

Tecnologie Cloud e Mobile

Lez. 16

XML Schema e WSDL

Giuseppe Psaila

Università di Bergamo

giuseppe.psaila@unibg.it

XML Schema

XML Schema (o XSD)

- XML Schema è nato per sostituire il DTD
- Ma non solo lo sostituisce, fornisce concetti che DTD non fornisce
 - Tipi di dati per gli attributi
 - Riutilizzo di strutture
- Nel seguito vedremo solo alcuni aspetti di XML Schema, perché è troppo vasto

StrutturaBase

- Il prefisso del namespace è xs

```
<?xml version="1.0" encoding="UTF-8" ?>
```

```
<xs:schema
```

```
  xmlns:xs="http://www.w3.org/2001/XMLSchema">
```

```
...
```

```
</xs:schema>
```

Definire un Elemento

- Se l'elemento da definire ha un contenuto solo testuale, la sua definizione è molto semplice

```
<xs:element name="orderperson"  
    type="xs:string" />
```

- Con type si può definire il tipo del contenuto, in questo caso è una stringa

Tipi di Dato

- Vari tipi di dato sono disponibili (alcuni)
- xs:positiveInteger
- xs:integer
- xs:decimal
- xs:string
- xs:date
- xs:time
- xs:dateTime

Definizione di Elemento Semplice

- Esempio:

```
<xs:element name="title"  
    type="xs:string" />
```

Elemento che contiene solo testo. Esempio:

```
<title>My Title</title>
```

Definizione di Elemento Complesso

- Si definisce un tipo complesso

```
<xs:element name="shiporder">  
  <xs:complexType>  
    ...  
  </xs:complexType>  
</xs:element>
```


Definizione di Elemento Complesso

- Che cosa c'è in un tipo complesso?
- `<xs:sequence>`
definisce sequenze di elementi
- `<xs:all>`
indica gli elementi che devono essere presenti,
indipendentemente dall'ordine

Definizione di Elemento Complesso

- `<xs:choice>`

Uno solo degli elementi indicati può occorrere nel contenuto

Può essere usato anche dentro `<xs:sequence>`

Definizione di Elemento Complesso

- Per vincolare la molteplicità, sia `xs:element` che gli altri elementi prevedono due attributi opzionali
- `minOccurs`
valori "0" o "1"
- `maxOccurs`
valore non negativo o "unbound"

Riuso di Elementi

- È possibile definire gli elementi per poi essere usati più volte
- Come usarli?
- `<xs:element ref="nome"/>`

Definizione di Attributi

- È possibile definire gli attributi degli elementi
- `<xs:attribute name=... typw=... use=.../>`
- use è opzionale e vale:
 - "optional"
 - "required"

Riuso di Attributi

- È possibile riusare attributi generici
- `<xs:attribute ref="nome"/>`

Altri Costrutti

- Si possono definire anche dei tipi riusabili
 - Per il contenuto degli elementi
 - Per gli attributi
- Si possono definire classi di tipi
- Si possono definire gerarchie
- Si possono definire elementi partendo da classi di tipi

Correttezza e Validazione

- Una specifica XSD sostituisce e potenzia il DTD
- Quindi, un documento viene detto «Valido» rispetto alla specifica XSD
- Come validare? Con i parser validanti XML Schema
XML-Schema Validating Parser

Esempio

- Vedi file XML_Schema.pdf

WSDL

WSDL

- Web
- Service
- Definition
- Language

Scopo

- Definire in modo formale un web service
- Per facilitare lo scambio dei dati con il web service
- Conseguenza: sviluppo di librerie e strumenti per effettuare la comunicazione in modo automatico, con poche linee di codice

Visione

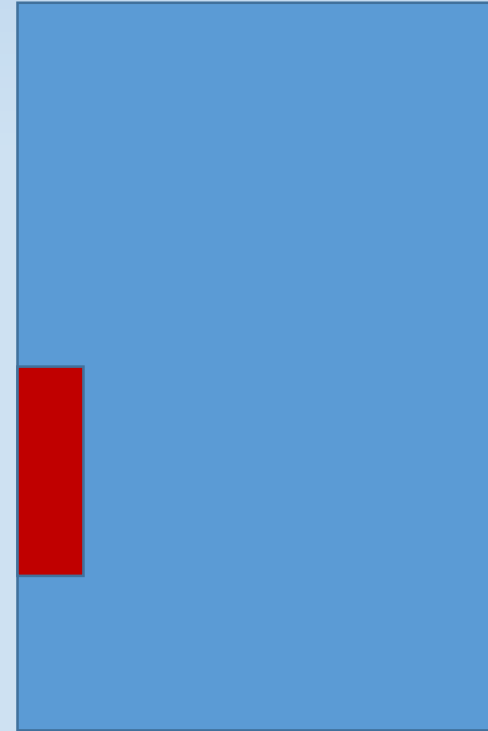
- Supponiamo di avere due sistemi informativi S1 e S2
- S2 deve ricevere dei dati da S1, da incorporare nelle sue strutture dati
- S2 espone un web service e genera una specifica WSDL che invia a S1

Visione

S1



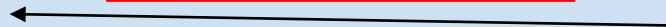
S2



Web Service



WSDL Spec.



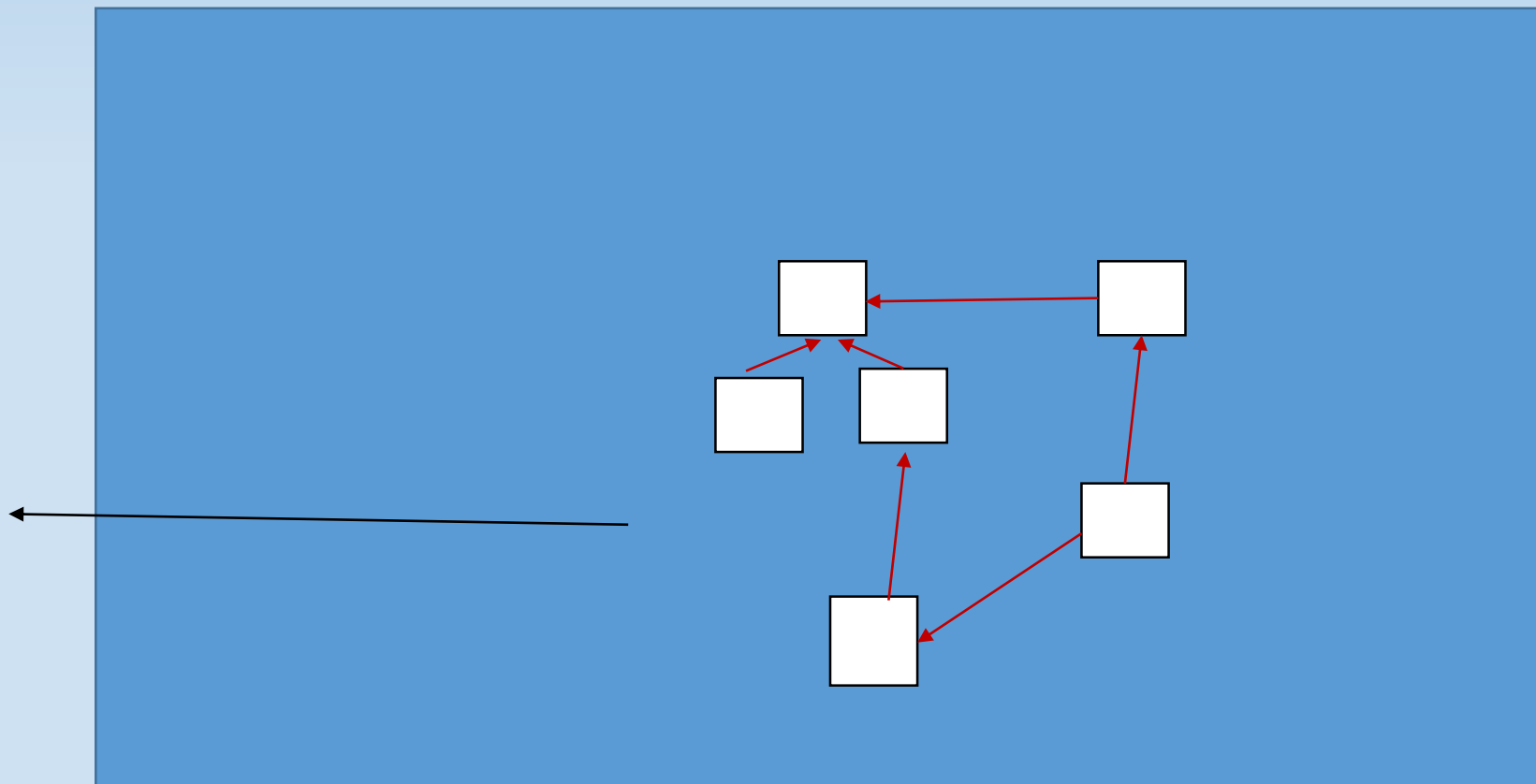
Visione

- Ma chi scrive la specifica WSDL?
- Nessuno,
- Viene generata automaticamente dalle strutture dati object-oriented che devono essere popolate con i dati ricevuti dal web service
- Partendo o dalle definizioni delle classi o dalle definizioni del database

Visione

S2

WSDL Spec.

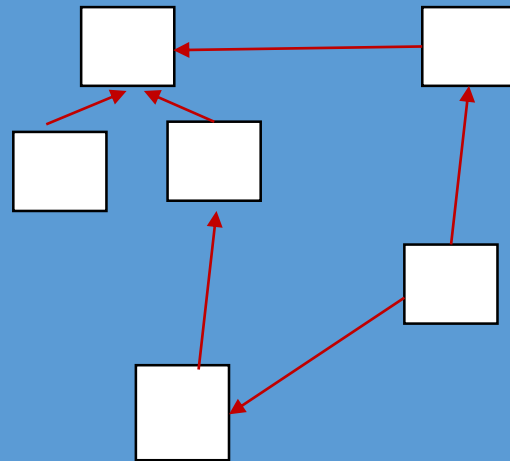


Visione

- S1 riceve la specifica WSDL
- Da essa si genera la definizione delle strutture dati che devono contenere i dati da inviare
- Nel linguaggio usato per implementare S1

Visione

S1



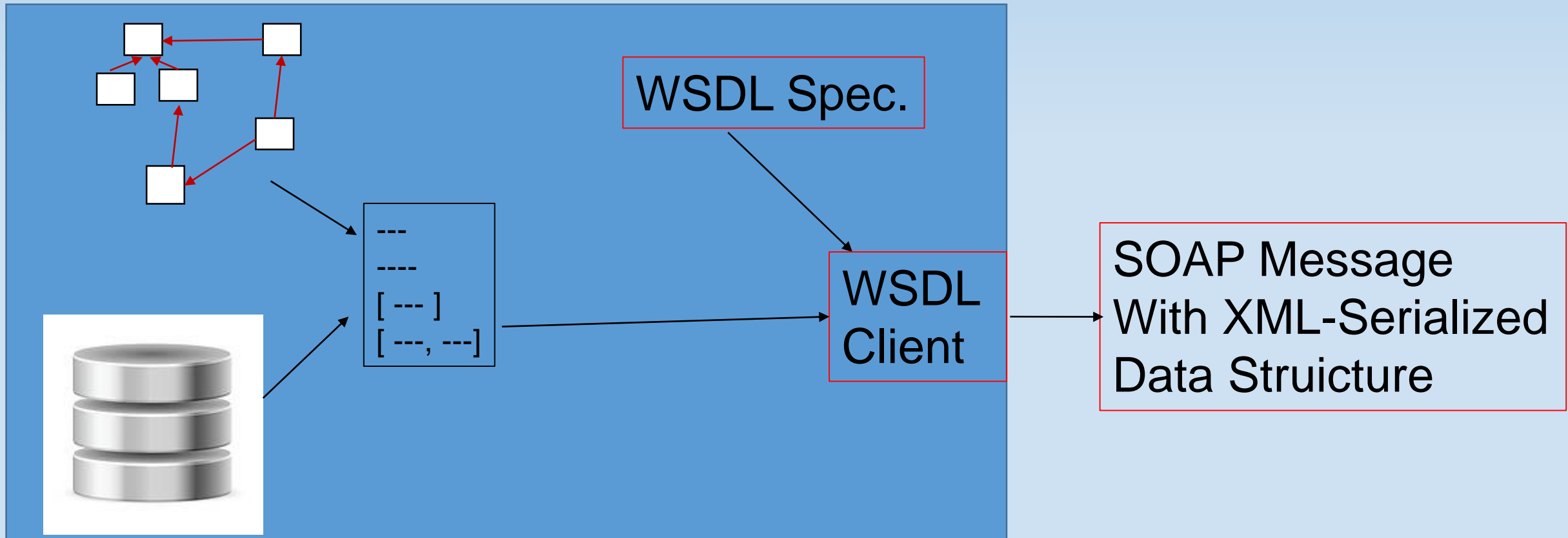
WSDL Spec.

Visione

- I programmatori di S1 popolano la struttura dati con i dati da inviare
- Usando le definizioni ottenute dalla specifica WSDL
- Il client WSDL in S1 serializza la struttura dati, usando la specifica WSDL per generare il contenuto XML del messaggio

Visione

S1

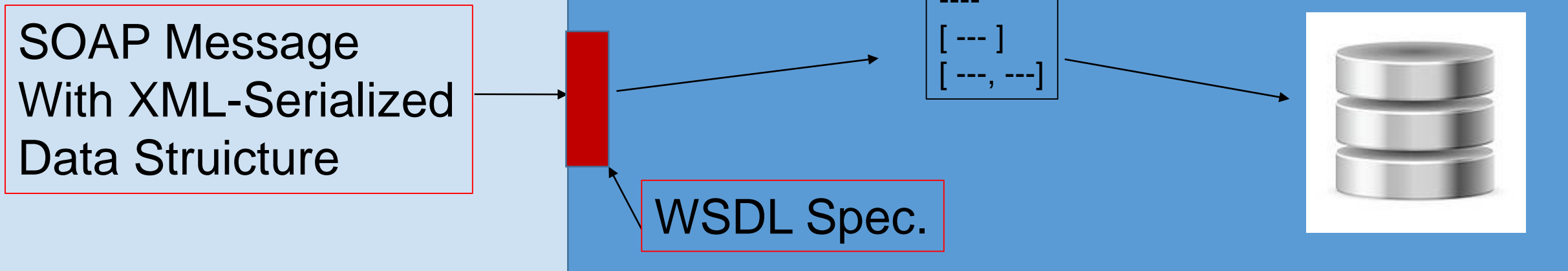


Visione

- Il web service esposto da S2 riceve il messaggio
- Usando la specifica WSDL deserializza il messaggio
- Popola la struttura dati interna a S2, nel suo linguaggio di programmazione
- Questa struttura dati verrà utilizzata dal resto del codice di S2

Visione

S2



Dettaglio di WSDL

- Non lo vediamo, perché allo stato attuale è inutile conoscerlo. Se siete interessati trovate facilmente molti documenti
- Che cosa occorre sapere?
- Lo vediamo nelle prossime slide

XML

- La specifica WSDL è basata su XML
- Fa parte di un namespace specifico
- Definisce i contenuti dei messaggi e come comunicare con il web service esposto

XSD

- Il contenuto dei messaggi è definito tramite una specifica XSD
- Quindi, WSDL si appoggia allo standard XML Schema e usa il suo sistema dei tipi
- Quindi, adotta una visione a oggetti

HTTP

- Il protocollo usato per inviare i messaggi è HTTP
- Trattandosi di web services, non poteva essere diversamente
- Il lato server del web service si appoggia ad un HTTP Server, occorre quindi aprire una porta specifica

SOAP

- Il contenuto dei messaggi (inviati via HTTP) è basato sul protocollo SOAP
- La envelope di SOAP contiene la serializzazione delle strutture dati
- La serializzazione è fatta con XML

Vantaggi

- Tutti i passaggi intermedi sono gestiti in automatico
- I programmatori lavorano direttamente sulle strutture dati di propria competenza
- Senza conoscere minimamente il WSDL
- La specifica WSDL viene generata automaticamente