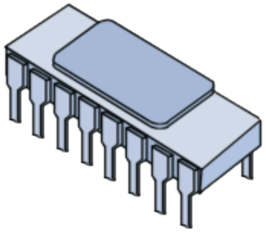
Architecture des Systèmes d’Information

JavaBeans



Traduction en cours

1996 Sun Definition

- "A JavaBean is a reusable software component that can be manipulated visually in a builder tool."

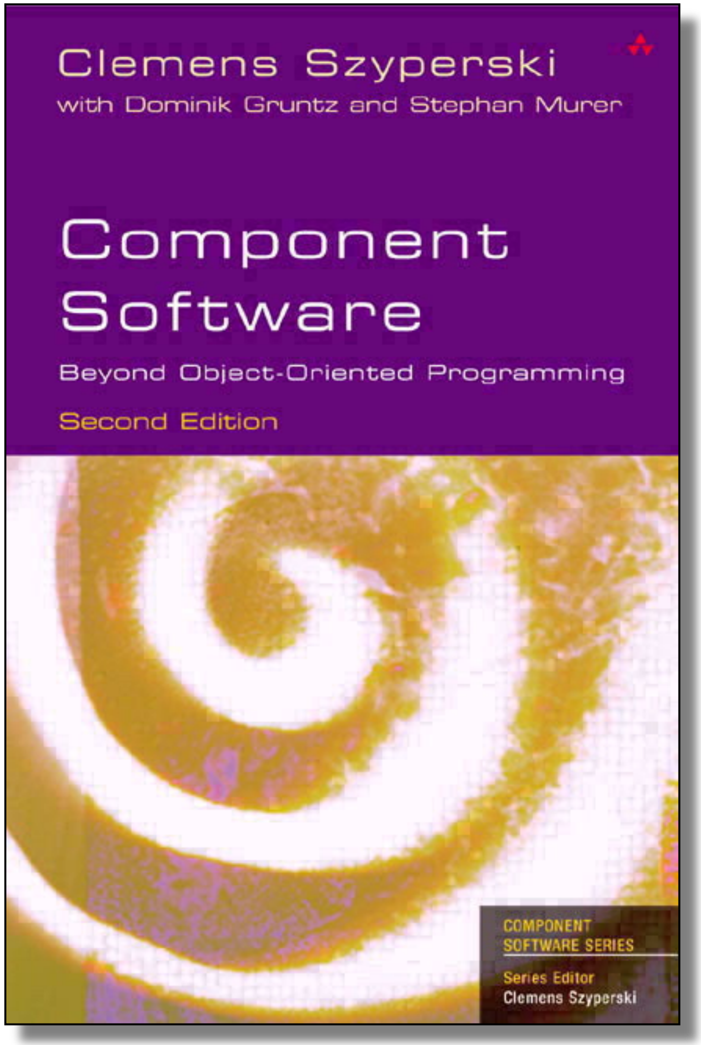
What is a component ?

- “A component is an object that can be used and tested as a unit, independent of the context in which the component is eventually used. The internal implementation of a component is completely hidden from the user.”

What is a component model ?

- “A software component model is a specification for how to develop reusable software components and how these component objects can communicate with each other”.

Clemens Szyperski



Clemens Szyperski

with Dominik Gruntz and Stephan Murer

Component Software

Beyond Object-Oriented Programming

Second Edition

COMPONENT SOFTWARE SERIES

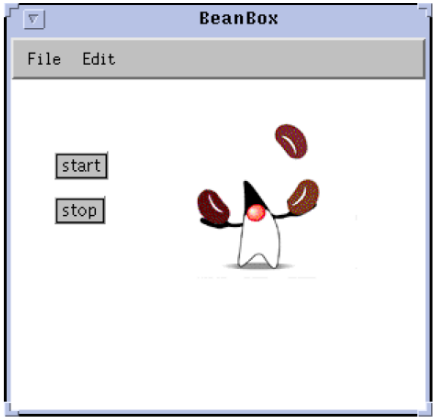
Series Editor Clemens Szyperski

JavaBean

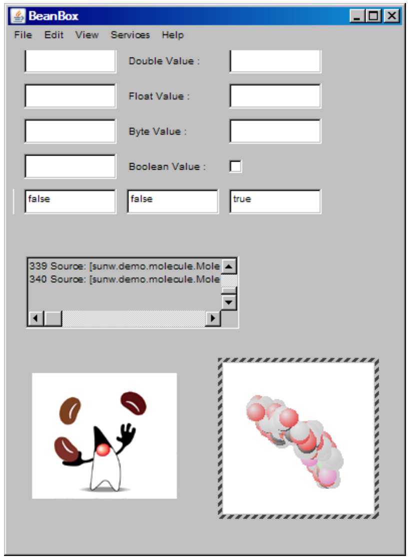
- JavaBean is just a Java class that respects the following rules :
  - it is a public class,
  - it has a public constructor with no arguments
  - it has public get and set methods to read and write to properties.

JavaBean component model

- JavaBeans component is any Java class that conforms to the JavaBeans component model.
- If a class follows the JavaBeans rules, it can work with visual design tools.



BeanBox



Double Value :

Float Value :

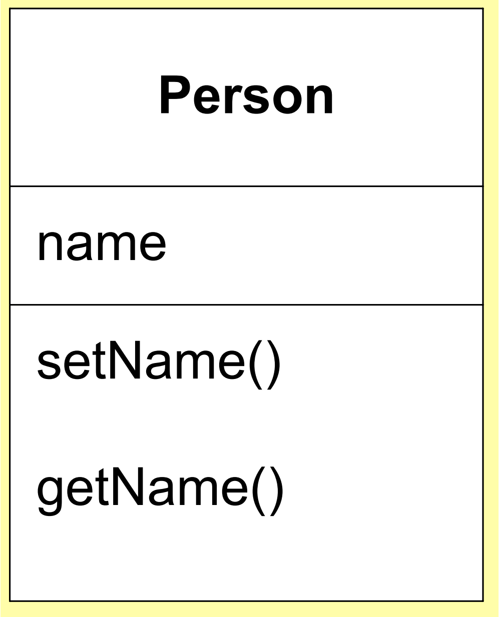
Byte Value :

Boolean Value : □

false false true

```
339 Source: [sunw.demo.molecule.Mole
340 Source: [sunw.demo.molecule.Mole
```

Person Bean

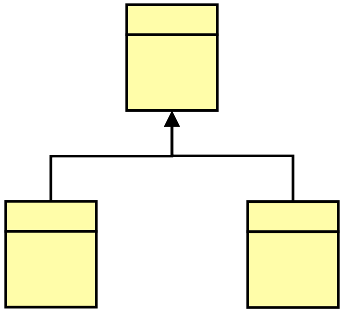


Person

name

setName()

getName()



Person bean

```
package bean;

public class Person implements java.io.Serializable
{
   private String firstName = null;
   private String lastName = null;

   public Person() {
   }
   public String getFirstName(){
      return firstName;
   }
   public String getLastName(){
      return lastName;
   }

   public void setFirstName(String firstName){
      this.firstName = firstName;
   }
   public void setLastName(String lastName){
      this.lastName = lastName;
   }
}
```

Person.java

JavaBean Events

- JavaBeans communicate: by passing Event objects, derived from java.util.event.
- The event type indicates what type of information the Event contains.

Introspection

- JavaBeans supports autodescription through an introspection mechanism.

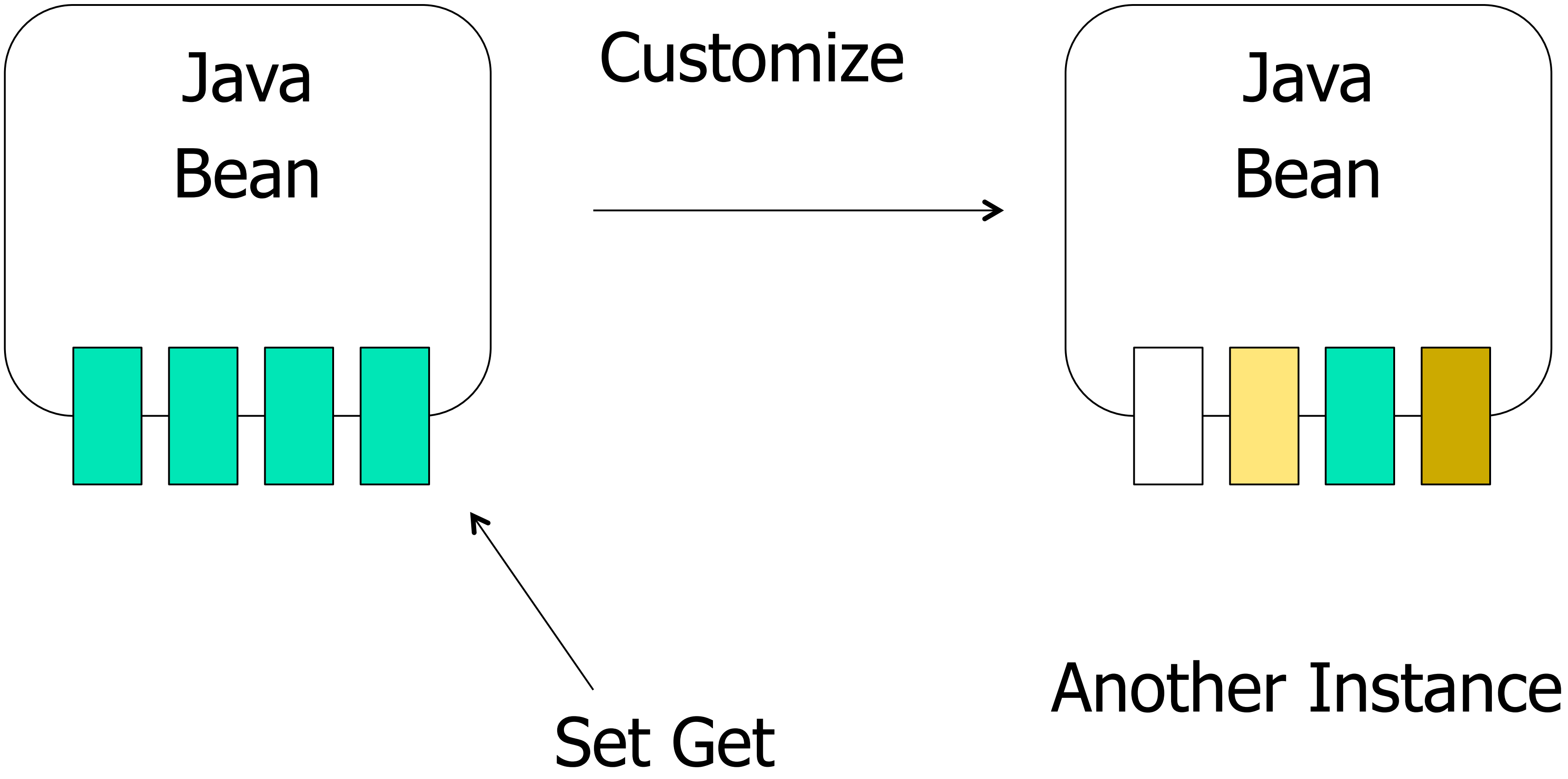
Bean Customization

- customization means configuring the *internal state of a bean*
- The internal state of a bean is defined by attributs like :
  - color,
  - size,
  - password string,
  - ....
- These attributs are called *properties*

JavaBeans Properties

- A JavaBean property is a named attribute that can be accessed.
- The attribute can be of any Java data type
- A JavaBean property may be read, write, read only, or write only.
- JavaBean properties are accessed through two methods in the JavaBean's implementation class:
  - property name :firstName
    - => getFirstName()
    - => setFirstName()

Java Bean Attributs and properties



Java

Bean

Customize

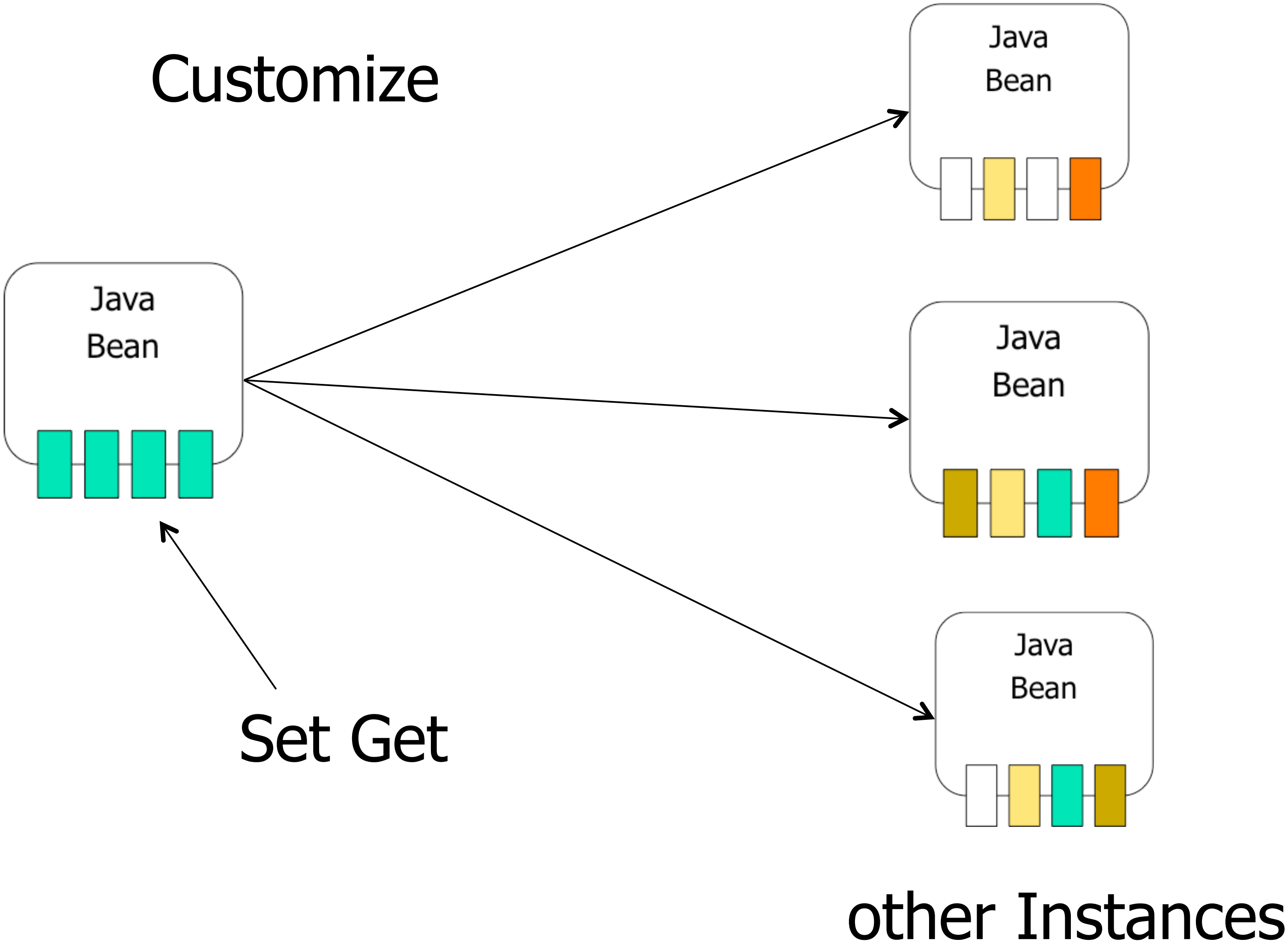
Java

Bean

Set Get

Another Instance

Java Bean Attributs and properties



JavaBeans components Lifecycle

- One of the most important aspects of JavaBeans concerns two different 'times' or moments in which an instance of a JavaBean can exist, they are called *Design-Time* and *Run-Time.*
- *Design-Time* :
  - makes reference to the moment when a JavaBean instance is being edited in a builder tool.
- *Run-Time* :
  - makes reference to the moment when a JavaBean instance is running as part of an application

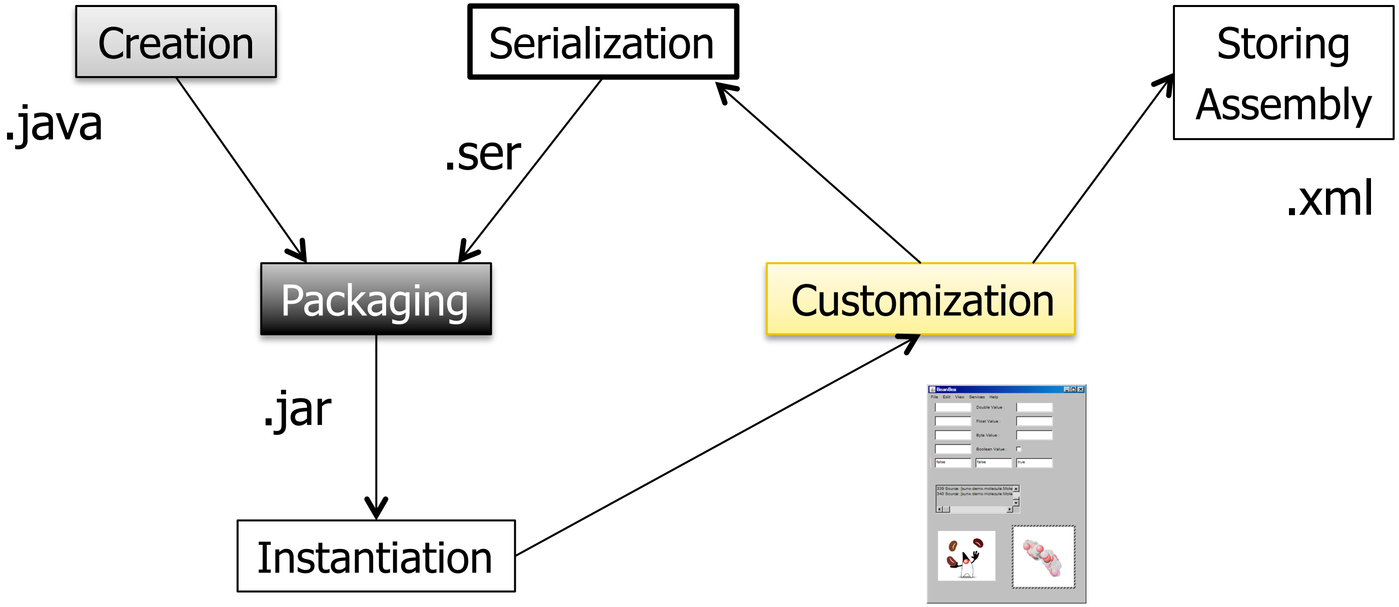
http://www.humbertocervantes.net/beansdiscussion.html#1

No need to access source code

- If an instance is customized, serialized and then packaged it becomes a *prototype* for other JavaBean instances.
- This sequence of steps is very important since it provides a method to set the equivalent of class values without the need to access source code of the JavaBean and to recompile it.

http://www.oracle.com/technetwork/java/javase/faq-135947.html

JavaBean LifeCycle : 6 pseudo States



Creation

Serialization

Storing Assembly

.java

.ser

.xml

Packaging

Customization

.jar

Instantiation

http://www.oracle.com/technetwork/java/javase/faq-135947.html