December 2004

A methodology for using Data Model Patterns

Solomon Antony
Texas Tech University

Vidhya Mellarkod
Texas Tech University

A methodology for using Data Model Patterns

Solomon Antony
Texas Tech University
solomon.antony@ttu.edu

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

ABSTRACT

Data model patterns are to be used as a repository of data models, from which an analyst can select a pattern that best suits the task he/she is modeling. There are handful resources for data models patterns, but there is no structured methodology or list of heuristics for using those patterns. In this paper we describe a methodology for matching patterns with tasks. To develop the methodology we observed the Pattern-Task matching processes of experts. Using Protocol analysis techniques we identified the process and using Gentner's Structure mapping theory heuristics we devised heuristics for pattern-task matching. The outcome of this exercise is a structured methodology for using data model patterns. We argue that even non-experts who have little or no experience with data modeling can use this methodology. We elaborate how this methodology can be used in data modeling instruction also

Keywords

Data models, patterns, methodology, reuse

INTRODUCTION

A software methodology is a systematic process of solving a task that is typically part of the systems development lifecycle. Methodologies guide users during the problem solving process by providing frameworks, tools to think with, and procedures to be followed. Benefits of using a methodology includes preventing the ill effects of cognitive biases, efficient management of the problem solving process, better measurability of outcomes and promotion of methods that are replicable across different projects by different actors. Researchers in the past have used best practices, personal experiences, ends-means approach and action research to create software methodologies. We used theoretical findings from similarity and analogical reasoning and empirical observation of experts to develop a methodology for using data model patterns. In this paper, results from a preliminary empirical study and plans for future studies have been outlined.

DATA MODEL PATTERNS

A data-model pattern is a representation of data structure that shares many similarities with commonly occurring solutions to data requirements problems. For example, a transaction involves exchange of resources between two entities. The application of this pattern can be found in trades, borrowing, selling of goods and selling of services etc. If a modeler has sufficient comprehension of the concept of transactions he can generalize that knowledge to other transaction like situations also. There are a handful of sources for data model patterns (e.g. Hay 1996, Silverston 1997, Coad 1997, Fowler 1997). The central theme of these books is reuse of data models with patterns as the repository of data models. There have been attempts at developing tools that can automatically select patterns for specific case (e.g. Purao 1998). However, a step-by-step methodology of how to use these patterns for a given style of requirements document was found to be lacking. This paper complements their work by providing such a methodology.

Some of the commonly occurring patterns, and the entities they use are listed in the table below. They have been compiled from Hay (1996), Silverstone et al (1997), and Fowler(1997).

No.	Generic name	Entities in the ER diagram
1	People	People, characteristics of people, contact information
2	Organization	Organizational unit, person, division, accountability
3	Accounts	Account, Account type, Transaction, entry, quantity
4	Planning and monitoring	Action, Plan, Procedure, Resource, Time
5	Transactions, and delivery of goods	Organization, product, product type, payment, contract
6	Assets, Ownership	Product, product type, structural component, Product family, Organizational unit
7	Membership, employment	Organizational unit, person, Time, Role, Responsibility,
8	Work orders (delivery of services, production order)	Procedure, organizational unit, activity, asset / product, people
9	Time related concepts (Timesheets, trends, historical records, plan schedules)	Time unit, Measure, measuring instrument, organizational unit
10	Portfolio	Contracts, quotes, trades, Risks,
11	Projects	Worker, Asset, task, Time, Location, Service

Table 1: Sample pattern relationships and their entities

SIMILARITY AND ANALOGICAL REASONING

To make use of patterns from a repository, the user must be able to match a task description with a candidate pattern. For example, given a task description the user must be able to recognize the task as describing a transaction pattern or as a membership pattern etc. Findings from research on similarity and analogical reasoning can be used to develop heuristics so that the process of matching can be more predictable. According to the Structure-Mapping theory (Gentner 1983), there are three ways comparisons can be made. They are (a) Literal comparison, (b) Abstraction, and (c) Analogy.

Literal similarity is a comparison where the most of the predicates of base is mapped to predicates in the target. Here the similarities of base object and target object are derived based on the mappings of the attributes of base object with target object. For example "The X12 star system is similar to our solar system" implies that X12 have similar aspects to our solar system (Gentner 1983).

Abstraction is a comparison in which the base domain's abstract relational structure is compared with the relational structure of the target (Gentner 1983). For example consider "A patron borrows books from the library" and "A customer renting videos from a video rental store". The first domain's relation can be abstracted as "borrowing" which can then be compared the abstraction of the second domain. The focus is more on the relational structure and less on the objects that participate in the relation. In abstraction, there may be mapping of some object attributes also. In this example, it can be seen that that some attributes of patron will map to attributes of a customer.

Analogy is about finding some parallels between the base relation and the target relation. There is few or no mapping of attributes or objects. For example, consider how "hydrogen atom" and "our solar system" are similar. There are surface similarities between the attributes of objects being compared. However, if we think harder about hydrogen atom and solar system are similar, we will find that that "An electron revolves around the nucleus in a hydrogen atom" and "Planets revolve around the sun in the solar system". Thus, analogy is found by in depth comparison of the two domains. Thus, analogy is the forced process of finding parallels between the sub-task relation and the pattern. (Gentner, 1983). Using the above findings, it is possible to develop heuristics for user of data model patterns.

PROTOCOL STUDY OF DATA MODEL PATTERNS USAGE

Concurrent capture of verbal protocols is considered valid and reliable for problems that involve high level cognitive processes, such as strategies and representations (Ericsson and Simon 1984). Protocol analysis techniques have been used in the past to develop software methods. For example, the pioneer researchers in human problem solving (Newell and Simon 1972) used protocol analysis for understanding the problem solving strategies of humans with the intention of developing computerized tools. We used Protocol analysis technique to capture the processes of data model pattern usage by two experts. We used the first process data set to test and revise the coding scheme presented in Irwin (2002). The revised coding scheme (See Appendix) was used in analyzing second process data. The strategies inferred by analyzing the episodes of the process data formed the basis of the methodology.

DATA MODEL PATTERN METHODOLOGY

STEP #1: Identify sub-tasks in the task description

Typically data requirements for a business situation include descriptions of concepts and the data that will be used, created or stored. For example, the information requirements for a Construction Project management system would include descriptions about workers, their skills, contractors, contractor specializations, equipment, and equipment types, building sites, architects, planners, city officials, work schedules, business rules and constraints. Typically, each of these individual concepts is presented independent of others. The first step in the DMP methodology requires that the user identify task description's sub-sections (hereafter called sub-tasks) that describe a functional concept.

STEP #2: Compare sub-tasks with patterns

We recommend that the user become familiar with the ER diagrams that represent each pattern and some of its variations. Matching patterns to sub-tasks involve comparing textual description to series of ER diagrams. There are two distinct steps involved in matching patterns. Firstly, the user needs to comprehend the meaning of interactions between the actors or things in the task description. Secondly, the user searches for a pattern that appears most similar to the relation described in the sub-task under consideration. 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 the following sub-task relations: (a) Worker – team membership, (b) work order to complete the task (c) properties of workers. Using the list in Table 1, we can identify that patterns 6 and 7 can be used. The third conceptual relation describes the properties of worker entity and can be modeled as an entity. To find matches between the sub-task and patterns, we can use the following heuristics.

- ➢ Literal similarity: It is possible that some entities in the sub-task are special cases of pattern's generalized entity. In data modeling terms, it is equivalent to finding the ISA- relationships or sub-types or synonyms between an entity in sub-task and an entity in pattern. For example, 'worker' entity in sub-task is literally similar to People pattern.
- ➢ Abstraction In using the abstraction heuristic, the user will convert the sub-task description to abstract conceptual relation. The relationship in the pattern is already in an abstract form for comparison. The abstraction mechanism is equivalent to finding that the sub-task relationship is a special case of the generic relationship in a pattern. In other words, abstraction is to relations as literal similarity is to entities. For example, consider a Transaction pattern (that has a seller, buyer and number of items being transacted) and a sub-task description that deals with Home owners purchasing services from a maintenance company. The attributes of a generic buyer and those of a home owner can be consider similar, and so are the attributes of transaction entities that mapped to the home service entities.
- ➢ Analogy: Analogy is about finding some parallels between the two relations or between two objects that participate in relations. It is possible that the entities of the sub-task relationship and the entities of the pattern relationship do not have any common attributes. Superficially, the sub-task relationship may not appear to be a special case of the pattern relationship. However on closer scrutiny, the similarities between them may become apparent. For example, consider the Transaction pattern and "fire-fighters offering safety programs that involve courses and lectures to local businesses" sub-task. At first glance, the two domains do not appear to have much in common. But, when you abstract the fire-fighter case as professionals providing packages of services to users, and each package includes a number of courses, the similarity becomes apparent. The entities of the pattern do not appear similat to entities of the sub-task case. The pattern and target solution are shown in the above Figure 2.

How does having these three ways to match sub-tasks with patterns enable a functional specialist to match patterns? According to Gentner and Medina (1998), the process of comparison has been a key mechanism for intellectual development of human beings. As children, we begin with 'concrete comparison' that is based on sight, touch, smell and taste. While into adulthood, we are well equipped to compare in abstract terms, sometimes even with rule-like regularities. So, similarity comparison is an innate skill that can be tapped upon at will. We expect that the users would attempt pattern matching first by trying literal similarity, then abstraction and analogy as the last resort.

STEP #3: Match pattern to sub-task

Using similarity comparison techniques, the user would find candidate patterns for each sub-task. While comparing the sub-tasks to the patterns, the user will attempt to create a one-to-one mapping between constructs in the sub-task and the constructs in the pattern, for every sub-task in the task description. The user will note instances where the mapping is not exact. In addition, the user will rate his/her confidence on the accuracy of each mapping.

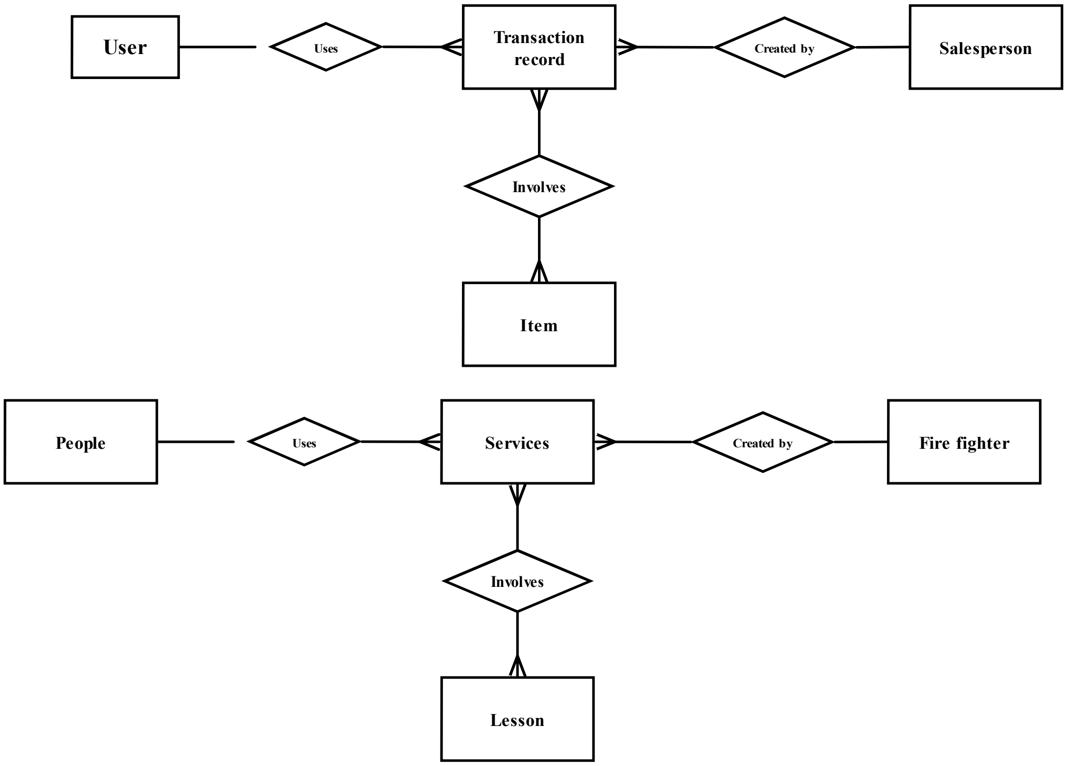

Figure 1: Analogy example

PROCEDURE TO USE OF THIS METHODOLOGY

The outcome of using the pattern matching methodology is a mapping of sub-task and patterns. A task description gets split up into sub-tasks and each sub-task is mapped to one more pattern from the patterns bank. The methodology does not provide heuristics for identifying the entities, since we expect the users will be able to identify the entities with relative ease. For identifying relationships the mapping will be useful. The mapping can then be handled by either by an experienced data modeler who will fine-tune the data model. We expect that such sharing of work between the experienced and non-expert designers (who can be a functional area specialist) is desirable in software project teams.

INTENDED USERS OF THE DATA MODEL PATTERNS

We believe that these patterns should be more useful to non-experts designers rather than experts. Because, experts already have many templates of data models from which they can draw from precluding the need for using external representations of patterns. Non-experts on the other hand, rely on knowledge that is in short-term memory.

We expect that this methodology will be useful for beginner database designers as well. Once the students have learnt to model entities and are trained on reading ER diagrams, they will be ready to use the patterns. They would need to know how to convert a task description into sub-tasks. Students can then use the repository of patterns to identify potential matches for relationships in the task description. Once the candidate pattern has been determined they can copy the ER diagram, sans labels, from the pattern, and insert the task labels. Then they would integrate the individual ER models so that the entire task is modeled.

VALIDATION OF THE METHODOLOGY

Any methodology is expected to meet certain qualifications before it can be deemed usable. Ruble (1997), and Yadav et al. (1988) provide a list of desirable characteristics for evaluating methodologies. We describe how our methodology measures up to those characteristics.

- An analysis technique must provide a framework for representation and discovery about the task domain: Input information for our methodology primarily comes from information requirements documents, which are typically not dynamic. Since the task is to model the information specified in the documents, and nothing more, there is little scope for discovery.
- The technique should adequately cover each aspect of the business problems: The scope of use for the methodology depends on how varied the patterns repository is. It is possible to create data model patterns for most business activities such as logistics, manufacturing, accounting etc. Since the methodology is not domain dependent, we believe, that it can be used with all type of patterns.
- The results of the analysis technique must be verifiable by the consumers of the outcomes of the analysis: The database designers can verify the accuracy of the sub-task to pattern matches, by analyzing the requirements document independently.
- The analysis method must also create measurable units for project managers to assess: We believe that the data model pattern matching can be completed in linear time with respect to the problem size. So, we believe that assessment of progress can be made with little effort.
- The outcomes of the analysis technique must be consumable so that it can be converted to designs and then to development of the system: The mapping of pattern and sub-task constructs will readily be usable by the data base designer, provided the user gets all parts of the solution correct.
- The analysis technique must score reasonably well in other characteristics such as ease-of-use and time taken: These empirical issues can be verified by field testing the methodology. Due to the lack of a competing methodology, the measures of ease-of-use and time-taken would be hard to interpret.

In addition we did a preliminary evaluation of this methodology; we found that a subject who had little knowledge about data modeling mapped six of seven sub-tasks correctly into patterns.

FUTURE RESEARCH

This paper describes a methodology, but does not provide a detailed empirical validation. Hence, the next step is to validate the methodology in a controlled setting. It is possible that factors such as subject's data modeling aptitude, subject's similarity finding aptitude, employment of heuristics, task characteristics, task-methodology fit and time pressure would affect the performance of the subjects. A number of controlled experiments are underway to address the effects of above factors.

CONCLUSIONS

Having a methodology for reusing software artifacts is desirable and will help organizations in cutting back the time required. In this paper we presented a methodology for using data model patterns. It is our contention that the methodology can be used by non-expert designers as well as functional area experts. Some of the factors that affect the performance of users of this methodology include perceived ease-of-use, users' analytical abilities, complexity of the task, and usability of the patterns, and time available. Our methodology promotes abstraction and analogy as two means of finding similarity between task and pattern.

REFERENCES

1. Coad, P., North, D., and Mayfield, M. (1997). *Object Models: Strategies, Patterns, and Applications*. Yourdon Press, NJ.
2. Ericsson, K.A. and H.A. Simon (1980). Verbal Reports as Data. *Psychological Review*, 1980(87): p. 215-251.
3. Fowler, M (1997). *Analysis patterns: Reusable object models*. Addison Wesley, Menlo Park CA.
4. Gentner, D. and Medina, J (1998). Similarity and the development of rules. *Cognition* 65, 263—297.
5. Hay, D. C. (1996). *Data Model Patterns: Conventions of Thought*. Dorset House Publishing, New York, NY.
6. Irwin, G. (2002) The Role of Similarity in the Reuse of Object-Oriented Analysis Models. *Journal of Management Information Systems*. Vol. 19 No. 2, Fall 2002 pp. 219 – 248
7. Newell, A. and Simon, H.A. (1972). *Human Problem Solving*, Englewood Cliffs, Prentice Hall.
8. Purao, S., Storey, V. C., and Han, T. (2003). Improving analysis pattern reuse in conceptual design: Augmenting automated processes with supervised learning. *Information Systems Research* 14, 3 (2003), 269—290.
9. Purao, S. (1998). APSARA: a tool to automate system design via intelligent pattern retrieval and synthesis. *ACM SIGMIS Database*. Volume 29 , Issue 4 Fall 1998 Pages: 45 - 57
10. Ruble, D. A. (1997). *Practical Analysis and Design for Client/Server and GUI Systems*. Yourdon Press, NJ.
11. Scheer, A-W, and Hars, A. (1992). Extending data modeling to cover the whole enterprise. *Communications of the ACM* 35, 9, 166—171.
12. Silverston, L., Inmon, W. H., and Graziano, K. (1997) *The Data Model Resource Book*. John Wiley & Sons, New York..
13. Todd, P. and Benbasat, I. (1987). Process tracing methods in decision support systems research: Exploring the black box. *MIS Quarterly* 11, 4, 493—512.
14. Yadav, S. B., Bravoco, R. R., Chatfield, A. T., and Rajkumar, T. M. (1988). Comparison of analysis techniques for information requirement determination. *Communications of the ACM* 31, 9, 1090—1097.

Appendix

Codes	Guidelines
1. Comprehend a pattern a. Reading a pattern b. Synthesize a pattern	*Comprehending a pattern* involves studying a pattern and/or coming up with higher level understandings of the pattern.
2. Comprehend the task a. Reading the task b. Synthesize the task	*Comprehending a task* constitutes studying a task and drawing higher level understandings of the task.
3. Setting Criteria for Selection	*Setting criteria for selection* implies specifying rules that a potential pattern should satisfy.
4. Evaluate a pattern for task	*Evaluating a pattern for task* involves making surface level mappings between the task and the pattern.
5. Verify correctness of pattern to task a. Look for attribute matches b. Look for entity matches c. Look for relationship matches d. Identify differences e. Resolve differences	*Verifying correctness of pattern* to task includes making deeper level comparisons between attributes, entities, and relationships required in the task with those provided by the pattern.
6. Recognize applicability of a pattern	*Recognizing the applicability* of a pattern implies identifying the appropriateness of a pattern to the task or a part of the task.
7. Recognize inapplicability of a pattern	*Recognizing the inapplicability* of a pattern implies identifying the inappropriateness of a pattern to the task or a part of the task.
8. Final Verification	*Final Verification* happens after different patterns have been identified as applicable to the task.
9. Process management a. Monitoring the pattern matching process b. Planning the pattern matching process c. Reflecting on pattern matching process	*Process management* includes monitoring, reflecting, and planning the pattern matching process
10. Miscellaneous utterances	*Miscellaneous utterances* are expressions irrelevant to the problem at hand
11. Analysis	*Analysis* includes delving deeper into the modeling techniques and using sophisticated analysis to draw conclusions.