Empirical Evidence for the Use of

Derivational Strategies in Analogical Problem Solving

Bachelor Thesis in Cognitive Science

December 2oo3

Sven-Eric M. Schelhorn (sven-eric.schelhorn@uos.de)

Department of Cognitive Science

University of Osnabrueck

Abstract

This paper presents evidence for the usage of derivational strategies in analogical problem solving. In cases of high structural similarity but low superficial similarity between two problem domains, subjects tend to use other strategies than transformational mapping to derive a new solution from a previously learned one. Methods used in the experiment include time-reaction measure and qualitative assessments. In conclusion, this paper suggests exploring strategies of analogical problem solving which include not only transformational approaches, but also derivational ones.

Contents

Introduction

How do individuals perform seemingly effortless comparisons between superficially very different objects and concepts? In how far do solutions to old problems help us solve new problems? Is it possible to liken abstract or unfamiliar concepts to something we know? The answers to these questions all have to involve the concept of analogy. But what is an analogy, after all? Consider the following two examples:

Fortress Problem: A small country was controlled by a dictator. The dictator ruled the country from a strong fortress. The fortress was situated in the middle of the country, surrounded by farms and villages. Many roads radiated outward from the fortress like spokes on a wheel. A general arose who raised a large army and vowed to capture the fortress and free the country of the dictator. The general knew that if his entire army could attack the fortress at once it could be captured. The general's troops were gathered at the head of one of the roads leading to the fortress, ready to attack. However, a spy brought the general a disturbing report. The ruthless dictator had planted mines on each of the roads. The mines were set so that small bodies of men could pass over them safely because the dictator needed to be able to move troops and workers to and from the fortress. However, any large force would detonate the mines. Not only would this blow up the road and render it impossible, but the dictator would then destroy many villages in retaliation. It therefore seemed impossible to mount a full-scale direct attack on the fortress. (Duncker, 1945)

Tumour Problem: Suppose you are a doctor faced with a patient who has a malignant tumour in his stomach. It is impossible to operate on the patient, but unless the tumour is destroyed the patient will die. There is a kind of ray that can be used to destroy the tumour. If the rays reach the tumour all at once with sufficiently high intensity, the tumour will be destroyed. Unfortunately, at this intensity the healthy tissue that the rays pass through will also be destroyed. At lower intensities the rays are harmless to the healthy tissue, but they will not affect the tumour either. What type of procedure might be used to destroy the tumour with rays without destroying the healthy tissue? (Duncker, 1945)

Although these problems look very different, their solutions are actually quite similar and indeed *analogous*, as they are based on the same underlying abstract strategy: *Divide and Conquer*.

The general could divide his army up into small groups and dispatch each group to the head of a different road. When all is ready he gives the signal, and each group marches down a different road. Each group continues down its road to the fortress, so that the entire army finally arrives together at the fortress at the same time. In this way, the general is able to capture the fortress and thus overthrow the dictator.

In the other problem, the ray can be divided into several low-intensity rays, no one of which will destroy the healthy tissue. If these several rays are positioned at different locations around the body and focused on the tumour, their effect will combine, thus being strong enough to destroy the tumour.

To find the correct solution concept, Divide and Conquer, with only one of these stories is surprisingly hard, but much easier when both stories are available (Gick & Holyoak, 1980). Why is that so? In order to solve the problems, the reader first has to recall the cover stories Fortress and Tumour. In our example, this *Retrieval* is quite simple because the memory trace is still fresh and only two possible candidates are presented. The next obvious step is the search for *similarities* between the stories. When talking about similarities in the context of analogical problem solving, semantically similar *objects* (generals and doctors, being vulnerable etc.) and semantically similar *relations* (like ray causes tissue to be destroyed, mine causes people to be killed, destruction of tumour is desired by doctor, destruction of fortress is desired by general etc.) are distinguished. The first set is roughly labelled *superficial similarity*, the latter set is labelled *structural similarity.*

Why this is so becomes clear at once when proceeding to the next step: the *Mapping.* During this process, the reader mentally connects similar entities (objects as well as relations) from one problem or domain to the other domain to find the common frame of reference that includes both stories and may help with the solution. In our example, it is very hard to find helpful superficial similarities between objects like armies, rays, tumours, fortresses, streets and tissue: by comparing their *attributes*, they are all very different objects. Only the general and the doctor are similar in the way that they are both persons who act to reach a certain goal.

Similarities between objects are labelled *superficially*, because they normally do not offer much information about the abstract role of objects within a problem and often construct useless mappings regarding the problem solution if unimportant attributes are mapped.

On the other side, we find quite a lot *structural similarities*, higher-order relations like x uses y to destroy z and so forth.

Similarities between relations are labelled *structural* because their mapping builds up complex structures of interconnected higher-order relations. They are often linked by *causation.* In our example, many subsequent steps of mapping may conclude to the concept of *Division of Forces* which forms the solution to both problems. This concept now may be *evaluated* for correctness by the reader and then learned in an *abstracted* form like "what I can't reach by a focused force I can possibly reach by intelligently splitting it up".

All this subsequent steps which allowed us to construct the correct solution concept may be labelled as one complex process: *Analogy-Making.*

While the set of mental processes roughly described in our example is a form concentrated on relational mapping and thus were an example for *Transformational*

Analogy Making, another way of action may have occurred if the solution trace to one of the two problems had already been available. In that case, the solution of the unsolved problem only had to be derived from the solution of the solved problem - this is termed *Derivational Analogy Making*. This thesis is aimed to summarize the field of research concerned with analogy-making and present evidence for the usage of exactly this special kind of the latter, namely *Derivational Analogy Making* used in analogical problem solving.

During the course of this paper we will explore under which conditions the one form of analogical problem solving or the other occurs. This is done by investigating how human subjects draw analogies to solve a set of *structurally isomorphic graph problems* in varying *degrees of surface similarity* with previously learned problems. The findings challenge the traditional transformational approach used in nearly all current models and propose that humans possess a set of strategies for analogical problem solving to select from.

These findings are in so far interesting for the cognitive sciences, as analogical reasoning and problem solving are core cognitive abilities. Evidence for the drawing of analogies is provided not only in problem solving and learning, but also in perception, creativity, communication and scientific discovery. Assuming the ability to draw and process analogies is not only a basic property of cognitive systems but actually the basis and foundation of reasoning itself may offer a new set of promising and consolidating theories for the cognitive sciences.

Examples for analogies range from mundane, low level recognition and categorization to more sophisticated and abstract mental models. Analogy making in humans seems to appear both spontaneously and deliberately in a wide spectrum of situations. This spectrum includes everyday life experience when we say "life is like a river", across perceptional analogies as in "this vegetable looks like a face" to more complex ones in scientific domains such as "an electrical circuit is like a system of water pipes" or "let's solve this problem like we solved the old one".

Scientists have accessed analogical reasoning and problem solving from different points of view, like general problem solving (Hellman, 1988), linguistics (Harris, 1994), scientific discovery (Holyoak & Thagard, 1995) or even design (Hofstadter et al., 1995). Current research not only provides us with new models for the understanding of human reasoning through consolidation of known cognitive aspects such as recognition, categorization, in/deduction, abstraction, similarity-estimation and generalization, but also for the construction of artificial reasoners in terms of automatic programming (Schmid, 1998) and analogical problem solvers in AI (Veloso, 1994).

Expanding on the examples provided before, analogy-making roughly can be defined as a set of processes within a cognitive system which *encode*, *map* and *transfer* entities and relations of domain-knowledge from a *base domain* to a *target domain*. These processes enrich knowledge about the target domain with the help of knowledge about the source domain.

Although this general intuition may be accepted by most scientists in the field, the question of how the concept of analogy making is properly defined or implemented remains unanswered and is heavily under discussion. This is mostly due to the generality and diversity of the topic. There not only exist a large amount of possible approaches, but due to the high cognitive level of analogy making it is difficult to determine the role of analogy making in the human cognitive system as a whole. Finding this role and constructing plausible models depends heavily on the progress in cognitive psychology, artificial intelligence and the theoretical neurosciences to develop feasible notions of knowledge-representation and similarity-estimation of natural and artificial cognitive systems in real world scenarios.

Although up to now a decent set of computational models based on substantial psychological research has been developed or is worked upon, each of them is only able to model some properties of analogy-making successfully and neglecting others. Therefore, this paper is aimed at highlighting the different aspects of analogy research and its models as well as show up different strategies used in analogical problem solving in an empirical investigation.

The thesis is structured seven parts. This first part introduced into the topic and provides the reader with the basic concepts of analogy; the second part provides an overview about core past research, strategies and processes in analogy-making, while the third part presents own experiments and empirical data. The fourth to seventh parts include the reference, appendices and formal considerations concerning the examination regulations.

Analogical Strategies in Problem Solving

Introduction

Major contributions to the field of analogical research came from many sides, which led to different assumptions about the role of analogy in human cognition. These assumptions range from theories about analogy in problem solving or perception to approaches which regard analogy making to be the underlying mechanisms of all higher reasoning. This chapter will summarize the most important theories and psychological models in short to provide an overview about the central processes and constraints of analogy making. Not all of the approaches presented are crucial for the understanding of the empirical part, but they provide extensive background knowledge about analogy making in general as well as derivational analogy and problem solving in special.

Research in analogy making is heavily influenced by the interplay of psychological experiments and computational modelling. While psychological evidence supports theories about analogy making and shows the multiple constraints which influence the process of analogy making in humans, computational models derived from these experiments allow confirmation of the results by means of implementation and evaluation. Also, the resulting expert systems can be used in various applications. A computational model of analogy making thus is as psychologically plausible as the degree of constraints that it can implement and the experimental results it can predict.

Most research groups do both, experimental work and computational modelling. While integrating the results of other groups, they construct a mutually supporting system of evidence, evaluation and justification. As we will see in detail later, this development on the one hand lead to certain shared assumptions which define a common notion of analogy but on the other hand also to crucial differences between accepted paradigms in analogy research. These differences concern the constraints included in the theories and are discussed later in more detail. First we will try to clarify the common notion of analogy.

In general, an analogy is a special relation drawn between at least two domains, termed the base domain and the target domain. Both domains contain certain entities, most often physical or mental objects like apples or letters. These objects have attributes with only one argument, like being red, and are connected within a domain through certain relations, like being earlier in the alphabet and so forth. These relations are called first-order relations, because they take objects as arguments. On top of these relations we have second-order relations, like causation, which take other relations as arguments. This network of objects, attributes and relations build up a system of statements about the domain, which allows logical reasoning, prediction and explanation within that domain.

The interesting point now is, why should we connect two domains which each other, and how do we do so? A need of most reasoning systems is to enlarge their knowledge about objects, their attributes and relations of certain domains to predict the outcome of actions and provide explanations. For this end, a common assumption is that principles or relations are often true in more than one domain, for example because all systems depend on the same law of nature and causation and good solutions tend to true and working in more than one domain. Assuming this to be true, it would be economically interesting to transfer knowledge from domains we know to domains we do not. But before we think about transfer, what is 'knowledge' in this case at all? In our notion, transferring knowledge means to transfer sets of relations. In general, promising domains between which we would like to transfer knowledge could be very different, that is, they could contain very different objects and attributes. Even the first-order relations are quite different normally. But what seems to be quite constant and therefore easily comparable are the second order relations. These meta-relations like causality bind first order-relations and thus objects in a certain framework which may be, abstractly seen, the same in many domains. So to transfer knowledge we should try to match groups of higher-order relations from one source-domain to a target-domain to gain knowledge about causations and the interplay of object and relations in the latter. This *mapping* by means of *structural* and *semantic similarity* is one of the core strategies of analogy making in general, although humans may rely on other strategies in cases of analogical problem solving if certain constrains apply, as this thesis will show.

Thus, structure and the superficial similarity of domains are essential for making useful analogies, that is, analogies which lead to transfer and, subsequent, learning. Other crucial elements of analogy making are retrieval, where a possible base domain is remembered from long term memory, evaluation, in which a mapping is rated for goodness, and transfer, where knowledge is transferred if a match is very promising. All these processes are influenced by a set of constraints. Both we will discuss in detail later.

Traditional models are focusing on structure and symbolic notions of similarity. They emphasize a serial and modular view, thus separating mapping and representation. This emphasis on structure is supported by a system of psychological experiments, showing the importance of second-order relations when comparing different domains and looking for similarities. The representations used within the traditional views and models are supposed to be pre-coded, what means either more or less neglected in psychological research or pre-formalized by an human instructor in computational modelling. This is due to the claims of simplicity and domain-independence most traditional views and models support. The seriality and modularity of these views and models is mostly due to the orientation on symbolic models and compositional theories which claim distinct sub-processes and stages of processing in analogy making. This approach can be termed symbolic-structural and has been refined over the years to comply with new psychological results.

Newer approaches are mostly connectionist models which implement multiple constraints. By automatic re-representation of knowledge and parallel integration of processes, they favor a more cognitively and computationally feasible approach. Most of these models rely on some form of connectionist network, may it be local or distributed, to include also cognitive and pragmatic constraints. The advantages of this technology are located in the implicit ability of similarity-processing of patterns and parallel processing by activation spreading between nodes representing features of knowledge. This approach can be termed connectionist-holistic and includes also hybrid systems.

Analogies itself are used in different meta-domains. The most important ones are explanatory analogies in analogical reasoning one the one side and analogies used problem solving on the other side. While analogical reasoning in general is mostly considered to be driven by inference possible through higher order relational mapping and the resulting power to construct explanatory analogies (Gentner, 1983), analogical problem solving is considered to be driven by the adaptation of a solution procedure from old solutions to new problems (Keane, 1996). Related to the latter and maybe including it is Case Based Reasoning (Kolodner, 1993). This distinction between analogical reasoning and analogical problem solving reflects not only different application but also two major strategies that are proposed to be used by human analogy makers: transformational strategies and derivational strategies. In the course of thesis I will try to discuss the properties of these approaches and provide evidence for the usage of the latter in complex problem solving scenarios.

Psychological Evidence

Psychological experiments in analogical reasoning mostly concentrate on two sub-processes, namely analogical *retrieval*, where a past situation is remembered and used as the base-domain, and the subsequent analogical *mapping*, where the relations and entities of the retrieved base-domain are associated with the relations and entities of a chosen (or encountered) target domain. By matching these domains, inferences can be *transferred* to enrich the target domain with knowledge from the base domain. The new knowledge then could be *abstracted* to a more general mental model.

Analogical Retrieval.

Preliminary to the process of mapping a matching base analogy within the source-domain has to be found. This process is termed analogical retrieval or retrieval in the context of this work. Interestingly, people often fail to retrieve a potentially useful base domain for an analogy from long term memory even if there is a high structural similarity between base and target and the base is fresh in memory. Experimental evidence shows that people often need more than one useful base domain or an explicit hint to use a certain base domain to successfully retrieve and start the mapping (Gick and Holyoak, 1983).

Retrieval can be facilitated by surface similarity, as the occurrence of highly similar objects in base and target domain. It was shown experimentally that people were four times more likely to retrieve a story from memory if it was superficially similar to a new story than when it was structurally similar to the new story. Interestingly, ratings for goodness of the mappings between retrieved stories were much higher if the stories shared structural similarities than when they shared superficial similarities. This finding is also verified in problem solving tasks (Ross, 1989), where retrieval is again affected by surface similarity but structural similarity is a better predictor for success in solving the problem.

Structural differences and similarities that influence pragmatic constraints also affect retrieval and transfer. The goodness of a source analogy may therefore be judged through the degree of similarity of initial state, goal state and a set of solution constraints of source and target. (Carbonell, 1983)

Domain expertise also may facilitate retrieval. There is evidence that experts and intermediates retrieve fewer surface similarities and lures in problem solving tasks and also reject them faster than novices. This may be due to a higher representational uniformity of the relations in the memory trace of the experts and thus a higher overall similarity of base and target domains. This is supported by experiments showing that retrieval is more likely if both domains share relational predicates with high similarity, termed manifest similarity (Clement et al, 1994). Mapping is lesser affected by this property, due to the higher degree of freedom of re-representation in the mapping process.

As might be expected, failure of retrieving a useful source analogy may also be grounded in an earlier step, for expertise seems to play an important role in representation of source analogies. Novices seem to rely more on superficial features representations of the source whereas experts construct more structurally sophisticated representations. (Chi, Feltovich & Glaser, 1981).

Analogical Mapping.

The earliest experiments regarding analogy making concentrated on relations like 'A corresponds to B like C corresponds to ?'. In the seventies this new field introduced formal systems for specifying the representations and developed computational algorithms for analogical matching and inference for implementation (Winston, 1979).

Researchers concerned with Artificial Intelligence soon became interested with the possibilities of using analogy making to solve problems, like modelling the transfer of solution strategies from a solved problem to a new one (Carbonell, 1983). Psychologists heavily participated in the field and started investigating analogy-making in humans under the aspects of problem solving and learning. Experiments in the eighties showed that humans prefer structurally consistent, one-to-one structural isomorphic correspondences between entities in different domains in their mappings (Gentner, 1983). These kinds of mappings which base on structural similarity result in a mapping with much higher overall consistency than alternatives with only superficial object similarity. Most influential in this claim is the structure-mapping theory (Gentner, 1983). This structural constraint was soon followed by the notion of pragmatic constraints by the pragmatic mapping theory (Holyoak, 1985), which focused on problem solving and goal directedness within the mapping process. Experimental evidence supports this claim, as could be shown that people preferred mapping entities which brought them closer in reaching their goals (Spellman & Holyoak, 1996). Structural and the pragmatic constraint were combined in the multi-constraint approach (Holyoak & Thagard, 1989). The notion of the structural constraint has been further refined to the principle of systematicity, which claims that systems of entities connected by higher order relations are better suited candidates for mapping than local matches due to their inherent causal importance and interrelatedness (Clement & Gentner 1991). The ease with which a candidate for mapping can be adapted to match with the target also predicts the selection of inferences through the adaptability constraint (Keane, 1996).

Later research also investigated non-isomorphic mappings (Spellman & Holyoak, 1996) and the impact of different degrees of structural overlap (Reed, Ackinclose & Voss 1990), (Schmid, Wirth & Polkehn, 2003). Results of these studies show that people tend to make partial mappings outgoing from known relations which are adapted and refined through the course of the process.

The presence or absence of systematicity in the base- and target domains is an important factor that influences performance. This is closely related to the transparency of the mapping, that is, the degree to which entities within similar relational systems are superficially similar. The influence of transparency on mapping has been shown with the teaching of algebra problems and the solving of structurally similar problems. People have been best in the high surface-similarity condition, in which the variables kept their names, and worst in cross-map condition, in which similar objects appeared in different roles across the two problems.

Properties of the reasoner herself like expertise in a domain or age have also influence on the mapping. In experiments with children (6 and 9 years), both younger and older children were influenced by transparency, but only older children could make use of systematicity (Gentner & Toupin, 1986). This development is hypothesized to be due to knowledge-driven relational shift from concentrating on object matches to focusing on relational matches (Gentner & Rattermann 1991) and the developmental increase of processing capacity (Halford et al., 1994).

Also influential on mapping are time, processing load and functional brain lesions. As has been shown (Goldstone and Medin, 1994), mapping relations seems to be more time consuming than mapping objects and their attributes. Mapping is also influenced by immediately proceeding experiences, for it has been shown that carrying out a similarity comparison just before mapping facilitates an higher order relational mapping (Markmann & Gentner, 1993). Also it could be shown that backwards-counting and similar ways to increase processing load leads to shallow or false mappings (Kubose, Holyoak & Hummel). Recent work by Holyoak has shown that damage to the prefrontal cortex leads to lower performance in mapping tasks, although causality and the functional dependencies which cause this are not clear yet.

Analogical Evaluation, Abstraction and Learning.

After retrieval and mapping are finished and the inferences have been drawn, the analogy can be evaluated for its goodness. Possible criteria for this evaluation are structural soundness, factual correctness and relevance of the drawn inferences and the resulting enriched target domain. The degree of influence of these criteria may vary depending on the situation and the goals of the reasoner. After evaluation the analogy is either judged as inferior and the process of retrieval, mapping and evaluation is repeated with another mapping or even another base domain. Or the common mental model that underlies the freshly gained knowledge is abstracted and stored in schema. This is partly confirmed by experimental data; people who read two analogous stories are better in retrieving the common structure than people who read only one story. Also actively comparing analogous stories enables better retrieval than reading the stories separately (Gick & Holyoak, 1983). Even if learning does not depend on analogy-making as a whole, the two topics are closely related. As we have seen, analogy making uses inferences to enrich the target domain with knowledge of the base domain. Together these processes of abstraction, inference drawing, difference detection and re-representation can lead to learning and schema abstraction (Gick & Holyoak, 1983). But apart from this, learning in relation to analogy making deserves a closer look, because analogies also have proven to be useful in teaching. Special problems of analogy makers, like the failure to access potentially useful analogies unless they share a high surface similarity with the target, also apply to known problems in education, for example the inert knowledge problem. The key feature of learning seems to be the refinement of certain competencies or skills, mostly achieved through the gathering of knowledge and the building of mental models of the world, may it be implicit or explicit. Learning seems to be a core feature of intelligent organisms in order to adapt to the environment and, more important, survive. Therefore the ability to learn seems to be a fundamental requirement and indicator of biological as well as artificial intelligence. Psychologists have termed different kinds of learning in the past, such as learning by Deduction, by Abduction, by Induction /Examples, by Memorization, by Analogy, by Instruction and by Observation. Even broader categories are established, like types that include a direct implanting of new knowledge like in learning by being programmed or modified by an external entity (Carbonell, Michalski & Mitchell, 1983). So learning is not only a crucial topic for biological organisms, but for all agents which live in a complex and dynamic environment. Learning by analogy means acquiring new knowledge about an input entity by transferring it from a known similar entity. As such it is a process of gaining information about a new domain by using existing knowledge about an older and analogical related domain. This seems particularly useful in the learning of complex systems, like scientific models and there like, because knowledge reusability seems especially important in a highly interrelated field like the sciences. Analogy and similarity play a central role in the conceptual change that characterizes learning, allowing the application of pre-existing conceptual structure to new problems and domains, and therefore supports the rapid learning of new systems. Of all the learning processes, analogy is the only one that offers a mechanism for the acquisition of substantial knowledge structures in a brief span of learning. This is supported by recent experiments (Nokes, Ohlsson & Corrigan-Halpern, 2002), which compare performance of learning in direct-instruction conditions and analogy conditions. Although overall accuracy was the same in both groups, the analogy-group took much less time to accomplish the task. By contrast, other learning processes, such as generalization, differentiation, accretion, or compilation, offer ways of refining, adding to or consolidating an existing system of beliefs. Further, similarity is a primary factor in categorization, retrieval, and organization of conceptual structure, and analogy is primary concerned with mapping of similarities, therefore being a core component of these processes. Due to this analogies are commonly used in science instruction, for example when billiard balls are compared to particles or an electrical circuit is compared to a water-flow system in physics. Similarity is used to extract principles from repeated, overlapping instructions to construct a generalized model that fits into the current world-knowledge of the learner. Student's models of a domain are shaped by such instructional analogies, using them to apply arguments used in examples or older instructions to new problems. As was stated before, also spontaneous analogies can occur and learners' first naive models are often formed around such implicit relations between domains. This can of course lead to partially wrong models, which dominate many of the mental models that we form of domains in which we are not experts. Only a very few computational models of analogy making account for learning by extracting a generalized abstract schema, like SEQL (Skorstad, Gentner & Medin, 1988), or LISA (Hummel & Holyoak 1997).

Derivational Strategies

While these traditional approaches in analogical reasoning stress the *transformational* aspect of the processes involved, other strategies than relational mapping of entities are promising. One of the most potential candidates for an additional strategy is *Derivational Analogy in Analogical Problem Solving* (Carbonell, 1986), a general form of case-based reconstructive reasoning. This analogical problem solving strategy emphasizes the adaptation of old problem-solving traces, that is, old solutions, to new but similar problems. Problem solving in this context is regarded to be a *controlled search* in a library of annotated solutions where different alternatives to reach sub-goals are *generated*, *explored* and *justified*. Thus, a large amount of knowledge is necessary.

The major contribution of this approach is the avoidance of a completely new search effort, as it is necessary in the mapping phase of transformational approaches. Instead, problem solving is divided into decision points and sub-goals which are stored as a solution-trace to a given problem. An interleaving process of justification and replay of multiple partial solution-traces guarantees applicability and soundness of the complete analogical solution to the problem. This reuse of solutions, the incorporated notion of justification and avoidance of exhaustive search make the derivational strategy feasible in complex domains and in cases when transformational mapping is complicated by other means like low superficial similarity and complex structures.

Derivational analogy has been successful in computational applications (Carbonell, 1986), especially as an effective strategy to adopt previously stored and annotated plans. It has later been refined to include also a retrieval process (Veloso & Carbonell, 1993).

A common mentioned example (Carbonell, 1986) may serve as illustration. Imagine, you want to translate a sort algorithm implementation from an imperative language into a functional one. It is quite hopeless to try a line by line translation from, for example, Pascal to LISP, due to the very different approaches of these languages. But what remains in both implementations are the major *structural and control decisions* the programmer has to make. Derivational analogy tries to capture and reuse these abstract lines of thought by replaying a previous plan or derivation and modifying it to satisfy new needs.

A common reason for the use of analogy in problem solving is the poor knowledge of relational structures in the target domain and the superficial differences between the objects of source and target domain. Transformational approaches can map entities and generate relations as long as the problems include only few steps, but they fail with more complex problems (Melis & Carbonell, 1993). Derivational analogy as control strategy for search and thus planning and problem solving may provide us with solutions to more complex problems.

Successful implementations of the derivational approach in domains like automatic programming, theorem proving and interface design include ABALONE (Melis & Whittle, 1997), APU (Bahansali & Harandi, 1993), BOGART (Mostow, 1989), REMAID (Blumenthal, 1990) and Prodigy/Analogy (Veloso & Carbonell, 1993). Computational models have to check after each decision point if the underlying assumptions and the context of the problem are still valid for transfer - otherwise more derivations have to be included or the strategy has to be changed.

Additional to their competence in complex environments, derivational approaches may produce concrete solutions that cannot be obtained by transformational approaches (Melis & Carbonell, 1993) and include justifications and also explanations which may possibly serve in tutoring systems. Up to now, most of the evidence generated in favor of the derivational approach is contributed by computational models. Questions like if derivational information can be found in spontaneous human analogy remained unsolved and will answered partially in this thesis.

Methodology

Analogical research is rather young and its processes are mostly higher-level and thus difficult to assess directly. Its methodology is not as well developed as in psycholinguistics or neuropsychology. Characteristic features of experimental task are recall of analogies and the production of mappings under different conditions, for example the superficial and structural similarity of the material. Reaction Time studies are nearly never done but were used with success as in the experiment presented in this paper. The stud of analogical capabilities of impaired patients is still in the beginning, but seems promising as analogy making relies on many different cog native processes connected to reasoning and memory.

Computational Models

Analogy making has been in the focus of computer scientists for quite a long time. Drawing inferences from known cases to new situation is a common technique in automatic programming, case-based reasoning and an important component of various expert-systems. Beside that, computational models of analogy making have proven useful in modelling aspects of reasoning of humans and animals as well. Two of the main challenges of these approaches to analogy-making are the so-called selection problem, that is, how to control the range of inferences that are drawn, and the problem of representational flexibility, that is, how to construct mappings that do not require absolute identity matches but allow 'fuzzy' associations. The solution to these problems is complicated by the fact that no normative notion of correctness or validity of a solution exists and therefore the evaluation of mappings is hard get. Also the grade of connectionism, parallelism, the modularity of the involved processes and the degrees of freedom such a system should have is still under discussion. Additionally, most implemented systems can use only heavily pre-structured data, don't provide a unified treatment of syntactic and semantic features or use unfeasible amounts of memory. In the following part I will try to highlight different approaches and processes that play an important role in computational analogy-making.

Constraints

All models try to consider a closed set of constraints, which influence the process of analogy making and serve as a possibility to evaluate the models. How the models try to fulfil these constraints is quite different and depends on the architecture of the model.

The structural constraint emphasizes the selection and mapping of sets of relations between entities, thereby ignoring superficial similarities of single entities and concentrating on the transfer of causal and other higher-order relations. This was the first constraint implemented and is widely regarded as the most important for making analogies.

The semantic constraint facilitates retrieval and mapping by taking semantic relations between entities into account, thereby providing a possibility to compute the similarity between two entities. It is difficult to fulfil, as a semantic network or a related hierarchy has to be established. This constraint is also valuable for pre-processing the data before retrieval in order to semantically unificate entities and relation before the structural constraint can be fulfilled.

The pragmatic constraint deals with the goals of the analogy maker and contextual effects. In some cases a special outcome of the analogy making process is wanted, and this constraint should combine goals, plans and other contextual data to enhance the process.

The cognitive constraint makes allowances for memory and processing capabilities of the analogy maker. This is mostly applied to lower the memory requirements of computational models to make them more plausible in regard to human reasoners.

Processes

Although many different approaches and models exist, a closed set of sub-processes of analogy making seems to be plausible. In how far these processes interact, are controlled or implemented is hugely different in each approach. Common is, that all models try to establish some kind of mapping between a given target and the best fitting of one of the many potential sources for an analogy. The mapping is influenced by the above-mentioned constraints. Encoding: this process encodes and pre-structures representations so that in can be later accessed as source in order to map analogies or gather information about entities or relations. As mentioned before, the process of building an own representation from real world data of any kind is absent in most models. Instead, these models rely on a pre-coded formalism that is fed into the mechanism. Only a few models construct their own high-level representations, for example ANALOGY (Mitchell 1993) or the trio Copycat, Tabletop and Metacat (Hofstadter et al 1995; French 1995). Other models rely on re-representations of older episodes, like AMBR (Kokinov & Petrov 2001). Symbolic models use predicate logic or vectorization to store entities and relations and distinguish in source and target domain, while models with connectionist components encode data in interlinked nodes or patterns of activation. Hofstadter has argued convincingly that analogy making cannot be adequately modelled by using explicitly structured representations fed to an analogy maker (Chalmers, French & Hofstadter, 1995).

Elaboration and Retrieval.

This process involves the selection of potential useful sources for analogies from long term memory under consideration of the above-mentioned constraints. Upon retrieving the relevant source, it is often necessary to determine additional features and relations of source. In some cases developing a specific problem solving strategy or explanation in the source domain for the target domain might be inevitable. This process has been comprehensively studied experimentally. It has been proven that retrieval of a base domain is easier if it is superficial similar to the target domain. Also facilitating is a higher familiarity of the base domain, a higher richness of representation, a generalized schema connecting the domains and, to a lesser degree, structural similarity. Most models are based on extensive search of long term memory.

Mapping.

This process involves building a map from the features of source to the features of the target domain under consideration of some of the above-mentioned constraints. This process can be seen as the core of analogy making and is implemented by all computational models. The difficulty is the selection of the right mapping out of the many possible ones. This task is computationally quite complex and cognitive unfeasible if the search is not properly constrained.

Inference and Transfer.

The target domain is enriched with entities and relations of the source domain under consideration of the constraints. This process is normally integrated into the mapping process and can be considered as the goal of the analogy making process, for new relations are discovered and knowledge about the target domain is gained.

Evaluation and Justification.

This process examines if the mapping between source and the target is valid and applicable and assigns a degree of confidence in the solution. If the goodness of the mapping is rated too low, modification of the original mapping and backtracking occur. Due to the high interrelatedness with mapping and transfer processes, it is often contained in these processes.

Learning and Storage.

This process stores the gained knowledge for future use, thereby enriching the pool of possible sources for analogies and eliciting changes in the overall knowledge structure of the system. This can be considered as learning and is implemented by only a very few models.

Implementations

Introduction.

We can differentiate between three kinds of computational models or paradigms which model analogy making in its various aspects. The traditional approach is based on symbolic architecture and uses distinct localist representation for the entities in the domains and a set of rules often combined with predicate logic to determine the mapping. In the end of the eighties more and more connectionist approaches were used and the entities were modelled as overlapping patterns of activity in a highly parallel neural network. Some of the latest developments combine these approaches and create a hybrid approach to combine symbolic representations with local and distributed connectionist activations. Since the first connectionist models of cognition were developed, the domain of computational modelling was enriched with the connectionist paradigm.

Whereas the more classical symbolic paradigm claims the ability to manipulate symbols to be a crucial or even necessary property for cognition and its modelling, symbolics is confronted with a set of problems which seem so far inherent to the approach itself. Belonging to these problems are at first hand the formal notions of undecidability and incompleteness and further as well as the biological implausibility of extensive search, immense serial processing effort, inflexible knowledge bases and the difficult implementation of semantic similarity.

Connectionist implementations offer new approaches to these problems, offering parallel models for activation spreading which accelerate search, incorporate learning and flexible representations. But they also pose new problems, like feature binding, huge memory consumption, unclear semantics and arbitrary selection of parameters which render models uninformative by making them able to comply with all sorts of data by the adjustment of the parameter values. Both paradigms gave rise to different architectures and approaches to models of analogy making and had success in different areas of the field. Whereas representation-building is not the strength of any of the following models, most of them are capable of retrieving, mapping and evaluating analogies in a satisfactory, although not always biological plausible way.

Symbolic Architectures.

The earliest model capable of analogy making was the so suitable called ANALOGY (Evans, 1964). It focused on solving multiple-choice geometrical problems as they are known from quantitative intelligence tests. The input was a low level description of the geometric shapes and the architecture was able to build high-level representations on its own and conduct a set of candidate rules through mapping. The most specific successful rule was chosen for the answer. This model has been proven to be doing well in this limited domain. The still most influential family of models of analogy making are based on the Structure Mapping Theory (Gentner, 1983). It has been the first one focusing on structural similarities between the domains and emphasizing the importance of relational mappings based on this theory is the Structural Mapping Engine (Falkenhainer et al, 1989) which models mapping as an isolated, modularized and serialized process. This first set of research can roughly be labelled as the symbolic-structural approach.

Gentner's Structure Mapping Theory (SMT) was the model in the field, computational implemented in SME by Falkenhainer, Forbus & Gentner and coupled to the refining pre-processor MAC/FAC (Forbus, Gentner & Law, 1995). In SMT process of analogy is traced back to the set of discrete subsystems of accessing, mapping, evaluating, storing and generalizing structural similarities and establishing an isomorphic mapping between the higher order relations of source and target domains. Characteristically for the Structure Mapping Theory are the Relational Matching Principle, which emphasizes the mapping of relations over the mapping of attributes, and the Systematicity Principle, which underlines the mapping of coherent higher order relations over first order relations. SMT has been successfully implemented in SME and the refined MAC/FAC model. Even with the MAC/FAC extension this system works highly redundant and with a huge need of memory, the data needs to be pre-structured and pragmatic and semantic constraints were not implemented. It focuses on higher-order relational matches and is only able to map identical relations. A set of promising solutions are evaluated by a greedy merge algorithm which finds the most consistent mapping. This approach implemented solely the structural constraints, although newer versions also contain weak pragmatic and semantic considerations.

To provide the possibility of analogical retrieval, the Structural Mapping Engine was later coupled with a model of analogical retrieval called MAC/FAC (Gentner & Forbus, 1991). The model tries to implement the experimental findings about retrieval, especially the dominance of superficial similarity in retrieval. Therefore the architecture is build around two stages and uses two kinds of representations, a short episode vector for representing superficial search and a detailed predicate calculus representation of the domains, representing later in-depth structural search.

The Incremental Model of Mapping IAM (Keane, 1993) focuses on the effects of the order in which material is presented to analogy makers and was the first model which tried to incorporate cognitive constraints like working memory limitation and background knowledge by implementing results of empirical findings of psychological experiments (Keane, 1993). While models like SME and ACME implemented Marr's Computational Level quite successfully, IAM was the first model to account for the Algorithmic Level of Marr's Metatheoretical Framework (Marr, 1982). The model uses several techniques to reduce processing effort by means of iterated serial constraint satisfaction and evaluation, producing systematically the same errors like humans in the same task.

Connectionist Architectures.

In the mid-1980 connectionist models got popular again and quickly the first analogy makers based on this approach were developed. Holyoak's and Thagard's ACME (1989) is a localist network using multiple constraint satisfaction by theoretically implementing pragmatic and semantic constraints together with a lower computational complexity but without a proper retrieval engine.

STAR-1 was the first distributed connectionist model of analogy-making (Halford, et al, 1994), followed by LISA (Hummel & Holyoak , 1997), which uses distributed representations of structure relying on dynamic binding and aiming at psychological plausibility and neural feasibility.

The newest models try to incorporate as well symbolic as connectionist architectures in a general cognitive design, thereby forming the hybrid approach, for instance in COPYCAT/TABLETOP (Mitchell, 1993, Hofstadter, 1995) and in AMBR (Kokinov, 1994). The central intuition is, that high-level cognition is an emergent result of locally interacting low-level processing units, thus incorporating contextual and semantic knowledge. These models are theoretically able to construct their own representations and they try to merge the processes to one interactive and parallel super process, discarding most of the sequential approach of older models.

Until recent times the scientific field was dominated by symbolic models due to their relative representative concreteness and simple seriality of processing which made them suitable for implementation on common platforms. Distributed representations re-emerged during the eighties as new mathematical models and applications were developed. These kinds of representations provide an inherent measure of similarity and processing of patterns in a highly parallel way. Specifically, this possibility of dealing with similarities is hard to implement with purely symbolic models, due to the natural fuzziness of the real world which resists to be formalized in atomic representations.

The first localist connectionist model that scored on the field was the analogical constrained mapping engine ACME (Holyoak & Thagard, 1989). It was based on the Multiple Constraints Theory, a concept still widely used in nowadays systems. The model tries to satisfy a set of three constraints simultaneously in mapping: structural similarity, semantic similarity and, opposed to Gentner's models, pragmatic importance. ACME is implemented as a cooperative algorithm for mapping based upon Marr's and Poggio's treatment of stereoscopic matching, in which this algorithm is a procedure for parallel satisfaction of a set of interacting constraints. Fed with representations, it builds a localist constraint-satisfaction network in which each node represents a possible hypothesis for an analogy. These candidate-hypotheses are connected through exhibitory and inhibitory, thus implementing the structural constraint. Through this links, consistent hypotheses strengthen each other and contradictory hypotheses. During the relaxation of the network, it gradually moves towards a state of equilibrium while the hypotheses are evaluated in a parallel way and the most active ones win the competition. This set is then considered the best mapping for an analogy. Deeply connected to this is ARCS, the retrieval engine of ACME. While the mapping is dominated by structural similarity, the retrieval is mostly influenced by semantic similarity. The major disadvantage of this early connectionist model is its huge need of working memory and its exploding complexity and amount of links. Specifically, ACME is too effective and memory-consuming to be biologically plausible. The arbitrary semantic interpretation of activation values as well as the presumed knowledge about construction of links and the arbitrary degree of freedom make the model difficult to justify empirically.

The first distributed connectionist models were the STAR family. STAR-1 (Halford et al, 1994) was based on the tensor product connectionist models developed by Smolensky. The newer STAR-2 (Wilson et al, 2001) is able to map complex analogies in a constraint satisfaction network similar to ACME, but with the ability to focus on different parts of the domain sequentially, thereby reducing memory load. The main interest of the developers of this model is in exploring limitations in performance in humans, and the model tries to imitate processing constraints by limiting the computational complexity of the task.

On of the newest models is LISA (Hummel & Holyoak, 1997). This model uses dynamic binding to establish the mapping. Whole patterns of oscillating activation are considered to be bound together. While the working memory representation of the architecture is distributed, the long term memory is localist with separate units for episodes, propositions and predicates. Predicates and their arguments are represented as distributed patterns of activation over units representing semantic primitives. Retrieval is done by spreading activation, mapping by learning new connections between the most active nodes. These phases are integrated, but still sequentially. This architecture is a substantial attempt to account for the cognitive constraint as we have discussed it earlier. Concepts like limited working memory capacity, partly serial mode of processing and organization of memory are at least partially accounted for. Due to this, LISA aims not only at psychological plausibility but also at neural feasibility and tries to provide a unified cognitive architecture for processes involved in analogy making, like perception, pattern categorization, schema construction and generalization. While comparable connectionist architecture like ACME focus on a complex architecture, LISA instead shifts to higher a complexity of processing by making use of temporal properties like oscillation of patterns and binding of nodes. The major drawbacks of the model are the same as in most other ones, for the representations have to be pre-fed to the model and the implementation of the semantic constraint remains unclear. Also the model has difficulties to deal with certain ambiguities of predicates which may arise as an intrinsic feature of the architecture itself.

Hybrid Architectures.

In the last years a new paradigm emerged between the symbolic and the connectionist approach: hybrid architectures, which try to incorporate the best of both worlds. It characteristic idea is that higher cognition is the result of continual interaction of simple units capable of local symbolic computations and connected with a network of activation spreading. The emerging systems are not only flexible, parallel and clearly structured but also able to compute semantic similarities and account for context-sensitivity. Successful examples for this type of models are Copycat/Tabletop and AMBR.

Copycat / Tabletop (Mitchell 1993; Hofstadter et al 1995; French 1995) is a computational analogy-maker characterized by its limited domain of input, letter strings like ABC:DEF::GHI:???, and its ability to build own, context-sensitive representations of a problem. It has been designed to combine bottom-up emergent processes and top-town semantic processes in a delicate way. The architecture consist of a short time working memory, a semantic long term memory in which concepts and relations are dynamically stored, and a procedural memory. In the procedural part of the memory, termed coderack, a set of simple, nondeterministic agents are creating and modifying themselves and structures from the working memory with help of the more static information from long term memory. The system gradually develops a consistent set of structure which makes up a context-induced mapping of concepts by disregarding their usual dissimilarities. Like other hybrid models of analogy-making, distinct processes like representation building, mapping and evaluation run in parallel and context dependant and influence each other constantly. Due to its ability to re-represent structures, its flexibility and its indeterminism, the architecture is also considered to model fluidity of concepts and thus properties of creativity. Drawbacks of this approach are the obvious limitation to artificial letter-strings, overgeneralization during the mapping and no implementation of learning.

The AMBR model (Kokinov, 1994) is chiefly characterized by its claim to implement the general cognitive architecture DUAL which tries to model human cognition by means of a hybrid multi-agent system. DUAL consists of two memory spaces, a long term memory and its active part, the short term memory. These memory spaces are filled with simple agents, each representing a piece of world knowledge. Coalitions if these agents form concepts and episodes. Each of these agents is a hybrid by itself. While the static-symbolic part processes declarative and procedural knowledge at a speed proportional to its relevance, the dynamic localist-connectionist part is connected in a network of spreading activation which computes this relevance of each agent in regard to the current context. AMBR implements this cognitive architecture in processes of recall, mapping and transfer which are computed in parallel by the collective of agents. While the symbolic level is the core of all processes and models the competence of the system while the connectionist level limits performance to a biologically feasible degree by allowing only the most active nodes to process data and thus construct working memory. The model is also able to successful model memory phenomena like blending, semantic order effects and context dependence. Major drawbacks of this model are the high amount of parameters and the low flexibility of the nodes. AMBR has been refined to AMBR-2 (Kokinov & Petrov, 2001) to de-centralize the processing and further the modelling of episodic memory, illusory memory, blending and priming.

Conclusion

As has been presented in the preceding part, analogy making in general and especially analogical problem solving are influenced by a wide amount of processes and constraints as well as implemented by a set of complementary models. So why introduce a further analogical strategy, and how to relate it to the existing approaches?

First of all, because it seems counterintuitive from a certain point of view that relational mapping is the core process of analogical problem solving. The memory and processing amount of the mapping, even if properly constraint and implemented, may be feasible for computer systems in artificial scenarios, but real world problems seem to be more complex on the one hand and unpredictable on the other to solve them by the limited abilities of human wetware. Also, at least the conscious self-observation of the writer's problem solving process in complex scenarios does not mainly show up highly complicated relational mappings; it rather seems to involve the recollection of old solutions methods. Therefore it seems economically and intuitively interesting not to save whole solutions and map them, but to store solution *traces* and important decision points which could be more easily adaptable to new problems. Derivational analogy seems to provide a compatible theory for these observations.

Derivational analogy should thus not be understood as an alternative to transformational approaches, it clearly is not, but complementary to it, limiting itself to problem solving in complex domains where old solutions are already available. Also, as derivational approaches are rather young compared to transformational ones, its implementations and supporting psychological theories are limited. This thesis is aimed at supporting the approach with these psychological evidence by exploring the use of derivational strategies in problem solving and some of the circumstances under which it occurs.

Empirical Part

Introduction

The experiment was designed to investigate the use of derivational strategies in mapping and transfer in certain cases of analogical problem solving.

The hypothesis is that subjects confronted with a relatively complex graph problem would rather adopt derivational than transformational strategies if a derivable solution is presented and relational mappings are made difficult by lowering superficial similarity between the problems.

This hypothesis seems reasonable (1) due to the need for alternative strategies in cases when relational mapping cannot be applied by humans by reason of missing constraints to limit the search process and (2) due to the competence of derivational approaches in exactly that kind of complex planning scenarios with the need of search control as mentioned earlier. If this hypothesis can be confirmed by the experimental results, humans may use different analogical strategies and the supposed dominance of transformational strategies in analogical problem solving is dissolved.

The experiment is separated into two steps. Firstly, we want to demonstrate that in the case of low superficial similarity between source and target-problem, participants use a strategy that is clearly non-transformational if they have a detailed solution to the source problem at hand. Secondly, if later asked, subjects would report that the strategy they used shares satisfactory more attributes with a derivational strategy than with a transformational strategy.

While we were able to use reaction time as the dependant variable for the first step, we were not able to find such a reliable variable for the second step. Therefore we used a qualitative questionnaire to access the self report of the subjects.

To reduce the bottleneck of retrieval in analogical problem solving as much as possible, we always explicitly provided 'helpful examples' (as was first used in Novick & Holyoak, 1991) together with a solution to a similar problem to simplify the process of retrieval. Thus we tried to enable the subjects to concentrate on the mapping- and transfer-strategies we wanted to test in this experiment.

Pilot Study

A pilot was conducted prior to the experiment to assess the saliency of the stimuli we used in the material. In this pilot, participants were asked to decide which of two source problems fit best to a target problem regarding general similarity and usefulness for teaching solutions of the problems. They then were asked to rate the degree of similarity and usefulness. Forty subjects participated in this experiment. As predicted, people rated stories with high superficial similarity to be satisfactorily more similar and useful for teaching. This confirmed the saliency of the material we used in the later experiments. For further explanation see Appendix A.

Material

Within each condition of the experiment, seven kinds of material were used. Four types of material are guided html-forms which were used to collect the answers of the subjects. One type of material was used as a general introduction to analogical problem solving. Two types of material were cover-stories for a class of isomorphic graph-problems (Hamilton-pathways). Due to the isomorphism between the stories, all graph problems used in the experiment were structurally identical. Therefore, only the theme of the story and superficial properties like the naming of entities as well as the order in which arcs were mentioned varied. The Solved Graph Problem and the Unsolved Graph Problem were presented in different versions for the High Agreement Group and the Low Agreement Group. The agreement between two stories in this context is considered to be high, if they are superficially similar and to be low if they are not. As mentioned before, the structural similarity did not vary and can be considered high between all stories. The agreement is the independent variable in the experiment.

Introductory Problems.

These two cover stories were used to prepare subjects for the task of analogical problem solving they had to work on later. The stories were different from the ones used to later, structurally as well as superficially. Subjects were not supposed to solve the analogical problem presented by these cover-stories, only to read them and to understand the solution. The solution was attached to the cover-stories and provided a way to solve the problem and transfer it to the other story by showing up the structural similarities between them.

Solved Graph Problem.

This cover story is used to introduce the subjects to graph problems and provide a guided solution for the problem presented in the cover story. This story called 'Boat' deals with a family that wishes to visit some locks which are located on channels connecting different cities. The locks are of architectonical value, and the family wants to visit each lock (that is, travelling each channel) exactly one time. From each city it is possible to travel to one of the adjacent cities through a channel, and on each channel a lock is located. So it is necessary and valid to visit cities more than one time, but undesirable and invalid to use a channel more then one time. The family starts at a particular city, designated as the start city. The problem posed to the reader is now to find a valid route that enables the family to travel through all channels without using one channel twice. Subjects are instructed to read the story, but not to solve it. Instead, they are asked to read and understand the solution which is attached to the cover story. In this solution they are instructed to visualize the structure of the underlying graph through using the first capital letters of the cities as nodes and connecting them with arcs in the same order as the channels are mentioned in the story. The resulting structure is a complete external representation of the graph. To solve the problem, it is now possible to try different routes from the start city over channels and cities by keeping in mind which arcs were already used. In the case of a dead end, backtracking has to occur and a decision met before has to be revised. This has to be repeated until all channels are used exactly one time. The formal solution of this problem is then defined as the concatenated string of the first letters of the cities in the order in which they were visited by travelling the channels. Subjects were instructed to re-read this solution and the procedure leading to it until they fully understand it.

For a detailed version of the problem see Appendix A. For a representation of the corresponding graph see Appendix B.

Unsolved Graph Problem.

This cover story poses a graph problem structurally identical to the one in the Solved Graph Problem. This story called 'Birthday' deals with a birthday party and the playing of a special messenger game: One person starts to write a word on a paper. The paper is then passed to another person who adds a second word, and so forth. To make things not too simple, the message passing follows a complicated protocol: The message has to be passed between all people knowing each other, but is only allowed to be passed between each acquainted pair of people exactly one time. As not all persons playing the game are acquainted, the problem is to find the order of valid connections between all players so that all friends transmit the message exactly one time. To this end, it is necessary and valid that the message is transmitted to a person more than one time, but undesirable and invalid to pass more than one time between two friends. Again, this is a graph problem, and as mentioned before it is isomorphic to the one in the second cover story. People are corresponding to cities and the property of acquaintance is corresponding to the channels. Nevertheless, it varies in the theme that is used and, depending on the version, in the agreement to the Solved Graph Problem concerning superficial similarity. Therefore, structurally the second and the third story are isomorphic in all conditions, but superficially they are highly similar in the high-agreement group and lowly similar in the low agreement condition.

For a detailed version of the problem see Appendix A. For a representation of the corresponding graph see Appendix B.

Solution Form.

This material consists of a simple html-form-field where subjects are instructed to enter their solution to the problem described in the Unsolved Graph Problem. Subjects are advised to enter the traversed nodes of the graph as the solution. This is identical to the string of first letters of the person's names between which the message circulated. An example of this syntax is first given in the Solved Graph Problem and repeated in this material to ensure the syntactical correctness of the input. Although subjects were advised to answer in this certain syntax, it was technically possible to answer in a wrong syntax or not at all. The answer was not case-sensitive. Forty-four out of several thousand possible permutations represent a valid solution to the graph-problem. These valid solutions were computed by a simple PROLOG script beforehand. All answers were stored in the database. If they belonged to one of the correct 44 solutions, the number of the solution was stored also for further assessment. Subjects got no feedback about the correctness of their input.

For a detailed version of the form see Appendix A.

Mapping Form.

This fifth type of material is a matrix where subjects could indicate correspondence of cities (see Solved Graph Problem) to persons (see Unsolved Graph Problem). This matrix is basically an html-formula to access a mapping between entities. Correspondence is indicated in the matrix through marking a field at the crossing-point of the matrix. Although subjects were advised to make a correspondence for each entity, it was technically possible to leave out some or all of them. 2 of 120 possible permutations represented a valid (which means structurally consisting) mapping. The answer was stored on the database as either correct or incorrect. Subjects got no feedback about the correctness of their input.

For a detailed version of the form see Appendix A.

Strategic Questionnaire.

This material consists of a set of 15 yes-no-questions. These qualitative questions were used to assess the problem-solving strategy subjects used to solve the problem posed in the Unsolved Graph Problem. Of these 15 questions, 5 were designed to specify certain features that would indicate transformational strategies, 5 were designed to indicate transformational strategies and 5 were fillers to indicate general strategies which do not fit into fixed categories. People were instructed to answer if they used the strategy posed in the question during the solving of the problem in Unsolved Graph Problem or not. Although subjects were advised to answer all questions, it was technically possible not to answer some or all questions. All answers were stored in the database as either answered positively or negatively.

For a detailed version of the questionnaire see Appendix A.

Demographic Questionnaire.

This last type of material consists of demographic questions about age, sex, occupation and expertise regarding graph-problems. All answers were stored in the database.

For a detailed version of the questionnaire see Appendix A.

Participants.

A total of 51 subjects participated in the experiment. Incorrect solutions were handed in by 9 (5 male, 4 female) participants. A valid solution was handed in by 42 (30 male, 12 female) participants. Seventy percent of these are students in computer-orientated programs; the others are computer-professionals. Eighty-one percent indicated they were familiar with graph-problems, the rest reported to be unfamiliar with such problems. The mean age was 25.5 years. All subjects were randomly divided to the High Agreement Version (22 correct solutions) and the Low Agreement Version (20 correct solutions) of the experiment. Mean self-report of subject's expertise on these problems on a scale of 0 to 5 was 2.67, SD = 1.44. This expertise was not significantly different between High Agreement Group (M = 2.95, SD = 1.29) and Low Agreement Group (M = 2.35, SD = 1.57, t (40) = 1.37, p = .18).

Procedure

The whole experiment was hosted on a web-server and conducted over the internet with a standard browser-interface at an internet-capable computer of the subject's own choice. The speed of presentation depended wholly on the subject's input. The participants were informed about the experiment through mailing-lists. Subscribers of these mailings lists were students in cognitive science and computer science as well as computer professionals.

Subjects were assigned randomly by the system to either to the *High-Agreement Group* or to the *Low-Agreement Group.*

Subjects in the High-Agreement Group received the Solved Graph Problem and the Unsolved Graph Problem in versions which *agreed best in surface similarity* and were rated as *highly similar* by subjects in the pilot study. Subjects in the *Low-Agreement Group* received the Solved Graph Problem and the Unsolved Graph Problem in versions which *agreed worst in surface similarity* and were rated *lowest in similarity* by subjects in the pilot study. Both groups first read an introductory text about psychological experiments and were advised to prepare a pen and a sheet of paper as well as plan in enough time and a constant internet connection.

After confirmation, the second screen with the Introductory Problems was presented. Participants were advised not to solve the problem but just to read the cover stories and the solution as long as they needed to understand them.

After confirmation, the third screen with the Solved Graph Problem was presented. Part of this material is not only the problem but also a solution and a detailed explanation of this solution. Participants were advised to read the story and the solution as long as they needed to understand them.

After confirmation, the forth screen with the Unsolved Graph Problem was presented. Participants were advised to read the story, but not to solve it.

After confirmation, the fifth screen was presented. This screen was split horizontally. In the upper part, the Solved Graph Problem, its solution, and the Unsolved Graph Problem were presented. In the lower part, people were instructed to now solve the Unsolved Graph Problem and insert its solution in certain syntax into the Solution Form. The syntax was explained in detail and consisted in the concatenated first letters of the persons between which the message was told. Time needed for the solution was measured for the whole time of presentation of this screen until subjects confirmed their solution.

After confirmation, the lower part of the screen changed, and the Mapping Form was presented. Participants were asked, to decide which city in the Solved Graph Problem corresponds best to which person in the Unsolved Graph Problem regarding their own solutions. Time needed for the mapping was measured for the whole time of presentation of this screen until subjects confirmed their mapping. Remember, that subjects still could refer to the Solved Graph Problem and the Unsolved Graph Problem in the upper part of the screen without effort, so the solution to the Solved Graph Problem was available the whole time.

After confirmation, the sixth screen with the Strategic Questionnaire and the Demographic Questionnaire was presented. People were asked to decide, if the strategies described in the questions fit to the strategy they used during the solution of the problem. After that, they were asked to fill in demographic data.

After confirmation, the last screen was presented with possibility to give feedback. At this time, the user's browser sent the collected data to the database-server together with a random number to identify the subject anonymously to prevent double postings.

Results

As described before, the two group's material differed only in superficial similarity of the stories in the Solved Graph Problem and the Unsolved Graph Problem. While the High Agreement Group received material with a high superficial similarity between these stories, the Low Agreement Group received material with a low superficial similarity.

The goal of this experiment was to encourage the use of other analogical strategies than transformational ones in problem solving by making it harder to use transformational strategies. To this end, we firstly look on the distribution of correct solutions of Unsolved Graph Problem between the groups. The distribution of correct solutions considering Agreement was not significantly different from chance, $\chi^2(1) = .19$. Thus, the variation of superficial similarity did not influence the correctness of the solutions.

As you may remember, the next task for the subjects was to construct a mapping between the cities in the Solved Graph Problem and the persons in the Unsolved Graph Problem considering structural similarity. Here we see that one of the groups is significantly faster in doing this. The High Agreement Group mapping time (n = 22, M = 730709 ms, SD = 100225 ms) was significantly faster than that of the Low Agreement Group (n = 20, M = 1710099 ms, SD = 217892 ms), t(26.80) = -4.08, $p < .001$.

This supports our hypothesis that the High Agreement Group used transformational mapping in their solution process, because there were very fast in constructing this mapping afterwards when asked. The Low Agreement Group on the other hand was comparatively much slower in constructing this mapping, indicating that they did not need to construct such a mapping in their solution and so were not able to recall it easily. As the mapping of entities between source and target problem is one of the crucial steps in transformational analogy, and the Low Agreement Group did not seem to rely on it, they apparently used a different strategy.

While we now have good evidence for the use of another strategy, we do not know which one it might have been. To assess what strategy used, the qualitative questionnaire was developed to narrow down the possible strategies In this questionnaire, five questions indicate the use of transformational strategies, 5 indicate derivational strategies and 5 were fillers to check for the validity of the responses (see Appendix A for these items) We thus were able to construct a transformational scale and a derivational scale. The results of this questionnaire were analyzed using two Mann-Whitney-U tests to evaluate whether both the mean ranked 'yes' responses for the transformational scale, and the mean ranked 'yes' responses for the derivational scale are dependent on the level of the agreement condition. For the transformational scale the High Agreement Group ($n = 22$) had a significantly higher mean rank of 24.95 than the Low Agreement Group ($n = 20$) had a mean rank of 17.70, $z = 1.96$, $p = .05$. For the derivational scale the Low Agreement Group had a significantly higher mean rank of 26.85 than the High Agreement Group (17.70), $z = 2.78$, $p = .005$. These differences in mean ranks indicate the relationship to the frequency of 'yes' responses is different for each level of the version faction. Thus, subjects from the High Agreement Group indicted significantly more use of transformational strategies than derivational strategies, and the Low Agreement Group indicated significantly more use of derivational strategies than transformational strategies.

Therefore we can conclude that subjects in both groups appeared to be using a strategy that shares more properties of the strategy expected for the version than the strategy expected for the other version. This corresponds well to the hypothesis stated before. Additionally to the described experimental setup, we also assessed which one of the 44 possible solutions to the graph problem subjects chose. Interestingly, this distribution of chosen solutions seemed randomly in both groups. This is most probably due to the very different external representations subjects chose.

During the course of this experiment we concentrated on the manipulation of surface similarity. Further experiments could continue this line of thought and modify structural similarity as well as surface similarity in varying degrees, maybe enriched with a refined formalization of similarity and complexity of problems and under consideration of omitted like retrieval and learning. As was stated before, no effects for sex and expertise have been found. In further refinements of this experiment, a better distribution of sex and expertise (see *Participants*) within the groups and tested against varying complexity of problems may lead to additional results in these areas.

Outlook

In this thesis only a very limited selection of possible effects concerning derivational analogy has been studied to remain within the boundaries of a bachelor thesis. Further studies like they were proposed in the empirical part should follow to further explore the field of derivational analogy under different conditions.

The results of the present study give some new insight about the nature of analogical problem solving as well as analogical reasoning in general. Apart from further exploring derivational approaches in its processes and constraints, connections to the transformational approach should be reevaluated. It seems obvious, that transformational and derivational approaches are not in competence but complement each other. While relational mapping will most probably stay the major strategy in analogical reasoning, new approaches may now be encouraged to model for analogical problem solving. Further on, consolidating frameworks of analogical reasoning which incorporate analogical problem solving, logical reasoning like in/deduction and computational implementations should begin to emerge in the future. Applications of derivational analogy making seem promising, especially in the field of complex tutoring systems where the strengths of this approach are met well.

Acknowledgements

I would like to briefly and cordially thank my colleagues and friends as well as my family and my lecturers for their support, their impulses and their love. Be compassionate and brave. All will be well.

References

Bhansali, S. & Harandi, M.T. (1993). Synthesis of UNIX Programs Using Derivational Analogy. *Machine Learning*, 10, 7–55.

Blumenthal, B. (1990). Strategies for automatically incorporating metaphoric attributes in interface designs. In *Third Annual Symposium on User Interface Software and Technology* (pp.66–75). New York, NY: ACM Press.

Carbonell, J.G. (1983). Derivational Analogy and Its Role in Problem Solving. In Genesereth, M.A. (Eds.), *Proceedings of the National Conference on Artificial Intelligence* (pp. 64–69). Washington, D.C.: AAAI Press

Carbonell, J. (1986). Derivational Analogy: A Theory of Reconstructive Problem Solving and Expertise Acquisition. In Michalski, R. and Carbonell, J. and Mitchell, T. (Eds.), *Machine Learning: An Artificial Intelligence Approach* (pp. 371–392). Los Altos, CA: Morgan Kaufmann.

Chalmers, D., French, R. & Hofstadter, D. (1995). High-level perception, representation, and analogy: a critique of artificial-intelligence methodology. In *Fluid concepts and creative analogies*. New York, NY: Basic Books.

Chi, M. T. H., Glaser, R. & Rees, E. (1981). Expertise in Problem Solving. In Sternberg, R. J. (Edt.), *Advances in the Psychology of Human Intelligence*. Hillsdale: Lawrence Erlbaum

Clement, C.A. & Gentner, D. (1991). Systematicity as a Selection Constraint in Analogical Mapping. *Cognitive Science*, 15(1), 89–132.

Duncker, K. (1945). On Problem Solving. *Psychological Monographs*, 58.

Evans, T.G. (1964). A Program for the Solution of a Class of Geometric-Analogy Intelligence-Test Questions. *Physical and Math. Sciences Res. Papers No. 64.*

Forbus, K. D., Gentner, D. & Law, K. (1995). MAC/FAC: A model of similarity-based retrieval. *Cognitive Science*, 19, 141–205.

Gentner, D. (1983). Structure-mapping: A Theoretical Framework for Analogy. *Cognitive Psychology*, 7, 155–170.

Gentner, D. & Toupin, C. (1986). Systematicity and surface similarity in the development of analogy. *Cognitive Science*, 10(3), 277–300.

Gentner, D. & Forbus, K.D. (1991). MAC/FAC: a model of similarity-based retrieval. *In Proceedings of the 13th Cognitive Science Conference* (pp. 504–509). Chicago: Erlbaum, Hillsdale.

Gentner, D., & Rattermann, M.J. (1991). Language and the Career of Similarity. In Gelman, S. & Byrnes, J.P. (Eds.), *Perspectives on Thought and Language: Interrelations in Development.* Cambridge University Press.

Gentner, G. & Markman, A.B. (1993). Analogy—Watershed or Waterloo? Structural Alignment and the Development of Connectionist Models of Cognition. In Hanson, S.J., Cowan, J.D. and Giles, C.L. (Eds.), *Advances in Neural Information Processing Systems* (pp. 855–862). San Mateo, CA: Morgan Kaufmann.

Gick, M.L. & Holyoak, K.J. (1983). Schema Induction and Analogical Transfer. *Cognitive Psychology*, 15, 1–38.

Goldstone, R.L. & Medin, D.L. (1994). Time Course of Comparison. *Journal of Experimental Psychology: Learning, Memory, and Cognition*, 20(1), 29–50.

Halford, G., Wilson, W., Guo, J., Gayler, R., Wiles, J. & Stewart, J. (1994). Connectionist implications for processing capacity limitations in analogies. In Holyoak, K. & Barnden, J. (Eds.), Advances in Connectionist and Neural Computation Theory, vol. 2 (pp. 363–415). Norwood, NJ: Ablex.

Harris, C.L. (1994). Backpropagation representations for the rule-analogy continuum. In Holyoak, K. & Barnden, J. (Eds.), Advances in Connectionist and Neural Computation Theory, vol. 2. Norwood, NJ: Ablex.

Hofstadter, D. & the Fluid Analogies Research Group (1995). *Fluid Concepts and*

Creative Analogies: Computer Models of the Fundamental Mechanisms of Thought. New York: Basic Books.

Holyoak, K.J. (1985). The Pragmatics of Analogical Transfer. In Bower G. H. (Edt.), *The Psychology of Learning and Motivation*. Orlando, FL: Academic Press.

Holyoak, K.J. & Thagard, P. (1989). Analogical Mapping by Constraint Satisfaction. *Cognitive Science*, 13(3), 295–355.

Holyoak, K.J. & Thagard, P. (1995). Mental Leaps: Analogy In Creative Thought. Cambridge, MT: MIT Press.

Hummel, J. & Holyoak, K. (1997). Distributed Representations of Structure: A Theory of Analogical Access and Mapping. *Psychological Review*, 104, 427–466.

Keane, M.T., Ledgeway, T. & Duff, S. (1993). Constraints on Analogical Mapping: A Comparison of Three Models.

Keane, M.T. (1996). On adaptation in analogy: Tests of pragmaticimportance and adaptablility in analogical problem solving. *The Quarterly Journal of Experimental Psychology*, 49(4), 1062–1085.

Kokinov, B. (1994). A Hybrid Model of Analogical Reasoning. In Holyoak, K. & Barnden, J. (Eds.), Advances in Connectionist and Neural Computation Theory, vol. 2 (pp. 247–318). Norwood, NJ: Ablex.

Kokinov, B. (1995). Integrating memory and reasoning in analogy-making: The AMBR model. In Gentner, D., Holyoak,K. & Kokinov, B. (Eds.), *The Analogical Mind: Perspectives from Cognitive Science* (pp. 59–124). Cambridge, MA: MIT Press.

Kolodner, J. (1993). *Case-Based Reasoning*. San Mateo, CA: Morgan Kaufmann.

Kubose, T., Holyoak, K.J., & Hummel, J.E. (1998). Strategic use of working memory in analogical mapping.

Marr, D. (1982). *Vision*, San Francisco: W.H. Freeman and Company.

Melis, E. & Whittle, J. (1997). External Analogy in Inductive Theorem Proving. In

Brewka, G. and Habel, C. and Nebel, B. (Eds.), *Proceedings of the 21st Annual German Conference on Artificial Intelligence* (pp. 111–122). Berlin: Springer.

Melis, E. & Carbonell, J.G. (1998). An Argument for Derivational Analogy. *Advances in Analogy Research*, 144–153.

Mitchell, M. (1993). *Analogy-making as Perception: A computer model.* Cambridge, MA: MIT Press.

Mostow, J. (1989). Design by Derivational Analogy: Issues in the Automated Replay of Design Plans. *Artificial Intelligence*, 40(1–2), 119–184.

Nokes, T., Ohlsson, S., & Corrigan-Halpern, A. (2002). Learning by analogy vs learning by instruction: Same knowledge, different representations. P*roceedings of the 9th Annual ACT-R Workshop.*

Ross, B.H. (1989). Distinguishing Types of Superficial Similarities: Different Effects on the Access and Use of Earlier Problems. *Journal of Experimental Psychology: Learning, Memory, and Cognition*, 15, 456–468.

Reed, S. K., Ackinclose, C. C., & Voss, A. A. (1990). Selecting analogous problems: similarity versus inclusiveness. *Memory & Cognition*, 18, 83–98.

Schmid, U. & Wysotzki, F. (1998). Induction of Recursive Program Schemes. In Claire Nédellec and Céline Rouveirol (Eds.), *Proceedings of the 10th European Conference on Machine Learning* (pp. 21–23). Berlin: Springer.

Schmid, U., Wirth, J. & Polkehn, K. (2003). A Closer Look at Structural Similarity in Analogical Transfer. Cognitive Science Quarterly, 3, 57–89.

Skorstad, J., Gentner, D., & Medin, D. (1988). *Abstraction processes during concept learning: A structural view.* Proceedings of the Tenth Annual Conference of the Cognitive Science Society, 419–425.

Spellman, B.A. & Holyoak, K.J. (1996). Pragmatics in analogical mapping. *Cognitive Psychology*, 31, 307–366.

Veloso, M. & Carbonell, J. (1993). Derivational Analogy in PRODIGY: Automating Case Acquisition, Storage, and Utilization. *Machine Learning*, 10(3), 249–278.

Veloso, M. & Carbonell, J. (1994). Case-Based Reasoning in PRODIGY. In Michalski, R. and Tecuci, G. (Eds.), *Machine Learning: A Multistrategy Approach* (pp. 523–548). Morgan Kaufmann.

Veloso, M. (1994). Planning and learning by analogical reasoning. In *Lecture Notes in Artificial Intelligence and Lecture Notes in Computer Science vol. 886.* New York, NY: Springer

Wilson, W., Halford, G. Gray, B. & Phillips, S. (2001). The STAR-2 Model for Mapping Hierarchically Structured Analogs. In Gentner, D. Holyoak, K. & Kokinov, B. (Eds.), The Analogical Mind: Perspectives from Cognitive Science. Cambridge, MA: MIT Press.

Winston, P.H. (1979). Learning and Reasoning by Analogy. *Journal of the Association for Computing Machinery*, 23, 689–703.

Appendix A: Material

Pilot Study

Introduction.

In this pilot prior to the final experiment we assessed information about the saliency of the names and arc-order we manipulated and used as independent variables in the final

experiment. Each material of the pilot consisted in three stories each covering a graph problem. All stories were isomorphic, that is, the problems used in all stories were structurally identical.

Subjects were advised to select which of the stories two (Birthday 1) or three (Birthday 2) in each material was most similar and useful regarding story one (Boat). Subjects were not advised to actually solve the problems.

Each story had four possible modifications: without surface similarity (0), with surface similarity due to same names (n), with surface similarity due to same order of arcs (a), with surface similarity due to same names and same order of arcs (na). Permutations of these possible modifications of stories resulted into five different versions (B, C, E, F, G) of material.

The results of the pilot were that only the difference between *0* and *na* was satisfactory salient (*Version C*). Thus, this version was used in the final experiment. Detailed versions of this pilot are following.

The following versions were used in the pilot:

Version B:

Story one (Boat): a, Story two (Birthday 1): a, Story three (Birthday 2): 0

Version C:

Story one (Boat): na, Story two (Birthday 1): na, Story three (Birthday 2): 0

Version E:

Story one (Boat): a, Story two (Birthday 1): a, Story three (Birthday 2): n

Version F:

Story one (Boat): na, Story two (Birthday 1): n, Story three (Birthday 2): na

Version G:

Story one (Boat): na, Story two (Birthday 1): na, Story three (Birthday 2): a

Pilot: Detailed Version B.

In the following you will be presented some thought problems and are asked for a short evaluation.

Imagine you are a teacher and your students have to solve the first problem (calles "Boat"). You want to select a suitable other problem which should help the students to solve the boat problem. Please read all stories carefully and answer the questions. You are not supposed to really solve the problems.

Boat The Johnsons are planning a river boat tour for their summer holiday visiting five cities: Schwetzingen, Marbach, Blaubeuren, Ludwigsburg and Ulm. The area they will visit is famous for its ancient river locks and they are looking forward to this experience. The Johnsons have heard that each of the eight river locks in this area has its own architectural value and technical concept, so they want to make sure to cross via each of the eight locks. But, as there is a fairly high toll for each lock, they also want to make sure not to travel through any lock more than once. The eight locks are located between the following pairs of cities: Ulm is connected to Blaubeuren and to Ludwigsburg via locks. These two cities are connected with all other cities via locks – that is, Blaubeuren is connected to Ulm, Ludwigsburg, Marbach and Schwetzingen; Ludwigsburg is connected with Ulm, Blaubeuren, Marbach and Schwetzingen. Marbach and Schwetzingen are connected with all cities except Ulm – that is, Marbach is connected with Blaubeuren, Ludwigsburg and Schwetzingen; Schwetzingen is connected with Blaubeuren, Ludwigsburg and Marbach. From Schwetzingen, they wish to travel along a route that will enable them to go though each of the eight locks exactly once. Note that their desire to travel through every lock once necessarily means that they will visit some of the cities more than once. Plan a route for the Johnsons so that they travel over through every lock exactly once and visit each city as many times as necessary. Feel free to use short notations for the people, as "U" for "Ulm" and so on.

Birthday 1 Five people attended a birthday party: Susan, Eric, Bill, Richard, and Mary. During the course of the evening they played different games. One game they played was a "messenger game" where one person started to write a word on a paper. The paper was then passed to another person who added a second word, and so on. To make things not too simple, the message passing followed a complicated protocol: The message had to be passed between all people knowing each other, but was only be allowed to be passed between each acquainted pair of people exactly once. The following pairs of people know each other: Mary knows two people – Bill and Richard. Bill and Richard both know all other people – that is, Bill knows Mary, Richard, Susan, and Eric; Richard knows Mary, Bill, Susan, and Eric. Eric and Susan know all people except Mary – that is, Susan knows Bill, Richard, and Eric; Eric knows Bill, Richard and Susan. When Susan was the person writing the first word, who is a possible person to write the last word? Give the order in which the message was passed person-to-person. Feel free to use short notations for the people, as "S" for "Susan" and so on.

Birthday 2 Five people attended a birthday party: Susan, Eric, Bill, Richard, and Mary. During the course of the evening they played different games. One game they played was a "messenger game" where one person started to write a word on a paper. The paper was then passed to another person who added a second word, and so on. To make things not too simple, the message passing followed a complicated protocol: The message had to be passed between all people knowing each other, but was only be allowed to be passed between each acquainted pair of people exactly once. The following pairs and triples of people know each other: Susan, Eric and Bill all know each other; Richard and Susan know each other; Bill, Mary, and Richard all know each other; Eric and Richard know each other. When Susan was the person writing the first word, who is a possible person to write the last word? Give the order in which the message was passed person-to-person. Feel free to use short notations for the people, as "S" for "Susan" and so on.

Read the stories above carefully. Assume that you do not know how to solve the boat problem. The solution of which of the other two problems would you find better suited to use for an explanation to students to help them solve the boat problem? Please answer the following questions. After you have done that, press the button far below to submit your answers.

1. Which problem do you find more useful for solving the Boat Problem?

Birthday 1 Birthday 2 (please circle one)

2. How useful? (please rate on the following scale, circle one)

Not Useful 1 2 3 4 5 Very Useful

3. Which problem do you find more similar to the Boat Problem?

Birthday 1 Birthday 2 (please circle one)

4. How Similar? (please rate on the following scale, circle one)

Not Similar 1 2 3 4 5 Very Similar

Thank-you for participating in the thought problem evaluation task. The responses are kept in strictest confidence. Your name will in no way be associated with the data or data analysis. Please do not discuss the task with anyone who might participate in this task later on, as this adversely affects the data.

Please answer two final questions by circling one response

1. Sex: Male Female

2. Do you have experience with solving problems of this type?

Yes No

If "yes" How would you rate your expertise with this type of problem

(please rate on the following scale)

Novice 1 2 3 4 5 Expert

Pilot: Detailed Version C.

Introduction omitted - see version B for details.

Boat The Johnsons are planning a river boat tour for their summer holiday visiting five cities: Cannenbach, Frankheim, Neustadt, Markburg and Behringen. The area they will visit is famous for its ancient river locks and they are looking forward to this experience. The Johnsons have heard that each of the eight river locks in this area has its own architectural value and technical concept, so they want to make sure to cross via each of the eight locks. But, as there is a fairly high toll for each lock, they also want to make sure not to travel through any lock more than once. The eight locks are located between the following pairs of cities: Behringen is connected to Neustadt and to Markburg via locks. These two cities are connected with all other cities via locks – that is, Neustadt is connected to Behringen, Markburg, Frankheim and Cannenbach; Markburg is connected with Behringen, Neustadt, Frankheim and Cannenbach. Frankheim and Cannenbach are connected with all cities except Behringen – that is, Frankheim is connected with Neustadt, Markburg and Cannenbach; Cannenbach is connected with Neustadt, Markburg and Frankheim. From Cannenbach, they wish to travel along a route that will enable them to go though each of the eight locks exactly once. Note that their desire to travel through every lock once necessarily means that they will visit some of the cities more than once. Plan a route for the Johnsons so that they travel over through every lock exactly once and visit each city as many times as necessary. Feel free to use short notations for the people, as "N" for "Neustadt" and so on.

Birthday 1 Five people attended a birthday party: Carry, Fred, Ned, Mike, and Babs. During the course of the evening they played different games. One game they played was a "messenger game" where one person started to write a word on a paper. The paper was then passed to another person who added a second word, and so on. To make things not too simple, the message passing followed a complicated protocol: The message

had to be passed between all people knowing each other, but was only be allowed to be passed between each acquainted pair of people exactly once. The following pairs of people know each other: Babs knows two people – Ned and Mike. Ned and Mike both know all other people – that is, Ned knows Babs, Mike, Fred, and Carry; Mike knows Babs, Ned, Fred, and Carry. Carry and Fred know all people except Babs – that is, Fred knows Ned, Mike, and Carry; Carry knows Ned, Mike and Fred. When Carry was the person writing the first word, who is a possible person to write the last word? Give the order in which the message was passed person-to-person. Feel free to use short notations for the people, as "C" for "Carry" and so on.

Birthday 2 Five people attended a birthday party: Susan, Eric, Bill, Richard, and Mary. During the course of the evening they played different games. One game they played was a "messenger game" where one person started to write a word on a paper. The paper was then passed to another person who added a second word, and so on. To make things not too simple, the message passing followed a complicated protocol: The message had to be passed between all people knowing each other, but was only be allowed to be passed between each acquainted pair of people exactly once. The following pairs and triples of people know each other: Susan, Eric and Bill all know each other; Richard and Susan know each other; Bill, Mary, and Richard all know each other; Eric and Richard know each other. When Susan was the person writing the first word, who is a possible person to write the last word? Give the order in which the message was passed person-to-person. Feel free to use short notations for the people, as "S" for "Susan" and so on.

Questionnaires omitted - see version B for details.

Pilot: Detailed Version E.

Introduction omitted - see version B for details.

Boat The Johnsons are planning a river boat tour for their summer holiday visiting five cities: Schwetzingen, Marbach, Blaubeuren, Ludwigsburg and Ulm. The area they will visit is famous for its ancient river locks and they are looking forward to this experience. The Johnsons have heard that each of the eight river locks in this area has its own architectural value and technical concept, so they want to make sure to cross via each of the eight locks. But, as there is a fairly high toll for each lock, they also want to make sure not to travel through any lock more than once. The eight locks are located between the following pairs of cities: Ulm is connected to Blaubeuren and to Ludwigsburg via locks. These two cities are connected with all other cities via locks – that is, Blaubeuren is connected to Ulm, Ludwigsburg, Marbach and Schwetzingen; Ludwigsburg is connected with Ulm, Blaubeuren, Marbach and Schwetzingen. Marbach and Schwetzingen are connected with all cities except Ulm – that is, Marbach is connected with Blaubeuren, Ludwigsburg and Schwetzingen; Schwetzingen is connected with Blaubeuren, Ludwigsburg and Marbach. From Schwetzingen, they wish to travel along a route that will enable them to go though each of the eight locks exactly once. Note that their desire to travel through every lock once necessarily means that they will visit some of the cities more than once. Plan a route for the Johnsons so that they travel over through every lock exactly once and visit each city as many times as necessary. Feel free to use short notations for the people, as "U" for "Ulm" and so on.

Birthday 1 Five people attended a birthday party: Susan, Eric, Bill, Richard, and Mary. During the course of the evening they played different games. One game they played was a "messenger game" where one person started to write a word on a paper. The paper was then passed to another person who added a second word, and so on. To make things not too simple, the message passing followed a complicated protocol: The message had to be passed between all people knowing each other, but was only be allowed to be passed between each acquainted pair of people exactly once. The following pairs of people know each other: Mary knows two people – Bill and Richard. Bill and Richard both know all other people – that is, Bill knows Mary, Richard, Susan, and Eric; Richard knows Mary, Bill, Susan, and Eric. Eric and Susan know all people except Mary – that is, Susan knows Bill, Richard, and Eric; Eric knows Bill, Richard and Susan. When Susan was the person writing the first word, who is a possible person to write the last word? Give the order in which the message was passed person-to-person. Feel free to use short notations for the people, as "S" for "Susan" and so on.

Birthday 2 Five people attended a birthday party: Carry, Fred, Ned, Mike, and Babs. During the course of the evening they played different games. One game they played was a "messenger game" where one person started to write a word on a paper. The paper was then passed to another person who added a second word, and so on. To make things not too simple, the message passing followed a complicated protocol: The message had to be passed between all people knowing each other, but was only be allowed to be passed between each acquainted pair of people exactly once. The following pairs and triples of people know each other: Fred, Carry, and Ned all know each other; Mike and Fred know each other; Ned, Babs, and Mike all know each other; Carry and Mike know each other. When Carry was the person writing the first word, who is a possible person to write the last word? Give the order in which the message was passed person-to-person. Feel free to use short notations for the people, as "C" for "Carry" and so on.

Questionnaires omitted - see version B for details.

Pilot: Detailed Version F.

Introduction omitted - see version B for details.

Boat The Johnsons are planning a river boat tour for their summer holiday visiting five cities: Cannenbach, Frankheim, Neustadt, Markburg and Behringen. The area they will visit is famous for its ancient river locks and they are looking forward to this experience. The Johnsons have heard that each of the eight river locks in this area has its own architectural value and technical concept, so they want to make sure to cross via each of the eight locks. But, as there is a fairly high toll for each lock, they also want to make sure not to travel through any lock more than once. The eight locks are located between the following pairs of cities: Behringen is connected to Neustadt and to Markburg via locks. These two cities are connected with all other cities via locks – that is, Neustadt is connected to Behringen, Markburg, Frankheim and Cannenbach; Markburg is connected with Behringen, Neustadt, Frankheim and Cannenbach. Frankheim and Cannenbach are connected with all cities except Behringen – that is, Frankheim is connected with Neustadt, Markburg and Cannenbach; Cannenbach is connected with Neustadt, Markburg and Frankheim. From Cannenbach, they wish to travel along a route that will enable them to go though each of the eight locks exactly once. Note that their desire to travel through every lock once necessarily means that they will visit some of the cities more than once. Plan a route for the Johnsons so that they travel over through every lock exactly once and visit each city as many times as necessary. Feel free to use short notations for the people, as "N" for "Neustadt" and so on.

Birthday 1 Five people attended a birthday party: Carry, Fred, Ned, Mike, and Babs. During the course of the evening they played different games. One game they played was a "messenger game" where one person started to write a word on a paper. The paper was then passed to another person who added a second word, and so on. To make things not too simple, the message passing followed a complicated protocol: The message

had to be passed between all people knowing each other, but was only be allowed to be passed between each acquainted pair of people exactly once. The following pairs and triples of people know each other: Fred, Carry, and Ned all know each other; Mike and Fred know each other; Ned, Babs, and Mike all know each other; Carry and Mike know each other. When Carry was the person writing the first word, who is a possible person to write the last word? Give the order in which the message was passed person-to-person. Feel free to use short notations for the people, as "C" for "Carry" and so on.

Birthday 2 Five people attended a birthday party: Carry, Fred, Ned, Mike, and Babs. During the course of the evening they played different games. One game they played was a "messenger game" where one person started to write a word on a paper. The paper was then passed to another person who added a second word, and so on. To make things not too simple, the message passing followed a complicated protocol: The message had to be passed between all people knowing each other, but was only be allowed to be passed between each acquainted pair of people exactly once. The following pairs of people know each other: Babs knows two people – Ned and Mike. Ned and Mike both know all other people – that is, Ned knows Babs, Mike, Fred, and Carry; Mike knows Babs, Ned, Fred, and Carry. Carry and Fred know all people except Babs – that is, Fred knows Ned, Mike, and Carry; Carry knows Ned, Mike and Fred. When Carry was the person writing the first word, who is a possible person to write the last word? Give the order in which the message was passed person-to-person. Feel free to use short notations for the people, as "C" for "Carry" and so on.

Questionnaires omitted - see version B for details.

Pilot: Detailed Version G.

Introduction omitted - see version B for details.

Boat The Johnsons are planning a river boat tour for their summer holiday visiting five cities: Cannenbach, Frankheim, Neustadt, Markburg and Behringen. The area they will visit is famous for its ancient river locks and they are looking forward to this experience. The Johnsons have heard that each of the eight river locks in this area has its own architectural value and technical concept, so they want to make sure to cross via each of the eight locks. But, as there is a fairly high toll for each lock, they also want to make sure not to travel through any lock more than once. The eight locks are located between the following pairs of cities: Behringen is connected to Neustadt and to Markburg via locks. These two cities are connected with all other cities via locks – that is, Neustadt is connected to Behringen, Markburg, Frankheim and Cannenbach; Markburg is connected with Behringen, Neustadt, Frankheim and Cannenbach. Frankheim and Cannenbach are connected with all cities except Behringen – that is, Frankheim is connected with Neustadt, Markburg and Cannenbach; Cannenbach is connected with Neustadt, Markburg and Frankheim. From Cannenbach, they wish to travel along a route that will enable them to go though each of the eight locks exactly once. Note that their desire to travel through every lock once necessarily means that they will visit some of the cities more than once. Plan a route for the Johnsons so that they travel over through every lock exactly once and visit each city as many times as necessary. Feel free to use short notations for the people, as "N" for "Neustadt" and so on.

Birthday 1 Five people attended a birthday party: Carry, Fred, Ned, Mike, and Babs. During the course of the evening they played different games. One game they played was a "messenger game" where one person started to write a word on a paper. The paper was then passed to another person who added a second word, and so on. To make things not too simple, the message passing followed a complicated protocol: The message

had to be passed between all people knowing each other, but was only be allowed to be passed between each acquainted pair of people exactly once. The following pairs of people know each other: Babs knows two people – Ned and Mike. Ned and Mike both know all other people – that is, Ned knows Babs, Mike, Fred, and Carry; Mike knows Babs, Ned, Fred, and Carry. Carry and Fred know all people except Babs – that is, Fred knows Ned, Mike, and Carry; Carry knows Ned, Mike and Fred. When Carry was the person writing the first word, who is a possible person to write the last word? Give the order in which the message was passed person-to-person. Feel free to use short notations for the people, as "C" for "Carry" and so on.

Birthday 2 Five people attended a birthday party: Susan, Eric, Bill, Richard, and Mary. During the course of the evening they played different games. One game they played was a "messenger game" where one person started to write a word on a paper. The paper was then passed to another person who added a second word, and so on. To make things not too simple, the message passing followed a complicated protocol: The message had to be passed between all people knowing each other, but was only be allowed to be passed between each acquainted pair of people exactly once. The following pairs of people know each other: Mary knows two people – Bill and Richard. Bill and Richard both know all other people – that is, Bill knows Mary, Richard, Susan, and Eric; Richard knows Mary, Bill, Susan, and Eric. Eric and Susan know all people except Mary – that is, Susan knows Bill, Richard, and Eric; Eric knows Bill, Richard and Susan. When Susan was the person writing the first word, who is a possible person to write the last word? Give the order in which the message was passed person-to-person. Feel free to use short notations for the people, as "S" for "Susan" and so on.

Questionnaires omitted - see version B for details.

Final Experiment

Here you find the detailed versions of the material used in the final experiment. For further explanation refer to the Empirical Part. Some graphics are omitted for the sake of shortness. For visualizations of the graphs pleaser refer to Appendix B

Introductory Problems.

Instructions Below these instructions you will see two problems, called 'Fortress' and 'Tumor'. Please read these problems carefully. Then proceed by reading the solution that is printed below the two problems. When you have understood the solution, please click on 'Next Page' on the bottom of this page to proceed to the next part of the experiment.

Fortress A small country was controlled by a dictator. The dictator ruled the country from a strong fortress. The fortress was situated in the middle of the country, surrounded by farms and villages. Many roads radiated outward from the fortress like spokes on a wheel. A general arose who raised a large army and vowed to capture the fortress and free the country of the dictator. The general knew that if his entire army could attack the fortress at once it could be captured. The general's troops were gathered at the head of one of the roads leading to the fortress, ready to attack. However, a spy brought the general a disturbing report. The ruthless dictator had planted mines on each of the roads. The mines were set so that small bodies of men could pass over them safely because the dictator needed to be able to move troops and workers to and from the fortress. However, any large force would detonate the mines. Not only would this blow up the road and render it impassible, but the dictator would then destroy many villages in retaliation. It therefore seemed impossible to mount a full-scale direct attack on the fortress.

Tumour Suppose you are a doctor faced with a patient who has a malignant tumour in his stomach. It is impossible to operate on the patient, but unless the tumour is destroyed the patient will die. There is a kind of ray that can be used to destroy the tumour. If the rays reach the tumour all at once with sufficiently high intensity, the tumour will be destroyed. Unfortunately, at this intensity the healthy tissue that the rays pass through will also be destroyed. At lower intensities the rays are harmless to the healthy tissue, but they will not affect the tumour either. What type of procedure might be used to destroy the tumour with rays without destroying the healthy tissue?

Solution Although the problems look very different, their solutions are actually quite similar and analogous if you think about it carefully. The solutions to both of the problems are based on the same strategy: Divide and Conquer. The general could divide his army up into small groups and dispatch each group to the head of a different road. When all is ready he gives the signal, and each group marches down a different road. Each group continues down its road to the fortress, so that the entire army finally arrives together at the fortress at the same time. In this way, the general is able to capture the fortress and thus overthrow the dictator. In the other problem, the ray can be divided into several low-intensity rays, no one of which will destroy the healthy tissue. If these several rays are positioned at different locations around the body and focused on the tumour, their effect will combine, thus being strong enough to destroy the tumour.

Problems in High-Agreement Version.

Solved Problem: Boat The Johnsons are planning a river boat tour for their summer holiday visiting five cities: Cannenbach, Frankheim, Neustadt, Markburg and Behringen. The area they will visit is famous for its ancient river locks and they are looking forward to this experience. The Johnsons have heard that each of the eight river locks in this area has its own architectural value and technical concept, so they want to make sure to cross via each of the eight locks. But, as there is a fairly high toll for each lock, they also want to make sure not to travel through any lock more than once. The eight locks are located between the following pairs of cities: Behringen is connected to Neustadt and to Markburg via locks. These two cities are connected with all other cities via locks – that is, Neustadt is connected to Behringen, Markburg, Frankheim and Cannenbach; Markburg is connected with Behringen, Neustadt, Frankheim and Cannenbach. Frankheim and Cannenbach are connected with all cities except Behringen – that is, Frankheim is connected with Neustadt, Markburg and Cannenbach; Cannenbach is connected with Neustadt, Markburg and Frankheim. From Cannenbach, they wish to travel along a route that will enable them to go though each of the eight locks exactly once. Note that their desire to travel through every lock once necessarily means that they will visit some of the cities more than once. Plan a route for the Johnsons so that they travel over through every lock exactly once and visit each city as many times as necessary. Feel free to use short notations for the people, as "N" for "Neustadt" and so on.

Solved Problem: Solution to Boat Remark: In the original version, this solution is supplied with detailed step-by-step pictures of the graph. For solving this problem, it is very helpful to visualize it with the help of a pen and a sheet of paper. First note each city mentioned in the problem by its first letter. For example, to represent the city 'Cannenbach' you draw a capital 'C', for the city 'Frankheim' a capital 'F' and so forth. When you did this with all of the cities mentioned in the problem, your sheet will look similar to the one displayed below. Do not worry if it does not exactly like yours, because there are several correct possbibilities.

After you have represented the cities on your sheet of paper, let us proceed with the river locks between the cities. Each river lock lies on a channel which connects two cities. To represent this channel, and with it also the lock, draw a line between two cities on your sheet of paper. For example, if the text says that a lock is located between the cities 'Behringen' and 'Neustadt', draw a line between the capital letters 'B' and 'N' on your sheet of paper. After you have done this with all the connections mentioned in the problem, your sheet will look similar to this one.

Now we have to plan a trip for the Johnsons, as it is mentioned in the problem. The Johnsons want to travel through every lock exactly one time. If we have our representation on the sheet of paper in mind, this means that we have to find a route, where every line is used exactly one time, starting with the city Cannenbach. Each time we cross a city, we note its capital letter somewhere to keep track of our position. Of course we also have to keep in mind, which lines we already used, because we are not allowed to use them twice. At some place it might be, that our partial solution can not be completed, because we made an error and end in an city without unused lines attached one to it. Then we have to go back and make another decision at an earlier point of the solving process. Let's have a closer look at one of the possible correct solutions. We start in Cannenbach, as mentioned in the problem. From there, we travel over the line to Neustadt, further to Behringen, Markburg, again Neustadt, then Frankheim, again Cannenbach, Markburg and finally again Frankheim. If the Johnsons follow this way, they will see each lock (that is, use each line) but no lock twice. Of course they travel through some cities twice, but that is not forbidden by the problem. So one possible solution to the problem would be 'CNBMNFCMF' - the order of visited cities the Johnson's could travel, represented by the first capital letters of the cities attached one to an another.

Unsolved Problem: Birthday Five people attended a birthday party: Carry, Fred, Ned, Mike, and Babs. During the course of the evening they played different games. One game they played was a "messenger game" where one person started to write a word on a paper. The paper was then passed to another person who added a second word, and so forth. To make things not too simple, the message passing followed a complicated protocol: The message had to be passed between all people knowing each other, but was only allowed to be passed between each acquainted pair of people exactly once. The following pairs of people know each other: Babs knows two people – Ned and Mike. Ned and Mike both know all other people – that is, Ned knows Babs, Mike, Fred, and Carry; Mike knows Babs, Ned, Fred, and Carry. Carry and Fred know all people except Babs – that is, Fred knows Ned, Mike, and Carry; Carry knows Ned, Mike and Fred. Carry was the person writing the first word. Give the order in which the message was passed person-to-person. Feel free to use short notations for the people, as "C" for "Carry" and so forth.

Problems in Low-Agreement Version.

Solved Problem: Boat The Johnsons are planning a river boat tour for their summer holiday visiting five cities: Schwetzingen, Marbach, Blaubeuren, Ludwigsburg and Ulm. The area they will visit is famous for its ancient river locks and they are looking forward to this experience. The Johnsons have heard that each of the eight river locks in this area has its own architectural value and technical concept, so they want to make sure to cross via each of the eight locks. But, as there is a fairly high toll for each lock, they also want to make sure not to travel through any lock more than once. The eight locks are located between the following pairs of cities: Schwetzingen and Blaubeuren, Schwetzingen and Ludwigsburg, Schwetzingen and Marbach, Marbach and Ludwigsburg, Marbach and Blaubeuren, Blaubeuren and Ludwigsburg, Blaubeuren and Ulm, Ulm and Ludwigsburg. The Johnsons plan to start their trip in Schwetzingen. From Schwetzingen, they wish to travel along a route that will enable them to go though each of the eight locks exactly once. Note that their desire to travel through every lock once necessarily means that they will visit some of the cities more than once. Plan a route for the Johnsons so that they travel over through every lock exactly once and visit each city as many times as necessary. Feel free to use short notations for the people, as "B" for "Blaubeuren" and so on.

Solved Problem: Solution to Boat Remark: In the original version, this solution is supplied with detailed step-by-step pictures of the graph. For solving this problem, it is very helpful to visualize it with the help of a pen and a sheet of paper. First note each city mentioned in the problem by its first letter. For example, to represent the city 'Schwetzingen' you draw a capital 'S', for the city 'Blaubeuren' a capital 'B' and so forth. When you did this with all of the cities mentioned in the problem, your sheet will look similar to the one displayed below. Do not worry if it does not exactly like yours, because there are several correct possibilities.

After you have represented the cities on your sheet of paper, let us proceed with the river locks between the cities. Each river lock lies on a channel which connects two cities. To represent this channel, and with it also the lock, draw a line between two cities on your sheet of paper. For example, if the text says that a lock is located between the cities 'Schwetzingen' and 'Blaubeuren', draw a line between the capital letters 'S' and 'B' on your sheet of paper. After you have done this with all the connections mentioned in the problem, your sheet will look similar to this one.

Now we have to plan a trip for the Johnsons, as it is mentioned in the problem. The Johnsons want to travel through every lock exactly one time. If we have our representation on the sheet of paper in mind, this means that we have to find a route, where every line is used exactly one time, starting with the city Schwetzingen. Each time we cross a city, we note its capital letter somewhere to keep track of our position. Of course we also have to keep in mind, which lines we already used, because we are not allowed to use them twice. At some place it might be, that our partial solution can not be completed, because we made an error and end in an city without unused lines attached one to it. Then we have to go back and make another decision at an earlier point of the solving process. Let's have a closer look at one of the possible correct solutions. We start in Schwetzingen, as mentioned in the problem. From there, we travel over the line to Blaubeuren, further to Ulm, Ludwigsburg, again Blaubeuren, then Marbach, again Schwetzingen, Ludwigsburg and finally again Marbach. If the Johnsons follow this way, they will see each lock (that is, use each line) but no lock twice. Of course they travel through some cities twice, but that is not forbidden by the problem. So one possible solution to the problem would be 'SBULBMSLM' - the order of visited cities the Johnson's could travel, represented by the first capital letters of the cities attached one to an another.

Unsolved Problem: Birthday Five people attended a birthday party: Richard, Eric, Mary, Susan, and Bill. During the course of the evening they played different games. One game they played was a "messenger game" where one person started to write a word on a paper. The paper was then passed to another person who added a second word, and so on. To make things not too simple, the message passing followed a complicated protocol: The message had to be passed between all people knowing each other, but was only allowed to be passed between each acquainted pair of people exactly once. The following pairs and triples of people know each other: Susan, Eric and Bill all know each other; Richard and Susan know each other; Bill, Mary, and Richard all know each other; Eric and Richard know each other. Susan was the person writing the first word. Give the order in which the message was passed person-to-person. Feel free to use short notations for the people, as "S" for "Susan" and so forth.

Solution Form.

Instructions In the frame above you will find the 'Boat' problem with solution and the new 'Birthday' problem that you already have read on the last page. You may have to scroll a bit to see everything. Your task now is to solve the 'Birthday' problem. You may use the solution to the 'Boat' problem if you like. Please solve the 'Birthday' problem and insert your solution into the box below this instruction. The form of the solution has to be analogous to the form of the solution of the 'Boat' problem. Of course there are no cities in the 'Birthday' problem, so use the first capital letters of the names of the friends between which the message was passed for your solution. This should look remotely similar to this example, but of course with different letters: [ABCBCA] When you inserted your solution, please click on the 'Submit Answers' button on the bottom of this page. Insert your solution for the 'Birthday' problem here: [ ] [Submit Anwers]

Mapping Form High Agreement Version.

Instructions Below this instruction you will see a small matrix. Please select, which city of the 'Boat' problem (on the left of the matrix) corresponds best to which person of the 'Birthday' problem. For doing this, you may have to compare the solutions of both problems and then click on the corresponding crossing-point in the matrix. When you have mapped each city to a name in this way, please click on 'Submit Answers' on the bottom of this page.

Mapping	Ned	Mike	Carry	Babs	Fred
Cannenbach	O	O	O	O	O
Frankheim	O	O	O	O	O
Neustadt	O	O	O	O	O
Markburg	O	O	O	O	O
Behringen	O	O	O	O	O

Mapping Form Low Agreement Version.

Instructions Below this instruction you will see a small matrix. Please select, which city of the 'Boat' problem (on the left of the matrix) corresponds best to which person of the 'Birthday' problem. For doing this, you may have to compare the solutions of both problems and then click on the corresponding crossing-point in the matrix. When you have mapped each city to a name in this way, please click on 'Submit Answers' on the bottom of this page.

Mapping	Richard	Bill	Susan	Mary	Eric
Schwetzingen	O	O	O	O	O
Marbach	O	O	O	O	O
Blaubeuren	O	O	O	O	O
Ludwigsburg	O	O	O	O	O
Ulm	O	O	O	O	O

Strategic Questionnaire.

Remark: Questions designed to assess transformational strategies: 01, 04, 08, 10,S11 Questions designed to assess derivational strategies: 02, 03,06, 12, 13, all other questions were dummy-fillers.

Instructions Now you are nearly done. Please answer some final questions for us. After you answered all the questions, please click on 'Submit Answers' on the bottom of this page. Describe how you solved the 'Birthday' problem. Check 'yes' if a sentence approximately fits your strategy and 'no' if it does not.

01 Yes O No O It was simple to use the 'Boat' solution to solve the 'Birthday' problem by replacing the names of the towns with the names of the people.

02 Yes O No O The 'Boat' problem and the 'Birthday' problem seemed similar but I could not figure out how the solutions were related

03 Yes O No O I remembered how I drew the graph with help of the solution of the 'Boat' problem. Through this I found the travel relations between the towns and used the same procedure to solve the 'Birthday' problem.

04 Yes O No O I did not go through the steps I used in the 'Boat' problem to solve the 'Birthday' problem because I just replaced the names in the boat problem with the names in the birthday problem.

05 Yes O No O I could not use the boat problem solution to solve the 'Birthday' problem, but when I finished with the 'Birthday' solution I realized how they are the same.

06 Yes O No O I used the travel strategy I remembered from the boat problem to solve the 'Birthday' problem, but I made a few (or one) mistakes as I went along and had to do some of the message route over again.

07 Yes O No O The 'Boat' problem and the 'Birthday' problem seemed too dissimilar for me to use the 'Boat' solution to solve the birthday problem.

08 Yes O No O I did not have to try to remember any of the steps from the 'Boat' problem solution.

09 Yes O No O I could not solve the 'Birthday' problem.

10 Yes O No O The solution to the 'Birthday' problem was obvious from near the beginning of the of trying to solve it because it is just like the 'Boat' problem.

11 Yes O No O I made a correspondence between the parts of the 'Boat' problem and the parts of the 'Birthday' problem and then wrote the solution to the 'Birthday' problem.

12 Yes O No O I did not make much of a link between the town names in the 'Boat' problem and the names of the people in the 'Birthday' problem. I just used the same route of the 'Boat' problem solution to solve the 'Birthday' problem.

13 Yes O No O I broke the 'Birthday' problem into smaller pieces and used the same strategies that were used in the 'Boat' problem solution.

14 Yes O No O I could not solve the 'Birthday' problem because I got stuck in the same way as I did trying to understand the solution to the 'Boat' problem.

15 Yes O No O I did not really use the 'Boat' problem solution because I already knew the general principle of how to solve problems of this kind.

Demographic Questionnaire.

Your sex:

male O female O

Your age: [ ]

Your occupation: [ ]

Do you have experience with solving problems of this type?

Yes O No O

If 'yes' How would you rate your expertise with this type of problem?

Novice O O O O O Expert

[Submit Anwers]

Appendix B: Solutions

Visualizations

Remark: The graph always at the bottom left. Note, that this visualizations are arbitrary and that many other visualizations of this graph are possible and valid.

High-Agreement Version Solved Problem.

```
   B
  / \
 /   \
M-----N
| \ / |
|  x  |
| / \ |
C-----F
```

High-Agreement Version Unsolved Problem.

```
   B
  / \
 /   \
M-----N
| \ / |
|  x  |
| / \ |
C-----F
```

Low-Agreement Version Solved Problem.

```
   U
  / \
 /   \
L-----B
| \ / |
|  x  |
| / \ |
S-----M
```

Low-Agreement Version Unsolved Problem.

```
   M
  / \
 /   \
B-----R
| \ / |
|  x  |
| / \ |
S-----E
```

Valid Paths through the Graph

High-Agreement Version.

CFMNBMCNF, CFMNCMBNF, CFMBNCMNF, CFMBNMCNF, CFMCNBMNF, CFMCNMBNF, CFNBMNCMF, CFNBMCNMF, CFNCMNBMF, CFNCMBNMF, CFNMBNCMF, CFNMCNBMF, CMNBMFNCF, CMNBMFCNF, CMNCFMBNF, CMNCFNBMF, CMNFMBNCF, CMNFCNBMF, CMBNCFMNF, CMBNCFNMF, CMBNFMNCF, CMBNFCNMF, CMBNMFNCF, CMBNMFCNF, CMFNBMNCF, CMFNMBNCF, CMFCNBMNF, CMFCNMBNF, CNBMNFMCF, CNBMNFCMF, CNBMCFMNF, CNBMCFNMF, CNBMFNMCF, CNBMFCMNF, CNFMNBMCF, CNFMBNMCF, CNFCMNBMF, CNFCMBNMF, CNMBNFMCF, CNMBNFCMF, CNMCFMBNF, CNMCFNBMF, CNMFNBMCF, CNMFCMBNF.

Low-Agreement Version.

SERBMRSBE, SERBSRMBE, SERMBSRBE, SERMBRSBE, SERSBMRBE, SERSBRMBE, SEBMRBSRE, SEBMRSBRE, SEBSRBMRE, SEBSRMBRE, SEBRMBSRE, SEBRSBMRE, SRBMREBSE, SRBMRESBE, SRBSERMBE, SRBSEBMRE, SRBERMBSE, SRBESBMRE, SRMBSERBE, SRMBSEBRE, SRMBERBSE, SRMBESBRE, SRMBREBSE, SRMBRESBE, SREBMRBSE, SREBRMBSE, SRESBMRBE, SRESBRMBE, SBMRBERSE, SBMRBESRE, SBMRSERBE, SBMRSEBRE, SBMREBRSE, SBMRESRBE, SBERBMRSE, SBERMBRSE, SBESRBMRE, SBESRMBRE, SBRMBERSE, SBRMBESRE, SBRSERMBE, SBRSEBMRE, SBREBMRSE, SBRESRMBE.

Correct Mappings

High-Agreement Version.

Cannenbach -> Carry

Frankheim -> Fred

((Neustadt -> Mike AND Markburg -> Ned) OR

(Neustadt -> Ned AND Markburg -> Mike))

Behringen -> Babs

Low-Agreement Version.

Schwetzingen -> Susan

Marbach -> Eric

((Blaubeuren -> Bill AND Ludwigsburg -> Richard) OR

(Blaubeuren -> Richard AND Ludwigsburg -> Bill))

Ulm -> Mary

Appendix C: Technical Background

General Remarks

While it is still relatively uncommon to use the internet for psychological experiments, the benefits of doing so are many. In this experiment it was possible to (1) effortlessly reach a large group of subjects without concerning ourselves with rooms and enough computers for the subjects to work on, while (2) the experimental data has been inserted and pre-computed without further human assistance. At the end, a pre-formatted excel sheet and the feedbacks by email were the only files that had to be processed by the experimenter. Drawbacks so far are fewer possibilities to control subjects for cheating and inserting false information, but this has not been an issue in our case anyway. It also would be possible to work with user accounts and validity checks of the inputs, but the experiences during the pilot were good so this did not seem to be necessary.

The central components of this self-developed technology used for the experiments are a php4-enabled web server with a coupled mysql3.x database system and a set of html-files on an arbitrary server. The html files display the content of the experiment to the subjects while the embedded javascript collects the data. At the end, all data is sent to the database server together with a unique number to exclude multiple runs of one subject. The script on the database server collects and pre-processes the incoming data and stores it into the database. At the end of the experiment, the database is converted into excel with a second, simple script. The excel file then can be imported into SPSS or other statistical programs. It is also possible to distribute the same or many different experiments on cheap and simple web accounts like all universities provides their staff with while having only one centralized and monitored database server. Password protection and data encryption are easily addable. This configuration would it make possible for a whole department to use web-based experiments in parallel with at cheap costs.

Database Script

```
<! File 'insert.php' a php 4 script that catches the experimental
data from the javascript-post. It checks the data for validity and
does some computation on correctness of the solutions and mappings
before storing everything in a mysql-database which is later
converted to excel. It also provides a possibility to email a
feedback over a form !>

<?php

    ## Catches http input from the experiment and stores data into database
    ## Provides possibility to send a feedback via mail to the experimenter

    ## Database stuff

    # Connect to Database
    $usr = "";
    $pwd = "";
    $db = "";
    $host = "";
    $cid = mysql_connect($host,$usr,$pwd);
    mysql_select_db($db);
    if (mysql_error()) { print "Database ERROR: " . mysql_error(); }

    ## Start catching the data from the http-call\\
    ## and translate the short variables in better readable ones

    # Catch person and version
    if (isset($HTTP_GET_VARS['per'])){$person   = $HTTP_GET_VARS['per'];}
    else {$person = 0;}
    if (isset($HTTP_GET_VARS['ver'])){$version  = $HTTP_GET_VARS['ver'];}
    else {$version = "null";}

    # Catch solution
    if (isset($HTTP_GET_VARS['sol']) && ($HTTP_GET_VARS['sol'] != "")){
        $solution  = strtoupper($HTTP_GET_VARS['sol']);
    }
    else {$solution = "null";}

    # Catch mapping between cities and persons in the low-constraint condition.
    # Variable names are cities and contain persons.

    if (isset($HTTP_GET_VARS['sch'])){$sch      = $HTTP_GET_VARS['sch'];}
    else {$sch = "null";}
    if (isset($HTTP_GET_VARS['mar'])){$mar      = $HTTP_GET_VARS['mar'];}
    else {$mar = "null";}
    if (isset($HTTP_GET_VARS['bla'])){$bla      = $HTTP_GET_VARS['bla'];}
    else {$bla = "null";}
    if (isset($HTTP_GET_VARS['lud'])){$lud      = $HTTP_GET_VARS['lud'];}
    else {$lud = "null";}
    if (isset($HTTP_GET_VARS['ulm'])){$ulm      = $HTTP_GET_VARS['ulm'];}
    else {$ulm = "null";}

    # Catch mapping between cities and persons in the high-constraint condition.
    # Variable names are cities and contain persons.

    if (isset($HTTP_GET_VARS['can'])){$can      = $HTTP_GET_VARS['can'];}
    else {$can = "null";}
    if (isset($HTTP_GET_VARS['fra'])){$fra      = $HTTP_GET_VARS['fra'];}
    else {$fra = "null";}
    if (isset($HTTP_GET_VARS['neu'])){$neu      = $HTTP_GET_VARS['neu'];}
    else {$neu = "null";}
```

```
if (isset($HTTP_GET_VARS['mar'])){$mar      = $HTTP_GET_VARS['mar'];}
else {$mar = "null";}
if (isset($HTTP_GET_VARS['beh'])){$beh      = $HTTP_GET_VARS['beh'];}
else {$beh = "null";}

# Catch reaction times

if (isset($HTTP_GET_VARS['rt5'])){$rtSolution   = $HTTP_GET_VARS['rt5'];}
else {$rtSolution = 0;}
if (isset($HTTP_GET_VARS['rt6'])){$rtMapping    = $HTTP_GET_VARS['rt6'];}
else {$rtMapping = 0;}

# Catch strategy & demographical part

if (isset($HTTP_GET_VARS['s01'])){$s01     = $HTTP_GET_VARS['s01'];}
else {$s01  = 0;}
if (isset($HTTP_GET_VARS['s02'])){$s02     = $HTTP_GET_VARS['s02'];}
else {$s02  = 0;}
if (isset($HTTP_GET_VARS['s03'])){$s03     = $HTTP_GET_VARS['s03'];}
else {$s03  = 0;}
if (isset($HTTP_GET_VARS['s04'])){$s04     = $HTTP_GET_VARS['s04'];}
else {$s04  = 0;}
if (isset($HTTP_GET_VARS['s05'])){$s05     = $HTTP_GET_VARS['s05'];}
else {$s05  = 0;}
if (isset($HTTP_GET_VARS['s06'])){$s06     = $HTTP_GET_VARS['s06'];}
else {$s06  = 0;}
if (isset($HTTP_GET_VARS['s07'])){$s07     = $HTTP_GET_VARS['s07'];}
else {$s07  = 0;}
if (isset($HTTP_GET_VARS['s08'])){$s08     = $HTTP_GET_VARS['s08'];}
else {$s08  = 0;}
if (isset($HTTP_GET_VARS['s09'])){$s09     = $HTTP_GET_VARS['s09'];}
else {$s09  = 0;}
if (isset($HTTP_GET_VARS['s10'])){$s10     = $HTTP_GET_VARS['s10'];}
else {$s10  = 0;}
if (isset($HTTP_GET_VARS['s11'])){$s11     = $HTTP_GET_VARS['s11'];}
else {$s11  = 0;}
if (isset($HTTP_GET_VARS['s12'])){$s12     = $HTTP_GET_VARS['s12'];}
else {$s12  = 0;}
if (isset($HTTP_GET_VARS['s13'])){$s13     = $HTTP_GET_VARS['s13'];}
else {$s13  = 0;}
if (isset($HTTP_GET_VARS['s14'])){$s14     = $HTTP_GET_VARS['s14'];}
else {$s14  = 0;}
if (isset($HTTP_GET_VARS['s15'])){$s15     = $HTTP_GET_VARS['s15'];}
else {$s15  = 0;}
if (isset($HTTP_GET_VARS['age'])){$age     = $HTTP_GET_VARS['age'];}
else {$age  = 0;}
if (isset($HTTP_GET_VARS['sex'])){$sex     = $HTTP_GET_VARS['sex'];}
else {$sex = "null";}
if (isset($HTTP_GET_VARS['exp'])){$expertise        = $HTTP_GET_VARS['exp'];}
else {$expertise = 0;}
if (isset($HTTP_GET_VARS['exR'])){$expertiseRating  = $HTTP_GET_VARS['exR'];}
else {$expertiseRating = 0;}
if (isset($HTTP_GET_VARS['occ']) && ($HTTP_GET_VARS['occ'] != "")){
    $occupation = $HTTP_GET_VARS['occ'];
    }
else {$occupation = "null";}


## Start processing and evaluating the catched data

    # Evaluate solution of the graph-problem for the high-constraint condition

    global $solutionCorrect; # By default, the solution is wrong
```

```
if ($version == "high") {

    switch ($solution) {

        case "CFMNBMCNF" :
            $solution = 1;
            $solutionCorrect = 1; break;

        case "CFMNCMBNF" :
            $solution = 2;
            $solutionCorrect = 1; break;

        case "CFMBNCMNF" :
            $solution = 3;
            $solutionCorrect = 1; break;

        case "CFMBNMCNF" :
            $solution = 4;
            $solutionCorrect = 1; break;

        case "CFMCNBMNF" :
            $solution = 5;
            $solutionCorrect = 1; break;

        case "CFMCNMBNF" :
            $solution = 6;
            $solutionCorrect = 1; break;

        case "CFNBMNCMF" :
            $solution = 7;
            $solutionCorrect = 1; break;

        case "CFNBMCNMF" :
            $solution = 8;
            $solutionCorrect = 1; break;

        case "CFNCMNBMF" :
            $solution = 9;
            $solutionCorrect = 1; break;

        case "CFNCMBNMF" :
            $solution = 10;
            $solutionCorrect = 1; break;

        case "CFNMBNCMF" :
            $solution = 11;
            $solutionCorrect = 1; break;

        case "CFNMCNBMF" :
            $solution = 12;
            $solutionCorrect = 1; break;

        case "CMNBMFNCF" :
            $solution = 13;
            $solutionCorrect = 1; break;

        case "CMNBMFCNF" :
            $solution = 14;
            $solutionCorrect = 1; break;

        case "CMNCFMBNF" :
            $solution = 15;
            $solutionCorrect = 1; break;
```

```
case "CMNCFNBMF" :
    $solution = 16;
    $solutionCorrect = 1; break;

case "CMNFMBNCF" :
    $solution = 17;
    $solutionCorrect = 1; break;

case "CMNFCNBMF" :
    $solution = 18;
    $solutionCorrect = 1; break;

case "CMBNCFMNF" :
    $solution = 19;
    $solutionCorrect = 1; break;

case "CMBNCFNMF" :
    $solution = 20;
    $solutionCorrect = 1; break;

case "CMBNFMNCF" :
    $solution = 21;
    $solutionCorrect = 1; break;

case "CMBNFCNMF" :
    $solution = 22;
    $solutionCorrect = 1; break;

case "CMBNMFNCF" :
    $solution = 23;
    $solutionCorrect = 1; break;

case "CMBNMFCNF" :
    $solution = 24;
    $solutionCorrect = 1; break;

case "CMFNBMNCF" :
    $solution = 25;
    $solutionCorrect = 1; break;

case "CMFNMBNCF" :
    $solution = 26;
    $solutionCorrect = 1; break;

case "CMFCNBMNF" :
    $solution = 27;
    $solutionCorrect = 1; break;

case "CMFCNMBNF" :
    $solution = 28;
    $solutionCorrect = 1; break;

case "CNBMNFMCF" :
    $solution = 29;
    $solutionCorrect = 1; break;

case "CNBMNFCMF" :
    $solution = 30;
    $solutionCorrect = 1; break;

case "CNBMCFMNF" :
    $solution = 31;
    $solutionCorrect = 1; break;
```

```
        case "CNBMCFNMF" :
            $solution = 32;
            $solutionCorrect = 1; break;

        case "CNBMFNMCF" :
            $solution = 33;
            $solutionCorrect = 1; break;

        case "CNBMFCMNF" :
            $solution = 34;
            $solutionCorrect = 1; break;

        case "CNFMNBMCF" :
            $solution = 35;
            $solutionCorrect = 1; break;

        case "CNFMBNMCF" :
            $solution = 36;
            $solutionCorrect = 1; break;

        case "CNFCMNBMF" :
            $solution = 37;
            $solutionCorrect = 1; break;

        case "CNFCMBNMF" :
            $solution = 38;
            $solutionCorrect = 1; break;

        case "CNMBNFMCF" :
            $solution = 39;
            $solutionCorrect = 1; break;

        case "CNMBNFCMF" :
            $solution = 40;
            $solutionCorrect = 1; break;

        case "CNMCFMBNF" :
            $solution = 41;
            $solutionCorrect = 1; break;

        case "CNMCFNBMF" :
            $solution = 42;
            $solutionCorrect = 1; break;

        case "CNMFNBMCF" :
            $solution = 43;
            $solutionCorrect = 1; break;

        case "CNMFCMBNF" :
            $solution = 44;
            $solutionCorrect = 1; break;

        default :
            $solution = $solution;
            break;

    } # End switch

} # End if

# Evaluate solution of the graph-problem for the low-constraint condition

if ($version == "low") {
```

```
switch ($solution) {

    case "SERBMRSBE" :
        $solution = 1;
        $solutionCorrect = 1; break;

    case "SERBSRMBE" :
        $solution = 2;
        $solutionCorrect = 1; break;

    case "SERMBSRBE" :
        $solution = 3;
        $solutionCorrect = 1; break;

    case "SERMBRSBE" :
        $solution = 4;
        $solutionCorrect = 1; break;

    case "SERSBMRBE" :
        $solution = 5;
        $solutionCorrect = 1; break;

    case "SERSBRMBE" :
        $solution = 6;
        $solutionCorrect = 1; break;

    case "SEBMRBSRE" :
        $solution = 7;
        $solutionCorrect = 1; break;

    case "SEBMRSBRE" :
        $solution = 8;
        $solutionCorrect = 1; break;

    case "SEBSRBMRE" :
        $solution = 9;
        $solutionCorrect = 1; break;

    case "SEBSRMBRE" :
        $solution = 10;
        $solutionCorrect = 1; break;

    case "SEBRMBSRE" :
        $solution = 11;
        $solutionCorrect = 1; break;

    case "SEBRSBMRE" :
        $solution = 12;
        $solutionCorrect = 1; break;

    case "SRBMREBSE" :
        $solution = 13;
        $solutionCorrect = 1; break;

    case "SRBMRESBE" :
        $solution = 14;
        $solutionCorrect = 1; break;

    case "SRBSERMBE" :
        $solution = 15;
        $solutionCorrect = 1; break;

    case "SRBSEBMRE" :
```

```
    $solution = 16;
    $solutionCorrect = 1; break;

case "SRBERMBSE" :
    $solution = 17;
    $solutionCorrect = 1; break;

case "SRBESBMRE" :
    $solution = 18;
    $solutionCorrect = 1; break;

case "SRMBSERBE" :
    $solution = 19;
    $solutionCorrect = 1; break;

case "SRMBSEBRE" :
    $solution = 20;
    $solutionCorrect = 1; break;

case "SRMBERBSE" :
    $solution = 21;
    $solutionCorrect = 1; break;

case "SRMBESBRE" :
    $solution = 22;
    $solutionCorrect = 1; break;

case "SRMBREBSE" :
    $solution = 23;
    $solutionCorrect = 1; break;

case "SRMBRESBE" :
    $solution = 24;
    $solutionCorrect = 1; break;

case "SREBMRBSE" :
    $solution = 25;
    $solutionCorrect = 1; break;

case "SREBRMBSE" :
    $solution = 26;
    $solutionCorrect = 1; break;

case "SRESBMRBE" :
    $solution = 27;
    $solutionCorrect = 1; break;

case "SRESBRMBE" :
    $solution = 28;
    $solutionCorrect = 1; break;

case "SBMRBERSE" :
    $solution = 29;
    $solutionCorrect = 1; break;

case "SBMRBESRE" :
    $solution = 30;
    $solutionCorrect = 1; break;

case "SBMRSERBE" :
    $solution = 31;
    $solutionCorrect = 1; break;

case "SBMRSEBRE" :
```

```
            $solution = 32;
            $solutionCorrect = 1; break;

        case "SBMREBRSE" :
            $solution = 33;
            $solutionCorrect = 1; break;

        case "SBMRESRBE" :
            $solution = 34;
            $solutionCorrect = 1; break;

        case "SBERBMRSE" :
            $solution = 35;
            $solutionCorrect = 1; break;

        case "SBERMBRSE" :
            $solution = 36;
            $solutionCorrect = 1; break;

        case "SBESRBMRE" :
            $solution = 37;
            $solutionCorrect = 1; break;

        case "SBESRMBRE" :
            $solution = 38;
            $solutionCorrect = 1; break;

        case "SBRMBERSE" :
            $solution = 39;
            $solutionCorrect = 1; break;

        case "SBRMBESRE" :
            $solution = 40;
            $solutionCorrect = 1; break;

        case "SBRSERMBE" :
            $solution = 41;
            $solutionCorrect = 1; break;

        case "SBRSEBMRE" :
            $solution = 42;
            $solutionCorrect = 1; break;

        case "SBREBMRSE" :
            $solution = 43;
            $solutionCorrect = 1; break;

        case "SBRESRMBE" :
            $solution = 44;
            $solutionCorrect = 1; break;

        default :
            $solution = $solution;
            break;

    } # End switch

} # End if


# Evaluate the mapping between cities and persons,
# taking symmetry into account
```

```
    global $mappingCorrect; # By default, the mapping is wrong

    # Evaluate the mapping for high-constraint condition
    if ($version == "high") {

        # Look into the variable with the name of the city
        # if the right person is in it
        if (($can == "car") && ($fra == "fre")
        && (  (($neu == "mik") && ($mar == "ned"))
            || (($mar == "mik") && ($neu == "ned"))  )
        && ($beh == "bab") ) {
                $mappingCorrect = 1; # Seems to be a good mapping
        }
        else {
        # in case we want to store wrong mappings
        }
    }

    # Evaluate the mapping for low-constraint condition

    if ($version == "low") {

    # Look into the variable with the name of the city
    # if the right person is in it

        if (($sch == "sus") && ($mar == "eri")
        && (  (($bla == "bil") && ($lud == "ric"))
            || (($bla == "ric") && ($lud == "bil"))  )
        && ($ulm == "mar") ) {
                $mappingCorrect = 1;  # Seems to be a good mapping
        }
        else {

        # in case we want to store wrong mappings
        }
    }

## Start filling the database with the evaluated and processed data
## if we are not in the mail-feedback condition

if (!isset($feedback)) {

        $rtSolution *= 100; # Hack for RT
        # Setup SQL statement
        $sql  = "INSERT INTO expReal ";
        $sql .= "(person, version, sex, expertise, expertiseRating,
        solution, solutionCorrect, mapping, mappingCorrect, rtSolution,
        rtMapping, age, occupation, s01, s02, s03, s04, s05, s06, s07,
        s08, s09, s10, s11, s12, s13, s14, s15) VALUES ";
        $sql .= "('$person','$version','$sex','$expertise','$expertiseRating',
        '$solution','$solutionCorrect','','$mappingCorrect','$rtSolution',
        '$rtMapping','$age','$occupation','$s01','$s02','$s03','$s04','$s05',
        '$s06','$s07','$s08','$s09','$s10','$s11','$s12','$s13','$s14','$s15'
        ) ";

        # Execute SQL statement

        $result = mysql_query($sql, $cid);

        # Check for errors

        if (mysql_error()) { print "Database ERROR: " . mysql_error(); }
```

```
    }
?>


<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN"
"http://www.w3.org/TR/1999/REC-html401-19991224/loose.dtd"> <html>
    <head>
        <meta http-equiv="content-type" content="text/html;charset=iso-8859-1">
        <meta name="ROBOTS" content="NONE">
        <meta http-equiv="pragma" content="no-cache">
        <meta http-equiv="expires" content="Thursday, 1-Jan-1991 01:01:01 GMT">
        <title>Experiment Finished</title>
        <link href="pathToStylesheet" rel="stylesheet" media="screen">
    </head>

    <body>
        <div id="content"><h1>Experiment finished!</h1>
        <div class="list">
The experiment is over and your data has been added to the
database. Thank you for participation!<br> The responses are kept
in strictest confidence. Your name will in no way be associated
with the data or data analysis.<br>Please do not discuss the task
with anyone who might participate in this task later on, as this
adversely affects the data.<br> Again, we appreciate your
participation.<br> If you would like to give us feedback, please
use the from below.<br> If you are interested in the results of
this experiment and the purpose behind it,<br> please visit us
again in about two weeks.
        </div>

<?php

    # Write from for feedback
    if (isset($feedback)) {
        $to = ""; // Enter YOUR email here
        $subject = "feedback pilot";
        $body = "A user has entered feedback on the site!\n";
        $body .= "Their feedback is:\n\n";
        $body .= $feedbacktext;
        mail($to, $subject, $body);
        print("<div class=\"dirslist\">Thanks for your feedback!</div><br>");
    }
    else {
        echo "<div class=\"dirslist\"><form action=\"insert.php\" method=\"POST\">
        <textarea cols=35 rows=15 name=\"feedbacktext\"></textarea><br>
        <input type=\"submit\" name=\"feedback\" value=\"Submit\"></form></div>";
    }
?>
        </div>
    </body>
</html>
```

Example Html File with Embedded Javascript

```
<! file example.html This html file with embedded javascript is
the prototype of the sequence of similar scipts which collect the
experimental data. After assigning a random id to the subject, the
script concatenates all data from the field filled in by the
subjects and posts it to the next script. The next, very similar
script deflates the data, concatenates its own data and posts it
to the next one etc. The last script in the row is the insert.php
on the databse server, which is called with a standard html-post
and then collects the data and organizes the database entry. While
this simple javascripts run on all browsers and webspaces
decentralized, the insert.php script is necessary only once and
can be installed on a central database server. RT-measure is done
also done by javascript down to the ms. !>

<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN"
"http://www.w3.org/TR/1999/REC-html401-19991224/loose.dtd"> <html>
    <head>
        <meta http-equiv="content-type" content="text/html;charset=iso-8859-1">
        <meta name="ROBOTS" content="NONE">
        <meta http-equiv="pragma" content="no-cache">
        <meta http-equiv="expires" content="Thursday, 1-Jan-1991 01:01:01 GMT">
        <link href="pathToStylesheet" rel="stylesheet" media="screen">
        <title>Psychological Experiment</title>
        <script language = "JavaScript1.2" type="text/javascript">
            <!-- hide script from older browsers

            // declare variables for time measure and person id
            var time1;
            var fuenf;
            var inhalt="&";

            // initialize random id to person
            function assign_id() {

                var bits = 5;
                var eins = Math.pow(10,bits);
                var zwei = Math.random();
                fuenf = Math.round(zwei*eins);
                if (fuenf < 10000) assign_id();
            }

            // initialize start time for rt measure
            function fill_id() {
                time1 = new Date();
            }

            // go to next page and post variables
            function next_page() {
                time2 = new Date();
                var rt5 = (time2.getTime() - time1.getTime());
                self.location = "page6.html"+"?per="+fuenf+get_forms()+"rt5="+rt5;
            }

            // collect the fields from the forms
            function get_forms() {
              for (var i=0; i<document.experiment.length; i++) {
                if (document.experiment.elements[i].type == "checkbox") {
                  if (document.experiment.elements[i].checked == true) {
                    inhalt += document.experiment.elements[i].name + "="
```

```
                    + document.experiment.elements[i].value + "&";
                  }
                }
        if (document.experiment.elements[i].type == "hidden") {
          inhalt += document.experiment.elements[i].name + "="
            + document.experiment.elements[i].value + "&";
        }
        if (document.experiment.elements[i].type == "radio") {
          if (document.experiment.elements[i].checked == true) {
            inhalt += document.experiment.elements[i].name + "="
              + document.experiment.elements[i].value + "&";
            }
          }
        if (document.experiment.elements[i].type == "select-one") {
            for (var j = 0; j < document.experiment.elements[i].length; j++) {
              if (document.experiment.elements[i].options[j].selected == true) {
                inhalt += document.experiment.elements[i].name + "="
                  + document.experiment.elements[i].options[j].value + "&";
              }
            }
        }
                    if (document.experiment.elements[i].type == "button") {
                        inhalt += "";
                    }
                    if (document.experiment.elements[i].type == "reset") {
                        inhalt += "";
                    }
                    if (document.experiment.elements[i].type == "submit") {
                        inhalt += "";
                    }
                    if (document.experiment.elements[i].type == "hidden") {
                        inhalt += "";
                    }
                    if (document.experiment.elements[i].type == "text" ||
                    document.experiment.elements[i].type == "textarea") {
                        inhalt += document.experiment.elements[i].name + "="
                        + document.experiment.elements[i].value + "&";
                    }
                }
                return inhalt;
            }

        //end hiding script form older browsers -->
        </script>
    </head>
    <body onload="assign_id(), fill_id()">
        <div id="content">
        <form name="experiment" method="get">
<input type="hidden" value="high" name="ver">
            <h1>Instructions</h1>
In the frame above you will find the 'Boat' problem with solution
and the new 'Birthday' problem that you already have read on the
last page. <br>You may have to scroll a bit to see everything.
<br>Your task now is to solve the 'Birthday' problem. You may use
the solution to the 'Boat' problem if you like.<br> Please solve
the 'Birthday' problem and insert your solution into the box below
this instruction.<br> The form of the solution has to be analogous
to the form of the solution of the 'Boat' problem.<br>Of course
there are no cities in the 'Birthday' problem, so use the first
capital letters of the names of the friends between which the
message was passed for your solution. <br>This should look
```

```
remotely similar to this example, but of course with different
letters: <img src="../img/04.jpg" align="middle"><br> When you
inserted your solution, please click on the 'Submit Answers'
button on the bottom of this page.

<div class="dirslist">
    <table>
        <tr>
            <td colspan="3">Insert your solution for the 'Birthday' problem here:</td>
        </tr>
        <tr>
            <td colspan="3"><input type="text" name="sol"></td>
        </tr>
    </table>
</div>

<br> <br>


<input type="button" value="Submit Answers" onclick="next_page()">

</form> </div> </body> </html>
```

Appendix D: Formalia

Hiermit versichere ich, dass ich die Arbeit selbstaendig verfasst habe und keine anderen als die angegebenen Quellen und Hilfsmittel benutzt habe.

(siehe PrOrd. 20 Abs. 6)

Ort Datum Unterschrift