The Knowledge Factory –

A Generic Knowledge Management Architecture

H.J. Müller
Deutsche Telekom AG
Technologiezentrum Darmstadt FE14k
D-64307 Darmstadt
Germany

A. Schappert
Siemens AG
ICN NT 13
D-89037 München
Germany

ABSTRACT

Knowledge management in enterprises often reduces to the application of a single data base with technical or personal data in order to produce some useful information for a certain task. However, analyzing the requirements of problems CEOs have when they like to apply knowledge management as a technology leads to the fact that the terms data, information and knowledge are used synonymously, that there is usually more than one source from which the "useful information" is extracted, and that there is no architectural structure which may be used to describe neither the requirements nor the realization of the problem.

A generic architecture will be presented which is based on the semiotic paradigm of information theory. The formal framework allows an adaptation of the architecture to special realizations and as such it covers standard information systems and data base application systems. The architecture will be the kernel the metaphorical description of a knowledge factory an may be enhanced with a collection of helpful software agents.

> *"Knowledge management is not a product in itself, nor a solution that organizations can buy off-the-shelf or assemble from various components. It is a process implemented over a period of time, which has as much to do with human relationships as it does with business practice and information technology"* (Benjamins, Fensel, Perez 1998)

DATA, INFORMATION, AND KNOWLEDGE

One major problem with knowledge management is the fact that despite of the intensive academic discourse on the terms data, information, and knowledge, in industrial practice they are used in an uncoordinated way. In the classical

interpretation data is associated with syntax, information corresponds to semantic and knowledge takes the pragmatic part. I.e. *data* per se has no meaning and may be seen as raw material for information. *Information* is context sensitive and meaningful in the sense that it is interpreted data. Since context is user (application) dependant information then may be enhanced by its use, i.e. the pragmatic. *knowledge*.
The semiotic correspondence of data, information, and knowledge thus interprets information as being the result of the transmitting knowledge and data as being the result of gathering information.

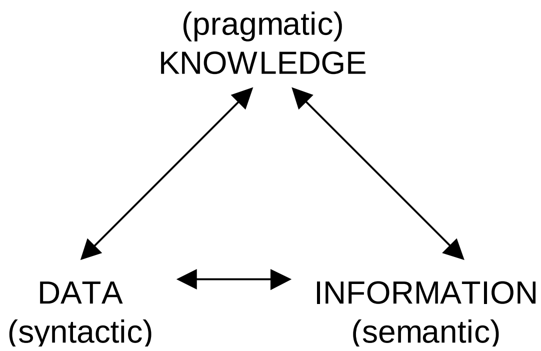

Figure 1a: The semiotic triangle

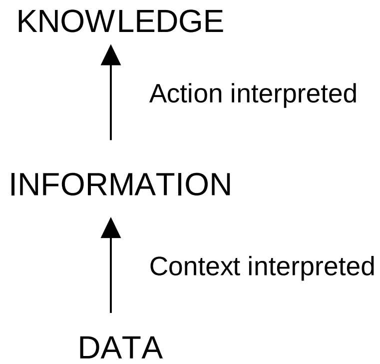

Figure 1b: Knowledge evolution

Turning the direction of reasoning leads to recent action oriented interpretations. According to (Nonaka 1994) knowledge is justified belief (i.e. information) that increases an entity's capacity for effective action, while information is the flow of messages or meaning which may add to, restructure, or change knowledge (Probst, Raub, Romhardt 1998). In that sense information is raw material for production of knowledge and information transforms to knowledge in the context of actions.

However, it would be wrong to imply a pure set inclusion between the three, i.e. knowledge is a subset of information which is a subset of data. Information may consist of many data items and knowledge may consist of information plus action rules.

- An example may be digital pictures: While on the data level only bit streams are represented the information level may contain additional format descriptions (especially those which identify the data as being a picture). Several and different information may be derived from the same data. On the knowledge level there may be semantic descriptors identifying the type of the picture (e.g. a landscape). Now searching for landscape pictures in the data base would have no result. The information system may select pictures from the data base and only on the knowledge level a landscape painting could be distinguished from a portrait.

It is crucial to note that knowledge emerges from using information, while information just adds additional items to data. In the example it might be argued that the type of the picture is just another meaning extending feature of the data, thus not knowledge but more detailed information. The important point is that something has to be done with the information, i.e. there has to be activity triggered or controlled by the information, in order to come to the knowledge level. Only since it is “known” that the class of pictures is divided into landscape, portrait, still life, abstract etc. it is possible to “use” the information in an adequate way and to reason about it.

- Suppose in the example we would know that trees often are typical for landscape pictures and that they are usually not part of a portrait. Further we know that wood consists of trees, and painting is a special word for picture. Then it would be possible via ontological reasoning to request for wood paintings. In the extended example information about the structure and meaning of expressions was used. Again, only the application of the information bridges the gap towards knowledge.

Ontologies play a major role in this process and nowadays knowledge management systems (Bejamins, Fensel, and Perez 1998) (O’Leary 1998). They describe an abstract model of some subject by explicitly defining relevant concepts and constraints between them. The model is accepted by experts of the subject and it is processable by machine.

A GENERIC KNOWLEDGE MANAGEMENT ARCHITECTURE

Enterprises are recognizing that the enterprise knowledge management rather than information gathering and data collection is becoming one of their main business factors. Total Quality Management and Business Process Reengineering support the companies to produce better products and to become more effective. However, these activities are usually not based on the enterprise’s experience and especially they do not support the talents of their best performers. Closest to knowledge management is the use of customized OLAP (Online Analytical Processing) tools to support planning activities. However, OLAP systems operate on large data bases tying to solve multi-dimensional requests for marketing, finance, and quality requests. Concerning the discussion in the last section, this means that information is generated out of data. The resulting information gives rise to (knowledge based) decisions made by human planners. In some cases expert systems are placed on top of OLAP tools in order to realize management support systems. If the expert system took care of using the companies expertise and practices, then it is a vertical knowledge management system in the sense of (Bejamins, Fensel, and Perez 1998). In the following we are interested in defining a horizontal knowledge management system which in contrast is not designed for a special business situation, but usable for different settings.

The Idea

The aim is to develop a generic architecture for knowledge management systems and processes which should

- respect the differentiation of data, information, and knowledge
- be used as a scheme to classify various types of enterprise business systems and knowledge processes
- support the flexible, though system-consistent modeling of knowledge management systems

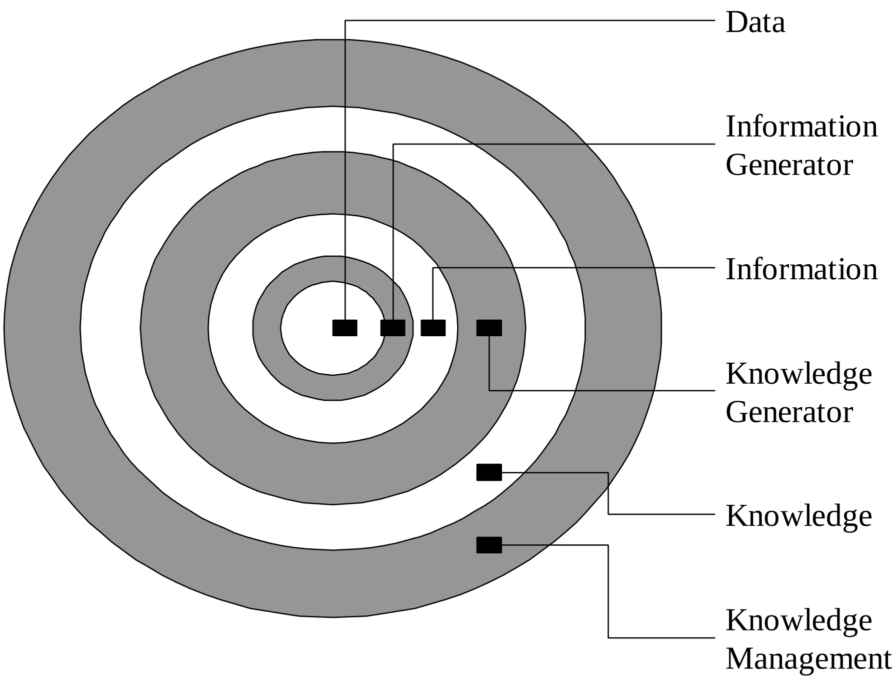

Figure 2: Knowledge Management Architecture Visualization

To visualize the knowledge management architecture the picture of an onion might be taken. It consist of circles which contains either container of material or tools to produce more complex material. To be more precise, the container are data-, information-, and knowledge bases. The tools are systems which use for instance data to produce information, like OLAP systems as discussed above. Cutting a piece from the center to the outside would then represent a specific knowledge management system, while the whole structure would represent the knowledge management facilities of an entire enterprise.

Formal Description

Though the general structure of the architecture is rather intuitive a formal definition is necessary in order to meet the consistency requirement. The kernel elements of the architecture are:

- A simple *data object* **sdo** is a name (sequence of ASCII-characters with a leading letter) or a number
- A data object **cdo** is either a simple data object or a list of data objects.
- A data base (**Dat**, **DS**) is a set of data objects **Dat** together with a structure description **DS**.
- An information object (**cdo, c**) is a data object together with a context description **c.**
- An information base (**Inf, Cxt**) is an data base ((**Inf, Cxt**), **DS**) where the data objects are information objects.
- A knowledge unit (**i, a, g**) is an information **i**, which is used to apply action **a** with goal **g**.
- A knowledge base (**Inf, Act, Gol**) is an extended information base ((**Inf** x **Act** x **Gol**), **Cxt**) where the information objects are knowledge units.

The definitions reflect the structure described in the previous sections. A data base distinguishes from a set of data by the structural description (i.e. the data model or data structure). Information objects are context extended data objects and knowledge

units couple information wit goal directed actions. It is notable that there might be identical data objects which, in different contexts, refer to different information objects. Hence, there is no one-to-one correspondence between data and information. The same is true for the knowledge where the identical information may be used in different actions or with different goals.
The defined structures define a hierarchy by construction.

Proposition1: Each knowledge unit (**i, a, g**) in a knowledge base **KB** is based on an information of an underlying information base **IB**, i.e.
$\forall(i,a,g) \in KB \quad \exists IB : i \in Inf.IB$

Proposition2: Each information object (**cdo, c**) in an information base **IB** is based on a data object of an underlying data base **DB**, i.e. $\forall(cdo,c) \in IB \quad \exists DB : cdo \in Dat.DB$

- A Knowledge Management System Kernel is a triple (**DB, IB, KB**) consisting of a data base **DB**, a Information base **IB** and a knowledge base **KB**.

Lemma: Each Knowledge Management Kernel has a hierarchical structure.

Given the basic components of a knowledge management system it is now possible to define operators on the elements. The idea is to define the use of tools which generate data, information and knowledge from the different sources. The definition is generic in that any transformation instrument may be used as an instant of one of the operators.

- Let HS be a hierarchical structure (*low, high*). Four sets $Local_l$, $Local_h$, *Lift, Combi* of possible operators are defined as follows:
  - Bottom-manipulator: $\forall f \in Local_l : f \in (low \rightarrow low)$

- Top-manipulator: $\forall g \in Local_h : g \in (high \to high)$
- Mutator: $\forall h \in Lift : h \in (low \to high)$
- Combinator: $\forall k \in Combi : k \in (low \times high \to high)$

- Two hierarchical structures HS1=(*low, mid*) and HS2=(*mid, high*) are compatible, if the Top-manipulators of HS1 are the identical to the Bottom-manipulators of HS2, i.e. $Local_l\ HS2 = Local_h\ HS1$

Note that the elements *mid* in both structures are identical by name.

Now, how does the abstract definition would apply to knowledge management scenarios. Intuitively all local operations on one level like read, write, and delete on a data, information, and knowledge base are covered by $Local_l$, $Local_h$. *Lift* contains operations which perform the transformation of a data object to an information object and an information object to a knowledge unit as indicated by the respective definitions. Data mining tools are typical representatives of those tools. The operations in *Combi* are more complex. They take non-interpreted data together with information objects and generate new information. This is mostly the case in enhanced OLAP systems. On the information/knowledge level they compute new activities based on known activities plus information. This is mostly the case in management decision support systems, where business actions are proposed on the basis of given information and descriptions of typical business activities.

THE KNOWLEDGE FACTORY

The architecture introduced in the previous section uses objects and methods or from the view of abstract data types data and operations in the traditional sense. Since the knowledge management architecture should be used in different contexts and by various people, it would be worthwhile to extend the presentation by incorporating the "agents" who will use the tools. Hence, there is a change from the rigid architectural description to a more vivid picture which we call the "knowledge factory". It extends the traditional view of having material and tools to work on in a natural way. Like in a factory beside the production there are the people who produce. In our scenario these are the "knowledge workers" and they will be incorporated into the framework by adding one more dimension. The following figure shows the new structure of the knowledge factory.

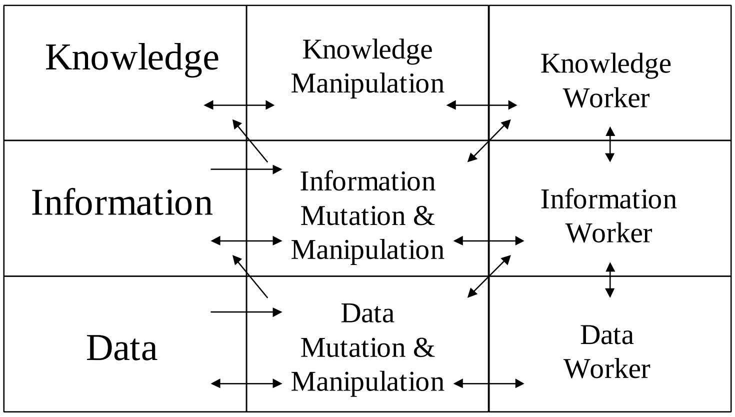

Figure 3: The Knowledge Factory Structure

The first column shows the hierarchical structure of the different types of basic objects: Knowledge bases are built on information bases which are built on data bases. With each level are associated the operator or tools used to work on the basic objects. Connecting the two columns with the arrows mirrors the simple parts of the operator definitions, namely *Local* (forth and back on the same level) and *Lift* (diagonal up). The *Combi* operators are implicitly presented with the third column. It represents the worker who use the tools of the second column. So the information worker applies tools of the data level and tools of the information level in order to produce new information or knowledge units. In contrast to the first and second column there is also a cooperation between the workers. Notice that we choose a hierarchy respecting model, i.e. it is not allowed to skip a level neither vertically nor horizontally. It may be a matter of discussion whether this strict proceeding is necessary. However, theoretically all missing cases can be constructed by combining the possible activities and on the practical side it is more secure if not everyone can do everything.

VARIATIONS

The knowledge factory structure may be used to describe standard solutions on one hand and advanced proposals on the other. In the following three scenarios will be discussed in order to demonstrate the generic power of the model.

OLAP systems may be interpreted as systems which generate information and in advanced cases knowledge on the basis of a database. The knowledge is the used by the management or the strategic department of a company to define future activities. Thus, the following structure may be extracted from the general model:

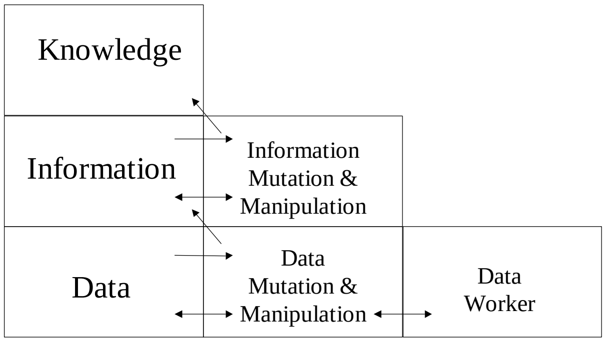

Figure 4: The OLAP variation of the knowledge factory

Worthy to note is that the OLAP control system it is represented as the data worker.

Another variation which may be easily constructed is a basic information system model consisting of the database and the information base together with the respective tools. It results in the following structure:

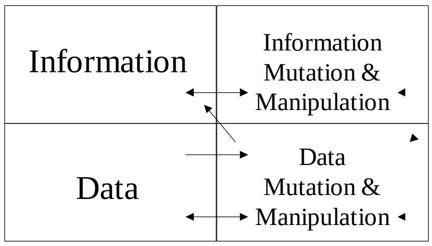

Figure 5: Classical information systems in the knowledge factory model

While the two scenarios demonstrate how existing system architectures may be easily generated from the entire knowledge factory model it will be shown in the following that the model serves also well in a scenario with distributed functionality. Of course, the model abstracts from data (information, knowledge) decomposition because at any level of the first column it is possible to have multiple entities. For the same reason it abstracts from functional decomposition as there may be a set of tools on each level at the second column. Thus it is possible to describe the whole IV-Structure of a company within the factory model. However, in mirroring the natural distribution of departments as would be inspired by the natural division of a company it is necessary to copy the factory model and to introduce the explicit interchange via the workers. This scenario is presented in figure 6.

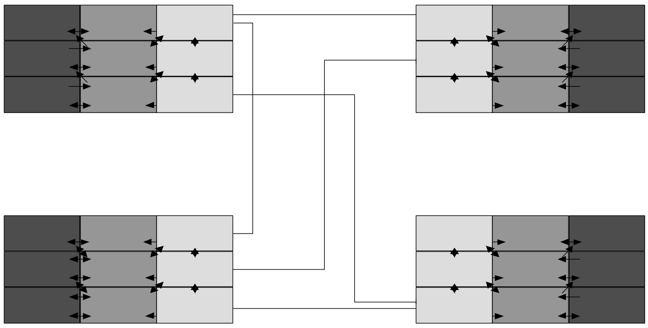

Figure 6: Four cooperating departments in the knowledge factory model

DISCUSSION

A generic architecture for knowledge management has been presented. Its primary features are: (1) that it is grounded on the basic insights of information theory respecting the division of data, information, and knowledge. (2) that it is multi-dimensional by associating tools and workers to each level. (3) that it offers clear interfaces to the different levels and dimensions through an abstract set of objects and operators. (4) that it is flexible enough to represent existing system architectures. (5) that it is component-compliant in the sense that it may be duplicated and connected. (6) that it is extendable to the incorporation other technologies like mobile agents as active workers.

REFERENCES

Benjamins, V.R., Fensel, D., Perez, A.G. (1998): Knowledge Management through Ontologies. In: Proceedings of the 2nd Int. Conf. On Practical Aspects of Knowledge Management (PAKM'98), Okt. 1998

Nonaka, I. (1994): A dynamic theory of organizational knowledge creation. In: Organization Science 5:1, pp. 14-37.

Probst, G., Raub, S., Romhardt, K. (1998): Wissen Managen. FAZ/Gabler.

O'Leary, D. E. (1998): Using AI in Knowledge Management: Knowledge Bases and Ontologies. IEEE Intelligent Systems, May/June 1998, pp. 34 – 39.