Epistemic Logic and Knowledge Management

Prof. Dr. Marcus Spies
Knowledge Management
Ludwig-Maximilians-Universität München
marcus.spies@computer.org

1 Introduction: Use or truth? Or both?

In most textbooks on knowledge management (KM; e.g., [5]), there is no reference to epistemic concepts at all. In [8], there is even an explicit statement that epistemologic concepts of knowledge are "useless" for KM, since KM seems to be about the use rather than the truth aspect of knowledge. However, there is a remarkably explicit reference to such concepts in [7] establishing the use of epistemic analysis in understanding and supporting the process of knowledge creation in today's innovative companies.

And, in fact, to what use could knowledge be to companies if it were not for the truths implicit in it? As an example, take knowledge of members of a software development team involved in requirements analysis. The value of knowledge is seen by the team's ability to translate requirement statements into a reasonably abstract algorithmic form in order to estimate programming, testing, and code optimizing efforts. If there were no reference to truth in this kind of teamwork, it certainly would be a useless exercise.

But, even in [7], the reference to epistemic concepts is not formulated in terms of epistemic logic. Thus, there is an opening of an important dialogue between KM and philosophy, but this dialogue seems to be still in its initial phase.

2 Some fundamental concepts of Epistemic Logic

In this paper, I will confine myself to the essential components of epistemic logic as they have been outlined in several writings of Jaakko Hintikka, [4] in particular. Syntactically, we need at least a first-order theory and a knowledge operator with two uses: First, it can qualify a formula as "known" by some individual, which corresponds to expressing factual knowledge (or “knowing that“). Second, it can qualify an existential quantifier, which means that values of its bound variable that satisfy the formula are known. This corresponds to “knowing which“. (For more philosophical background on these types of knowledge, see [3]). Additionally, it seems reasonable to allow nesting of these operators, which expresses so-called “transactive knowledge“ [6], i.e. knowledge about other individuals' knowledge.

Semantically, epistemic logic can be based on Hintikkas Game-Theoretic-Semantic (GTS; again, see [4]). The essential idea of GTS is to define a game that, given a first-order sentence and any interpretation in the standard sense of predicate logic, allows to check whether the interpretation is a model for the sentence in question. Assuming the sentence to be in prenex normal form (i.e., all quantifiers precede the body of the formula), the game is performed according to the sequence of quantifiers by two different players: the falsifier selects domain elements for universally quantified variables, the verifier selects domain elements for existentially quantified variables (depending on the falsifier's previous selections). The goal of the play is to find domain elements by successive selections such that the sentence is true in the traditional sense. If a disjunction occurs in the body, the verifier selects on which disjunct the play continues, if a conjunction occurs, the falsifier selects on which conjunct to continue. If negation occurs, the roles of verifier and falsifier are simply exchanged and the play continues with the remaining formula body without the negation operator. The play stops once a false atomic statements is reached (win of the falsifier) or once the last true atomic statement is checked (win of the verifier). Truth (falsity) of a sentence is then defined as the existence of a winning strategy for the verifier (falsifier). The great advantage of the truth definition in GTS is that it can easily cope with undecidable sentences, since for such a sentence there is simply no winning strategy for either the verifier or the falsifier, which is a common situation in game theory. (Hintikka [4] shows the impact of this argument w.r.t. Gödels and Tarskis incompleteness results.)

GTS is not primarily intended to be used in the context of epistemic logic, however, it provides a useful approach. With GTS, besides proof games, interrogation games can be described that are played with the goal of seeking knowledge.

3 Making epistemic sense of knowledge sharing

The implications of GTS for KM seem highly interesting: If truth can be defined as an interplay of testing agents and searching/selecting agents, we have a nice philosophical basis for one of the cornerstones of KM, namely, that knowledge should be both shared and distributed among different team members or teams, or even different parts of companies. And it's more than just nice: KM consultants and practitioners often have to solve the problem of just *how to design processes and infrastructures for sharing or distributing* knowledge. It is usually a bad approach to make knowledge available in a codified version (like in a database with some access management and searching amenities) without defining the interests of buyers and sellers in such a knowledge marketplace. According to epistemic logic, this is no surprise. In order to "justify a concept" (as [7] put it), i.e. to test whether some suggestion is true for the company, what is needed is a coordination of knowledge of different types. Factual knowledge will lead to formulating goals or subgoals in this process (corresponding to the abstract role of the falsifier), "knowing which" will help to fill in the information needed to reach or abandon such goals (corresponding to the abstract role of the verifier), and, finally, transactive knowledge tells which real agent might play the role needed next. That is, there must be structure to all knowledge sharing or distributing, a structure which is related to the content, the meaning of what is or should be known. Therefore, a pure management approach to KM that disregards content cannot be successful – a fact that has been observed plenty of times and that drops out as an immediate conclusion if one adopts the essentials of epistemic logic.

Practically, it means, that a KM-supporting system has to be defined along with use cases that specify the typical interactions of people with the system and between people in the different roles. Doing so explicitly can be supported by the formalisms of epistemic logic. A research project using the modeling standard UML is being prepared at my chair together with Fujitsu Enabling Software Technology and Fraunhofer research. The aim is to have something that could be called K-UML, that is a set of innovative tools for model-based knowledge management.

A further practical consequence is that information brokerage and transactive memory are just one step towards a solution to KM sharing problems. This is because sharing is no reasonable goal in itself, it becomes useful only if new games (in the sense defined above) can be played and, hopefully, won. In KM terms, sharing is useless if it does not open new product or service offering potentials. However, in order to achieve this, goals and methods for such games must be set. A mere knowledge sharing agent, human or computer-based, cannot do this. Therefore, if a company wants to set up knowledge based services, it should not build a mere "brokerage" team, but a team that includes or has permanent access to real experts, whether in the factual or in the knowing-which sense.

4 Knowledge changes and company knowledge maps

There is a second area of KM for which epistemic analysis even on a very rough level allows some helpful conclusions. It is the area of organizational knowledge maps, nowadays often termed "corporate taxonomies" or even ontologies (where this last concept has a somewhat different meaning within the philosophical community), see [1]. These maps are built in order to have conceptually founded systems of document descriptors and, thus, to enable intelligent web services on the basis of semantically organized documents. One major issue with corporate taxonomies is that, unlike their biological counterpart, these taxonomies need to be constantly evolving. A recent simulation study [2] has shown that there is a dramatic decrease in search efficiency if only a few new products or services have been added to the company's portfolio without sufficient adaptation of the taxonomy underlying the search engine's indexing procedure.

From the point of view of traditional first-order logic, dynamic languages are not quite in the focus of the analysis. But, from the point of view of epistemic logic, "coming to know" something should result in a change of the underlying language. An example discussed in [4] is the empirical investigation of the effect of an independent variable x on a dependent variable y. Logically, this amounts to interpreting the formula $(\forall x)(\exists y)S(x, y)$. After suitably many observations, we come to know some more or less precise functional relationship between x and y. This can be expressed by one or more functions g, such that we know that $(\forall x)S(x, g(x))$. Such functions are well-known especially from the AI literature and are called Skolem-functions. Skolem-functions are added to a language in order to eliminate existential quantifiers in the scope of universal quantifiers from sentences of a logical problem. Hintikka's argument is that these functions imply the existence of a winning strategy for the verifier of the formula in question.

It is easy to transfer this example to KM. Take a team involved in maintenance of a complex production plant. Hopefully, with some experience, they come to know that for each plant failure there is an appropriate set of indicators which helps planning the actual maintenance work. So, team experience takes the form of a Skolem function that maps plant states into suitable indicator values, a classical example of "knowing which". Now, assume that the parts and processes of the production plant are modelled in an ontology of some useful format. The task of good KM is, then, to make sure the team's experience is entered into this ontology. Actually, this task boils down to adding suitable Skolem functions to relevant statements in the rule-base accompanying the ontology.

Of course, there are less simple examples and epistemic logic might not be equally helpful in all cases. But I hope to have provided some convincing support for the hypothesis that there are some fruitful implications for KM if epistemic logic is taken into account. Further research will most probably add more evidence in favor of this hypothesis.

References

[1] Dieter Fensel. *Ontologies: A Silver Bullet for Knowledge Management and Electronic Commerce*. Springer, Berlin, 2001.

[2] Martin Hepp and Klaus Böhnlein. Modellierung der pflege von standards mit hilfe höherer petri-netze. In Jörg Biethahn, editor, *Proc. 8. Symposium: Simulation als betriebliche Entscheidungshilfe*, pages 3–13. Goettingen, 2002.

[3] Jaakko Hintikka. *Knowledge and the Known.* Synthese Historical Library. D. Reidel Publishing Company, Dordrecht, 1974.

[4] Jaakko Hintikka. *The principles of Mathematics revisited.* Cambridge University Press, Cambridge, 1996.

[5] Gilbert Probst, Steffen Raub, and Kai Romhardt. *Wissen managen. Wie Unternehmen ihre wertvollste Ressource nutzen*. Gabler, Wiesbaden, 3rd edition, 1999. KnowMgmt.

[6] Marcus Spies, Elisabeth Brauner, Rudolf Kerschreiter, and Andreas Mojzisch. Sozialpsychologische erfolgsfaktoren des wissensmanagement. In Elke van der Meer and et al., editors, *43. Kongreß der Dt. Ges. f. Psychologie*, page 347, Berlin, 2002. Pabst Science Publishers.

[7] Georg von Krogh, Kazuo Ichijo, and Ikujiro Nonaka. *Enabling Knowledge Creation*. Oxford University Press, Oxford, 2000.

[8] Helmut Willke. *Systemisches Wissensmanagement*. Lucius & Lucius, Stuttgart, 1998.