
Mining HCI Data for Theory of Mind Induction

Oksana Arnold and Klaus P. Jantke

Additional information is available at the end of the chapter

<http://dx.doi.org/10.5772/intechopen.74400>

Abstract

Human-computer interaction (HCI) results in enormous amounts of data-bearing potentials for understanding a human user's intentions, goals, and desires. Knowing what users want and need is a key to intelligent system assistance. The theory of mind concept known from studies in animal behavior is adopted and adapted for expressive user modeling. Theories of mind are hypothetical user models representing, to some extent, a human user's thoughts. A theory of mind may even reveal tacit knowledge. In this way, user modeling becomes knowledge discovery going beyond the human's knowledge and covering domain-specific insights. Theories of mind are induced by mining HCI data. Data mining turns out to be inductive modeling. Intelligent assistant systems inductively modeling a human user's intentions, goals, and the like, as well as domain knowledge are, by nature, learning systems. To cope with the risk of getting it wrong, learning systems are equipped with the skill of reflection.

Keywords: user modeling, inductive modeling, theory of mind, theory of mind induction, theory induction, inductive learning, identification by enumeration, logical refutation

1. Introduction

The present work originates from the authors' earlier work in the field of digital games with emphasis on the impact of game play, in general, and on game-based learning, in particular [1–5]. The original approach has been generalized, and algorithms have been extended toward business applications far beyond the limits of gaming [6]. Essentials have been carried over to the study of scenarios of data analysis, visualization, and exploration [7, 8].

Motivated by questions for the impact of playing digital games, the authors analyzed game play represented as (sets of) sequences of actions. Seen from the application point of view, the

task in focus is player modeling or, more generally, user modeling. Seen from the data point of view, it is string mining. Seen from the viewpoint of algorithms deployed, the task is pattern inference.

To achieve a high expressiveness, the authors prefer logical terminology powerful enough to approximately represent human goals, intentions, preferences, desires, fears, and the like. Seen this way, the task is theory induction, and the method is hypotheses refutation [9, 10].

2. From software tools to intelligent assistant systems

No doubt, *digitalization* pervades nearly every sphere of life. Humans are facing more and more digital systems at their workplaces, in everyday education, in their spare time, and in health care. With the US Food and Drug Administration's approval of aripiprazole tablets with sensors in November 2017 [11], the digitalization reaches the inside of the human body.

Frequently, the process of digitalization is placing on humans the burden of learning about new digital systems and how to use them appropriately. More digital systems do not necessarily ease the human life. To use them effectively, users need to become acquainted with software tools, have to understand the interfaces, and have to learn how to wield the tools. "A tool is something that does not do anything by itself unless a user is wielding it appropriately. Tools are valuable for numerous simple tasks and in cases in which a human knows precisely how to operate the tool. Those tools have their limitations as soon as dynamics come into play. There are various sources of dynamics, such as a changing world or human users with different wishes, desires, and needs" (see [12], p. xii). As the present authors put it earlier, the digitalization process "bears abundant evidence of *the need for a paradigmatic shift from digital tools to intelligent assistant systems*" (see [7], p. 28).

Thinking about human assistance, the most helpful assistants are those who have own ideas, go their own ways, and—from time to time—surprise us with unexpected outcomes. This does apply to digital assistant systems as well.

Approaches to intelligent system assistance are manifold (e.g., see [13, 14] and the references therein including the authors' contributions [15, 16]).

Digital assistants are programmed to behave differently in different conditions such as varying environmental or infrastructure contexts and varying human users with different prior knowledge, preferences, skills, needs, desires, fears, and the like. To adapt accordingly, *assistant systems need to learn* from the data available. In a sense, a digital assistant system has "to ask itself," so to speak, how to learn what the user needs from sparse information such as mouse clicks or wisps over the screen.

Seen in its right perspective, digital assistant systems are facing problems of learning from incomplete information sometimes called *inductive inference* [17]. Digital assistant systems are necessarily *learning systems*.

The purpose of the system's learning is understanding the context of interaction to adapt to. In this chapter, the authors confine themselves to understanding the human user.

Conventionally, this is called *user modeling* naming a rather comprehensive field of studies and applications (see, e.g., [18–22] or any of the earlier UMAP conference proceedings).

By way of illustration, [23] provides a comprehensive digital game case study of mining large amounts of human-computer interaction data—in fact, data of game playing behavior—for the purpose of classification according to psychologically based personality traits [24].

This exemplifies a particular way of user modeling by means of HCI data mining.

3. Perspectives at inductive modeling and data mining

3.1. Approaches and opinions

Already for decades, the misconception of *data mining as digging for golden nuggets* is spooking through the topical literature [25, 26]. Some authors believe that data mining means somehow squeezing out insights from the given data and put this opinion in words such as “visualization exploration is the process of extracting insight from data via interaction with visual depictions of that data” (see [27], p. 357).

Instead, *data mining is a creative process of model formation based on incomplete information* (see [7], p. 108). In brevity, *data mining is inductive modeling*.

The details of the inductive modeling process depend substantially on the data, on the goal of modeling, of the algorithmic technologies in use, and on the underlying model concept including the syntax of representing models [28, 29]. For going into the details of the model concept, [30] is of particular interest.

In the thesis [30, 31] is taken as a basis for a systematic framework of data mining design in which the model concept resides in the center. An outer frame, so to speak, consists of the application domain, the methods of model construction, and the methods of model use. Every concrete model depends (a) on the context of modeling, (b) on communities of practice, (c) on the purpose of modeling, and, possibly, (d) on models generated earlier (see [31], p. 37). This covers Fayyad’s KDD process [32] and the CRISP data mining model [33].

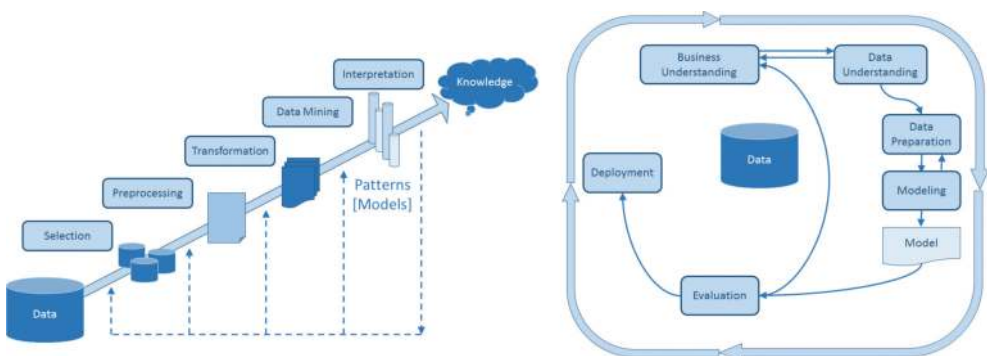


Figure 1. Fayyad’s KDD process according to [32] vs. the CRISP data mining model as in [33].

With respect to the difficulty of learning from incomplete information, process models of data mining allow for cyclic processing as shown in **Figure 1**. Consequently, data mining has to be seen as a process over time that does not result in an alone model, but in an unforeseeably long *sequence of subsequently generated hypothetical models*. This does perfectly resemble the learning systems perspective of [17].

In the authors' opinion, the thinking about *emerging sequences of hypotheses* is badly underestimated in contemporary data mining investigations. Pondering model concepts is not sufficient. We need to put emphasis on the investigation of suitable *spaces of model hypotheses*.

3.2. Theories of mind

Throughout the rest of this chapter, spaces of hypotheses will be *spaces of logical theories*. The concept *theory of mind* is adopted and adapted from behavioral research in animals [34]. There is much evidence that certain animals reflect about intentions and behaviors of other animals [35, 36]. Birds of the species *Aphelocoma californica*—the western scrub jay, esp. the California scrub jay—are food-caching. They do not only cache food but also colorful and shiny objects such as plastic toys. In case such a bird, let us name it A, is caching food or other treasures, and if it is watched by another bird of its species, we name it B, then A returns shortly after to unearth the treasures cached before. The interpretation is, loosely speaking, that the bird A thinks about the possibly malicious thoughts of the bird B. It builds its own theory of mind. More generally speaking, thinking about another one's thoughts means to build a theory of mind.

The authors aim at digital assistant systems able “to understand their human users” by hypothesizing theories of mind. Anthropomorphically speaking, digital assistant systems shall be enabled “to think about their human user's thoughts.” The cornerstone has been laid in [1, 2]. Case studies as in [4, 5, 37] demonstrate that this is possible.

For this purpose, user models are seen as theories—just formalizations of theories of mind—such that human user modeling becomes theory induction. The conceptual approach is called *theory of mind modeling and induction* [1, 2] based on human-computer interaction data.

3.3. Data mining as theory induction based on HCI data

What the system “knows” about its human user comes from an analysis of interaction data. [3] describes a study based on a commercial digital game. When playing the game, players may learn about pieces of legerdemain. They play successfully when being able to script the necessary steps for doing conjuring tricks. Patterns of game playing behavior reveal the human players' success or failure. Instances of those patterns are shown in recorded game play. It is the system's task to learn patterns from their instances. This approach is generalized toward theory induction.

The concept of a pattern in science dates back to work by Alexander in architecture [38, 39]. Angluin redefines the pattern concept for purposes of formal language investigations and,

most importantly, provides algorithmic concepts for learning patterns from instances [40]. Exactly this is what an assistant needs to do when collecting sequences of interaction data.

In game studies such as [3, 5], interaction is abstractly represented by finite sequences over an alphabet A that contains identifiers of all possible activities of all engaged agents such as human players, non-player characters (NPCs), and other computerized components. A^+ denotes the set of all those strings, and given any particular game G , $\Pi(G)$ is the subset of all strings that can occur according to the rules and mechanics of G . Angluin's pattern concept describes string properties of a certain type. If instances $\omega_1, \dots, \omega_n \in \Pi(G)$ occur, the learning task consists in finding a pattern p that holds in all the observed strings. In a sense, p is a theory with $\{\omega_1, \dots, \omega_n\} \models p$, where \models denotes the logical consequence operator.

The authors generalize the before-mentioned approach toward human-computer interaction in general beyond the limits of game play as undertaken in [6, 7].

The validity of the logical expression $\{\omega_1, \dots, \omega_n\} \models p$ means *consistency* of the hypothesis p with the set of observations $\Omega_n = \{\omega_1, \dots, \omega_n\}$ it is built upon. In conditions more general than pattern inference according to [40], the choice of the logic is decisive to consistency. In the conventional case, the question for consistency $\Omega_n \models p$ is recursively decidable but NP-hard. Concerning the background of computability and complexity, the authors rely on [41, 42]. For the moment, let us assume any suitable logic. Details will be discussed as soon as they become interesting.

A closer look at conventional data mining process models as in **Figure 1** reveals that original data appear somehow static. Both models on display show data represented by a drum icon. An emergence over time is beyond the limits of conventional perspectives. In contrast, human-computer interaction data emerge over time [3, 5, 7]. This leads to the learning task of processing sequences $\Omega_1 \subseteq \Omega_2 \subseteq \dots \subseteq \Omega_n \subseteq \dots$ of growing finite data sets of observations. When learning patterns according to [39], the learner returns hypotheses $p_1, p_2, \dots, p_n, \dots$ such that every hypothesis p_n is consistent with the underlying data set Ω_n .

Consistence is a critical requirement and may be refined by approximations in different ways. In learning theory, it is known that algorithms that are allowed to temporarily return inconsistent hypotheses are of higher effectiveness [17, 43, 44]. The authors refrain from a detailed discussion of these effects, for reasons of space.

Extending the abovementioned approach, one arrives at an understanding of mining HCI data as the induction of theories over emerging sequences $\Omega_1 \subseteq \Omega_2 \subseteq \dots \subseteq \Omega_n \subseteq \dots$ of data. The result is a corresponding sequence of logical theories $T_1, T_2, \dots, T_n, \dots$ which, if possible, should converge to an ultimate explanation T of the observed human-computer interaction, i.e., $\Omega_n \models T$ for all sets of observations.

By way of illustration, [5] is based on a case study in which the sequence of theories begins with some default T_0 and consists of 35 subsequent hypothetical theories. In this sequence, subsequent theories remain unchanged frequently. There are only nine changes of hypotheses. The final theory reasonably explains the overall human user's behavior.

Note that there are varying other approaches to deal with dynamics such as [45, 46]. The authors, however, stick to the logical approach for its declarativity and expressiveness.

Different from other approaches, they dovetail logical reasoning and inductive inference [17, 43]. In this way, logics and recursion theory are underpinning data mining on HCI data.

3.4. Formalization and operationalization of theory induction on HCI data

The logical background of the authors' approach includes reasoning about changes in time. This leads directly to temporal logics that are around for already more than half a century [47, 48]. In these good old days, time was tense, but in conditions of digitalization, time became digital as well [49]. It was already known before that this makes a difference [50].

In the simple digital game case study [4, 5], it is sufficient to choose the Hilbert-style logic K (see [49], Section 1.6).

Which logic to choose depends on the particular domain of application. In particular, there is an indispensable need (i) to formalize background knowledge. The logic must allow for the representation of knowledge in such a way that it is easy (ii) to refute hypotheses [9, 10]. Below, we will come back to these two issues. Logics taken into account come from [49–56]. For the generic approach discussed in this section, however, the choice is subordinate.

Speaking about human-computer interaction with the intention of user modeling by theories of mind, the fundamental question is what to take into account. Interaction may be represented on largely varying levels of granularity [57] ranging from keystrokes and wisps over the screen through compound actions to activities on a task level (named quests in the world of digital games, where the approach originated). The authors are engaged in a joint project in which even the exact position of a document on the screen plays a role in mining HCI data and, thus, must be documented in interaction representations [58].

The (finite) set of actions of interest is denoted by A . It is considered an alphabet. As usual in theoretical computer science, A^* denotes the set of all finite strings over A including the empty string ϵ , $A^+ = A^* \setminus \{\epsilon\}$. If it makes sense, one may restrict A^+ to the set Π of only those sequences that can occur in practice, $\Pi \subseteq A^+$. In conditions of strictly regulated interaction possibilities, Π is a formal language [59].

Every string $\pi \in \Pi$ abstractly represents some process of human-computer interaction such as a game play [3] or a session with a data analysis tool [7]. When trying “to understand the human user,” π is subject to investigation. Sometimes, there is a finite subset of Π available. By way of illustration, see **Figure 2** adopted and adapted from ([3], p. 89, Figure 6.3).

According to the game magazines worldwide, allegedly, the innovation of the commercial game studied in [3] consists in the unprecedented feature of players doing conjuring tricks. To investigate this in more detail, the alphabet A of actions contains player actions such as clicking to a magic head (denoted by mh in the strings on display in **Figure 2**), opening a grimoire, in the game called a magic book (mb), turning pages of the book when searching for an appropriate trick (tp), selecting a trick from the book (st), scripting the steps of the trick in preparation of performance (sc), and presenting the trick by means of a magic wand either successfully (mw) or not ($mw-$). The digital game system's actions indicated by boxes ranging from yellowish to reddish in **Figure 2** are the presentation of a magic head (mh) indicating that

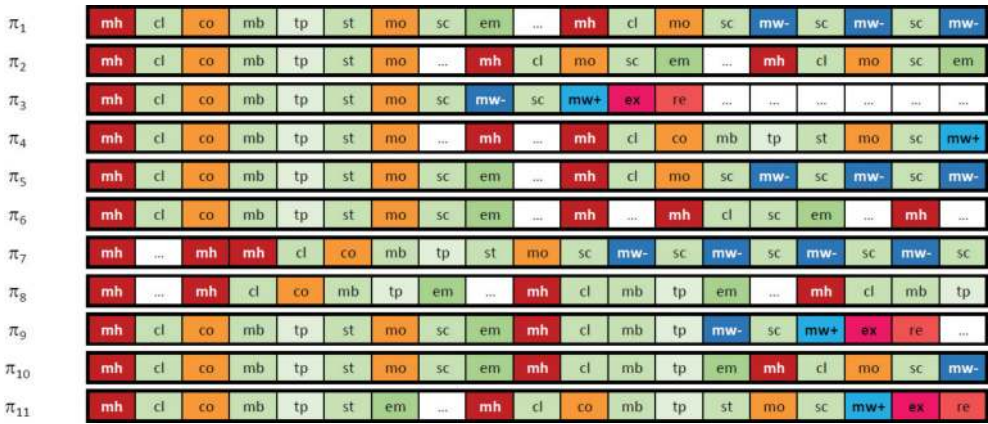


Figure 2. Excerpts from recorded game play of 11 subjects striving to do a conjuring trick.

there is an opportunity of witchcraft, a comment (co) in response to a human user’s click to inform the player what to do, the opening of the magic book (mo) to allow for scripting tricks, and, in case the trick has been scripted correctly and the user has triggered its execution by clicking to the magic wand, a virtual execution (ex) of the trick by means of a cut scene and some response (re) to the player about the success of the performance.

The (cutouts of) strings on display in **Figure 2** have different properties that are indicators of the players’ mastery of game play, in general, and of scripting tricks, in particular [3]. For a precise and readable treatment, actions in A are written in brackets such as [mh] and [ex]. [...] abbreviates an action not of interest. Using this convention, the cutout of π_1 is [mh][cl][co][mb][tp][st][mo][sc][em][...] [mh][cl][mo][sc][mw-][sc][mw-][sc][mw-]. Readers may easily recognize that the player has a problem. The substring [sc][mw-] indicates a failed effort of scripting and doing a conjuring trick. This will be discussed in some detail.

Suppose that \leq denotes the substring relation. $\pi_1 \leq \pi_2$ means that there are (possibly empty) strings π' and π'' satisfying $\pi' \pi_1 \pi'' = \pi_2$. In other words, π_1 occurs somewhere in π_2 .

By way of illustration, the following two sample formulas $\varphi_2 = [sc][mw-][sc][mw-] \leq \pi$ and $\varphi_3 = [sc][mw-][sc][mw-][sc][mw-] \leq \pi$ describe certain string properties. This justifies logical expressions such as $\pi \models \varphi_2$ and $\pi \models \varphi_3$ meaning that the string π satisfies the corresponding property. It is custom to say that π is a model of φ_2 or φ_3 , respectively. The intuitive meaning is quite obvious. When φ_3 occurs in a string π describing human game play, the player appears to stab around in the dark. According to **Figure 2**, it holds $\pi_1 \models \varphi_3$, $\pi_5 \models \varphi_3$, and $\pi_7 \models \varphi_3$. Properties of this type are called *patterns*. Patterns according to Angluin [40] are properties of strings that are decidable. This does obviously apply to both φ_2 and φ_3 as well.

Because the information about the other eight strings of game play in **Figure 2** is incomplete, we are not sure whether or not one of the patterns φ_2 and φ_3 is satisfied. With respect to the information available, all we know is that we are *not* able to *disprove* one of these patterns.

Needless to state that in computational logics, double negation cannot be removed [60, 61]. In other words, $(\neg\neg p \rightarrow p)$ is no valid axiom of (propositional) computational logics.

Jantke [37, 62] has developed a family of games based on the Post correspondence problem (see [63], Section 2.6, pp. 88ff). Patterns that occur in game play are of higher complexity than those sketched above. Computational learning of these patterns—theory of mind induction—is possible but considerably more involved. As illustrated in [64], a computer program may even learn skills the human player is not aware of. The authors confine themselves to a sketch of the essentials of PCP games within the following four paragraphs.

A Post correspondence system (see [63], p. 89)—in PCP games, this is called a pool—is a finite set of pairs of strings that may be visualized as dominos. Some of these systems have solutions; others have not. Playing a PCP game means to incrementally modify a common pool according to some rules of play. The goal is to make a pool solvable and to prevent others from doing so. Who makes a pool solvable and declares victory accordingly wins the game by showing a solution. If the player's demonstration fails, the game is lost.

Interestingly, the solvability of Post correspondence systems is algorithmically undecidable (for a comprehensive treatment of undecidability, [65] is recommended). As a consequence, a player might be unaware of being able to declare victory and to win the game accordingly. There is the phenomenon of missing a win. This may occur repeatedly.

Using elementary formalizations (see [62, 64] for details), one may write down formulas φ_n of first-order predicate calculus saying that a player never misses more than n wins in a game. Whether or not ψ_n holds in recorded game play π is effectively undecidable. But the problem is effectively enumerable (some call it semi-decidable).

Therefore, a computer program can watch a human playing PCP games. It can analyze strings describing the human-computer interaction for the occurrence of missing wins. The program's first hypothesis may be ψ_0 . In case a missing win is detected, the hypothesis is changed to ψ_1 . If ψ_n is hypothesized, but one more missing win is diagnosed, the hypothesis is changed to ψ_{n+1} . The underlying process is identification by enumeration [66].

Let us have a look—quick and dirty—at the principle of identification by enumeration from a logical viewpoint. A space of hypotheses is an effective enumeration $T_0, T_1, T_2, T_3, T_4, T_5, \dots$ of theories; in the paragraph before, these theories are the singleton sets $\{\psi_n\}$. When sets of observations $\Omega_1 \subseteq \Omega_2 \subseteq \Omega_3 \subseteq \Omega_4 \subseteq \Omega_5 \dots$ come in subsequently, learning means to search the given enumeration of hypotheses for the first theory that does not contradict the current information. Formally, a learner L getting fed in Ω_n searches for $k = \mu m [\neg(\Omega_n \not\equiv T_k)]$ and hypothesizes $L(\Omega_n) = T_k$. The symbol μ represents the minimum operator [41].

As explicated already much earlier [67], the key logical reasoning problem in learning from incomplete information is refutation. This is sound with related philosophical positions [11]. The crux is that $\neg(\Omega_n \not\equiv T_k)$ is usually undecidable as seen in the PCP game case study. This leads to the authors' original pattern concept. Whereas in [3, 68]—adopted from [40]—the assumption is that the validity of a pattern in a stream of HCI data is decidable, the ultimate

approach weakens the requirement (see [37], p. 12): Patterns are logical theories that are co-semi-decidable. In other words, under the assumption of an underlying logic with (i) its consequence operator \models , (ii) the operator's implementation \vdash , (iii) background knowledge, and (iv) current observations, the implementation \vdash may be used to find out in a uniform way whether any set of observations and any theory are inconsistent. Furthermore, according to scenarios of analyzing human experience of patterns in HCI data [69], patterns should have the property of locality. Informally, once a pattern instance occurred, it does not disappear throughout subsequent interaction. In formal terminology, for any pattern φ and for any $\pi_1, \pi_2 \in A^*$, the validity of $\pi_1 \models \varphi$ implies the validity of $\pi_1\pi_2 \models \varphi$.

3.5. Theory of mind model induction via identification by enumeration

To sum up, theory induction on HCI data is operationalized by construction of theories and sticking to them as long as they are not refuted. The underlying decisive knowledge forms an effectively enumerable space of hypotheses. In formal language learning, the appropriate technical term is called an indexed family of formal languages [70]. For the purpose of theory induction, this concept has been slightly generalized. The authors coined the term of an indexed family of logical formulas [5]. Because logic in general is more expressive than formal languages are, there is a need for requirements that are weaker but still sufficient to allow for inductive learning.

Assume any logic that does not exceed the expressive power of first-order predicate calculus to allow for a completeness theorem [71]. The logic brings with it its well-formed formulas, its consequence operator \models and the operator's implementation \vdash (due to completeness). Practically, refutation completeness is sufficient [67]. By way of illustration, the authors' recent application uses Horn logic and relies on the refutation completeness of Prolog [72].

Given domain-specific background knowledge BK, an indexed family F of logical formulas is defined by the following conditions. $F = \{\varphi_n\}_{n=0,1,2,\dots}$ such that the sequence of formulas φ_n is effectively enumerable. Furthermore, for any two indices m and n with $m < n$, the formula that occurs later in the enumeration does not imply the earlier one, i.e., $BK \not\models (\varphi_n \rightarrow \varphi_m)$.

Note that the sequence of formulas $\{\psi_n\}_{n=0,1,2,\dots}$ discussed in the context of PCP games above meets the conditions and, thus, is an example of an indexed family of logical formulas. The corresponding background knowledge comprises the rules of play including Peano arithmetic.

Apparently, the authors' approach is a two-stage process above the granularity of the more conventional processes depicted in **Figure 1**. First, one selects an effectively enumerable space of hypotheses. Second, one performs identification by enumeration as the key learning methodology. Other conventional steps such as data selection, data preparation, and data preprocessing occur as well [8]. However, the latter are not in focus of this chapter.

As the choice of spaces of hypothetical models—an issue ignored in conventional approaches—is decisive, it is worth to take updates and revisions into account. The authors introduced a generalization for which they coined the term dynamic identification by enumeration [73].

4. An inductive inference perspective at mining HCI data

In contrast to earlier approaches that are widespread (see **Figure 1**, where in the CRISP-like model on the right, the “model” node is hatched, as it is missing in the original figure [33]), the authors stress the aspects illustrated by the (four groups of) darker boxes in **Figure 3**. First of all, data are not seen as a monolithic object within the process concept but as an emerging sequence. Second, whereas in the Fayyad process (see Figure [1] and the source [32]), the pattern concept appears from nowhere, the terminology of forming hypotheses is seen a central issue—the selection of a logic and the design of suitable spaces of hypotheses, both potentially subject to revision over time. Third, the inductive modeling procedure discussed in some more detail throughout this chapter is identification by enumeration.

Involved logical reasoning may easily become confusing—not so much to a computer or to a logic program [4], but to a human being. Within the digital game case study [4, 5], the generation of a single indexed family of logical formulas has been sufficient. Identification by enumeration works well for identifying even a bit perfidious human player intentions. Business applications as in [6] are more complex and may require unforeseeable revisions of the terminology in use, i.e., the dynamic generation of spaces of hypotheses on demand [73].

The present section is aimed at a clarification of the core ideas and technicalities. For this purpose, the approach is stripped to the essentials. Recursion-theoretic inductive inference as in [17, 43, 44] is the most lucid area in which problems of inductive learning can be explicated

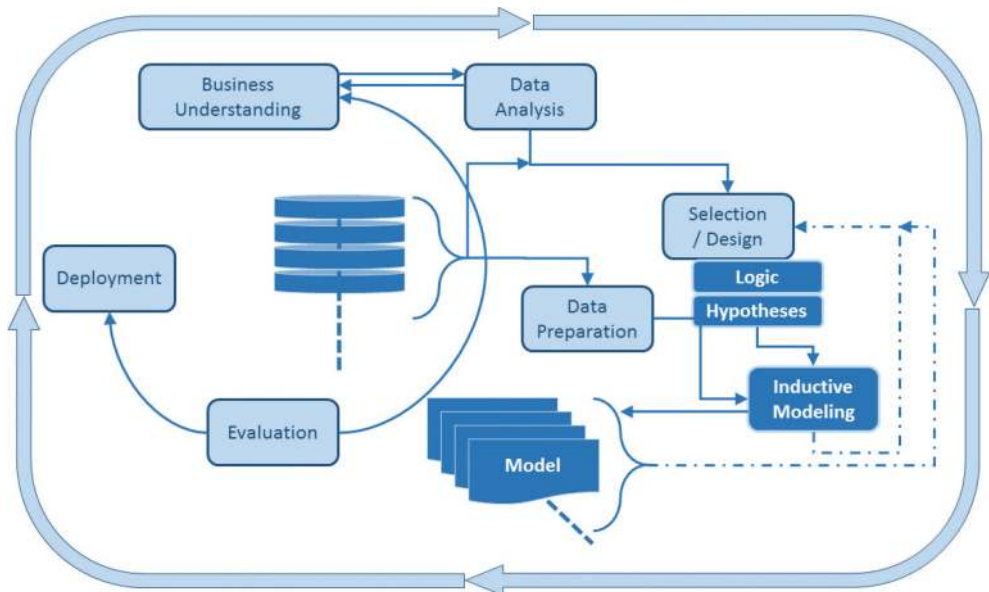


Figure 3. HCI data mining approach with emphasis on aspects of inductive modeling.

without any need for dealing with syntactic sugar. The underlying apparatus of mathematics can be found in textbooks such as [41, 63].

In **Figure 4**, the darker boxes with white inscriptions denote conventional concepts of recursion-theoretic inductive inference [44]. The other boxes reflect formalizations of this chapter’s core approaches to HCI data mining by means of identification by enumeration. The concepts derived from the present chapter’s practical investigations form a previously unknown infinite hierarchy between the previously known concepts NUM and TOTAL.

Throughout the remaining part of this section, the authors confine themselves to only elementary concepts.

Learning logical theories is very much like learning recursive functions. Both have finite descriptions but determine a usually infinite amount of facts—the theorems of a theory and the values of a function, respectively. In both cases, the sets of facts are recursively enumerable but usually undecidable. The deep interplay of logic and recursion theory is well understood for almost a century and provides a firm basis of seminal results [74]. Inductively learning a recursive function means, in some sense, mining the function’s graph which is presented in growing chunks over time, a process very similar to mining HCI data.

A few notions and notations are inevitable. \mathbb{N} is the set of natural numbers. P^n denotes the class of n -ary partial recursive functions mapping from \mathbb{N}^n into \mathbb{N} . $R^n \subset P^n$ is the subclass of all total recursive functions. Assume any ordering of \mathbb{N} written in the form $X = \{x_0, x_1, x_2, \dots\}$. For any function $f \in R^1$, the sequence of observations $(x_0, f(x_0)), (x_1, f(x_1)), (x_2, f(x_2)), (x_3, f(x_3)), \dots$ provides growing but incomplete information about f . With respect to the ordering X , the amount of information up to the timepoint n is encoded in $f_X[n] = ((x_0, f(x_0)), \dots, (x_n, f(x_n)))$. If X_0

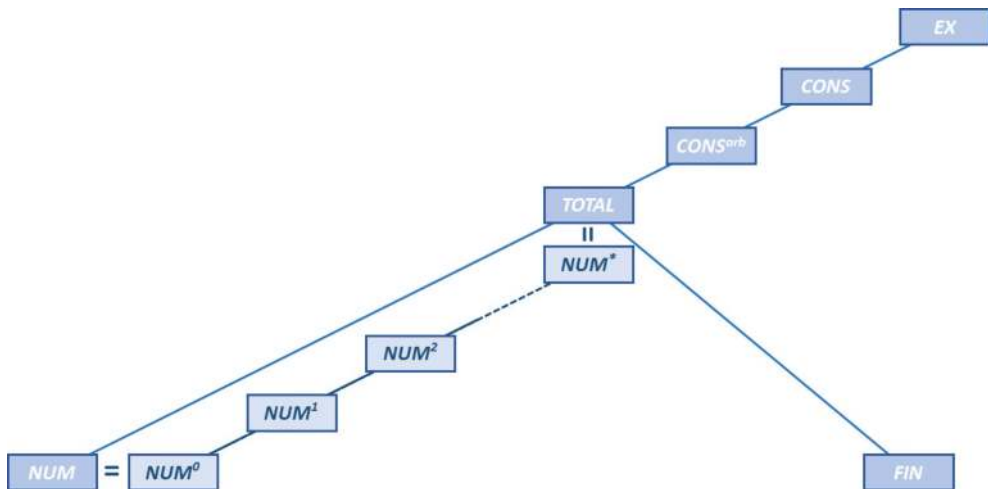


Figure 4. Abstractions of fundamental inductive learning concepts compared and related; ascending lines mean the proper set inclusion of the lower learning concept in the upper one.

is the standard ordering $0,1,2,3,4, \dots$, the index is dropped such that the notation is $f[n]$. Throughout any learning process, hypotheses are natural numbers interpreted as programs according to some Gödel numbering φ . Because any two Gödel numberings are recursively isomorphic, the choice of a numbering does not matter. Learnability is transcendental.

Assume any class $C \subset R^1$ of total recursive functions. The functions of C are uniformly learnable by an effectively computable learner $L \in P^1$ on the ordering of information X_0 , if and only if the following conditions are satisfied. For all $f \in C$ and for all $n \in \mathbb{N}$, the learner computes some hypothesis $L(f[n])$. For every $f \in C$, the sequence of hypotheses converges to some c that correctly describes f , i.e., $\varphi_c = f$. EX denotes the family of all function classes learnable as described. $EX(L) \in EX$ is the class of all functions learnable by L . In the case arbitrary arrangements of information X are taken into account, the definition is changed by substituting $f_X[n]$ for $f[n]$. The class of functions learnable by L is named $EX^{arb}(L)$, and the family of all function classes learnable on arbitrary X is EX^{arb} . The term EX is intended to resemble *explanatory learning*; this is exactly what theory of mind induction is aiming at.

The equality of EX and EX^{arb} is folklore in inductive inference. Therefore, arbitrary orderings are ignored whenever possible without loss of generality.

Intuitively, it seems desirable that a hypothesis reflects the information it is built upon. Formally, $\forall m \leq n (\varphi_h(x_m) = f(x_m))$ where h abbreviates $L(f_X[n])$. In the simpler case of X_0 , every x_m equals m . The property is named *consistency*. The families of function classes uniformly learnable consistently are $CONS$ and $CONS^{arb}$, resp., and $CONS^{arb} \subset CONS \subset EX$ is folklore as well. Apparently, the message is that consistency is a nontrivial property.

Consistency may be easily guaranteed, (T) if all hypotheses are in R^1 or (F) if it is decidable whether or not a hypothesis is finally correct. Adding (T) or (F) to the definition of EX and EX^{arb} , one gets learning types denoted by $TOTAL$, $TOTAL^{arb}$, FIN , and FIN^{arb} , respectively. In inductive inference, $FIN = FIN^{arb} \subset TOTAL = TOTAL^{arb} \subset CONS^{arb}$ is folklore as well [44].

Under the prior knowledge of $FIN = FIN^{arb}$, $TOTAL = TOTAL^{arb}$, and $EX = EX^{arb}$ (see [44]), all the abovementioned inclusions are on display in **Figure 4**.

NUM is the learning type defined by means of identification by enumeration as discussed in the previous section. A class $C \subset R^1$ belongs to NUM, if and only if there exists a general recursive enumeration h with $C \subseteq \{\varphi_{h(n)}\}_{n \in \mathbb{N}} \subset R^1$. A partial recursive learning device $L \in P^1$ learns via identification by enumeration on h , if and only if $L(f[n]) = h(\mu m [\varphi_{h(m)}[n] = f[n]])$. Interestingly, this extremely simple concept reflects exactly the application in [4, 5].

The potential of generalizing the learning principle of identification by enumeration is practically demonstrated in [6]. Accordingly, [73] introduces the novel concept of dynamic identification by enumeration. In terms of recursion theory, this looks as follows.

For simplicity, the authors confine themselves to X_0 . Note that we adopt a few more notations. If h is an enumeration or, alternatively, if n is an index of the enumeration h , C_h and C_n denote the class of all functions enumerated by h . From Grieser [75], we adopt the notation $[C]$ to denote all initial segments of functions in C , i.e., $[C] = \{f[n] \mid f \in C \wedge n \in \mathbb{N}\}$.

A class of functions $C \subseteq \mathbb{R}^1$ belongs to NUM^* , if and only if there exists a computable generator function $\gamma \in \mathcal{P}^1$ such that for all $f \in C$, it holds (I) for all $n \in \mathbb{N}$ that $\gamma(f[n])$ is defined, $\varphi_{\gamma(f[n])} \in \mathbb{R}^1$, $C_{\gamma(f[n])} \subseteq \mathbb{R}^1$, and $f[n] \in [C_{\gamma(f[n])}]$ and (II) there is a critical point $m \in \mathbb{N}$ such that for all $n \in \mathbb{N}$ larger than m , it holds $\gamma(f[m]) = \gamma(f[n])$ and (III) $f \in C_{\gamma(f[m])}$.

The criteria (I), (II), and (III) are practically motivated [6]. They are called *operational appropriateness*, *conversational appropriateness*, and *semantic appropriateness*, respectively. Usually, the change of $\gamma(f[n])$ to another $\gamma(f[n + 1])$ means an extension of terminology [6, 73]. The condition (II) of conversational appropriates prevents us from a Babylonian confusion.

According to [73], it holds $\text{NUM}^* = \text{TOTAL}$. This proves the enormous gain of learning power by means of *dynamic* identification by enumeration. Whereas NUM is incomparable to FIN , NUM^* is lying far above FIN ; [44] provides much more information about the space between FIN and TOTAL .

In this chapter, the authors are going much further by introducing a family $\{\text{NUM}^k\}_{k \in \mathbb{N}}$ of infinitely many refinements of NUM^* . A class C in NUM^* belongs to NUM^0 , if and only if there exists some generator function γ that is constant and identical for all functions f of C . For a positive number k , a class C in NUM^* belongs to NUM^k , if and only if there exists a γ that, for every function f of C , does generate at most k different spaces of hypotheses $\gamma(f[n])$. Intuitively, γ suggest at most k times an extension of terminology for the purpose of more appropriately expressing hypotheses throughout the process of data analysis and learning.

Jantke [76] provides a detailed discussion of benchmarks to prove that $\{\text{NUM}^k\}_{k \in \mathbb{N}}$ forms an infinite hierarchy as on display in **Figure 4**. For brevity, just two benchmarks are presented. $C^1_{q\text{-like}} = \{f \mid f \in \mathbb{R}^1 \wedge \forall x \in \mathbb{N} (x > 0 \rightarrow f(x) > 0) \wedge \varphi_{f(0)} = f\}$. Apparently, $C^1_{q\text{-like}} \in \text{NUM}^1 \setminus \text{NUM}^0$. $C^{k+1}_{q\text{-like}} = \{f \mid f \in \mathbb{R}^1 \wedge \exists g \in C^k_{q\text{-like}} \wedge \exists n \in \mathbb{N} (f(n) = 0 \wedge \forall x \in \mathbb{N} (x > n \rightarrow f(x) > 0) \wedge \forall x \in \mathbb{N} (x < n \rightarrow f(x) = g(x)) \wedge \forall x \in \mathbb{N} (x > n \rightarrow f(x) = \varphi_{f(n+1)}(x-n-1))\}$. This allows to separate NUM^{k+1} from NUM^k .

5. Process models and heuristics for mining HCI data

By the end of Section 3, the authors have summarized their HCI data mining approach and visualized essentials of inductive modeling in **Figure 3**. We take up the thread once again. The selection or the design of a terminology is essential. The terminology determines the space of hypothetical models that may be found. Throughout the process of data mining, model spaces may be subject to revision repeatedly (see preceding Section 4).

The world of models is overwhelmingly rich. Models may be characterized by properties, by purpose, by function, by model viability, or by model fitness [30]. As Thalheim puts it, “models are developed within a theory” ([30], p. 117).

Every concrete application domain provides such an underlying theory. It is a necessary precondition to data mining to specify all the aspects of the underlying theory that should be taken into account (see [30], p. 115, for mapping, truncation, distortion, and the like). Revisions

may turn out to be necessary, when inductive modeling, i.e., learning proceeds. Therefore, the word “data understanding” in the CRISP model (see **Figure 1**) is considered inappropriate and, hence, substituted by “data analysis” in the approach shown in **Figure 3**. This figure is intended to visualize both the dynamics of the data and of the model spaces. *Hypothetical* data understanding is seen as the *preliminary* result of data mining.

When speaking about logics and its algorithmic use, it is strictly advisable to stay within the limits of first-order predicate calculus [71]. The selection or the design of a logic means to decide about the signature of the language and about axiom sets of background knowledge.

Under the assumption of a given logic, business understanding and data analysis underpin an impression of what the current analysis process is about. To say it more practically, what might be typical statements arrived at by the end of the data mining process? In the authors’ digital game case study, by way of illustration, typical statements explain a human player’s action under conditions of a play state [4, 5]. In their business intelligence application [6–8], formulas relate business data and temporal information of largely varying granularity. As soon as the type of expected formulas becomes clear, the next design task is to specify an indexed family of logical formulas. This forms the first space of hypothetical models.

Within the authors’ framework, a crucial step is the modification of a space of hypotheses. There are heuristics discussed in [8] that shall be briefly surveyed. An automation may require, to some extent, natural language processing.

The human user’s activities are syntactically analyzed. In case there occur terms that have no corresponding sort, constant, function, or predicate names in the formulas of the current space of hypotheses, a limitation of the terminology is detected. The system is “unable to speak about what the user is doing.” A case discussed in [8], p. 234, is “retracement of business volume.” Retracement is interpreted as inequality with a (large) factor in it, and some sequence of such formulas of properly increasing strength is automatically generated.

Methodologies, guidelines, and process models aiming at (logical) model space construction are worth much more future research work and practical exploration.

6. Summary, conclusion, and outlook

The authors’ present approach to mining human-computer interaction data works well in applications that provide larger amounts of data [3–6]. The novel dynamic approach to the generation of model spaces exceeds the power of preceding approaches significantly [6, 73].

However successful in the cited prototypical applications, the approach may fail under conditions of small amounts of data. Consequently, it seems inappropriate to applications such as recommender systems. Perhaps, the authors’ approach would work when applied to accumulated data of larger numbers of users. If so, the particular outcome would be something like a theory of mind of a user stereotype. Related questions are still open.

Another question derives from the authors' generalization of identification by enumeration. The authors are convinced that it is possible to generalize their recent approach to dynamic identification by enumeration even further. This requires a careful easing of one or more of the requirements named operational appropriateness, conversational appropriateness, and semantic appropriateness. The related open questions need some more research effort.

Finally, the authors want to attract the audience's attention to a larger and rather involved field of research problems beyond the limits of this chapter: *reflective artificial intelligence*.

There are rarely any bug-free software systems. In the future, there will be rarely any bug-free assistant systems. However, even if a future assistant system were to be totally free of bugs, it would hardly be able to solve every imaginable problem. Digital assistant systems may fail. In response to this severe problem, it is necessary to work toward digital systems able to ponder their own abilities and limitations. Systems that do so are called reflective.

Limitations of learning systems are unavoidable [17]. In response, approaches to reflective inductive learning have been developed and investigated in much detail [75]. The results demonstrate the possibility to design and implement reflective artificial intelligence.

The authors' step from the conventional approach to dynamic identification by enumeration reveals a feature of reflection. A learning digital assistant system that gives up a certain space of hypotheses—in formal terms, $\gamma(f[n]) \neq \gamma(f[n + 1])$ resp. $\gamma(f_x[n]) \neq \gamma(f_x[n + 1])$ —with the intention to change or to extend the terminology in use is, in a certain sense, reflective. It “worries” about the limits of its current expressive power and aims at fixing the problem. Vice versa, a system able to change spaces of hypotheses, but not doing so (formally, it holds $\gamma(f[n]) = \gamma(f[n + 1])$ or $\gamma(f_x[n]) = \gamma(f_x[n + 1])$, resp.), shows a certain confidence in its abilities to solve the current problem.

This leads immediately to a variety of possibilities to implement reflective system behavior. First, a system changing its space of hypotheses may inform the human user about its recent doubts as to the limitations of terminology. Second, a bit further, it may inform the human user about details of the new terminology. Third, such a system may also report confidence.

As a side effect, so to speak, the authors' work leads to concepts and algorithmic approaches to reflective AI. This bears strong evidence of the need for further in-depth investigations.

Acknowledgements

After the second author—inspired by some fascinating results in behavioral sciences—has introduced the concept and coined the term of *theory of mind modeling and induction* in 2012, the two authors' student Bernd Schmidt has undertaken the endeavor to provide the first theory of mind modeling and induction application. The authors are grateful to him for his engaged and excellent work and for his continuous willingness to meet whatsoever requirements.

Working on an internship, Rosalie Schnappauf, then a student of the University of Rostock, took part in a series of experiments demonstrating that Bernd Schmidt's implementation of identification by enumeration does really work and allows for the fully computerized induction of a human game player's goals and intentions—a very first case of, so to speak, *mining HCI data for theory of mind induction*.

Rosalie's and Bernd's success encouraged the authors to attack harder application problems and to develop the generalized approach to *dynamic identification by enumeration*.

Part of the work reported in this chapter has been supported by the German Federal Ministry for Education and Research (BMBF) within the joint research project ODIN aiming at Open Data INovation. The authors' subprojects KVASIR (Erfurt University of Applied Sciences) and BALDUR (ADICOM Software) are currently administrated by the agency Projektträger Jülich (PtJ, see www.ptj.de) under the contracts 03PSWKPA and 03PSWKPB, respectively.



Author details

Oksana Arnold¹ and Klaus P. Jantke^{2*}

*Address all correspondence to: klaus.p.jantke@adicom-group.de

1 Erfurt University of Applied Sciences, Erfurt, Germany

2 ADICOM Software, Weimar, Germany

References

- [1] Jantke KP. User Modeling with Theories of Mind: An Introductory Game Case Study. Report KiMeRe-2012-05, Fraunhofer IDMT, Children's Media Dept., Erfurt, Germany, November 2012
- [2] Jantke KP. Theory of Mind Induction in User Modeling: An Introductory Game Case Study. Report KiMeRe-2012-06, Fraunhofer IDMT, Children's Media Dept., Erfurt, Germany, December 2012

- [3] Jantke KP. Patterns of game playing behavior as indicators of mastery. In: Ifenthaler D, Eseryel D, Ge X, editors. *Assessment in Game-Based Learning: Foundations, Innovations, and Perspectives*. New York, Heidelberg, Dordrecht, London: Springer; 2012. pp. 85-103
- [4] Schmidt B. *Theory of Mind Player Modeling [Bachelor Thesis]*. Erfurt, Germany: University of Applied Sciences; 2014
- [5] Jantke KP, Schmidt B, Schnappauf R. Next generation learner modeling by theory of mind model induction. In: *Proceedings of the 8th International Conference on Computer Supported Education (CSEDU 2016)*, 21–23 April 2016, Rome, Italy. Sétubal: SCITEPRESS, p. 499-506
- [6] Arnold O, Drefahl S, Fujima J, Jantke KP. Co-operative knowledge discovery based on meme media, natural language processing and theory of mind modeling and induction. In: *Proceedings of the International Conference on e-Society*; 20–22 April 2017; Budapest, Hungary. Sétubal: IADIS Press; 2017. pp. 27-38
- [7] Jantke KP, Fujima J. Analysis, visualization and exploration scenarios: Formal methods for systematic meta studies of big data applications. In: Grand E, Kotzinos D, Laurent D, Spyrtatos N, Tanaka Y, editors. *Information Search, Integration, and Personalization, 10th International Workshop (ISIP 2016)*, 10–12 October 2015, Grand Forks, ND, USA, Revised Selected Papers. Heidelberg, Dordrecht, London, New York: Springer; 2016. pp. 107-127
- [8] Fujima J, Arnold O, Jantke KP, Schmidt B. Interaction semantics vs. interaction syntax in data visualization and exploration. Design, implementation and utilization of meme media. In: *Proceedings of the 30th International Florida Artificial Intelligence Research Society Conference (FLAIRS-30)*, 22–24 May. Vol. 2017. Marco Island, FL, USA. Palo Alto, CA: AAAI; 2017. pp. 231-234
- [9] Popper KR. *Logik der Forschung*. Tübingen; 1934
- [10] Popper KR. *Conjectures and Refutations*. London: Routledge & Kegan Paul; 1963
- [11] FDA. FDA approves pill with sensor that digitally tracks if patients have ingested their medication [Internet]. 2017. Available from: <https://www.fda.gov/NewsEvents/Newsroom/PressAnnouncements/ucm584933.htm> [Accessed: November 22, 2017]
- [12] Jantke KP. Foreword. In Kreuzberger G, Lunzer A, Kaschek RH, editors. *Interdisciplinary Advances in Adaptive and Intelligent Assistant Systems: Concepts, Techniques, Applications, and Use*. Hershey, PA, USA: Idea Group Inc.; 2011. pp. vii-viii
- [13] Kaschek RH, editor. *Intelligent Assistant Systems: Concepts, Techniques and Technologies*. Hershey, PA, USA: Idea Group Inc.; 2007
- [14] Kreuzberger G, Lunzer A, Kaschek RH, editors. *Interdisciplinary Advances in Adaptive and Intelligent Assistant Systems: Concepts, Techniques, Applications, and Use*. Hershey, London, Melbourne, Singapore: Idea Group Inc.; 2011

- [15] Jantke KP, Igel C, Sturm R. From e-learning tools to assistants by learner modeling and adaptive behavior. In: Kaschek RH, editor. *Intelligent Assistant Systems: Concepts, Techniques and Technologies*. Hershey, PA, USA: Idea Group Inc.; 2007. pp. 212-231
- [16] Jantke KP, Müller C. Wrapper induction programs as information extraction assistants. In: Kaschek RH, editor. *Intelligent Assistant Systems: Concepts, Techniques and Technologies*. Hershey, PA, USA: Idea Group Inc.; 2007. pp. 35-63
- [17] Jain S, Osherson DN, Royer JS, Sharma A. *Systems That Learn: An Introduction to Learning Theory*. Cambridge, MA, USA: MIT Press; 1999
- [18] Carberry S, Weibelzahl S, Micarelli A, Semeraro G. User Modeling Adaptation, and Personalization, Proceedings of the 21st International Conference (UMAP 2013), 10–14 June 2016, Rome, Italy, Ser. LNCS, Vol. 7899. Heidelberg, Dordrecht, London, New York: Springer; 2013
- [19] Dimitrova V, Kuflik T, Chin D, Ricci F, Dolog P, Houben GJ. User Modeling Adaptation, and Personalization, Proceedings of the 22nd International Conference (UMAP 2014), 7–11 July 2016, Aalborg, Denmark, Ser. LNCS, Vol. 8538. Heidelberg, Dordrecht, London, New York: Springer; 2014
- [20] Ricci F, Bontcheva K, Conlan O, Lawless S. Editors. User Modeling Adaptation, and Personalization, Proceedings of the 23rd International Conference (UMAP 2015), 29 June - 3 July 2016, Dublin, Ireland, Ser. LNCS, Vol. 9146. Heidelberg, Dordrecht, London, New York: Springer; 2015
- [21] Vassileva J, Blustein J, Aroyo L, D’Mello S, editors. User Modeling Adaptation, and Personalization, Proceedings of the 24th International Conference (UMAP 2016), 13–17 July 2016, Halifax, NS, Canada. New York, NY, USA: ACM; 2016
- [22] Bielikova M, editor. User Modeling Adaptation, and Personalization, Proceedings of the 25th International Conference (UMAP 2017), 9–12 July 2017, Bratislava, Slovakia. New York, NY, USA: ACM; 2017
- [23] Lengyel D. A Bottom-Up Approach to Reveal a Mapping between Personality Traits and Gameplaying Traces [Diploma Thesis]. Ilmenau, Germany: Ilmenau Univ. of Technology; 2012
- [24] Leary TF. *Interpersonal Diagnosis of Personality: A Functional Theory and Methodology for Personality Evaluation*. New York, NY, USA: Ronald Press; 1957
- [25] Veluswamy R. Clinical quality data mining in acute care. *The Physician Executive*. 2008; 48-53
- [26] Zhang D, Zhou L. Discovering golden nuggets: Data mining in financial application. *IEEE Transactions on Systems, Man, and Cybernetics Part C: Applications and Reviews*. 2004; 34:513-522
- [27] Jankun-Kelly T, Ma KL, Gertz M. A model and framework for visualization exploration. *IEEE Transactions on Visualization and Computer Graphics*. 2007;13:357-369

- [28] Bridewell W, Langley P, Todorovski L, Džeroski S. Inductive process modeling. *Machine Learning*. 2008;**71**:1-32
- [29] Overmars, KP, de Groot, WT, Huigen, MGA. Comparing inductive and deductive modeling of land use decisions: Principles, a model and an illustration from the Philippines. *Human Ecology* 2007;**32**:439-452
- [30] Thalheim B. The conception of the model. In: Abramowicz W, editor. *International Conference on Business Information Systems (BIS 2013)*. Ser. LNBIP. Vol. 157. Berlin, Heidelberg: Springer; 2013. pp. 113-124
- [31] Jannaschk K. *Infrastruktur für ein Data Mining Design Framework* [Ph.D. thesis]. Kiel, Germany: Christian-Albrechts-Univ.; 2017
- [32] Fayyad U, Piatetsky-Shapiro G, Smyth P. The KDD process for extracting useful knowledge from volumes of data. *Communication ACM*. 1996;**39**:27-34
- [33] Wirth R, Hipp J. CRISP-DM. Towards a standard process model for data mining. In: *Proceedings of the 4th International Conference on the Practical Applications of Knowledge Discovery and Data Mining*. 2000. pp. 29-39
- [34] Carruthers P, Smith PK. *Theories of Theories of Mind*. Cambridge, UK: Cambridge University Press; 1996
- [35] Emery NJ, Clayton NS. Comparative social cognition. *Annual Review of Psychology*. 2009;**60**:87-113
- [36] Emery NJ, Dally JM, Clayton NS. Western scrub-jays (*Aphelocoma californica*) use cognitive strategies to protect their caches from thieving conspecifics. *Animal Cognition*. 2007;**7**: 37-43
- [37] Jantke, KP. PCP-Spiele. Report KiMeRe-2012-04, Version 2.0, Fraunhofer IDMT, Children's Media Dept., Erfurt, Germany, February 2016
- [38] Alexander C. *A Pattern Language*. New York, NY, USA: Oxford University Press; 1977
- [39] Alexander C. *The Timeless Way of Building*. New York, NY, USA: Oxford University Press; 1979
- [40] Angluin D. Finding patterns common to a set of strings. *J. Computer and System Sciences*. 1980;**21**:46-62
- [41] Rogers H. *Theory of Recursive Functions and Effective Computability*. New York, NY, USA: McGraw-Hill; 1967
- [42] Garey MR, Johnson DS. *Computers and Intractability: A Guide to the Theory of NP-Completeness*. San Francisco, CA, USA: W. H. Freeman & Co.; 1979
- [43] Angluin D, Smith CH. Inductive inference: Theory and methods. *ACM Computing Surveys*. 1983;**15**:237-269

- [44] Jantke KP, Beick HR. Combining postulates of naturalness in inductive inference. *EIK*. 1981;17:465-484
- [45] Yang Q, Wu X. 10 challenging problems in data mining research. *International Journal of Information Technology & Decision Making*. 2006;5:597-604. DOI: 10.1142/S0219622006002258
- [46] Roddick JF, Spiliopoulou M. A bibliography of temporal, spatial and spatio-temporal data mining research. *ACM SIGKDD Explorations*. 1999;1:34-38
- [47] Prior AN. *Time and Modality*. Oxford, UK: Clarendon Press; 1957
- [48] Prior AN. *Past, Present and Future*. Oxford, UK: Clarendon Press; 1967
- [49] Blackburne, P; De Rijke, M, Venema, Y. *Modal Logic*. Ser. Cambridge Texts in Theoretical Computer Science, Vol. 53. Cambridge, UK: Cambridge University Press, 2011
- [50] van Benthem, JFAK. *Logical Dynamics of Information and Interaction*. Cambridge, MA, USA: Cambridge University Press, 2010
- [51] Gabbay DM, Hodkinson, I, Reynolds, M. *Temporal Logic: Mathematical Foundations and Computational Aspects*. Ser. Oxford Logic Guides, Vol. 1. Oxford, UK: Clarendon Press; 1994
- [52] van Benthem, JFAK, Smets, S. Dynamic logics of belief change. In: van Ditmarsch, H, Halpern, JY, van der Hoek, W, Looi, B, editors. *Handbook of Logics for Knowledge and Belief*. London, UK: College Publications, 2015. p. 299-368
- [53] Fisher M. Temporal representation and reasoning. In: van Harmelen F, Lifschitz V, Porter B, editors. *Handbook of Knowledge Representation*. Amsterdam, Oxford: Elsevier; 2008. pp. 513-550
- [54] Mueller ET. Event calculus. In: van Harmelen F, Lifschitz V, Porter B, editors. *Handbook of Knowledge Representation*. Amsterdam, Oxford: Elsevier; 2008. pp. 671-708
- [55] Doherty, P, Kvarnström, J. Temporal action logics. In: van Harmelen, F, Lifschitz, V, Porter, B, editors. *Handbook of Knowledge Representation*. Amsterdam, Oxford: Elsevier, 2008. p. 709-757
- [56] Peppas P. Belief revision. In: van Harmelen F, Lifschitz V, Porter B, editors. *Handbook of Knowledge Representation*. Amsterdam, Oxford: Elsevier; 2008. pp. 317-359
- [57] Lenerz C. Layered languages of Ludology – Eine Fallstudie. In: Beyer A, Kreuzberger G, editors. *Digitale Spiele – Herausforderung Und Chance*. Ser. Game Studies. Boizenburg, Germany: Whv; 2009. pp. 39-52
- [58] Schedel T, Atzenbeck C. Spatio-temporal parsing in spatial hypermedia. In: *Proceedings of the 27th ACM Conference on Hypertext and Social Media (HT 16)*, 10–13 July 2016, Halifax, Canada. New York, NY, USA: ACM Press. pp. 149-157

- [59] Hopcroft JD, Ullman JE. Introduction to Automata Theory, Languages, and Computation. Menlo Park, London, Amsterdam, Ontario: Reading; 1979
- [60] Heyting A. Die formalen Regeln der intuitionistischen Logik. Sitzungsberichte der Preussischen Akademie der Wissenschaften, Physikalisch-mathematische Klasse. 1930: 158-169
- [61] Kolmogorov AN. Zur Deutung der intuitionistischen Logik. Mathematische Zeitschrift. 1932;**35**:58-65
- [62] Jantke KP. PCP-Spiele. Technical Report KiMeRe-2012-04. Erfurt, Germany, Fraunhofer IDMT, Children's Media Dept.; 2012
- [63] Machtey M, Young P. An Introduction to the General Theory of Algorithms. Elsevier North-Holland: New York, NY; 1978
- [64] Jantke KP. Theory of mind modeling and induction: Ausdrucksfähigkeit und Reichweite. ADISY Technical Report 03/2016. Weimar, Germany: ADISY Consulting GmbH; 2016
- [65] Davis M, editor. The Undecidable: Basic Papers on Undecidable Propositions, Unsolvability Problems and Computable Functions. New York, NY; 1965
- [66] Gold EM. Language identification in the limit. Information and Control. 1967;**17**:447-474
- [67] Jantke KP. The main proof-theoretic problems in inductive inference. In: Wechsung G, editor. Frege Conference, 10–14 September 1984, Schwerin, Germany. Berlin, Germany: Akademie-Verlag, pp. 321-330
- [68] Jantke KP, Arnold O. Patterns – The key to game amusement studies. In: Proceedings of the 3rd Global Conference on Consumer Electronics (GCCE 2014), 7–10 October 2014, Makuhari Messe, Tokyo, Japan. IEEE Consumer Electronics Society 2014, pp. 478-482
- [69] Jantke KP. The pattern experience evaluation program. In: Proceedings of the Intl. Workshop on Design, Evaluation and Refinement of Intelligent Systems (DERIS 2009), 28 November. Vol. 2009. Kraków, Poland: AGH Univ. of Science and Technology; 2009. pp. 70-75
- [70] Angluin D. Inductive inference of formal languages from positive data. Information and Control. 1980;**45**:117-135
- [71] Lindström P. On extensions of elementary logic. Theoria. 1969;**35**:1-11
- [72] Clocksin WF, Mellish CS. Programming in Prolog. Berlin, Heidelberg, New York: Springer; 1981
- [73] Arnold O, Drefahl S, Fujima J, Jantke KP, Vogler C. Dynamic identification by enumeration for co-operative knowledge discovery. IADIS International Journal on Computer Science and Information Systems. 2017;**12**:65-85

- [74] Gödel K. Über formal unentscheidbare Sätze der "Principia Mathematica" und verwandter Systeme. Monatshefte für Mathematik und Physik. 1931;**38**:173-198
- [75] Grieser G. Reflective inductive inference of recursive functions. Theoretical Computer Science. 2008;**397**:57-69
- [76] Jantke KP, Drefahl S, Arnold O. Characterizing the power and the limitations of concepts for adaptivity and personalization by benchmark results from inductive inference. In 12th Intl. Workshop on information search, integration, and personalization (ISIP 2018), 14-15 May 2018, Kyushu University, Fukuoka, Japan, 2018