Knowledge sharing in distributed organisations

IJCAI-99 Workshop on
Knowledge Management and Organizational Memories
July 31, 1999
Stockholm, Sweden

Are Sørli
Gunnar John Coll
Einar Dehli
Kjell Tangen

Computas AS, P.O.Box 482, N-1327 Lysaker, Norway
http://www.computas.com
Phone: +47 67831000, Fax: +47 67831001
{Are.Sorli, Gunnar.Coll, Einar.Dehli, Kjell.Tangen}@computas.com

This paper discusses how agent technology can help solve the knowledge management challenges faced by evolving organisations. It addresses the problem of sharing and distributing knowledge across an enterprise where knowledge assets are highly decentralised. An organisation-specific scenario and a technological solution are presented.

Keywords: *Knowledge sharing, knowledge management, organisational memories, agents, agent communication, ontologies, KQML, FIPA ACL, XML*

1 Introduction

Recent work in the field of knowledge management centres on how increasing globalisation and the explosive growth of the Internet are forcing businesses to adapt to accelerating changes in their environment. It is argued that organisations need to be able to evolve rapidly in order to stay competitive. This means that increasingly adaptable, flexible and organic organisational structures emerge. Moreover, since most companies have come around to viewing their knowledge as a main asset, managing and evolving this knowledge has become a key concern. This paper addresses technology-supported management of organisational knowledge.

The first part of the paper will concentrate on organisational aspects. In section 2, we will give a practicable characterisation of what we mean by the term 'organisational knowledge'. In section 3, we will present a real-world knowledge management scenario from a world-wide organisation.

The second part will focus on technology. In section 4, we will outline the technologies we think best suited to tackling knowledge management issues. In section 5, we will show how these may be applied to handle the previously presented scenario.

2 Organisational knowledge

To talk meaningfully about managing knowledge in an organisation, we need a rough-and-ready definition of the term 'organisational knowledge'. First, we define 'knowledge' for our purposes, then what we mean by prefixing 'organisational'.

2.1 Knowledge

'Knowledge' is basically a folk-psychological ([Stich, 1983]) term. In addition to its imprecise everyday connotations, it has numerous meanings within more specialised disciplines. We will not challenge pedagogues, epistemologists, psychologists, sociologists, anthropologists, cognitive scientists, neuroscientists, organisational theorists or knowledge management theorists as to the definitive nature of knowledge. Rather, we will propose a narrowed, practical perspective, suitable for tackling knowledge management challenges with the help of technology. This technology-slanted view, however, leaves certain important knowledge management aspects, such as motivation, emotion, ethics and power structures, to be addressed by others.

2.1.1 The knowledge cycle

The concept of knowledge in folk psychology is usually strongly **content-oriented**; knowledge is *something* a subject may possess (witness such expressions as "What knowledge do you have?", "How much have you learned" etc.), a kind of mental content. Consequently, attempts at understanding knowledge are often structural in nature, using taxonomies to classify *kinds* of knowledge (e.g. 'factual', 'procedural', 'formal', 'practical' 'explicit', 'tacit' etc.). Furthermore, it is not uncommon to see the notion of knowledge as detachable from mind, in the sense that many would be willing to say that knowledge can be present in a book or other passive extensional medium.

Our position differs by viewing knowledge as a **relation-oriented** concept; it is intimately tied to behaviour in the real world; knowledge presupposes an **actor** with a task agenda. ('Actor', here, means any entity able to display intelligent behaviour relative to events external to itself.)

A simplified model, which we call the **knowledge cycle**, shows how an actor interacts with the environment in order to create *and use* knowledge:

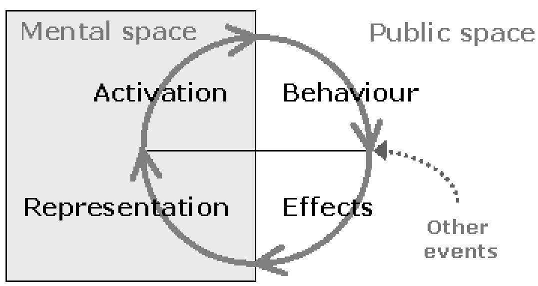

Figure 1: The Knowledge Cycle

This model is essentially behaviouristic ([Skinner, 1953]), in the sense that it says that knowledge is a relation between an actor and its environment, and can only be studied and validated by assessing the quality and the contingencies of the actor's behaviour. It is also cognitivistic, because we allow the study of the intermediate function: How is it possible to build representations which sustain knowledgeable behaviour?

The knowledge cycle comprises four simultaneously active stages:

- The actor's **behaviour** in the external environment. The more rational the behaviour (or, equivalently, the more competent the actor), the more we may assume activated knowledge (the previous stage) is the cause. Note that in this view, competence is in the eye of the beholder; an actor can be said to be knowledgeable only to the extent that he demonstrates it through rational action. See [Russell and Wefald, 1991] for a definition of rationality well suited to this view.
- The observable **effects** on the external environment of the actor's behaviour.
- The **representation** in the actor's mental state of significant traces resulting from public events. Observing the effects of his actions on the environment, the actor creates and stores new knowledge,

possibly integrating it with existing knowledge, or even revising or discarding old knowledge. In this phase, the actor's mental state is used as a knowledge *sink*.

- The **activation** of represented traces. Stored knowledge is brought up and used by the actor in order to behave rationally. In this phase, the actor's mental state is used as a knowledge *source*.

Note that the transition interfaces between the mental and public spaces are the actor's **sensors** (for seeing, hearing etc.) at the effects-representation interface, and its **effectors** (for moving, speaking etc.) at the activation-behaviour interface.

Learning is often defined as a relatively lasting change in behaviour, as a result of experience (see e.g. [Hilgard et al., 1975]). Learning is the main key to **adaptivity**: Using the terminology of the knowledge cycle, we may say that adaptive actors are those who use knowledge representation and activation effectively in order to increase behavioural accuracy in an ever-changing public environment, and who are able to revise their representations in accordance with the quality of the effects. How knowledge is represented and activated thus become the key factors for successful behaviour:

Representation

- Which public phenomena are selected for representation?
- How rich is the knowledge representation (how complex phenomena and relations can be represented)?
- Is knowledge stored interconnectedly or disconnectedly?
- Is new knowledge integrated with existing knowledge?
- Can knowledge be modified or criticised based on new and consistent evidence?

Activation

- Is relevant knowledge readily accessible in a given situation?
- Can retrieved knowledge be combined to suit a new situation?
- Is the activation mechanism fast enough to operate effectively within the given time window?
- Is there enough interconnected background knowledge to establish a context for the situation at hand?

2.1.2 Knowledge profiles

To help answer these questions, we propose the following more specific criteria for establishing whether a given encoding supports effective knowledge representation and activation, i.e. the criteria for good **knowledge encoding**.

We consider bipolar parameters for assessment of encoding types. We call the left pole **knowledge-centric** (with a strong bearing on learning or acting), and the right pole **information-centric** (unrelated to an actor's adaptive behaviour).

- **Subjective** vs. **objective**. Knowledge is always interpreted by an actor, involving a perspective, or a frame of reference. Information, on the other hand, can be said to exist independently of actors. To illustrate this, consider an ancient manuscript written in a hitherto undeciphered script. When scientists then decipher the script, the information content of the manuscript remains the same as when it was written, while lost knowledge is recreated, courtesy of an actor interpreting the information.

- **Fuzzy** vs. **exact**. An actor will often have less than perfect information about its environment. Useful knowledge representations should support non-measurable or limited information, as well as acting under uncertainty.

- **Associative** vs. **fragmentary** (mainly influences acting). Associativity is a key factor in how the human mind achieves effective knowledge activation. A single keyword may open doors to wide areas of long-disused knowledge. 'Relevance' as a term is less applicable to information than to an actor's purposive interpretation of it.

- **Goal-driven** vs. **neutral** (mainly influences acting). Representation and activation of knowledge is always driven by some goal, which an actor wants to accomplish. This has a direct influence on both *what* is stored and *how* it is stored.
- **Active** vs. **passive** (mainly influences acting). A knowledge representation causes problem solving, or other competent behaviour, to happen when the appropriate context occurs. A knowledge representation must support action relative to brief time windows. Information representations are passive in that they do not in themselves cause action.
- **Dynamic** vs. **static** (mainly influences learning). Knowledge representations get modified through being used. By formulating an answer or an explanation, you may trigger further reflection that adds new knowledge, even while your information remains the same. Using an information representation, e.g. a book, does not alter it.
- **Changeable** vs. **rigid** (mainly influences learning). Efficient learning exerts an evolution pressure on the represented knowledge, enforcing revision as new knowledge arrives. Merely adding information to already existing information is not an evolutionary process; indeed, this may even *hinder* the process of extracting knowledge because a large amount of non-integrated information becomes unwieldy in practice (information overload).
- **Adaptive** vs. **planned** (mainly influences learning). In the real world, unforeseen things happen. A good knowledge encoding should be open-ended and general enough to accommodate reasonable responses to changes in the environment.

Different representation formats may be characterised along these bipolarities, giving different profiles of knowledge-centricity. The human brain is an example of an ideal format, since it is the base case of knowledge representation and activation, supporting highly adaptive human behaviour:

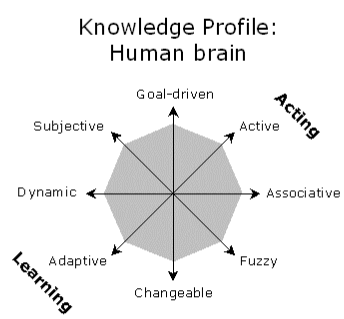

Figure 2: Knowledge Profile: Human brain

2.2 Knowledge in an organisational context

So, what do we mean by organisational knowledge? We believe that it is fruitful to generalise the knowledge cycle to entire organisations. In doing this, we see an organisation as an actor; we assume the **intentional stance** ([Dennett, 1971]) towards organisations.

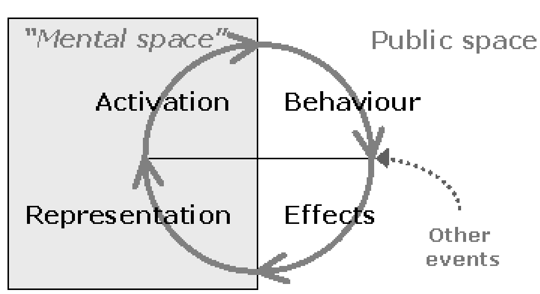

Figure 3: The Organisational Knowledge Cycle

This organisational level of description allows us to explore the intermediate functions of knowledge processes on a purely extensional basis. Our actor is the organisation, not the individual. An organisation will then be seen as trying to be a rational actor in its environment by exploiting its represented knowledge. Our focus is on specific representation formalisms, not on postulated private phenomena. Our pragmatic aim is improving the organisation's knowledgeable behaviour. The "mental space" of an organisation is a metaphoric notion, comprising all encoding media at its disposal: from archives and databases to buildings and personnel. The "mental" representations of the organisation, then, are indeed open to study and manipulation.

The individual employee is, perhaps, the most important knowledge source for an organisation. **Individual knowledge** is private; it resides in the heads of competent people within the organisation. **Shared knowledge**, on the other hand, is public, as it resides in repositories that can be activated by different actors during organisational problem solving.

2.2.1 Knowledge sharing

Facilitating the creation and deployment of individual knowledge in an organisation is a legitimate and important knowledge management concern, especially since all shared knowledge must have started its life as private. However, we think that where companies can *really* gain competitive advantage is by converting human capital assets to structural capital assets, thereby facilitating knowledge sharing. There are two main reasons for wanting this:

- Organisational knowledge assets may be **too valuable** to entrust to a single individual (ideally for no other reason than that the human body is not very durable).
- If different actors within the organisation are able to activate the same shared knowledge, you have a foundation for **spreading best practices**.

The rest of this paper will concentrate on **managing shared knowledge within an organisation**. The motivation for this is that we see severe shortcomings in current KM approaches, in this respect. Specifically, nearly all current KM efforts are information-centric rather than knowledge-/actor-centric: Their emphasis is on representation of data, with little thought for adaptive real-time activation of what is represented. A typical organisational knowledge cycle may look like this:

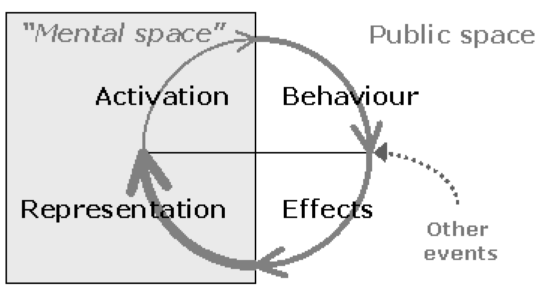

Figure 4: Unbalanced Knowledge Cycle in an information-centric organisation

Consider this quote from a practitioner in the field ([Sieloff, 1998]):

> *[...] The authors overlook one aspect that contributes to a knowledge management project's failure. The scarsest resource in the emerging knowledge economy is not knowledge itself, but people's attention: the limited capacity that we have to acquire, use or transfer available knowledge. [...] Too many knowledge management projects focus on increasing the supply of available knowledge rather than on increasing people's capacity to use that knowledge effectively. When the resulting repositories sit unused and the groupware applications die for lack of participation, the blame is usually assigned to some combination of corporate culture (knowledge hoarding, competitiveness, "not invented here" attitude), incentives (insufficient rewards for sharing or reusing knowledge), or technology (too difficult to use, too expensive). But the results of cultural assessment surveys within HP point to a different culprit: the lack of time available to deal with the overwhelming volume of knowledge. [...] Are knowledge management efforts helping people better use their limited attention capacity or simply adding another element to the many demands for their scarsest resource? [...]*

We believe that a key reason for these problems is that the knowledge encodings in dominant use within today's organisations are information-centric, and accordingly do not support knowledge activation well. To appreciate this, consider the following knowledge profiles:

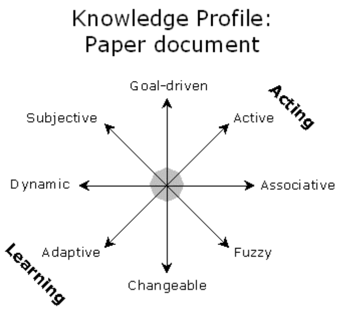

Figure 5: Knowledge Profile: Paper document

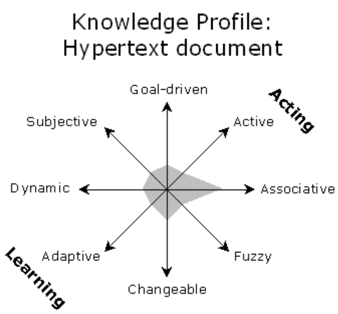

Figure 6: Knowledge Profile: Hypertext document

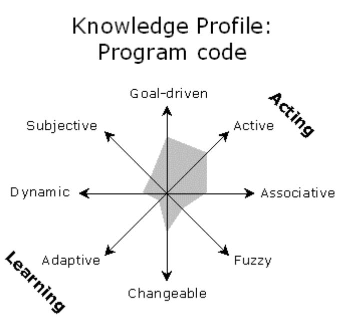

Figure 7: Knowledge Profile: Program code

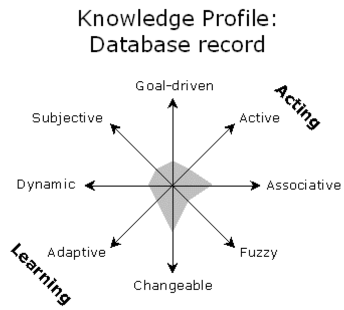

Figure 8: Knowledge Profile: Database record

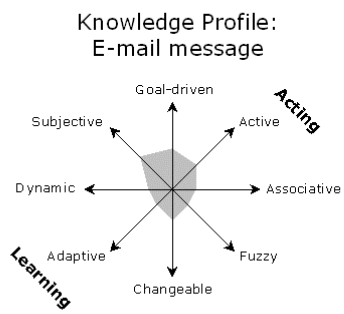

Figure 9: Knowledge Profile: E-mail message

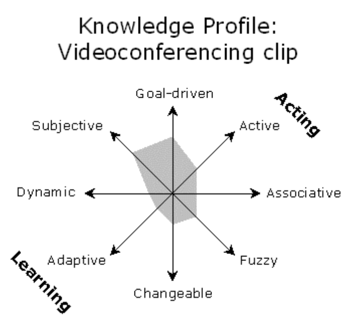

Figure 10: Knowledge Profile: Videoconferencing Clip

2.2.2 Active organisational memories

Generalising the knowledge cycle to apply to the aggregate organisational actor may give us a clearer perspective on what is actually referred to by such popular terms as 'learning organisation' and 'organisational memory'. To be successful in its environment, an organisation needs to establish a well-lubricated knowledge cycle, incorporating representation and activation of shared knowledge, together with efficient organisational sensors and effectors. This amounts to introducing technology into the organisation in such a way that the roles of human knowledge are carefully interwoven with technology-borne knowledge support, forming a knowledgeable socio-technical total system. This perspective gives new meaning to statements such as "the organisation is more than a sum of its individuals" (*yes, it exploits* *shared best practises*) or “the organisation has not realised its human potential” (*no, there are problems with knowledge representation, knowledge activation, sensors or effectors*).

Crucial for success, then, is a KM infrastructure for representing and activating shared knowledge (organisational sensors and effectors are covered by other management disciplines than KM). We will call such an infrastructure an **active organisational memory** (to reflect actor focus).

We will now turn to describing a real-life scenario where such an active corporate memory is proving itself extremely helpful.

3 DNV scenario

Det Norske Veritas (DNV) is one of the leading ship classification societies in the world. It conducts its ship classification activities all over the globe. DNV’s objective in the maritime business is to ensure the safety of ships world-wide. Let us briefly describe the ship classification business:

In order to operate their ships, ship owners are obliged by national authorities to obtain safety certificates for each individual vessel in their fleet. DNV is among those entities that issue these certificates. So, when a new vessel is built according to DNV classification rules, the vessel must be approved by DNV before it can sail. This is a systematic process in which virtually every piece of equipment must be approved, and the ship’s structure will be analysed in order to assess its strength.

During the lifetime of a ship, the owner must renew the certificates at regular intervals. Therefore, DNV will periodically perform on-board surveys of the ship in order to assess its condition.

DNV performs a highly knowledge-intensive business. Knowledge is the main product - knowledge about how to build and operate ships in ways that protect the environment, but still are both safe and economic for the owner. In addition, it is a highly competitive business, as well as a globally distributed one. Efficient distribution and sharing of relevant information and knowledge is critical to the business. Providing appropriate knowledge services as part of the IT solutions used in daily operation is key to enabling efficient knowledge distribution and sharing.

This scenario describes a knowledge-intensive, distributed work process; a DNV survey job. The purpose of the survey job is to renew certificates that are expiring. By doing so, DNV assures that the ship is still safe to operate. Operating a ship involves many highly specialised disciplines, and requires the combined knowledge of many people. So how is it that one DNV surveyor can survey the ship and DNV as a corporation can renew the certificates of the ship based on his judgements? The answer is a highly developed knowledge management practice with systematic knowledge sharing in the form of classification rules and experience notes.

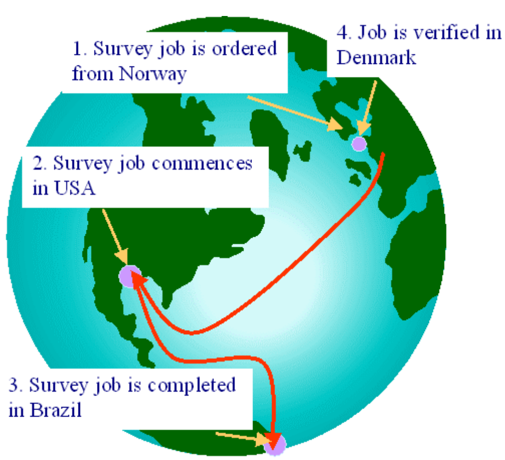

Figure 11: A DNV survey job

Knowing that the main class certificate of one of his ships soon expires, the ship owner contacts the DNV customer manager in order for DNV to renew this certificate for another five years. This results in an order for DNV to do a survey job on the ship. The survey will start the next time the ship is in harbour.

The DNV customer manager defines the scope of this job. Checking the schedule of all the ship's certificates, he finds that it would be wise to extend the survey in order to cover other certificates that approaches expiration. The ship owner approves.

Having set the scope for the job, the job is handed over to one of the surveyors at the DNV office closest to the harbour where the survey will take place. Prior to boarding, the surveyor must carefully plan the survey. The ship is large, and he is incapable of surveying all parts of it in a limited time. Given the scope of the survey, DNV classification rules define what parts of the ship should be surveyed and what its condition must be in order for the certificates to be renewed. Depending on the actual condition of the ship, other parts of the ship may be subject to inspection as well. For instance, the ship owner has recently reported minor damages on the ship's hull and some equipment on board that need to be checked. Furthermore, DNV has learnt that on this particular type of ship, there is at this point in the ship's lifecycle a high probability of significant corrosion, and subsequent weakness, in certain spots. This experience has been gained through analysis of data gathered from surveys on similar vessels.

While planning the survey, the surveyor needs to gather relevant information about the ship, drawings, notes from previous surveys, etc., which he brings with him as he enters the ship.

Once on board the ship, the surveyor starts executing the survey plan. The survey plan is a checklist, defining in detail what parts of the ship to survey. The result of each checklist item is recorded. If there is a finding, e.g. a damage, a "condition of class" may be ordered. The condition of class conditions a repair before the certificate is issued. Damages may be documented with photos, sketches or other relevant documentation.

Since the survey is quite extensive in scope, the ship has to sail before the survey is completed. Therefore, the survey has to be continued in the next harbour, and the job must therefore be transferred to another surveyor. Upon completion, the survey job is sent to a qualified person for verification and approval. Note that, in general, several people might be involved in job verification.

In this case, the job is rejected with comments, and sent back to the last surveyor for further revision. Finally, the job is approved, and the new certificates are issued to the ship owner.

The main products of DNV surveys are certificates, and the quality assurance process of these certificates must be performed by a qualified person. This means that every job will be approved or rejected based on human judgement; verification and approval can not be automated.

The wide distribution of work in this case is a big challenge for the DNV organisation; it implies that at any location in the DNV organisation, surveys will be conducted according to state-of-the-art theoretical knowledge as well as DNV's best practices in many different technical disciplines. The rest of the paper discusses how DNV applies technology to overcome these challenges.

4 Organisational knowledge technology

The vision, then, of a successful active corporate memory, where every organisational actor has access to best practice knowledge relevant to the task at hand, and where the task performed feeds into future best practices via the knowledge cycle, is a compelling one. Which technological tools are available in order to make this a reality?

Although there is currently a rapidly growing market for IT tools which are marketed as knowledge management solutions, much recent thinking in the field has warned against technology-driven KM and argued, rightly *per se*, that it is the people that matter (see e.g. [Davenport and Prusak, 1998]). There is, however, a risk that the pendulum will swing back too far; technology is one of the key driving forces behind changes in the competitive environment, and should not be dismissed. [Malhotra, 1998] is a constructive contribution to this debate. Malhotra argues plausibly (but unfortunately with excessive KM jargon such as "re-everything", "wicked environment", "intrapreneurship" etc.) that current knowledge management systems are mostly inadequate for supporting rapidly changing organisations. Using our previous terminology, their knowledge profiles are firmly rooted towards the information end of the scale. But then he goes further than just criticising; he sketches what should replace them. His suggestion is systems that act as “knowledge ecologies”:

- They should focus on **social networks** and **people** rather than IT networks and computers
- They should be **adaptive** rather than pre-programmed for only anticipated situations
- They should consist of differentiated **knowledge nodes** that (temporarily) collaborate via **knowledge flows** to accomplish specific goals.
- Knowledge in such systems should be tightly coupled to **actions** and **context**..
- They should thrive on knowledge **diversity**, and support both **collaborative** and **competitive** modes of operation.
- They should handle knowledge **dynamically** and **evolutionary**, creating, recreating and even destroying knowledge as necessary.

We would contend that there exist good candidate technologies for implementing such KM systems.

4.1 Knowledge engineering

From its inception, a prime concern in the field of artificial intelligence has been to represent knowledge in a format that is interpretable by computers. The AI community has a long tradition in investigating how to build symbolic reasoning systems ([Newell and Simon, 1961]).

The AI subfield that grew out of this tradition, **knowledge engineering**, concerns itself with building **knowledge-based systems**. To this end, it has developed several practical, mature formalisms for encoding knowledge.

A key idea here has been to see knowledge as a separate category of data to be manipulated upon by a specialised piece of software, a reasoning mechanism. A **knowledge base** is then the repository for knowledge data structures, while a **knowledge engine** is used to manipulate the contents of the knowledge base as new circumstances arise.

Current trends within enterprise software engineering emphasise the need for separation between domain-specific and system-specific components**:** **business logic** should be separate from **application logic** to facilitate rapid software adaptation to new business needs. This is usually done by keeping pieces of code dealing with the business domain isolated from those dealing with system infrastructure.

Knowledge engineering, however, goes a step further: Knowledge about the specific business domain is explicit as declarative data in knowledge bases, and not implicit in procedural code (note that the code for the knowledge engine itself should be independent of the business domain).

Making business knowledge explicit in this way carries several benefits:

- Explicit business knowledge data can be made available in a form that makes sense to non-programmers. This has wider implications than might at first be thought. It allows what may be called **knowledge stakeholders**, those people within the organisation who really know the business domain, to get actively involved in shaping the knowledge content within the technology solution at hand. Far too often nowadays, systems for IT support fail to deliver benefits because IT professionals make (the wrong) business decisions; there is a wide chasm between “high-level” organisational goals and “low-level” IT systems coding. In our view, this chasm must be bridged from both sides. Knowledge stakeholders need to get “down and dirty” in shaping fine-grained business knowledge in technology solutions, while software engineers need to incorporate support in their solutions for easily understandable knowledge formalisms which are free of purely technical details and considerations; in short, they need to become **knowledge engineers**. Streamlined interactions between knowledge stakeholders and knowledge engineers will breed better business support systems, and might even, given the goal-oriented and practical emphasis, lead to fruitful feedback for organisational rethinking.

- Additionally, explicit knowledge bases enable business knowledge, being data, to be changed without involving software engineering. Armed with the right tools, the appropriate knowledge stakeholder (a Chief Knowledge Officer, say) could be able to edit the knowledge base directly in response to a new business situation.
- The third point to make is that explicit, declarative knowledge is easily transportable. We will have much to say about this when discussing knowledge sharing, au-dessous.

There are several kinds of AI techniques that we think will prove useful from an organisational point of view:

- **Rule-based reasoning** (logical, rule-based knowledge). Ultimately based on mathematical logic, and refined through real-world experiences with expert systems, rule-based systems are perhaps the most widely used AI technique. 'If-Then' reasoning rules are useful for representing formal regulations and complex conditions involved in day-to-day business operations.
- **Case-based reasoning** (structural, example-based knowledge). Inspired by ideas on how humans learn by examples, CBR deals with storing specific solutions for problems already encountered, and using these as guidelines for solving new problems. A CBR approach is useful for storing and reusing organisational best practises on a fine-grained level.
- **Constraint satisfaction** (structural/logical, constraint-based knowledge). Used for expressing and solving problems where you have a composite structure, each part dependent in complex ways on other parts, CSP will be useful for organisations which deliver complex, component-based products (physical or informational) where individual components may change rapidly.
- **Planning and scheduling** (logical/practical, goal-based knowledge). Inspired by human practical reasoning, planning techniques are able to synthesise complex courses of actions for accomplishing given goals. This is useful in operational business situations, where the organisation needs to get something done in a resource-optimised manner.
- **Probabilistic networks** (practical, utility-based knowledge). Grounded in probability and decision theory, probabilistic networks embody algorithms for acting on uncertain information. This is useful for dealing with real-time/complex/incomplete organisational data flows where swift decisions are paramount.

In addition to these techniques from the AI field, we want to emphasise three other, primarily non-AI, technologies which can nevertheless be seen as legitimate technologies for knowledge engineering:

- **Object-oriented analysis/design** (structural, object-based knowledge). Although invented for software engineering purposes, we wish to treat an object model as just another knowledge base. This will be extremely useful for representing knowledge about the entities in an organisation's business domain.
- **Workflow technology** (procedural, process-based knowledge). This is technology for specifying and supporting explicit business processes. Again, we see this as a knowledge engineering technique. Workflow technology is useful for supporting organisational business processes that can be formalised to a large extent.
- **Online Analytical Processing/data mining technology**. This is technology typically used in data warehousing applications, and is valuable for extracting knowledge implicit in high volumes of data.

4.2 Knowledge sharing technology

In the scenario au-dessus we encountered a global organisation doing a job requiring many types of knowledge. Accordingly, knowledge engineering for such an organisation involves two additional challenges:

- Support for **knowledge distribution**. Knowledge might be created, stored, modified and used by different actors in different physical locations. How to ensure that organisational knowledge is decentralised and mobile, yet globally accessible?

- Support for **knowledge interoperability**. For an organisational actor, heterogeneous knowledge might be needed to support rational action. How should the interfaces between different sorts of knowledge be realised?

We use **knowledge sharing technology** to accomplish these two things.

Knowledge sharing technology ([Genesereth and Ketchpel, 1994]) is a discipline within the larger (and still somewhat unsettled and hype-ridden) field of agent technology. It deals with the challenge of constructing a **distributed, knowledge-based system** by having **software agents** as basic software components in a distributed system. Each agent will be the custodian of a **virtual knowledge base** (VKB), so called because whether realising it as a traditional knowledge base or not is a choice left to the implementor. This approach is clearly a descendant of work done on co-operative problem solving within the distributed AI (DAI) community ([Bond and Gasser, 1988]). Such knowledge-based agents will communicate and collaborate via a well-defined **agent communication language** (ACL).

For knowledge sharing purposes, the criterion for being a software agent is just that of being able to communicate in an agent communication language. An ACL is a language and protocol for knowledge-based messaging. It is inspired by speech act theory ([Searle, 1969]), and consists of two main parts: an inner **semantic** core language for **expressing** knowledge, and an outer **pragmatic** wrapper language for expressing an **attitude** towards the knowledge that is being communicated. Thus, an agent might *ask* (attitude, wrapper) another agent if <something> (knowledge, core) is the case, *tell* another agent about <something> being the case, command another agent to *achieve* <something>, *advertise* its willingness to be *ask*ed about <something>, or even *subscribe* to another agent's *tell*ing when <something> has changed. A main idea here is that the semantic core (message "body") is opaque to the pragmatic wrapper (message "heading"); an ACL is content-neutral in that it doesn't mandate what type of knowledge is communicated.

An agent-oriented, distributed knowledge-based system, in sum, will consist of:

- An **agent platform** for deploying agents, including agent lifecycle components, network transparency components, agent messaging infrastructure components, "white pages" components (for finding agents) and "yellow pages" components (for publishing and finding knowledge-based agent services).
- A well-defined ACL **message catalogue** detailing which speech acts are available at the pragmatic level; what you can do with the contents of knowledge bases. The two ACL standards that are currently available differ somewhat in their emphasis here: FIPA ACL has strong protocols for negotiation- and market-based speech acts, while KQML is strong on collaborative reasoning speech acts.
- A well-defined ACL **message content ontology** that describes the knowledge formats available at the semantic level; how to express contents of knowledge bases in an interchangeable manner. This amounts to establishing an agent-shared protocol for declaratively expressing such things as rules, object models, processes, plans etc. A promising format for doing this is XML (together with one of its related initiatives for supporting standardised semantics).
- The **software agents** themselves, which will then be able to interoperate transparently across the network on the knowledge level ([Newell, 1982]) by exchanging well-defined ACL messages.

In addition to addressing knowledge distribution and interoperability, adopting knowledge sharing technology has some "incidental" software engineering benefits:

- **Actor-centric abstraction**. Agents can be viewed as full-blown actors (with their own "individual" knowledge cycles) in the active organisational memory (we then adopt the intentional stance towards software entities). Our software engineering entities and abstractions will then align nicely with the organisational entities and abstractions the system should support.
- **Implementation neutrality**. Knowledge sharing technology forces us to deal with knowledge in an explicit, declarative manner (which we have already argued is a good idea). Having gone to this trouble, however, we find ourselves in an interesting software engineering position: Since ACL messages are basically ASCII strings, the only technical constraint on implementing (new) agents in the system is that they be able to send and receive strings over the given network infrastructure.

5 Technology-supported DNV scenario

DNV has a history of applying new technology as a strategic means of improving its business processes. Now, let us outline the **Nauticus** vision and how it contributes to process improvement and knowledge management in DNV:

Nauticus is an active organisational memory, a distributed, knowledge-based systems built on top of a modern, component-based distributed architecture (Windows DNA).

The core of Nauticus is the **product model**, a comprehensive, distributed, “live” object model which serves as an ontology (ships in general) and a knowledge base (specific ships) for knowledge about the condition of each vessel for which DNV has an engagement. This includes all parts and aspects of the ship that are relevant for ship classification: a detailed 3D model of the ship, its parts, damages, survey plans, etc., with all kinds of related information (drawings, calculation spreadsheets, documents, etc.). The main difference between the product model and a traditional database is that the product model makes its knowledge content explicitly available. More specifically, in the parts of the domain where there is an ontology, each object in the product model will publish references to the ontology. This makes the product model the most important knowledge base of DNV, because it acts as the main reference for other knowledge sources.

In addition to the product model, there are knowledge bases of a more task-oriented nature: process bases, rule bases, case bases, constraint bases and plan bases. Knowledge base distribution and interoperability is achieved by putting each knowledge base inside an agent; agents communicate and collaborate asynchronously by exchanging ACL messages.

Here is a simplified sketch of the Nauticus layered architecture:

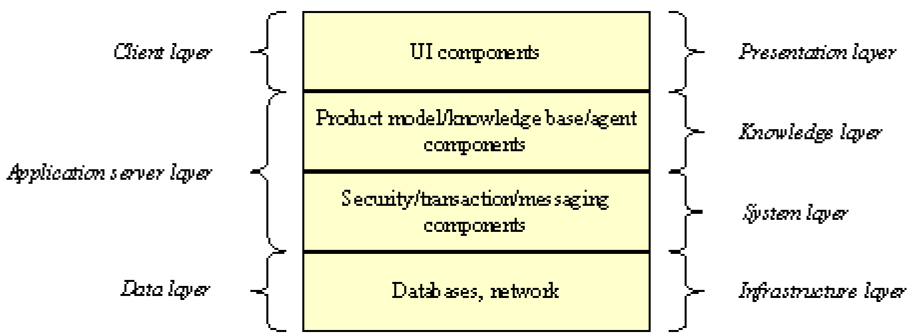

Figure 12: Nauticus layered architecure

Now, as knowledge flows into the product model, it becomes an invaluable source for systematic analysis by the other, task-oriented knowledge bases, enabling DNV to feed back operational experience - and finally to improve ship design.

The result of this is organisational process improvement: learning how to make future ships safer.

The DNV survey job involves collaboration between at least six roles that are played by one or more **persons** in the scenario:

- **Ship Owner** role**.** For the purpose of this scenario, this is the ship certificate customer.
- **DNV Customer Manager** role**.** For the purpose of this scenario, this is the certificate supplier.
- **DNV Certificate Authority** role**.** This is DNV’s certificate issuer.
- **DNV Survey Authority** role**.** This is DNV’s internal survey verifier. Traditionally, this has been the DNV station manager at the DNV office where the survey takes place. However, with the appropriate technology in place, any person in DNV may take on this role provided he possesses the right skills.
- **Surveyor** role**.** Persons who do onboard inspections fill this.

- **Ship Operator** role**.** For the purposes of this scenario, this is the role that needs to be alerted before an onboard survey.

(Note that roles are identified to pinpoint areas of organisational responsibility. There is not necessarily a one-to-one correspondence between a person and a role; several persons may play a role, and a person may play several roles. A common case of the latter within DNV is that a person with the title 'Customer Manager' plays both the Customer Manager and the Certificate Authority role.)

The sequential interactions between the roles in the scenario can then be illustrated in this manner:

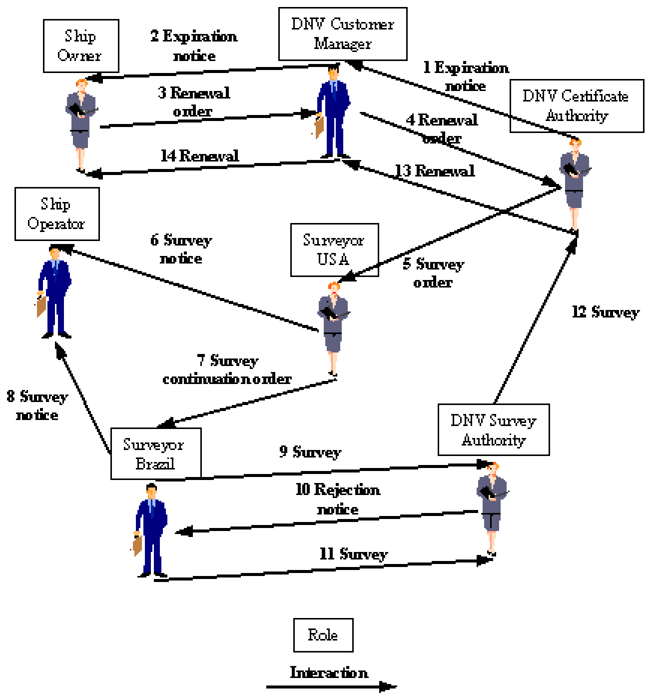

Figure 13: Interactions between human roles in the survey job scenario

To support those who fill the Customer Manager, Certificate Authority, Survey Authority and Surveyor roles in DNV's day-to-day survey jobs, and ensure end-to-end quality through best practices, we give them access to Nauticus. If desirable, the non-DNV roles (Ship Owner and Ship Operator) can also be given access to the system; it is an active memory for virtual organisations as well.

Agents will then assist end-users of Nauticus in representing and activating knowledge.

(Note, however, that this agent assistance isn't of the "digital assistant" kind, as seen in e.g. Microsoft's "Office assistant" technology. Nauticus agents are invisible background mechanisms; whether to "agentify" the user interface is a separate issue.)

In order to implement end-user knowledge assistance, we find it useful to distinguish between **role agents** and **knowledge agents**:

- A knowledge agent is responsible for representing and activating a particular type of knowledge needed by a particular role for accomplishing a particular task. Knowledge agents are the base mechanism for knowledge distribution.
- A role agent is responsible for assisting an organisational role through the whole of its interaction with the active organisational memory, delegating to, and assimilating from, heterogeneous knowledge agents for particular needs. When roles collaborate, their role agents communicate. Role agents are the base mechanism for knowledge interoperation.

The idea is that a human user that is playing a certain organisational role is assigned a specific role agent, which tracks the user's context throughout the interaction with the system. That agent, in turn, will invoke the appropriate knowledge agents at any given time to provide the user with **just-in-time knowledge support**. It is just this which makes the organisational memory **active**. An organisational role that needn't be filled by a human can be filled by a role agent instead: The role is automated; the agent supports itself.

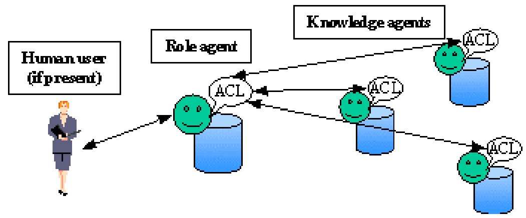

Figure 14: Just-in-time knowledge support

Let us now consider knowledge assistance for the knowledge-intensive survey job roles (DNV Certificate Authority, DNV Survey Authority, Surveyor) in turn, referring back to the human role interactions shown previously. Note that the focus has moved from interaction to action; how a particular role is supported actively at task execution.

In our previous figure, however, we left out one all-important organisational role: the DNV Ship Structure Authority, which is responsible for knowledge concerning general and specific ship structures.

The reason for this "omission" is simple: As said before, the core of Nauticus is the product model, an extensive object model concerning ships. In fact, so much organisational knowledge is encoded here that for certain well-defined purposes, including survey jobs, this role can be automated; no human involvement is necessary, at least in non-exceptional situations.

In the job survey scenario, nevertheless, this role is involved at nearly all stages, since knowledge about ship structure is a prerequisite for both certification and surveying activities. Let us therefore start by describing knowledge assistance for ship structure.

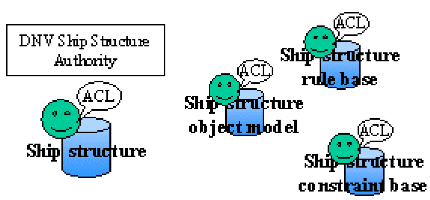

Figure 15: Knowledge assistance for ship structure

When knowledge about ship structure is needed (as when the surveyor is planning an on-board inspection of a particular ship between interactions 5 and 7, or when a survey is verified between 9 and 10), other role agents can ask the ship structure role agent to activate knowledge about ship layout (the object model agent), ship construction regulations (the rule agent) or ship design constraints (the CSP agent).

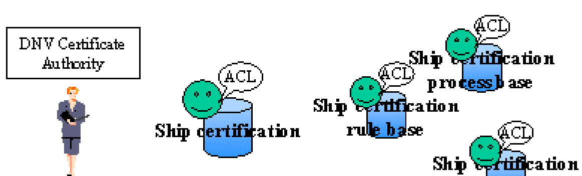

Figure 16: Knowledge assistance for ship certification

The ship certification agent will monitor existing electronic ship certificates, and notify the appropriate Customer Manager when they are about to expire (interaction 1). When the Certificate Authority receives a renewal order (4), the certification agent will assist in setting up the re-certification workflow by contacting the certification process agent. One step in the certification process is ordering a ship survey from the appropriate Surveyor (5). When a verified survey is returned (12), the certification agent will assist the user in controlling that all certification criteria is met by communicating with the certification rule agent. The certification CBR agent is available for auxiliary purposes, if the Certificate Authority should take an initiative to compare with previous certifications. When a certification is renewed (13), it will be recorded by the certification CBR agent.

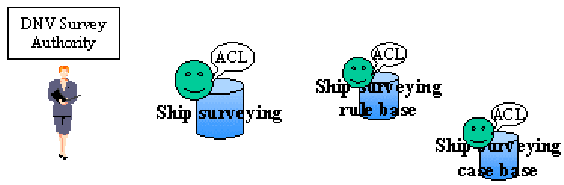

Figure 17: Knowledge assistance for ship surveying

The ship surveying agent provides background knowledge for supporting the Survey Authority in his decision-making: When he considers a survey (after 11), it assists in looking up rules and previous cases; when he finally verifies the survey (14), it is recorded in the case base.

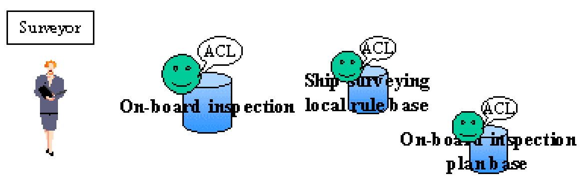

Figure 18: Knowledge assistance for on-board inspection

The surveyor's job is the most knowledge-intensive of all. When planning an on-board inspection (after interaction 6), or finishing the survey after the on-board visit (before 13), the on-board inspection agent helps by taking into account both general surveying rules and previous survey cases (via the agent belonging to the Survey Authority), as well as any site-specific regulations from the local rule agent. If inspection time is sparse, the on-board inspection planning agent will be asked for help in scheduling the resources effectively.

6 Conclusions

In sum, we have shown how knowledge engineering and knowledge sharing technologies contribute to realising an active organisational memory, with a well-rounded knowledge profile suited for:

- **Just-in-time knowledge support** - putting task-specific global best practices at employees' fingertips (organisational acting)
- **Just-in-place knowledge management** - ensuring that shared knowledge is deployed and changed locally by its stakeholders (organisational learning)

7 Acknowledgements

Elling Rishoff, DNV, has provided invaluable directions in preparing this paper.

8 Authors

Are Sørli, Senior Knowledge Engineer, Computas

Responsible for knowledge sharing technology design and implementation in Computas' FrameSolutions™ frameworks. Has done knowledge and software engineering for several Computas solutions.

Gunnar John Coll, Chief Engineer, Computas

Responsible for workflow studies, knowledge analysis, system design and human factors supervision in numerous projects carried out by Computas for DNV and other clients. Participated in several knowledge management and human-computer interaction research projects.

Einar Dehli, Vice President, Computas

Co-founder of Computas. Currently Vice President and Process Owner Technology and R&D. Has broad experience from technical and management positions in a variety of knowledge management-related projects.

Kjell Tangen, Chief Engineer, Computas

Responsible for the software architecture of Nauticus. Has worked with software architecture, distributed component technology and model-based knowledge systems in a wide range of engineering domains: offshore, process, space and marine industries.

Computas was established in 1985, and has grown to become the leading supplier of technology solutions for knowledge management in Scandinavia. Customers include large industrial enterprises, government agencies and international organisations. Our mission is to capture and represent the knowledge inherent in the tasks and processes of the organisation, and deliver working solutions that bring the knowledge to life. Computas FrameSolutions™ is our unique architecture and technology platform, accumulating years of experience with practical application needs and state-of-the-art tools and techniques from the fields of artificial intelligence, object technology and user interaction.

9 References

[Bond and Gasser, 1988]

Bond, A. H. and Gasser, L. (eds.). Readings in Distributed Artificial Intelligence. Morgan Kaufman Publishers, San Mateo, California, 1988.

[Davenport and Prusak, 1998]

Davenport, T. H. and Prusak, L. *How organizations manage what they know*. Harvard Business School Print, Boston, Massachusetts, 1998.

[Dennett, 1971]

Dennett, D. C. Intentional Systems. *Journal of Philosophy*, 8, pp. 87-106.

[Genesereth and Ketchpel, 1994]

Genesereth, M. R. and Ketchpel, S. P. Software Agents. *Communications of the Association for Computing Machinery*, 37(7), pp 48-ff.

[Hilgard et al., 1975]

Introduction to Psychology. Harcourt Brace Jovanovich inc., New York, 1975.

[Malhotra, 1998]

Malhotra, Y. Toward a Knowledge Ecology for Organizational White-Waters. Keynote Presentation at the *Knowledge Ecology Fair 98: Beyond Knowledge Management*, February 2 - February 27, 1998, Sponsored by @BRINT, Dow Chemical, Fast Company, Hewlett-Packard, Intel, Monsanto, Skandia, and Society for Organizational Learning (Currently available on the World Wide Web at the URL http://www.brint.com/papers/ecology.htm).

[Newell, 1982]

Newell, A. The Knowledge Level. *Artificial Intelligence*, 18(1), pp 87-127.

[Newell and Simon, 1961]

Newell, A.and Simon, H. A. GPS, a program that simulates human thoughts. In Billing, H., editor, *Lernende Automaten*, pp. 109-124, R. Oldenbourg, Munich, Germany, 1961.

[Russell and Wefald, 1991]

Russell, S. J. and Wefald, E.H. *Do the Right Thing: Studies in Limited Rationality*. MIT Press, Cambridge, Massachusetts, 1991.

[Searle, 1969]

Searle, J. R. *Speech Acts: An Essay in the Philosophy of Language*. Cambridge University Press, Cambridge, 1969.

[Sieloff, 1998]

Sieloff, C. Letter to the Editor. *Sloan Management Review*, 39(3), MIT, Cambridge, Massachusetts, 1998.

[Skinner, 1953]

Skinner, B. F. *Science and Human Behavior*. The MacMillan co., New York, 1953.

[Stich, 1983]

Stich, S P. *From Folk Psychology to Cognitive Science – The Case Against Belief*. Bradford Books, Cambridge, Massachusetts, 1983.