self-giving love, generosity, care
- Juridical aspect: 'what is due', rights, responsibilities
- Aesthetic aspect: harmony, surprise and fun
- Economic aspect: frugality, skilled use of limited resources
- Social aspect: social interaction, relationships and institutions
- Lingual aspect: symbolic signification
- Formative aspect: formative power, achievement, history, technology
- Analytical aspect: distinction, conceptualizing and inferring
- Sensitive aspect: sense, feeling and emotion
- Biotic (organic) aspect: life functions, integrity of organism
- Physical aspect: energy and mass
- Kinematic aspect: flowing movement
- Spatial aspect: continuous extension
- Quantitative aspect: discrete amount.

As an example, the constellation of the lingual aspect includes (and is expanded later):

- ways in which things may be meaningful: e.g. writing, rather than marks on paper (which would be sensitive) or poetry (which would be aesthetic), medium rather than substrate (which would be physical)
- ways in which things make sense: e.g. flow of argument or narrative
- modes of being: e.g. word, sentence, paragraph, diagram, noun, verb, bullet list, abbreviation, hyperlink
- ways of functioning: e.g. speak, write, read, translate, search.
- basic kind of properties: e.g. emphasis, clarity of wording, connotation
- ways in which things can relate: e.g. cross-reference, synonym
- sphere of normativity: e.g. words should be drawn from a vocabulary, text should obey laws of syntax.

3.3 Characteristics of aspects

Aspects are irreducible to each other, in that the meaning of one aspect can never be derived from, nor explained solely in terms of, that of others. Attempts to do so denature it and end in antinomy or paradox (Dooyeweerd used the example of Zeno's paradox to illustrate reduction of the kinematic to the spatial aspect). What this means is that each aspect is equally important and that the beings, functions, etc. that different aspects make possible must be considered separately. By contrast, systems theory sees aspects (levels) as 'emergent' from others by virtue of structure and behaviour rather than meaning.

Though irreducible to each other, there is also an "indissoluble inner coherence" between aspects (Dooyeweerd, 1955,I:3), which is of two types, analogy and dependency. Analogy means that each aspect contains echoes of the others (e.g. logical entailment (analytic aspect) echoes causality (physical aspect)). But it is inter-aspect dependency that more directly concerns us here. Aspects depend on each other for their full meaning. Dependency defines the order in which the aspects are normally listed, with pistic being 'latest' and quantitative, 'earliest' (note, not 'highest, lowest', to avoid any connotation that any aspect is more important than others). It is in two directions: an aspect depends on earlier ones for its implementation and on later ones for its full meaning. Thus, for example, the lingual aspect could not operate without formative structure, and symbolic signification would be of limited use were it not to allow us to communicate and thus relate socially. Note that dependency does not entail reducibility.

The earlier aspects are determinative, in that the outcome of response to their laws (e.g. laws of physics) is (largely) determined, whereas the later aspects are normative (e.g. laws of linguistics), allowing and enabling freedom.

Finally, no aspect is absolute, in the sense of being a self-sufficient foundation of meaning for the others. Nor is the entire suite of aspects taken together absolute, but all depend ultimately for their meaning on a Divine Origin.

One important corollary of this is that, as Dooyeweerd explicitly stated in (1955,II:556), "the system of the law-spheres [aspects] designed by us can never lay claim to material completion. A more penetrating examination may at any time bring new modal aspects of reality to the light not yet perceived before. And the discovery of new law-spheres will always require a revision and further development of our modal analyses." This is because to make a distinction among aspects we have to function in the analytic aspect, and if this is non-absolute then it is impossible to rely on the distinctions that it produces, such as a suite of aspects. However, this is not a recipe for universal skepticism since, as Dooyeweerd claimed, the aspects may be known by the intuition. Intuition, to Dooyeweerd, is not some mysterious nor instinctual way of knowing, but an awareness in which all aspects play their part.

This relates to Dooyeweerd's theory of everyday life (lifeworld), in which everything involves (i.e. can have meaning in) every aspect. For example, as I write this, I am functioning primarily in the lingual aspect, but I am also forming my thoughts (formative), distinguishing what to write (analytic), seeing the screen (sensitive), breathing (biotic), and exerting force on the keyboard (physical). I am also trying to give the reader what is due (juridical), to present a coherent argument (aesthetic), within a limited number of pages (economic), and I am treating the reader not as mere information sink but as fellow human being (social aspect).

4. A DOOYEWEERDIAN APPROACH TO APPROPRIATENESS

In like manner, use of an IS is multi-aspectual lifeworld (see chapter 11). Therefore, in principle, every aspect is important, and knowledge from every aspect should be represented in an IS. If this is to be done easily by 'the people', then the KRF they use should facilitate, and not hinder, the representation of knowledge of each aspect. We will now discuss how we may apply these portions of Dooyeweerd's philosophy to understand appropriateness, and thence to propose an approach to designing KRFs. First, we employ our knowledge of some aspects to design aspectually-oriented KRFs, then consider new aspects, and finally discuss the integration of such KRFs into a single, overall KRF.

4.1 Aspects of knowledge and individual KRFs

What we called aspects of knowledge correspond quite closely with Dooyeweerd's modal aspects (the use of the word 'aspect' in both cases is fortuitous since at the time the author knew nothing of Dooyeweerd):

- items: analytic aspect (distinct concepts)
- relationships: formative aspect (formed structure)
- values: quantitative aspect (discrete amount)
- spatial: spatial aspect (continuous extension)
- text: lingual aspect (symbolic signification).

We will examine these five aspects and see how, by reference to some of their philosophical roles, we can design a KRF for the corresponding aspect of knowledge. Referring back to the design decisions in section 1:

- Mode of being indicates types of 'things' to make provision for.

- Way of functioning suggests activity to provide as procedures.
- Basic types of property suggests attribute types.
- Way of relating suggests type of relationship.
- Aspectual rationality indicates inferences to be built into our KRF.
- Normativity suggests constraints that would be meaningful.

We work these out tentatively (and without any attempt at completeness) for several aspects as follows:

For the quantitative aspect (discrete amount),
- 'Things': integers, ratios, fractions, proportions, etc.; also types that anticipate later aspects such as 'real numbers' for the spatial aspect.
- Inferences: those of arithmetic
- Attributes: quantity, accuracy, units, etc.
- Relationships: greater and less than, sets, etc.
- Procedures: incrementing, scaaing, statistical functions, etc.
- Constraints: e.g. a given quantity remains that quantity until changed
- Example KRF: APL. Example software: calculators, statistical packages.

For the spatial aspect (continuous extension),
- 'Things': shapes, lines (straight or curved), areas, regions, dimensional axes, etc.
- Inferences: those of geometry
- Attributes: size, orientation, distance, side (in, out, left, right), etc.
- Relationships: spatial alignments and arrangements, touchings, crossings, surroundings, toplogy, etc.
- Procedures: join, split, stretch, deform, rotate, overlap, expand, etc.
- Constraints: e.g. boundaries should not have gaps
- Example KRF: Funt's (1980) geometric reasoner. Example software: geographic information systems, computer aided design, drawing packages.

For the analytic aspect (distinction),
- 'Things': concepts, objects, labels to identify things, etc.
- Inferences: those of formal logic
- Attributes: truth values, difference and sameness, etc.
- Relationships: contradiction, logical entailment, identity, etc.
- Procedures: e.g. distinguish, deduce
- Constraints: e.g. principle of non-contradiction, entity integrity (as in relational databases)
- Example KRF: PROLOG. Example software: cognitive mapper.

For the formative aspect (formative power),
- 'Things': structuring, relationships, plans, means and ends, goals, intentions, power, etc.
- Inferences: those of synthesis
- Attributes: feasibility, efficacy, version, strength (as of a relationship), etc.

- Relationships: means and ends, the purpose of something, sequence of operations (history), part-whole, etc.
- Procedures: form, compose, relate, revise, undo (to previous version), seek, effect a meaningful change (change a state), plan, etc.
- Constraints: e.g. referential integrity
- Example KRF: production rules. Example software: planning software, project control software.

For the lingual aspect (focusing on syntax and semantics, of text r.t. graphics):

- 'Things': nouns, verbs, etc.; words, sentences, etc.; bullet lists, headings, cross references, quotations, etc.; word roots, languages
- Inferences: those of syntax, semantics, etc.
- Attributes: tense, case, emphasis, cultural connotation, etc.
- Relationships: synonyms, antonyms, opposites, cross references, rhymes, thesaurus relationships, etc.
- Procedures: write, draw, understand, send message, text search, find equivalent meaning, translate, etc.
- Constraints: e.g. words to be within a vocabulary (spell checkers), syntax to conform to a grammar (grammar checkers)
- Example KRF: SGML, HTML. Example software: word processor, knowledge representation software.

Even though the features we have mentioned for each aspect are just examples, they exceed those supported by most extant KRFs. Yet they are not esoteric, but are features encountered in everyday living, of which some are already found in practical software oriented to the lifeworld). So why are they not implemented in KRFs? Such an aspectual approach to the design of KRFs might yield more comprehensive KRFs, which would go a long way to fulfilling the aim of 'KR to the people'.

But a number of issues still require our attention.

4.2 Change and qualitative values

Basden (1993) suggested that change (events, processes) is an aspect of knowledge, but there is no Dooyeweerdian aspect of change. Instead, to Dooyeweerd, change is bound up with time and inherent in every aspect.

The aspect of knowledge, values, originally included qualitative values (e.g. truth values), but has been aligned solely with the Dooyeweerdian quantitative aspect above. Like change, qualitative values are different for each aspect, and are constituted of its basic types of property. Table 2 gives a few examples of types of change and of qualitative for several aspects.

Table 2. Examples of change and qualitative value for certain aspects

Aspect	Example Change	Example Qualitative Value
kinematic	movement and flow	(mostly quantitative)
physical	causal reactivity	state (gas, liquid, solid)
biotic	growth and decay	alive v. dead
sensitive	stimulation and response	colour
analytic	from premise to conclusion	true v. false
formative	creation, construction and shaping	difficulty
lingual	utterance and understanding	which language
... up to ...		
pistic	the act of committing	sacred v. profane.

4.3 New aspects?

Basden (1993) discussed only four aspects of knowledge, conflating items and relationships into a single aspect and ignoring text. The question was raised whether there were other aspects of knowledge, and even whether those aspects are valid. At the time he had no other basis for answering this than intuition and observation of practical difficulties of trying to reduce one to another.

Dooyeweerd's aspects provides a sounder basis on which to answer these questions. Not only do the characteristics and philosophical roles of aspects provide a systematic way of considering aspects of knowledge and their corresponding KRFs, but it also provides, on one hand, a basis for splitting aspects of knowledge should that be necessary, and, on the other, a way of identifying possible new aspects and their KRFs, such as the economic or juridical. Some of the work on KM 'ontologies' may be seen as reaching for KRFs of later aspects. For some, such as the economic aspect, a useful initial KRF might be constructed, but for others, such as the ethical and pistic, it is not yet clear what shape a KRF would take.

4.4 Integrating KRFs

So far we have linked KRFs separately to different aspects. But any given application will involve representation of knowledge of several (if not all) aspects, and so the KRF we seek does not centre on a single aspect, but includes KRFs for all relevant aspects integrated together. To achieve this we might combine several aspectual KRFs, but we do not want to end up with a confusing muddle. Dooyeweerd's theory of inter-aspect relationships, however, can help us: just as the aspects relate to each other by dependency and analogy, and yet remain distinct, so aspectual KRFs may be integrated into a system by dependency and analogy and yet remain distinct. That dependency does not entail reducibility is an important philosophical insight by Dooyeweerd, because it provides us with the possibility of integrating aspectual KRFs.

Inter-aspect dependency in the foundational direction implies that a KRF for any aspect will require the facilities offered by a KRF of an earlier aspect. For example, a KRF for the lingual aspect will require at least the (formative) ability to relate and structure things like sentences, and to make distinctions such as for purpose of emphasis (analytic aspect). This suggests that the architecture of an aspectual KRF will involve a module for each aspect, and that if aspect Y is later than aspect X, then a KRF which has a module for Y will also need a module for X.

Dependency in the anticipatory direction suggests that some of the things, functions, constraints, etc. in aspect X are only meaningful in anticipating a later aspect, Y. (For example, irrational numbers are meaningful by reference to the spatial aspect, and linguistic pragmatics by reference to the social.) Since some of the later aspects have yet to be opened fully, it is likely that there are things in earlier aspects whose full significance is yet to be recognised. This means that when we implement a KRF module for any aspect (possibly excepting the pistic), we should make it possible to extend its things, procedures, constraints, inferences, etc. by such means as call-back hooks. (This is, of course, standard practice in reusable software today, but an awareness of aspects might be a useful guide to this practice.)

By its nature, inter-aspect analogy is more difficult to clearly define. It may be that extensibility features similar to those required for anticipatory dependency will be adequate, but since their exact form is likely to be different, it would be wise to keep them separate.

4.5 Discussion

These are only initial suggestions that illustrate how KRFs might be designed by considering Dooyeweerdian aspects and their constellations of meaning. We have considered only a few aspects, and the approach still has to be properly explored, tested and refined, but there is reason to believe that, if reasonably full account is taken of the constellation of meaning of each aspect, it will address the problems we identified in section 2.3 that result from inappropriate KRFs:

a) **Richness** need no longer be reduced inadvertently because all that is meaningful in an aspect is provided.

b) For a similar reason, **distortion** need no longer occur.

c) Implementation by **side-effects** is no longer necessary if all relevant aspects are included.
d) KB **complexity** is reduced to that which is intrinsic to the domain and not artificially increased by extraneous programming needed to implement other aspects.
e) KR becomes more 'natural' because all that is intuitive in each aspect is provided by atomic features and the KE no longer has to implement them, so less **KR effort** is required.
f) Thus there is less danger that a **simplified version** will be implemented.
g) Since all that is meaningful in each aspect is provided as atomic features, the **inner workings** of their implementation are no longer exposed.
h) So the attendant **dangers** are reduced.
i) In an integration of aspectual KRFs **interfacing** between the KRFs is already built-in and the KE no longer has to take responsibility for it.

The vision of an integrated, multi-aspectual KRF presented here is a long-term one. Not only are individual extant KRFs impoverished versions of the full aspectual KRFs envisaged, but the challenge of integration has yet to be met. It may be possible, during a period of research and experience, to gradually build up a sophisticated KRF that is appropriate to many aspects. The author's Istar KBS software (Basden and Brown, 1996) started to be developed along these lines.

However, there are several problems with this proposal that need to be addressed over the longer term. First, it is not always clear what shape a KRF might take for the later aspects, such as the juridical. Before we can tell whether the proposal made here has any merit, we should attempt to define KRFs for every aspect. There is evidence that this may be possible and even desirable, because features from later aspects - such as the juridical feature of copyright notices and authentication checks - are becoming important in practical software even today. At present, they are being introduced on an ad-hoc basis, which might insert problems for later; it may be that Dooyeweerd's aspects can enable a more systematic and future-proof approach.

Second, if we are to integrate aspectual KRFs, meaning in one aspect has to be translated to that of another. How this may best be done is not yet clear. The given suite of aspects proposed by Dooyeweerd does at least motivate us to do this, and his notion of the two types of relationship between aspects (dependency and analogy) can inform such consideration.

Third, if a complete set of facilities were implemented for every aspectual KRF would not the resulting software package be completely unwieldy? Though this has not been researched yet, there are two reasons for believing this might not be so. One is that the aspects are readily learned, and intuitively grasped; as Winfield (2000) and Lombardi (2001) have found, it becomes second-nature to consider them in any situation; see chapter 4. The other is that since the aspects have irreducible meaning, then it is possible that the symbols that express that meaning can be designed to be relatively independent of each other.

5. CONCLUSION

This is the only chapter in this book devoted to information technology as such. It has been concerned with the basic knowledge representation formalisms from which information systems may be constructed. It has been argued that, if we are to achieve Brachman's (1990) call for 'KR to the people' - a call that has yet to be fulfilled - then the KRF should be appropriate to all the diversity of knowledge found to be relevant in the domain of interest. This means that the facilities available to the knowledge engineer should enable all the richness of the domain knowledge to be fully represented, without distortion, without undue complexity and yet without simplification, without making the whole process more prone to errors than it need be, and without the need for esoteric knowledge of how to represent one aspect of knowledge in terms of another. We have argued that one way to achieve this is to ensure that the KRF gives every aspect of knowledge its due.

Aspects of knowledge, it has been further suggested, may be derived from the modal aspects proposed as a philosophical ontology by Dooyeweerd (1955), and that from these appropriate KRFs may be designed. All human life, including using an IS to aid our tasks, is multi-aspectual, involving every aspect. Appropriateness of a KRF is constituted in its ability to enable the knowledge engineer to give aspects of the everyday life of using an IS their due.

Modal aspects are irreducible spheres of meaning that are more than categories; they enable being, doing, relating, properties, norms, and each of these indicates a different portion of a KRF. Thus our proposal is that we should first devise a KRF for each aspect, and then integrate them together. There are philosophical reasons to believe (in the general case) not only that every aspectual KRF is needed (because of aspectual irreducibility) but also that integration is possible (because of an "inner indissoluble coherence" between the aspects).

A number of limitations in the proposal have been discussed. These may be taken to motivate future work - the development of KRFs for every aspect, of interfaces between aspects, and of ways to make the whole easy to use.

There is a surprising twist in this proposal. There is a general assumption that the inner design of technology is a purely 'technical' matter, divorced from the 'human' side, and that for a 'socio-technical' approach the technical must somehow be plugged into the human or social. The assumption may even be detected in chapter 1 of this book, in that the 'integrated vision for technology' sees the human, social context as outside the technical. But here we have seen that the very inner design of information technology - that central part which enables us to write programs - is intimately connected with all aspects of human everyday living. The human, social context is, or should be, already inside the very guts of information technology.

http://www.springer.com/978-0-387-32150-9

In Search of an Integrative Vision for Technology
Interdisciplinary Studies in Information Systems
Strijbos, S.; Basden, A. (Eds.)
2006, XX, 310 p., Hardcover
ISBN: 978-0-387-32150-9