

The Axlog Guide

Bogdan-Eugen MARINOIU

July 22, 2009

Chapter 1

The User's Guide

1.1 Defining views: the axlog subscriptions

```

    subscribe
    subscribe
    subscribe
    subscribe
    @Axlog
    vice. AxlogSer-
    http://localhost:6969/MyPeer/services/AxlogService.
    MyPeer subscribe localhost subscribe
    subscribe subscribe
    n/a. subscribe
    subscribe subscription.
```

1.1.1 Active Document

```

    document
    subscription
    document
    docID document
    <document docID="doctest.xml"/>.
    peerURL
    where peer.
    peer
    document where
    included document
```

subscription-
query.

1.1.2 Tree-pattern Query

subscription-
query.

tree conditions,
nodes
label link label * n
link
/ or //
output, true id

selectionStyle nodes
node text selectionStyle
conditions condition

timecondition

```
1 <condition>  
    <op>i1</op><relation>equ</relation><op>i4</op>  
</condition>
```

```
    i1 i4.
```

```
2 <timecondition>i5 - i1 lt 15000</timecondition>  
    i5 - i1 < 15000, i1 i5  
    i5 i1  
    lt (<) gt (>)  
    lte (<=) gte (>=)
```

1.1.3 The template

```

<id>
<a><b>{ $1 }</b><c>{ $2 }</c></a>
i1 i2.

```

1.1.4 Optional: the BY clause

```

BY
<by type="serviceCall">
  <ns2:address xmlns:ns2="http://futurs.inria.fr/gemo/axml/service/Algebra"
  endpoint="http://localhost:6969/MyPeer/services/ReceiveOperator">
    <ns2:currentID>
      <ns2:peerID>MyPeer</ns2:peerID>
      <ns2:docID>docsub.xml</ns2:docID>
      <ns2:nodeID>task1</ns2:nodeID>
    </ns2:currentID>
  </ns2:address>
</by>

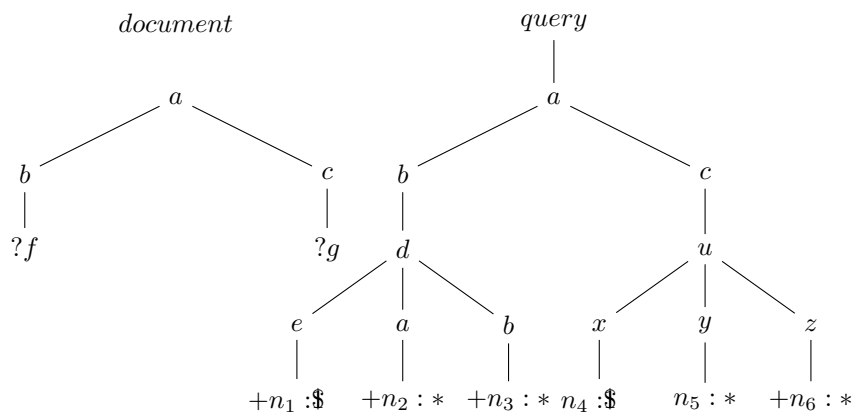
```

1.1.5 An example

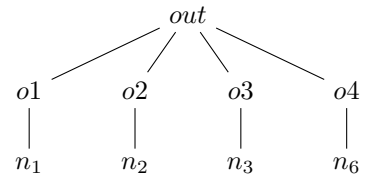
```

n1 d
n4
n1 d n5)
n1, n2, n3 d n6.
doctest.xml
<a xmlns:axml="http://futurs.inria.fr/gemo/axml/">
  <b>
    <axml:sc axml:id="f">
      <axml:return>

```



$\tau(n_5) - \tau(n_1) < \theta \quad s \quad \text{template}$



~~blm~~

```

        <axml:append/>
    </axml:return>
    <axml:ws-soap endpoint="http://localhost:6969/MyPeer/services/DummyStreamService">
        <s:streamToMe xmlns:s="n/a">
            <s:max-timeout>5</s:max-timeout>
            <s:query>for $i in doc('/db/sourcef.xml')/** return $i</s:query>
        </s:streamToMe>
    </axml:ws-soap>
</axml:sc>
</b>
<c>
    <axml:sc axml:id="g">
        <axml:return>
            <axml:append/>
        </axml:return>
        <axml:ws-soap endpoint="http://localhost:6969/MyPeer/services/DummyStreamService">
            <s:streamToMe xmlns:s="n/a">
                <s:max-timeout>10</s:max-timeout>
                <s:query>for $i in doc('/db/sourceg.xml')/** return $i</s:query>
            </s:streamToMe>
        </axml:ws-soap>
    </axml:sc>
</c>
</a>

```

~~Esprit~~
~~peut-être~~
~~à la~~

docsub.xml.1
doctest.xml,

BY u

```

<trace xmlns:axml="http://futurs.inria.fr/gemo/axml/">
  <monitoringTask>
    <axml:sc axml:id="task1">
      <axml:return>
        <axml:append/>
      </axml:return>
      <axml:ws-soap
        endpoint="http://localhost:6969/MyPeer/services/AxlogService">
        <s:subscribe xmlns:s="n/a">
          <s:subscription>
            <document docID="doctest.xml"/>
            <query>
              <tree>
                <node label="a" link="/">
                  <node label="b" link="/">
                    <node label="d" link="/">
                      <node label="e" link="/">

```

```

        <node id="i1" label="*" link="/" output="true"/>
    </node>
    <node label="a" link="/">
        <node id="i2" label="*" link="/" output="true"/>
    </node>
    <node label="b" link="/">
        <node id="i3" label="*" link="/" output="true"/>
    </node>
</node>
</node>
<node label="c" link="/">
    <node label="u" link="/">
        <node label="x" link="/">
            <node id="i4" label="*" link="/">
        </node>
        <node label="y" link="/">
            <node id="i5" label="*" link="/">
        </node>
        <node label="z" link="/">
            <node id="i6" label="*" link="/" output="true"/>
        </node>
    </node>
</node>
</node>
<conditions>
    <condition>
        <op>i1</op><relation>equ</relation><op>i4</op>
    </condition>
    <timecondition> i5 - i1 lt 150000 </timecondition>
</conditions>
</tree>
</query>
<template><out><o1>{i1}</o1><o2>{i2}</o2><o3>{i3}</o3><o4>{i6}</o4></out>
</template>
<by type="serviceCall">
    <ns2:address xmlns:ns2="http://futures.inria.fr/gemo/axml/service/Algebra"
        endpoint="http://localhost:6969/MyPeer/services/ReceiveOperator">
        <ns2:currentID>
            <ns2:peerID>MyPeer</ns2:peerID>
            <ns2:docID>docsub.xml</ns2:docID>
            <ns2:nodeID>task1</ns2:nodeID>
        </ns2:currentID>
    </ns2:address>
</by>
</s:subscription>
</s:subscribe>

```



```

    </axml:ws-soap>
  </axml:sc>
</monitoringTask>
</trace>

```

1.2 Configuration

Figure 1.1

web.xml file

- *host_name* localhost
- *peer_url* http://localhost:6969/MyPeer
- *exist_url* xmldb:exist://localhost:6969/exist/xmlrpc/db
- *port* 0
- *timestamping* false
- *reevaluation* true
- *persistence* no *yes*
- *test* no *yes*
- *storage* no *yes*
- *debug* no *yes*

1.3 Installing AXML with Axlog and testing it

Figure 1.2

1.3.1 Install Binaries

```
1) unzip AXMLaxlogDistrib1_0.zip, exist.war et demo.zip en
   h Files en ICDE2009proj en
   s https://gforge.inria.fr/frs/?group_id=1356

2) unzip AXMLaxlogDistrib1_0.zip, exist.war
   MyPeer et peer1

3) exist.war
   MyPeer en
   peer1

4) CATALINA_HOME en
```

1.3.2 Obtain the sources and create a development environment

```
1) ant build.xml en
   ant b

2) <AXLOG_HOME> en
   https://gforge.inria.fr/scm/?group_id=1356

3) AXLOG_HOME en
   <AXLOG_HOME> en

4) build.xml en
   build.xml en <AXLOG_HOME> en
   <AXLOG_HOME>/services/subscription
```

1.3.3 Demonstration

```
MyPeer. en
test en yes en
MyPeer. en
ls
ls docsub.xml, doctest.xml, sourcef.xml et sourceg.xml. en
g /db/MyPeer, source*.xml en /db/
http://localhost:6969/MyPeer. en
docsub.xml et task en
doctest.xml en
```

afistch fbbch
fch doctest.xml, fch f ch
fch sourcef.xml) ch g ch
fch sourceg.xml) fch
ch docsub.xml.

Chapter 2

The Developer's Guide

queries. *active updates, end-of-stream d time*

<http://jacop.osolpro.com/>

2.1 Model (axlog.model.query)

QueryTree -

QueryNode -

Constraint

2.2 Model to XQuery (axlog.querytranslator)

```

class QueryTranslator {
    QueryTranslator() {
        generateXQuery*
    }
    plan
}

```

2.3 Engine (axlog.engine)

```

class Engine {
    ViewCompiler, a Maintainer
    Tuner
    Application
    ...
}

```

2.4 View (axlog.view)

```

class View {
    Manager Views
}

```


stg (6) 11
h

2.8 Periodic Tasks (axlog.timedtasks)

h (6) h (6) h (6)
h (6) h (6)
h (6) h (6)
h (6)
h (6) h (6) h (6)
h (6) h (6) h (6)
h (6) h (6) h (6)
h (6) h (6) h (6)
h (6) h (6) h (6)
h (6) h (6) h (6)
h (6) h (6) h (6)
h (6) h (6) h (6)
h (6) h (6) h (6)
h (6) h (6) h (6)
h (6) h (6) h (6)

Bibliography

1. *Journal of the ACM*, 1961, 4(1), 1-22.
2. *Journal of the ACM*, 1961, 4(1), 23-36.
3. *Journal of the ACM*, 1961, 4(1), 37-50.

4. *Journal of the ACM*, 1961, 4(1), 51-64.
5. *Journal of the ACM*, 1961, 4(1), 65-78.
6. *Journal of the ACM*, 1961, 4(1), 79-92.

7. *Journal of the ACM*, 1961, 4(1), 93-106.
8. *Journal of the ACM*, 1961, 4(1), 107-120.
9. *Journal of the ACM*, 1961, 4(1), 121-134.

Proc. PODS, 1998