Analogical thinking as a cognitive strategy to develop models in information systems

Susanne Schwarzer, Matthias Schrepfer
Växjö University, Sweden

Susi.Schwarzer@web.de
Matthias.Schrepfer@gmx.de

Shortened English version of

Analogisches Denken als Erkenntnisstrategie für Modellbildung in der Wirtschaftsinformatik

by Alfred Holl, Robert Auerochs

(Frank, Ulrich (ed.): *Wissenschaftstheorie in Ökonomie und Wirtschaftsinformatik. Theoriebildung und –bewertung, Ontologien, Wissenmanagement.* Wiesbaden: DUV 2004, 367-389, ISBN 3-8244-0738-8)

Abstract: Information systems as an empirical science require models to represent and to design structures of companies. How do humans gain knowledge to develop these models? This paper discusses analogical thinking as a possible cognitive strategy for this purpose. This requires to epistemologically define the terms of analogy and analogical thinking and to explain their relation to inductive and deductive thinking. Our epistemological basis is Karl Popper's ontological theory. A formal method to gain scientific knowledge will be introduced as well as a method to transfer knowledge: analogy induction and analogy deduction are established as new terms. While we construct new models and test them, we unconsciously use cognitive strategies. With the conscious use of these strategies in a scientific cyclic process, we can support the formal correctness, relevance, transparence and comparability of such models.

1 Intention, general concept of “model” and overview

One of the aims of information systems is to support and optimize information processing in companies with the help of information technology. Therefore, it is necessary to analyze structures in companies (e.g. behavior (business processes) and data structures), to document all the information gained and finally to optimize these structures. The development of IT solutions should be oriented towards these structures. An *information model* ([Holl99] 174) is based on the interpretation of the relations and structures in companies ([Schü95] 435). The process in which the model developer acquires knowledge is described in literature as an intuitional, speculative, creative act or as inspiration and, consequently, cannot be assumed to produce objective knowledge. Previous characterizations of knowledge acquisition in information systems are mainly confined to the *syntactical formulation of models* (e.g. EPC, ERM, UML etc). The process of *knowledge acquisition* was often not regarded critically and, if so, often not discussed properly ([HoKr02] 65). One possible *strategy to gain knowledge* is analogical thinking. This strategy can be formalized in a mathematical way. The aim of this paper is to characterize type constructions based on analogical thinking. We will show new aspects regarding the methodology of modeling. For this purpose, we introduce Karl Popper’s ontological theory of the three worlds in Chapter 2 and epistemological definitions of analogy, analogy induction and analogy deduction in Chapter 3. In Chapter 4 we deal with the exactness of models designed using the strategies mentioned above as well as with limitations and consequences of analogical thinking.

2 Epistemological background

In order to adequately discuss the process of knowledge acquisition, especially regarding model design, a *model of model building processes* (epistemological meta-model) is required. A plausible meta-model is Karl Popper’s theory of the three worlds as it is easily adaptable to the particularities of information systems. Nevertheless, we have to be aware of the fact that there is a lot of critique against Popper’s theory. With the transfer of his ontological levels to information systems, the cognitive processes can be described much better than before (2.1). Furthermore, essential terms can be defined (2.2).

2.1 Popper's model of the three worlds

In addition to the characterization and description of Popper's ontological levels (= levels of existence, 2.1.1), the transitions and interrelations between the different levels of existence are important (2.1.2).

2.1.1 Levels of existence

Popper postulates the three ontological levels: "nature" (world 1), "consciousness" (world 2) and "culture" (world 3).

World 1: Reality, universe (especially companies)

The definition of Popper's world 1 covers nature, all material, physical objects (e.g. the physical aspect of staff members, documents etc.) as well as immaterial artifacts created by human beings (e.g. states etc.). This world corresponds to the "absolute" reality. Within information systems, these objects have to be examined in order to adequately design a model.

World 2: Individual experiences (especially consciousness of model designers)

Every model designer gains his own understanding of physical objects during knowledge acquisition. This leads to a *reconstruction of reality* (in the form of *phenomena*) influenced by the model designer's individual disposition. There are as many worlds 2 as humans. Knowledge gained from world 3 (e.g. theories) is individually learned and used by every human. Learning world 3 knowledge is called *activation*.

World 3: Culture, world of models (especially enterprise models)

It contains trans-subjective knowledge gained from *externalized human thoughts* (e.g. concepts, ideas, language, theories etc.). This world exists as well as world 1 only once. Its contents are temporally dynamic and subject to an evolutionary process.

Direct / indirect objects of cognition

Objects in world 1 are *indirect* objects of cognition. They can be perceived by humans, for example by touching or seeing them in reality, but knowledge about them is not directly available for humans.

Direct objects of cognition are objects in world 2 and 3. They only exist in human consciousness or in developed models. Knowledge about these objects of cognition is immediately available for humans.

2.1.2 Interrelations between Popper's three worlds

For a complete discussion of the three worlds, an analysis of all of the interrelations between them would be necessary. Within this paper, only interrelations regarding world 2 are explained.

World 1 → World 2: A knowledge gaining process runs between a subject which perceives and an object which shall be perceived. This process can be influenced by the object of cognition as well as by the subject (individual influence of the model designer) ([Voll98] 41). This becomes obvious at the transition from world 1 to world 2 in the form of *primary* (phylogenetically determined) and *secondary* (ontogenetically determined) *distortions*. Primary distortions cannot be made conscious; they are produced by cognitive strategies and abilities. Secondary distortions can be made conscious; they occur due to knowledge, perspectives, views, or fears that are individual to every model designer. The resulting problem of isomorphy can be reduced by an iterative cyclic process where models are designed, tested and modified.

World 2 → World 3: Individual knowledge can be available in non-verbal pre-scientific form (e.g. feelings, intuition) as well as in verbal form (e.g. theories, conversation, presentation). This world 2 knowledge is externalized, whenever it is communicated to other people or represented in natural or formal language, and thus becomes part of world 3.

World 3 → World 2: The internalization of knowledge is done by learning and activating the contents of model representations in world 3. World 3 is not fictive, but real as every human is individually influenced by knowledge from world 3. It is important to distinguish between activated models (world 2) and only trans-subjective written knowledge (world 3). If the interpretation of a model in world 3 is activated or at least reconstructable in the individual consciousness of a world 2, this knowledge is not lost, otherwise only notations are available.

2.2 Definitions of terms and consequences

We define essential concepts and its consequences using Popper's three worlds in information systems.

2.2.1 Model

Within Popper's epistemological meta-model, the term "model" is split up into three parts. In this paper, the term "model" subsumes *the representation of models* (world 1), *the contents of a model* (world 3) and the *interpretation of a model* (world 2). Representations of models in world 2 can be distinguished from models in world 3 by their duration. Orally communicated models lead to short-term externalized models compared to those documented in written form (world 3).

2.2.2 Features

Within Popper's world 2 or 3, features can be assigned to objects of cognition belonging to Popper's world 1. A feature *F* is defined as a tuple of a *feature dimension D* and a *feature value V*, similar to the terms "variable", "type of a variable" and "value of a variable". A feature value *V* of a feature *F* is a concretization of a feature dimension *D*. Feature values can be divided into nominal (e.g. color), ordinal (e.g. date of expiry), cardinal (e.g. height) and logical ones (e.g. validity). Feature dimensions create the *domain* of all possible feature values.

The intuitive assignment of a *semantic network* of tuples of features *F(D, V)* to an object of cognition requires a mental will. The concept of this assignment is based on the evolutionarily affected way of how we perceive reality and the way of how we think in concepts.

Furthermore, we can divide features into two groups based on Aristotle's theory of reasons: intrinsic and extrinsic features. *Intrinsic features* are form (e.g. structure) and substance (e.g. material), and they are directly perceivable. *Extrinsic* ones, mode of operation (e.g. principle) and aim (e.g. function), are not directly perceivable. They can only be perceived by using an object in its proper context. This is done by putting the object in an imaginary black box and analyzing input and output. By doing this, we can determine the feature "aim". If the imaginary black box is opened, the working principle of the object can be detected.

2.2.3 Event

The feature values of objects of cognition are only partly static. Depending on the duration of the constancy of the values, (quasi-)static and dynamic features can be distinguished. A change of the value of a feature is called *event E* (e.g. change of the color of a traffic light). A unique name and time stamp can be assigned to every event ([HoKM00] 204).

3 Analogy and assumption of analogy – definition

During the history of evolution, the human brain has not been an organ that just acquires knowledge. The primary purpose of the human brain is to guarantee the survival of a human being. Our mental resources are limited compared to the unlimited universe. In order to survive, we have to make effective predictions. The cognitive strategies and structures that are developed during adaptation to world 1 are sufficient for the survival among competitors, but they are not perfect ([Voll98] 93). A possible cognitive (survival) strategy is the abstraction of individual phenomena to classes of similar phenomena. This is done by concentrating on essential features and dropping accidental ones. Types created inductively in this way can be deductively applied to the perceived reality in order to make predictions. For instance, predictions helped our forefathers to recognize a rock falling down and escape its consequences ([Ried80] 32 and 84). This stone-age way to derive a conclusion is symptomatic for humans. For the design of modern scientific models, it is still relevant. Therefore, efficient model design needs awareness of procedures, problems and boundaries of model design. In order to get this awareness, we discuss analogical classification (3.1), inductive type construction (3.2) and deductive use of types (3.3) in the framework of the three world model.

3.1 Analogy - coincidence of feature values

Analogical thinking is a cognitive strategy. It has the aim to put similar objects of cognition together in groups so that we can perceive structures (3.1.1). To make this possible, we need a similarity measure (3.1.2) which measures up to which degree objects in a group coincide depending on relevant parameters (3.1.3). We discuss these aspects in the following sections with the help of the terminology of data mining.

3.1.1 Arrangements in groups – how and why?

There are two procedures to divide a set of objects of cognition into groups. The *agglomerative* approach starts with small one-element groups. The objects with the greatest similarity are pooled together in one group (analogy). For the *divisive* approach we have one big group starting from which we separate dissimilar objects in different subgroups. Both procedures have the goal to construct groups with a high similarity within the groups, but a high dissimilarity between different groups. The essential similarity within such a group is the basis of model design, as this similarity is regarded as typical and, based upon it, a corresponding type is inductively constructed (3.2). The assumption, that a specific similarity (a set of common feature values) is representative for a type, seems to be arbitrary at first sight. Vollmer, however, shows that the assumption of structures of world 1 and their (at least partly) perceptibility are a plausible starting point which supports type construction ([Voll98] 28).

3.1.2 Similarity measure – distance functions and their parameters

To identify the similarity degree between two or more objects of cognition, we have to compare these objects. We need a *similarity measure* which measures the analogy degree between two objects of cognition on the basis of features. The result of a *distance function* is defined as this similarity measure. The value of the distance function is higher if the objects of cognition are more similar ([Moog85] 73). Let the variables of the distance function be the feature values of the feature dimensions of the objects of cognition. The result of the distance function is the *analogy degree* which measures the proximity of the objects. The higher the analogy degree, the higher is the proximity of objects inside a group. One possible distance function f is to *count the coincidences* of the feature values of two objects of cognition where V_{ij} is the feature value of a feature i of an object of cognition j:

$$f(V_{11}, V_{12}, V_{21}, V_{22}, ..., V_{n1}, V_{n2}) = \sum_{\substack{i=1..n \\ V_{i1}=V_{i2}}} 1$$

A result is that the two objects of cognition in Figure 1 have an analogy degree of 3 because they coincide in the three features "purpose", "material" and "speed". Riedl assigns this calculation to a "ratiomorphic apparatus" which manages our rational thinking and acting ([Ried80] 52). If a major role has to be assigned to special features (e.g. the causal behavior in process models), they are ranked with a *weighting factor* W. We get the following weighting function to calculate the analogy degree:

$$f(V_{11}, V_{12}, V_{21}, V_{22}, .., V_{n1}, V_{n2}) = \sum_{\substack{i=1..n \\ V_{i1}=V_{i2}}} W_i$$

We can use weighted features for the definition of system boundaries and for a solution of the isolation problem of model design ([Holl99] 192). Systems can be defined as areas in which certain combinations of features considered as important coincide. Weak system boundaries can be constructed where the feature combinations start to change on the margin of a problem area.

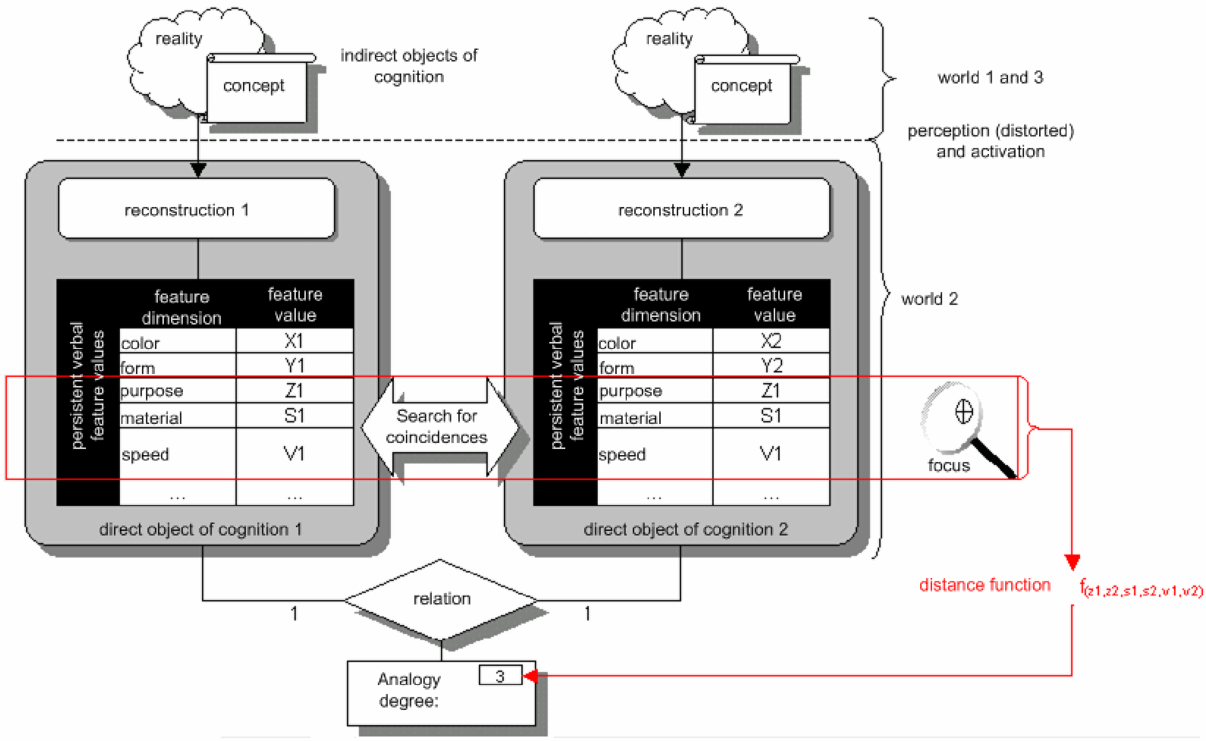

Fig. 1: Analogies are based on coincidences of features. Analogy is an attribute of the relation between two direct objects of cognition and their pendants in world 1 and 3.

The weighting function can assess the identity of feature values (e.g. equal shape) as well as coincidences of the simultaneity and succession of feature changes, that is the coincidence of events. We can divide, for instance, traffic lights into pairs because the light switches from red to green at the same time, whereas the process structure can be derived from the repetitive switching of these traffic light pairs. For Riedl, simultaneity means the simultaneous occurrence of features without a time component ([Ried80] 96), but in this paper, simultaneity means the simultaneous change of dynamic features. In order to recognize changes of feature values, we need a feature memory. We have to pay attention to the fact that implicit information about changes of features ("blinking") can be included in "static" feature values. With the help of temporal structures (sequences of events), we can drop the division between functional, structural and temporal analogies in favor of structural analogies.

3.1.3 Selection of relevant parameters

In order to judge direct objects of cognition, only a relevant subset of the features associated is used for the distance function. The number of coincidences inside this subset determines the analogy degree (Figure 1). Whether a feature is placed in the relevant subset or not, depends on the purpose of a model. Therefore, it is necessary to explain this purpose. The subset is constituted by feature dimensions which are selected according to explicit purposes. We need a temporal sequence of actions, e.g. regarding persons and departments involved, in order to design process models. The analysis method "*specific search for analogies*" is flexible regarding the selection of feature dimensions to be compared and can, therefore, be adapted to the model's purpose ([Schü95]). This is one important strength of this strategy of cognition as the analysis method is adapted to the task, not the other way around. Elements of models that are based on analogy (3.2) meet the requirement of relevance of proper modeling ([Schü95]).

3.2 Analogy induction – assigning equal to equal

General statements in terms of modeling need a differentiation between accidental and essential features (3.2.1) as well as the abstraction of accidental ones ([Holl99] 174). Structures perceived inside a group of analogous objects of cognition are considered as essential in the induction step. We define a new type using coinciding and, therefore, essential features (3.2.2) and dropping accidental ones. This method improves the aspects of economy, transparency and understandability of the model designed (3.2.3).

3.2.1 Differentiation between accidental and essential features

We can divide the features of objects of cognition in *essential distinctive* and *accidental non-distinctive* features. For this purpose, we have to analyze the analogous objects of cognition in a comparative way. Let O_i describe n analogous objects of cognition F be a verbal feature and V the feature value of this feature. Then we have following relations:

F essential $\leftrightarrow \forall 1 \leq i, j \leq n : V(F,O_i) = V(F,O_j)$

F accidental $\leftrightarrow \exists i, j : V(F,O_i) \neq V(F,O_j)$

A feature is essential if and only if its feature value is pair-wise identical for all objects of cognition inside the analogous group (e.g. purpose). A feature is accidental if a pair of objects inside the analogous group has different feature values (e.g. color).

3.2.2 Capturing of essential features in types

Isolation, definition and description of essential features lead to type construction. We can create a type by dropping accidental features and defining essential features (Figure 2). An example from information systems is the entity type "customer" which has essential features such as "name" and "address" ect.

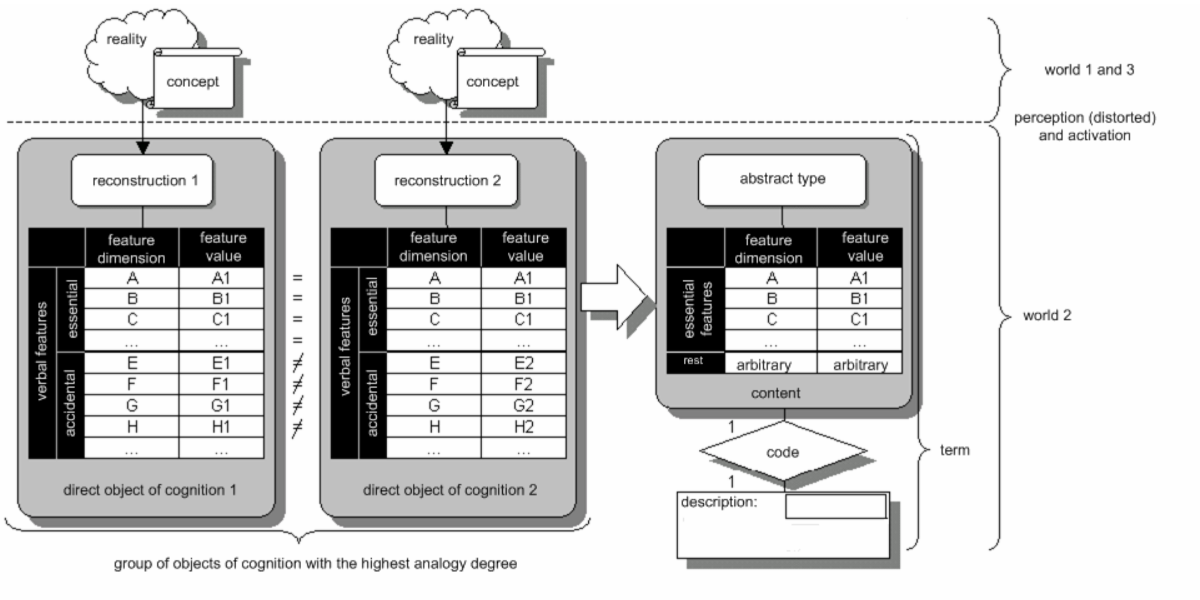

Fig. 2: Induction step due to postulated analogy

During type construction, we make a transition from specific (individual) features to general (common) features of objects of cognition. This method is generally called *induction*, in this special case *analogy induction*. If we assign a unique identifier to a type, we get a bilateral linguistic sign which consists of form (signifiant) and meaning (signifié): a term ([Dörn99] 222; cf. 2.2.1).

3.2.3 Advantages of types

The development of algorithms for information objects only makes sense if they fit for multiple uses. This reusability means that there must be a set of analogous problems a solution fits. During model construction, the model designer has to show all the analogous problems in some company. It is important for the designer to regularly look for analogies so that he or she does not have to design so many algorithms and software systems. For instance, one can avoid the double development of algorithms for customers and suppliers as both can be modeled together as one type called “business partners”.

Analogies improve the understandability and the transparence of a model, regarding both the model’s derivation and its internal structures. Then structures made evident with types (for instance object-oriented glossary) provide starting points for the decomposition of a model (cf [HoKM00] 205). For this reason, the cognitive strategy of analogical thinking meets Schütte’s modeling principle of *highest possible transparency* ([Schü95] 438). In Section 4.3, we deal with higher degrees of abstraction and the decrease of complexity and of the number of model components.

In addition to the transparency of a model, the explication and documentation of analogical thinking make it possible to compare the contents of models. Types that are explicitly defined with features form the semantic synchronization points for the comparison of two models. The range of interpretation of glossaries is restricted because of the explicit formal definition of the analogy used for type construction. That is why the identification of analogous components in other models is possible. This creates the basis for further inductive-abstract reference modeling.

3.3 Analogy deduction – pars pro toto

In addition to the inductive type construction, analogical thinking allows us to deductively use types. Once we know the essential features of a type, we can – on the basis of an analogy to a perceived or activated object of cognition (3.3.1) – try to draw a conclusion from a partial similarity in primary essential features (key features) to an advanced similarity in secondary essential features (3.3.2) in a *pars-pro-toto-step*. Therefore, the type becomes a reference model (3.3.3).

3.3.1 Classification analogy – identification of an object of cognition

We have a (partial) analogy if a direct object of cognition has special features that are equal to a subset of features of a learned or inductively designed type. Based on these key features, *identification and description* of an individual object of cognition (e.g. a perceived strawberry) with its type (“strawberry”) takes place. We speak of *classification analogy*. We can divide the essential features of types in *primary (key features)* and *secondary essential features*. This differentiation is not immanent in a type, but is assumed due to a partial classification analogy. Features are primary essential features if they are well-known or observed for some object of cognition and equal to the features of its type. Other well-known features of that type are called secondary essential features (cf. Fig 3; [Holl02] 2; [Lore43] 240).

3.3.2 Transfer analogy – assumption of analogy

When attempting to draw a conclusion from partial to extended similarity, one transfers one or more secondary essential features of a type to an analogous object and, therefore, implicitly to its pendant in world 1 and 3. With this step, the type becomes a reference model and the object becomes a use case. Accidental features are not affected by this transfer. That means we extend the domain of the partial classification analogy (in primary essential features) to a (complete) *transfer analogy* (in all essential features). We draw a conclusion from the general (the common) to the special (the individual). In general, the transfer of a feature from a type to an individual object is called *deduction*. In this case, as the transfer is based upon analogy, it is called *analogy deduction*. Due to the selection of features during type construction, the results of such conclusions are likely, but not mandatory. A discussion about function and limits of this method can be found in section 4.

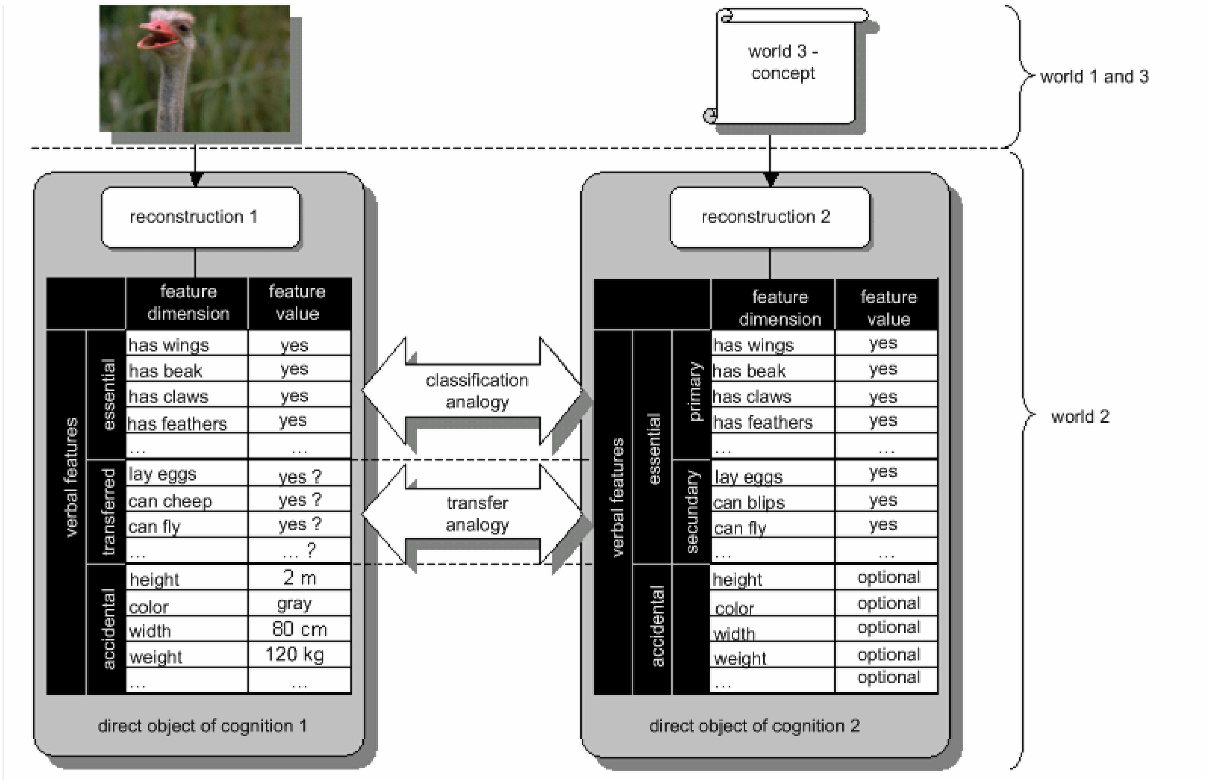

Fig. 3: Deductive conclusion with the help of analogy

A formal method for the correct logical transfer from individual features is Aristotle's *syllogisms*. For this type of logical conclusion, we need two formal statements called *premises*. The first one contains the classification analogy due to partial coincidence of two objects of cognition in essential (key) features (e.g., Socrates is a human). The second premise contains the assignment of further essential or common accidental features to one of the two objects of cognition (e.g., all humans are mortal). Using a conclusion of the type *Modus ponens* (analogy deduction), we get Socrates' mortality (a common accidental feature) as a result ([Dörn99] 271). The 19 possible logical conclusions explained in literature are useful for information systems to transfer reference models to the reality in companies.

We assume that we encounter an animal in the nature like in Figure 3 so that we can put it into a classification analogy because of its primary essential features (feathers, beak and wings). We recognize our well known type "bird" that has further (secondary) essential features, e.g. "can cheep". In a deduction step, these features are assigned to our perceptional reconstruction of the animal. The term "bird" becomes a reference model. "The perceived phenomenon corresponds to a bird. Birds can cheep". That means that the animal that we see can cheep even if that doesn't happen at the moment. This method works for common accidental features as well, such as "lays eggs"; this feature can be considered as not essential (not distinctive) as there are other vertebrates that also lay eggs, e.g. lizards, turtles, snakes. Regarding the feature "can fly", one can say that it is essential as there are only a few others vertebrates which can fly (bats). If one includes all animals, this feature is no longer essential, as there are lots of insects which can fly.

3.3.3 Reference model – transfer of complex types

We have to distinguish between *reference models* that are explicitly designed for reuse and models that become a reference model because of (analogy) deduction. The former ones are designed by inductive abstraction of several specific models and by deductive use of theoretical knowledge ([Schü95] 436). We can use reference models effectively if we can show sufficient analogies between the facts of a business system and models of other business systems. Therefore, the search for analogies becomes an economic paradigm. The *intensity of modeling* is limited due to the lack of money. Analogical thinking as a cognitive strategy, however, can lead to cost reduction as it saves money later in the software development process.

In the prescription phase we can achieve cost reduction with a less complex model design process by reusing parts of already existing reference models. Instead of producing new software components in the software design phase, we can use old ones and adapt them. In the descriptive phase, reference models can be used to minimize the risk of forgetting something: the software designers search for similarities between examined phenomena as well as between phenomena and well known concepts ([Ho99] 191). Using reference models deductively as *analysis patterns* can thus help to avoid future corrections and higher costs.

4 Nature, limitations and consequences of analogical thinking

In this section, we deal with the questions whether models designed with analogy induction are correct and represent reality adequately (4.1). Where are boundaries of using analogical thinking as a cognitive strategy (4.4)? Which consequences regarding model quality can arise from the use of analogical thinking (4.3)? In order to discuss these questions, we have to define the term “model correctness” and to deal with possible effects within the modeling process (4.2).

4.1 Correctness of information models

The *correctness* of an information model covers the aspects of syntactic as well as semantic correctness. A model is called *syntactically correct* if its model representation meets the syntactic rules of the modeling syntax (e.g. UML) used ([Schü95] 437). As this aspect belongs to knowledge representation and not to knowledge acquisition, the use of analogical thinking has no influence on the syntactic correctness of a model. *Semantic correctness*, however, is determined by comparing the *model’s structure and behavior* to the corresponding segment of reality. A model is semantically correct if its logical implications coincide with the reality observed.

A model designer only analyzes a subjective segment of reality (part of world 1). As one could only have observed special cases, Popper’s falsification theory judges statements on the basis of observations as mere hypotheses ([Voll98] 34). If a certain observation, however, is made more often, its reliability increases while the probability of an error decreases: the probability of the exactness of the corresponding model increases according to this maximum-likelihood-principle (see Fig. 4).

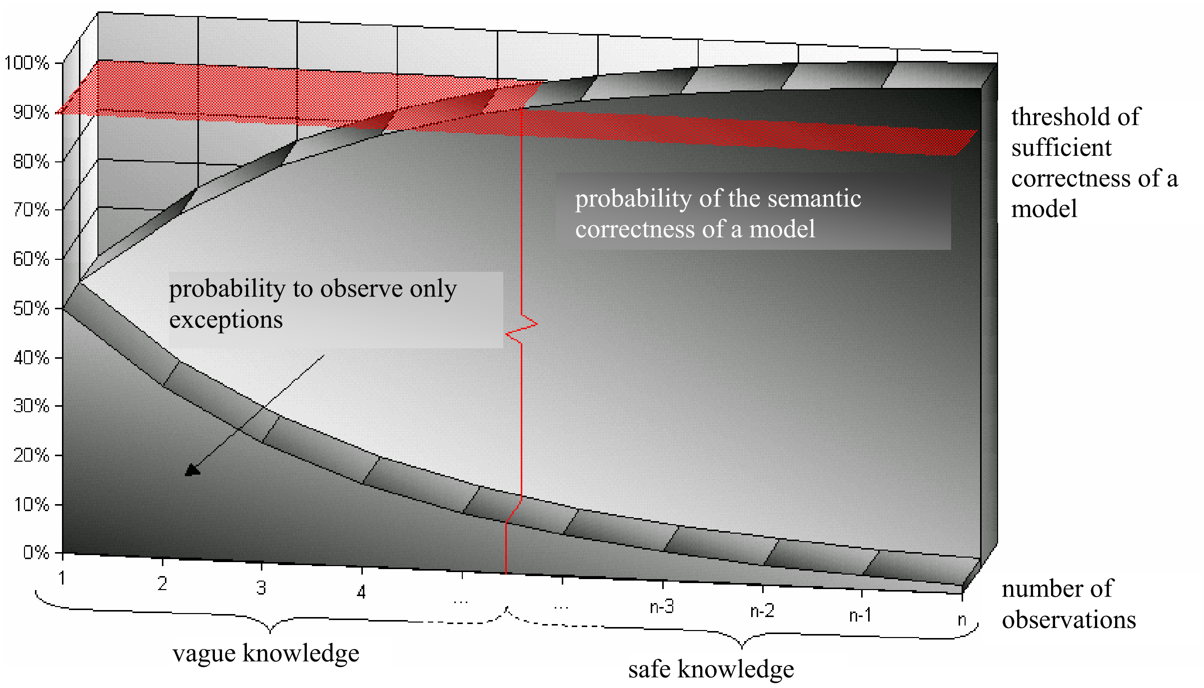

Fig. 4: Probability of semantic model correctness

Finally, the probability of an error can never be equal to zero. Therefore, the model one designed is always a preliminary model whose correctness one cannot prove ([Voll98] 38). The interpretation of the word “exactness”, however, can be changed from a final to a sufficient correctness ([Popp71] 47): one defines all the purposes of the model and checks if it meets them. If so, the model can be assumed to be sufficiently correct. As the check depends on the model’s particular purpose, it is subject to different criteria depending on the individual situation. Therefore, it is not possible to give a general definition of model parameters which have to be met in order to assume the sufficient correctness of a model.

4.2 Mayeutic cycles – approximative correctness

Since we do not have any final guarantee for a model to be correct, the correctness of a model is always only preliminary. Therefore, the correctness of models and theories is tested by deducing forecasts about reality using analogy deduction; the forecasts are verified using further observations. If the model reconstructs the perceived (distorted) reality incorrectly or insufficiently, the model is modified using the recently gained knowledge and tested again. In this *mayeutic cycle* (also called *Experiencial Learning Model* or literally “knowledge birth cycle”), the model is improved iteratively and approximated even better to the given reality.

Model correctness can only be achieved with regard to a threshold determined by the model’s purposes. In general, model correctness can never be reached completely ([Holl99] 189). The problem of isomorphy can be solved approximately using a scientific cyclic process in which the model is modified and tested. One-to-one images of reality are still not possible.

We can distinguish between two activities within a mayeutic cycle: *creating* and *modifying* models and theories as well as testing whether they represent reality adequately. These two steps are executed until the testing unit cannot falsify the created hypothesis any longer. After these steps, the model is sufficiently correct in relation to the reality perceived ([HoSc98] 11). This process can be compared to the concept of biological evolution: apparently, model modification corresponds to mutation while model test corresponds to selection. The interplay between principle of analogy and mayeutic cycle is shown in Fig. 5.

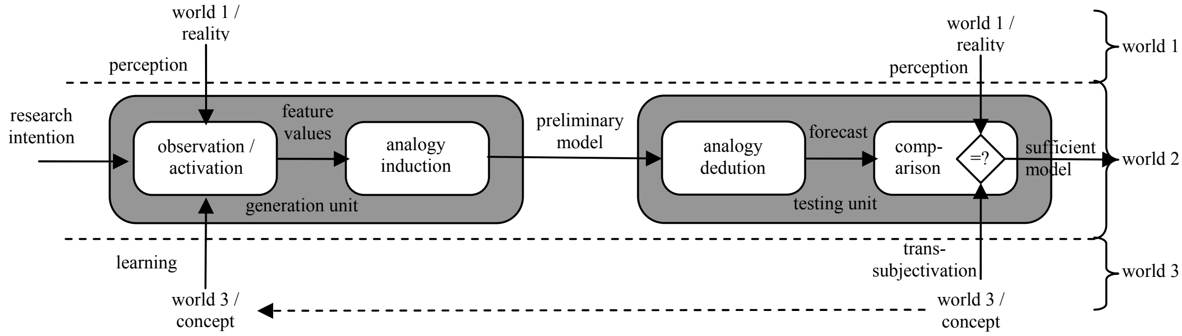

Fig. 5: Analogy induction and deduction in the context of a mayeutic cycle

We are now going to show a model which we created by observing nature (here: ostriches) and then modify it in a mayeutic cycle. We assume with a high probability that ostriches cannot fly, although we cannot be sure about this fact. Therefore, the more general type "bird" has to be modified with regard to the feature "can fly" in a mayeutic cycle. There are two possibilities to improve this type. Either we keep the previous term "bird" and classify the feature of "can fly" as accidental or we split the type "bird" into two new sub-types (birds which can fly and birds which cannot fly) where the feature "can fly" ("cannot fly") becomes essential. Thus, we have redefined the type "bird" using taxonomies: a bird is either a bird which can fly or a flightless bird. Caused by the higher coincidence of the feature values, we will now associate an ostrich to be a flightless bird and avoid the false conclusion of analogy deduction that every bird can fly. The model of the reality perceived is now better structured and more exact than before.

By embedding analogy induction and deduction in one or more endogenous (executed in the consciousness of the model designer) or exogenous (executed in relation to model trans-subjectivation and activation) mayeutic cycles, we can achieve sufficient model correctness within certain boundaries. Therefore, the method described is a possibility to meet the principle of (semantic) correctness of a model ([Schü95] 437). The boundaries mentioned above are discussed in 4.4 in more detail.

4.3 Repeated application of mayeutic cycles within a modeling process

If one has designed a model which describes reality sufficiently, the descriptive work of a model designer (information systems expert) normally ends and he starts to deduce a prescriptive model by optimizing the descriptive model. In the analyzing step, one possibly stated that there are different human categories in an enterprise; such as customers and suppliers, workers and employees. The majority of computer scientists would now immediately start to transform the achieved model into classes and program code. But the model can still be improved. One can start to analyze the given model and look for analogies within it. Doing this, one will realize that we can establish abstract types, such as business partners (customers and suppliers) or staff members (workers and employees).

Using analogies within models, we can define *generic terms* and *generalizations* in order to get even better models that are more efficient, free of redundancies, more transparent and better comprehensible. Additionally, they are easier to test. Using existing knowledge, one can acquire new insights about a model: the mayeutic cycle turns out to be a helix of knowledge acquisition ([Ried80] 74 and 104). We can compare this situation to a little child who learns its mother tongue: while the child grows, it reaches a critical point and starts to use language in order to extend its command of language ([Hofs91] 316).

These analyzing steps lift our models to qualitatively higher levels. In order to use our models, for instance, as a basis for mathematical models, we have to test whether they are mathematically well-defined and complete. A mathematical model, in turn, is the basis for axiomatisation. Axiomatisation is based on the analogies found in the mathematical model ([Hofs91] 39). Therefore, one of the main statements of this paper is the following: the modeling process should not stop at the end of the first phase of analysis. The developed model, however, should be made the basis for a new analysis which is, of course, less complex. Looking systematically for analogies within a model (or in relation to other well-known models) can support the design of more economic models and, therefore, reduce the costs for optimization and implementation.

4.4 Boundaries of analogical thinking

Analogical thinking cannot be used as a general strategy of cognition. We have to be aware of this and to check whether it is possible to use it. If we encounter phenomena which are either intentionally misleading (e.g. *mimikry*) or have their origin in a *chaotic system*, we can obviously not use analogical thinking. Chaotic systems are infinitely sensitive to their starting conditions: very small changes of the initial conditions are leading to completely different results (e.g. “wing beat of a butterfly” ([Davi88] 105). The similarity measure for the analogy degree cannot consider such tiny differences correctly. In these situations, type definitions achieved by induction as well as consequences achieved by deduction are subject to random to a great extent. We have to relativize these situations considering a huge set of observation data. The leveling of feature values (e.g. using mean values) allows other mathematical approaches, e.g. statistics ([Holl02] 2).

In contrast to objects of cognition in natural sciences, the objects of cognition in information systems have structures that are more easily perceivable for humans since they are created by humans. Business systems are frequently pre-formalized (e.g. accounting) or at least accessible to formalization (e.g. business processes). Business conditions confirm the actions of the actors to a calculable degree. With increasing complexity, however, we have to expect chaotic behavior (e.g. behavior of stock markets). In these areas, it is not possible to use analogical thinking as a knowledge gaining strategy.

5 Summary and Résumé

Similarities between perceived or activated indirect objects of cognition are considered to be typical. Starting from that, abstract direct objects of cognition are created. The coinciding features of analogous direct objects of cognition are defined as essential and constitute a (new) type. If we detect coincidences in primary essential features, we can transfer secondary essential features deductively in order to derive forecasts as well as in order to use reference models. The designed models can then be improved iteratively in mayeutic cycles using false forecasts. Verified types (e.g. customers) and the interrelations between them constitute sufficiently correct (information) models. With regard to chaotic systems, however, this cognitive strategy fails.

Analogical thinking is, of course, not the only method to build high-quality models. Compared to other alternatives, especially speculative or free associative modeling, we consider analogical thinking a very good cognitive strategy in the field of information systems. Nevertheless, model design remains a subjective process. Usually, the model designer is not aware of this fact. If model building is done using explicit type definitions based on analogies, it is at least followably subjective instead of only intuitively subjective. This is only a small step for the model designer, but a huge one for information systems.

6 Literaturverzeichnis

[Davi88] Davies, P.: Prinzip Chaos. Bertelsmann, München, 1988.

[Dörn99] Dörner, D.: Bauplan für eine Seele. Rowohlt, Hamburg, 1999.

[HoKr02] Holl, A.; Krach, T.: Ubiquitäre IT – ubiquitärer naiver Realismus? In (Britzelmaier, B. et al. ed.): Der Mensch im Netz – Ubiquitous Computing. Teubner, Stuttgart, 2002; 53-69.

[HoKM00] Holl, A.; Krach, T.; Mnich, R.: Geschäftsprozessdekomposition und Gestalttheorie. In (Britzelmaier, B. et al. ed.): Information als Erfolgsfaktor. Teubner, Stuttgart, 2000; 197-209.

[Hofs91] Hofstaedter, D.: Gödel, Escher, Bach – Ein endlos geflochtenes Band. dtv, München, 1991.

[Holl02] Holl, A.: Nutzen und Tücken von Analogieschlüssen in der Verbalmorphologie: Rückläufige Ähnlichkeit als tertium comparationis in ausgewählten romanischen und germanischen Sprachen. In (Heinemann, S. et al. ed.): Roma et Romania. Niemeyer, Tübingen, 2002; 151-167.

[Holl99] Holl, A.: Empirische Wirtschaftsinformatik und Erkenntnistheorie. In (Becker, J. et al. ed.): Wirtschaftsinformatik und Wissenschaftstheorie – Bestandsaufnahme und Perspektiven. Gabler, Wiesbaden, 1999; 163-207.

[HoSc98] Holl, A.; Scholz, M.: Objektorientierung und Poppers Drei-Welten-Modell als Theoriekerne in der Wirtschaftsinformatik. In (Schütte, R. et al. ed.): Wirtschaftsinformatik und Wissenschaftstheorie – Grundpositionen und Theoriekerne. Universität Essen, 1999; 91-105.

[Lore43] Lorenz, K.: Die angeborenen Formen möglicher Erfahrung. In: Zeitschrift für Tierpsychologie 5 (1943); 235-409.

[Moog85] Moog, W.: Ähnlichkeits- und Analogielehre. VDI, Düsseldorf, 1985.

[Popp71] Popper, K.: Logik der Forschung. Mohr, Tübingen, 1971.

[Ried80] Riedl, R.: Biologie der Erkenntnis. Paul Parey, Berlin / Hamburg, 1980.

[Schü95] Schütte, R. et al.: Grundsätze ordnungsgemäßer Modellierung. In: Wirtschaftsinformatik 37 (1995); 435-445.

[Voll98] Vollmer, G.: Evolutionäre Erkenntnistheorie. Hirzel, Stuttgart, 1998.