Chapter 6.

A FRAMEWORK FOR UNDERSTANDING INFORMATION SYSTEMS DEVELOPMENT

Of course information systems development (ISD) is more than programming. But what is it? As Vignette 3 in the Preface shows it does involve the creativity and joy of programming, but it also involves a host of other things -- knowledge elicitation, understanding what knowledge is, respect for others, resource planning, responsiveness to the changing desires of users, awareness of the role of the IS in everyday life, hard work, frustration, satisfaction, and much more. Others have found similarly over the past 50 years, though they might have described them in more conventional ways, such as user requirements analysis, design, implementation, validation, verification, documentation, maintenance, and so on.

Information systems used by us need to be designed and developed. ISD is human activity. The main theme of most research in ISD is methodology to guide that activity. The central philosophical question addressed in this chapter is: what is the nature of ISD, including its norms? The central practical question is: what should guide ISD? These are the ways this chapter tries to formulate a framework for understanding ISD.

This chapter explores how Dooyeweerd's philosophy might help us understand the challenges and issues in ISD. The information system that is developed includes both the technical artefact or system and the human context of its use, which is often organisational. The communities of practice and research in this area include those involved in programming, system design, systems analysis, organisational analysis, knowledge elicitation, modelling, and many more. First this chapter reviews the history of ISD and paradigms, and shows briefly why a new paradigmatic approach might be useful. Then it applies Dooyeweerd's notion of multi-aspectual functioning to understand what goes on in ISD, and derives a tentative framework for understanding it.

6.1 APPROACHES TO ISD

6.1.1 Brief History of ISD

In the early days ISD was programming, and this was largely a technical creative activity, which could be quite unstructured, though techniques found to work in one project would be carried over to others. But, as Hirschhein, Klein and Lyytinen [1996:29] put it, "projects failed due to the lack of methodical guidelines and theoretical conceptions of IS." ISD methodology became an important issue, and a topic for research. Hirschhein et. al. give a brief historical overview of the field, as seven generations of ISD methodology which they discern to have arisen since the mid 1960s,

each in response to a particular problem that was perceived:

- Life-cycle methods: concerned to control the whole life of an ISD project (from user requirements analysis, through the actual programming and testing stages, to delivery), usually by means of standardization;

- Structured approaches: concerned to increase productivity of the development team and ensure that the IS developed are more maintainable

- Prototyping and evolutionary approaches: concerned about the rigidity of the above, and that it is more important to get the right system rather than get the system right, by exposing users frequently to version of the system and responding to their criticisms;

- Socio-technical, participatory approaches: concerned to ensure participation of users so that they, rather than the development team, are in final control of ISD;

- Sense-making and problem-formulation approaches: concerned to ensure that multiple perspectives (not just those of users or developers) have influence in ISD;

- Trades union led approaches: concerned that workers' rights and industrial democracy should prevail in ISD and/or in the social situation in which the IS is used;

- Emancipatory approaches: concerned about barriers to effective communication due to power and social differentiation, by encouraging the questioning of dominant forms of thinking and access to information.

(Note, however, that unstructured programming has continued to this day, especially for small programs or in amateur situations, such as early games development.)

This picture, however, should perhaps be extended with two other 'generations'. One is approaches to the development of knowledge based systems (KBS) such as Elsie in chapter 4; though this could, arguably, be seen in the above terms, there are specific challenges around elicitation of good quality knowledge that should be encapsulated into the KBS. The other is so-called agile system development methods (SDMs), many of which have developed since 1995, such as Beck's [2000] Extreme Programming, which seek to achieve the above benefits but without a heavy overhead of questioning discourse that often attends sense-making and emancipatory approaches. Agile methods, more than most, aspire to an everyday stance rather than one driven by a particular theory or method.

6.1.2 ISD Paradigms

What drives the development of ISD approaches is the paradigm or

world-view that researchers and practitioners in the area hold as part of a social system or culture. A classic model of sociological paradigms is that by Burrell and Morgan [1979]. Paradigms are seen as meta-theoretical assumptions about the nature of the subject of study. Four paradigms are differentiated by the interaction of two dimensions: objectivism-subjectivism and order-conflict. Objectivism assumes that the world is 'given' and we can apply models and methods derived from the natural sciences to study it, whereas subjectivism denies both of these and instead probes the subjective experience and beliefs of individuals. 'Order' assumes the social world is, or should be, stable, consensual and integrated, whereas 'conflict' assumes change which is conflictual and coercive in nature. The four paradigms are:

- functionalism (objectivism-order)
- social relativism (subjectivism-order)
- radical structuralism (objectivism-conflict)
- neohumanism (subjectivism-conflict).

This has been referred to in two ways to understand perspectives that have driven the development of ISD methods.

6.1.2.1 Systems Approaches

Jackson [1991] uses the Burrell-Morgan model to inform a study of systems approaches. All presuppose that some situation (system state) needs to be changed. The organisation-as-system approach treats the whole organisation as a system, which has goals and the means to achieve them by the operation of its subsystems. But in most organisations it is situations rather than the organisation as a whole that needs changing, and the remaining systems approaches address that.

Hard systems thinking (HST) assumes we can know both the state of the current system and the state we desire. It sees the main challenge as to identify how to reach one from the other, preferably employing mathematical equations or propositional logic. The potential of computers to facilitate this, especially for administrative or industrial processes, was early recognised. In ISD driven by HST it is assumed that the role of ICT is to control or 'objectify' the situation, removing uncertainty.

Checkland [1981] argued that HST is fundamentally unsuited to management decisions, in which different players appreciate the situation in different ways and see different things as problematic in it. Hence they cannot necessarily agree on what the relevant system is, let alone what the desired new state should be. He proposed the term 'soft systems thinking' (SST) to differentiate what motivated his Soft Systems Methodology (SSM) from HST. SST is more suited to 'human activity systems'. The focus in SST is to expose the diversity of perspectives, and, welcoming all, try to reach a consensus about what should be done, including what IS should be developed.

SST however has been criticised by Jackson [1991] and Ulrich [1994], as being isolationist, assuming consensus rather than conflict,

and taking participation for granted, unable to handle social power structures and not critically self-reflective. They have developed two strands of what is known as Critical Systems Thinking (CST), both of which see themselves as in a line of progress: HST -- SST -- CST. Using Habermas' social theory, especially from [1972], CST believes there to be three types of rationality and interest, the instrumental interest of the empirical-analytic sciences, which characterizes HST, interpretive rationality of historical-hermeneutic sciences, which characterizes SST, and emancipatory interest of the critical sciences, which characterizes itself. In emancipation -- for example from oppressive work conditions common in the 1970s or from unconscious compulsions -- CST recognises a transcending normativity in systems design.

Jackson [1991] discusses these in more depth, but the influence of HST may be traced in the first two, SST in the next three and CST in the last two historical generations above.

6.1.2.2 Direct use of Burrell and Morgan

Instead of going via systems approaches, Hirschheim, Klein and Lyytinen [1995] have applied Burrell and Morgan's model directly to ISD. After examining the ontological epistemological and value assumptions of the four paradigms, they discuss the impact each paradigm would have on ISD as such (including role of the IS designer, nature of IS application, objectives for IS design and use, legitimation of the objectives, and deficiencies of each paradigm), on ISD functions (including preferred metaphor for defining information and for framing ISD, problem finding and formulation, analysis, logical design, 'physical' design and technical implementation, organisational implementation and maintenance), and on aspects of the developed system (including technology architecture, kind of information flows, control of users, control of systems development, access to information, error handling, training and raison d'etre). A selection of their analysis is presented in Table 6.1.2.2.

Table 6.1.2.2 Impact of paradigms on some aspects of ISD

ISD aspect	Functionalist	Social Relativism	Radical Structuralism	Neohumanism
Role of IS designer	Expert (master engineer)	Catalyst	Warrior	Emancipator
Information seen as:	Product, made available and traded	Reflective journey with partner	Means of manipulation; weapon in ideological struggle	Means of control, sense-making, argumentation
Raison d'tre	Maximizing savings, minimising costs, improving competitive advantage	Improving creativity, sense-making	Reducing alienation by giving the workforce the control of the productive resources	Emancipation from unwarranted constraints (physical, social) by improved control and understanding

6.1.3 Practical Critique of Paradigms

Burrell and Morgan's model has thus been used, both directly and indirectly, as a framework to understand ISD, and still is today.

It has received some criticism as being over-simplified [Hirschheim et. al., 1995:49]. For example, it seems to overly narrow one's view, making it difficult to be open to the diversity of everyday experience. For example, the roles Hirschheim et. al. recognise for IS designers -- expert, catalyst, warrior and emancipator -- do not exhaust all the roles in the experience of the designer. Even though there might be situations when it is valid for each of these roles to predominate, there are many others when other roles might seem more appropriate. For example, the roles which this author has taken on in his experience of ISD include many that go beyond these four. They are listed in Table 6.1.3, referred to later. (Explorer means that ideas were explored by means of ISD, stimulant, that the client's views were to be stimulated, butler, that he was serving the needs of a project but had considerable responsibility, but was not in the role of expert, artist, that he tried to generate something beautiful, teacher, that he taught others about ISD and IS.)

Table 6.1.3. ISD roles assumed by the author

Project	Reference	ISD role
C.A.D.	Basden & Nichols (1973)	Explorer
Industrial data	-	Hired hand
Clinics	Basden & Clark (1980)	Butler (respobsible servant)
AusCor	Basden & Hines (1985)	Stimulant/teacher, Explorer
Wheat Counsellor	Jones & Crates (1984)	Explorer, Expert, a bit of Warrior
Elsie	Brandon, Basden, Hamilton, Stockley (1988)	Explorer, Teacher
Istar	Basden & Hibberd (1996)	Emancipator, Explorer
KgSvr	Basden (2000)	Butler
IRKit	-	Explorer, Artist
'New View' web site	www.basden.demon.co.uk.xn/nv/	Warrior, Stimulant
BHG web site	www.basden.demon.co.uk.xn/bhg/	Stimulant

Likewise, information can be seen in other ways than product, journey, weapon and means of control or argumentation, and the raison d'etre for the IS can extend into many aspects, as discussed in chapter 4. Moreover, these paradigms do not capture the experience of the technology as such with which IS developers must engage (which is discussed in chapter 7).

Similarly, real-life ISD cannot be compartmentalized into HST, SST or CST, especially if it is of good quality. The good developer does take some things about the domain of application to be given (reflecting an aspect of HST) but does try to sensitive to a wide range of interpretations (SST) and does question the status quo (CST). It is often not entrapment by one of these systems approaches that prevents this so much as an attitude such as laziness. Just as with Burrell and Morgan's four paradigms, so these three systems approaches do not adequately represent fully the everyday approach taken in ISD.

6.1.4 Philosophical Critique of the Paradigms

However, the Burrell-Morgan paradigms can also be criticised philosophically. Hirschheim et. al. [1995:49] do cite criticisms of over-simplification, especially in its treatment of functionalism, which is much more nuanced than their model would suggest, and that the dichotomies are artificial. Nevertheless, they conclude, they see it as the best available at the time for ISD. Indeed, unless there is a fundamental philosophical problem, over-simplifications can be ameliorated.

A Dooyeweerdian point of view does indeed suggest there might be a fundamental problem. Eriksson [2006] has examined the systems approaches through the lens of Dooyeweerd's theory of ground-motives. He argues that HST, SST and CST all presuppose the Nature-Freedom Ground-Motive. HST is self-evidently of the Nature pole, SST is of the Freedom pole and "Compared with HST and SST, CST makes a serious attempt at representing the complete Nature-Freedom Ground-Motive. It not only articulates explicitly the two realms but also attempts to provide a link between the two." This is shown in Table 6.1.4 (which is adapted from Eriksson [2006:226]).

Table 6.1.4. The ground-motive commitments of varieties of systems thinking

	Systems Thinking Framework			
	HST	SST	CST	MST / DST
Dominating Religious Ground-Motive	Nature	Freedom	Nature - Freedom	Creation - Fall - Redemption

But, according to Eriksson [2006,p.226], CST "has not succeeded in solving the very fundamental tension between the realm of nature and that of freedom, which is: How can man maintain his autonomous freedom in a mechanistically determined world?" This is because it is rooted in Kant, whom Dooyeweerd criticised for not being critical enough. (Eriksson provides a cogent summary of Dooyeweerd's critique of Kant.)

MST, Multimodal Systems Thinking [De Raadt, 1991], conjoins Dooyeweerd's aspects to Beer's [1979] Viable Systems Model. Its use of aspects provides diversity of norms to act as practical guide. It claims to be founded on the Creation-Fall-Redemption Ground-Motive, but Eriksson criticises it for not taking seriously enough issues of Biblical interpretation and suggests that, because VSM presupposes the NFGM, so does MST. MST does indeed betray strong influences of the Nature pole. Nevertheless, De Raadt is to be applauded as the first attempt to define a systems approach based on Dooyeweerd. It is still in development and used in practical analysis.

DST, Disclosive Systems Thinking [Strijbos, 2006], seeks to develop a systems approach based on some parts of Dooyeweerd's thinking without the help of others. Its central idea is that of diverse

intrinsic normativity and sees ISD (and other business analyses) as a process of 'disclosing' the innate normativity of a situation. It has links with Schuurman's [1980] 'liberating vision for technology' discussed in chapter 8. It does not make use of Dooyeweerd's aspects. It has yet to be extensively applied to practical situations; Eriksson does little more than mention it.

Following his analysis of HST, SST, CST, MST, Eriksson discusses Jackson's use of Burrell and Morgan's [1979] model of sociological paradigms. He argues [2006,p.231-2] the model is based on the NFGM (both dimensions are expression of Nature v. Freedom) and thus "This both articulates and forces the investigated systems thinking paradigms into the unbridgeable tension of dualism, founded on the assumption of autonomous reason." As a result, "it does not allow the detection of problems in the very ground-motive that governs many of the systems thinking paradigms", "it mis-conceptualizes systems thinking paradigms that are not based on the Nature-Freedom Ground-Motive {e.g. MST}" and "does not inquire explicitly the sources of norms of these paradigms." The problem, Eriksson argues, is the Nature-Freedom Ground-Motive itself.

A new framework, based on a different ground-motive, CFR, is explored here, using Dooyeweerd's ideas, but not quite in the way either MST or DST does so.

6.1.5 Towards a Different Framework for Understanding

The central proposal of this chapter is that ISD may be understood as multi-aspectual human functioning, in which every aspect is important and deserving of attention. Each of Hirschheim et. al.'s generations, being an historical response to certain perceived problems (often brought about by previous generations), is likely to point to something important in the pursuit of ISD. The issue addressed in each generation becomes meaningful and problematic with reference to a particular aspect or two; Table 6.1.5 shows this author's analysis of them.

Table 6.1.5. Main aspects of ISD generations

Generation	Concerns	Aspect
Lifecycle	"to control .. standardization"	Formative
Structured approaches	"to increase productivity"	Economic
Evolutionary, Prototyping	"conncerned .. to get the right system"	Juridical
Socio-technical	"to ensure participation .. resolve conflicts"	Social
Sense-making and problem-formulation	"sense-making" "consensus among multiple perspectives"	Analytic Aesthetic
Trades union	"that workers' reights .. should prevail" ""that .. democracy should prevail"	Juridical Social
Emancipatory	"barriers to communicationg" "questioning of dominant forms of thinkin"	Lingual Pistic

The shalom principle (§3.4.3) implies that focus on one aspect to the exclusion of others will inevitably give rise to problems, which will be associated with one or more of the ignored aspects. This can account for why each generation led to others. But it also suggests that for high quality ISD all aspects need to be taken into account in ISD. This does not, however, indicate that we should simply amalgamate all the generations. For one thing, the ethical aspect is under-represented. For another, the concerns shown do not always align with the kernel meaning of the aspect, but sometimes only with one small part thereof (e.g. Trades Union approach is concerned with juridical 'what is due' but only for a small section of stakeholders).

Therefore, we will explore how a framework for understanding ISD may be formulated beginning with the proposal that ISD is multi-aspectual functioning.

6.2 ISD AS MULTI-ASPECTUAL HUMAN ACTIVITY

It is tempting, because it is concerned with a technology and development, to jump to the conclusion that IS development is qualified by the formative aspect. But such an approach too quickly narrows the focus, robbing us of an everyday perspective. DST exhibits this tendency. A Dooyeweerdian approach that retains the lifeworld perspective is to begin not with a hidden normativity but with the very visible multi-aspectual human functioning that is ISD.

Aspectual normativity is as important in ISD as it was in human use of computers (chapter 4), but it is important in a different way. When considering HUC, the normativity of aspects was important to allow us to differentiate beneficial from detrimental impacts for purposes of evaluation. In ISD normativity is important as guidance for methodology.

6.2.1 Several Multi-aspectual Functionings

Following the approach in chapter 4, several multi-aspectual human functionings may be discerned in the lifeworld of ISD, which are interlaced with each other. Four will be considered:

- the overall ISD process
- anticipating usage: how possible use impacts on design of the IS and vice versa
- the creation and crafting of the technical artefact and its context of use
- elicitation of domain meaning.

Most ISD methods include these in some form. For example 'anticipating usage' is often narrowed to 'user requirements analysis', prior to design, implementation and testing, which constitute 'creation of artefact'. But Dooyeweerd leads us to emphasise things differently, and allows all four to take place in parallel. The advent of iterative methods has shown this is correct.

Hirschheim et. al.'s [1996] 'generations' may be seen as emphasising one or other of these. Unstructured methods tended to focus on creating the IS, then life-cycle, structured and prototyping approaches emphasised the overall ISD process, as did participatory approaches though it also focused on anticipating use. Sense-making, trades union and emancipatory approaches focused on anticipating use (including impacts on social structures in the latter two). Of the two additional approaches mentioned, KBS methodology focused on elicitation, and agile SDMs return the focus to creating the IS, though they also recognise the other three.

In most extant approaches, the first is seen as a whole of which the others are parts, but assuming a part-whole relationship tends to separate out the various functionings into sequential stages (as in the waterfall model), and deprives the supposed parts of meaning. But each of them can be meaningful outwith the overall ISD process (for example knowledge elicitation may be seen as analysis unconnected with ISD), and each has a different qualifying aspect. So the relationship between them is not part-whole but enkaptic, which is a structural relation among wholes.

The qualifying aspects of these are, respectively, the aesthetic with social (orchestrating the whole process), juridical (responsibility for how it can be used), formative with lingual, and analytic (identifying relevant concepts). The reasons are given below, as the four are examined in more detail.

6.3 ASPECTS OF OVERALL ISD PROCESS

Most of the aspects of the overall process of ISD will now be discussed. But in which order should they be considered? In ISD a number of people are involved, team members, participants or other stakeholders and they must be involved not as individuals but as a cohesive group, so the social aspect is of central importance. It is therefore useful to consider it first.

6.3.1 The Social Aspect

Hirschheim and Klein's [1994] Emancipatory ISD (EISD) is one approach in which the social aspect is central. Power relations is emphasised, but the lifeworld of ISD involves many social relations that are not based on power -- from 'social' events to helping each other out. To what extent should control be exercised, and to what extent should the team comprise unstructured social relations?

Dooyeweerd's [1986] theory of social institutions can help us understand and manage power relations among stakeholders in ISD. Intracommunal relationships, that is within a true social institution, he suggested, are dependent upon relationships of 'authority and subordination' [Kalsbeek, 1975,p.199-200], though this is not to be seen as coercive power so much as mutual recognition and assumption of role within the institution. But no such power structures should be imposed in the interaction between communities or individuals. As Kalsbeek [1975,p.200] points out, "we never find an authority

structure in intercommunal and interpersonal relationships."

The norm of the social aspect is respect, but what form should this 'respect' take? ISD involves intracommunal, intercommunal and interpersonal relationships, exhibited, respectively, in the need to make a cohesive team (and hence often a formal structure), the interests of the various external stakeholders and in the social friendships of all participants. Dooyeweerd suggested we should treat all three differently and beware of assuming the norms for one type carry over into others. This allows a valid but limited place for 'power relations': deference may only be expected in the first type -- and even this must be tempered by other aspects, especially the juridical and ethical. It is not appropriate in the others. It is not valid for one stakeholder's interest to subordinate that of another. However, there is also respect for special expertise, which may be seen as intercommunal in that the expert, as expert, is member of another community. Kalsbeek continued: "this does not deter certain people or groups or classes from exerting a considerable influence outside of their area of authority because they have special gifts or capital at their disposal." Table 6.3.1 summarises how the norm of respect is different for each type and might be useful in avoiding inappropriate social functionings in practice.

Table 6.3.1. Norms for different social relations in ISD

Type of Social Relationship	**Type of Respect**
Intracommunal	Authority + Deference, assumption of fole (tempered)
Intercommunal	Giving due importance to each interest and view Respect for expertise
Personal	Friendship, Consideration

6.3.2 Pre-social aspects

The lingual aspect of ISD overall is manifested in communication within the team and with all other stakeholders and also in such things as seeking information and reporting. Open dialogue, central to EISD and to Checkland's [1981] Soft Systems Methodology (SSM), is what this refers to. (The lingual aspect of the representation and coding of knowledge is of a different multi-aspectual functioning, dealt with below.)

The formative aspect is manifested here in planning of the ISD process and in the history of the project. The difference between formal structure (e.g. Waterfall model), or informal (iterative development), or 'pondering' the idea in unstructured development, is visible at the formative aspect. What is common to all approaches, however, is the primary manifestation of the formative aspect in creating the IS itself, which is discussed below.

Functioning in the analytic aspect involves making distinctions

relating to usage, which include such things as between who are stakeholders and who are not, between relevant and irrelevant issues, and between what the IS should be called upon to do and what it should not. These are all versions of what Midgley [2000] calls boundary critique in his Systemic Intervention. These three are found in SSM: the first relates to C (customer), the second to building the rich picture, and the third to T (transformation). In unstructured programming, analytic functioning generates the idea of what the IS is about. Perhaps one of the most important analytic functionings is the identification of secondary users, as discussed in chapter 4.

The sensitive-psychic aspect of ISD is manifested in the emotions and feelings of team members and other stakeholders, and the biotic is their health.

6.3.3 Post-social aspects

Of the post-social aspects, the economic is exhibited in management of the project, and in the limited resources of time, budget, expertise, personnel, access to participants, and the like, which the structured (waterfall) approach emphasises. It has received considerable attention in research and practice, so little more need be said here.

It was suggested that the aesthetic is the aspect that qualifies the ISD process as a whole. This ISD project may be likened to a symphony, the ISD team being the orchestra. There are main players -- strings, wind, percussion, brass sections -- with the occasional emphasis on one instrument type or another. Solo pieces come in from the outside, as it were. Each plays to the best of their ability (anticipating the juridical aspect) but each (anticipating the ethical) subordinating themselves to serving and supporting the others rather than seeking aggrandisement. Even the soloists serve to beautify the rest and make it exciting. The playing has a certain overall plan, but there is much improvisation, which is seen not as unfortunate deviation from plan but as making the whole even more meaningful. And yet improvisation is the minor element within the context of the major plan, and is done with economy. Moreover, the symphony does not stand alone as a masterpiece, but beautifies and enhances its surrounding pieces, bringing out their own beauty and integrity.

One would never outsource the playing of sections of the piece! It unduly elevates the economic aspect in ISD. One of the strengths of the original unstructured approach is that, when done well, it facilitates an aesthetic holism. (But when done badly, it is destructive.)

The juridical aspect is important, not just because of implied or actual contracts of delivery between the parties, but more importantly because of responsibility for repercussions when the system is in use, and thus towards all stakeholders, whether acknowledged or not. This responsibility is the main reason why ISD should be orientated so as always to critically and sensitively anticipate usage.

Dooyeweerd's ethical aspect is specifically centred on self-giving -- in contrast to traditional approaches to ethics -- implying a norm of

generosity. This is recognised in Gerald Weinberg's [1999] notion of egoless programming.

The pistic aspect is concerned with faith, loyalty and vision of who we are as part of the ISD project. It is manifested, for example, in treating some stakeholders and issues 'sacred' and others 'profane' [Midgley, 2000]. But it is more centrally manifested in our commitment to a vision of what the meaning of the IS is that we are developing, what eventual usage will mean, and in loyalty to the project. This may be different for different stakeholders, who hold deeply-held beliefs and perspectives that Checkland [1981] calls (Weltanschauungen).

Variety in such deep, pistically-held perspective can result in conflict. The more intractable conflicts arise from undue absolutization either of some concrete thing, such as one's own preferred solution, or an aspect. Such conflicts cannot be resolved by open dialogue, partly because their holding is pistic functioning, and partly because each aspect provides a distinct rationality (§3.1.5). Habermas' notion of ideal dialogue, which is at the centre of EISD, perhaps proves less than useful to us, even as a counterfactual ideal, because there is no logical link between what is sense in different aspects. Holding disparate visions together involves pistic functioning, which itself depends, if Dooyeweerd is correct, on good aesthetic and ethical functioning. Assuming such an attitude, the Dooyeweerdian notion of aspectually centred perspectives, as discussed in §3.3.8, can help.

6.3.4 Aspectually-centred Perspectives

Table 6.3.4 Aspectually-centred perspectives and roles

Aspect	**Emphasis (e.g.)**	**Role (e.g.)**
Quant'ive	Number	Accountant
Spatial	Distance, layout	Geographer, draftsman
Kinematic	Movement	Transport planning
Physical	Energy, forces	Energy analyst
Biotic	Health	Nurse, Doctor
Psychic	Emotion, sensory-motor functions	Psychologist
Analytic	Logic, Analysis	Analyst
Formative	Achievement, Power, History	Planner
Lingual	Communication, documentation	Communicator, External relations
Social	Social activity, Relationships	Host, Group therapist
Economic	Resources, frugality	Manager, Economist
Aesthetic	Harmony, Fun	Coordinator, Clown
Juridical	Due, Contracts	Lawyer
Ethical	Generosity	Charity worker
Pistic	Vision, Loyalty, Identity Religious activity	'Champion', Padre

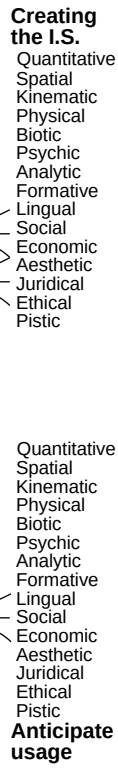
Many perspectives or Weltanschauungen centre on one aspect and these are often associated with the holder's role in the project or organisation. Assuming the above attitude it can often be useful to surface them. Table 6.3.4 gives examples of what might be emphasised under each, as a factor of great importance and as a lens through which to see the world, together with associated organisational roles.

6.3.5 All Aspects Together

These aspects of the overall ISD process are summarised in Table 6.3.5. One of the benefits of seeing ISD in this way is that it gives a good picture of what might indicate high quality ISD.

Table 6.3.5. Aspects of the ISD Process as a Whole

Aspect	.. of overall ISD process	ISD Approach
Biotic / Organic	Health of members of development team.	
Psychic / Sensitive	Pain, pleasure experienced by the ISD team	
Analytic	Clear objectives, goal.	SI boundary ctq
Formative	Planning, History. Development of artefact and context of use.	Iterative development
Lingual	Knowledge representation. Documentation, Archives. Open dialogue. Seeking information.	EISD, SSM
Social	Respect for views. Authority in the ISD team.	Structure of team
Economic	Budget, deadlines. Expertise, access. Technical limitations.	Structured development
Aesthetic	Orchestrating Fun	Unstructured development
Juridical	Contract to deliver. Responsibility.	CMMI. CCA, EISD
Ethical	Attitude of self-giving.	Egoless programming
	Vision for project. Loyalty to project.	SSM *Weltanschauung* SI 'sacred'



Many approaches to ISD are mentioned in the table, as giving emphasis to certain aspects -- such as unstructured development as practised by early programmers, structured development, the Waterfall Model and organisational versions like CMMI, iterative development and Boehm's [1988] Spiral Model, Gerald Weinberg's [1999] egoless programming, and Beck's [2000] extreme programming and other 'agile' methods, Basden, Watson and Brandon's [1995] Client Centred Approach (CCA), Checkland's [1981] Soft Systems Methodology (SSM), Jackson's [1991] Critical Systems Thinking, Ulrich's [1994] Critical Heuristics, Hirschheim

Pistic

and Klein's [1994] Emancipatory ISD (EISD), Avison and Wood-Harper's [1990] Multiview, and more). Seldom are the latter mentioned alongside the former, but this approach brings all together into a single picture.

The aspectual approach to normativity, adoption of a suite of aspects like Dooyeweerd's, and the shalom principle give us a precise and yet flexible basis on which to understand the quality of the ISD process. According to this, ISD projects will tend to run well to the extent that those involved fulfil the underlying norms of all the aspects. Seriously ignoring any of the norms can jeopardise the overall success of the project. But it must be reiterated that since aspectual norms can never be fully known by theoretical thought, nor even defined, it is dangerous to try to make them into a method or set of rules. But Table 6.3.5 can help point to the kinds of normative issues that demand attention during an ISD project.

The validity of this multi-aspectual approach can be neither proved nor disproved scientifically because it concerns the lifeworld of ISD. However, support for it may be found in the records or the honest recollections of many ISD projects. In the author's experience of ISD project failure or difficulty:

- In one project, a key player left half way through, and the project could never recover (pistic loyalty).
- In many student group projects there is lack of commitment (pistic).
- In another, a competitive, self-seeking attitude pervaded the team (ethical).
- In yet another, insufficient attention was given to the core knowledge and too much to the usability features, so the knowledge base was never completed (juridical, aesthetic).
- Yet another project was too fragmented (aesthetic).
- Many projects exceed time or cost budgets (economic).
- Another project was jeopardised by animosity between team members (social).
- Some projects fail technically (formative).
- Some projects lack clear objectives (analytic).

In such an analysis, are we merely filling slots? This common self-criticism in Dooyeweerdian application is discussed in chapter 9. But even if it is, using the aspects as slots to fill can provide useful methodological guidance in ISD because it stimulates the developer to think of things often overlooked -- as long as the suite of aspects used broadly covers the wide range of issues in the lifeworld of ISD. This is precisely what is claimed for Dooyeweerd's suite.

6.4 ANTICIPATING USE

Anticipating use of the IS goes beyond 'user requirements analysis' in three ways. First, it does not assume that finding out what users believe they want is what will bring increased shalom in practice; it is well known that what brings real benefit is often other than that which users express in their wish-lists. Second, it recognises that IS use (especially HLC) involves all aspects, not just those of which users

might be aware during analysis. Third, there must be a need for creative change of work and other practices as a result of introducing the IS, and these changes must be anticipated.

Whereas in use, the focus is mainly on present and past actuality, in ISD we look forward, anticipating future possibilities for such usage, which guide the creative process of design and development. Thinking philosophically, the notion of law-based functioning gave Dooyeweerd a means of understanding anticipation of future possibility [Dooyeweerd, 1984,I,p.105]:

> "Everything that has real existence, has many more potentialities than are actualized. Potentiality itself resides in the factual subject-side [entity-side]; its principle, on the contrary, in the cosmonomic-side [law-side] of time. The factual subject-side is always connected with individuality (actual as well as potential), which can never be reduced to a general rule. But it remains bound to its structural laws, which determine its margin or latitude of possibilities."

This latitude of possibilities is governed by the normativity of the aspects, each aspect providing latitude of a different kind.

Whereas development itself might be formative, possibility implied responsibility. That is, the IS developers are responsible for what they develop and, though never solely so, for the uses to which the IS is put and the impacts thereof because these are influenced by its design. For this reason we may see the qualifying aspect of anticipating use as the juridical.

To fulfil this juridical norm every stakeholder should be identified and involved in the ISD process as far as possible. Both the Client Centred Approach (CCA) [Basden, Watson and Brandon, 1995] Midgley's [2000] Systemic Intervention emphasises this aspect. But neither give guidance on how to ensure all the stakeholders are identified. Dooyeweerd's suite of aspects can, however, help us do so very practically.

- Asking of each aspect "What roles are there connected with this aspect in the situation of use?" will identify those who should participate; Table 6.3.4. can help.

- Asking "What repercussions will use of the system have in this aspect, and on what or whom will it have repercussions?" can help identify other stakeholders who would be affected even though they have no role connected with it -- such as animals, the public, society, environment. A host of likely repercussions can be exposed, and thus help distinguish the important from the trivial.

In practice, it can be useful to consider the three types of usage we identified in chapter 4, HLC, ERC, HCI. Here is a selection of questions that developer might ask, mainly geared to HLC.

Analytic aspect: HLC: When they use our artefact, users will make all kinds of distinctions relevant to their living or work that we have not considered. To do so appropriately they must understand it clearly: build in transparency so they can understand the structures of

the system and how it works.

Formative: HLC: Will this be used for purposes we cannot envisage (c.f. Elsie in chapter 4), because their context is not ours? So build in flexibility and robustness.

Lingual: HLC: Will users be able to explain or communicate better to their colleagues, or keep more salient records because of the IS we deliver? HCI and ERC: Users must interpret what we put on screen and give input. To ensure they will understand what we intend, we must carefully design wording and explanations.)

Social: HLC: Will use of this make people less or more socially active in healthy ways? HCI: Cultural connotations may cause problems if it will be used globally (e.g. a web page).

Economic: HLC: As a result of using this, will the way resources are managed (e.g. raw materials, paper, users' time) change? Ensure this always tends towards more frugality rather than waste or superfluity. Remember climate change: not only energy consumed by a computer left on for hours when not used, but will use of it result in more flights, road journeys, etc.? Though SSM [Checkland, 1981] deals with 'environmental constraints' it does not adequately recognise the flexible responsibility involved; see later.

Aesthetic: HLC: Users should find all they do with your system interesting and stimulating, and the IS should fit harmoniously with its lifeworld context(s) of use, yet provoking new thinking therein.

Juridical: HLC: Ensure that, as a result of using this system, all stakeholders will be given their due. This is more important even than ensuring adherence to national and international law, which may be seen as (deficient) attempts to define 'due'. The notion of emancipation in Hirschheim and Klein's [1994] EISD is mainly juridical.

Ethical: HLC: Will use of our system make users more selfish and competitive or more self-giving, generous, collaborative? This was one reason why Elsie (chapter 4) was a success. ERC: Will users come to 'love' the IS?

Pistic: HLC: It might surprise us how important is the pistic aspect. Churchman [1971] suggested:

> "If we look at faith from the design point of view, we ask whether a faithful inquiring system is better than a faithless one. No matter how slight the chance, the gambler must in some sense have faith in the one possibility that is favourable to him." [p.240]

The 'gambler' may be the IS developer. But they can also be the user, who is using the IS in new ways. ERC: Will its users and all other stakeholders trust it, or might it let them down? HLC: Is the use of this system in line with the vision of the users and their organisation? If so, is that vision appropriate? When using it, will users be stimulated towards questioning deficient visions? Will 'good faith' be encouraged or will it be hindered (see §4.6.2 and Walsham

[2001])?

There is much more to be said in each aspect, and the reader is left to fill the gaps. Many of these impacts in HLC will be unexpected, indirect or long-term, and it is the IS developer's responsibility to design for these as far as possible. Of course, many such impacts cannot be predicted, but an aspectual analysis can help indicate the general kinds of issues that might have been overlooked.

One example: As developer of the Wheat Counsellor system [Jones and Crates, 1985], which advised farmers on use of fungicides, the author took on himself a responsibility of a pistic nature. Realising that British agriculture was then too heavily dependent on chemicals, and that the pendulum would soon begin to swing away from this state, he probed the source agronomists for their expertise about what they would advise if a farmer wished to use fewer chemicals. After some initial resistance, they divulged the advice quite readily, and Wheat Counsellor became not only a better product but one that more trusted. (This is why, in Table 6.1.3, he had a bit of a Warrior role.)

Functionalistic 'user requirements analysis' is a pale shadow of such multi-aspectual analysis anticipating usage, and no prior specification can every adequately capture its richness. But neither can the usual spiral or 'iterative' methods. Therefore the IS developer has a (God-given?) responsibility to ensure they are sensitive to a wide range of aspectual issues. This is perhaps what DST is aiming at, and it is a pity that the use of Dooyeweerd's aspects is eschewed. It is certainly what good developers already do, and what this author had been trying to achieve in most of his ISD projects listed in Table 6.1.3, most of which were undertaken long before he discovered the great benefit of Dooyeweerd's suite of aspects.

6.5 ASPECTS OF CREATING THE IS

Creating the IS is not just about constructing a technical artefact or system. Creating the IS involves shaping both the technical artefact to be used, the user's knowledge and the human and organisational context of use (see Structure of ERC in chapter 4).

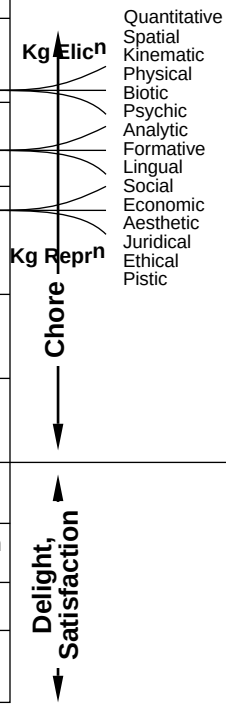
That both knowledge in the artefact and that instilled in the context of use are important is indicated by the Dooyeweerdian belief, expounded in chapters 4 and 5, that the knowledge represented 'in' the computer is nothing without the user's subject-functioning, it is the user's knowledge which is the more important. This echoes West's [1992] view that, even in the case of expert systems technology, in which the knowledge represented in the artefact reaches its peak of sophistication, "it may not be necessary to convert this information ... for inclusion within the expert system" but to rely on it being active in the users. This involves training users -- if you like, domain knowledge must be 'put into' both the computer system and the users!

6.5.1 Aspects of Creating the IS

The shaping of both is a multi-aspectual human activity. Though the aspects of ISD overall and anticipating usage were each discussed, here only a few indicative examples are given, of aspects of artefact creation and of shaping the context of use. Table 6.5.1 shows these; the knowledgeable reader is encouraged to criticise this and make their own more detailed analysis.

Table 6.5.1 Aspects of creating the information system

Aspect	... of shaping context	.. of creating artefact
Analytic	Who may and may not use it; What they must know	Clarifying concepts, Relevant v. irrelevant knowledge, concepts.
Formative	Purpose it serves (HLC) Planning the context	Structuring data, Designing algorithms, Getting it working.
Lingual	Training users	Representing knowledge as the program. Writing documentation, tutorials, help systems, etc.
Social	Organisational structures.	Relationship with all involved in ISD, esp. domain experts.
Economic	Management of IS use.	Program efficiency, Use of limited screen area. Scaling up.
Aesthetic	Making use enjpyable. Artefact fits context.	Style of UI. Beauty of program,
Juridical	Ensuring use is appropriate to context.	Doing justice to information and knowledge.
Ethical		Loving the program.
Pistic	Shaping visions of users.	Impact of religious views on how we program.



The main (qualifying) aspect of creating the IS is the formative -- planning and shaping both the artefact and its context of use. The ISD community as a whole has considerable experience of the analytic to lingual aspects, and books on, and methodologies about, these abound. This is called the chore, which is distinguished, perhaps rather unfairly, from the delight that of creating IS.

Creating the IS links strongly with knowledge elicitation and representation, via the analytic, formative and lingual aspects as shown, and this is explored below, and further in chapter 7.

6.5.2 The 'Chores' of Creating the IS

It is well known that the task of creating the IS is hard work, and most methodologies designed to guide it concern themselves with the analytic to economic aspects in varying degrees. Central, of course, is the discipline of actual programming (or database creation, OO

class creation, etc.) -- good program structure, meaningful variable names, comments that explain 'why' as well as 'what; this is all covered by the cosmic normativity of the lingual aspect. But it has long been recognised that many other aspects are also relevant, such as the economic aspect of keeping to time, the formative aspect of planning the IS, and so on.

'Agile' SDMs have returned the focus to IS creation, and each seems to address a particular problem. For example the Crystal series of ASDMs [Cockburn, 2005] is aimed at overcoming barriers to communication, while Extreme Programming [Beck, 2000] has the slogan "Embrace change". Most ASDMs, however, seem to be attuned to the everyday life of programming. Table 6.5.2, shows that the 15 principles that support XP's values cover a wide range of the human aspects. ('Embracing change' was difficult to assign to an aspect. The most obvious aspect is the formative (deliberate shaping) but this principle expresses rather the welcoming of change that occurs, hence the aesthetic aspect of harmony. It might have been the ethical aspect (self-giving: willingness to bend to the will of others), which is otherwise missing.) Such an aspectual analysis of an SDM can indicate the degree to which it is open to everyday reality.

Table 6.5.2 Aspects emphasised in various agile methods

XP Principle	Aspect
Assuming simplicity	Aesthetic
Incremental change	Formative
Embracing change	Aesthetic
Quality work	Juridical
Teaching learning	Lingual
Small initial investments	Economic
Playing to win	Pistic
Concrete experimentation	Formative
Open, honest communication	Lingual
Working with people's instincts	Social
Accepting responsibility	Juridical
Local adaptation	Social
Travelling light	Economic
Honest measurement	Juridical
Rapid feedback	Lingual

6.5.3 The Delight that is Creating IS

But other aspects are also important, including some rather surprising ones. Yet, in Things a Computer Scientist Rarely Speaks Talks About, Donald Knuth [2001], the designer of TeX, recalls [p.130]:

> "I got hold of a program from IBM called SOAP, written by Stan Poley. That program was absolutely beautiful. Reading it was just like hearing a symphony, because every instruction was sort of doing two things and everything came together gracefully. I also read the code of a compiler that was written by ...: that code was plodding and excruciating to read, because it just didn't possess any wit whatsoever. It got the job done, but its use of the computer was very disappointing. So I was encouraged to rewrite that program in a way that would approach the style of Stan Poley. In fact, that's how I got into software."

What Knuth refers to is the aesthetic aspect of programming. Pacey [1996,p.80-81] likewise refers to the "existential joy" in technology. One can 'love' a program, giving oneself for it (ethical aspect), do justice to all the knowledge and information represented in it (juridical), and Knuth is also quite open about how his religious beliefs have impacted his programming. What is it that makes creation of the artefact satisfying and a delight, rather than a chore? (I am not aware that this question has ever been addressed.) It is these four post-economic aspects -- which, sadly, receive little attention. But seeing creation of the IS as multi-aspectual, in which the shalom principle is important, can help ensure such issues are given due, but not undue, importance.

6.6 KNOWLEDGE ELICITATION AND REPRESENTATION

The terms 'knowledge elicitation' and 'knowledge representation' are taken from the KBS community, though they are equivalent to analysis and design, implementation and testing in structured SDMs. But 'elicitation' and 'representation' are used here because it was there they reached their pinnacle of sophistication and responsiveness to the lifeworld of the application, because in KBS it was important to elicit and represent very complex, highly nuanced, meaning of the domain, and current methods owe a lot to KBS research and practice. Knowledge elicitation is concerned with identifying relevant 'knowledge' of the domain and conceptualizing it, and knowledge representation is concerned with expressing that in computer-readable symbolic form.

Knowledge elicitation has traditionally been seen in mining metaphors: extracting nuggets of 'knowledge' from those who are expert in the domain, which are subsequently represented in propositional form. But this metaphor was critiqued early in the early 1990s on the grounds that some knowledge is un-extractable, some takes the form of 'stuff' rather than nuggets, while yet other knowledge is generated by the very process of elicitation and representation (see Basden and Hibberd [1996] for discussion of this). Fiol and Huff [1992], and many others, found that the very act of representation stimulates elicitation rather than being merely a mechanical expression of what has been elicited.

But under Dooyeweerd, to think of knowledge as either extractable or un-extractable, or nuggets or stuff, is misleading. Rather, it is better to focus on the multi-aspectual human activity that

is knowledge elicitation-and-representation (KER). It concerns itself with cosmic and concrete meaning relevant to the domain of application, involving the analytic aspect of distinguishing what is relevant meaning from what is less so, and conceptualizing it, formative aspect, of structuring and relating it, and lingual aspect, of expressing it in a computer-readable knowledge representation language. These are the arrows shown in Table 6.5.1. The type of language is discussed in chapter 7. The qualifying aspect of elicitation on its own is the analytic, and that of representation is the lingual, but it is better to see all three as intertwined.

A benefit of this view is that it covers many eventualities, including ill-structured domains of application and fictitious domains (such as virtual reality), and also gives due recognition to other aspects of the process.

The social aspect is important because knowledge elicitation from people involves forming relationships of trust and friendship with those who hold the necessary knowledge, otherwise knowledge might be withheld. The other aspects contribute, such as the juridical (do justice to the meaning of the domain), economic (time limits), aesthetic (harmony of the meaning represented), etc.

6.6.1 Doing Justice to Domain Meaning

In chapter 4 it was shown how the norm that should guide ERC (engagement with represented content) is that of doing justice to domain meaning. One technique that has emerged to assist this is the domain ontology, a computer-processible statement of, as Guarino, Masolo and Verere [1999], put it, "a set of things whose existence is acknowledged by a particular theory or system of thought." But such a definition presupposes such a theory, which precludes a lifeworld attitude towards knowledge elicitation. Perhaps better is Gruber's [1995] definition, "a formal explicit specification of a shared conceptualization" because while it acknowledges the need for formal specification (readable by computer as a data model) it seems to leave open the possibility of this attempting to represent the lifeworld of the domain. But there is a hidden presupposition, that it is clear what a shared conceptualization is or that it is straightforward for the developer to know it.

Because (in general) the IS will be part of the everyday life of the user, the knowledge that is represented in it must relate naturally to everyday use. It is thus multi-aspectual. This means that the analytic aspect, formative and lingual aspects of KER must reach out to all the aspects, as already mentioned and indicated in Table 6.5.1. The knowledge the developer seeks is neither 'objective' nor purely personal (so-called subjective) because Dooyeweerd goes beyond both (see §3.3.2). The challenge is, therefore, to ensure:

- that every relevant aspect of the domain is recognised,
- that each of these aspects is appropriately understood in terms of its kernel meaning, and
- that all relevant concepts within each aspect are elicited (including things and their types, properties, constraints,

relationships, operations, and so on).

The first two refer to the law side of reality, the third to the entity or fact side. Traditionally, only the third has been recognised, leading to the erroneous notion of 'complete and accurate knowledge'.

6.6.2 Virtual Reality

In the case of systems that provide the users with a virtual world, however, these challenges must be modified, because the virtual world might be imaginary rather than real, especially in computer games. As discussed in chapter 4 the quality of virtual world is governed more by its (virtual, represented) law side than by its entity side, so the third challenge above is even less important in relation to the other two. Table 4.3.2 showed how the virtual world in the game ZAngband invokes nearly every aspect; all these must be represented explicitly to form that software.

The virtual environment community differentiates believability from plausibility. Both refer to the virtual law side that is the program. Believability may be seen as arising from faithful representation of the laws of all aspects (e.g. there is gravity, social grouping), and plausibility as minor modification of these laws (e.g. gravity strength varies with time), enabling the exploration of interesting fictional possibilities.

6.6.3 Everyday Experience and Understanding

Knowledge elicitation has undergone two shifts in emphasis. The first was from computer models, which modelled theories (especially physical), but in the 1970s it became clear that such 'book knowledge' could not form the basis of KBS (then called 'expert systems') that could provide useful advice or solve real-life problems. So the possibility of encapsulating actual problem-solving experience, as rules of thumb or heuristic rules, was explored. Two classic KBSs of this era were MYCIN [Shortliffe, 1976] and Prospector [Hart and Duda, 1977].

But basing a knowledge base on experience led to 'brittle' systems, whose knowledge bases were opaque and for which the explanation facilities were poor. The author's own experience, mentioned in Vignette 3 of the Preface, for example, recommended that what they called 'understanding' should be sought instead. It could be extracted from experience by separating out context-dependent problem-solving knowledge [Attarwala and Basden, 1985]. 'Understanding' was very definitely not a return to theoretical knowledge, but included 'everyday' understanding. Incorporating everyday understanding was the second step, and it became known as 'second generation expert systems' by Steels [1985] and Weilinga and Breuker [1986]. Pat Hayes' [1978], [1985] Naive Physics Manifesto was an early visionary contributor to this movement. Systems incorporating such understanding would 'degrade gracefully'.

But, at the time, neither Steels nor this author had any

philosophical justification for believing there was 'understanding' to be sought and separated; it just seemed to work. However, it may now be justified by Dooyeweerd's distinction between law- and entity-side knowledge. Heuristics, as expressions of concrete experience, are entity-side knowledge and therefore cannot express that which is generally so, while understanding is law-side and can do so.

If this is so, then there is a different type of understanding for each aspect. So-called causal nets were devised at the time, on the assumption of a universal causality, but lack of insight into the difference between kernel causality (physical aspect) and its analogical echoes (§3.1.4) led to problems. What this implies, and will be explored in chapter 7, is that it may be worthwhile to prepare a knowledge representation facility in which the different type of understanding in each of the aspects is made available.

6.6.4 Tacit and Explicit Knowledge

The issue of tacit knowledge was first highlighted by Collins [1974], as a problem for knowledge elicitation. 'Knowledge' is conveyed lingually, but speaker and hearer might assume subtly different meanings in the words used; Collins' example is what 'short' means in reference to a wire in an early laser. During the 1980s and 1990s especially interest in tacit knowledge mushroomed, first because of the challenges in building KBS, then because of misunderstandings of knowledge stored in organisational repositories. Such misunderstandings can be both hidden, because each assumes that others understand things in the same way as they do, and dangerous, because the stored knowledge can mislead without any warning.

The challenge is to make tacit knowledge explicit. Polanyi's monograph, The Tacit Dimension [1967], has been a core reference in this discourse, but he maintained that tacit knowledge can never be made explicit; most of his examples are of sensory-motor knowledge. But many, including the author of this work, have found otherwise for much taken-for-granted conceptual knowledge; such knowledge, long forgotten and now apparently second-nature in operation, can often be made explicit by judicious interview techniques, such as Winfield's [2000] MAKE. Meanwhile, Baumard [1999] discusses 'tacit knowledge in organisations', which takes on a metaphorical meaning: knowledge can be 'tacit' to the organisation-treated-as-person because it is not recorded, even though someone in the organisation might have the knowledge required in explicit form.

The Dooyeweerdian notion of multi-aspectual knowing can provide some insight to distinguish several reasons why knowledge might be 'tacit':

Intuitive grasp of aspectual meaning (§3.1.4, §3.3.4) fundamentally cannot be explicated fully, so it might be more fruitful to rely on its influence on user's activity, by judicious design of the artefact, rather than to try to represent it explicitly therein.

The different aspectual ways of knowing implies tacit knowing in several ways. Knowing in the pre-analytic aspects (e.g. our experience of colour, sound or muscular feeling, our knowledge of how to ride a bicycle) is likely not to be explicated without significant distortion, because such things are continuous rather than crisply distinguishable. Therefore it is usually unwise to attempt precise and full description of this, and wiser to capitalize on the user's functioning.

The non-absoluteness of the lingual aspect implies that language can never fully carry the meaning intended, so misunderstandings can arise.

However, the lingual aspect does have considerable power to carry meaning, and a lot can be explicated usefully. Such knowledge is tacit by virtue of being taken for granted, which is often due to cultural reasons and is often possible to explicate to some degree by stimulating the expert's memory, either by listening to stories or by seeking understanding as Attarwala and Basden's [1985] approach sought to do. In Collins' [1974] example what 'short' means was misunderstood in this way, but eventually explicated.

Baumard's [1999] 'tacit knowledge in organisations' is socially rather than lingually or analytically tacit and hence might be explicated by functioning in the social rather than merely lingual aspect, which encourages individuals holding the required knowledge to release it into the public space.

6.6.5 Limits to Knowing

Just as none of our aspectual functioning is absolute (see §3.1.4), so none of our knowing can be absolute either. We cannot, and should not, ever hope for 'truth' in the way objectivism has conceived it. Dooyeweerd very explicitly stated that, even in the ideal, "There is no truth in itself" [Dooyeweerd, 1984,III,p.577] whereas subjectivism reacted against this by denying the notion of truth, Dooyeweerd denied 'in itself'; i.e. self-dependent and able to stand as truth without reference to anything else. It is not that there is no truth, but there is no truth 'in itself'. "Hypostatized 'truth' is a lie," he said [Dooyeweerd, 1984,II,p.561], "there is no selfsufficient partial truth," and so "2 x 2 = 4 becomes an untruth if it is absolutized into a truth in itself" [Dooyeweerd, 1984,II,p.572]. This means that "our insight is fallible" [Dooyeweerd, 1984,II,p.574]. The root of the non-absoluteness lies not in any imperfection in our knowing-functioning but that all meaning refers beyond itself to its Origin. This has implications for knowledge elicitation in that one should always be cautious about the meaning represented in the computer.

But, we should not despair of thought altogether. Dooyeweerd said [1984,II,p.556] that thought "remains bound to a modal horizon which has a constant determining character as to all the changing concrete facts." In other words, thought is 'friendly' to the Cosmos

rather than inimical to it. Reality, he believed, has a tendency to reveal rather than hide itself, to inform us rather than mislead us. This belief sounds unfamiliar to Western ears that are used to what Tarnas [1991,p.366] has called 'secular skepticism' by which we have long believed there is absolutely no connection between knowing and known, and our capacity to be misled is, consequently, infinite. But that is a presupposition as to what the relationship is between thought and thing, and Dooyeweerd happens to have made a different presupposition, and these led him to acknowledge the ability of intuition to grasp cosmic meaning more fully than analytic thinking can.

What this implies for ISD is that, on the one hand, we must always be cautious about the claims we make for elicited knowledge, but on the other hand we should not despise our intuitive grasp of domain meaning during knowledge elicitation. Rather, we should allow it to speak to us about the domain of application and explore what it tells us.

6.7 PRACTICAL DEVICES

6.7.1 Aspectual Analysis

As in chapter 4, aspectual analysis is useful. Here it has been used to ensure quality (shalom) of the overall ISD process and creation of the IS and to stimulate the consideration of possibilities in anticipating usage and identifying responsibilities. But it is perhaps less useful in those simple forms for knowledge elicitation. Instead, for this, a 'multi-aspectual knowledge elicitation' method has been developed.

6.7.2 Multi-Aspectual Knowledge Elicitation: MAKE

Winfield [2000] has done some very interesting work, in devising a sophisticated methodology for analysis of domain knowledge, that is centred on Dooyeweerd's notion of aspects: Multi-Aspectual Knowledge Elicitation (MAKE). MAKE combines the identification of aspects of an application with more detailed analysis of the individual issues, concepts, laws, etc. that are important for the application, and is useful for generating domain ontologies and for making some types of tacit knowledge explicit.

His approach is to start by asking the experts in the application domain to identify a couple of the aspects they deem most important, and to grow a recognition of the relevance of others aspects by a gradual process. The participants first identify a few whole aspects, then they start to identify concepts and laws etc. within aspects. In the process, concepts come to mind that do not fit well within currently identified aspects, and the participants are thus led in a very natural manner to identify other relevant aspects. The steps of the MAKE process may be seen as:

1. Introduction (e.g. explanation of kernel meanings of aspects, and obtain statement of requirements)
2. Identify a few (e.g. a couple) important aspects.

3. Focus on one of these aspects and specify any laws, axioms, data, definitions, and constraints that apply to the domain.
4. Identify as many concepts as possible that lie in this aspect. (Note: May need to check the concepts at a later stage to ensure they fall within the correct aspect.)
5. Apply Low Level Abstraction to each concept, which needs, or is thought to need exploding.
6. Repeat steps 3-6 as necessary.
7. Use the aspectual template to identify any new aspects, which may apply to the concepts specified but (build bridges between concepts and aspects), and return to step 3.

Low Level Abstraction was a concept that Winfield developed from the 1991 edition of Clouser [2005] and refers to becoming aware of the various aspectual properties of things yet without isolating them from the things. An aspect map is drawn as the analysis proceeds, an example shown in Fig. 6.5.2.

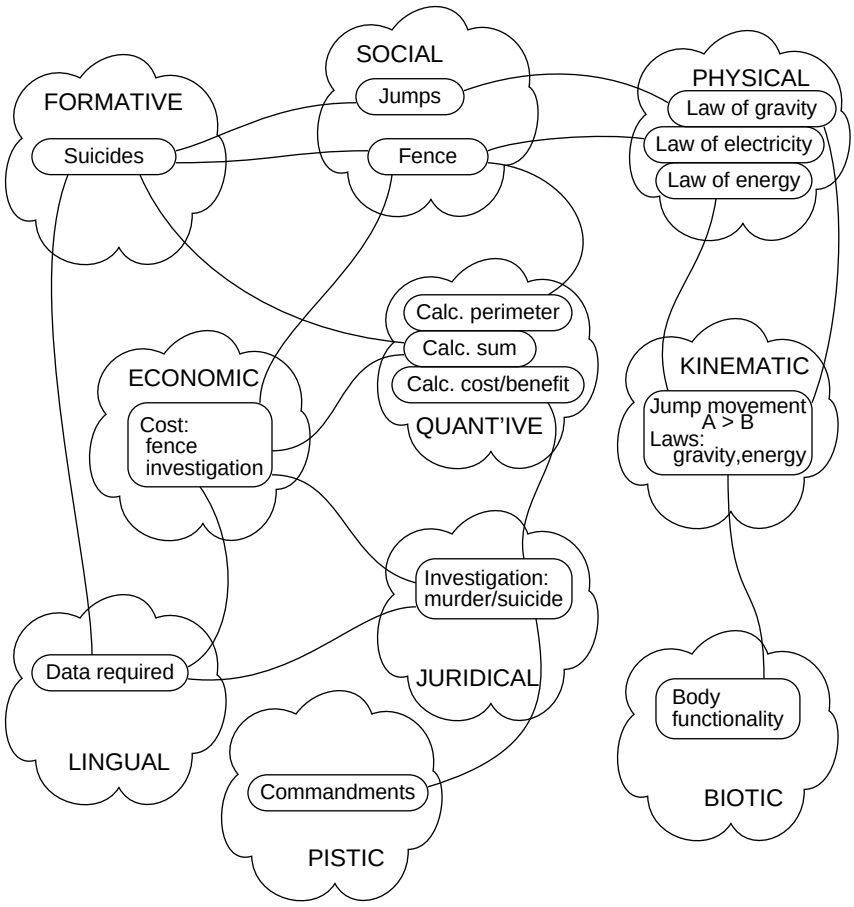


Figure 6.5.2. Aspectual map generated in Multi-Aspectual Knowledge Elicitation

Winfield refined and tested MAKE on eight case studies, mostly with participants who had never heard of Dooyeweerd or the aspects before, and found consistently that MAKE was easy to learn (both by experts and Winfield's students who carried out some of the case studies) it was not difficult for participants to grasp enough

understanding of the aspects in order to undertake this process, that nearly every aspect (typically 13 out of 15) was identified in each of the case studies, that it is useful over a wide variety of domains (tree planting, sustainability, vetinary practice, Islamic food laws, youth advice and management of a local housing business unit).

6.7.3 Characteristics of MAKE

MAKE has a flexible structure that places the interviewed expert very much at the heart of the knowledge elicitation process, and works even when participants understand aspectual meaning differently from the way Dooyeweerd did. It was found that MAKE improved on current methods of knowledge elicitation in a number of ways, by:

- improving the breadth of knowledge elicited,
- aiding the elicitation of (certain types of) tacit knowledge,
- facilitating multiple views of knowledge,
- encouraging reflection by the expert,
- eliciting the underlying 'theories' relating to the domain.

If the aspectual law-and-meaning pertains across all cultures, as claimed in §3.1.4, and to the extent that participants' intuitive grasp of aspectual meaning is not too distorted, then MAKE might be particularly useful in inter-cultural or cross-cultural analysis. The interviewee in the case of Islamic food laws said he found it extremely useful.

One general benefit of Dooyeweerd's approach based on law rather than entity is that it is more tolerant of novel ideas. For example, in considering virtual organisations, an entity-centred perspective would see that all organisations (as entities) have involved the spatial aspect, suggesting that virtual organisations could not work, whereas if we considered the aspects, we can see that the social and spatial aspects are distinct and thus virtual organisations might work.

MAKE is a simple and effective method for analysing a domain. But we can also see an example of how a method of analysis can arise directly out of a Dooyeweerdian framework for understanding knowledge domains. The use of MAKE has confirmed empirically a number of things:

- that applications knowledge is multi-aspectual,
- that the aspects are distinct and thus deliberately thought about,
- that most aspects can expect to be relevant,
- that each aspect gives rise to concepts, known laws, axioms, constraints, and the like,
- that these are all interconnected, that such interconnections include connections with other aspects (e.g. inter-aspectual echoes, and
- that the kernel meanings of the aspects may be intuitively grasped.

We cannot say that MAKE was derived deductively from the

framework, but rather that it was inspired by a knowledge of Dooyeweerd. The development of MAKE exemplifies how I anticipate those working in the various of areas of information systems to be able to develop methods, theories, classifications and the like on the basis of frameworks for understanding inspired by Dooyeweerdian philosophy.

6.8 ENRICHING SSM

This section shows how Dooyeweerd has been used by several thinkers to engage with an existing framework for understanding ISD. The framework is Checkland's [1981] soft systems methodology (SSM). This discussion should not be seen as a full critique of SSM, but rather as a demonstration of some ways in which Dooyeweerd's thought can engage with existing frameworks, not to supplant them but to support, underpin and enrich them.

Whereas SST, CST, MST and DST have presented themselves as replacing or absorbing earlier paradigms, several thinkers have attempted to enrich Checkland's [1981] SSM using Dooyeweerd explicitly, in effect transplanting it as it is or wishes to be from the Nature-Freedom to the Creation, Fall, Redemption Ground-Motive.

Bergvall-Kåreborn [2001] characterizes the aim of SSM as "to improve real-world situations by orchestrating changes of appreciation through a cyclic learning process." This speaks of a diversity of norms, relationships and perspectives among those involved in, or affected by, the situation. Such diversity must be not only acknowledged but 'orchestrated,' so that all those involved or affected may receive their due, and do so in a coherent way. 'Changes' speaks of dynamically bringing about new insights into the situation rather than merely gathering existing perspectives together. SSM has four stages: of finding out about a situation that needs to be improved, modelling, comparison and taking action, all usually undertaken by a group of participants. Finding out yields a Rich Picture, Modelling yields Root Definitions and Conceptual Models of systems that might effect improvements, comparison is between these these models and the real situation, and taking action should result. Checkland [1981,p.224-5] suggested that a good Root Definition and Conceptual Model will specify at least six elements:

C - customers: "beneficiaries or victims affected by the system's activities"
A - actors: "agents who carry out, or cause to be carried out, the main activities of the system, especially its main transformation"
T - transformation process: "the means by which defined inputs are transformed into defined output" (where input is current situation and output is desired situation)
W - Weltanschauung: "an outlook, framework or image that makes this particular root definition meaningful"
O - ownership of the system: "some agency having a prime concern for the system and the ultimate power to cause the system to cease to exist"
E - environmental constraints: "features of the system's environments and/or wider systems which it has to take as

’given’"

which has since then been known as ’CATWOE’. T is the desired improvement, but the W concept is, as Checkland [1981,p.18] points out, "the most important one in the methodology". By making Ws (perspectives) explicit during analysis new insights can be generated that might, as Bergvall-Kåreborn [2001] puts it, "break away from self-imposed constraints and frames of mind." Though SSM has been in use for well over 20 years, almost in its original form, there are a number of problems, which are discussed by Bergvall-Kåreborn, Mirijamdotter and Basden [2004], and which have motivated various attempts to enrich SSM using Dooyeweerd.

Bergvall-Kåreborn [2006] sees a human activity system (such as an hospital) as functioning in all aspects, but with a primary purpose or meaning that is led by a qualifying aspect (§3.2.5). She shows that how a system functions is governed by what we treat as the qualifying aspect. The notion of qualifying aspect was embedded in SSM and this was tested by applying it to a case study that studied two programs aimed at creating new work opportunities in a small municipality in Sweden.

Whereas most Dooyeweerdian scholars assume, and try to identify, a single qualifying aspect of things, Bergvall-Kåreborn allowed the qualifying aspect to vary to reflect different perspectives. By asking "What if we treat the qualifying aspect of our system as X?", she found, helped clarify especially the meaning of W and T in SSM. It might stimulate fresh insights and overcome SSM’s tendency to generate regulative solutions.

Mirijamdotter and Bergvall-Kåreborn [2006] carried out an ’appreciative critique and refinement’ of SSM as a whole. First they demonstrated, by case studies (of attitudes taken by young people to the area in which they live), how aspectual analysis can enrich SSM’s Rich Picture, by being sensitive to the diversity of aspects and also be differentiating good from bad in each. In the design and comparing phases, five criteria are traditionally employed to judge quality of transformation. It is easy to see these are arbitrarily chosen (they all begin with E), but Mirijamdotter and Bergvall-Kåreborn showed how Dooyeweerd’s aspects not only justified these five, but also extends the set of criteria. This then led them to devise useful evaluative questions. Finally, they show how, by the worked example of an analysis of the Estonia ferry disaster, aspectual analysis can help build a good conceptual model and CATWOE. Thus, in every phase of SSM, Dooyeweerd is able to enrich its activity.

Basden and Wood-Harper [2006] have undertaken a detailed critical re-interpretation of CATWOE using Dooyeweerd’s ideas. Most of the elements may be understood in aspectual terms. To summarize (and over-simplify): T is multi-aspectual functioning. C is no longer individual people or organisation but repercussions in each aspect of T. E is primarily the intrinsic normativity of all the aspects, which are the law side, and only secondarily the constraints afforded by the entity side. A is multi-aspectual competences. W is perspectives centred on various aspects. O is responsibility for C and

A in particular. These re-definitions overcome a number of extant problems in SSM.

They claim that in effect they have transplanted CATWOE from the sterile soil of NFGM into the more fertile soil of CFR, so that it can bear more fruit. They also discuss the validity of doing this. Though both they and De Raadt [1991] have brought Dooyeweerdian thought into contact with extant systems models, they have done so in different ways. De Raadt accepted Beer's Viable Systems Model and its various concepts largely as they were given. Though he criticised the narrow biotic foundation of VSM, he did not criticise its structure but merely suggested multi-aspectual versions of each element. He also treated Ashby's notion of variety in the same uncritical way. By contrast, Basden and Wood-Harper critically re-interpreted the elements of CATWOE in the light of Dooyeweerd's Meaning- and Law-centred approach and accounted for why each is necessary to the model. They also justified doing this, especially on the basis that if Checkland's writing is examined, Meaning proves to be much more important than Being, and so his attempt to ground SSM in systems theory, which presupposes Being, may be seen as mistaken and also unnecessary. Dooyeweerd could, they claim, provide a much sounder philosophical foundation for SSM. In this way they might have avoided Eriksson's criticism above that De Raadt's MST in fact ended up back in the NFGM.

6.9 CONCLUSION

The starting point for formulating a philosophical framework for understanding IS development is to see it not as a technical, nor even socio-technical operation but as a human activity that exhibits a strong everyday flavour even though theories may be made about it and methodologies created to guide it.

Upon briefly reviewing the history of the area that is IS development (including recent developments like agile system development methods), and examining paradigms that pertain in ISD, as suggested by the time-honoured Burrell-Morgan model, it was found that it would be useful to attempt to understand ISD from the standpoint of the CFR ground-motive. The review was brief and limited, intended merely to indicate the usefulness and value of CFR (as understood by Dooyeweerd), rather than to demonstrate conclusively that CFR is the only possible standpoint or to show it is superior to the others. Attempting such a demonstration is left to another occasion, and would have to take into account approaches not based on the Burrell-Morgan model.

However, one reason why CFR is likely to be useful is because of its sensitivity to everyday experience, especially its ability to address complexity (cohering diversity) and normativity. Therefore the first principle developed is:

ISD is multi-aspectual human functioning, with especial focus on the normativity of the aspects.

This is similar to the framework formulated for computer usage, but

there is a difference. In usage, aspectual normativity is directed towards the past and present, and to evaluation thereof, while in ISD ...

Aspectual normativity is directed to the future, and to guidance towards it. Spheres of law are seen as enabling possibility.

Approaching ISD as everyday experience revealed, as with usage, ...

ISD is constituted of several different multi-aspectual functionings, each of which needs to be treated in a different way, but which interweave with each other in an enkaptic relationship.

These include the overall ISD project or process, anticipating usage, creating the IS, and knowledge elicitation and representation. Each was then examined, to yield the following understanding of them:

The overall project is to be guided by the shalom principle, but the aesthetic and social aspects are key, the aesthetic aspect in its focus on harmony to achieve a coherent project as its qualifying aspect, and the social because ISD is teamwork.

An insight from Dooyeweerd about the difference between types of social institutions was helpful in establishing the proper place of power and non-power relations, including authority, subordination, respect, friendship and intimate trust. Aspects of the overall project reach out to the other three human activities examined.

Anticipating usage is qualified by the juridical aspect of responsibility for all outcomes in future use, and aspectual normativity as possibility is a key insight.

Creating the IS is likewise multi-aspectual functioning, and the analytic to economic aspects thereof have been long recognised, but, less recognised, it can be a delight as well as a chore if the four latest aspects are given their due.

Knowledge elicitation and representation are qualified by the analytic, formative and lingual aspects and are best seen as enkaptically interwoven with each other rather than separate, in that each stimulates and depends on the other. Each of these aspects reaches out to the diverse aspectual meaning of the domain of application, which should be respected.

For the latter, Dooyeweerd's theory of knowing was important, including the insights this offered about the nature of tacit or taken-for-granted knowledge, which is currently recognised as a major challenge, and how to tackle it.

The fact that only certain of the aspects have been used as inter-process links, as shown in Fig. 6.3.4, suggests there might be other processes, but the discussion in this chapter indicates the ways by

which different processes can be understood and how they can be integrated into the whole of ISD.

Two practical devices were offered, simple aspectual analysis as was found useful in usage but with an orientation to future possibilities and to guidance of the ISD process, and Multi-Aspectual Knowledge Elicitation, developed by Winfield [2000], which has proven useful in developing domain ontologies and making some kinds of taken-for-granted knowledge explicit.

It was also shown how, instead of constructing a purely Dooyeweerdian methodology such as MAKE is, extant methodology can be critiqued, supported and enriched by Dooyeweerd. How this might be achieved was illustrated by briefly examining the work of several authors who have applied Dooyeweerd in this way to Checkland's Soft Systems Methodology. The reader is referred to the cited literature to find out more.

The benefits of this framework, if developed into guidelines, are that the overall ISD process should be healthier and more satisfying to all involved, as well as more efficient, because it attends to all aspects of the process. The frequency of developing the wrong IS should reduce because the framework provides a way of imagining the potential of the domain more clearly, in all its aspects. Artefact creation should become a more human, and yet more efficient, process. Knowledge elicitation and representation should be more complete because of the encouragement to explicitly consider more aspects of the domain meaning.

One question has not been considered in this chapter: from what raw materials do the IS developers work? A house is built from bricks, timber, nails, copper piping and the like. From what such basic building blocks should the IS (at least its technical component) be constructed? This is the topic of the next chapter.

References

Attarwala, F. T., & Basden, A. (1985). A methodology for constructing Expert Systems. R&D Management, 15(2), 141-149.

Avison, D. E., & Wood-Harper, A. T. (1990). Multiview: An exploration in Information Systems development. Henley-on-Thames, England: Alfred Walker.

Basden, A. & Nichols, K. G. (1973). New topological method for laying out printed circuits. Proc. IEE. 120(3), 325-328.

Basden, A. & Clark, E. M. (1980). Data integrity in a general practice computer system (CLINICS). Int. J. Bio-Medical Computing, 11, 511-519.

Basden, A. (2000). Some technical and non-technical issues in implementing a knowledge server. Software Practice and Experience 30(10), 1127-1164.

Basden, A., & Hibberd, P. R. (1996), User interface issues raised by knowledge refinement, International Journal of Human-Computer Studies. 45, 135-155.

Basden, A., Watson, I. D., & Brandon, P. S. (1995). Client centred: an approach to developing knowledge based systems. Chilton, England: Council for the Central Laboratory of the Research Councils.

Basden, A., & Wood-Harper, A. T. (2006). A philosophical discussion of the root definition in soft systems thinking: An enrichment of CATWOE. Systems Research and Behavioral Science, 23, 61-87.

Baumard, P. (1999). Tacit knowledge in organisations. London: Sage.

Beck, K. (2000). Extreme programming explained: Embrace change. Boston: Addison-Wesley.

Beer, S. (1979). The heart of the enterprise. New York: Wiley.

Bergvall-Kåreborn, B. (2001). The role of the qualifying function concept in systems design. Systemic Practice and Action Research, 14(1), 79-93.

Bergvall-Kåreborn, B. (2006). Reflecting on the use of the concept of qualifying function in

system design. In S. Strijbos, & A. Basden (Eds.), In search of an integrative vision for technology: Interdisciplinary studies in Information Systems (pp. 39-62). New York: Springer.
Bergvall-Kåreborn, B., Mirijamdotter, A., & Basden, A. (2004). Basic principles of SSM modeling: An examination of CATWOE from a soft perspective. Systemic Practice and Action Research, 17(2), 55-73.
Boehm, B. W. (1988, May). A spiral model of software development and enhancement. IEEE Computer, 21(5), 61-72.
Brandon, P. S., Basden, A., Hamilton, I., Stockley, J. (1988) Application of Expert Systems to Quantity Surveying. The Royal Institution of Chartered Surveyors, London.
Burrell, G., & Morgan, G. (1979). Sociological paradigms and organizational analysis. London: Heinemann.
Checkland, P. (1981). Systems thinking, systems practice. New York: Wiley.
Checkland, P. (1991). Systems thinking, systems practice: A 30-year retrospective. New York: Wiley.
Churchman, C. W. (1971). The religion of inquiring systems. In C. W. Churchman, The design of inquiring systems: Basic concepts of systems and organization (pp. 237-246). New York: Basic Books.
Clouser, R. (2005). The myth of religious neutrality: An essay on the hidden role of religious belief in theories (2nd ed.). Notre Dame, IN: University of Notre Dame Press.
Cockburn, A. (2005). Crystal Clear: a human-powered methodology for small teams. Boston: Addison-Wesley.
Collins, H. M. (1974). The TEA-Set: Tacit knowledge and scientific networks. Science Studies, 4, 165-186.
De Raadt, J. D. R. (1991). Information and managerial wisdom. Pocatello, ID: Paradigm Publications.
Dooyeweerd, H. (1984). A new critique of theoretical thought (Vols. 1-4). Jordan Station, Ontario, Canada: Paideia Press. (Original work published 1953-1958)
Dooyeweerd, H. (1986). A Christian theory of social institutions (M. Verbrugge, Trans.). La Jolla, CA: Herman Dooyeweerd Foundation.
Duda, R., Gaschnig, J., & Hart, P. (1979). Model design in the Prospector consultant system for mineral exploration. In D. Michie (Ed.), Expert systems in the microelectronic age (pp. 153-167). Edinburgh, Scotland: Edinburgh University Press.
Eriksson, D. M. (2006). Normative sources of systems thinking: An inquiry into religious Ground-Motives of systems thinking paradigms. In S. Strijbos, & A. Basden (Eds.), In search of an integrative vision for technology: Interdisciplinary studies in Information Systems (pp. 217-232). New York: Springer.
Fiol, C. M., & Huff, A. S. (1992). Maps for managers: Where are we, where do we go from here? Journal of Management Studies, 29, 267-286.
Flood, R. L., & Jackson, M. C. (1991). Critical systems thinking: Directed readings. Chichester, England: Wiley.
Gruber, T. R. (1995). Toward principles for the design of ontologies used for knowledge sharing. International Journal of Human-Computer Studies, 43, 907-928.
Guarino, N., Masolo, C., & Verere, G. (1999). OntoSeek: content-based access to the Web. IEEE Intelligent Systems, 14(3), 70-80.
Habermas, J. (1972). Knowledge and human interests (J. J. Shapiro, Trans.). London: Heinemann.
Hart, P. E., & Duda, R. O. (1977). Prospector: A Computer Based Consultation System For Mineral Exploration. Menlo Park, CA: SRI.
Hayes, P. J. (1978). The naive physics manifesto. In D. Michie (Ed.), Expert systems in the micro electronic age (pp. 242-270). Edinburgh, Scotland: Edinburgh University Press.
Hayes, P. J. (1985). The second naive physics manifesto. In R. J. Brachman, & H. J. Levesque (Eds.), Readings in knowledge representation (pp. 467-486). Los Altos, CA: Morgan Kaufmann.
Hines, J.G., Basden, A. (1986). Experience with use of computers to handle corrosion knowledge. British Corrosion Journal, 21(3), 1511-156.
Hirschheim, R., & Klein, H. K. (1994). Realizing emancipatory principles in information systems development: The case for ETHICS. MIS Quarterly, 18(1), 85-109.
Hirschheim, R., Klein, H. K., & Lyytinen, K. (1995). Information systems development and data modeling: conceptual and philosophical foundations. London: Cambridge University Press.
Jackson, M. C. (1991). System methodology for the management sciences. New York: Plenum Press.
Jones, M. J., & Crates, D. T. (1985). Expert systems and videotext: And application in the marketing of agrochemicals. In M. A. Bramer (Ed.), Research and development in expert systems: Proceedings of the Fourth Technical Conference of the British Computer Society Specialist Group for Expert Systems. London: Cambridge University Press.
Kalsbeek, L. (1975). Contours of a Christian philosophy. Toronto, Ontario, Canada: Wedge.
Knuth, D. (2001). Things a computer scientist rarely talks about. Stanford, CA: CSLI.
Midgley, G. (2000). Systemic intervention: Philosophy, methodology and practice. New York: Kluwer/Plenum.
Mirijamdotter, A., & Bergvall-Kåreborn, B. (2006). An appreciative critique and refinement of Checkland's soft systems methodology. In S. Strijbos, & A. Basden (Eds.), In search of an integrative vision for technology: Interdisciplinary studies in Information Systems

(pp. 79-102). New York: Springer.
Pacey, A. (1996). The culture of technology. Cambridge, MA: MIT Press.
Polanyi, M. (1967). The tacit dimension. London: Routledge and Kegan Paul.
Schuurman, E. (1980). Technology and the future: A philosophical challenge. Toronto, Ontario, Canada: Wedge.
Shortliffe, E. H. (1976). Computer-based medical consultations: MYCIN. New York: Elsevier.
Steels, L. (1985). Second generation expert systems. Future Generation Computer Systems. 1, 213-221.
Strijbos, S. (2006). Towards a 'Disclosive Systems Thinking'. In S. Strijbos, & A. Basden (Eds.), In search of an integrative vision for technology: Interdisciplinary studies in Information Systems, (pp. 235-256). New York: Springer.
Tarnas, R. (1991). The passion of the Western mind. Pimlico, London: Random House.
Ulrich, W. (1994). Critical heuristics of social planning: A new approach to practical philosophy. Chichester, England: Wiley.
Walsham, G. (2001). Making a world of difference: IT in a global context. Chichester, England: Wiley.
Weinberg, G. M. (1999). Egoless programming. IEEE Software, 16(1), 118-120.
West, D. (1992). Knowledge elicitation as an inquiring system: Towards a 'subjective' knowledge elicitation methodology. Journal of Information Systems, 2, 31-44.
Wielinga, B. J., & Breuker, J. A. (1986). Models of expertise. Proceedings 7th European Conference on Artificial Intelligence, Brighton, England, July 1986, 306-318.
Winfield, M. (2000). Multi-aspectual knowledge elicitation. Unpublished doctoral thesis, University of Salford, England.