A Knowledge Development Conception and its Implementation: Ontology Categories, Knowledge Ontology, Rule System and Application Scenarios

Eckhard Ammann
School of Informatics
Reutlingen University, Reutlingen, Germany
Eckhard.Ammann@Reutlingen-University.de

Ismael Navas-Delgado, José F. Aldana-Montes
E.T.S.I. Informática
University of Málaga, 29071 Málaga, Spain
{ismael, jfam}@lcc.uma.es

Abstract**-- Knowledge development in an enterprise is about approaches, methods, techniques and tools, that will support the advancement of individual and organizational knowledge for the purpose of an improvement of businesses. An approach for knowledge development in a company is described in this paper. This approach is based on a new conception of knowledge, with the introduction of three knowledge dimensions and conversions between knowledge assets. To guide a formalized implementation of this conception in the form of a knowledge ontology, we discuss principal ontology categories of things and adapt them to the domain of knowledge development and management. Thus, we can take advantage of reasoning and rules processing, provided by a reasoner in combination with a rule engine. Important scenarios for knowledge development in a company are identified and it is shown how these scenarios can be supported by processing the developed rules. For example, knowledge requirements for a new or existing employee can be gained once the appropriate requirements for a planned project are known as well as the learning options in the company.**

Keywords-- *Conception of knowledge, knowledge development, ontology categories, knowledge ontology, rule system, application scenarios.*

I. INTRODUCTION

This is an extended version of a conference paper [1].

Knowledge development in an enterprise is about approaches, methods, techniques and tools, that will support the advancement of knowledge for the purpose of an improvement of businesses. This notion includes as well individual knowledge as group and organizational knowledge. It can be seen as integral part of knowledge management; see [2], [3] and [4] for a description of several existing approaches for knowledge management. One specific approach for enterprise knowledge development is EKD (Enterprise Knowledge Development), which aims at articulating, modeling and reasoning about knowledge, which supports the process of analyzing, planning, designing, and changing your business; see [5] and [6] for a description of EKD. EKD does not provide a conceptual description of knowledge and knowledge development.

In this paper, we present a new conception of knowledge and knowledge development and describe an implementation of this conception based on a knowledge ontology, reasoning support and a rule system.

For the conception part, there exists one well-known approach by Nonaka/Takeuchi [7], which is built on the distinction between tacit and explicit knowledge and on four knowledge conversions between the knowledge types (SECI-model). However, is explicit knowledge still bound to the human being, or already detached from him? Also, the linear spiral model of knowledge development is limiting.

Concepts for organizational learning, which is closely related to knowledge management, are given by Argyris and Schön [8, 9] and by Senge [10]. The latter refers to system thinking as very important fifth discipline of the learning organization; also, see [11] for system thinking.

Approaches for knowledge transfer are surveyed in [12]. An approach for knowledge access and development in firms is given by Boisot [13]. Here, development scenarios of knowledge in the Information Space are provided.

Our conception of knowledge is represented by a three-dimensional model of knowledge with types, kinds and qualities. General knowledge conversions between the various knowledge variants are introduced as a model for knowledge dynamics in the enterprise. First, a basic set of such conversions is defined. Building on this set, general knowledge conversions can be defined, which reflect knowledge transfers and development and do not suffer from the restrictions of the SECI-model.

In order to formalize this conception of knowledge and knowledge dynamics, we shortly introduce top level categories of ontologies as described in [14] and apply them to the domain of knowledge and knowledge development. Hence, major categories of knowledge management can be identified. A similar approach is given in [15] for the intellectual capital domain.

Following this path, a knowledge ontology with a corresponding rule system is described in this paper, which implements as well the appropriate top level ontology categories as the described conception of knowledge and knowledge conversions. Everything was developed by using the web ontology language OWL [16]. The reasoning support in combination with a rule system allows for a formal treatment of important knowledge development scenarios.

Application scenarios for knowledge development are classified and described in this paper. They can be represented by general knowledge conversions, which are subject to rule processing. A set of corresponding rules for addressing these scenarios and their representations has been developed and is described in this paper. Therefore, possible solutions for those scenarios can be gained.

The structure of the paper is as follows. After an introduction, Section II introduces our conception of knowledge and knowledge dynamics. Section III overviews the top level categories of things and applies this to the domain of knowledge development. Using this, Section IV describes the knowledge ontology and the corresponding rule system. Afterwards, Section V classifies and describes application scenarios for knowledge development, which partly can supported with the formalizations of the previous section. A summary and outlook section will conclude the paper.

II. A Conception of Knowledge and Knowledge Dynamics

A conception of knowledge and knowledge dynamics in a company is described. More details are given [2].

A. *Knowledge Conception*

We provide a conception of knowledge with types, kinds and qualities. As our base notion, knowledge is understood as justified true belief (in the propositional kind), with a dimension of purpose and intent, identifying patterns in its validity scope, brought to bear in action, see [3] and [9]. It is a perspective of "knowledge-in-use" [17] because of the importance for its utilization in companies and for knowledge management. In contrast, information is understood as data in relation with a semantic dimension, but without the pragmatic and pattern-oriented dimension, which characterizes knowledge.

1) Type Dimension of Knowledge

The type dimension is the most important for knowledge management in a company. It categorizes knowledge according to its presence and availability. Is it only available for the owning human being, or can it be communicated, applied or transferred to the outside, or is it externally available in the company's organizational memory? It is crucial for the purposes of the company, and hence a main goal of knowledge management activities, to make as much as possible knowledge available, i.e., let it be converted from internal to more external types.

Our conception for the type dimension of knowledge follows a distinction between the internal and external knowledge types, seen from the perspective of the human being. As third and intermediary type, explicit knowledge is seen as an interface for human interaction and for the

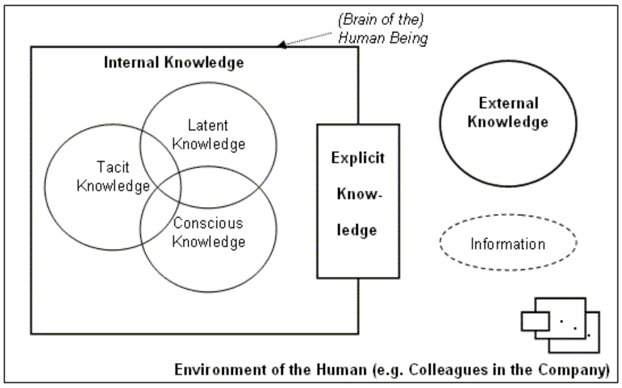


Figure 1. Conception of knowledge types

purpose of knowledge externalization, the latter one ending up in external knowledge. Internal (or implicit) knowledge is bound to the human being. It is all that, what a person has "in its brain" due to experience, history, activities and learning. Explicit knowledge is "made explicit" to the outside world, e.g., through spoken language, but is still bound to the human being. External knowledge finally is detached from the human being and may be kept in appropriate storage media as part of the organizational memory. Figure 1 depicts the different knowledge types.

Internal knowledge can be further divided into tacit, latent and conscious knowledge, where those subtypes do partly overlap with each other; see [18]. Conscious knowledge is conscious and intentional, is cognitively available and may be made explicit easily. Latent knowledge has been typically learning as a by-product and is not available consciously. It may be made explicit, for example in situations, which are similar to the original learning situation, however. Tacit knowledge is built up through experiences and (cultural) socialization situations, is specific in its context and based on intuition and perception.

2) Kind Dimension of Knowledge

In the second dimension of knowledge, four kinds of knowledge are distinguished: propositional, procedural and strategic knowledge, and familiarity. It resembles to a certain degree the type dimension as described in [17]. Propositional knowledge is knowledge about content, facts in a domain, semantic interrelationship and theories. Experience, practical knowledge and the knowledge on "how-to-do" constitute procedural knowledge. Strategic knowledge is meta-cognitive knowledge on optimal strategies for structuring a problem-solving approach. Finally, familiarity is acquaintance with certain situations and environments; it also resembles aspects of situational knowledge, i.e., knowledge about situations, which typically appear in particular domains.

Knowledge kinds go along with knowledge types in the sense, that they occur in most knowledge types. The

TABLE I TYPE/KIND-MATRIX OF KNOWLEDGE

Knowledge Type / Kind	internal: tacit	internal: latent	internal: conscious	explicit	external
propositional	X	X	X	X	X
procedural	X	X	X	(X) [1]	(X) [2]
strategic	X	X	X	(X) [3]	—
familiarity	(X) [4]	X	X	X	? [5]

Legend:
1 can be demonstrated, not to be articulated easily
2 partly through intelligent application systems
3 partly, can be demonstrated
4 if at all, unconscious acquaintance
5 if at all, possibly in future intelligent application systems

type/kind-matrix given in Table 1 indicates, which type/kind-pairs normally appear. One interesting content is, that external knowledge does not appear in the strategic and.familiarity kinds, the latter case with the potential of becoming possible with future intelligent application systems.

3) Quality Dimension of Knowledge

The quality dimension introduces five characteristics of knowledge with an appropriate qualifying and is independent of the kind dimension; see [17]. The level characteristics aims at overview vs. deep knowledge, structure distinguishes isolated from structured knowledge. The automation characteristic of knowledge can be step-by-step-doing by a beginner in a domain of work or automated fast acting by an expert.

Modality as the fourth quality of knowledge asks for the representation of it, be it words versus pictures in situational knowledge kinds, or propositions versus pictures in procedural knowledge kinds. Finally, generality differentiates general versus domain-specific knowledge. Knowledge qualities apply to each knowledge asset.

4) The Knowledge Cube

Bringing all three dimensions of knowledge together, we gain an overall picture of our knowledge conception. It can be represented by the knowledge cube as shown in Figure 2.

Note, that the dimensions in the knowledge cube behave different. In the type and kind dimensions, the categories are mostly distinctive (with the mentioned exception in the sub-types), while in the quality dimension each of the given five characteristics are always present for each knowledge asset.

B. Knowledge Dynamics

Here we give a conception of knowledge conversions. The transitions between the different knowledge types, kind and qualities are responsible to a high degree for knowledge development in an organization. These general knowledge conversions are the building blocks to model knowledge dynamics, i.e., all of acquisition, conversion, transfer, development and usage of knowledge, in an enterprise.

Most important for knowledge management purposes are conversions between the knowledge types, especially those making individual and internal knowledge of employees usable for a company. The explicitation and externalization conversions described in this section achieve this. Implicitly, socializations between tacit knowledge of different people also contribute to this goal.

1) Basic Knowledge Conversions

Five basic knowledge conversions in the type dimension are distinguished here: socialization, explicitation, externalization, internalization and combination. Basic conversion means, that exactly one source knowledge asset is converted into exactly one destination knowledge asset and exactly one knowledge dimension (i.e., the type dimension in this case) is changed. More complex conversions may be easily gained by building on this set as described in the next sub-section. They will consist of n-to-m-conversions and include information assets in addition.

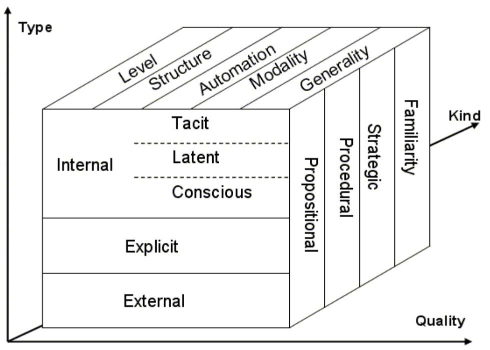


Figure 2. The knowledge cube

Socialization converts tacit knowledge of a person into tacit knowledge of another person. This may succeed by exchange of experience or in a learning-by-doing situation. Explicitation is the internal process of a person, to make internal knowledge of the latent or conscious type explicit, e.g., by articulation and formulation (in the conscious case) or by using metaphors, analogies and models (in the latent case). Externalization converts from explicit knowledge to external knowledge or information and leads to detached knowledge as seen from the perspective of the human being, which can be kept in organizational memory systems. Internalization converts either external or explicit knowledge into internal knowledge of the conscious or latent types. It leads to an integration of experiences and competences in your own mental model. Finally, combination combines existing explicit or external knowledge in new forms.

Basic knowledge conversions in the kind dimension of knowledge do not occur. Those in the quality dimension are mostly knowledge developments aiming at quality improvement. Examples are basic conversions changing the overview, structure and automation quality, respectively.

2) General Knowledge Conversions

Our conception allows the generalization of the basic five knowledge conversions described above. General knowledge conversions are modeled converting several source assets (possibly of different types, kinds and quality) to several destination assets (also possibly different in their knowledge dimensions). In addition, information assets are considered as possible contributing or generated parts of general knowledge conversions.

For example, in a supervised learning-by-doing situation seen as a complex knowledge conversion, a new employee may extend his tacit and conscious knowledge by working on and extending external knowledge in a general conversion, being assisted by the tacit and conscious knowledge of an experienced colleague. As a result of the conversion we have extended internal knowledge of the new employee and extended external knowledge. Figure 3 shows this general knowledge conversion in the proposed BPMN-KEC2 notation, which extends the well-known business process modeling notation BPMN with constructs related to knowledge and knowledge dynamics; see [3] for more details of this notation.

III. ONTOLOGY CATEGORIES

In order to formalize this conception of knowledge and knowledge dynamics, we shortly introduce top level categories of ontologies as described in [14] and apply them to the domain of knowledge and knowledge development. Hence, major categories of knowledge management can be identified. A similar approach is given in [15] for the intellectual capital domain.

" Ontology . . . is the study of existence, of all the kinds of existence – abstract and concrete that make up the world" (Sowa in [14, p.51]). According to Peirce ([19], also in [14]), one fundamental distinction of categories in ontologies is reflected by the idea of so-called triads, which he called Firstness, Secondness, and Thirdness. Here we follow Sowa [14] and call them Independent, Relative, and Mediating, respectively. Independents represent actual entities, which are identified by qualities and can exist independent from any other entities. In logic, they can be represented by a monadic predicate (e.g., person(z)). Relatives denote entities, which exist relative to other entities. In logic, their representation would be in the form of a dyadic predicate (e.g., mother(y,z), where y would be the mother of z here). Mediating (or Thirdness) brings the first and second categories in relation. In our example, a ternary predicate in logic could be motherhood(x, y, z), where the motherhood x brings together the mother y and

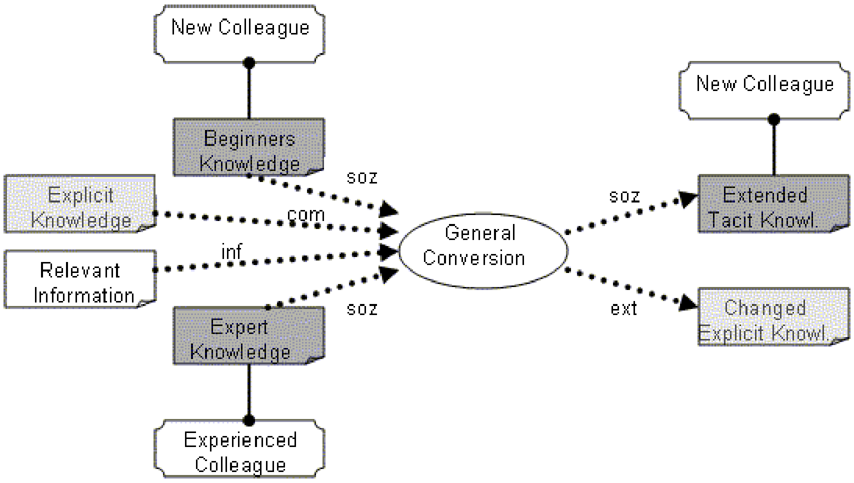


Figure 3. Examples of a general knowledge conversion

	Physical		Abstract	
	Continuant	**Occurent**	**Continuant**	**Occurent**
Independent	Object	Process	Knowledge	Single-Loop Learning
Relative	Relation	Participation	Rules	Double-Loop Learning
Mediating	Organizational Context	Community of Practice	Knowledge Strategy	Deutero Learning

Figure 4. Categories of a knowledge management ontology (Sources: [19], [15], author's adaptation for knowledge management)

her child person z. It is not possible, to express this mediating entity with a monadic or dyadic predicate or a conjunction of dyadic predicates. In general, mediating entities can be represented in logic by ternary or higher predicates.

The second distinction of ontology categories addresses the physical vs. abstract contrast of entities. This distinctionis independent of an observer's viewpoint. This is different for the third and final distinction of categories, where the contrast between continuant and occurent is stated. Continuants are entities, which are recognizable as one and the same entity at different times. Occurents flow in the sense, that they can only be identified by their location in the space-time region.

Having a triadic and two dual distinctions, we end up in a set of twelve categories of entities. In Figure 4, we list these categories in a matrix representation according to [14]. We changed the categories to reflect important concepts for the domain of knowledge development and management. Those changes are partly already proposed in [15] for the domain of intellectual capital, which is related to knowledge management.

On the physical side, we only want to emphasize the occurent and mediating entry in Figure 4. Here communities of practice in companies can be seen, which relate processes and participation in knowledge themes of interest. On the abstract side, continuants in the knowledge development and management domain are knowledge, rules, and knowledge strategy, taken from independent over relative to mediating. Abstract occurents are organizational learning entities, namely single-looplearning, double-loop learning, and deutero learning, see [8,9] for this organizational learning loops.

IV. THE KNOWLEDGE ONTOLOGY

In this section, we present the Knowledge Ontology, which implements the conception of knowledge and knowledge dynamics, as described in Section II. As one main goal the ontology will enable the discovery of the crucial knowledge conversions for a company. The ontology (as visually shown in Figure 5) is divided in four core concepts: *Knowledge*, *Information*, *Knowledge_Conversion* and *Knowledge_Dimension*. The three different knowledge dimensions are represented as: *Type_Dimension*, *Kind_Dimension* and *Quality-Dimension*. *Knowledge* is defined according to these dimensions. Properties are used to model the relationships between *Knowledge* and *Dimensions*: *hasType*, *hasKind* and *hasQuality*. For example, *Explicit_Knowledge* is defined as every piece of knowledge, which is related to the instance *Explicit_Type* via the *hasType* property. In the same way, *Knowledge* in general must be related to every quality sub-dimension through the *hasQuality* property.

In the case of the type dimension of knowledge, we have defined disjoint axioms in order to make explicit the fact that a piece of knowledge cannot be simultaneously external and internal - except in the case of *Latent*, *Conscious* and *Tacit Knowledge*, which can actually overlap (compare with Figure 1). There are also disjoint axioms for the kind dimension, since a propositional piece of knowledge cannot be *Procedural*, neither *Strategic* nor *Familiarity*.

Two properties have been defined to model the knowledge conversions: *hasSource* and *hasDestination*, with knowledge conversions as ranges, and pieces of knowledge and information as domains.

A General Conversion is modeled through the *Knowledge Conversion* concept, and its only restriction the fact that it must have at least one source asset and one destination asset. *Basic Conversions* are more specific, in the sense that they have only one source and only one destination. Eight basic conversions (five in the type dimension, three in the quality dimension) are defined in the ontology.

The concept *Crucial_Conversion* gathers those conversions that contribute to the goal of making the knowledge available for the company.

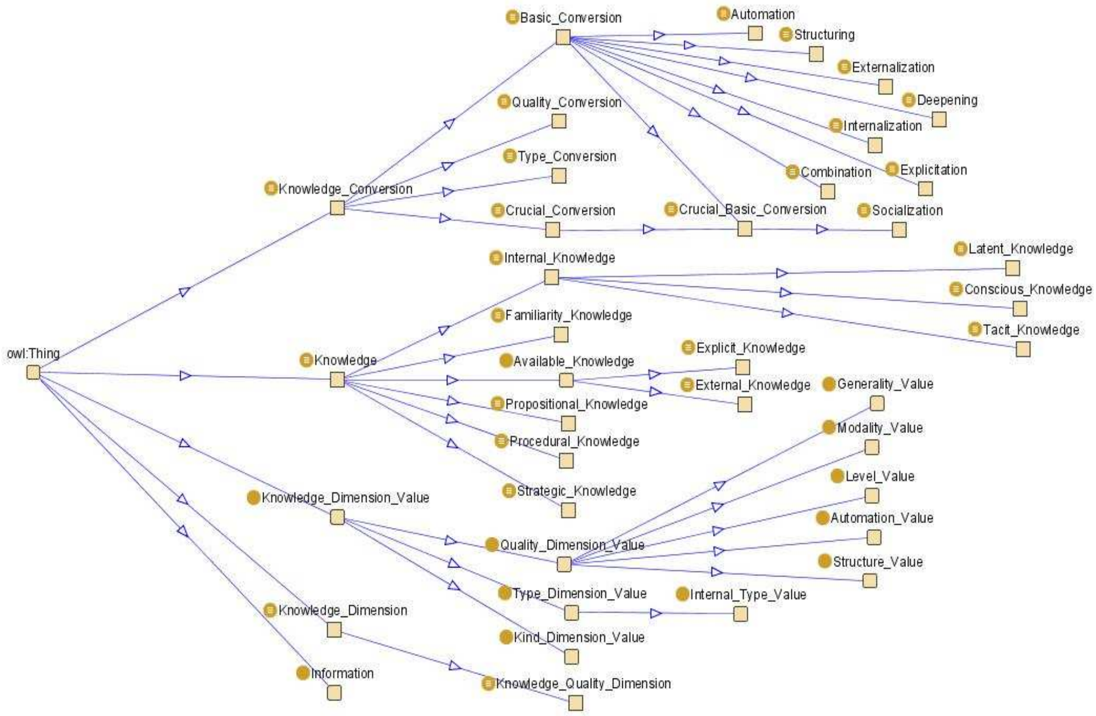


Figure 5. Knowledge ontology hierarchy

A. Restrictions and Reasoning

Basic reasoning is based on subsumption mechanisms that deal with the ontology hierarchy. However, ontologies can contain more complex elements to enable advanced reasoning. In this way, the Knowledge Ontology has been extended with OWL restrictions to enable new ways of generating interesting new knowledge.

Here, we will only describe some of the most interesting restrictions. Let us imagine that we have two pieces of knowledge in our company: *knowledge1* and *knowledge2*. Both pieces of knowledge have as type *Explicit* (is related to the instance of *Type_Dimension_Value* called *Explicit* through the property *hasTypeValue*). Additionally we have defined *Explicit_Knowledge* as follows:

Available_Knowledge AND
∃hasTypeValue has Explicit

Thus, a reasoner will identify both pieces of knowledge as *Explicit_Knowledge* (and using subsumption also as *Available_Knowledge*).

We can consider two different conversions *conversion1* and *conversion2*: one that converts *knowledge1* in *knowledge2* and vice versa. Then, we have defined a *Crucial_Conversion* as:

Knowledge_Conversion AND
∃hasDestination some Available_Knowledge

Thus, we can infer that *conversion1* is a *Crucial_Conversion* for the company.

B. Rules

Ontology restrictions allow us to infer new characteristics of a given concept or instance. However, in some cases we could require to generate new instances in the ontology depending on certain situations. In this case we have used rules, so the knowledge ontology will be able to infer all the possible conversions given some pieces of knowledge. First, the rule engine will create basic conversions with all the possible source-destination pairs, and then, the same engine will characterize these conversions, inferring the changing dimension for each case.

SWRL [20] rules have been defined and the Jess rule engine [21] has been used. The main rule for our model is the one that creates new conversions for the knowledge assets that we have stored in our ontology:

Knowledge(?k1) ^ Knowledge(?k2) ^
hasDimensionValue(?k1, ?v1) ^
hasDimensionValue(?k2, ?v2) ^
differentFrom(?k1, ?k2) ^ differentFrom(?v1, ?v2) ^
swrlx:makeOWLThing(?c, ?k1, ?k2)
→
Knowledge_Conversion(?c) ^ hasSource(?c, ?k1) ^
hasDestination(?c, ?k2)

Thus, this rule is activated when we have two different pieces of knowledge with different dimensions values. In this case, a new instance is created for providing a new knowledge conversion between both pieces of knowledge.

Then, we have six rules to infer the changing dimensions of each of the new discovered conversions: one for the type dimension and five for the quality ones. For example, the rule for the type dimension is as follows:

Knowledge(?k1) ^ Knowledge(?k2) ^
hasTypeValue(?k1, ?v1) ^ hasTypeValue(?k2, ?v2) ^
differentFrom(?v1, ?v2) ^ Knowledge_Conversion(?c1) ^
hasSource(?c, ?k1) ^ hasDestination(?c, ?k2)
→
hasChangingDimension(?c,
Knowledge_Type_Dimension)

Suppose that we have two pieces of knowledge in our company (*knowledge1* and *knowledge2),* which are related through the *hasTypeValue* property to *Explicit* and *External*, respectively. Both are related to the values *Familiar* and *Step by step*. Using the defined rules, new instances are produced. Thus, the rule engine has inferred two conversions, one for "knowledge1 → knowledge2", and another for "knowledge2 → knowledge1". Then, the reasoner can infer additional facts:

- About the pieces of knowledge:
  - They are both *Familiar_Knowledge*.
  - One of them is *External_Knowledge,* the other is *Explicit_Knowledge*.
  - Both are *Available_Knowledge*.
- About the conversions:
  - They are both *Basic_Conversion*.
  - Both are *Crucial_Conversion* (since they have *Available_Knowledge* as destination).
  - Both are *Type Conversions* (since they change the type dimension).

V. APPLICATION SCENARIOS

Application scenarios for knowledge development in a company can be related with our model of knowledge dynamics. Two categories of scenarios exist. The first one is constructive and builds knowledge development chains (see [2] for a modeling approach). Here we focus on the second scenario category, which consists of analytic scenarios. They can be represented by general knowledge conversions and are subject to rule processing as described in Section IV. In these scenarios we face gaps in knowledge dynamics chains as provided by knowledge conversions. These gaps will be closed by applying appropriate rules to the relevant instances of knowledge assets and conversions, which have been instantiated in our knowledge ontology.

As an example suppose, that the knowledge requirement for a project and the learning options in the company are known. The task would be to identify the minimal knowledge requirements for a new employee, who should work on the project and should be able to fulfill the requirements at least after some learning efforts. This scenario is modeled in Figure 6, where the first part shows the analytic scenario, and the second part is giving the solution after rule application. As notation BPMN-KEC2 is used again.

Part I: Basic analytic scenario

Part II: Solution after rule application

Figure 6. A basic scenario and its solution

Our principle approach to deal with analytic scenarios is shown in Figure 7. The bold arrow in the first line indicates the knowledge development activity, which is needed in order to resolve an application scenario with unknown part. Our approach first represents the application scenario as a general knowledge conversion, applies an appropriate rule of our rule system to it, and finally interprets the completed knowledge conversion as solved application scenario.

For example, the knowledge requirements for a project are known as well as the learning options in the company. From that, one would try to identify minimal knowledge requirements for a new employee, who should work in the project and should be able to fulfill the requirements at least after some learning efforts.

Our representation of this scenario is that we know the result of a knowledge conversion as well as the conversion itself, but we do not know the source knowledge asset. A rule application should deliver the missing knowledge asset.

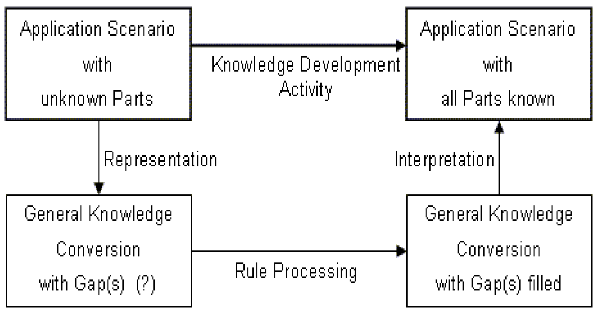


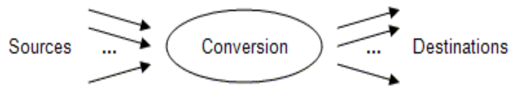
Figure 7. Support of knowledge development scenarios

A. Analytic Application Scenarios and their Representation

Analytic application scenarios for knowledge development are characterized by gaps in the corresponding knowledge dynamics chains. Without restriction of generality, we focus on simple scenarios, which can be represented by a single general knowledge conversion. More complex scenarios should be composed of simple ones.

A representation as a general knowledge conversion leads to a set of eight possible scenarios. In the conversion definition with sources, conversion and destinations we can apply zero or more question marks, i.e., gaps of unknown parts, to the conversion. Out of the eight possible scenarios, we do not further consider two of them. The case with no gap is a constructive scenario really, while the case without any known part is not a realistic one. The other six scenarios are outlined in the following and shown in Figure 8.

Scenarios with known destination parts of the conversion and with gaps on the sources side represent situations, where the target of knowledge development activities is known. A known conversion part in the knowledge conversion in this scenario would indicate existing knowledge development options in the company, while a gap indicates missing development support (Scenarios 1 and 2). Scenario 5 describes known sources and destination parts, but missing development options and support in the company. Scenarios 3 and 4 have a complete sources part of the knowledge conversion and gaps in the destinations part. If existing knowledge development options are available, then the scenario would ask for the potential of evolving knowledge applying these options (Scenario 3). If no such options exist, the question of the scenario would be, which knowledge development activities should be initiated and to which possible result in extended and new knowledge this could lead (Scenario 4). Finally, Scenario 6 assumes existing knowledge development options in the company, but incomplete sources and destinations parts. If only very few out of the sets of sources or destinations are unknown, this scenario can be partly handled with our approach also. Otherwise, especially in the case of completely exclusively unknown sources and destinations, no further treatment is possible.



Scenario	Sources	Conversion	Destinations
1	?	known	known
2	?	?	known
3	known	known	?
4	known	?	?
5	known	?	known
6	?	known	?

Legend:
? for Sources (Destinations): One or more ot the Sources (Destinations) are unknown
? for Conversion: Conversion is unknown

Figure 8. Application scenarios and representations

B. Rules Application to Representations of Scenarios

As described in Section IV, a rule system has been developed, which is applied to instances of knowledge and conversions introduced in the knowledge ontology.

Only rules for basic knowledge conversions in the type dimension with only one gap exist until now. We therefore are restricted currently to the corresponding 1-to-1 cases of scenarios 1, 3 and 5 as described before in Figure 8. A rule for Scenario 5 case has already been given in Section IV. For each of Scenarios 1 and 3, there exist five such 1-to-1 cases, because the known conversion part must be one of the five basic knowledge conversions in the type dimension. Here we analyze the cases and provide appropriate rules. Note that in some scenarios there is unambiguous implication result of the corresponding rules.

1) Scenario: Source → Socialization → ?
The following rule produces a new destination *Tacit_Knowledge:*

```
Knowledge(?k1) ^ Socialization(?s) ^
hasSource(?s, ?k1) ^
swrlx:makeOWLThing(?k2, ?k1)
→
Tacit_Knowledge(?k2) ^ hasDestination(?s, ?k2)
```

2) ? → Socialization → Destination
A new source Tacit_Knowledge is produced:

```
Knowledge(?k2) ^
Socialization(?s) ^
hasDestination(?s, ?k2) ^
swrlx:makeOWLThing(?k1, ?k2)
→
Tacit_Knowledge(?k1) ^
hasSource(?s, ?k1)
```

3) Source → Explicitation → ?
A new destination Explicit_Knowledge is produced:

```
Knowledge(?k1) ^
Explicitation(?e) ^
hasSource(?e, ?k1) ^
swrlx:makeOWLThing(?k2, ?k1)
→
Explicit_Knowledge(?k2) ^
hasDestination(?e, ?k2)
```

4) ? → Explicitation → Destination
In this case, we produce latent or conscious knowledge as the source. However, it is not possible to decide on one of them so this rule has been generalized to produce a new destination internal knowledge.

5) Source → Externalization → ?
This kind of conversion produces information or external knowledge. So, this case will depend on the user perspective and decision, so it cannot be solved at rule level.

6) ? → Externalization → Destination
A new source explicit knowledge is produced:
Knowledge(?k2) ^
Externalization(?e) ^
hasDestination(?e, ?k2) ^
swrlx:makeOWLThing(?k1, ?k2)
→
Explicit_Knowledge(?k1) ^
hasSource(?e, ?k1)

7) Source → Internalization → ?
We produce a new source internal knowledge:
Knowledge(?k1) ^
Internalisation(?i) ^
hasSource(?i, ?k1) ^
swrlx:makeOWLThing(?k2, ?k1)
→
Internal_Knowledge(?k2) ^
hasDestination(?i, ?k2)

8) ? → Internalization → Destination
This kind of conversion can work over information or external knowledge. So, this case will depend on the user perspective and decision, so it cannot be solved at rule level.

9) Source → Combination → ?
This kind of conversion can produce Information or External Knowledge. So, this case will depend on the user perspective and decision, so it cannot be solved at rule level.

10) Scenario: ? → Combination → Destination
The following rule produces a new source *Available_Knowledge*, it cannot decide on a specific type of *Explicit_Knowledge* or *External_Knowledge*:
Knowledge(?k2) ^ Combination(?c) ^
hasDestination(?c, ?k2) ^
swrlx:makeOWLThing(?k1, ?k2)
→
Available_Knowledge(?k1) ^
hasSource(?c, ?k1)

C. *Setting the Approach into Perspective*

In this section, application scenarios have been discussed so far, with focus on analytic scenarios. In this sub-section, the findings are set into perspective.

As can be seen in Figure 7, there is a distinction between the application scenario level and the formal representational level with general knowledge conversions.

In a real knowledge development situation in a company, as first activity the corresponding application scenario must be identified and described. The following steps of representation as general knowledge conversions with gaps, rule processing, and the interpretation of the results as application scenario with all parts known are shown in Figure 7. As a final activity these outcomes have to be implemented in the company.

According to this discussion, we can in principle identify a hierarchy consisting of knowledge development instances, descriptions of them, and a model for formally handling them:

- Knowledge development situation/need in the company.
- Identification and description of it as application scenario with unknown parts.
- Modeling of this scenario with general knowledge conversions with gaps, which allows for rule processing in order to fill the gaps.

What is needed in the future is, firstly, to build up experience in concrete knowledge development situations in companies and to apply the steps to them as described above. This would verify our approach and indicate its value for knowledge development. Secondly, the body of rules on the modeling level must be extended beyond basic knowledge conversions and augmented with heuristics, where rules application could not possibly show up a unique result. Heuristics should provide good or acceptable solutions, where exact solutions are not possible.

VI. SUMMARY AND OUTLOOK

A conception of knowledge development in an enterprise has been given. It is based on a concept of knowledge and knowledge dynamics.

In order to formalize this conception of knowledge and knowledge dynamics, top level categories of ontologies have been given and applied to the domain of knowledge and knowledge development. Hence, major categories of knowledge management could be identified.

In order to implement this conception, a knowledge ontology has been built and described in this paper, together with reasoning support and in combination with a rule engine. This has opened the path, to solve open questions in application scenarios for knowledge development.

With the help of representations, these scenarios can be mapped to general knowledge conversions, which are subject to rule processing in relation to the knowledge ontology. A final interpretation steps leads back to the solved scenario. In effect, the knowledge development activity has been undertaken with this procedure. A clear distinction between the application scenario level and the formalized model level is shown by this approach, with representation and interpretation as bridging “operators”.

Until now only simple application scenarios and their representations are covered by the set of developed rules. In more complex scenarios, possible solutions are no longer unique. With the help of heuristics, which have to be developed, good or acceptable solutions may be identified. Those heuristics should be developed having results of several fields of study in mind, e.g., learning psychology and organizational learning.

What is needed in the future is, to build up experience in concrete knowledge development situations in companies and to apply the steps to them as described above. This would verify our approach and indicate its value for knowledge development.

ACKNOWLEDGEMENT

Supported in parts by TIN2008-04844 (Spanish Ministry of Education and Science).

REFERENCES

[1] Ammann, E., Ruiz-Montiel, M., Navas-Delgado, I., and Aldana-Montes, J.F.: “A Knowledge Development Conception and its Implementation: Knowledge Ontology, Rule System and Application Scenarios”, Proc. of the 2nd Int. Conf. on Advanced Cognitive Technologies and Applications (COGNITIVE 2010), Lisbon, Portugal, 21-25 November, 2010, pp. 60-65.

[2] Ammann, E.: “A Meta-Model for Knowledge Management”, Proc. of the 5th Int. Conf. on Intellectual Capital and Knowledge Management (ICICKM), New York 2008, pp 37-44.

[3] Ammann, E.: “The Knowledge Cube and Knowledge Conversions”, Proc. of the World Congress of Engineering, International Conference on Data Mining and Knowledge Engineering (ICDMKE), London, UK 2009, pp.319-324.

[4] Lehner, F.: Wissensmanagement (in German), 3rd ed., Hanser, München 2010.

[5] Bubenko, J.A., Jr., Brash, D., and Stirna, J.: EKD User Guide, Dept. of Computer and System Science, KTH and Stockholm University, Elektrum 212, S-16440, Sweden.

[6] EKD – Enterprise Knowledge Development, ekd.dsv.su.se/home.html. Last access: January 5, 2012.

[7] Nonaka, I. and Takeuchi, H.: The Knowledge-Creating Company – How Japanese Companies Foster Creativity and Innovation for Competitive Advantage, Oxford University Press, London 1995.

[8] Argyris, C. and Schön, D.A.: Organizational Learning: a Theory of Action Perspective, Addison-Wesley, Reading, Massachusetts 1978.

[9] Argyris, C. and Schön, D.A.: Organizational Learning II – Theory, Method, and Practice, Addison-Wesley, Reading, Massachusetts 1996.

[10] Senge, P.: The Fifth Discipline, Currency Doubleday, New York 1994.

[11] Sterman, J.D.: Business Dynamics – Systems Thinking and Modeling for a Complex World, McGraw Hill, Boston, USA 2000.

[12] Ling, L.H.: “From Shannon-Weaver to Boisot: A Review on the Research of Knowledge Transfer”, in: Proc. of Wireless Communications, Networking and Mobile Computing (WiCom 2007), 2007, pp. 5439-5442.

[13] Boisot, M.H.: Knowledge Assets, Oxford University Press, 1999.

[14] Sowa, J.F.: Knowledge Representation – Logical, Philosophical, and Computational Foundations, Brooks/Cole Thomson Learning, Pacific Grove, 2000.

[15] Vlismas, O. and Venieris, G.: “Towards an Ontology for the Intellectual Capital Domain”, in: Journal of Intellectual Capital, Vol.12, No. 1, 2011, pp. 75-110.

[16] OWL Web Ontology Language Reference, http://www.w3.org/standards/history/owl-ref. Last access: January 5, 2012.

[17] De Jong, T. and Fergusson-Hessler, M.G.M.: “Types and Qualities of Knowledge”, Educational Psychologist, 31(2), 1996, pp. 105-113.

[18] Hasler Rumois, U. : Studienbuch Wissensmanagement (in German), UTB orell fuessli, Zürich 2007.

[19] Peirce, Ch.S.: Writing of Charles S. Peirce, vols.1-5, Indiana University Press, Bloomington, 1982-1993.

[20] SWRL: A Semantic Web Rule Language Combining OWL and RuleML, http://www.w3.org/Submission/SWRL/ . Last access: January 3, 2012.

[21] Jess Rule Engine, http://www.jessrules.com. Last access: January 4, 2012.