

JAXB 2.x

Sang Shin
Michèle Garoche
www.javapassion.com
“Learning is fun!”



Agenda

- Improvements of JAXB 2.x over JAXB 1.x
- Architecture
- Default data type mapping
- Java to Schema annotations

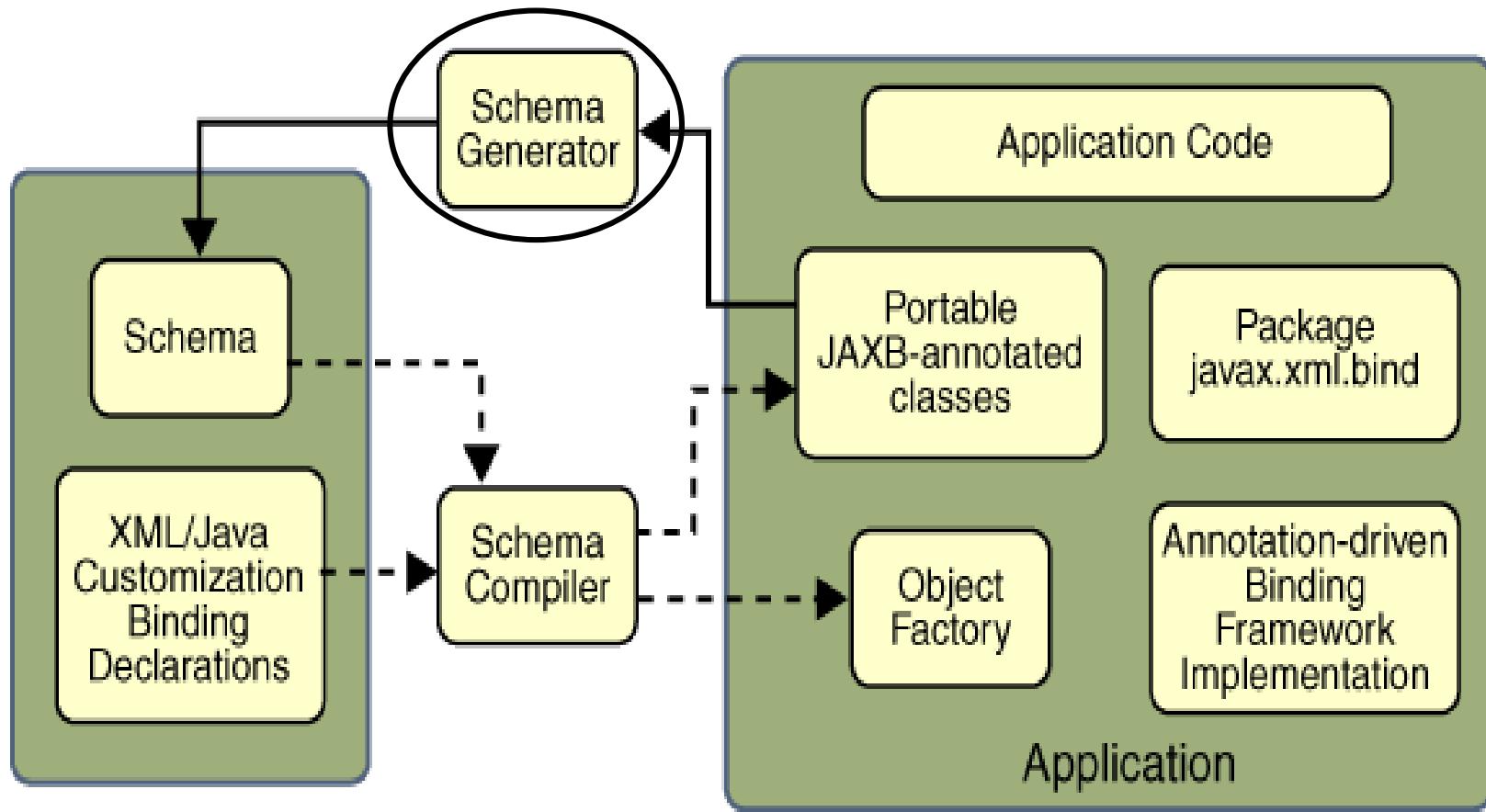
JAXB 2.x Improvements over JAXB 1.x

JAXB 2.x Improvements over JAXB 1.x

- Support for binding Java-to-XML Schema, with the addition of the *javax.xml.bind.annotation* package to control this binding. (JAXB 1.x specified the mapping of XML Schema-to-Java, but not Java-to-XML Schema.)
- A significant reduction in the number of generated schema-derived classes.
- Additional validation capabilities through the JAXP 1.3 validation APIs.
- Smaller runtime libraries.

Architecture

Architecture



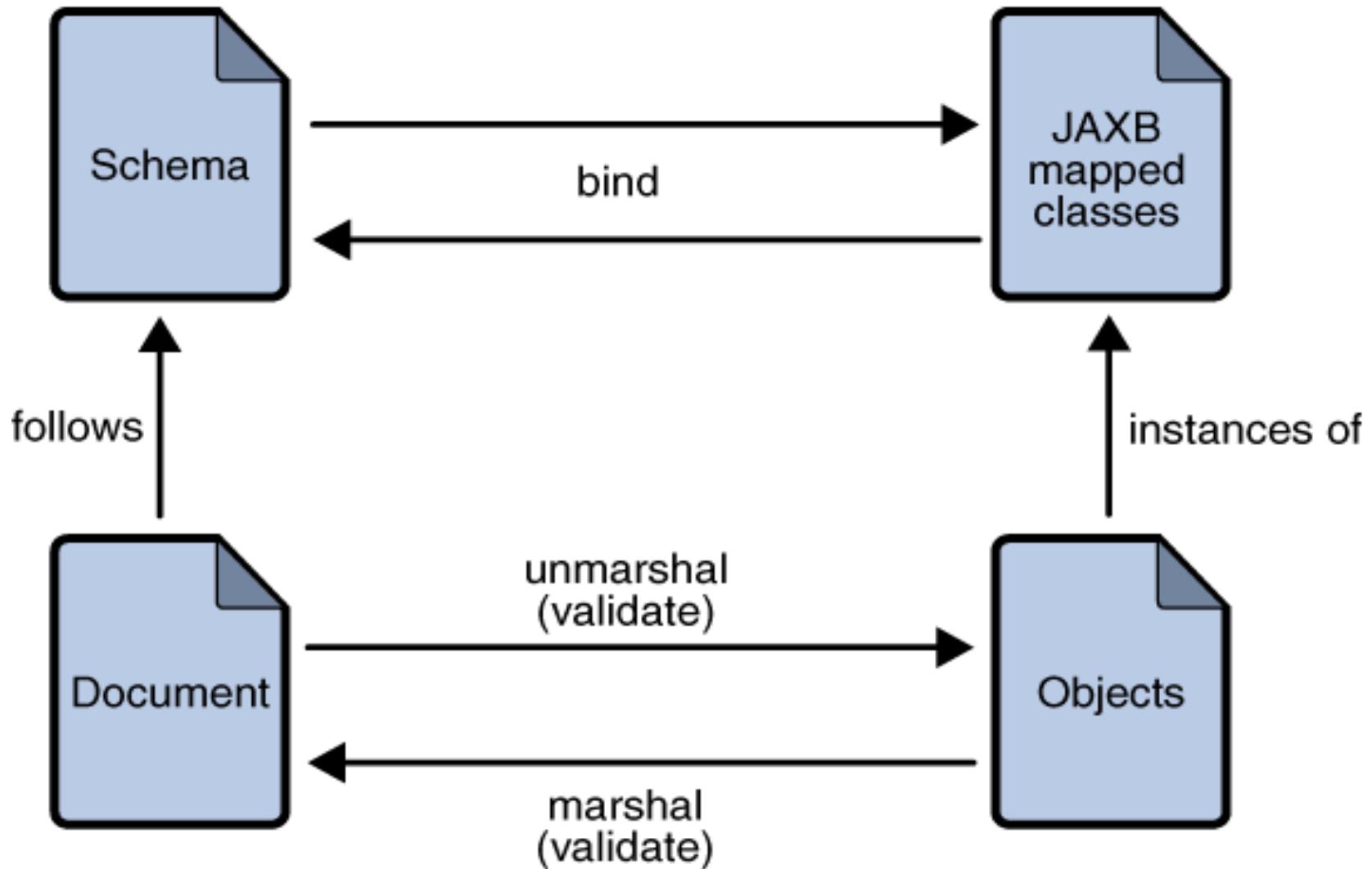
-----→ Schema to Java

→ Java to Schema

Architectural Components

- Schema compiler
 - > Binds a source schema to a set of schema-derived program elements. The binding is described by an XML-based binding language.
- Schema generator (from JAXB 2.x)
 - > Maps a set of existing program elements to a derived schema. The mapping is described by program annotations.
- Binding runtime framework
 - > Provides unmarshalling (reading) and marshalling (writing) operations for accessing, manipulating, and validating XML content using either schema-derived or existing program elements.

JAXB Binding Process



Unmarshalling

- Provides a client application the ability to convert XML data into JAXB-derived Java objects.

Marshalling

- Provides a client application the ability to convert a JAXB-derived Java object tree back into XML data.
 - > By default, the Marshaller uses UTF-8 encoding when generating XML data.
 - > Client applications are not required to validate the Java content tree before marshalling
 - > There is also no requirement that the Java content tree be valid with respect to its original schema to marshal it back into XML data.

Validation

- Validation is the process of verifying that an XML document meets all the constraints expressed in the schema
- JAXB 1.x provided validation at unmarshal time and also enabled on-demand validation on a JAXB content tree
- JAXB 2.x only allows validation at unmarshal and marshal time.

Default Datatype Mapping

Schema to Java

- The Java language provides a richer set of data type than XML schema

XML Schema Type	Java Data Type
xsd:string	java.lang.String
xsd:integer	java.math.BigInteger
xsd:int	int
xsd:long	long
xsd:short	short
xsd:decimal	java.math.BigDecimal
xsd:float	float
xsd:double	double
xsd:boolean	boolean
xsd:byte	byte
xsd:QName	javax.xml.namespace.QName
xsd:dateTime	javax.xml.datatype.XMLGregorianCalendar
xsd:base64Binary	byte[]
xsd:hexBinary	byte[]

Schema to Java

xsd:unsignedInt	long
xsd:unsignedShort	int
xsd:unsignedByte	short
xsd:time	javax.xml.datatype.XMLGregorianCalendar
xsd:date	javax.xml.datatype.XMLGregorianCalendar
xsd:g	javax.xml.datatype.XMLGregorianCalendar
xsd:anySimpleType	java.lang.Object
xsd:anySimpleType	java.lang.String
xsd:duration	javax.xml.datatype.Duration
xsd:NOTATION	javax.xml.namespace.QName

JAXBElement

- When XML element information can not be inferred by the derived Java representation of the XML content, a *JAXBElement* object is provided
- This object has methods for getting and setting the object name and object value.

Schema-to-Java

- Two ways to customize an XML schema:
 - > As inline annotations in a source XML schema
 - > As declarations in an external binding customizations file that is passed to the JAXB binding compiler

Java to Schema Annotations

Java to Schema: Annotations to Java Package

Annotation	Description and Default Setting
@XmlSchema	<p>Maps a package to an XML target namespace.</p> <p>Default Settings:</p> <pre>@XmlSchema (xmlns = {}, namespace = "", elementFormDefault = XmlNsForm.UNSET, attributeFormDefault = XmlNsForm.UNSET,)</pre>
@XmlAccessorType	<p>Controls default serialization of fields and properties.</p> <p>Default Settings:</p> <pre>@XmlAccessorType (value = AccessType.PUBLIC_MEMBER)</pre>
@XmlAccessorOrder	<p>Controls the default ordering of properties and fields mapped to XML elements.</p> <p>Default Settings:</p> <pre>@XmlAccessorOrder (value = AccessorOrder.UNDEFINED)</pre>
@XmlSchemaType	<p>Allows a customized mapping to a XML Schema built-in type.</p> <p>Default Settings:</p> <pre>@XmlSchemaType (namespace = "http://www.w3.org/2001/XMLSchema", type = DEFAULT.class)</pre>
@XmlSchemaTypes	<p>A container annotation for defining multiple @XmlSchemaType annotations.</p> <p>Default Settings:</p> <pre>None</pre>

JAXB Annotations to Java Class

Annotation	Description and Default Setting
@XmlType	<p>Maps a Java class to a schema type.</p> <p>Default Settings:</p> <pre data-bbox="462 558 1071 858">@XmlType (name = "##default", propOrder = {""), namespace = "##default" , factoryClass = DEFAULT.class, factoryMethod = "")</pre>
@XmlRootElement	<p>Associates a global element with the schema type to which the class is mapped.</p> <p>Default Settings:</p> <pre data-bbox="462 1019 956 1189">@XmlRootElement (name = "##default", namespace = "##default")</pre>

Thank you!

**Sang Shin
Michèle Garoche
www.javapassion.com
“Learning is fun!”**

