Durch den Wert "Permit" des FulfillOn XML Attributs des rewrite <xacml3:ObligationExpression> Elements ist festgelegt, dass die in der Regel beschriebenen Modifikationsanweisungen im Falle einer positiven Zugriffskontrollentscheidung durchzusetzen sind. Erwähnt sei an dieser Stelle, dass es sinnvoll sein kann, Modifikationsanweisungen in positiven und in negativen Zugriffsregeln zu definieren. Eine Modifikationsanweisung in einer positiven Regel hat das Ziel, die Weiterleitung einer Nachricht zu erlauben, wenn sie modifiziert wurde. Eine negative rewrite Regel hat hingegen das Ziel, das Subjekt von der Modifikation seiner Nachricht zu informieren und es selbst entscheiden zu lassen, ob es auch die modifizierte Nachricht an den WFS senden möchte. Für weitere Details und Beispiele wird auf Abschnitt 6.1 verweisen.

```
1  <Rule Effect="Permit" RuleId="example:x" xmlns="xacml3.0" ...>
2    <Target>
3      <AnyOf>
4        <AllOf>
5          <Match MatchId="&string-equal;">
6            <AttributeValue DataType="&string;">Alice</AttributeValue>
7            <AttributeDesignator Category="&access-subject;"
                 AttributeId="&subject-id;" DataType="&string;"
                 MustBePresent="true"/>
8          </Match>
9          <Match MatchId="&xpath-node-equal;">
10           <AttributeValue DataType="&xpath;" Category="&message;">
11 /wfs:GetFeature/wfs:Query[@typeName="Building"]
12           </AttributeValue>
13           <AttributeDesignator AttributeId="&content-selector;"
                 DataType="&xpath;" Category="&message;"
                 MustBePresent="true"/>
14         </Match>
15       </AllOf>
16     </AnyOf>
17   </Target>
18   <ObligationExpressions>
19     <ObligationExpression ObligationId="&rewrite-obligation;"
         FulfillOn="Permit">
20       <!-- the rewrite expressions in form of an ISLT Stylesheet -->
21       <AttributeAssignmentExpression
           AttributeId="&xslt-rewrite-stylesheet;">
22         <AttributeValue DataType="&xslt;">
23           <xsl:stylesheet ... version="2.0">
24             <!-- hook for argument passing mechanism -->
25             <xsl:param name="&init-select-node-argument;:xslt-arg-1"
                   select="<!--dynamically-assigned-by-context-handler-->"/>
26             <!-- static predicate that shall be added to selection
                   predicate of the intercepted GetFeature request -->
```

```
27              <xsl:param name="predicate-to-add">
28                <ogc:Within>
29                  <ogc:PropertyName>location</ogc:PropertyName>
30                  <gml:Polygon srsName="osgb:BNG">
31                    <gml:outerBoundaryIs><gml:LinearRing>
32                      <gml:coordinates> 528000.000,178856.330 ...
33                      </gml:coordinates>
34                    </gml:LinearRing></gml:outerBoundaryIs>
35                  </gml:Polygon>
36                </ogc:Within>
37              </xsl:param>
38              <xsl:template match="node()|@*">
39                <xsl:choose>
40                  <xsl:when
41               test="self::node()=$&init-select-node-argument;:xslt-arg-1">
42                    <xsl:call-template name="modify-query"/>
43                  </xsl:when>
44                  <xsl:otherwise>
45                    <xsl:copy>
46                      <xsl:apply-templates select="node()|@*"/>
47                    </xsl:copy>
48                  </xsl:otherwise>
49                </xsl:choose>
50              </xsl:template>
51              <xsl:template name="modify-query">
52              <xsl:copy>
53                <xsl:apply-templates select="@*"/>
54                <xsl:apply-templates select="wfs:PropertyName"/>
55                <ogc:Filter>
56                  <xsl:if test="not(ogc:Filter)">
57                    <xsl:copy-of select="$predicate-to-add"/>
58                  </xsl:if>
59                  <xsl:if test="ogc:Filter">
60                    <ogc:And>
61                      <xsl:if test="not(ogc:Filter/ogc:And)">
62                        <xsl:copy-of select="$predicate-to-add"/>
63                        <xsl:apply-templates select="ogc:Filter/node()"/>
64                      </xsl:if>
65                      <xsl:if test="ogc:Filter/ogc:And">
66                        <xsl:apply-templates
                              select="ogc:Filter/ogc:And/node()"/>
67                        <xsl:copy-of select="$predicate-to-add"/>
68                      </xsl:if>
69                    </ogc:And>
70                  </xsl:if>
71                </ogc:Filter>
72                </xsl:copy>
73              </xsl:template>
```

```
74            </xsl:stylesheet>
75          </AttributeValue>
76        </AttributeAssignmentExpression>
77        <!-- argument that shall be passed to the xslt stylesheet -->
78        <AttributeAssignmentExpression
              AttributeId="&init-select-node-argument;:xslt-arg-1"
              Category="&xop;:category:obligation">
79          <AttributeDesignator AttributeId="&content-selector;"
                DataType="&xpath;" Category="&message;" MustBePresent="true"/>
80        </AttributeAssignmentExpression>
81        <AttributeAssignmentExpression
              AttributeId="&adr-representation-to-map;"
              Category="&obligation;">
82          <AttributeValue DataType="&string;">&content-based;
83        </AttributeValue></AttributeAssignmentExpression>
84      </ObligationExpression>
85    </ObligationExpressions>
86  </Rule>
```

Listing 4.14: XSLT-basierte Definition einer XACML v3.0 rewrite Regel

Unterhalb des <xacml3:ObligationExpression> Elements der oben abgebildeten rewrite Regel werden drei <xacml3:AttributeAsssignmentExpression> Elemente beschrieben:

Das &xslt-rewrite-stylesheet; <xacml3:AttributeAssignmentExpression> Element beinhaltet die Definition der Modifikationsanweisungen in Form eines XSLT Stylesheets. Das Stylesheet sorgt dafür, dass multiple Zugriffsentscheidungsanfragen derart umgeschrieben werden, dass darin beschriebene wfs:GetFeature/wfs:Query[@typeName = "Building"] Elemente nur mehr Gebäudefeatures innerhalb Deutschlands selektieren können. Um dies zu erreichen, wird das Prädikat "within($location$, $Polygon_{Germany}$)" (s. Z. 28-38) geeignet unterhalb betroffener <wfs:Query> Elemente eingefügt. Hierbei müssen dreierlei Fälle separat behandelt werden: die abgefangene Anfrage hat kein <ogc:Filter> Element, sie hat ein <ogc:Filter> Element ohne <ogc:And> Kind und sie hat ein <ogc:Filter> Element mit <ogc:And> Kind. Je nachdem, welcher dieser Fälle vorliegt, muss das neue Prädikat an unterschiedlichen Stellen eingefügt werden und vorhandene Teilbäume müssen unter Umständen verschoben werden (vgl. Z. 57-71). In den Zeilen 60 und 62 wird beispielsweise geprüft, ob unterhalb des aktuell betrachteten <wfs:Query typeName = "Building"> Elements ein <ogc:Filter> Element vorhanden ist, das allerdings kein <ogc:And> Element als Kindknoten hat. Ist dies der Fall, wird unterhalb des <ogc:Filter> Elements ein <ogc:And> Element hinzugefügt, dessen erstes Kind das neu hinzuzufügende Prädikat ist. Durch Reorganisation des Baumes muss das ehemalige Kind des <ogc:Filter> Elements als zweites Kind des neuen <ogc:And> Elements eingefügt werden.

Neben dem &xslt-rewrite-stylesheet; <xacml3:AttributeAssignmentExpression> Element beinhaltet das rewrite <xacml3:ObligationExpression> Element ein xslt-argument <xacml3:AttributeAssignmentExpression> Element. Kommt die rewrite Regel im Zuge der Auswertung einer individuellen Zugriffsentscheidungsanfrage zur Anwendung, wird neben dem XSLT Stylesheet das xslt-argument <xacml3:AttributeAssignment> Element übergeben. Der Context Handler nutzt das übergebene Argument, um das zugehörige <xsl:param> Element im XSLT Stylesheet zu initialisieren. Im Beispiel wird der Wert des &content-selector; <xacml3:Attribute> Elements der aktuell betrachteten individuellen Zugriffsentscheidungsanfrage an das XSLT Stylesheet übergeben. Der Context Handler sorgt dafür, dass der Wert des "select" XML Attributs des &init-select-node-argument;:xslt-arg-1 <xsl:param> Elements mit dem Wert des &content-selector; <xacml3:Attribute> Elements belegt wird. Das <xsl:param> Element entspricht daher dem Knoten, auf den sich die individuelle Zugriffsentscheidungsanfrage bezieht (d.h. auf einen der wfs:GetFeature/wfs:Query[@typeName = "Building"] Elementknoten der abgefangenen GetFeature Anfrage).

Im Stylesheet wird geprüft, ob es sich bei dem aktuell zu transformierenden Knoten um den im <xsl:param> Element definierten Kontextknoten der individuellen Zugriffsentscheidungsanfrage handelt (s. Z. 41 und 42). Ist dies der Fall, wird für diesen Knoten die modify-query XSLT Template Rule aufgerufen, die dafür sorgt, dass das entsprechende <wfs:Query> Element wie oben beschrieben umgeschrieben wird. Durch das an das Stylesheet übergebene Argument wird gesteuert, welche Knoten von den Modifikationsanweisungen betroffen sein sollen. Nur durch die Übergabe des XPath Ausdrucks, der den Knoten selektiert, auf den sich die individuelle Zugriffsentscheidungsanfrage bezieht, kann im XSLT Stylesheet stets der gewünschte Teilbaum des Evaluationskontexts und ausschließlich dieser modifiziert werden. Riefe man die modify-query XSLT Template Rule immer dann auf, wenn der zu transformierende Knoten vom Typ "wfs:Query" ist, würden fälschlicherweise auch <wfs:Query> Elemente umgeschrieben werden, die sich nicht auf die Building Featureklasse beziehen.

Das letzte unterhalb des rewrite <xacml3:ObligationExpression> Elements definierte <xacml3:AttributeAssignmentExpression> Element steuert, welche Repräsentation der im Evaluationskontext beschriebenen OWS Nachricht zurück ins originäre Format zu transformieren ist. Details zur Semantik und Verwendung dieses Elements finden sich in Abschnitt 4.4.6.4.

Wenn ein Context Handler die in Abbildung 4.6 (oberer Teil) auszugsweise und schematisch visualisierte multiple Zugriffsentscheidungsanfrage erzeugt und die entstehenden individuellen Zugriffsentscheidungsanfragen anschließend gegen die Regel aus Listing 4.14 ausgewertet werden, wird die Repräsentation der abgefangenen GetFeature Anfrage im

Evaluationskontext wie in Abbildung 4.6 (unterer Teil) gezeigt modifiziert.

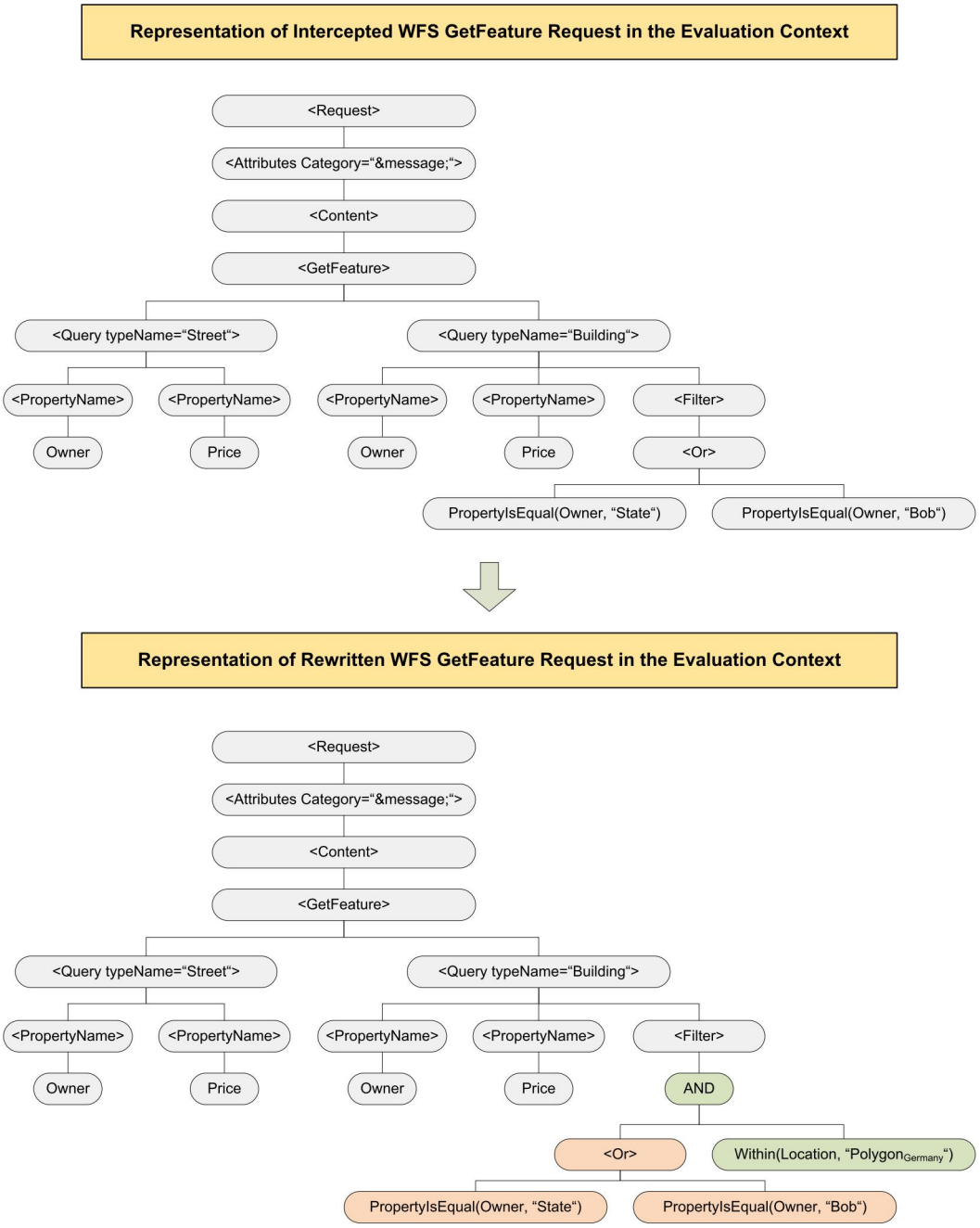


Abbildung 4.6.: Auszug einer multiplen XACML Zugriffsentscheidungsanfrage vor und nach dem Rewrite