

Object-oriented Analysis using Patterns: A Review and Research Opportunities

Felix Leung
Department of Information Systems
City University of Hong Kong
isfelix@cityu.edu.hk

Narasimha Bolloju
Department of Information Systems
City University of Hong Kong
narsi.bolloju@cityu.edu.hk

Abstract

The application of analysis patterns has gained an increasing interest among researchers and software analysts over the last few years. Their beliefs of advocating analysis patterns are to support knowledge reuse and produce more reliable conceptual design at reduced time frame. The idea of documenting analysis patterns is to capture proven expert solutions to recurring business problems. However, the strength of analysis patterns is not fully achieved and their reuse is still limited. In this paper, we formally give an overview on the state of Object-oriented modeling using patterns. In order to strengthen the groundwork for object-oriented analysis patterns (OOAP), we suggest five future OOAP research directions which are establishing theoretical/empirical foundation, proposing appropriate structure, understanding influences on different system analysts, enriching modeling capabilities and suggesting other kinds of measurement.

Keywords: Object-oriented Modeling, Analysis Patterns, System Analysis

1. Introduction

In the late 1980s, using object-oriented analysis (OOA) techniques in system analysis has received a great attention from the information systems (IS) groups and organizations (Coad and Yourdon, 1991, Rumbaugh *et al.*, 1991, Embley *et al.*, 1992). OOA is data-oriented approach and focuses on identifying objects and their activities (Rumbaugh *et al.*, 1991). In recent years, an advance OOA modeling practise has been proposed by introducing analysis patterns for the representation of conceptual models. Patterns embody the knowledge and experience of many developers. A common belief is that analysis patterns can improve conceptual modeling quality because they are proven solution and have been scrutinized by many developers. Using patterns is not a new idea in software design for reuse, but it is recognized as a new and less matured research discipline in system analysis. Most of the existing pattern-related research papers are either concentrating on discovering commonalties forming an analysis pattern for a particular commerce process or writing an analysis pattern for an existing application domain. The aim of this work is neither of these approaches, but to provide an understanding on those findings across existing studies and identify new directions in the pattern-based conceptual modeling.

In the section below, we first provide an overview on the state of OO modeling using patterns. Next, we define our view on the state of conceptual modeling researches using object-oriented analysis patterns (OOAP) and their motivation in Section 3. The most relevant empirical and non-empirical studies published to date are described. We address prior studies on related research topics, approaches and methods, etc. In Section 4, we suggest the future directions to do. Finally, the conclusions are presented in Section 5.

2. Background

The development of pattern-oriented design techniques has introduced in the software engineering community since 1987 (Beck and Cunningham, 1987). Over the past decade, a vibrant research and user community has sprung up around software patterns. The evolution of pattern discipline can be traced through three primary sources. The first is the famous book of the Gang of Four (GOF) on design patterns (Gamma *et al.*, 1995). The GOF patterns are generally considered the foundation for all other patterns. Besides, there were also a number of books (Coad *et al.*, 1997, Nicola *et al.*, 2001, Yacoub and Ammar, 2003, etc.) which were devoted entirely to patterns have been published. The second is a series of conferences dedicating to software pattern. In addition, some pattern-related discourses have flourished at some famous object-oriented conferences. The application of patterns became popular in the object-oriented community after the 9th annual conference on Object-Oriented Programming Systems, Languages, and Applications (OOPSLA) in 1994. In the same year, the first conference on Pattern Languages of Programs (PLoP) which focus is improving the expression of patterns was also held. The third is the Hillside Group has been established. It is an educational non-profit organization to further the field. Although most of the common use of pattern until now has been at the design stage, many people believe that analysis patterns can contribute more to reusability and software quality than the other varieties (Fernandez and Yuan, 2000, Purao *et al.*, 2003).

2.1 Object-oriented Analysis Pattern

The basic idea behind OOA is to reflect the problem domain, system's responsibilities and user requirements by means of identifying and defining classes and their objects found in the specified domain. It focuses on what the system is supposed to do rather than how the system is to be built. In principle, the notion of "analysis" comes close to the word's original meaning and helps requirements engineers understand the domain. As denoted by its name, an OOAP regards itself as a pattern used in the OOA. Like any other software patterns, OOAP share the main definition with other kinds of patterns. It is an abstraction of a group of related generic objects (meta-classes) with expected relationships which is likely to be helpful again and again in OO development. Major development in analysis patterns can be attributed to Peter Coad *et al.* (1992, 1997, 1999) and Martin Fowler (1996). Analysts can reuse and incorporate analysis patterns into their work to address problems of knowledge representation in describing and understanding an application area, and in this manner to improve the time and resources consumed in the conceptual modeling. Also, analysis patterns can allow other analysts to benefit from the experience of pattern designers. To summarize, some of the benefits brought by OOAP are:

Reuse solution – by reusing already established design, system analysts get a head start on their problems. System analysts do not have to reinvent solutions for commonly repeating problems. In this way, system analysts can minimize their intellectual effort as well as time and cost.

Establish common terminology – from a solution readability and understandability perspective, analysis patterns provide a common base vocabulary and a common viewpoint of the problem for system analysts.

Promote modeling quality – as promoting proven knowledge and solution to a recurring business problem, analysis patterns can give some indications to entire stakeholders about the quality of the overall conceptual model. Particularly, system analysts get the benefit of learning from the experience of others. A preliminary study (Bolloju, 2004) has highlighted the effectiveness of applying analysis patterns in helping the analyst identify missing classes, associations and aggregations.

3. Overview of some existing studies on OOAP

This paper takes the results and findings reported in other studies at face value. Our emphasis is to understand the influence of this analysis technique. As pattern-based Object-oriented analysis and design (OOAD) is still in its infancy, those selected papers for this study came from both conference proceedings and journal articles. Papers were searched based on the words “Analysis Pattern”, “Design Pattern” and “Object-oriented Modeling” within the title, abstract or keyword fields. The journal articles that were used in this study are come from the Information Systems, System Engineering and Computer Science fields like IEEE Transactions on Software Engineering (TSE), IEEE Software, Communications of the ACM (CACM), Information and Software Technology (IST), Software Practice and Experience (SPE), Journal of System and Software (JSS) and Requirements Engineering (RE). The other papers are come from the following conferences: Americas Conference on Information Systems (AMCIS), International Conference on Software Engineering (ICSE), Pattern Languages of Programs Conference (PLoP), European Conference for Object-Oriented Programming (ECOOP) and Conference on Object-Oriented Programming, Systems, Languages, and Applications (OOPSLA). In order to get a significant and current snapshot of the field of pattern research, we have examined related publications during the past decade.

3.1 Prior Studies

During the 1990s, much research was done on pattern is subjective. In the earliest beginning, the research topics are usually related to the use assessment and refinement of individual patterns. The research approaches were mainly “discovery” or “invention” of new patterns in a particular type of business problem. In recently year, the focus shift to the creation of complete pattern languages which is a way to describe existing patterns and their collaboration networks. Our following review of prior studies is not intended to be comprehensive. We simply aim to illustrate the kinds of work that have been done.

3.1.1 Development of Analysis Patterns

Pattern is not created but discovered or identified from existing world. There are four main approaches that have been used to develop analysis patterns (Hamza, 2004). They are the direct approach, specialization approach, analogy approach, and stability approach. The Fowler's (1996) analysis patterns are belong to the direct approach category. They are not generalized or abstracted further after they are identified. In the specialization approach, identified patterns are abstracted so that they are easier to apply themselves in similar and related applications than those patterns formed by direct approach. Analysis patterns described in Coad *et al.*'s (1997) and Nicola *et al.*'s (2001) belong to this approach. In the analogy approach, patterns are abstract to construct templates such as “Resource Rental” pattern (Braga *et al.*, 1998). An analogy between the analysis pattern and the new application is conducted in order to adopt the identified pattern template to fit into the new application. Finally, the stability approach is a layered approach for developing new application (Hamza, 2002). All these different approaches share the common theme of capturing core knowledge of the real-world problem for future modeling.

3.1.2 Classification of Analysis Patterns

Currently, the two main streams of OOAPs can be classified as application-independent and application-dependent patterns. The application-independent patterns like Coad *et al.*'s (1997) 31 analysis patterns and Nicola *et al.*'s (2001) 12 analysis patterns are all small fundamental patterns for object business modeling. They appear in a 2-class structure deal mostly with the four generic classes (i.e., people, places, things, and events) and relationships among them (i.e., transactions, roles). In Figure 1, examples of Coad *et al.*'s analysis patterns

(1997) are shown and we have shown how do these six patterns collaborate together to form a larger transaction model for a video shop in Figure 2.

The application-dependent patterns are those most generic application-specific patterns such as the “Party” patterns (Fowler, 1996), the “Resource Rental” pattern (Braga *et al.*, 1998), etc. In Figure 3, the Braga *et al.*’s “Resource Location” pattern is shown and it’s objective is to provide an abstraction that can be reused to model the problem of renting any kind of things. Besides application-specific patterns, there are others which are less generic and may be reused in different applications inside a certain domain. For example, insurance application patterns (Robertson and Strunch, 1993) and banking application patterns (Marsura, 1998) are more domain-specific in nature.

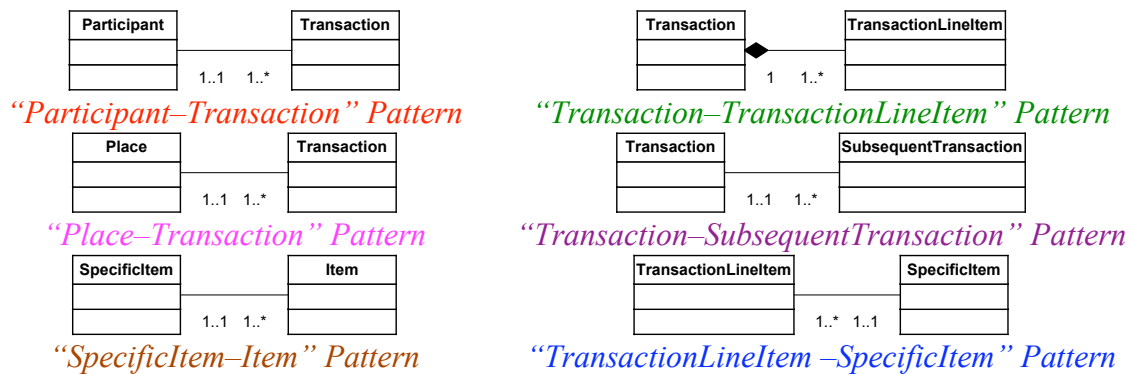


Figure 1 – Example of Coad *et al.*'s analysis patterns (1997)

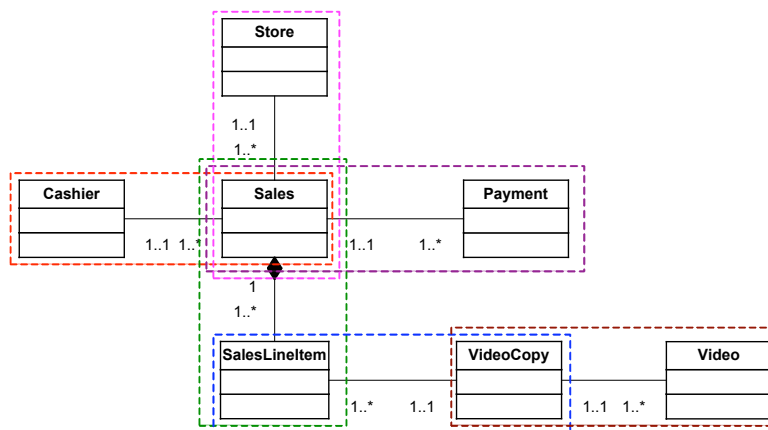


Figure 2 - Modeling video store transaction using Coad *et al.*'s analysis patterns (1997)

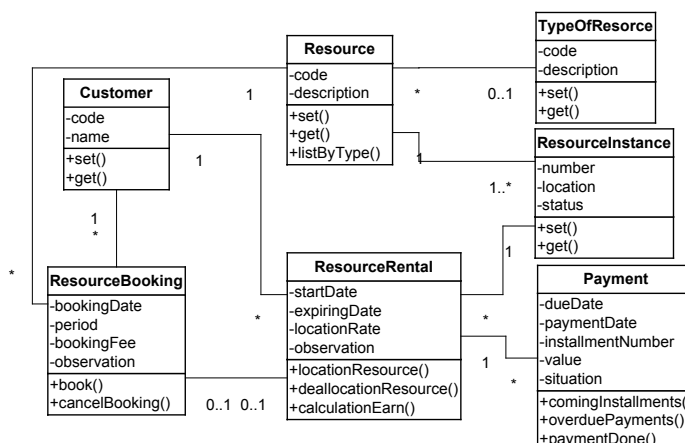


Figure 3 –Braga et al.'s “Resource Rental” pattern (1998)

4. Some Future Research Opportunities

In this section, we will address the future research opportunities of OOAP. In general, future research on OOAP can be undertaken in at least five areas:

4.1 Establishment of theoretical/empirical foundation for OOAPs

Most of the works on system analysis using pattern is based on practice. Although design patterns have emerged as a promising technique for improving the quality and reducing the cost and time of software development (Gamma *et al.*, 1995, Schmidt *et al.*, 1996), the objectives and requirements of system analysis are different from system design so the understanding of analysis patterns' impact remains fragmented and superficial. Besides a theoretical foundation could help place this line of study on a solid grounding, other kind of empirical study (e.g., Bolloju, 2004) can also help to explore the effectiveness of applying analysis patterns in helping the analyst.

4.2 Evaluation of OOAPs formalism

Communication and teamwork requires a common base vocabulary. Conventionally, patterns can provide such a common point of reference during the analysis phase of a project by adopting a well-known modeling language like Unify Modeling Language (UML) to describe successful solutions on common software problems. However, the current situation is that a series of research works on developing analysis patterns for the representation of conceptual models were inconsistent. There is no single agreed style or template for patterns, they can be described in a free form of narrative (Hay, 1995, Fowler, 1996) or in a template (Meszaros and Doble, 1997). As the definition of an object-oriented analysis pattern is still imprecise, there is no an agreed conclusion on the presentation of analysis patterns.

Besides the presentation problem, there is not a common viewpoint of similar problem among different pattern writers. In the pattern development process, different pattern writers may view similar business problem differently. One of the consequences of this phenomenon is that several patterns have been proposed to model the same problem. For example, three different kinds of "Account" pattern (i.e., Fowler, 1996, Fernandez and Liu, 2002, Fayad and Hamza, 2003) are all promised to model any kind of account in any application. Different patterns for the same problem complicates the use of analysis patterns, and hence, may discourage developers from reusing patterns of different structures (Hamza, 2004).

Because of the different levels of abstraction and different formats of presentation on OOAPs, it complicates the formalism of analysis patterns thereby diminishing reusability. This is because the reusability of a specific pattern is limited to those patterns that follow the same structure and abstraction. An appropriate form of OOAPs must be addressed so that pattern developers can develop formal models and system analysts can easily understand the patterns' semantics.

4.3 Assessment on the performance of using OOAPs

Different system analysts will structure the problem differently. Experts are prone to a top-down procedure while novices tend to use a bottom-up procedure (Mackay and Elam, 1992). Experts are able to group problems with the same underlying structure but novices group problems by surface characteristics and they usually lack a comprehensive plan. In fact, most of the related researches to date have little understanding about how pattern-based object-oriented methods are used by practicing professionals. These researches usually used novices because of existing practical constraints. Traditionally, people always argue that there are many difficulties in OO modeling such as steep learning curve, lack of domain knowledge, imprecise business rules involved, lack of modeling expertise, etc. As patterns are reusing already established design, system analysts suppose can get a head start on their problems.

System analysts can also get the benefit of learning from the experience of others. It is an interesting area to study whether patterns really can help novice and experienced analysts to reduce the difficulties in OO modeling.

4.4 Development of patterns for other analysis artifacts

Among those UML diagrams, the domain model plays an important role in the OOA process. It serves as a fundamental blueprint for the subsequent development activities. Owing to the importance of domain model, current works only focus on the development of OOAP using in this particular UML diagram. However, just a domain model cannot completely describe the whole analysis model. A complete analysis model also requires a dynamic model and functional model. In order to achieve the completeness of modeling using OOAP, pattern designer need to consider other form of patterns for different UML diagrams such as Use Case diagram, Sequence diagram and Statechart diagram, etc.

4.5 Measurement on the quality of OOAPs

The generality and understandability are among the main qualities that determine the usefulness of analysis patterns (Johannesson and Wohed, 1999). From a research perspective, user understanding may not be enough to measure the goodness of OOAPs. We also need alternative measures to evaluate the impact of using OOAPs. Further empirical researches can concentrate more on problem-solving measures (i.e., effectiveness and efficiency). One possible question is what are, in quantitative terms, the gains in productivity, quality, and time to analysis when using pattern-based object-oriented modeling?

5. Conclusion

The objective of applying meta research method on OOAP is to take stock and provide direction for future research, rather than to determine composing patterns to design software system is a must. Our experience has been that methods grow and mature by a combination of new ideas from the research community and stress testing on real world projects. Our expectation look forward to seeing how a variety of researches address these issues, and expect all the methods to improve and expand as a result. In general, there is a great interest of ongoing research in the area of OOAPs' foundation. Consequently, theoretical and empirical contributions are needed in order to formalize better OOAPs and also to understand the quantitative contribution of using them in conceptual modeling. We also expect future development of patterns on other kind of analysis artefacts. Otherwise, it will obstruct the future research development on this modeling technique.

6. References

- Beck, K. and Cunningham, W. 1987. Using Pattern Languages for Object-oriented Programs. Proceedings of the *2nd Annual Conference on Object-Oriented Programming Systems, Languages, and Applications*, Orlando, Florida, USA.
- Bolloju, N. 2004. Improving the quality of business object models using collaboration patterns. *Communications of the ACM*. **47** (7) 81-86.
- Braga, R. T. V., Germano, F. S. R. and Masiero, P. C. 1998. A Confederation of Patterns for Resource Management. Proceedings of the *5th Annual Conference on Pattern Languages of Programs*, Monticello, Illinois, USA.
- Coad, P., North, D. and Mayfield, M. 1997. *Object Models: Strategies, Patterns, and Applications (2nd ed.)*. Prentice Hall, Upper Saddle River, N.J.
- Coad, P. and Yourdon, E. 1991. *Object-Oriented Analysis*. Prentice Hall, Englewood Cliffs, N.J.

- Embley, D. W., Kurtz, B. D. and Woodfield, S. N. 1992. *Object-oriented Systems Analysis: A Model-driven Approach*. Yourdon Press, Englewood Cliffs, N.J.
- Fayad, M. E. and Hamza, H. 2003. The AnyAccount Pattern. Proceedings of the *10th Conference on Pattern Languages of Programs (PLoP'03)*, Monticello, Illinois.
- Fernandez, E. B. and Liu, Y. 2002. The Account Analysis Pattern. Proceedings of the *7th European Conference on Pattern Languages of Programs*, Irsee, Germany.
- Fernandez, E. B. and Yuan, X. 2000. Semantic Analysis Patterns. Proceedings of the *19th International Conference on Conceptual Modeling*, Salt Lake City, Utah, USA, 183-195.
- Fowler, M. 1996. *Analysis Patterns: Reusable Object Models*. Addison Wesley, Reading, Mass.
- Gamma, E., Helm, R., Johnson, R. and Vlissides, J. 1995. *Design Patterns: Elements of Reusable Object-oriented Software*. Addison-Wesley Pub Co, Reading, MA.
- Hamza, H. 2002. *A Foundation for Building Stable Analysis Patterns*. Master thesis. Dept. of Computer Science & Engineering, University of Nebraska, Lincoln, USA.
- Hamza, H. S. 2004. Improving Analysis Patterns Reuse: An Ontological Approach. Proceedings of the *19th Annual ACM Conference on Object-Oriented Programming, Systems, Languages, and Applications*, Vancouver, British Columbia, Canada.
- Hay, D. C. 1995. *Data Model Patterns: Conventions of Thought*. Dorset House Publishing Company, New York.
- Johannesson, P. and Wohed, P. 1999. The Deontic Pattern - a Framework for Domain Analysis in Information Systems Design. *Data & Knowledge Engineering*. **31** (2) 135-153.
- Mackay, J. M. and Elam, J. J. 1992. A Comparative Study of How Experts and Novices Use a Decision Aid to Solve Problems in Complex Knowledge Domains. *Information Systems Research*. **3** (2) 150-172.
- Marsura, P. (1998), Vol. 2004.
- Meszaros, G. and Doble, J. 1997. A pattern language for pattern writing. In Martin, R. C., Riehle, D. and Buschmann, F. (eds.) *Pattern Languages of Program Design 3*. Addison-Wesley, Reading, Mass, 529-574.
- Nicola, J., Mayfield, M., Abney, M. and Abney, M. 2001. *Streamlined Object Modeling: Patterns, Rules, and Implementation*. Prentice Hall PTR, Upper Saddle River, N.J.
- 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) 269-290.
- Robertson, S. and Strunch, K. 1993. Reusing the Products of Analysis. Proceedings of the *2nd International Workshop on Software Reusability*, Lucca, Italy.
- Rumbaugh, J. R., Blaha, M. R., Lorensen, W., Eddy, F. and Premerlani, W. 1991. *Object-Oriented Modeling and Design*. Prentice Hall, Englewood Cliffs, N.J.
- Schmidt, D. C., Johnson, R. E. and Fayad, M. 1996. Software Patterns. *Communications of the ACM*. **39** (10) 37-39.
- Yacoub, S. M. and Ammar, H. H. 2003. *Pattern-Oriented Analysis and Design: Composing Patterns to Design Software Systems*. Addison-Wesley Pub Co., Reading, Mass.