

**FACHHOCHSCHULE WEDEL
SEMINARARBEIT**

in der Fachrichtung
Informatik

Seminar: „Service-orientierte Architektur (SoA)“

Thema Nr. 3:

Web Services MetadataExchange

Eingereicht von: Arne Simonsen (Matrikelnr. 5914)
Pestalozzistraße 26 F
25421 Pinneberg
Tel. (04101) 68133
Email: wi5914@fh-wedel.de

Erarbeitet im: 7. Semester

Abgegeben am: 6. Dez. 2006

Betreuer: Prof. Dr. Sebastian Iwanowski
Fachhochschule Wedel
Feldstraße 143
22880 Wedel

Inhaltsverzeichnis

Abbildungsverzeichnis.....	II
1 Einleitung.....	1
1.1 Motivation.....	1
1.2 Zielsetzung.....	2
1.3 Zusätzliche Bemerkungen.....	2
2 Web Services Metadata	3
2.1 Metadata Resource.....	3
2.2 Web Services Metadata.....	3
3 Web Services Transfer.....	6
3.1 WS Transfer Get.....	7
3.2 WS Transfer Get Response.....	8
3.3 Fazit zu WS Transfer Get.....	10
4 Get Metadata	11
4.1 Get Metadata Request.....	11
4.2 Get Metadata Response.....	13
4.3 Get Metadata und WS Transfer	13
5 Metadaten in Endpunkt Referenzen.....	15
6 Fazit.....	16
Anhang.....	17
Verwendete XML Präfixe.....	17
Literaturverzeichnis	18

Abbildungsverzeichnis

Abbildung 1: Web Services Metadata Element.....	3
Abbildung 2: Vordefinierte Dialekte.....	4
Abbildung 3: Empfehlungen zur Identifier Wahl.....	4
Abbildung 4: WS Transfer Get.....	7
Abbildung 5: Header einer WS Transfer Get Response Nachricht.....	8
Abbildung 6: SOAP Body mit Metadaten.....	9
Abbildung 7: WS Transfer Get	10
Abbildung 8: Get Metadata Request.....	12
Abbildung 9: SOAP Body Get Metadata Response.....	13
Abbildung 10: Einbettung in WS Addressing 2004 Endpunkt Referenz.....	15
Abbildung 11: Einbettung in WS Addressing 1.0 Core Endpunkt Referenz.....	15
Abbildung 12: Verwendete XML Präfixe und Spezifikationen.....	17

1 Einleitung

1.1 *Motivation*

In Web Services werden Metadaten genutzt, um zu beschreiben, was bei der Benutzung des Services zu beachten ist und auf welche Art andere Endpunkte den Service benutzen können. Die drei am häufigsten benutzten Metadaten Dialekte sind dabei WS-Policy, WSDL und XML Schema. WS-Policy dient dabei zur Beschreibung von Fähigkeiten, Anforderungen und anderen Eigenschaften des Services, in WSDL werden Schnittstellen mit ihren Funktionen, Protokollen und Adressen beschrieben und in den XML Schemata der Aufbau und die Elemente der XML basierten Nachrichten, um mit dem Web Service zu kommunizieren.

Wenn jetzt aber ein Client einen Web Service benutzen will und dafür die Metadaten benötigt, da er ansonsten nicht in der Lage ist, den Service zu benutzen, steht er vor dem Problem, wie er an diese Metadaten herankommt.

Ohne die in dieser Arbeit und in der Spezifikation „WS MetadataExchange“ beschriebenen Methoden hat der Client des Web Services grundsätzlich nur vier verschiedene Möglichkeiten an die Metadaten zu gelangen¹:

- Manuell nach veröffentlichten Metadaten des Services und Einbindung in die eigene Anwendung. (z.B. im Internet)
- Manuell den Serviceanbieter kontaktieren und sich die Metadaten zukommen lassen.
- Automatisches Abrufen über proprietäre Schnittstellen des Serviceanbieters. Auch hier müssen erst manuell Informationen über die Schnittstelle gesammelt werden.
- Automatisches Abrufen aus einem zentralem Register (UDDI)

Von diesen vier Möglichkeiten ist allerdings nur die letzte wünschenswert, da die anderen drei einen hohen bis sehr hohen manuellen Aufwand erfordern

¹ Vgl. Ertl, Thomas: Service-Oriented Architecture, 2005, S. 248

Einleitung

und daher dem Wunsch der automatischen Maschine-Maschine Kommunikation widersprechen. Da jedoch viele Serviceanbieter ihre Metadaten nicht in zentralen Registern publizieren, ist es notwendig, standardisierte Mechanismen einzuführen, die den automatischen Metadatenaustausch zwischen Client und Service organisieren.

1.2 Zielsetzung

Ziel dieser Arbeit ist die drei Mechanismen zum Erlangen von Metadaten, wie sie in der Spezifikation „WS MetadataExchange“ eingeführt werden, vorzustellen und zu erklären.

Analog zur Spezifikation wird dabei erst ein Format zur Kapselung von Metadaten eingeführt. Danach werden die drei Methoden einzeln vorgestellt und deren Unterschiede beleuchtet und aufgezeigt, wie diese Methoden miteinander zusammenarbeiten und welche Position die Spezifikation im WS Framework einnimmt.

1.3 Zusätzliche Bemerkungen

Die in den XML Beispielen verwendeten XML Namespaces werden aus Gründen der Übersichtlichkeit überwiegend nicht in den Beispielen selbst definiert, sondern sind im Anhang als Tabelle aufgelistet².

Die verwendeten XML Beispiele sind der Spezifikation „Web Services MetadataExchange“ entnommen und leicht verändert worden.

² Siehe Abbildung 12: Verwendete XML Präfixe und Spezifikationen

2 Web Services Metadata

2.1 Metadata Resource

Unter „Metadata Resources“ werden Web Services verstanden, die über eine Endpunkt Referenz adressierbar sind und die als WS Transfer Ressource behandelt werden können. „Metadata Resources“ müssen die WS Transfer Get Operation unterstützen und einem der in „WS MetadataExchange“ definierten Dialekte³ entsprechen.

2.2 Web Services Metadata

Ein Web Services Metadata Element enthält eine Liste von verschiedenen Metadaten Elementen wie zum Beispiel WS Policy, WSDL oder XML Schemata. Die einzelnen unterscheidbaren Metadaten werden dabei in „MetadataSection“ Elementen geschachtelt und können entweder direkt eingebettet oder über URL oder über eine Endpunktreferenz erreichbar sein⁴.

```
<xs:Metadata xmlns:xs="http://www.w3.org/2001/XMLSchema">
<mex:Metadata xmlns:mex="http://schemas.xmlsoap.org/ws/2004/09/mex">
<mex:Metadata ...>
  <mex:MetadataSection Dialect="xs:anyURI"
  (Identifier="xs:anyURI"? ...>
  (
  <mex:MetadataReference ...>
    endpoint-reference
  </mex:MetadataReference>
  |
  <mex:Location>xs:anyURI</mex:Location>
  |
  DialectSpecificElement
  )
  </mex:MetadataSection>*
  ...
</mex:Metadata>
```

Abbildung 1: Web Services Metadata Element

3 Siehe Abbildung 2: Vordefinierte Dialekte

4 Siehe Abbildung 1: Web Services Metadata Element

Web Services Metadata

Dialect URI	Metadata Format
http://www.w3.org/2001/XMLSchema	xs:schema [XML Schema Part 1]
http://schemas.xmlsoap.org/wsdl/	wsdl:definitions [WSDL 1.1]
http://schemas.xmlsoap.org/ws/2004/09/policy	wsp:Policy [WS-Policy]
http://schemas.xmlsoap.org/ws/2004/09/policy/attachment	wsp:PolicyAttachment [WS-PolicyAttachment]
http://schemas.xmlsoap.org/ws/2004/09/mex	mex:Metadata [This specification]

Abbildung 2: Vordefinierte Dialekte

Metadata

Enthält für jedes unterscheidbare Metadatenelement ein MetadataSection Element. Zum Beispiel ein Element für die WSDL Definitionen, ein Element für die WS Policy Informationen und zwei Elemente für zwei verschiedene XML Schemata.

MetadataSection

Enthält genau ein Metadatenelement (zum Beispiel eine WS Policy) welches entweder eingebettetes dialektspezifisches XML, eine URL oder eine Endpunkt Referenz ist.

Dialect: Eine URI die angibt um welche Art von Metadaten es sich handelt.

Identifier: Ein dialektspezifischer Zusatz, dient allgemein zur weiteren Identifikation und Unterscheidung von Metadatenelementen desselben Dialektes. Für einige Metadatendialekte gibt es Empfehlungen, wie der Identifier zu wählen ist⁵.

Dialect URI	@Identifier value
http://www.w3.org/2001/XMLSchema	xs:schema/@targetNamespace
http://schemas.xmlsoap.org/wsdl/	wsdl:definitions/@targetNamespace
http://schemas.xmlsoap.org/ws/2004/09/policy	wsp:Policy/@Name
http://schemas.xmlsoap.org/ws/2004/09/policy/attachment	Not defined
http://schemas.xmlsoap.org/ws/2004/09/mex	Not defined

Abbildung 3: Empfehlungen zur Identifier Wahl

⁵ Siehe Abbildung 3: Empfehlungen zur Identifier Wahl

Web Services Metadata

MetadataReference

Das Element `MetadataReference` enthält eine Endpunkt Referenz zu einer Metadata Ressource wie in Kapitel 2.1 beschrieben. Die Metadata Ressource muss die WS Transfer Get Operation derart unterstützen, dass das Metadanelement, welches dem Dialekt und dem Identifier des Väterelementes entspricht, heruntergeladen werden kann.

Location

Das Location Element enthält eine URL auf eine Metadatenfile. Die Metadatenfile muss nur den primären Zugriffsmechanismus des verwendeten Protokolls unterstützen, zum Beispiel HTTP GET bei einer HTTP URL.

Dialect Specific Element

Statt einer Location oder einer `MetadataReference` können die Metadaten auch direkt als eingebettetes XML gegeben sein. Der XML Code wird dann anhand der angegebenen Dialect und Identifier Attribute interpretiert.

3 Web Services Transfer

Die erste Methode zum Abruf von Metadaten von einer „Metadata Resource“, ist das Senden eines WS Transfer Get an den Endpunkt der „Metadata Resource“. Als Antwort würde der nachfragende Client eine SOAP Nachricht mit allen sich an diesem Endpunkt befindenden Metadaten bekommen.

Zur Erläuterung von Inhalt und Aufbau der WS Transfer Get und WS Transfer Get Response und des Zusammenhangs zwischen den beiden Nachrichten zu erläutern, soll ein kleines Beispiel dienen.

Angenommen ein Onlineshop will seinen Versand optimieren, indem er sich zu jedem Paket, das verschickt werden soll, per Web Service Schnittstelle aus einer Menge von Spediteuren denjenigen heraussucht, der ihm das beste Angebot macht. Um die Preisabfragen, die von mehreren Parametern (z.B. Größe und Gewicht) abhängen, und die eventuell darauf folgenden Bestellungen durchzuführen, muss der Onlineshop jedoch wissen, wie er diese Anfragen zu stellen hat und welchen Beschränkungen der jeweilige Spediteur unterliegt. Für den Fall, dass der Onlineshop nicht nur die Endpunktadresse des Services sondern auch die der zugehörigen Metadaten kennt, könnte eine Abfrage der Metadaten mit WS Transfer Get nun folgendermaßen aussehen⁶.

⁶ Siehe Abbildung 4: WS Transfer Get

3.1 WS Transfer Get

```
<s11:Envelope xmlns:s11='http://schemas.xmlsoap.org/soap/envelope/'
  xmlns:wsa10='http://www.w3.org/2005/08/addressing'>
  <s11:Header>
    <wsa10:Action>
      http://schemas.xmlsoap.org/ws/2004/09/transfer/Get
    </wsa10:Action>
    <wsa10:To>http://services.example.org/spediteurX/metadata
    </wsa10:To>
    <wsa10:ReplyTo>
      <wsa10:Address>http://shop.example.org</wsa10:Address>
    </wsa10:ReplyTo>
    <wsa10:MessageID>
      urn:uuid:1cec121a-82fe-41da-87e1-3b23f254f128
    </wsa10:MessageID>
  </s11:Header>
  <s11:Body />
</s11:Envelope>
```

Abbildung 4: WS Transfer Get

Mit dieser Nachricht würde der Onlineshop den Endpunkt „<http://services.example.org/spediteurX/metadata>“, hinter dem sich die Metadaten des Spediteurs X verbergen, auffordern, ihm eine Repräsentation der Metadaten an die Adresse „<http://shop.example.org>“ zu schicken. Im Header befinden sich dabei die nötigen WS Addressing Informationen, um die Nachricht richtig zuzuordnen. Der Body der Nachricht bleibt leer.

Wie zu sehen ist, geht aus der WS Transfer Get Nachricht nicht explizit hervor, dass die Daten, nach denen gefragt wird, Metadaten sind. Würden sich hinter der Adresse „<http://services.example.org/spediteurX/metadata>“ keine Metadaten verbergen, würde der Endpunkt unter Umständen mit normalen Daten antworten, sofern er WS Transfer überhaupt unterstützt.

3.2 WS Transfer Get Response

Die Antwort des Spediteurs auf diese Anfrage könnte, in Header⁷ und Body⁸ unterteilt, wie folgt aussehen. (Das Envelope Element der SOAP Nachricht, welches Header und Body umschließt, wurde hier weggelassen.)

```
<s11:Header>
  <wsa10:Action>
    http://schemas.xmlsoap.org/ws/2004/09/transfer/GetResponse
  </wsa10:Action>
  <wsa10:To>http://shop.example.org</wsa10:To>
  <wsa10:RelatesTo>
    urn:uuid:1cec121a-82fe-41da-87e1-3b23f254f128
  </wsa10:RelatesTo>
</s11:Header>
```

Abbildung 5: Header einer WS Transfer Get Response Nachricht

Im Header befinden sich nur die für die Versendung wichtigen Daten, wie in diesem Fall die Adresse des Shops (`<wsa10:To>http://shop.example.org</wsa10:To>`) und die Angabe der Aktion (`<wsa10:Action>http://schemas.xmlsoap.org/ws/2004/09/transfer/GetResponse</wsa10:Action>`) und die für die Verarbeitung beim Onlineshop wichtige Angabe der Message - ID der Anfrage. Wie in der WS Transfer Get Message geht auch hier aus dem Header nicht hervor, dass es sich bei den übertragenen Daten um Metadaten handelt.

Im Body befindet sich ein einzelnes *Metadata* Element, in dem sämtliche sich an dem Endpunkt befindlichen Metadaten in jeweils eigenen *MetadataSection* Elementen enthalten sind.

Das erste *MetadataSection* Element enthält die WSDL Definitionen des Spediteurs X, wie man am *Dialect* Attribut erkennen kann, die in diesem Fall direkt in das *MetadataSection* Element eingebettet ist. Hier würden zum Beispiel die Methodenbeschreibungen des Services stehen, mit deren Hilfe der Onlineshop in der Lage wäre, Service Aufrufe an den Spediteurservice zu senden.

⁷ Siehe Abbildung 5: Header einer WS Transfer Get Response Nachricht

⁸ Siehe Abbildung 6: SOAP Body mit Metadaten

Web Services Transfer

Im zweiten *MetadataSection* Element befindet sich eine XML Schema Definition, die in diesem Fall nicht direkt eingebettet sondern nur als *Location* angegeben ist. Die XML Schema Metadaten könnten in diesem Fall mit einem HTTP GET abgefragt werden.

```
<s11:Body>
  <mex:Metadata>
    <mex:MetadataSection Dialect='http://schemas.xmlsoap.org/wsdl/'>
      <wsdl:definitions ...>
        omitted for brevity
      </wsdl:definitions>
    </mex:MetadataSection>
    <mex:MetadataSection Dialect='http://www.w3.org/2001/XMLSchema'
      Identifier='http://services.example.org/spediteurX/schemas'>
      <mex:Location>
        http://services.example.org/spediteurX/schemas
      </mex:Location>
    </mex:MetadataSection>
    <mex:MetadataSection
      Dialect='http://schemas.xmlsoap.org/ws/2004/09/policy'
      Identifier='http://services.example.org/spediteurX/policy'>
      <mex:MetadataReference>
        <wsa10:Address>
          http://services.example.org/spediteurX/policy
        </wsa10:Address>
      </mex:MetadataReference>
    </mex:MetadataSection>
  </mex:Metadata>
</s11:Body>
```

Abbildung 6: SOAP Body mit Metadaten

Im letzten *MetadataSection* Element befindet sich eine *MetadataReference*, die eine Endpunkt Referenz auf ein WS Policy Dokument enthält. Dieses Dokument, welches in unserem Spediteurs-Beispiel Beschränkungen von Paketgrößen oder Fähigkeiten wie minimale und maximale Lieferzeiten enthalten könnte, könnte mit einem weiteren WS Transfer Get an die Adresse „<http://services.example.org/spediteurX/policy>“ abgerufen werden.

3.3 Fazit zu WS Transfer Get

Die Benutzung von WS Transfer Get bietet eine gute Möglichkeit, gezielt Metadaten abzurufen, sofern man die Adressen der Metadaten Ressourcen kennt. Wenn jedoch nur die Adresse des Service Endpunktes und nicht die Adressen der Metadaten Ressourcen bekannt sind, ist es nicht möglich die Metadaten auf diese Art abzurufen, da ein WS Transfer Get, wie in Abbildung 7 verdeutlicht, für den Zweck des Metadatenabrufes nur auf Metadaten Ressourcen funktioniert.

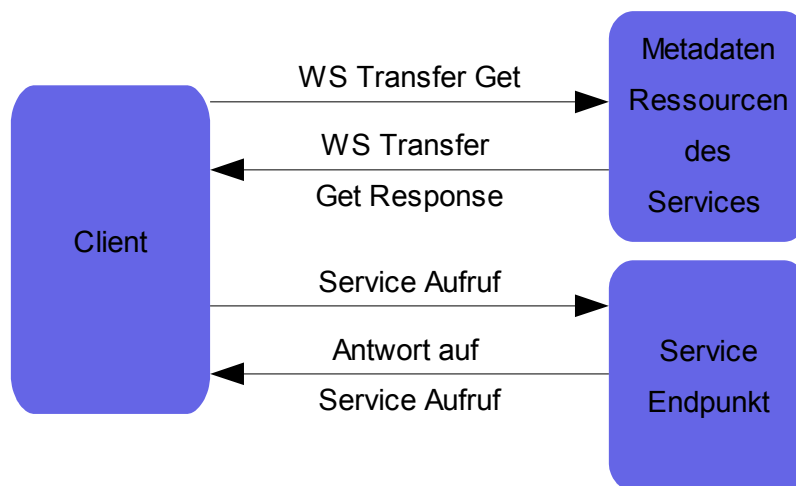


Abbildung 7: WS Transfer Get

4 Get Metadata

„Get Metadata“ ist ein speziell für die Übertragung von Metadaten entworfenes Format, das aus einer Anfrage, „Get Metadata Request“, und einer dazu gehörenden Antwort, „Get Metadata Response“, besteht.

Gegenüber dem Abruf mit WS Transfer hat Get Metadata den Vorteil, dass die Anfrage nicht an eine Metadata Ressource gestellt werden muss, sondern direkt der Service Endpunkt angesprochen werden kann. Außerdem ist es möglich, mit Get Metadata die Metadaten nach *Dialect* und *Identifier* zu filtern, so dass zum Beispiel nur nach *WS Policy* Metadaten gefragt werden kann, oder sogar nur nach einem *WS Policy* Element mit einem bestimmten Identifier.

4.1 Get Metadata Request

Der Aufbau der *Get Metadata Request* Nachricht soll wie *WS Transfer* am Beispiel des Onlineshops und der Spediteur Services erläutert werden. Dazu wird zuerst die Nachricht des Shops an den Spediteur X gezeigt⁹.

Im Header ändert sich gegenüber einem Abruf mit WS Transfer¹⁰ zum einen die ausgewählte *Action* von *WS Transfer Get* auf *Get Metadata Request* und zum anderen die Zieladresse des Abrufs, da jetzt nicht mehr die Metadaten angesprochen werden müssen, sondern direkt am Service Endpunkt nachgefragt werden kann.

Im Gegensatz zu *WS Transfer Get* bleibt der Body der SOAP Nachricht nicht leer, sondern enthält optionale Filter. In diesem Beispiel würde nur das *WS Policy* Element mit dem *Identifier* „<http://services.example.org/spediteurX/policy>“ ausgewählt werden.

Würde in diesem Beispiel der *Identifier* weggelassen werden, würden als Antwort alle *WS Policy* Elemente des Service Endpunktes übermittelt werden. Wenn auch der *Dialect* wegfallen, würden nicht nur die *WS Policy* Elemente sondern alle Metadaten des Services zurück übermittelt werden.

⁹ Siehe Abbildung 8: Get Metadata Request

¹⁰ Siehe Abbildung 4: WS Transfer Get

Get Metadata

```
<s11:Envelope xmlns:s11='http://schemas.xmlsoap.org/soap/envelope/'
  xmlns:wsa10='http://www.w3.org/2005/08/addressing'>
  <s11:Header>
    <wsa10:Action>
      http://schemas.xmlsoap.org/ws/2004/09/mex/GetMetadata/Request
    </wsa10:Action>
    <wsa10:To>http://services.example.org/spediteurX
    </wsa10:To>
    <wsa10:ReplyTo>
      <wsa10:Address>http://shop.example.org</wsa10:Address>
    </wsa10:ReplyTo>
    <wsa10:MessageID>
      urn:uuid:1cec121a-82fe-41da-87e1-3b23f254f128
    </wsa10:MessageID>
  </s11:Header>
  <s11:Body>
    <mex:GetMetadata>
      <mex:Dialect>
        http://schemas.xmlsoap.org/ws/2004/09/policy
      </mex:Dialect>
      <mex:Identifier>
        http://services.example.org/spediteurX/policy
      </mex:Identifier>
    </mex:GetMetadata>
  </s11:Body>
</s11:Envelope>
```

Abbildung 8: Get Metadata Request

Einschränkend muss jedoch gesagt werden, dass nur nach einem *Dialect* bzw. einem *Identifier* pro Aufruf gefiltert werden kann, das heißt, ein Aufruf, in dem gleichzeitig alle WS Policy Elemente und alle WSDL Definitionen abgerufen werden sollen, ist nicht möglich.

Get Metadata

4.2 Get Metadata Response

Die zu dieser *Get Metadata Request* passende *Response* Nachricht ähnelt im Header sehr stark der *WS Transfer Get Response* Nachricht¹¹, der einzige Unterschied ist, das die *Action* „<http://schemas.xmlsoap.org/ws/2004/09/transfer/GetResponse>“ durch „<http://schemas.xmlsoap.org/ws/2004/09/mex/GetMetadata/Response>“ ersetzt wird.

Im Body der Nachricht ändert sich nur, dass durch den Filter in der *Get Metadata Request* Nachricht alle *MetadataSection* Elemente, mit Ausnahme des Elementes mit dem *Dialect WS Policy* und dem *Identifier* „<http://services.example.org/spediteurX/policy>“, heraus gefiltert werden¹².

```
<s11:Body>
  <mex:Metadata>
    <mex:MetadataSection
      Dialect='http://schemas.xmlsoap.org/ws/2004/09/policy'
      Identifier='http://services.example.org/spediteurX/policy'>
      <mex:MetadataReference>
        <wsa10:Address>
          http://services.example.org/spediteurX/policy
        </wsa10:Address>
      </mex:MetadataReference>
    </mex:MetadataSection>
  </mex:Metadata>
</s11:Body>
```

Abbildung 9: SOAP Body Get Metadata Response

4.3 Get Metadata und WS Transfer

Der Vorteil von *Get Metadata* gegenüber der Abfrage in *WS Transfer* liegt darin, dass zum einen nicht Metadaten Ressourcen direkt angesprochen werden müssen, sondern sich ein Client direkt an den Service Endpunkt wenden kann, und zum anderen darin, dass es mit *Get Metadata* möglich ist nach bestimmten Metadaten Dialekten und sogar nach *Identifier* zu filtern, um so das Ergebnis und die zu übertragenden Metadaten einzuschränken.

¹¹ Siehe Abbildung 5: Header einer WS Transfer Get Response Nachricht

¹² Siehe Abbildung 9: SOAP Body Get Metadata Response

Get Metadata

Zudem ist ein Zusammenspiel von *Get Metadata* und *WS Transfer* denkbar, welches wie folgt aussehen könnte:

1. Der Client gelangt über ein öffentliches Register an die Endpunkt Adresse eines Services, der ihn interessiert.
2. Er sendet eine *Get Metadata Request* Nachricht an den Service und erhält als Antwort eine Liste mit Endpunkt Referenzen und *Locations* für alle Metadaten des Services.
3. Der Client kann nun mit *WS Transfer Get* die Metadaten abrufen, für die Endpunkt Referenzen vorhanden sind und die ihn interessieren. Für das Abrufen der Metadaten, für die nur *Locations* vorhanden sind, muss er den primären Zugriffsmechanismus¹³ der URLs benutzen.

Auf diese Weise kann sich ein Client zuerst einen groben Überblick über die vorhandenen Metadaten verschaffen und dann gezielt die Metadaten abrufen, die ihn interessieren.

¹³ Siehe Kapitel 2.2: Location

5 Metadaten in Endpunkt Referenzen

Zusätzlich zu den beiden Möglichkeiten Metadaten eines Services mit WS Transfer oder GetMetadata abzurufen, so genannten „pull“-Mechanismen, bei denen die Initiative für einen Metadatenaustausch vom Client ausgeht, wird in der Spezifikation „WS MetadataExchange“ noch ein „push“-Mechanismus eingeführt, bei dem die Metadaten des Service in einer Endpunkt Referenz auf ihn eingebettet werden.

Das führt dazu, dass jeder Client, der zum Beispiel über das zentrale Register UDDI auf Grund einer Suchanfrage die Endpunkt Referenz des Services zugeschickt bekommt, zusätzlich schon mit allen nötigen Metadaten ausgestattet ist, die er für die Benutzung des Services benötigt.

Die Einbettung der Metadaten in eine Endpunkt Referenz der Spezifikation *WS Addressing 2004* erfolgt durch das Einfügen eines Metadata Elementes mit den enthaltenen MetadataSection Elementen als direktes Kind Element der Endpunkt Referenz¹⁴.

```
<wsa04:EndpointReference>
  <wsa04:Address>http://services.example.org/spediteurX</wsa04:Address>
  <mex:Metadata> omitted for brevity
</mex:Metadata>
</wsa04:EndpointReference>
```

Abbildung 10: Einbettung in WS Addressing 2004 Endpunkt Referenz

In der *WS Addressing 1.0 Core* wurde ein zusätzliches Element eingebaut, so dass dort die Einbettung leicht verändert wurde¹⁵.

```
<wsa10:EndpointReference>
  <wsa10:Address>http://services.example.org/spediteurX</wsa04:Address>
  <wsa10:Metadata>
    <mex:Metadata> omitted for brevity </mex:Metadata>
  </wsa10:Metadata>
</wsa10:EndpointReference>
```

Abbildung 11: Einbettung in WS Addressing 1.0 Core Endpunkt Referenz

¹⁴ Siehe Abbildung 10: Einbettung in WS Addressing 2004 Endpunkt Referenz

¹⁵ Siehe Abbildung 11: Einbettung in WS Addressing 1.0 Core Endpunkt Referenz

Fazit

6 Fazit

Durch die Spezifikation *WS MetadataExchange* werden drei Methoden zur Verteilung von Metadaten definiert und damit standardisiert. Durch diese Standardisierung des Metadaten austauschs werden mehrere Ziele erreicht:

- Der Austausch von Metadaten wird vereinfacht und ist nun automatisierbar. Die manuelle Suche nach Metadaten wird überflüssig.
- Die Interoperabilität zwischen Web Services wird erhöht, da durch die Standardisierung Metadaten einfach austauschbar sind.
- Die Unabhängigkeit wird gestärkt, da keine Absprachen über proprietäre Mechanismen zum Austausch von Metadaten mehr nötig sind.
- Die Wiederverwendbarkeit wird verbessert, da Clients Metadaten, die nicht mehr aktuell sind, einfach aktualisieren können.

Zusätzlich wurde ein Metadaten Format definiert, das beliebige Arten von Metadaten kapseln und einfach durch neue Metadattypen erweitert werden kann.

Anhang

Anhang

Verwendete XML Präfixe

Prefix	XML Namespace	Specification(s)
wxf	http://schemas.xmlsoap.org/ws/2004/09/transfer	This specification
s11	http://schemas.xmlsoap.org/soap/envelope/	SOAP 1.1
s12	http://www.w3.org/2003/05/soap-envelope	SOAP 1.2
s	Either of s11 or s12	SOAP
wsa	Either of wsa1 or wsa2	WS-Addressing
xs	http://www.w3.org/2001/XMLSchema	XML Schema
wsdl	http://schemas.xmlsoap.org/wsdl	WSDL/1.1
wsa1	http://schemas.xmlsoap.org/ws/2004/08/addressing	WS-Addressing
wsa2	http://www.w3.org/2005/08/addressing/	WS-Addressing 1.0 Core

Abbildung 12: Verwendete XML Präfixe und Spezifikationen

Literaturverzeichnis

Literaturverzeichnis

Literaturverzeichnis

Ertl, Thomas (2005), *Service-Oriented Architecture Concepts, Technology, and Design*; Print by R.R. Donnelley; Crawfordsville, Indiana; © Pearson Education Inc.

Quellen im Internet

Ballinger, Keith / u. a. (2006) *Web Services MetadataExchange*,
<http://download.boulder.ibm.com/ibmdl/pub/software/dw/specs/ws-mex/metadataexchange.pdf> , Stand August 2006, Abruf 05.12.2006

Alexander, Jan / u. a. (2006) *Web Services Transfer*,
<http://www.w3.org/Submission/WS-Transfer/> , Stand September 2006, Abruf 05.12.2006