Objects and Components: What’s the difference?

John Daniels
Syntropy Limited
john@syntropy.co.uk

Agenda

- Components in context
- Object principles
- Component principles
- Components vs. objects
- Examples
- Modelling components with UML
Object principles

Identity: Every object has a unique tag by which it is accessed.

Object: A way of representing in software an idea, a thing, or an event according to a chosen set of principles.

Interface: Defines capabilities of the object. Many objects can share the same interface, and an object may provide many interfaces.

Class: Defines how the object works. Many objects can share the same class. An object is an instance of a class.

Unification: An object holds data and provides operations on them.

Client/Supplier independence: The user of an object cares only about the interface, not the class. Client code will work with objects of different classes if they support the same interface.
What is an Object?

- Specification unit
- Implementation unit
- Execution unit

Component principles

- **Component** is not a well-defined term
- Many people believe they are “doing components” but they are doing many different things
- Ranges from loose notion of “useful package” (c.f. UML) to fully standardised plug-in part
- I’m using **component** to mean something quite specific: a piece of software conforming to a defined component standard
Aspects of a component

- It has an implementation

```java
for (int i=0; i<limit; i++)
    { list[i] = ... }
```

- It conforms to a standard

- It has a specification

- It can be deployed

- It can be packaged into modules

Industry component standards

- Microsoft COM+
  - Set to become the dominant standard in the NT world

- CORBA Components
  - EJB without the Java?

- DCOM
  - CORBA & DCOM for the plumbing
Component standard features

- **Component Model:**
  - defined set of services that support the software
  - set of rules that must be obeyed in order to take advantage of the services

- **Simple programming model, no need to design/know about the infrastructure**

- **Services include:**
  - remote access, transactions, persistent storage, security
  - typically use services by configuring not programming

---

**What is a Component?**

**Objects:**
- Component Implementation
- Component Specification
- Interface
- Component
- Component Module
- Execution unit

**Classes:**
- Component Implementation
- Component Specification
- Interface
- Component
- Component Module

**Relationships:**
- Specification unit
- Implementation unit
- Execution unit

---

email:john@syntropy.co.uk  
Syntropy Limited
Components and Classes

- A component is a special kind of class
  - Has hooks to connect into an environment
  - Can provide other classes (dependent on standard)
- The component (with any provided classes) is the unit of deployment and replacement

Compile-time and run-time

Compile-time

Run-time

At run-time we can create many independent instances of the component and its provided classes.
**Interfaces vs Component Specs**

- **Interface**
  - Represents the usage contract
  - Provides a list of operations
  - Defines an underlying logical information model specific to the interface
  - Specifies how operations affect or rely on the information model
  - Describes local effects only

- **Component Specification**
  - Represents the realization contract
  - Provides a list of supported interfaces
  - Defines the run-time unit
  - Defines the relationships between the information models of different interfaces
  - Specifies how operations should be implemented in terms of usage of other interfaces

**Component deployment**

- **Component**
  - 1..* \( \rightarrow \) Component Module
  - Installed Component
    - \( ^1 \ast \) installed as
    - \( ^1 \ast \) file

- **Component Module**
  - 1 * \( \rightarrow \) Installed Module
    - 1 * copy
    - 1 * server

- **Registration unit**
- **Installation unit**
Example - Enterprise Java Beans

- UML Glossary: “a physical, replaceable part […] that packages implementation and […] provides the realization of a set of interfaces”
UML Component (v1.3)

- UML has very loose notion of component and interface
- It isn’t possible to express all the subtleties we want without extending UML
- We need an OMG initiative to define a “UML Profile” for components
Want to know more?

- UML Components by John Cheesman and John Daniels, Addison-Wesley
- http://www.umlcomponents.com