Association for Information Systems
AIS Electronic Library (AISeL)

AMCIS 2009 Proceedings | Americas Conference on Information Systems (AMCIS)

2009

Data Modeling Patterns: A Method and Evaluation

Solomon Anthony
Murray State University, solomon.anthony@murraystate.edu

Vidya Mellarkod
Texas Tech University, vidya.mellarkod@ttu.edu

DATA MODELING PATTERNS: A METHOD AND EVALUATION

Solomon Antony
Murray State University
Solomon.antony@murraystate.edu

Vidya Mellarkod
Texas Tech University
Vidya.mellarkod@ttu.edu

ABSTRACT (REQUIRED)

Patterns capture abstractions of situations that occur frequently in data modeling. Effective use of data modeling patterns can lead to high quality designs and productivity gains. Data modeling patterns are widely available in the public domain, yet there is a lack of studies on usability of such patterns. In this exploratory study we examine the usability of data modeling patterns. Effective use of patterns presupposes the users' ability to find similarities between task and pattern. We present and evaluate some heuristics for finding the similarities. The results of the empirical evaluation indicate that the heuristics are useful and can lead to accurate solutions. Future research as well as implications for researchers and practitioners is also discussed.

Keywords (Required)

Data modeling, patterns, conceptual modeling.

INTRODUCTION

Software reuse has been regarded as a panacea for the escalating demands for software and software engineers in the software development industry (Ravichandran and Rothenberger 2003). It has also been considered by the IS researchers as well as the practitioners as an effective strategy for advancing the software development efficiency, the quality of software applications, the competitive edge and time to market of software development enterprises (Rine and Nada 2000). In the conceptual data modeling area, patterns can be reused in creating new models. Such data modeling patterns (DMPs) are extensively available in the public domain. While attempting to use the patterns, the user will compare the task in hand with an existing pattern and choose the one that is most similar. Finding similarities and analogies is a well researched area within problem solving research (Chen 1995). However, it has not been well utilized in IS development research.

DMP are patterns that capture abstractions of situations that occur frequently in conceptual modeling. The use of DMP provides the data modelers with the benefit of high quality conceptual models developed within short periods of time. Nevertheless, a data modeler is not equipped with a systematic process for using DMPs. This is a major drawback for the data modelers as the extant literature shows that "…human beings do not always discover for themselves clever strategies that they could readily be taught" (Simon 1996). Therefore, human beings do not always have the capability to ascertain cleaver strategies while they can easily learn the same through instruction. This implies that the lack of a method that illustrates the step-by-step process to reuse DMP, will hamper the progress in using DMP and thereby will not allow the database modeler to garner the potential of DMP.

Reusing data modeling patterns can be an effective way to solve problems. It has been found that worked out examples are effective in learning algebra (Sweller and Cooper 1985). It can be expected that research project in reuse of DMPs can also use findings from research in examples based problem solving. The objective of this paper is to propose a method for using DMPs and to test its effectiveness. To assess the effectiveness of the method, analytical as well as empirical validations were conducted. The following section provides a brief background of DMPs and discusses the potential users of the patterns. While section 3 describes the procedure used to develop the method, section 4 illustrates the method itself. Section 5 explicates the analytical and empirical evaluation of the method. Lastly, section 6 discusses the implications and future research. This research can contribute towards better understanding of use of similarities in analogies during design of software artifacts as well as help practitioners with a validated step-by-step for reusing data modeling patterns.

DATA MODELING PATTERNS

A DMP is a representation of data structure that shares many similarities with commonly occurring data models. For example, consider the Ownership pattern that involves an owner and an object. Similarities to this pattern can be found in finance (customer owning accounts), resource management (computer server owned by a department), and many other domains. Many such patterns can be found in the practitioner oriented publications (Coad, North et al. 1997; Fowler 1997; Silverston, Inmon et al. 1997) and in research oriented publications .

(Hay 2000) presents a rich yet parsimonious collection of DMPs that were developed based on studies of various industries. He discusses syntactic, positional, and semantic conventions followed by the description of specific business situations and DMPs that represent them. These patterns can be used by a data modeler as a basis to model an organization's data structures. However, (Hay 2000) presents the DMPs at an abstract level, by leaving out the details, and thereby requires a reasonable level of expertise to understand them. Likewise, the library of DMPs provided by (Fowler 1997) will have to be customized to suit the needs of a specific enterprise. They use specific entity names and instances so that non-experts would find them easy to understand. However, the DMPs and examples are intertwined, and so applicability to other situations is not quite direct. Academic researchers have focused more on groups of patterns. For example, (Batra 2005) describes patterns such as transactions, entity type, recursive data, hierarchical data, plan, event, and generalization. In addition to detailed description of the patterns, an empirical evaluation of existence of these patterns is also provided. There are numerous sources for DMP's (Coad, North et al. 1997; Fowler 1997; Fernandez and Yuan 2000).

For a pattern to be usable in different domains, it must have low granularity (Batra 2005). Each pattern must not have more than three or four entities. In addition, the DMPs must represent a generic concept rather than appear to be a solution to a specific problem. From existing literature, we identified some of the commonly occurring DMPs that satisfy the above-mentioned criteria. There were many other patterns that did not qualify to be included, because of their granularity and generalizability. In this study we used 15 patterns. The list of patterns is shown in

No.	Pattern name	Description of pattern	Entities in the pattern
1	Asset and Asset types	An asset is classified into an asset type	Asset, Asset type
2	Transaction	Items are transacted between a provider and receiver	Receiver, Provider, Transaction, Items
3	Subsequent Transaction	A transaction leads to more transactions	Transaction
4	Accounting	An account has transactions	Account, Transaction
5	Time related relationships	To capture time dependent data	Any Entity and another entityTime,
6	Ownership	Owner owns one or more assets	Owner, Asset
7	Hierarchy	Organization, consists of sub-units	Organization, Sub-unit
8	Allocation	Resource is allocated to consumer	Consumer, Resource
9	Membership	Organization having members	Person, Organization
10	Employment	Person having a job in an organization	Person, Organization
11	Business Policy	Artificial requirement of relationship between entities	Entity 1, Entity 2
12	Plans	A plan having activities	Plan, Activity
13	Action	Actor performing certain action on an object	Actor, object
14	Work Order	An organization prepares a work order that involve activities	Organization, Work order and activity
15	Observation	Phenomenon, observed by an observer using an instrument	Phenomenon, Observation, Instrument, Observer

Table 1.

No.	Pattern name	Description of pattern	Entities in the pattern
1	Asset and Asset types	An asset is classified into an asset type	Asset, Asset type
2	Transaction	Items are transacted between a provider and receiver	Receiver, Provider, Transaction, Items

3	Subsequent Transaction	A transaction leads to more transactions	Transaction
4	Accounting	An account has transactions	Account, Transaction
5	Time related relationships	To capture time dependent data	Any Entity and another entityTime,
6	Ownership	Owner owns one or more assets	Owner, Asset
7	Hierarchy	Organization, consists of sub-units	Organization, Sub-unit
8	Allocation	Resource is allocated to consumer	Consumer, Resource
9	Membership	Organization having members	Person, Organization
10	Employment	Person having a job in an organization	Person, Organization
11	Business Policy	Artificial requirement of relationship between entities	Entity 1, Entity 2
12	Plans	A plan having activities	Plan, Activity
13	Action	Actor performing certain action on an object	Actor, object
14	Work Order	An organization prepares a work order that involve activities	Organization, Work order and activity
15	Observation	Phenomenon, observed by an observer using an instrument	Phenomenon, Observation, Instrument, Observer

Table 1: Patterns used in this study

The books and research papers on DMPs list most of the DMPs relevant to the business data requirements (Batra 2005) (Coad, North et al. 1997; Fowler 1997). There is little doubt that readers of those books and articles would learn more about DMPs in general. But, who is most likely to benefit by using DMPs? Experts and experienced database designers are less likely to use the DMPs, since they already possess the mental scripts of commonly occurring situations, and they would tend to draw from their own expertise and experience. Non-expert designers, on the other hand, would find DMPs more useful. There are three groups of non-expert designers who will find the DMPs useful. They are: (a) students in data modeling classes, (b) participants in IS projects who have little knowledge of data modeling (such as functional area specialist who lends a hand to analysts in IS projects), and (c) beginner level database designers.

- Students in business data modeling classes: Learners in any domain, be it computer programming, solving math problems or decision making, benefit by studying examples, analogies, and sample solutions. Hence the students learning to model data will benefit by studying and using DMPs. The knowledge gained by studying DMPs can be used in solving data modeling problems in the future.
- Participants in IS projects: Entity Relationship (ER) diagrams depict entities and relationships, and are developed by the analysts. It should be noted that ER diagrams are also used as common media for communication between functional area specialists and the analysts (Maes and Poels 2006). Therefore the functional specialists can be expected to read and interpret conceptual data models, i.e. ER Diagrams. While the *creation* of ER diagrams is a difficult process, reading and interpreting the same can be considerably easier. Having been trained on interpreting ER diagrams, if the functional specialists are trained on using patterns, they can help the analysts during conceptual modeling.
- Beginner level database designers: Beginner data base designers will also benefit by using DMPs. The patterns would enable them to develop a wider perspective on modeling.

Based on the above arguments, it is conceivable that non-expert designers stand to benefit most from using DMPs. Hence, the user group of interest is non-expert designers. To use the patterns, the designer must find similarities between the task at hand and a pattern.

A METHOD FOR USING DATA MODELING PATTERNS

A method should describe a systematic process of accomplishing a task (Martin 1995). Well developed methods can guide problem-solvers during their problem-solving activity. They limit the cognitive strain of a problem-solver by narrowing the problem space, and by providing strategies for efficient search within the problem space (Martin 1995). Generally, methods can be devised by individuals with an extensive insight into the best practices, or those with extensive past experiences, or by ends-means analysis, or by action research. For this study we developed a method for using DMPs employing verbal protocol analysis based action research method.

To build a conceptual data model (typically an ER Diagram) the user will (a) read the Task document and then identify the sub-tasks, (b) identify the entities and relationships in the sub-tasks, (c) compare the sub-task relationship with list of patterns, (d) identify a pattern that is similar to the sub-task relationship, (e) map the entities from the sub-task to entities in the pattern and (f) instantiate the pattern with entities from the sub-task. Repeat these steps for each relationship in the sub-task. Repeat these steps for each sub-task in the task description. Once all the ER diagrams have been instantiated, integrate them to create a data model for the entire task. The method is illustrated in the diagram below (Figure 1). A detailed description of the method follows.

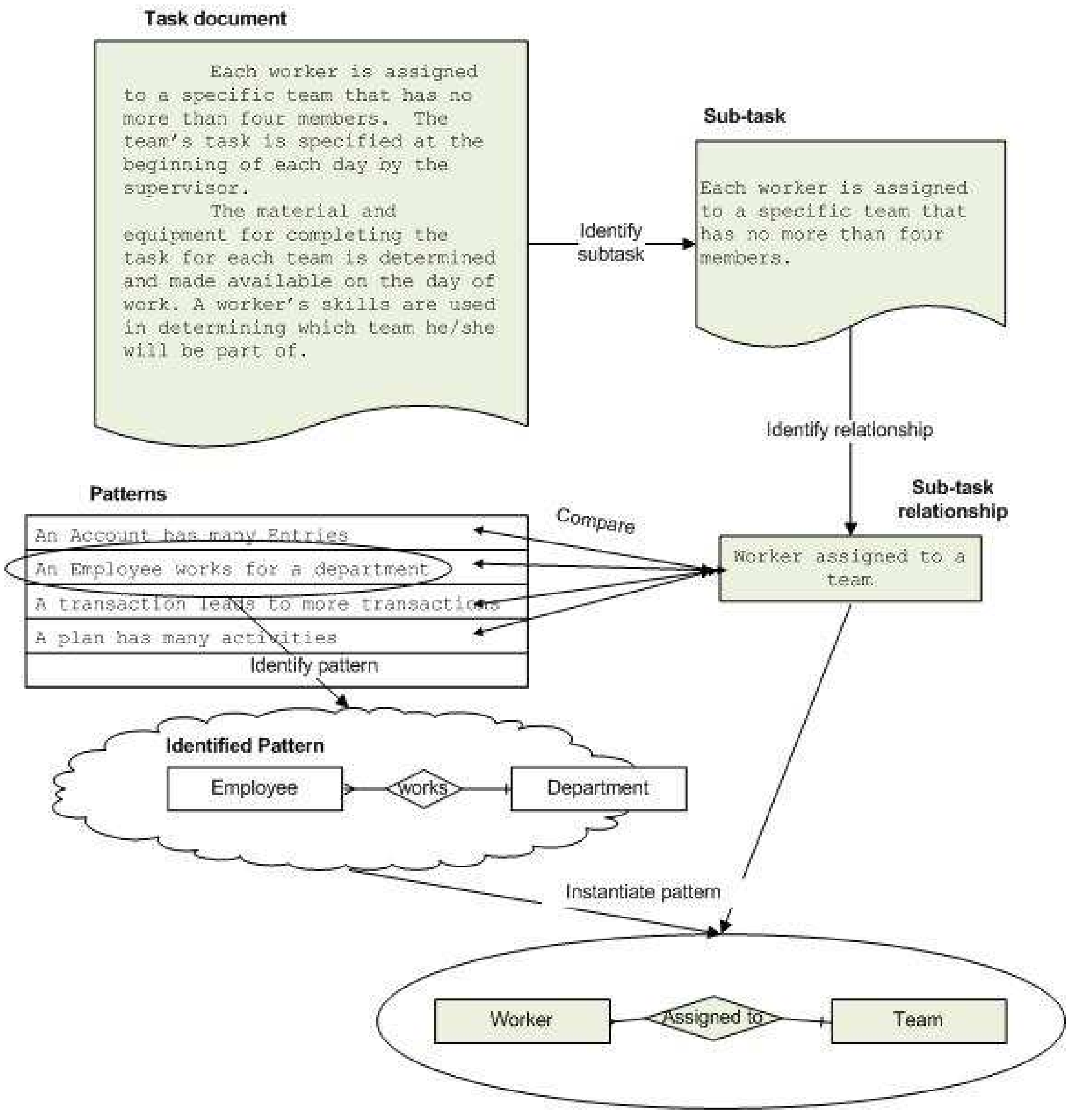

Figure 1: Method for using Patterns

Step 1: Identify sub-tasks

The information requirements document (IRD) must be decomposable into smaller sub-tasks. Since the each pattern we use have a few entities, our focus is to identify sub-tasks that describe a small group of entities and the relationships among them. For example, the IRD for a *Construction Project Management System* may include descriptions about (i) *workers*, (ii) *their skills*, (iii) *contractors*, (iv) *contractor specializations*, (v) *equipment*, (vi) *equipment* types, (vii) *building sites*, (viii) *architects*, (ix) *building codes*, (x) *city officials*, (xi) *inspections*, (xii) *work schedules* etc. Among these descriptions about *workers* and *their skills* can be considered a sub-task.

Step #2: Identify entities and relationships within sub-task

Next, the designer must identify the entities and relationships between the entities in the sub-task. Entities typically form the subjects and objects of a statement whereas the relationships are typically the verbs that connect subjects and objects. For example, consider the following sub-task that describes the assignment of workers to tasks in the building construction project.

> Each worker is assigned to a specific team that has no more than four members. The team's task is specified at the beginning of each day by the supervisor. The material and equipment for completing the task for each team is determined and made available on the day of work. A worker's skills are used in determining which team he/she will be part of.

On studying the above description, the user can identify entities- namely Team, Member, Supervisor, task, material, and equipment - the following sub-task relations and entities: (a) *worker – team membership*, (b) *work order to complete the task* (c) *properties of workers*. Next, the user searches for a pattern that appears most similar to the sub-task relationships under consideration. For each of these sub-task relationships, the modeler must look for best matching patterns.

Step #3: Match patterns to sub-task relationship

Compare the sub-task relationship and identify the pattern that best matches it. It is preferable that the DMP user compares the sub-task relationship with each pattern in the list. It is possible that there may be more than one pattern that matches the sub-task relationship. The modeler would then choose the best fitting pattern among the short-listed patterns.

Step #4: Instantiate ER diagram for sub-task relationship

Once the user finds candidate patterns for each sub-task, he creates an instance of the pattern substituting the entities in the pattern with entities from the sub-task relationship.

Step #5: Find data model for each relationship

The user repeats steps from 2 to 4 for each relationship in the sub-task. Creating a list of ER diagrams.

Step #6: Analyze and model all other sub-tasks

The user will repeat steps 1 through 5 so that all the sub-tasks are analyzed.

Step #7: Integrate the models

In this step, the user would integrate the all ER models into a larger model for the entire task.

The first step (Identify sub-tasks), would be fairly straight forward, since the requirements document would present the requirements already classified into various sub-sections. Identifying relationships from textual description (step #2) can also be easily accomplished by using heuristics on how to look for entities, and heuristics on how to identify sentences that relate entities. The third step of identifying matching patterns for each relationship can be difficult. Step 4 can be accomplished with a CASE tool. Analyzing and integrating the individual ER models would call for using semantic or syntactic matches. These matches can be done at the schema level or at elements level. The focus of this paper is on steps 2, 3, 4 and 5 only. It is not implied that steps 1 and 7 are less important in developing data models based on reusable patterns. We wished to limit the scope of the paper to a subset of the steps so that deeper exploration of those steps can be accomplished.

HEURISTICS FOR DISCOVERING SIMILARITIES

While a designer can himself determine if a sub-task matches a pattern or not, training on *how* to find the matching patterns can be useful. We can derive some heuristics based on the Structure Mapping Theory (Gentner and Medina 1998). In the Structure Mapping theory framework, there are two ways to find similarities between a source relation and a target relation. They will be known as Abstract heuristic and Analogy heuristic.

Abstract level heuristic

The Abstract level heuristic[1] involves converting the sub-task relationship into an abstract conceptual relationship and then comparing it with the pattern. Some sub-task relationships are *special cases* of a pattern. For example, a *construction task and related activities* is a special case of the *Plan-activity* pattern. Another example is "City inspector inspecting the foundation and reporting" can be a special case of the *Observation* pattern. Finding patterns based on the 'Special case' heuristic can be taught to modelers by requiring them to ask questions like "*Is the sub-task relation a special case of the relation in the pattern?*" or "*Would a generic case of sub-task relation be similar to the pattern?*". For more examples of the sub-task patterns that are abstractly similar see Table 2.

Pattern name	**Examples**
Asset and Asset types	Assets in an office can be classified as furniture, electronics, fixtures, etc.
Ownership	Department owns computers
Hierarchy	Region, and sub-region; Boss and subordinate
Allocation	Labs allocated to department
Membership	Department belonging to a college
Business Policy	Certain employees eligible to lead projects
Plans	Planning a birthday party
Action	Salesman serves a territory
Observation	Employee review, product quality control

Table 2: Examples of Abstract heuristic

Analogy heuristic

Analogies are quite useful in finding similarities between two objects. Even children are equipped with learning about their environment and solve problems through the use of analogies (Goswami 1986). The essence of analogical thinking is the transfer of knowledge from once situation to another by a process of mapping – finding a partial set of correspondences between the elements (objects, attributes, and relations) that form mental representations of the two situations (Hesse 1963). In the current domain – data modeling - the Abstract level heuristic may not be sufficient to find a suitable match. There may be some cases where there is little or no similarity between entities of the pattern and the entities of the sub-task. In such cases, the modeler will have to employ the Analogy heuristic, where he finds if sub-task and pattern are analogous. Analogies cannot be found by superficial comparison of pattern relationships with task relationships. Instead, the modeler must actively seek and discover parallels between the sub-task relationship and the pattern. For example consider a group of fire fighters offering safety programs that involve courses and lectures to local business people. At first glance, the transaction pattern and the fire-fighters' safety programs do not appear to have much in common. But, when we are asked, "How are fire -fighters offering safety courses similar to organization selling products to customer?", we discover that they both involve transfer of things from a source to a target, and possibly for a price. In other words, the firefighter case can be

[1] Gentner calls this the Abstraction heuristic, but owing to different definition of the term Abstraction by IS practitioners and researchers, we call it the Abstract level heuristic.

abstracted as "professionals providing packages of services to users, and each package includes a number of courses" (A graphic representation of the two diagrams are shown in Figure 2).

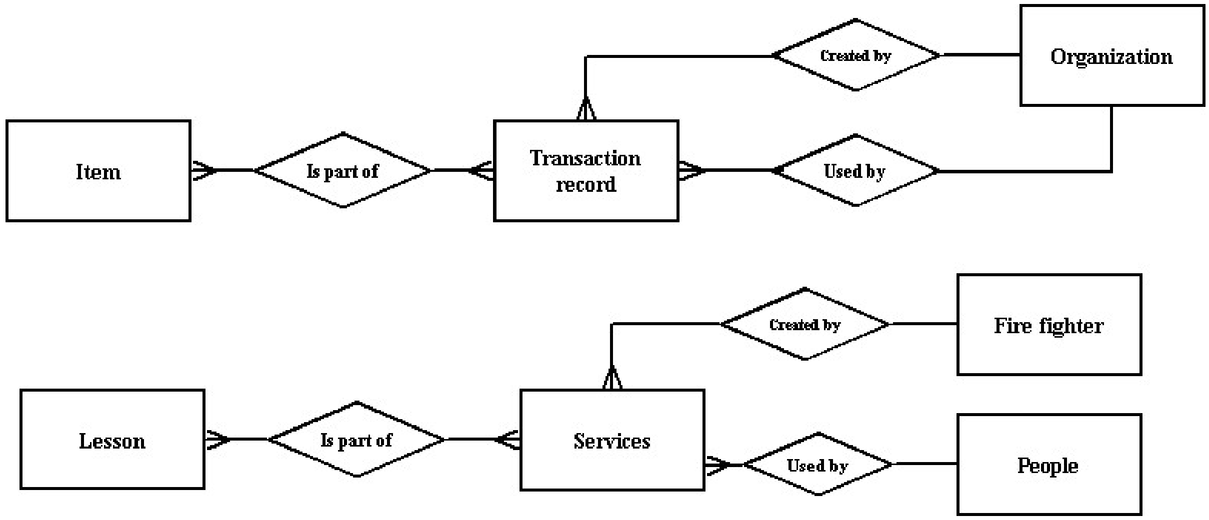

Figure 2: Analogy example

When the Abstract based heuristic does not help in similarity between a pattern and the sub-task relationship, the user must use the Analogy heuristics. To use the Analogy heuristic, the modeler would have to ask questions such as "*What makes the sub-task relationship similar to the pattern?*" or "*How can the sub-task relation be similar to the pattern?*". Some more examples of similarity that can be discovered using analogies are shown in Table 3.

Sub-task relationships	Patterns that are analogically similar
Dietary requirements of animals at a zoo	Resource allocated to consumer (Resource allocation)
Cooking recipe having specific steps	A Plan having many activities (Plan pattern)
Watering schedule at a conservatory	Time dependent resource allocation pattern
Christmas wish-list	Transaction - Items are transacted between a provider and receiver

Table 3: Examples for Analogy heuristic

While use of heuristics can be expected to help, it is possible to find similarities between the source sub-task relationships and target patterns. It is desirable to verify if use of heuristics can indeed improve the pattern matching, so that it can be incorporated into methodology. Hence, we decided to explore the consequences of using heuristics based method for finding similarities. Specifically we are interested in learning if heuristics based approach to similarity finding (a) improves the accuracy of the data models, (b) enable modelers to consider more patterns, and (c) whether modelers can identify more matching relationships or not. In addition, it will be useful to find (d) if there is a *preferred* set of patterns that have higher appeal to the modeler. We included (d) for the following reason. Although the number of patterns we set out to use were not many (only 15), the kinds of relationships were not that varied. Most of them were binary 1-to-many or Many-to-Many type relationships. So, it was debated that subjects could find the solution by considering only a sub-set of these patterns. Hence we decided to find if there is a preferred set of patterns.

RESEARCH METHOD:

We employed a quasi-experimental design in this study. There is a control group and one treatment group. The treatment group received special training in using the heuristics for identifying similarities. We employed novice database designers. Accuracy of solution, number of distinct relationships modeled, are main outcome variables.

SUBJECTS:

The objective of the study was to find if heuristics can be used to find similarities between task relationships and patterns. For this purpose, we recruited 59 MBA students who were enrolled in an Information Systems class and trained them on concepts of ER diagrams. Specifically, they were trained on how to read and interpret entities, and relationships in an ER diagram. The training lasted an hour. After the training the subjects completed a quiz that evaluated their ability to read and interpret ER diagrams. We used *E-R Diagram for the Pine Valley Furninture Company* from (Hoffer, Prescott et al. 2002). The mean of 12.3 was found to be significantly higher than expected mean of 8.5. (*t value = 15.17, p < 0.001*). Hence we can conclude that the subjects did learn to read and interpret ER diagrams. The subjects received 5% course credit for participating in the study and completing both the sessions.

PROCEDURES:

As a next step in the experiment, we wished to test if the subjects can successfully identify patterns for a given task. To test this we invited the same set of MBA students to training sessions on data modeling patterns. There were two class sections. Each class section was randomly assigned to one of the two groups – a control group and a treatment group. The control group subjects were introduced to data modeling patterns and were shown how to use patterns to create data models for a given problem. As part of the training, they solved a practice problem also. Their training lasted 40 minutes. The treatment group subjects were also introduced to data modeling patterns and shown how to use them for creation of data models. In addition, they were shown how to use the heuristics for matching the pattern to a sub-task. Their training session and problem solving session lasted for about 50 minutes. After training, the subjects were handed a description of a data modeling task. The task description was already decomposed into sub-tasks. . They were given the URL of a web site where the images of the 15 data modeling patterns could be viewed. The subjects were required to identify all the sub-tasks, all the sub-task relationships, and for each sub-task relationship they were required to identify a pattern and indicate how the entities in the sub-task relationships can be mapped to the entities in pattern. The names of patterns, relationship names, and entity mappings constituted their solutions. The subjects were not required to integrate the individual ER diagrams.

GRADING

The correct ER model for the task contains four sub-tasks, and a total of 14 relationships. Individually counting, there were 10 entities. In all, 56 subjects participated in the study of which 54 were present during the data modeling training session as well as the actual study. The subjects submitted their solutions using a web browser. A subjects solution is made up of a list of matchings. A matching is a dyad of a sub-task name and a pattern. A total of 351 matchings from 56 subjects were recorded. Of the 351, 286 were found to be usable. The unusable matchings were either (a) duplicate entries (when subject submitted the same solution more than once), or (b) the sub-task modeled was not part of the correct solution and had no standard to compare with or (c) solution was submitted by a subject who did not participate in all sessions. From the sub-task and patterns matched, it was possible to instantiate the relationship. Each of the instantiated relationship was checked for correctness. If the entities, relationship and connectivity were correct, it was awarded 1 point. If the relationship was partly correct, it was awarded 0.5 points, and if it was incorrect, no points were awarded. While grading the relationships, the grader was not aware of the subject's identity or the experimental group from which solution is from. The points for individual matchings were added to arrive at an overall grade for each subject. In addition, the total number of relationships modeled, and the number of unique patterns used by each subject were also recorded.

RESULTS:

Although the treatment group's mean grade is slightly higher than the control group's grade, the results of the ABOVA test revealed (Table 4) no significant difference. However, the treatment group subjects modeled significantly more relationships than the control group subjects (see Table 5).

Accuracy	*Control group*	*Treatment group*
Mean	3.39	3.44
Variance	2.79	4.34
Observations	29	25
Df	28	24
F	0.64	
P(F<=f) one-tail	0.130	
F Critical one-tail	0.52	

Table 4: Scores comparison

Number of relationships	*Control group*	*Treatment group*
Mean	4.76	5.72
Variance	3.33	8.54
Observations	29	25
Df	28	24
F	0.39	
P(F<=f) one-tail	0.009	
F Critical one-tail	0.52	

Table 5: Number of relationships modeled

The treatment group ended up using significantly more number of unique patterns than the control group (Table 6)

Unique Patterns	*Control group*	*Treatment group*
Mean	3.52	4.16
Variance	1.33	3.31
Observations	29	25
Df	28	24
F	0.40	
P(F<=f) one-tail	0.011	
F Critical one-tail	0.52	

Table 6: Unique patterns used

Among the patterns selected, the Action pattern (Actor performing certain action on an object) was the most popular. See Table 7 for top four patterns of choice.

Pattern Selected	Counts of matchings
Action	49
Hierarchy	37
Employment	30
Event	25

Table 7: Most frequent patterns used

The Action pattern was the most frequently used pattern of choice for Crime, Driving, and Training sub-task relationships. For the Employment sub-task the Employment and Hierarchy patterns were most frequently used. For the experimental task we used, there appears to be a clear list of preferred patterns.

DISCUSSION AND CONCLUSION

In this paper, we presented a method for using data modeling patterns. Among all the steps of the methodology, determining similarity between a pattern and the sub-task was believed to be the hardest. Hence, a method that uses some heuristics that are based on Structure mapping theory was developed. The two methods were compared using non-expert data base modelers. Although the results are not strong, the usability of patterns as an alternative to traditional data modeling has been tested. The subjects had no experience in modeling, but were able to correctly find matching patterns to some of the sub-task relationships. The training session included description of the method, demonstration of how to use it and an opportunity to practice the method. Feedback from the subjects indicated that more practice could have improved performance.

Our experience with this study has bolstered our belief that DMPs are usable by non- experts. In this study, the subjects compared a textual representation of a sub-task relationship with a graphic and textual representation of the pattern. In other domains, language based analogies were found to be more facilitative in problem solving (Lane and Schooler 2004). Hence, it is possible that comparing textual description of sub-task relationship to textual description of pattern would lead to improved effectiveness. Application of the Structure mapping theories to other areas of IS development can also be explored.

There was clear preference for choosing a few patterns (Action, Employment, Hierarchy, Event) by most of the subjects. In future studies, we may not have to use all fifteen patterns, just handful of them would be sufficient. Very few subjects actually used the variations to patterns. A qualitative study that uses process tracing or protocol analysis can help in understanding why the subjects chose these few patterns.

REFERENCES

Batra, D. (2005). "Conceptual Data Modeling Patterns." Journal of Database Management **16**(2): 84-106.

Chen, Z. (1995). "Analogical transfer: from schematic pictures to problem solving." Memory & cognition **23**(2): 255.

Coad, P., D. North, et al. (1997). Object models: strategies, patterns, and applications, Yourdon Press Upper Saddle River, NJ, USA.

Fernandez, E. B. and X. Yuan (2000). Semantic Analysis Patterns. Procs. of 19 th International Conference on Conceptual Modeling, ER2000.

Fowler, M. (1997). Analysis Patterns: Reusable Object Models, Addison-Wesley Professional.

Goswami, U. (1986). "Children's use of analogy in learning to read: A developmental study." Journal of Experimental Child Psychology **42**(1): 73-83.

Hay, D. C. (2000). Data Model Patterns: Conventions of Thought, Dorset House Publishing Co., Inc. New York, NY, USA.

Hesse, M. B. (1963). Models and analogies in science, Sheed and Ward.

Hoffer, J. A., M. B. Prescott, et al. (2002). "Modern Database Management."

Lane, S. M. and J. W. Schooler (2004). "Skimming the surface. Verbal overshadowing of analogical retrieval." Psychological Science **15**(11): 715–719.

Maes, A. and G. Poels (2006). "Evaluating Quality of Conceptual Models Based on User Perceptions." Lecture Notes in Computer Science **4215**: 54.

Martin, M. P. (1995). Analysis and Design of Business Information Systems. Upper Saddle River, NJ, USA, Prentice Hall.

Ravichandran, T. and M. A. Rothenberger (2003). "Software reuse strategies and component markets." Communications of the ACM **46**(8): 109-114.

Rine, D. C. and N. Nada (2000). "An empirical study of a software reuse reference model." Information and Software Technology **42**(1): 47-65.

Silverston, L., W. H. Inmon, et al. (1997). The Data Model Resource Book: A Library of Logical Data Models and Data Warehouse Designs, Wiley.

Simon, H. A. (1996). The sciences of the artificial, MIT press.

Sweller, J. and G. A. Cooper (1985). "The use of worked examples as a substitute for problem solving in learning algebra." Cognition and Instruction: 59-89.