



UNIVERSITY OF
CALGARY

**A tutorial report for SENG 609.22
Agent Based Software Engineering
Course Instructor: Dr. Behrouz H. Far**

Comparison of Objects and Agents

By Louis Wang

Introduction

All software development involves abstraction. It can be argued that the complexity of the problem you can solve is directly related to the kind and level of abstraction. This tutorial intends to focus on object-oriented and agent-based software engineering in terms of abstraction. For example, assembly language is a small level of abstraction over underlying machines. Imperative languages such as C and Fortran were abstraction of assembly languages. These languages were big improvements over assembly language, but their primary abstraction still requires you to think in terms of the structure of the computer rather than the structure of the problem you are trying to solve. Agent-based and object-oriented analysis and design have taken software engineering from the domain of solutions to the domain of problems that software is designed to solve. This tutorial intends to compare objects and agents.

What is an Object?

An object is a bundle of attributes and method calls. The basic idea behind object-oriented programming is simulation. A program should be written to simulate the states and activities of real world objects. An important concept of object-oriented programming is the concept of class. Class can be thought of as a cookie cutter, where it can create many different cookies with the same shape. For example, a mammal class can include many different objects such as people, lion, deer and cow. These characteristics represent a pure approach to object oriented programming:

- Everything is an object.
- A program is a bunch of objects telling each other what to do by sending messages.
- A program is a bunch of objects telling each other what to do by sending messages.
- All objects of a particular type can receive the same messages .

What is an Agent?

Agents are often described as entities with attributes considered useful in a particular domain, functions continuously and autonomously in a particular environment often inhabited by other agents and processes. Some people consider agents as “objects with attitudes”. Some people look at agents and objects as different things although they share many things in common. A list of common agent attributes is shown below:

- Adaptively
- Autonomy:
- Collaborative behavior:
- Knowledge-level communication ability
- Inferential capability
- Temporal continuity
- Personality
- Adaptability
- Mobility

What are the similarities between agents and objects?

1. Both accept the principle of encapsulation and information hiding
2. Both recognize the importance of interactions.

What are the differences between agents and objects?

1. Agents have control over their own behavior, whereas objects are totally obedient to one another and do not have autonomy over their choice of action. For an example, if a function call is made for an object's method, the object does not have the choice not to process the function call. The method can be implemented in such a way that it does nothing in the case of a particular function, but it does not have the ability to reject the function call. Agents have control over their own behavior.
2. Object-oriented approaches defines relationship by static inheritance hierarchies, where as agent-based approach defines a much more complex relationship between agents.
3. Intelligent agents typically constitute a conceptually higher abstraction than objects. Agent-oriented programming can be thought as a specialization of object-oriented approach, with what kind of constraints on what kinds of state-defining parameters, message types and methods are appropriate. From this perspective, an agent is essentially an object with 'altitude'.
4. Agents are multiple and dynamic classification. In object-oriented design, once an object is created. It can never change the class it belongs. Agents can be more flexible. For example, a particular agent can be a spouse, an employee, a customer and a landowner.
5. Agents have the ability to learn. It can add or subtract features dynamically.
6. Objects can be thought of as centrally organized, because an object's methods are invoked under the control of other components in the system. Agent-based environments can employ both centralized and decentralized processing. While agents can certainly support centralized systems, they can also provide us with the ultimate in distributed computing.

Conclusion

This paper covers the ideas of objected-oriented and agent-based software engineering, including their similarities and differences. Agent-based software can be view as somewhat higher level than objected-oriented software engineering. One of the key benefits of agent-based paradigm is that it allows us to capture intention. Another key benefit of agent-based paradigm is that it is easier to combine pre-existing system together. Agent-based analysis can be implemented using Objected-oriented methodology. However, the current methodology might not be sufficient enough to cover the complexity of agent-based paradigm.

References

Thinking in Java, Bruce Eckel

SENG 609.22 Agent-based Software Engineering notes, Behrouz Homayoun Far
<http://www.enel.ucalgary.ca/People/far/>

Bridging the Gap between Between Objects and Agents, Benjamin N. Grosff
http://www.acm.org/sigplan/oopsla/oopsla99/2_ap/tech/2d1a_bridge.html