The Adaptive Tutoring System RATH
a Prototype

Cord Hockemeyer and Dietrich Albert

Department of Psychology
University of Graz, Austria

The theory of knowledge spaces developed by Doignon and Falmagne provides a theoretical framework for adaptive tutoring systems (ATS). RATH is a domain–independent web–based ATS built on a connection between knowledge space theory, a relational hypertext model, and relational database technology.

The current prototype includes a course on elementary probability theory based on prior analyses.

Keywords: Adaptive Tutoring, Hypermedia, Relational Modeling, Knowledge Space Theory

Adaptive tutoring systems (ATS) adapt to the student's knowledge in the domain taught. This means, that lessons, examples, or test and training problems are presented to students only if they know all prerequisites for understanding the material. A theoretical framework for such individual selection of material exists in the theory of knowledge spaces (Doignon and Falmagne, 1985, 1998; Albert, 1994; Albert and Lukas, 1999). So far, there exist two tutoring systems based on knowledge space theory. ALEKS[1] (Doignon and Falmagne, 1998) is a web–based system applying knowledge space theory developed especially for K12 mathematics. AdAsTra (Dowling et al., 1996) is a general purpose system for testing and training which, however, does not offer web access and does not include a possibility for teaching lessons. The RATH [2] system (Hockemeyer, 1997; Hockemeyer et al., 1998) was developed in order to obtain a web–based, domain–independent ATS built on knowlede space theory.

[1] See `http://www.aleks.com`.
[2] See `http://wundt.kfunigraz.ac.at/rath/`.

1 Knowledge space theory

If Q is a set of items, the *knowledge state* of a student can be described as the subset of items, this student masters. Due to prerequisite relationships between items, the set of possible knowledge states, the *knowledge space*, is restricted to a subset of the power set of Q. One way to represent such prerequisite relationships is the *surmise relation*. Two items $x, y \in Q$ are in prerequisite relation ($x \sqsubseteq y$) if, from a correct answer to item y, we can surmise a correct answer to item x. Each surmise relation describes a unique knowledge space.

Surmise relations are assumed partial orders on Q. Therefore, they can be illustrated through Hasse diagrams. Figure 1 shows such Hasse diagrams of a surmise relation and the corresponding knowledge space. Knowledge spaces corresponding to a surmise relation are

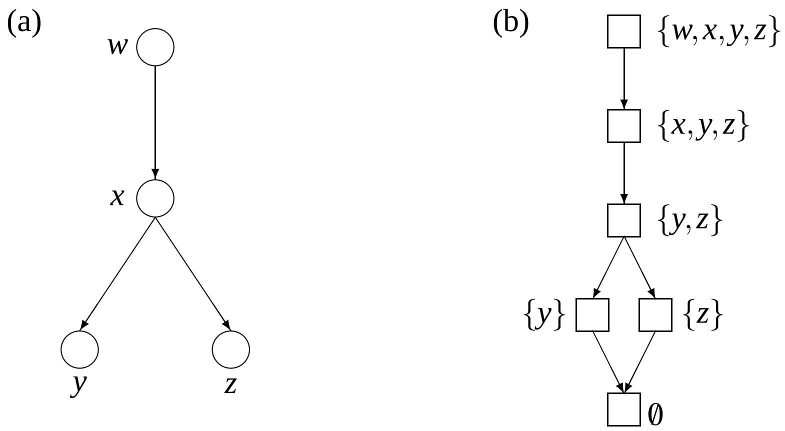

Figure 1: Surmise relation (a) and knowledge states (b) for the problems in Q. (Items are marked by circles and states are marked by squares)

closed under union and intersection, i. e. for any two knowledge states S, S', their union $(S \cup S')$ and their intersection $(S \cap S')$ are also knowledge states.

Applying knowledge space theory for tutoring, we obtain the concept of *learning paths* (Falmagne, 1989, 1993) describing students' possible ways from the complete novice (knowledge state $\emptyset$) through the knowledge space to the complete expert (knowledge state Q). We will come back to this approach in Section 3.1.

A more general concept for representing prerequisite relationships has been specified through *surmise systems* (Doignon and Falmagne, 1985). Graphically, they can be illustrated through and–or–graphs. Knowledge spaces corresponding to surmise systems are closed under union but not necessarily under intersection. This approach has not been applied in RATH, yet. The ALEKS and AdAsTra systems mentioned above apply surmise systems instead of surmise relations.

2 Hypertext theory

Based on hypertext models suggested by Tompa (1989) and by Halasz and Schwartz (1990, 1994), Albert and Hockemeyer (1997) have used a relational notation for formally describing hypertext structures. Within this model, a *hypertext* $H = (C, L)$ consists of a set C of *components* and a set L of *links*. A component $c = (b, S_c, D_c)$ is built by a *base component* b (i. e. the basic unit of information), a set S_c of *source anchors*, and a set D_c of *destination anchors*. The source anchors $s_c \in S_c$ and the destination anchors $d_c \in D_c$ are located on the base component. The sets of all source anchors and destination anchors are denoted by S and D, respectively. A *link* $l \in L$ with $l = ((c, s_c), (c', d_{c'}))$ connects the source anchor s_c located on the component c with the destination anchor $d_{c'}$ located on the component c'. We also call l a link from c to c'. Based on this formalization of a hypertext, we introduce a binary relation $\vdash \subseteq C \times C$ for a hypertext $H = (C, L)$ by $c \vdash c'$ if and only if there exists a link $l = ((c, s_c), (c', d_{c'})) \in L$.

By distinguishing a subset $P \subset S$ of source anchors we also obtain a distinct subset $L^P \subset L$ by defining $L^P = \{l = ((c, s_c), (c', d_{c'})) \mid s_c \in P\}$. Consequently, we can define a binary *prerequisite link relation* $\vdash^P$ on the set C of components by applying the definition of $\vdash$ to the reduced set L^P of Links. This mechanism can be used to define different *link types*. One example for such link types is the *prerequisite link* in a hypertext tutoring system. Such a prerequisite link from c to c' would specify that, for understanding the content of c, a student must know the content of c'. Figure 2 illustrates prerequisite links within a set $C = \{u, v, w, x\}$ of four components.

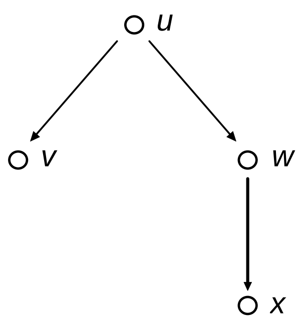

Figure 2: Prerequisite links in a set of components

3 The RATH system

3.1 Connecting knowledge space and hypertext theory

Based on a prerequisite link relation as described above, an adaptive hypertext tutoring system should restrict the student's navigational possibilities to those components they can understand with their existing knowledge. This restriction of navigation can be realized in the framework of knowledge space theory as presented in Section 1.

If we assume the student's knowledge to comprise the components presented to the student so far, we only have to regard the subsets of components they have visited and/or mastered (cf. Albert and Hockemeyer, 1997).

Because of the restriction of the student's navigation only some subsets of components can be reached in this way. Figure 3 shows those subsets of components which may have been presented to a student if we assume the prerequisite link structure from Fig. 2.

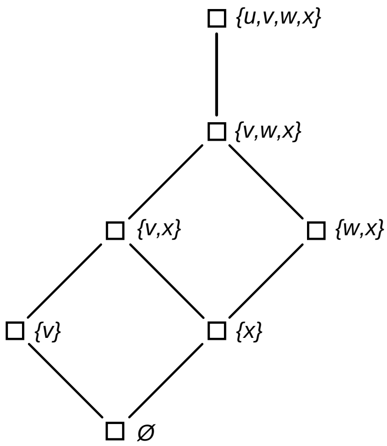

Figure 3: Structure according to the prerequisite relation above

Taking into account this structure we obtain that there is only a limited number of ways to get from the empty set (or another starting point) to the set of all components (or another learning goal. For a Hypertext $H = (C, L)$ and a prerequisite link relation $\vdash^P$, a sequence $X_0, X_1, \ldots, X_n$ of subsets of components is called a *learning path* if and only if the following two conditions hold: (i) For any $i = 0, 1, \ldots, n$, for any $c \in X_i$, and for any $c' \in C$, if $c \vdash^P c'$ then $c' \in X_i$. (ii) For any $i = 1, 2, \ldots, n$ there exists a $c \in C \setminus X_{i-1}$ such that $X_u = X_{i-1} \cup \{c\}$. Figure 4 shows one possible learning path from the empty set to the complete set in the structure from Figure 3. In this example there exist three possible paths. This notion of a learning path

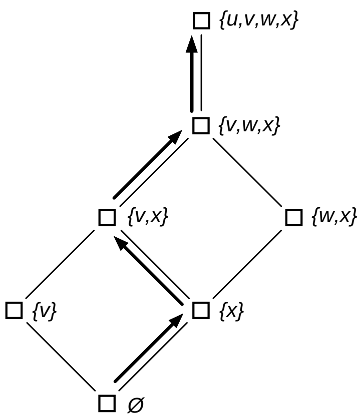

Figure 4: One possible path through the structure from Figure 3

corresponds to that from knowledge space theory mentioned in Section 1 above.

3.2 Technical aspects of RATH

Applying relational models for the structure of the domain of knowledge and of the hypertext yields an important technical advantage: the relational notation allows a rather direct transfer of the mathematical models to a relational database system; only a few concepts have to be represented differently due to normalization issues. The relational database then can also maintain data about the students' performance and progress. Thus, efficient procedures for accessing, evaluating, and administering large data sets can be used which are available as part of professional database systems.

In Figure 5, an application oriented view of the architecture of RATH is depicted. We look at the architecture with special respect to adaptive tutoring systems (ATS). We start with

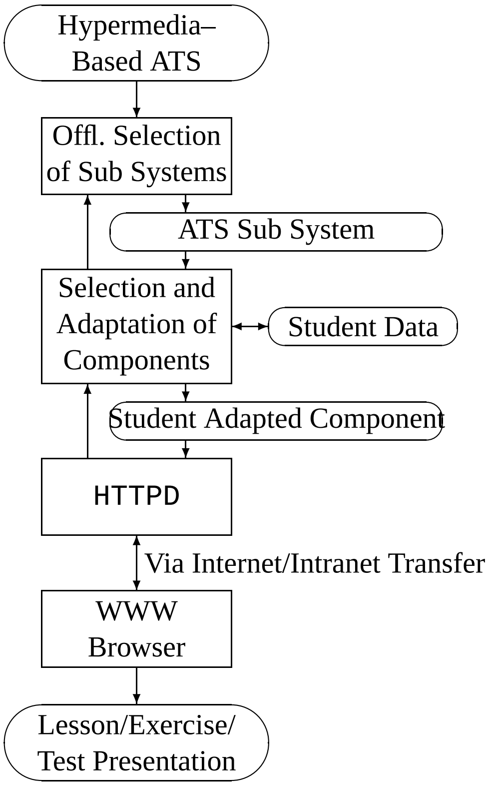

Figure 5: RATH architecture

a hypermedia–based adaptive tutoring system which corresponds to the complete hypertext. ATS sub systems can be obtained through offline selection functions. Offline here means that these functions are applied once, e. g. for tailoring specific courses before they are made available to students.

From this ATS sub system, certain components, e. g. lessons or exercises, are selected and adapted online, i. e. during the learning process, according to data stored about the student's knowledge and preferences. This online selection and adaptation of single documents is done by CGI programs. The adapted component is sent via the HTTPD to the student's WWW browser which presents the lesson, exercise, or test to the student. In case of exercises or tests, each answer given by the student is transferred by the WWW browser via the HTTPD back to the CGI programs which evaluate the answer and update the student database accordingly. For an intermediate student who has visited a number of lessons and may visit all but two lessons (drawing without replacement and generalized descriptions of events) and the final certification page, Fig. 6 shows the content overview of the RATH example course on elementary probability theory.

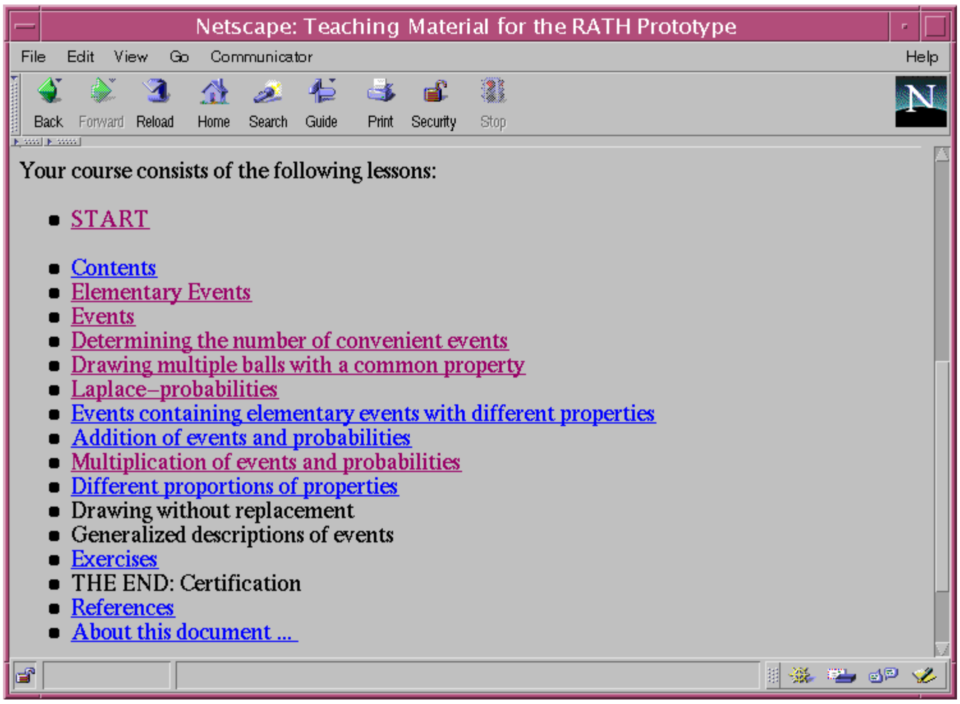

Figure 6: RATH contents overview for an intermediate student

In the current state of RATH, several features are still missing. One example is the introductory assessment of knowledge. Currently, it is assumed that new students start with zero knowledge. There exist, however, several procedures applying knowledge spaces for the assessment of knowledge (see, e. g., Doignon and Falmagne, 1998; Dowling and Hockemeyer, to appear). Another example is the offline selection of sub systems mentioned above.

4 Structuring contents

An important task in the development of ATS is the determination of structures for lessons and problems serving as a basis for the adaptivity. In our case, these structures are the surmise relations and their corresponding knowledge spaces. During the last fifteen years, numerous procedures have been suggested which involve mass data analysis, querying experts with computer support, and analysis of competencies and performances, or of cognitive processes, and the cognitive demand analysis and systematic construction of test items (for an overview, see Lukas and Albert, 1999). In the course used in the RATH prototype, the latter method was applied. We describe the method using the course on elementary probability theory as an example.

The basis of the analysis is a set of problems spanning the domain of knowledge, in our case a set of typical problems on drawing balls from an urn. For each of the problems, the *demands* posed upon students by the problem must be identified. These demands point to abilities or entities of knowledge for which corresponding lessons (including examples etc.) must be developed. The ten demands for urn models in elementary probability theory listed in Table 1 have been identified by Held (1999). In our RATH course, each of these demands

Table 1: Demands in the field of elementary probability theory

1. Knowledge that, in general, Laplace probabilities are computed as the ratio between the number of *convenient* events and the number of *possible* events.
   [Laplace–probabilities]
2. Ability to determine the number of possible events.
   [Events]
3. Ability to determine the number of convenient events if one ball is drawn.
   [Determining the number of convenient events]
4. Ability to determine a convenient event if one ball is drawn, or if the sample for which the probability has to be computed consists of equally colored balls.
   [Drawing multiple balls with a common property]
5. Knowledge that if an outcome like "exactly/at least n balls are of colour x" is asked for, all possible sequences of drawings are convenient events.
   [Events containing elementary events with different properties]
6. Knowledge that $p(A \cup B) = p(A) + p(B)$, for two disjoint events A and B.
   [Addition of events and probabilities]
7. Knowledge that $p(A \cap B) = p(A) \cdot p(B)$, for two events A and B that are (stochastically) independent.
   [Multiplication of events and probabilities]
8. Knowledge that the probability of drawing a ball of a specific colour is not equal to 0.5 if there are different numbers of balls of different colors in the urn.
   [Different proportions of properties]
9. Knowledge that drawing *without* replacement reduces both, the total numbers of balls in the urn as well as the numbers of balls that have the same number as the drawn ball.
   [Drawing without replacement]
10. Knowledge that drawing *at least* a number of certain balls includes the — not explicitly stated — results of drawing more balls of the certain kind.
    [Generalized description of events]

is represented by a lesson. The titles of the corresponding lessons are specified in brackets in Table 1.

On the other hand, problems are described through *components*, i. e. properties, which may take certain values, called *attributes*. For the urn models, Held specified three components[3],

[3] A fourth component, "wording" is not relevant for our work.

(a) the numerical ratio of differently colored balls, (a) the way of drawing, and (a) the specification of the asked event. For the component a, e. g., there exist the attributes a_1 "equal to one" and a_2 "not equal to one".

By assigning to each of these attributes those subsets of demands which are required for solving problems with the respective attribute, Held (1999, p. 91) determined component–wise attribute orders based on a set inclusion principle (Birkhoff, 1967).

For obtaining a structure between the demands (and, thus, between the lessons), the inverse assignment is regarded that maps each demand onto the set of attributes for which the respective demand is required. Analogously to the attribute orders, a demand structure can be obtained (see Hockemeyer, 1997, App. D). This demand structure is shown in Fig. 7.

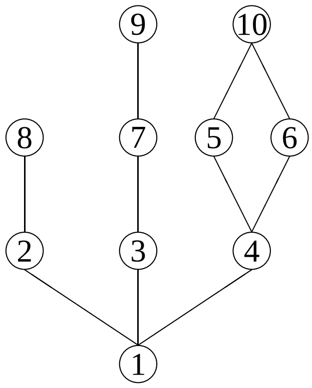

Figure 7: Demand structure

Finally, the problems used for the demand identification in the first phase are also analyzed in order to obtain a problem structure that can be merged with the demand structure. Based on the attribute orders, Held (1999) has ordered the problems applying the principle of component–wise ordering of product sets. Such a problem structure can be embedded into a demand structure (Hockemeyer, 1997). The result is shown in Fig. 8 as a common structure of demands (i. e. lessons) and problems. The lower part of this structure has been slightly changed and extended compared to the original result from Fig. 7: The demand "Laplace–probabilities" has been moved for didactical reasons, and a new demand and lesson "Elementary events" has been inserted providing some very elementary knowledge.

5 Conclusions

The implementation of the RATH system has proven that the theoretical concepts introduced in Sections 2 and 3.1 can be put into practice. Nevertheless, there are still a number of improvements necessary in order to get a system ready for application. These improvements include practical issues (e. g. user interface or administrative facilities) as well as theory–based ones (e. g. an introductory knowledge assessment, selecting sub ATS systems online, using surmise systems as a more general representation for prerequisite relationships, or comparative investigation of different methods for navigation support).

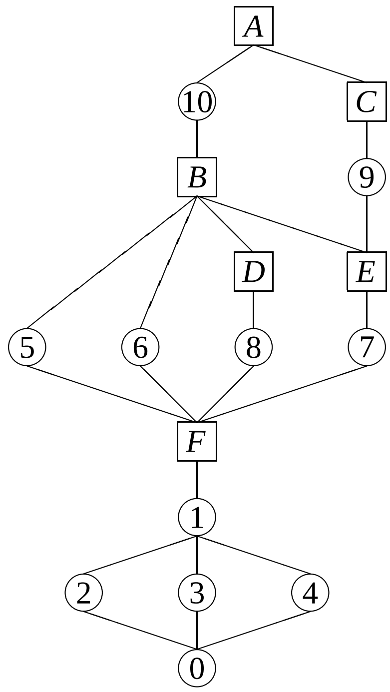

Figure 8: Modified structure of teaching contents (circles) and training problems (boxes)

Another issue for further research is the technical evaluation of the system. First tests have shown that the system — hosted on a SPARC-20 serving a SUN cluster — is capable of serving a class synchronously. However, using it with larger number of students in parallel has not yet been tested.

A psychological and pedagogical evaluation would be concerned with the effects of a system like RATH on the students' learning progress. In order to test these effects, a larger course would be needed containing more learning contents. Such a course should also have a different topic to demonstrate that RATH is independent from the domain of knowledge.

Acknowledgments The work reported in this paper was supported by the European Commission through the HCM grant ERBCHBICT941599 to Theo Held and Cord Hockemeyer.

References

Albert, D. (Ed.). (1994). *Knowledge structures*. New York: Springer Verlag.

Albert, D. & Hockemeyer, C. (1997). Dynamic and adaptive hypertext tutoring systems based on knowledge space theory. In B. du Boulay & R. Mizoguchi (Eds.), *Artificial Intelligence in Education: Knowledge and Media in Learning Systems* (pp. 553–555). Amsterdam: IOS Press.

Albert, D. & Lukas, J. (Eds.). (1999). *Knowledge spaces: Theories, empirical research, applications*. Mahwah, NJ: Lawrence Erlbaum Associates.

Birkhoff, G. (1967). *Lattice theory* (third ed.). Number XXV in American Mathematical Society Colloquium Publication. American Mathematical Society.

Doignon, J.-P. & Falmagne, J.-C. (1985). Spaces for the assessment of knowledge. *International Journal of Man-Machine Studies*, *23*, 175–196.

Doignon, J.-P. & Falmagne, J.-C. (1998). *Knowledge spaces*. Berlin: Springer–Verlag.

Dowling, C. E. & Hockemeyer, C. (to appear). Automata for the assessment of knowledge. *IEEE Transactions on Knowledge and Data Engineering*.

Dowling, C. E., Hockemeyer, C., & Ludwig, A. H. (1996). Adaptive assessment and training using the neighbourhood of knowledge states. In C. Frasson, G. Gauthier, & A. Lesgold (Eds.), *Intelligent Tutoring Systems* (pp. 578–586). Berlin: Springer Verlag.

Falmagne, J.-C. (1989). A latent trait theory via stochastic learning theory for a knowledge space. *Psychometrika*, *53*, 283–303.

Falmagne, J.-C. (1993). Stochastic learning paths in a knowledge structure. *Journal of Mathematical Psychology*, *37*, 489–512.

Halasz, F. & Schwartz, M. (1990). The Dexter hypertext reference model. In J. Moline, D. Benigni, & J. Baronas (Eds.), *Proceedings of the Hypertext Standardization Workshop* (pp. 95–133). Gaithersburg, MD 20899: National Institute of Standards and Technology.

Halasz, F. & Schwartz, M. (1994). The Dexter hypertext reference model. *Communications of the ACM*, *37*(2), 30–39.

Held, T. (1999). An integrated approach for constructing, coding, and structuring a body of word problems. In D. Albert & J. Lukas (Eds.), *Knowledge Spaces: Theories, Empirical Research, Applications* (pp. 67–102). Mahwah, NJ: Lawrence Erlbaum Associates.

Hockemeyer, C. (1997). *RATH — a relational adaptive tutoring hypertext WWW–environment*. (Tech. Rep. 1997/3), Institut für Psychologie, Karl–Franzens–Universität Graz, Austria. To be obtained from Cord.Hockemeyer@kfunigraz.ac.at.

Hockemeyer, C., Held, T., & Albert, D. (1998). RATH — a relational adaptive tutoring hypertext WWW–environment based on knowledge space theory. In C. Alvegård (Ed.), *CALISCE'98: Proceedings of the Fourth International Conference on Computer Aided Learning in Science and Engineering* (pp. 417–423). Göteborg, Sweden: Chalmers University of Technology.

Lukas, J. & Albert, D. (1999). Knowledge structures: What they are and how they can be used in cognitive psychology, test theory, and the design of learning environments. In D. Albert & J. Lukas (Eds.), *Knowledge Spaces: Theories, Empirical Research, Applications* (pp. 3–12). Mahwah, NJ: Lawrence Erlbaum Associates.

Tompa, F. W. M. (1989). A data model for flexible hypertext database systems. *ACM Transactions on Information Systems*, *7*(1), 85–100.