Reconsidering Knowledge… And Business Improvement

Larry Lucardie[1,2], Paul Hendriks[3] and Joost van Ham[1]
[1]**Knowledge Values, Schiphol Airport, The Netherlands**
[2]**Uppsala University, Sweden**
[3]**Radboud University Nijmegen, The Netherlands**
Larry.Lucardie@knowledge-values.com
P.Hendriks@fm.ru.nl
Joost.van.Ham@knowledge-values.com

Abstract: The ever growing complexity of market processes continues to increase the importance of knowledge as the organization's core capability to maximize business performance. Current conceptions of knowledge and knowledge representation, however, prove to be highly unproductive. A fundamental problem here is that insight into the nature of knowledge is an inevitable requirement for adequate knowledge management that, nevertheless, is hardly met in business. In this article, we claim that adopting a functional view of the nature of knowledge reveals and restores the relation between knowledge and corporate effectiveness.

In a functional approach to conceptualization, functional equivalence instead of observable similarity serves as the basis for classification. The sets of conditions that have to be met in a particular situation are here taken as functional demands. These functional demands may vary across situations, thus precluding the valid possibility of a static one-on-one connection between functions and individual objects. Not the objects as potential instances of classes, but the relationships between objects given their properties and situations, defined in terms of functional demands, become central. These relationships define the concepts, and thus what we know. Classification amounts to relational matching of specified situations to specified objects.

The functional view not only enables content improvement through rational classifications, but also enhances process designs, implementations and process maintenance. It also aligns information technology to the new demands set by the knowledge economy by enabling goal-oriented, transparent and easy-to-use-and-modify knowledge structures. The paper further describes a real world case taken from the financial services industry to exemplify how a functional analysis of knowledge -including to the functional view aligned Match™ Technology- realizes great improvements in business performance.

Keywords: knowledge representation formalisms, functional view, rational classifications, functional equivalence, Match™ technology

1. Introduction

Maximising the performance of a company involves managing both its functioning in the short term and the company's long-term competitive position. Many authors have recognized that a focus on knowledge provides the adequate conceptual backdrop for combining both management objectives (e.g. Zack, 1999; Hislop, 2005). Adequate knowledge management (KM) therefore precedes optimal business performance and, so we argue, an adequate definition of the nature of knowledge precedes adequate KM. Yet to date, views on the relation between business improvement and knowledge on the one hand and knowledge and knowledge representation on the other hand are not convincing. What hampers knowledge realizing its full potential is the lack of systematic attention for knowledge in business. As a consequence, current KM initiatives tend to produce more information yet less knowledge.

What appears as a fundamental problem haunting KM debates is that adequately defining knowledge remains problematic. Definitions are either too specific to distinguish knowledge from information or too vague to warrant the design of truly knowledge-based management practices. In this article we argue that an interesting middle-ground between rigidity and vagueness emerges when knowledge is defined functionally. The functional approach allows flexible categorization by relating knowledge about referents to the functions these referents should perform, instead of treating referents around which organizational knowledge is developed and used as given objects.

The objective of the paper is to show that in-depth analysis of functional characteristics of knowledge helps to design smarter classifications in business processes and enables rational choices for knowledge representation formalisms. In addition, the paper explores which methodologies are needed to let knowledge structurally improve processes (transparent descriptions, clear relations between process levels, smooth work instruction integration in process steps) and Information and Communication Technology (ICT, improved software requirements engineering, shorter and better application development, and flexible architectures). A case taken from the financial services industry illustrates how a functional analysis of

Reference this paper as:
Lucardie, L., Hendriks, P. and van Ham, J. "Reconsidering Knowledge… and Business Improvement." *The Electronic Journal of Knowledge Management* Volume 6 Issue 2 2008, pp 135 - 144, available online at www.ejkm.com

knowledge and the Match™ Technology make the organization's knowledge the basis of business performance.

2. Functional classification as knowledge model

KM debates have produced a plethora of definitions of what knowledge is, and perhaps even more of what knowledge is not. Alvesson and Kärreman (2001, p. 997) warn us that the great diversity of approaches to knowledge has made the concept 'inconsistent, vague, broad, two-faced and unreliable' and that going for all too tangible definitions of knowledge may be unwise or counterproductive. Alvesson and Kärremans complaint is reminiscent of Dewey and Bentley's lamentation in 1949 that 'knowledge is one of those vague words one is at times compelled to use, a loose name that has been used to refer to a great many, often different things' (cited in Gourlay, 2006, p.1424). In fact, if it is management that we are interested in, engaging in philosophical debates regarding the nature of knowledge should be avoided because knowledge by its very nature cannot be managed. "Knowledge management is thus threatened by falling into pieces if both the two ingredients are taken too seriously" (Alvesson & Kärremans, 2001, p. 1015). Yet because knowledge matters and because many KM discussions indeed are built on highly debatable definitions of knowledge, not taking knowledge seriously is no option either. Indeed knowledge is different from information, and any knowledge manager should avoid using the terms interchangeably because that would involve downplaying knowledge.

When considering the need and possibility of defining knowledge in a KM context, it is important to note that the general distinctions used in the KM debates have only limited value. For instance, the distinction between tacit and explicit knowledge is used most often to separate information from knowledge. Particularly tacit knowledge is then heralded as the most valuable organizational resource, because it can produce a competitive edge (e.g. Zack, 1999). However, as Achterbergh and Vriens (2002) rightly stress, the fact that knowledge is tacit as such cannot guide an inquiry into which knowledge will gain value for the individual organization. The tacitness of knowledge as such cannot help decide which knowledge is relevant for management, and which is not (Jimes and Lucardie, 2004). The same criticism applies to the numerous other taxonomies that have been proposed, sometimes related to the tacit-explicit dichotomy, leading to such distinctions as overt versus covert knowledge, declarative versus procedural knowledge, object versus process knowledge and so on. Exactly because of their general nature, the relevance of these distinctions to KM debates is rather limited. What is needed therefore is (1) a broadly accepted, general definition of the concept of knowledge and (2) a specification of that general definition that may serve to draft management tools, and particularly the ICT tools for KM that are the focus of this paper.

No taxonomy of knowledge types can produce a general definition of knowledge, but only a specification of the roles or functions knowledge plays. Allen Newell (1982) offers such a functional definition in his famous discussion of the 'knowledge level' at which artificially intelligent – or knowledge-based – systems should be defined according to Newell. He argues that knowledge is 'whatever can be ascribed to an agent such that its behaviour can be computed according to the principle of rationality', explaining that thus knowledge 'is characterized entirely functionally, in terms of what it does, not structurally, in terms of physical objects with particular properties and relations' (Newell, 1982, p. 9-11). While Newell's definition has been the object of criticism, for instance for lacking a crisp delineation of rationality, it should be noted that it contains a valuable referral to behaviour. It is in line with the pragmatists' approach that knowledge has no existence separate from behaviour. The strong connection between knowledge and behaviour is specified in John Dewey's concept of productive inquiry. As Cook and Brown (1999, p. 388) explain, "productive inquiry is that aspect of any activity where we are deliberately (though not always consciously) seeking what we need, in order to do what we want to do." Dewey maintains that when our experiences are non-reflectional, as is characteristic of our everyday living, they produce know-how or result from know-how. On the other hand, when our experiences are reflective, as is typical of the professional behaviour of academics, they shape and are shaped by know-that (Dewey, 1922). Thus the artificial separation of tacit (know-how) from explicit (know-that) knowledge is lifted when knowledge is not treated as something that in a Cartesian sense is fully separable from behaviour. It is important to note that these functional approaches to defining knowledge can only sketch the confines of the knowledge landscape, but do not produce any tangible guidelines for engaging in or abstaining from KM activities.

An important step towards KM tool design can be made when linking to the distinction of knowledge-as-possession versus knowledge-as-practice (Cook & Brown, 1999; Hislop, 2005). This distinction appears to have become the standard divide in KM debates separating the older, ICT-focused approaches to KM from the newer, people-centred ones (Chiva & Alegre, 2005). When seen as possession, knowledge is treated as an entity, as objective facts, and as derived from intellectual process (cf. Hislop, 2005; Chiva & Alegre, 2005). Practice-based approaches stress that knowledge and knowing are inseparable, knowledge is embedded in practice, and it is socially constructed, culturally embedded and situated (cf. Hislop, 2005; Chiva & Alegre, 2005). They see tacit and explicit knowledge as aspects and not types of knowledge. Knowledge is both at the same time. Newell et al. (2002) show how the two knowledge models lead to distinct understandings of KM. Following a possession approach, KM's primary aim is to codify and capture knowledge, and the critical success factor is technology. From a practice stance, KM's primary function is encouraging knowledge sharing, and the critical success factor is trust and collaboration.

Given the basic connection of knowledge to behaviour, any satisfactory approach to KM will need to connect the two approaches (Cook & Brown, 1999; Chiva & Alegre, 2005). A key concept in linking a possession to a practice approach is that of 'dynamic affordance', which refers to what becomes possible when knowledge is used as a tool for action in concrete situations (Cook & Brown, 1999). 'Affordance' refers to how a situation "affords" doing. For instance, a doorknob (Cook & Brown, 1999, p. 389), when appropriately designed, affords opening a door smoothly. Dynamic affordance adds the notion that actual doing is established through interaction in concrete situations or concrete contexts. The same doorknob may prove functional to one person, yet dysfunctional to another with different physical abilities (e.g. a midget or someone with arthritis). What then is a doorknob? Is it the well-crafted, appealing physical object or a string hanging above the keyhole that opens the door? What is a table? Is it the flat stone in the countryside or the elaborately decorated Louis Seize piece of furniture that facilitates eating and writing? The answer can only be that the crafting and design are important, but only when we link them to the purpose these objects may serve in factual situations, can we say what doorknobs, tables or any other objects 'are'.

What then does this imply for drafting intelligent, knowledge-based ICT tools for KM? As we will argue, the line of reasoning developed above (1) shows the prime importance of classification as manifestation of knowledge, and (2) points to relational matching as the key mechanism in achieving classification. Both points will now be elaborated. Figure 1 serves to illustrate the argument.

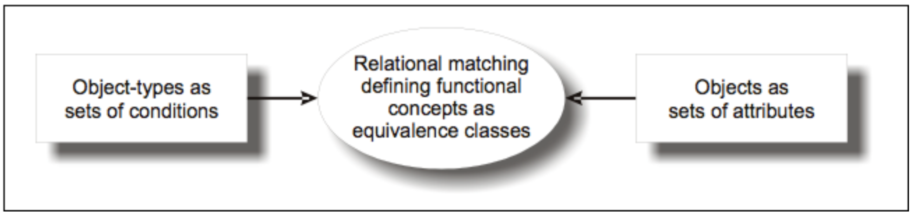

Figure 1: Functional classification (adapted from Reitsma, 1990; Lucardie, 1994)

What the argument proposes is to reverse our common way to look at knowledge and knowing. Knowledge is not defined as separately existing entity with particular perceptions, e.g. problem detection, and actions, e.g. problem solving, as potentially useful by-products. Rather knowledge appears as a particular way in which human behaviour deals with circumstances. Therefore what guides actions and behaviour, whether these are seen as directed by goals and objectives or conceived as opportunistic and chaotically-reactive, embraces knowledge and simultaneously defines the potential value of knowledge.

What defines knowledge is not some taxonomy of related insights deposited in concepts and conceptual relations, with ignorance being defined as the lack thereof. Instead, the problems to be detected and solved, the opportunities to be seen and seized, or any other objectives to be perceived and achieved, provide the more appropriate starting point for understanding knowledge and thus for modelling it. Knowing then amounts to classifying objects with respect to a given objective, and knowledge to the rules that guide classification (Lucardie, 1994). Particulars of this functional approach, which centres on establishing functional equivalence classes via matching contexts and objects, are discussed below (see also Reitsma, 1990, and Lucardie, 1994). This approach is at odds with the standard practice of understanding and building ontologies that describe instances, concepts, attributes and relations between concepts (e.g. Sharman & Kishore, 2007). These focus on the structural similarities of objects in an unduly static fashion. They look at the right hand side of Figure 1 in isolation, instead of on the relational matching mechanism that gives the right hand side its meaning. The underling model is one that has been labelled above as the possession model. Therefore, ontology-based knowledge representation cannot escape possession approaches to knowledge and KM.

3. Business enhancement viewed from a functional knowledge perspective

By disentangling knowledge (content) from knowledge representation formalisms (data structures and access mechanisms), the functional view on knowledge systematically changes the ways business performance enhancements can be realised. Firstly, it helps to re-organise knowledge by installing the rationality principle in the content that is a key driver for business performance. Secondly, it helps to map knowledge to representation formalisms as human resources, documents, (formal) techniques and applications such as database management systems and flow engines.

3.1 Creating rational classifications

Business cannot do without classifications and associated interactions. Due to lack of transparency and because historical artefacts have crept into classifications, many classifications that are operational in business are flawed. The overwhelming volume of information that classifications can generate, their often doubtful validity and unnecessary complexity are important problems that, when viewed from a positive angle, provide space for significant business improvements.

A strong asset of the functional view is that it helps organisations to start working from a goal or system of goals. A goal-oriented approach disentangles the confusion that often occurs when an organization attempts to execute an object-type such as ‘valid service request’ or ‘optimal product (portfolio) offering’ while not taking into account that defining object-types can and should only be accomplished through a goal or multiple goals.

This leads to the phenomenon of functional equivalence, which involves the assessment that objects are identical—even if they possess quite different attributes—because they can perform the same function. In other words, objects may vary in attributes, but if they match one of the constructs of a goal-constituted object-type, they are functionally equivalent. Functional equivalence involves three basic mechanisms: conditional relevance, conceptual interaction and variation limited to goal-constructed categories.

- *Conditional Relevance.* Under specific conditions, other attributes may become important for determining class membership. Their relevance is conditional upon circumstances that also need to be incorporated into the object-type.
- *Conceptual Interaction.* Categorizations of attributes of objects may influence each other.
- *Variation limited to goal-constructed categories.* Objects may have different attribute values, but this variation is limited to, or falls within, goal-constructed categories.

What benefits does the installation of goal-oriented and functional equivalence principles yield? We argue that the advantages relate to transparency of classifications, sparseness of information, simplification and correctness of classifications.

As an example, consider the case of a computer system that had been developed to determine a students’ eligibility for university scholarships. The object-type ‘scholarship student’ as it had been incorporated into the system led to complaints from students who were overlooked for a scholarship because the system mistakenly failed to classify them as a ‘scholarship student’ (mismatch). It subsequently appeared that the rather complex object-type was constructed using the government’s goal ‘should suit budget,’ while the universities linked to the scholarships had the implicit goal to acquire as many scholarship students as possible. Analysis revealed that at least two distinct object-types ‘scholarships’ should have been distinguished based upon the different goals of the actors involved.

In addition to the efforts spent handling students’ complaints, the costs to reconcile both object-types in an adapted system were substantial. The inclusion of goals and the related distinction of several object-types (and objects) would have eliminated irrelevant information, reduced complexity and increased transparency of knowledge. When goals determine which conditions are relevant for the definition of an object-type, knowledge becomes something in use as a function of the organization’s goals. This prevents knowledge from becoming obsolete, or just a sitting stock of information: for when the goals change, knowledge changes with it. This is true irrespective of whether knowledge is processed through humans, systems or both.

The correctness or validity of classifications is often damaged by reification, which underlies the sometimes highly cloaked tendency to use taxonomies in goal contexts that are different from the ones for which they were created.

Unfortunately, a huge number of classifications in business processes display the characteristics of reification creating exceptions as a rule. And dealing with exceptions is not cheap. Processing them requires the employment of additional human resources. What in fact occurs is a compensation of (partly) invalid classifications by additional human resources in operations departments of organisations!

3.2 Boons and limits of codification and automation

What about the roles of codification, formal knowledge representation and automated support via ICT? Are they out of the picture once the predominance of tacit knowing has been recognized and its social-practice character accepted? We claim that in the process of enriching limited possession thinking with conceptually richer social-practice elements, codification has lost nothing of its appeal. In an interesting essay on performance variability, or the phenomenon that businesses cannot maintain the same level of performance even if circumstances appear familiar, Matson and Prusak (2003) distinguish between situations of high and low predictability and of high and low frequency. In the terminology of this paper, these distinctions would refer to classifications for which the rules (knowledge) are known (=predictable) or not (=unpredictable) and those that have to be made often (=frequently) or not (=rarely). Matson and Prusak describe how content-based ICT (e.g. implementing a decision algorithm) can be economically helpful for dealing with high-frequency problems, and person-targeted ICT (e.g. expert-locator systems) for low-frequency problems. The approach described here is best applicable to the high-frequency class of problems. In situations where a specific classification has to be made only occasionally it would probably not be economically interesting to support it with an automated system.

However, the intrinsically more interesting dimension is that of high and low predictability. Matson and Prusak identify different classes of ICT applications for dealing with predictable and unpredictable solution paths. For the former they suggest decision algorithms and best practice databases, and for the latter case-based learning and Q&A databases. We agree with Matson and Prusak's implicit assertion that codification can be useful for any performance-variability-related problem class, but we disagree with treating the various predictability-frequency combinations as fundamentally different, as they – also mostly implicitly – do. What we argue instead is that in all situations – whether predictable or not, and whether frequent or not – tacit knowledge is indispensible and suitability of ICT support lies in how well it connects to that aspect of knowledge. Therefore not different application types are needed in different situations, but a similar knowledge modelling approach with a different focus on the amount and perhaps nature of expertise required in the applications' users. Automated systems cannot and should not replace individual knowing; they are knowledge-based to the degree that they allow hooking into subsidiary background knowledge that gives meaning to the content of the systems. Tacit (or background) knowledge is not just required to deal with situations of low predictability, also a predictable classification that is exclusively or mostly based on existing knowledge cannot do without tacit knowledge – at least as this was understood by Polanyi (1966), who introduced the concept. Tacit knowledge – or subsidiary background knowledge – is always needed to be able to understand the generalizations, to make exceptions, to check the appropriateness of descriptive accounts of the individual objects, to verify the completeness of the classification-decision table, and to assess the suitability of the resulting classification. Obviously, functional modelling does not and cannot result in codifying the indispensible tacit aspects of knowledge into a system. However, what it can claim to achieve is that it offers a perspective on the explicit or focal aspects of knowledge that does not unduly separate those explicit aspects of knowledge from tacit knowing. The focal knowledge is here not represented starting from the generalizations, the rules, but taking the goal as a context for using these generalizations as the starting point. Thus the logic of classification follows the practice of knowing, and needs no extra translation step to see how individual situations make use of or alter existing 'knowledge as generalization'.

3.3 Mapping rational classifications

Improved rational classifications should be mapped to well-chosen knowledge representation formalisms, and supported by smart technologies. Software tools that follow a functional view enable the development of smart classifications. The Match™ Suite of technologies is an example of a well-elaborated tool kit to do exactly this. It offers a range of representation formalisms that are function-oriented and can deal with functional equivalence. Basically, it enables capturing smart classifications through digital knowledge bases (Match™ Developer) and ensures that classifications are interactively executable (Match™ Player).

Match™ Developer allows constructing and maintaining a knowledge-based system consisting of knowledge tables that are linked together. A knowledge table is a table that represents the exhaustive whole of mutually exclusive conditional statements within an a priori defined problem domain (Verhelst, 1980). Match™ Developer adheres to that definition. Therefore it can perform automated checks on domain completeness, exhaustiveness and consistency of created knowledge tables.

Figure 2 is an example of a knowledge table created with Match™ Developer where a goal is assessed and used as a guideline for establishing the relevant conditions and their respective alternatives. In case of our example: determine the 'Type of Client'.

Type of Client

C1	Duration of account (in months)	=<12				>12	
C2	Business performance (C/I ratio)	=<50	>50 AND =<75		>75	=<60	>60
C3	Wealth	-	Wealthy	Not wealthy	-	-	-
A1	Type of Client	Special	Special	Normal	Problem	Special	Normal
	Rules	1	2	3	4	5	6

Figure 2: A Match™ table defining 'Type of Client' based on functional equivalence

This table also shows the concept of functional equivalence as it was unfolded above into the three mechanisms of conditional relevance, conceptual interaction and variation limited to goal-constructed categories. Firstly, conditional relevance shows in how the condition 'Wealth' is introduced in the decision logic, since that condition is only relevant in case the 'Duration of account' is less then 12 months and the 'Business performance' is between 50 and 75. In other circumstances, wealth is not relevant for classifying clients. Secondly, conceptual interaction becomes visible in the fact that the categorization of 'Business performance' is influenced by the value of 'Duration of account'. When account duration is less then 12 months, 4 categories of business performance become relevant. When the account duration is over 12 months, differences in business performance are considered to be accurately captured by only 2 classes. This phenomenon of conceptual interaction becomes manifest in the mutual influence of the attributes' categorizations. Thirdly, reduction in variation through goal-constructed categories is established by the fact that clients who are described by very different combinations of characteristics will still be categorized into a limited number of goal-constructed categories. Regardless of the duration of the account, the height of the client's cost-income (C/I) ratio or that person's wealth, all clients will be classified in one of the categories 'Problem', 'Normal' or 'Special'.

With Match™ Player the knowledge bases that are created with Match™ Developer, can be executed through the use of a meta-interpreter (or inference or reasoning engine). This interpreter will allow for automated evaluation of the goal of the knowledge base. In the example of Figure 2, it will try to assess the goal 'Type of Client', by evaluating the conditions C1-3. The conditions can receive their value in several different ways: for instance by asking an end-user to provide an answer, or retrieving it from a database. This will allow for several business improvements ranging from process optimizations to streamlining a company's ICT-organization.

3.4 Business improvements

The functional theory and the Match™ Suite of technologies especially improve business in the domains of processes and ICT.

3.4.1 Process improvements

A functional conception of knowledge supports the alignments of processes to goals. First of all, a rational classification can be a process description taking care, for instance, of conditional relevance, which points to the fact that in certain contexts a process step does not need to be performed. Secondly, it helps to keep the levels in process descriptions transparently connected, so that implications of changes in different layers are traceable. Thirdly, the usual but artificial separation between a process description and the knowledge relevant to perform a step in the process is being eliminated which gives space for transparent interaction between 'process' knowledge and 'execution' knowledge. A smooth and flexible integration of work instruction and process is the result. Fourthly, flow analysis with large volumes of objects flowing through the process, is possible using the Match™ Player in batch mode even before the process is implemented.

3.4.2 ICT improvements

ICT may produce a multitude of benefits, such as faster and more transparent requirements engineering, shortened application development, rationalisation of the application portfolio and increased flexibility of IT architectures. The benefits of ICT-based functional modelling in all these domains will be shortly addressed now.

Requirements engineering is one of the most critical activities in software development (Hickley & Davis, 2004). Here, the functional view does not only help to organise the software requirements specifications, but it also supports the selection of techniques that increase transparency for non-computer-oriented persons. In Match™ Developer, for instance, we carefully selected knowledge tables, frames and a logic programming language to assist the analyst dealing with functional equivalence in unprecedented visual and iterative ways. In all stages of requirement engineering -from elicitation up to verification- the chosen combination of techniques proved to be fruitful in various application domains. Especially when it comes to determining completeness and consistency in the verification stage and to provide a sound basis for design and implementation and testing.

An essential 'side effect' of the approach described above is that the addition of a meta-interpreter forms a new dimension to the efficiency of application development. While the data structure component consisting of hierarchical knowledge table systems, frames and logic programming is extremely powerful and expressive already, the meta-interpreter as the process component render the specifications executable. This takes application development to a level beyond model-driven architectures (MDA) where specifications, for instance written in UML, are stepwise and semi-automatically translated into code bases (.NET and/or J2EE). In all these cases the programme itself will need to be repaired to deal with flawed classifications or new insights. Apart from a stronger business/ICT alignment, the approach advocated here leads to considerable reductions in resources used for designing, testing and programming applications and especially in their maintenance (Figure 3).

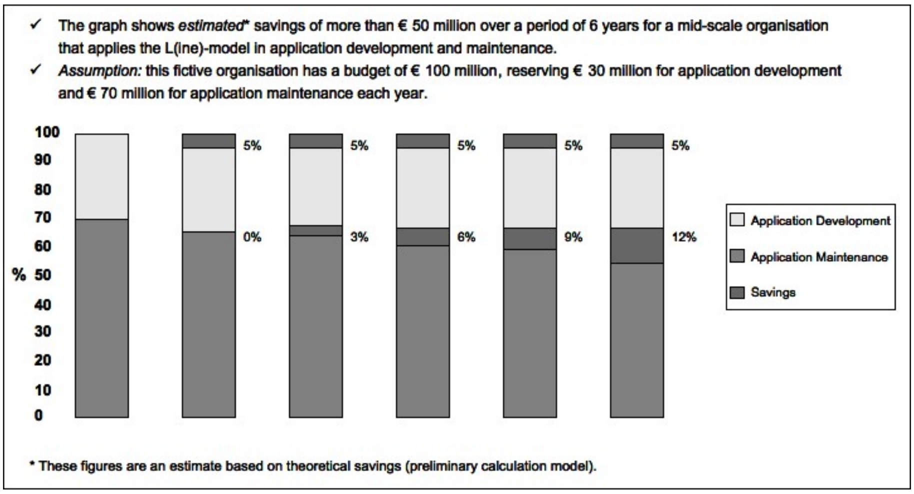

Figure 3: Effectiveness of knowledge-based application development

Current IT systems often do not meet organisations' objectives concerning functionality and flexibility. Executive management is therefore taking measures to improve their performance (Debevoise, 2005). One option is to rationalise a company's application portfolio by replacing legacy IT systems with more intelligent systems that allow business rules and processes to be abstracted and formalised in such a manner that they can be improved without the necessity to re-develop the complete system. This can be done by the concept of flexible IT architectures: building a layer of agile knowledge bases on top of legacy systems and below the end users in various processes! Another approach is to deploy functional knowledge bases as web services that function as self-describing, self-contained software modules available via a network such as the internet for fully automated complex transactions. (Papazoglou, 2008).

4. Case study: Intelligent debtor management

We now turn to a case study to illustrate ICT-supported modelling of functional equivalence classes. That case study concerns the debtor management process for a financial service provider. Debtor management comprises three sub processes (see Figure 4): signalling (identifying debtors), decision-making (selecting a debtor strategy) and execution (conducting a selected strategy). In the case organisation a variety of applications supported these processes, with a dedicated Cobol application specifically functioning as the decision-making engine. That application was the source of many problems. The knowledge of the decision-making process was spread out over multiple Cobol modules that were often inconsistent and hardly accessible. The original functional specifications were largely outdated. Deciphering the original Cobol code would have been the only way to access vital knowledge, and to alter or expand it. Since many of the original Cobol programmers had been hired from outside or had left the company, their argumentation was not easily accessible.

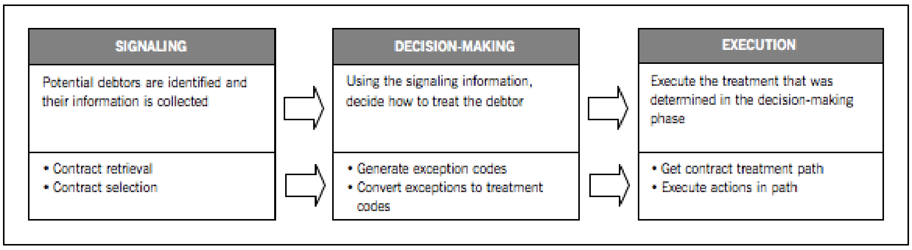

Figure 4: Debtor Management Process at a global financial institution

4.1 Phase I: Extracting knowledge from systems

The first phase towards a more intelligent debtor management process was the extraction of relevant knowledge out of the various sources and formalise it into knowledge bases through use of the Match™ Suite. A thorough analysis of the working of the Cobol application was performed to ensure that the signalling/decision-making rules in the knowledge base produced the same results as the Cobol modules before the system became operational.

This could be tested and simulated by using a Match™ Player to process a batch of debtors while recording the results and comparing them to the output of the old system.

By restructuring the knowledge and processes according to certain design principles (Lucardie et al., 2004) handled by the Debtor Management decision-making engine in goal-oriented, transparent and easy-to-modify digital knowledge bases knowledge fragmentation was eliminated and costs driven down significantly, highlighting the optimisation opportunities of the efficiency of debt collections.

Additional benefits are:

- Increased process maintenance efficiency: Using digital knowledge bases, process changes can be efficiently and transparently specified and tested before actual programming takes place further reducing operational costs.
- Clear insight into process flow and structure: Possible omissions and irrelevant process steps become immediately noticeable enabling process optimization.
- Increased leverage of organisational knowledge: Knowledge previously residing in the heads of individuals and stored in hard-coded applications becomes transparent and accessible organisational knowledge leveraged through digital knowledge bases.
- As maintenance of the Cobol modules was outsourced due to complexity that now disappeared, outsourcing can be viewed from a different perspective.

4.2 Phase II: Smart functional classifications

Phase II starts by identifying the goals of the debtor management process. Once these have been determined, variables that are required to establish these goals can be identified.

A single set of generic conditions is used to determine the class of a debtor. Based on this classification, strategy assignment takes place. By intelligently choosing the generic classification conditions, strategy redundancy is eliminated and the number of strategies can be limited. Due to reduced decision-making complexity, the transparency of the process is greatly enhanced. Because of the goal oriented nature of the knowledge base, product specific information is retrieved only when the goal requires it (Figure 5).

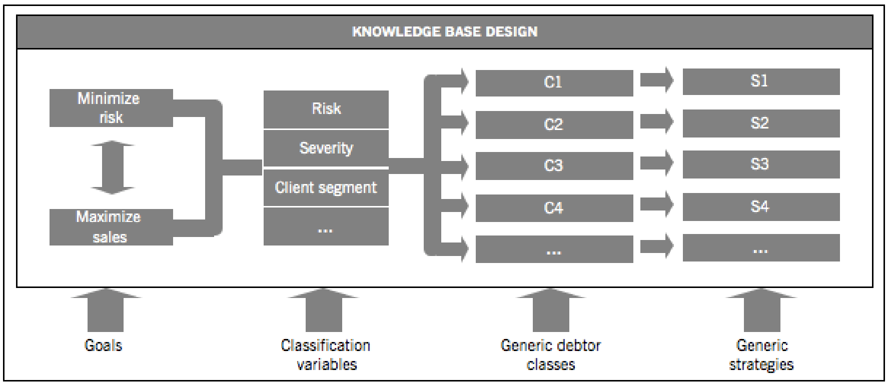

Figure 5: Using smart functional classifications to reduce complexity

As a result of increased transparency and flexibility, maintenance and modifications require fewer FTE. Increased flexibility also makes it possible to shorten the time to market of new strategies thereby enhancing debt collection efficiency.

5. Conclusion

In this paper a functional approach to knowledge modelling has been described and illustrated. That approach is characterized by a focus on classification as combined knowledge and knowing, the concept of relational matching as the basis for classifying individual objects into object-types, the notion of functional equivalence as the basis for understanding sameness and difference, and conditional relevance and conceptual interaction as descriptive of how characteristics of objects feed into their classification. Knowledge by its nature combines abstraction with concreteness, and group elements with individual elements. Tsoukas' (2001, p. 983) definition of organizational knowledge neatly shows all these elements: "Organizational knowledge is the capability members of an organization have developed to *draw distinctions* in the process of carrying out their work, in particular *concrete contexts*, by enacting sets of generalizations (*propositional statements*) whose application depends on historically evolved *collective understandings* and experiences" (emphasis in the original). A functional modelling approach fits seamlessly in the combined possession-practice approach to knowledge this definition reveals. As to the abstract side of knowledge, the key focus of a functional approach lies on modelling classification, and not generalization. Generalization comes into the picture as the argument leading to classification decisions, and the justification of resulting class composition. Thus, a functional approach may help cross the divide between cognitive codification and social-practice approaches to KM where these two perspectives engage in the kind of generative dance that Cook and Brown (1999) describe. In parts of the KM discussion, knowledge codification and knowledge exploitation appear to have received a bad name. This applies particularly to those contributions to the KM debate that focus on power, emotions, situatedness and conflict in relation to knowledge questions. We claim that, by definition, advanced knowledge services based on smart codification are a prerequisite for business enhancement. Services that permit knowledge level analysis according to the functional view, will lead to improved engineering of knowledge representation formalisms, moving beyond yet encapsulating ontological engineering topics as semantic web (Gomez-Perez et al., 2004). Social-practice approaches to KM should embrace codification and ICT with as much vigour as the indeed flawed and inherently limited cognitive-possession tradition of KM did (and partly still does). Adequate knowledge services require the disentanglement of knowledge and knowledge representation, the availability of a knowledge theory and ICT aligned to this view. Impressive business improvements are indeed feasible, by reconsidering knowledge and thus…. business performance.

References

Achterbergh, J.M.I.M., & Vriens, D.J. (2002). "Managing Viable Knowledge". *Systems Research and Behavioral Science,* 19(3), 223-241.

Alvesson, M., & Kärreman, D. (2001). "Odd couple: Making sense of the curious concept of knowledge management". *Journal of Management Studies, 38*(7), 995-1018.

Chiva, R., & Alegre, J. (2005). "Organizational learning and organizational knowledge - Towards the integration of two approaches". *Management Learning,* 36(1), 49-68.

Cook, S.D.N., & Brown, J.S. (1999). "Bridging epistemologies: The generative dance between organizational knowledge and organizational knowing". *Organization Science,* 10(4), 381-400.

Debevoise, T. (2005). *Business Process Management with a Business Rules Approach*. Business Knowledge Architects, Roanoke.

Dewey, J. (1922). *Human nature and conduct: an introduction to social psychology*. Holt, New York.

Gomez-Perez, A., Fernandez-Lopez, M, & Corcho, O. (2004). *Ontological engineering*. Springer-Verlag, London.
Gourlay, S. (2006). "Conceptualizing knowledge creation: A critique of Nonaka's theory". *Journal of Management Studies,* 43(7), 1415-1436.
Hickley, A., & Davis, A. (2004). "A Unified Model of Requirements Elicitation". *Journal of Management Information Systems*, 20(4), 64-84.
Hislop, D. (2005). *Knowledge management in organizations: a critical introduction*. Oxford University Press, Oxford.
Jimes, C., & Lucardie, L. (2004). "Reconsidering the tacit-explicit distinction – A move toward functional (tacit) knowledge management". *Electronic Journal of Knowledge Management*, *1*(1), 23-32.
Lucardie, G.L. (1994). *Functional Object-types as a Foundation of Complex Knowledge-based Systems*. Technische Universiteit Eindhoven, Eindhoven.
Lucardie, G.L., Eriksson-Lundström, J. & Ezitis, J. (2004). "Design Principles for Table-Based Knowledge Bases - A Case Study from the Financial Sector". *Proceedings of the 5th European Conference On Knowledge Management Management.* 491-504.
Matson, E. & Prusak, L. (2003). "The performance variability dilemma". *MIT Sloan Management Review*, *45*(1), 39-44.
Newell, A. (1982). "The Knowledge Level". *Artificial Intelligence,* 18(1), 87-127.
Newell, S., Robertson, M., Scarbrough, H., & Swan, J. (2002). *Managing knowledge work*. Palgrave, Houndmills.
Nonaka, I., & Takeuchi, H. (1995). *The Knowledge-Creating Company*. Oxford University Press, New York.
Papazoglou, M. (2008). *Web Services: principles and technology*. Pearson Education Limited, Harlow.
Polanyi, M. (1966). *The Tacit Dimension*. London: Routledge and Kegan Paul.
Reitsma, R.F. (1990). *Functional classification of space: aspects of site suitability assessment in a decision support environment*. International Institute for Applied Systems Analysis, Laxenburg.
Sharman, R., & Kishore, R. (2007). *Ontologies: a handbook of principles, concepts and applications in information systems*. Springer, New York.
Tsoukas, H., & Vladimirou, E. (2001). "What is organizational knowledge?" *Journal of Management Studies, 38*(7), 972-993.
Verhelst, M. (1980). *De praktijk van beslissingstabellen*. Kluwer, Deventer.
Zack, M.H. (1999). *Knowledge and strategy*. Butterworth-Heinemann, Boston.

