

IMPROVING THE QUALITY OF BUSINESS OBJECT MODELS USING COLLABORATION PATTERNS

Project managers and systems analysts can tackle the challenges of representing system requirements using the object-oriented approach.

<< BY NARASIMHA BOLLOJU >>

Business object models are conceptual models built using object-oriented concepts during systems analysis to represent the requirements of an information system. Developing business object models is a difficult activity for most systems analysts because it demands both modeling experience and domain knowledge. As a result, the object models can be incomplete and possibly erroneous representations of business requirements, particularly when novice analysts perform the activity. Many studies have identified incomplete requirements as a major factor contributing to project failures [1]. Software project managers and systems analysts can benefit from any assis-

tance in addition to that provided by systems development methodologies. Essentially, they need resources to help them deal with the difficulties associated with capturing and representing information systems requirements and help them build better quality business object models.

Many practitioners advocate the application of patterns for efficient and effective business object modeling [5, 8]. Patterns provide generalized solutions to common problem situations and can be adapted to specific problem situations. The application of patterns became popular in the object community after the first conference on pattern languages in 1994 and the publication of Gamma

et al.'s book on design patterns [9]. Benefits from pattern applications in various activities of object-oriented systems development include more rapid underlying activity, more opportunities for reuse, and improved communication and documentation. The object community, recognizing the potential of patterns, has been active in creating numerous patterns in different categories, such as in programming for coding elegant, readable, and efficient programs [7]; in design for creating reusable object-oriented designs [9]; in architecture for describing and documenting large-scale applications [2]; and in analysis for developing business object models [4, 5, 8, 10].

Analysis patterns are designed to assist systems analysts in capturing information systems requirements, with object model fragments representing commonly occurring groups of classes and relationships in various business domains. Major developments in analysis patterns can be attributed to Peter Coad et al. [4–6] and Martin Fowler [8]. Coad's initial work on object-oriented patterns provided examples of simple analysis patterns and showed how they could be combined for building larger models [4]. An illustration of how patterns speed up the modeling process can be observed in [5] where the authors used appropriate analysis patterns, from a set of 31, for developing five different object models. Coad et al. [6] introduced the notion of archetypes and identified four types of interconnected archetypes—moment-interval, role, description, and party-place-or-thing—forming a generic analysis pattern called a domain-neutral component. While the types of analysis patterns identified by Coad et al. dealt mostly with four generic classes—people, places, things, and events—and relationships among them—transactions and roles—the types of patterns developed by Fowler [8] are more specific and closer to functional areas of business domains, such as organization structures, planning, inventory, accounting, and trading. Despite the difference in orientation between these two pattern types, their applications are expected to assist in the process of business object modeling.

Even with ample evidence of the benefits and limitations of design patterns in object-oriented systems development [3, 11], the contribution of analysis patterns to business object modeling has not been investigated empirically (with the exception of a recent publication by Purao et al. [12], in which the authors

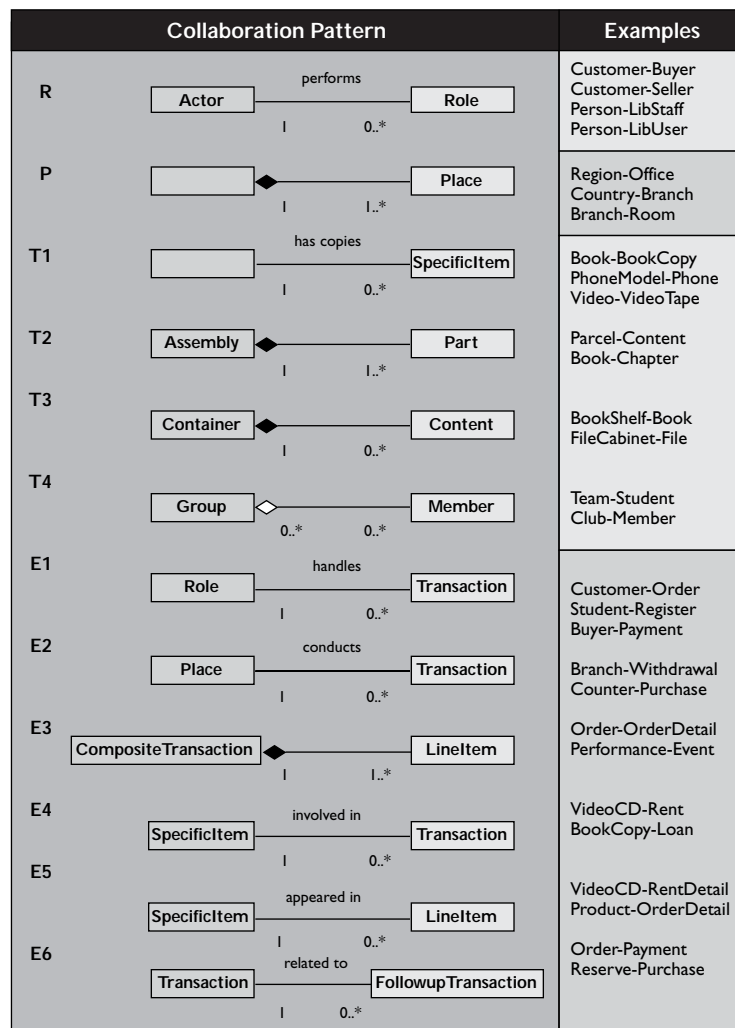


Figure 1. Sample applications of proposed and empirically tested a collaboration patterns [8].

methodology involving reuse of analysis patterns in conceptual design). This study aims to help understand how collaboration patterns can assist in enhancing the quality of business object models developed by novice systems analysts. Based on the results of the study, I offer project managers and systems analysts recommendations on the effective application of analysis patterns for business object modeling.

Collaboration patterns, as defined and described in [10], involve pairs of classes belonging to the four types of pattern players: people, places, things, and events. Nicola et al. [10] transformed the 31 object modeling patterns presented in [5] into a set of 12 primary patterns (see Figure 1). Based on their experience in consulting and development work using them, the authors asserted they represent a set of irreducible patterns and that all other patterns can be derived either by snapping together patterns with a common pattern player or by merging objects that act as several different pattern players. Generalization-specialization relationships are

excluded from this set because no collaboration exists among subclasses and superclasses, and it is possible that classes representing any of the four types of pattern players can be specialized independently. Collaboration patterns and other types of analysis patterns can be used either for developing business object models (see the sidebar for an illustration) or for improving initial versions of object models, which is the focus of the study.

To study the effect of applying collaboration patterns on the quality of object models, I examined the project work submissions of 13 student teams (47 students) in a final-year undergraduate course on object-oriented analysis and design. All students completed a course on structured systems analysis and design in their second year. As part of their project work, the student teams prepared and submitted object models for the project topic assigned to them. During a 45-minute tutorial session, preceded by a 20-minute introduction to patterns, each team applied collaboration patterns and identified changes to their object model. Each team selected and ranked five key classes from their original model to start the pattern application process. Table 1 lists the projects, the number of students in each team, and the number of classes used in the object model. One member of each team noted the details of each pattern application, including a comment describing the need for that application. Each team also provided written feedback on its experience applying patterns during the session.

Team Performance

The case study material collected from each team at the end of the session included four components: a one-

	Project Topic	# Students in Team	# of Classes
1	Credit-Card System at a Bank	4	13
2	International Freight Service	5	24
3	Hotel Operations	4	18
4	Fast-Food Outlet Chain	3	15
5	Movie Theater Ticketing	3	23
6	Public Road Transport	4	11
7	Public Utility—Electricity	3	16
8	Airline Reservations	3	15
9	Hospital Operations	4	26
10	Department Store	4	21
11	Travel Agency	3	17
12	Mobile Telecom Company	3	17
13	Insurance Company	4	25

Table 1. List of projects.

Pattern	# of applications	% of applications
E1: Role-Transaction	58	33.1
E6: Transaction-FollowupTransaction	21	12.0
E5: SpecificItem-LinItem	20	11.4
E3: CompositeTransaction-LinItem	12	6.9
E2: Place-Transaction	11	6.3
R: Actor-Role	10	5.7
T1: Item-SpecificItem	10	5.7
T4: Group-Member	9	5.1
E4: SpecificItem-Transaction	9	5.1
T2: Assembly-Part	7	4.0
P: OuterPlace-Place	5	2.9
T3: Container-Content	3	1.7
	175	100.0

page list of business requirements prepared and used in developing the original object model for each project, the original object model, a list of pattern applications identifying changes to the original object model, and feedback on the team's experience in applying patterns. I analyzed each pattern application using all the related material to determine its effect on the original object model.

Most teams applied a variety of collaboration patterns to revise their object models. On average, each team used about 19 patterns (minimum of 11, maximum of 26). Table 2 shows the distribution of different types of patterns applied by all the teams. Pattern E1 was the most commonly used pattern (about a third of the total number of patterns used by all teams), and patterns P and T3 were the least used.

Each pattern application resulted in improvement to the original model, introduction of a new problem (negative effect), or an insignificant change to the original object models. About 19% of the 175 pattern applications belonged in the last category. The pattern application turned out to be simple documentation of an existing collaboration, sometimes with an insignificant change, such as renaming an association to reflect the change in reading direction, and thus made no difference to the original models. All such pattern applications were removed from the subsequent analysis. Some pattern applications resulted in more than one change, such as renaming an existing association and changing its multiplicity at the

Effect	# of observations	% of observations	# of projects observed
Identified new classes and associations	31	19.9	7
Changed association multiplicity	28	17.9	7
Identified new associations	17	10.9	7
Renamed existing associations	16	10.3	7
Swapped association multiplicities	10	6.4	5
Changed association to aggregation	7	4.3	3
Identified new redundant associations	11	7.1	4
Improperly used aggregation	8	5.1	3
Identified improper classes	6	3.8	3
Others (either positive or negative)	22	14.1	13
	156	100	

Table 3. Effects of pattern applications.

same time. A summary of different effects obtained from all 13 projects is presented in Table 3. Details of the effects observed across at least three different teams with possible sources of those effects are presented as follows:

New classes and associations. As anticipated, the most positive effect of pattern application was the identification of new classes and the relation of these new classes with existing classes. Although most patterns contributed to this effect, patterns T1, T4, and R were used in over 50% of the applications in this category. Pattern T1 helped separate multiple copies or instances of an object. Pattern T4 helped in finding compositions of teams and groups. Pattern R often resulted in differentiating role from actor, and in turn made it possible to represent multiple roles of an actor. The overall effect can be summarized as finding new classes and associations through elaboration of things and roles.

Alteration in association multiplicity. This effect includes changes made to multiplicity values on one or both ends of association and excludes the minor change from * to 0-* on any end of the association and reversing multiplicity values (such as from 1 to * to * to 1). Nearly 90% of these changes are contributed by patterns (in descending order of usage frequency) E1, E5, and E3, which included changes from * to 1-*, from 1 to *, and from * to 1. It appears that these patterns forced the teams to critically analyze each association between collaborating classes and revise the multiplicity values to make them more accurate.

New associations. About 90% of the changes of this type resulted in identifying associations between existing classes using patterns (in descending order of usage frequency) E1, E6, and E2. Pattern E1 helped in identifying missing associations between classes representing role and transaction—which were sometimes shown indirectly in original object models. Pattern E6 was helpful in linking transaction classes with follow-up transaction classes. Application of pattern E2 resulted in linking transaction classes with place classes. The net effect is best described as anchoring transaction classes to other classes representing role (who), follow-up transaction (next), and place (where).

Renaming of existing associations. This effect includes both renaming existing associations and naming associations not named in the original object model. Over 80% of the changes of this type resulted from the applications of patterns E1 and E6. The majority of the changes were related to the associations between role and transaction. The results enhanced the semantics of associations.

Swapping of association multiplicities. This effect includes only 1 to * associations, where the multiplicity values are simultaneously changed on both ends. About 50% of the changes resulted from the application of pattern E5 and E4, where the multiplicity value of 1 on the side of a specific item is changed to *. It appears that 5 of the 13 teams made this change because they used a chain (or sequence) of 1 to * associations ending at a

specific item (instead of reversing the last 1 to * association).

Altered association to aggregation. Application of pattern E3 usually resulted in this change, in which teams made the association between a composite transaction and its detail line item.

While these changes resulted in improvements in object models, the following three changes contributed negatively to the object model:

Identification of redundant associations. Pattern E1, observed to have contributed to most of the positive results, misled teams into identifying redundant associations. This could be due to the fact that the pattern application procedure being employed did not facilitate the removal of existing associations. Curiously, most of these redundant associations appeared between a role class and subclasses of transactions.

Improper use of aggregation. This group of changes is related to the use of aggregation in place of a typical 1 to * association. Applications of patterns T4 and E3 contributed most to this negative effect. This result also illustrates the common difficulty in selecting between association and aggregation to represent 1 to * relationships.

Identification of improper classes. This category of changes includes those classes that are required, at least at the analysis level, in the business object model. Application of pattern E1 contributed to this problem when teams proposed additional classes for either role or transaction. If attributes of various classes were also considered part of the application, this result might not have been present.

Analysis of written feedback collected from the teams regarding their experience in applying patterns identified how the 45-minute session helped them improve object models and their modeling skills, as well as some limitations in the set of patterns used. All the teams commented on the benefits (39 comments from 13 teams), and some teams mentioned limitations of patterns (eight comments from four teams). Most teams found that pattern application is an easy-to-use technique that provides guidance in identifying classes and associations and correcting multiplicity of associations. One team noted that dividing patterns into four categories helped it identify different types of relationships in a more systematic way. Members of another team observed that by following the procedure, they were “less likely to omit important transaction classes” and found it “easier to ensure completeness.” Yet another team noted the exercise enabled it to get “a clear picture of the daily operations of the company” (problem domain knowledge) and assisted in the critical analysis of the object model to identify improvements.

Regarding the limitations of patterns, one team expressed uncertainty as to “whether all relationships are identified.” Members of another team commented that there are “too many possible relationships to be considered” and that they “might miss some relationships in the process (of identifying relevant relationships).” While one team complained that the set of patterns was limited to represent a specific relationship, another team solved that problem in its original object model by introducing a new class. The former team tried to represent collaboration between Location (a place) and CreditCard (a thing), although no pattern was available. The latter team, upon finding no pattern available to represent a Supplier-Item (role-thing) collaboration, introduced a new class, Supply (a transaction), and then established two collaborations between Supplier-Supply and Item-Supply.

Conclusion

The findings from this study highlight the effects of applying collaboration patterns to improve business object models by identifying missing classes, associations, and aggregations. Other improvements include the assignment of proper names and multiplicity values for associations. Despite some minor negative effects of pattern application, significant improvements in the quality of object models were observed across all 13 projects. Pattern application procedure was also found to be an efficient and effective technique for identifying classes and relationships, as well as for ensuring their completeness and for fixing problems in the original object models. It is also worth noting that these effects of pattern application both on the quality of object models and on learning were achieved in a short session of less than an hour. It is possible that the extent of benefits observed in this study might be attenuated when experienced analysts are involved in modeling large-scale applications. Analysis patterns, however, could benefit even experienced analysts with respect to reuse, communication, and documentation, particularly when business object models are to be completed with such details as attributes and services to capture and represent business processes and rules.

The following recommendations for systems analysts relate to gathering and organizing business requirements using the object-oriented approach, and for bringing improvements to initial versions of business

object models. For developing new object models, analysts can start with a few important classes representing problem domain objects, then select collaboration patterns from Table 2 (in order of usage frequency) for identifying other classes and collaborations. Following this procedure, analysts can efficiently gather requirements from clients considering all possible collaborating classes and their collaborations and thereby minimize the risks associated with incomplete systems requirements.

Regarding object models, four guidelines can help improve the quality of initial versions of business object models:

- Elaborate the classes representing things using patterns T1, T2, T3, and T4 and classes representing roles using pattern R to identify new classes, and associate those classes with existing classes;
- Verify association multiplicities and names using patterns E1, E35, and E53. Check for possible cases of reversed multiplicity when noticing pattern E5 in the object model;
- Identify any missing associations using patterns E1, E2, and E6. Look for redundant associations (such as a specific role handling a transaction and a follow-up transaction where the association between role and follow-up transaction is redundant) and remove them; and
- Identify aggregations or change associations to aggregations using the pattern E3. Avoid defining an aggregation relationship between classes belonging to different types (such as Place and Transaction), other than those identified by E3.

Considering the typically short time frames available for application development, pattern application can help project teams and managers capture system requirements. Using patterns, as observed in the case of design patterns, can also enhance communication with clients and communication among project teams and thereby minimize the risks associated with improper business object models. Project managers must therefore prepare, with the help of senior analysts, a small set of analysis patterns based on types of applications being considered and develop guidelines for using these patterns for gathering and organizing business requirements. They should also encourage all team members to use the identified patterns and define or derive new

Developing a Business Model of a Video Store System Using Collaboration Patterns

Classes relevant to model a video rental store system include people, such as customers and staff; places, such as stores; things, such as videotapes and DVDs; and events, such as rentals, returns, and payments.

Collaboration patterns can be applied for developing business object models starting with a few important classes, such as Rental, because it captures a key business event in a video rental store. For example, pattern E1 can be applied to identify the class **CounterStaff** (Role) and capture its association with **Rental** (Transaction). Starting with such a simple model segment, appropriate patterns can be repetitively used to expand the model by adding other classes and different types of relationships (associations and aggregations). For example, starting with **Rental** class patterns E2, E3, and E6 can be applied to identify **Store** (Place), **RentalDetail** (LineItem), and **Payment** (FollowupTransaction) classes respectively and link these classes to **Rental**. It is also possible to name the associations and specify relevant multiplicity values, as relationships between classes are identified.

patterns or pattern compositions suitable for applications within the organization.

Although the focus of this article is on improving the quality of models defined using object-oriented concepts, the pattern application and the associated benefits are also relevant to conceptual data modeling activities, such as entity-relationship modeling. This study also provides clues about common errors committed by novice systems analysts in business object modeling. The results, therefore, can be employed to address difficulties in teaching and learning object modeling.

Overall, using analysis patterns in object modeling results in better business object models that represent business requirements more accurately and completely. The application of such patterns can potentially save the time and effort required to incorporate missing requirements in the later phases of systems development. **C**

REFERENCES

1. The Chaos Report. The Standish Group, 1994; see www.standishgroup.com/sample_research/chaos_1994_1.php.
2. Buschmann, F., Meunier, R., Rohnert, H., Sommerlad, P., and Stal, M. *Pattern-Oriented Software Architecture: A System of Patterns*. John Wiley & Sons, Ltd., Chichester, U.K., 1996.
3. Cline, M.P. The pros and cons of adopting and applying design patterns in the real world. *Commun. ACM* 39, 10 (Oct. 1996), 47–49.
4. Coad, P. Object-oriented patterns. *Commun. ACM* 35, 9 (Sept. 1992), 152–159.
5. Coad, P., North, D., and Mayfield, M. *Object Models: Strategies, Patterns, & Applications, 2nd Ed.* Prentice Hall, Upper Saddle River, NJ, 1997.
6. Coad, P., Lefebvre, E., and De Luca, J. *Java Modeling in Color with UML: Enterprise Components and Process*. Prentice Hall, Upper Saddle River, NJ, 1999.
7. Coplien, J.O. and Schmidt, D.C. (Eds.) *Pattern Languages of Program Design*. Addison-Wesley, Reading, MA, 1995.
8. Fowler, M. *Analysis Patterns: Reusable Object Models*. Addison-Wesley, Menlo Park, CA, 1997.
9. Gamma, E., Helm, R., Johnson, R., and Vlissides, J. *Design Patterns: Elements of Reusable Object-Oriented Software*. Addison-Wesley, Reading, MA, 1995.
10. Nicola, J., Mayfield, M., and Abney, M. *Streamlined Object Modeling*. Prentice Hall, Upper Saddle River, NJ, 2002.
11. Prechelt, L., Unger-Lamprecht, B., Philippsen, M., and Tichy, W.F. Two controlled experiments assessing the usefulness of design pattern documentation in program maintenance. *IEEE Trans. on Software Engineering* 28, 6 (June 2002), 595–606.
12. Purao, S., Storey, V.C., and Han, T. Improving analysis pattern reuse in conceptual design: Augmenting automated processes with supervised learning. *Information Systems Research* 14, 3 (Sept. 2003), 269–290.

NARASIMHA BOLLOJU (narsi.bolloju@cityu.edu.hk) is an associate professor in the Department of Information Systems at the City University of Hong Kong.

© 2004 ACM 0001-0782/04/0700 \$5.00
