8-15-1997

An Approach to Transformational Reengineering of SSADM Application Specifications to Object-Oriented Specifications

Narasimha Bolloju
City University of Hong Kong, isnarsi@cityu.edu.hk

An Approach to Transformational Reengineering of SSADM Application Specifications to Object-Oriented Specifications

Narasimha BOLLOJU
Department of Information Systems, City University of Hong Kong
HONG KONG
E-mail: isnarsi@cityu.edu.hk

Abstract

Many organizations are planning for transition to object-oriented software development for savings in development and maintenance costs. Appropriate tools and techniques to support this process of transition will help such organizations in overcoming many problems associated with the transition process. This paper outlines an approach to transformational reengineering of SSADM application specifications to Object-Oriented specifications.

Introduction

Object-oriented software development (OOSD) promises substantial savings in development and maintenance costs (Fayad *et al* 1996). A major difficulty in transition to OOSD is related to the understanding of the culture change associated with the migration. Fayad *et al* (1996) present a detailed approach on managing the transition of development teams to OOSD. They observed that on average development teams would require about one year for the transition to OOSD. Appropritate tools and techniques supporting the migration process will minimize many such difficulties.

Many organizations have been using the Structured Systems Analysis and Design Method (SSADM) for computer-based information systems development with professionals educated and trained in structured methodologies. It is expected that these organizations would be planning for a transition to Object-Oriented software development (OOSD) to realize the benefits in the form of savings in development and maintenance costs.

This paper presents outlines an approach to this transformation process. This research has implications for practitioners in planning for migration to OOSD and for researchers in addressing various problems associated with the transformation of specifications made using structured methods to OO specifications.

Transformational Reengineering

Reengineering of existing applications to exploit new techniques and technologies has been an active research area in recent years. The complete transformation of application systems includes reverse engineering of code to analysis products, transformation of analysis products to new paradigm (e.g., object-oriented) products and forward engineering of transformed analysis products to code.

Wang and Archer (1994) state that a difficulty in object-oriented analysis is the lack of tools to support the "artistic" or unstructured work of identifying class inheritance structures required for reuse. They present a semi-structured approach based on a pattern-matching algorithm supported by semantic analysis to aid systems analysts. Reporting on the reverse engineering of relational databases to object models, Premerlani and Blaha (1994) suggest that it is important to recognize generalization early in the reverse-engineering process. Gall and Klosch (1995) present an approach to object identification in procedural programs integrating human expertise and external domain and application specific knowledge. Sneed and Nyary (1995) also support idea of using different software components (e.g., maps or panels for communicating with the user, databases, job control procedures, programs) for identifying objects in existing systems.

Leite *et al.* (1995) report results of their research on recovering high level abstractions (business rules and domain facts) from structured specifications. Many CASE tools support transformation to a single target

methodology (Fayad *et al.* 1996). Newcomb and Kotik (1995) present a reengineering approach to transform COBOL programs into functionally comparable object-oriented systems based on Hierarchical Object-Oriented State-Machine Model. Liao *et al.* (1996) describe their experience with reengineering of an existing SSADM-designed system to Booch's object-oriented method using a real-life retailing application. A detailed account of advantages and disadvantages of this reengineering process is given based on their experience in manually transforming SSADM-designed system to Booch's object-oriented method (Booch 1994).

Blaha *et al.* (1994) describe an approach for mapping object-oriented models to relational data models and describe a tool to support this mapping. They present the rules for converting object models to relational models and discuss possible ways of mapping inheritance relationships to relational models. Bolloju and Toraskar (1996) present an approach for effective mapping of object models to relational models. This approach illustrates how the systems analysed using object-oriented techniques can efficiently use the popular relational database management systems for implementation.

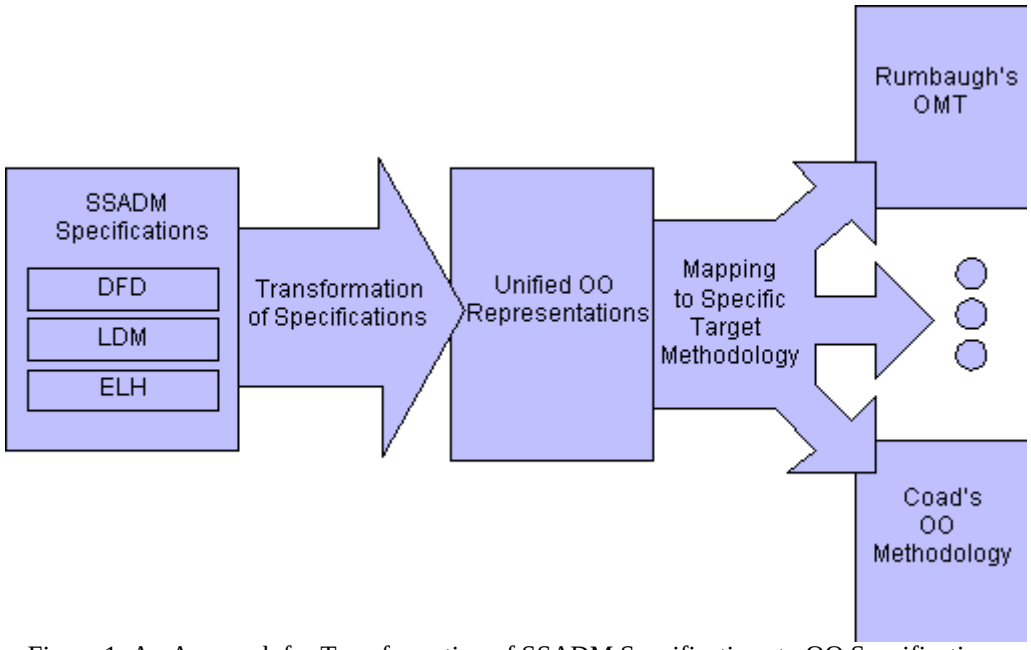

Figure 1: An Approach for Transformation of SSADM Specifications to OO Specifications

An Approach to Transformational Reengineering

This paper outlines an approach to the transformation of structured application systems specifications to object-oriented specifications with emphasis on the transformation to multiple target OO specifications. The proposed approach is illustrated in Figure 1 for transformation of SSADM-based application system specifications.

Fayad *et al* (1996) stress the transformation of an application system to multiple target methodologies should be supported by a CASE tool. Based on the current trends in the standardization of OO methodologies, we need to consider methodologies such as Object-Modeling Technique (Rumbaugh *et al.* 1991), Coad-Yourdon Methodology (Yourdon 1994), Unified Modeling Language (proposal of Booch-Rumbaugh-Jacobson) and Open Modeling Language (Open Consortium of about a dozen companies) as the target methodologies. These four methodologies cover a broad spectrum of OO methodologies and it is expected that many of the problems associated with transformation will be dealt with using such a scheme. Subsequently, it should be relatively easier to extend this approach to include other target methodologies.

Effective mapping of various SSADM analysis stage products such as Logical Data Model (LDM), Data Flow Diagrams (DFD), and Entity Life Histories (ELH) to the equivalents of multiple target OO

methodology products is an important issue for research. Mapping possibilities need to be identified for selection between the extremes of complete automation and manual process of mapping along with the preconditions. Definition of mapping rules should be made based on various forms of dependencies between the target and SSADM analysis stage products. Research findings and experiences related to the transformations (Gall and Klosch 1995; Leite *et al.* 1995; Newcomb and Kotik 1995; Premerlani and Blaha 1994; Wang and Archer 1994; Bolloju and Toraskar 1996; Liao *et al.* 1996) will be useful in defining the mapping process.

Proper identification of objects from LDM and DFDs, and inheritance relationships (generalization-specialization hierarchies) to define class hierarchies is necessary for reusability since it contributes directly to the savings in development and maintenance costs. The level of automation or intelligent support that can be provided in the transformation process will ultimately establish the practical utility of the proposed approach.

Summary

An approach to transformational reengineering of SSADM application specifications to Object-Oriented specifications is outlined in this paper. Issues related to the selection of target methodologies, definition of mapping rules, and identification of reuse are discussed. Research in this direction is expected to contribute to a mapping theory for transformation of SSADM analysis stage products to equivalents of popular OO methodologies. Areas such as the correctness of mapping, ability to identify reuse opportunities, and the level of automation achievable with respect to each of the target object-oriented methodologies are some potential candidates for further research. Any tools for transformation based on this approach can also support the education and training in migration of information systems professionals trained in SSADM or other conventional structured systems analysis methods to object-oriented software development.

References

1. Blaha, M., Premerlani, W., and Shen, H. (1994), Converting OO models into RDBMS schema. IEEE Software, May, 28-39.
2. Bolloju, N. and Toraskar, K.(1996), Data Clustering for Effective Mapping of Object Models to Relational Models, *Journal of Database Management*, (accepted for publication).
3. Booch, G. (1994), *Object-Oriented Analysis and Design with applications*, Benjamin/Cummings.
4. Fayad, M.E., Tsai, W.T., and Fulghum, M.L. (1996), Transition to Object-Oriented Software Development, *Communications of the ACM*, Vol. 39, No. 2, pp. 108-121.
5. Gall, H. and Klosch, R. (1995), "Finding Objects in Procedural Programs: An Alternative Approach" in the *Proceedings of the Second Working Conference on Reverse Engineering*, Wills, L., Newcomb, P. and Chikofsky, E. (Eds.) IEEE Computer Society Press, pp. 208-216.
6. Leite, J.C.S.P. and Cerqueira, P.M. (1995), "Recovering Business Rules from Structured Analysis Specifications" in the *Proceedings of the Second Working Conference on Reverse Engineering*, Wills, L., Newcomb, P. and Chikofsky, E. (Eds.) IEEE Computer Society Press, pp. 13-21.
7. Liao, S.Y., Shao, Y.P. and Tsang, W.H. (1996), SSADM-Designed System to Object-Oriented System: A Re-engineering Approach, *Journal of Object-Oriented Programming* (accepted for publication).
8. Newcomb, P. and Kotik, G. (1995), "Reengineering Procedural Into Object-Oriented Systems" in the *Proceedings of the Second Working Conference on Reverse Engineering*, Wills, L., Newcomb, P. and Chikofsky, E. (Eds.) IEEE Computer Society Press, pp. 237-249.
9. Premerlani, W. and Blaha, M. (1994), An approach for reverse engineering of relational databases, Communications of the ACM, 37(5):42-49, 134.
10. Rumbaugh, J., Blaha, M., Premerlani, W., Eddy, F., and Lorensen, W. (1991), Object-Oriented Modeling and Design, Prentice-Hall.

11. Sneed, H.M. and Nyary, E. (1995), "Extracting Object-Oriented Specifications from Procedurally Oriented Programs" in the *Proceedings of the Second Working Conference on Reverse Engineering*,
12. Wang, S. and Archer, N.P. (1994), "Identifying inheritance structure in object-oriented systems analysis: A pattern matching approach", *Journal of Object-Oriented Programming*, May, pp. 47-55.
13. Wills, L., Newcomb, P. and Chikofsky, E. (Eds.) IEEE Computer Society Press, pp. 217-226.
14. Yourdon, E. (1994), *Object-Oriented Systems Design: An integrated approach*, Prentice-Hall.