

Reuse and the Unified Modeling Language (UML)

By Sinan Si Alhir (January 17, 1999)
Updated January 17, 1999

Abstract

The Unified Modeling Language (UML) is a modeling language for specifying, visualizing, constructing, and documenting the artifacts of a system-intensive process. It was originally conceived by Rational Software Corporation and three of the most prominent methodologists in the information systems and technology industry, Grady Booch, James Rumbaugh, and Ivar Jacobson (the Three Amigos). The language has gained significant industry support from various organizations via the UML Partners Consortium and has been submitted to and approved by the Object Management Group (OMG) as a standard (November 17, 1997).

This paper elaborates on *reuse* and the application of the UML.

Contents

- [Introduction](#)
 - [System Development](#)
 - [Reuse Culture](#)
 - [Reuse Strategy](#)
 - [The UML](#)
 - [Conclusion](#)
 - [References](#)
-

Introduction

Businesses define success as achieving profitability via strategic initiatives that maximize their return-on-investment, where people, processes, and tools/technology are strategic weapon in enabling organizations to have a competitive advantage in achieving their business goals. In today's a highly volatile environment involving a global, vigorously competitive, and unpredictable market filled with opportunities and risks resulting from constant change and ever-increasing complexity, businesses can only succeed by addressing market challenges and competitively achieving business goals. Competitiveness (or having a competitive advantage) involves the ability of an organization to proactively minimize risks and capitalize on opportunities while, not simply sustaining, but thriving in a highly volatile environment and adding value by increasing quality, reducing costs, and reducing time-to-market. Fundamentally, productively delivering quality in a timely and cost-effective manner.

These challenges are best addressed by applying the fundamental engineering principle of reuse — do not reinvent, but adapt existing proven quality elements, and construct new quality elements only as necessary — and the fundamental natural principle of evolution — do not simply adapt, but develop from adapting via a process of adapting and cultivating the ability to better adapt by addressing immediate needs and continuously maturing to anticipate and be better positioned to address immanent needs.

The **UML** is an evolutionary general-purpose, broadly applicable, tool-supported, and industry-standardized modeling language for specifying, visualizing, constructing, and documenting the artifacts of a system-intensive process. The language is broadly applicable to different types of systems (software and non-software), domains (business versus software), and methods and processes. The **UML** enables and promotes (but does not require nor mandate) a use-case-driven, architecture-centric, iterative, and incremental process that is object oriented and component based. The **UML** enables the capturing, communicating, and leveraging of knowledge: models capture knowledge (semantics), architectural views organize knowledge in accordance with guidelines expressing idioms of usage, and diagrams depict knowledge (syntax) for communication.

Achieving reusability in system development involves an *appropriate* culture and strategy.

[Return to Contents](#)

System Development

System development may be characterized as problem solving, including understanding or conceptualizing a problem, solving the problem, and realizing the solution. Conceptualizing a problem involves understanding the problem and representing it in some form. Solving the problem involves manipulating representational constructs from the problem domain and the solution domain to derive a representation of the desired solution. Realizing a solution involves implementing the solution by mapping the representational constructs of the solution into a usable form. Paradigms determine the possible types of representations that may be used in such a process. The following paradigms are common:

- The object-oriented paradigm focuses on constructing reusable units and encompasses the conceptualization and specification principles of abstraction, encapsulation, inheritance, and polymorphism.
- The component-oriented paradigm focuses on the assembly of reusable units and encompasses the specification and realization principles of components, interfaces, and infrastructure.

[Return to Contents](#)

Reuse Culture

Reuse is the promised holy grail of object orientation that enables us to improve value by increasing quality, reducing costs, and reducing time-to-market. Technology and processes enable people to practice reuse; however, enabling reuse does not ensure that reuse will occur. An organizational culture is required to promote reuse. Such a culture encompasses the following:

- A standardized language for communication -- the **UML**.
- A paradigm optimized for the conceptualization and specification phases of problem solving -- object orientation.
- A paradigm optimized for the specification and realization phases of problem solving -- component-based development.
- Other paradigms supporting higher-level concepts (frameworks, patterns, etc.).
- Processes, people, and tools that establish a cohesive reuse strategy.

[Return to Contents](#)

Reuse Strategy

While a reuse culture establishes the foundation for reuse, a reuse strategy encompassing the following is required to actualize the benefits of reuse:

- A vision supported by an infrastructure.
- A repository for housing and managing assets (reusable artifacts).
- Reuse champions, mentors, and evangelists chartered to promote and be catalysts for reuse.
- A standard for defining assets.
- A quality system for evaluating the quality of assets.
- A metric system for evaluating asset reuse.
- An incentive system encouraging the realization of reusable artifacts and asset reuse.
- Processes and methods focusing on reuse throughout the system development life cycle.
- Tools that enable execution of the strategy.

[Return to Contents](#)

The UML

The language used within a culture provides the means for communication and collaboration among members of the culture, and establishes the foundation upon which evolution and reuse may be achieved. In our attempts to "industrialize" system development through object orientation and component-based development, significant focus is placed on tools and processes; however, minimal emphasis is placed on the development culture. To realize the benefits of reuse, development cultures must promote reuse, and because the UML is an evolutionary general-purpose, broadly applicable, tool-supported, and industry-standardized modeling language, it may be used as the foundation for such a culture.

[Return to Contents](#)

Conclusion

The UML has the potential to enable us to realize the holy grail of object orientation and component-based development — reuse. However, it is experience and gradual adoption and *evolution* of the standard that will reveal its true potential and enable organizations to realize its benefits.

[Return to Contents](#)

References

[Alhir]

Sinan Si Alhir. "UML in a Nutshell : A Desktop Quick Reference". O'Reilly Associates, Inc., 1998.

[Return to Contents](#)

Email salhir@earthlink.net
Web Site home.earthlink.net/~salhir
Copyright © 1999 Sinan Si Alhir. All rights reserved.