IDEA GROUP PUBLISHING
701 E. Chocolate Avenue, Suite 200, Hershey PA 17033-1240, USA
Tel: 717/533-8845; Fax 717/533-8661; URL-http://www.idea-group.com
ITB12848

This paper appears in the book, *Emerging Trends and Challenges in Information Technology Management, Volume 1 and Volume 2* edited by Mehdi Khosrow-Pour © 2006, Idea Group Inc.

Improving the Software Development Process by Improving the Process of Relationship Discovery

Joseph T. Catanio, LaSalle University, Computer Science Dept, Philadelphia, PA 19141, T: 1-215-951-1142, catanio@lasalle.edu

Michael Bieber, New Jersey Institute of Technology, Information Systems Dept, Newark, NJ 07102, bieber@oak.njit.edu

ABSTRACT

A significant aspect of systems analysis involves discovering and representing entities and their inter-relationships. Guidelines exist to identify entities but none provide a rigorous and comprehensive process to explicitly capture relationships. Whereas, other analysis techniques lightly address the relationship discovery process, Relationship Analysis (RA) provides a comprehensive list of questions to determine a domain's relationship structure.

The quality of design artifacts, such as class diagrams can be improved by first representing the complete relationship structure of the problem domain. RA can significantly enhance the systems analyst's effectiveness, especially in the area of relationship discovery, resulting in improved analysis and design artifacts.

INTRODUCTION

The literature indicates that the best way to improve the software development life-cycle is to improve it during the early stages of the process (Sommerville, 2001) (Faulk, 2000) (Booch et al., 1998); in particular, during the analysis phase. During the analysis phase, components are determined through the identification process of the system's entities and relationships. Informal guidelines exist to help identify entities. However, no defined processes exist to explicitly and systematically assist in eliciting relationships or documenting them in class diagrams or entity-relationship (E/R) diagrams (Beraha & Su, 1999). Relationships constitute a large part of an application domain's implicit structure. Completely understanding the domain relies on knowing how all the entities are interconnected. Relationships are a key component, yet lightly addressed by E/R and class diagrams (Catanio, 2004a,b). These diagrams capture a limited subset of relationships and leave much of the relationship structure out of the system model. While analyses and models are meant to be a limited representation of a system, the incomplete relationship specification is not by design, but rather a lack of any technique to determine them explicitly (Bieber & Yoo, 1999). As a result, many analyses miss aspects of the systems they represent. Relationship Analysis (RA) addresses these concerns and provides a straightforward way to identify the relationship structure of a problem domain, and thus fills a void in the systems analysis process.

RELATIONSHIPS

Relationships within information systems can be depicted in diagrams using various modeling techniques. These techniques identify system components and properties to build a conceptual model of the system. A conceptual model must be powerful enough in its semantic expressiveness and easily comprehensible, as it serves as a communication medium among stakeholders who interact with it during the stage of analysis and modeling (Topi & Ramesh, 2002).

Entity-Relationship (E/R) modeling (Chen, 1976) is one of the best known semantic data modeling approaches and is often used to represent the conceptual schema of the problem domain by identifying its entities, properties, and relationships. Although relationships are depicted on the E/R diagram, the amount of information these relationships convey is rather limited and only provides minimal information describing relationships.

Within the object-oriented methodology, conceptual models are represented as a collection of interacting objects. These objects help to encapsulate an abstract concept into a self-contained unit organized by their similarities into classes. Relationship classification caters towards class representation and there is no prescribed way to determine classes or relationships.

Semantic classification within information systems strongly categorizes main system entities but poorly classifies how they are related. Modeling techniques focus on identifying main system components but loosely identify how components are related and interrelated. These semantic models offer the modeler a small set of the fundamental abstractions needed to identify the relationship structure of the domain. None of the existing techniques explicitly helps the analyst in determining the detailed relationship structure of the domain, and therefore they are not as comprehensive as analysts treat them. For any analysis to be truly effective, it needs to identify a complete set of relationships. RA is such a technique that can supplement and "complete" the existing approaches.

RELATIONSHIP ELICITATION QUESTIONS

The following elicitation questions are a subset of those that the RA (Catanio, 2004a) process uses to explicitly identify relationships.

- Does the item have a definition?
- How can this item be expanded?
- What is this item a part-of?
- What is part-of this item?
- Which other items are similar/opposite to this item?
- What precedes/follows this item for a given purpose?
- What output results from the item's inputs?
- What can modify this item?
- How can this item be reused?
- What is this item dependent on?
- What is dependent on this item?

EXPERIMENT

An experiment was performed using RA (Catanio, 2004a) to assess whether RA is an effective technique to explicitly identify the relationship structure of a problem domain.

This study measures the quality of class diagrams generated by groups that accomplished the same task utilizing different means. The benefits of expertise are less pronounced when analyzing and solving a problem with a well-defined technique (Spence & Brucks, 1997).

We speculate that low experience groups utilizing RA questions will produce design artifacts of equal quality as high experience groups.

A. Method

The method of the experiment is a 2 x 2 factorial design. The two independent variables are experience and analysis tool and the four conditions in this experiment are:

- Use-case, low experience
- Use-case & RA, low experience
- Use-case, high experience
- Use-case & RA, high experience

The use-case analysis tool represents the control group category and the treatment group represents the use-case & RA category.

Experience has been used extensively in experiments to determine its effect on the learning process (Amento et al., 2000) (Schenk et al., 1998) (Spence & Brucks, 1997). To determine experience level, subjects completed a pre-experiment questionnaire that identified academic background, software background, and professional work experience relating to software analysis. Experts divided the subjects into low and high experience based upon the criteria determined from the pre-experiment questionnaire. The low experience subjects were randomly selected and placed in a team consisting of three low experience individuals. Similarly, high experience subjects were randomly selected and placed in a team consisting of three high experience individuals.

B. Procedures

The main experiment lasted one week, whereby the first day included a training session. Each team, from the four different group types, performed the same task. This permits the pure effect of the treatments to be isolated because the difference in tasks is controlled. This will increase the internal validity of the research (Rosenthal & Rosnow, 1991).

All subjects were taught how to develop use-case analysis diagrams and generate class diagrams. The treatment groups were trained in RA. To eliminate any training effect, the control groups were provided an equivalent enrichment topic, namely entity relationship (E/R) analysis. After the training, all groups were provided the same task to solve with their team members. All groups had one hour to create the use-case analysis diagram. At the conclusion of the session, all groups were provided with an expert-generated use-case analysis diagram to the problem statement to use as a basis to complete the remaining experimental steps. The control groups generated class diagrams after use-case analysis. The treatment groups performed RA and then generated class diagrams.

C. Measures

Expert judges rated the quality of each group's generated class diagram utilizing a 10-point scale whereby a 10 represents a perfect score. Expert judges have been used in many studies to evaluate quality of system design and decision-making (Shaft & Vessey, 1998).

D. Empirical Evidence

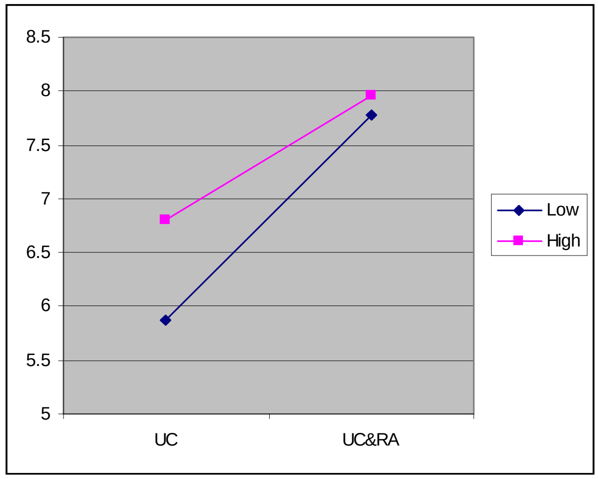
Table 1 provides the quality grade mean and standard deviation calculations for each of the conditions.

The mean score of those using RA is better than not using RA for both low and high experience groups. This represents a positive synergistic effect and suggests that low experience analysts utilizing RA could be more effective than high experience analysts without RA. Figure 1 depicts the quality grade for the high and low experience level groups.

Table 1. Quality grade mean and standard deviation calculations

2 x 2 Factorial Design		**Experience**	
		Low	**High**
Analysis Tool	**Use-case**	Mean = 5.87 SD = 2.53 N =15	Mean = 6.81 SD = 0.97 N = 13
	Use-case & RA	Mean = 7.78 SD = 0.55 N = 16	Mean = 7.96 SD = 0.78 N = 13

Figure 1. Quality grade for groups



CONCLUSION

Relationship Analysis (RA) provides a usable set of questions that improves an analyst's effectiveness in relationship discovery. A study showed that RA questions provide a fuller and richer analysis, resulting in improved quality of class diagrams, and that RA enables analysts of varying experience levels to achieve a similar level of quality of class diagrams. RA significantly enhances the systems analyst's effectiveness, especially in the area of relationship discovery, resulting in improved analysis and design artifacts.

The encouraging results provide convincing incentive to begin developing RA into a practical technique that can be utilized by the software community. In a post-experiment debrief session many subjects mentioned that current software object-oriented analysis techniques provide little assistance in identifying classes and how they interrelate. Much of the system analysis process and class diagram creation is delegated to highly experienced system analysts. RA attempts to level the playing field among analysts of varying experience by providing a set of questions to explicitly identify relationships, which in turn help to identify classes.

Although a methodology independent technique, RA can be positioned seamlessly between the use-case analysis and class diagram generation steps of the widely-used object-oriented paradigm. In future work we hope to conduct field trials at various types of organizations showing that RA improves the development process, is compatible with current approaches, and that practitioners are satisfied with and accept it will be the first stage towards RA's inclusion into the object-oriented paradigm and toolkits used by software engineers.

ACKNOWLEDGMENTS

Many thanks to Teri, Chason, John, and Shads.

REFERENCES

Amento, B., Terveen, L., & Hill, W. (2000). "Does Authority Mean Quality? Predicting Expert Quality Ratings of Web Documents," AT&T Shannon Laboratories.

Beraha, S., & Su, J. (1999). "Support for Modeling Relationships in Object Oriented Databases," Data & Knowledge Engineering, Vol. 29, No. 3, pp. 227-257.

Bieber, M., & Yoo., J. (1999). "Hypermedia: A Design Philosophy," ACM Computing Surveys 31(4es).

Booch, G., Jacobson, I., & Rumbaugh, J. (1998). The Unified Modeling Language Users Guide, Addison Wesley, Massachusetts.

Catanio, J. (2004a). "Relationship Analysis: Improving The Systems Analysis Process," Ph.D. Dissertation, New Jersey Institute of Technology.

Catanio, J., Nnadi, N., Zhang, L., Bieber, M., and Galnares, R., (2004b) "Ubiquitous Metainformation and the WYWWYWI* Principle," Journal of Digital Information, Volume 5, Issue 1, April 2004. (*What you want, when you want it)

Chen, P. (1976). "The Entity-Relationship Model – Toward a Unified View of Data," ACM Transactions on Database Systems, Vol. 1, No. 1.

Faulk, S. (2000). "Software Requirements: A Tutorial," Software Requirements Engineering, Second Edition, IEEE, Los Alamitos, California, pp. 158-179.

Rosenthal, R. & Rosnow, R. (1991). Essentials of Behavior Research Methods and Data Analysis, Second Edition, McGraw-Hill, New York.

Schenk, K.D., Vitalari, N., & Davis, K. (1998). "Differences Between Novice and Expert Systems Analysts: What Do We Know and What Do We Do?," Journal of Management Information Systems, Vol. 15, No. 1, pp. 9-50.

Shaft, T. M. & Vessey, I. (1998). "The Relevance of Application Domain Knowledge: Characterizing the Computer Program Comprehension Process," Journal of Management Information Systems, Vol. 15 No. 1, pp. 51-78.

Sommerville, I. (2001). Software Engineering, Sixth Edition, Addison-Wesley Publishers, Massachusetts.

Spence, M. T., & Brucks, M. (1997). "The Moderating Effects of Problem Characteristics on Experts' and Novices' Judgments," Journal of Marketing Research, Vol. XXXIV, pp. 233-247.

Topi, H., & Ramesh, V. (2002). "Human Factors Research on Data Modeling: A Review of Prior Research, an Extended Framework and Future Research Directions," Journal of Database Management, Vol. 13, No. 2, pp. 3-19.

Related Content

Analyzing Key Decision-Points: Problem Partitioning in the Analysis of Tightly-Coupled, Distributed Work-Systems
Susan Gasson (2012). *International Journal of Information Technologies and Systems Approach (pp. 57-83).*
www.irma-international.org/article/analyzing-key-decision-points/69781

State of the Art and Key Design Challenges of Telesurgical Robotics
Sajid Nisarand Osman Hasan (2018). *Encyclopedia of Information Science and Technology, Fourth Edition (pp. 6872-6881).*
www.irma-international.org/chapter/state-of-the-art-and-key-design-challenges-of-telesurgical-robotics/184383

New Factors Affecting Productivity of the Software Factory
Pedro Castañedaand David Mauricio (2020). *International Journal of Information Technologies and Systems Approach (pp. 1-26).*
www.irma-international.org/article/new-factors-affecting-productivity-of-the-software-factory/240762

Business Continuity Management in Data Center Environments
Holmes E. Millerand Kurt J. Engemann (2019). *International Journal of Information Technologies and Systems Approach (pp. 52-72).*
www.irma-international.org/article/business-continuity-management-in-data-center-environments/218858

Team Characteristics Moderating Effect on Software Project Completion Time
Niharika Dayyala, Kent A. Walstromand Kallol K. Bagchi (2021). *International Journal of Information Technologies and Systems Approach (pp. 174-191).*
www.irma-international.org/article/team-characteristics-moderating-effect-on-software-project-completion-time/272765