

The Object–Oriented Paradigm

By Sinan Si Alhir (October 23, 1998)
Updated October 23, 1998

Abstract

The object–oriented paradigm is a concept–centric paradigm encompassing the following pillars (first principles): abstraction, encapsulation, inheritance, and polymorphism. It was originally conceived by three of the most prominent Greek philosophers, Socrates (470–399 BC), Plato (428–348 BC), and Aristotle (384–322 BC) in the theory of Forms. The paradigm has evolved since its original conceptualization to become a pivotal force in the evolution of science, technology, and any other domain in which it is applied.

This paper elaborates on the *definition* of the object–oriented paradigm.

Contents

- [Introduction](#)
 - [The Theory of Forms](#)
 - [The Real World](#)
 - [The Object–Orientated Paradigm](#)
 - [Objects and Classes](#)
 - [Links and Associations](#)
 - [Scenarios and Interactions](#)
 - [The Theory of Forms and the Object–Oriented Paradigm](#)
 - [Conclusion](#)
 - [References](#)
-

Introduction

Problem solving involves understanding or conceptualize a problem, solving the problem, and implementing or realizing the solution. Conceptualizing a problem involves representing the problem using *representational constructs* (mental notions or ideas). 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 mapping those representational constructs of the solution unto the solution world, that is, constructing the solution. The use of representational constructs is a very natural process that often occurs subtly and sometimes unconsciously in problem solving. Underlying this scheme is the use of a paradigm in determining the possible types of representations utilized in problem–solving efforts.

The object–orientated paradigm is derived from the convergence of other fundamental paradigms, and is reducible to the other paradigms as required by its application via a language or method. The flexibility provided by utilizing this paradigm can be best understood by examining the roots of the paradigm itself within Plato’s theory of Forms.

[Return to Contents](#)

The Theory of Forms

The theory of Forms was conceived and evolved over three distinct periods within Plato's life (and philosophy) to address questions concerning the nature of knowledge. Socrates, regarded as the wisest man in the world during his lifetime, conceived the theory with Plato. Plato formalized the theory. And, Aristotle, a pupil of Plato, scrutinized and evolved the theory with Plato.

The first period of Plato's philosophy involved Socrates and Plato collaborating to conceive the theory of Forms. The quest for understanding the nature of knowledge began with the premise that something is understood if it can be defined. Socrates proposed that the Socratic method be used as the means for attaining definitions. The *Socratic method* is a dialectic (converse or discourse) method in which a teacher guides a pupil into recognizing a true conclusion by progressively questioning the pupil, rather than simply indicating that the conclusion is true, until the conclusion is reached by the pupil. This method was demonstrated by Socrates on an individual who is guided to recognize the truth of the Pythagorean theorem (within a right triangle, the square of the length of the hypotenuse is equal to the sum of the squares of the lengths of the other two sides). Socrates and Plato concluded that learning involves a pupil's *recollection* of Ideas (or Forms).

The second period of Plato's philosophy involved Plato's formalization of the theory of Forms. Plato theorized that Forms or Ideas are the central elements of knowledge, and to know the Form of a thing is to understand the nature of that thing. *Forms* are properties or abstract non-material essences (qualities) of things. *Things* in the real world are *particulars*, Forms are *universals*, and particular entities instantiate certain universal entities. Plato conjectured that the search for knowledge involves defining Forms by using the dialectic method to progress through a series of hypotheses that lead to the recollection of knowledge. The objective of the search is formalized within the notion of Forms, and the search itself involves the Socratic method.

The third period of Plato's philosophy involved Aristotle and Plato collaborating to scrutinize and evolve the theory of Forms. Plato refined the dialectic method into a *procedure of collection and division*. Collection involves the identification of an embracing Form from a number of given specific Forms, and the embracing Form is divided by the selection of differences into the more specific Forms. Plato advanced the one over many principle as the motivation for the procedure of collection and division. The *one over many principle* specifies that where there are a number of elements (particulars) of the same kind sharing a set of properties, there is a single thing (Form) that is the kind being and having this set of properties. The principle motivates the postulating of universals from a collection of particulars; that is, a plurality of things may have a single ideal Form apart from the particulars themselves. Aristotle's *third man argument* maintains that if a plurality of things are defined by participating in a single ideal Form, another Form is required to explain both the plurality of particulars and the ideal Form. Fundamentally, because Forms themselves have the properties they are Forms of (particulars), the original plurality plus the ideal Form constitute a new plurality which requires the existence of another third ideal Form, and so on to an infinite regress for ideal Forms.

Aristotle attempted to apply a highly mathematical understanding of the Forms. He argued that a formal logic must be used in place of the dialectic method, where certain things are stated and something other than what is stated follows of necessity from what is stated (deduction). He also argued against the separation between particulars and Forms on the basis that substances in the real world consist of both matter and form, and for Forms to be instantiated in many particulars, they cannot be separable or have an independent existence. In his own philosophy, Aristotle proposed the notion of *Categories* to refer to ten classes that together covered all modes (states) of being. Particular things were called *primary substances*, and species and genera of primary substances were called *secondary substances*. The species of a particular thing defines what the thing is. Both primary substances and secondary substances are real, and

the separation of particulars and Forms was avoided.

Plato and Aristotle sought after a master science (the First Philosophy) where all things are systematically related in a single all encompassing system of knowledge. The theory of Forms embodies the germination for the First Philosophy and defines the fundamental concepts of the object–oriented paradigm.

[Return to Contents](#)

The Real World

The real world is the domain encompassing problems, solutions, and problem–solving efforts. The real world includes the following:

- Things or *entities* that have a purpose or role within a world. Entities have *structural characteristics* that represent what a thing "knows" and *behavioral characteristics* that repress what a thing "does". An aggregate entity may contain component entities that are structural characteristics and behavioral characteristics.
- *Relationships* among entities. Relationships may be treated as structural characteristics shared among multiple entities that have a relationship with one another. An aggregate entity may contain component entities that are related via an attribute of the aggregate entity.
- *Occurrences* among entities. Occurrences may be treated as behavioral characteristics shared among multiple entities that interact with one another. An aggregate entity may contain component entities that are manipulated via an operation of the aggregate entity.

These concepts are fundamental for interacting with the real world.

[Return to Contents](#)

The Object–Orientated Paradigm

The object–oriented paradigm focuses on the behavioral and structural characteristics of entities as complete units. It is concept–centric (holistic) in that it focuses on all the types of features that constitute any given concept. The paradigm encompasses and supports the following pillars (first principles):

- *Abstraction* involves the formulation of representations by focusing on similarities and differences among a set of entities to extract intrinsic essential characteristics (relevant common features) and avoid extrinsic incidental characteristics (irrelevant distinguishing features) in order to define a single representation having those characteristics that are relevant to defining every element in the set.
- *Encapsulation* involves the packaging of representations by focusing on the hiding of details to facilitate abstraction, where *specifications* are used to describe what an entity is and what an entity does and *implementations* are used to describe how an entity is realized.
- *Inheritance* involves the relating and reusing of existing representations to define new representations.
- *Polymorphism* involves the ability of new representations to be defined as variations of existing representations, where new implementations are introduced but specifications remain the same such that a specification has many implementations.

These pillars are used to facilitates communication, increase productivity and consistency, and enable the management of change and complexity within problem–solving efforts.

[Return to Contents](#)

Objects and Classes

Objects and classes abstract *entities*.

Objects are representational constructs of entities. Objects encapsulate structural characteristics known as attributes and behavioral characteristics known as operations. Attributes are representational constructs of structural characteristics of entities and determine the possible states of an object. Operations are representational constructs of behavioral characteristics of entities and determine the possible behaviors of an object as invoked in response to receiving a message. Objects have identity and are instances of classes. Fundamentally, objects are abstracted entities that encapsulate state and behavior.

Classes are descriptions of objects with a common implementation. Classes are concerned with the implementation of uniform structural characteristics and behavioral characteristics. Fundamentally, classes are descriptions of objects with common attributes, operation implementations, semantics, associations, and interactions.

Types are descriptions of objects with a common specification. Types are concerned with the specification of uniform structural characteristics and behavioral characteristics. Types may be explicitly related to classes, where a class receives an interface from a type and provides an implementation for the interface, or types may be implicitly related to classes, where a class defines an interface and provides an implementation. Fundamentally, types are descriptions of objects with common attributes, operation interfaces, semantics, associations, and interactions.

There are various other concepts regarding objects and classes that are not discussed above.

[Return to Contents](#)

Links and Associations

Links and associations abstract *relationships* among entities.

Links are representational constructs of entities that relate other entities. Links are instances of associations. Fundamentally, links are abstracted relationships among objects.

Associations are descriptions of links with a common implementation. Aggregations are associations that specify a whole–part relationship among an aggregate and component parts. Compositions are aggregations with strong ownership and coincident lifetime constraints among a composite and component parts. Generalizations are associations specifying a taxonomic relationships that relate more general representational constructs and more specific representational constructs. The more specific representational constructs are derived from existing more general representational constructs and acquire the characteristics of the more general representational constructs via inheritance. The more specific representational constructs may override characteristics received via inheritance and introduce new characteristics. Because the more specific representational constructs receive the characteristic of the more general representational constructs, instances of the more specific representational constructs may be substituted for instances of the more general representational constructs. Polymorphism enables the same message (operation interface) to invoke the appropriate method (operation implementation) based on the class of the receiver when a more specific instance is substituted for a more general instance. Fundamentally, associations are descriptions of links with common attributes, operation implementations, semantics, associations, and interactions.

There are various other concepts regarding links and associations that are not discussed above.

[Return to Contents](#)

Scenarios and Interactions

Scenarios and interactions abstract *occurrences* among entities.

Scenarios are representational constructs of entities that are conduits for a sequence of message exchanges among other entities. Scenarios are instances of interactions. Fundamentally, scenarios are abstracted message exchanges among objects.

Interactions are descriptions of scenarios with a common implementation. Message exchanges involve a sender who is said to apply an operation on a receiver by sending a message (operation interface) that invokes a method (operation implementation) within the receiver. Fundamentally, interactions are descriptions of scenarios with common message exchange sequences, classes, and associations.

There are various other concepts regarding scenarios and interactions that are not discussed above.

[Return to Contents](#)

The Theory of Forms and the Object–Oriented Paradigm

The following brief correlation can be made between the theory of Forms and the object–oriented paradigm to exemplify the value of the theory in shaping the paradigm:

<i>The Theory of Forms</i>	<i>The Object–Oriented Paradigm</i>
The Socratic method (and recollection)	Abstraction (as a means for gaining knowledge)
Forms, universals, and secondary substances	Classes (and encapsulation)
Things, particulars, and primary substances	Objects (and encapsulation)
The procedure of collection and division, and the one over many principle	Inheritance (and polymorphism)
The third man argument	Abstraction (within a conceptual framework for modeling involving multiple levels of abstraction)

There are various semantic variations or interpretations of the object–oriented paradigm due to the definition and interpretation of the concepts that constitute the paradigm, application and utilization of the paradigm, and the closure and extensibility of the paradigm. However, all variations must support the pillars (first principles) of the paradigm.

[Return to Contents](#)

Conclusion

Because the object–oriented paradigm is vividly applied in software and hardware engineering, most practitioners understand it from a technology perspective and believe that it is *only* concerned with technology–based systems; when actually, the paradigm is *not* simply concerned with technology–based systems but with systems in general.

In our evolution to understand the world and the nature of knowledge, there have been many paradigms *preceding* the object–oriented paradigm and there will be many paradigms *succeeding* the object–oriented paradigm; therefore, an understanding of the paradigm and its origin is very crucial in enabling us to progress to the next paradigm in this evolutionary process.

[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 © 1998 Sinan Si Alhir. All rights reserved.