Active XML: Peer-to-Peer Data and Web Services Integration

Serge Abiteboul, Omar Benjelloun,
Ioana Manolescu, Tova Milo, Roger Weber

INRIA, Tel-Aviv University, ETH-Zurich

What is Active XML (AXML)?

```
<directory>
  <dep name="Toy“>
    <sc>toy.xyz.com/GetToyPersonel()</sc>
  </dep>
  <dep name=“DVD“>
    <sc>dvd2000.com/GetDVDPersonnel()</sc>
  </dept>
</directory>
```

AXML documents: XML documents with embedded calls to (AXML) web services.

AXML web services: defined using XQuery over AXML documents.

```
let service Get-Toy-Personnel( ) be
  for $a in document("toy.xyz.com/members.axml")/member,
      $b in $a//name,
      $c in $a//phone,
      $d in $a//pda
   return
   <person pname={ $b/text() }> { $c } { $d } </person>
```

Peer-to-peer architecture, where each peer:

- manages AXML documents
- provides AXML web services

⇨ The Goal: scalable data integration

AXML Documents

```
<knownAuctions ID="peer10">
  <category name="Toys">
    <sc>eBay.net/getOffers("Toys")</sc>
    <auction id="1254" >
      <heldBy>eBay.net</heldBy>
      <item>Pink panther</item>
    </auction>
    …
    <sc>babel.org/translate("Czech",
                            "English",
      <sc>crystal.cz/getToys()</sc>)</sc>
    <sc>peer25/getAuctions([../@name])</sc>
  </category>
  ...
  <sc frequency="once" >getMyAuctions()</sc>
  …
</knownAuctions>
```

May contain calls to any SOAP web service
- e-bay.net, google.com, babel.org, etc.
- AXML peers also offer web services.
- Locally defined services can be called.

Are enriched by each service call's results

The returned nodes are inserted as brothers of the corresponding <sc> element.

Can use XPath expressions for call parameters

Relative path expressions are evaluated starting from the <sc> element.

Activation of calls and data lifespan are controlled
- frequency: when is the service called ?
- validity: how long is the retrieved data kept ?
- mode: immediate or lazy ?

⇨ AXML Documents are powerful data integrators.

AXML Services

A simple, declarative way to create web services…

- A service operation is specified as a query with parameters.
- It may query (local) AXML documents.
- It is made available on the web using the SOAP protocol.

```
let closeAuction($a) be
  for $b in $a/bid
  where $b/amount = max($a/bid/amount)
return
  <sc mode="immediate" frequency="once">
    notifyWinner($b/who, $a/aID, $b/amount)
  </sc>
  <status>closed</status>
```

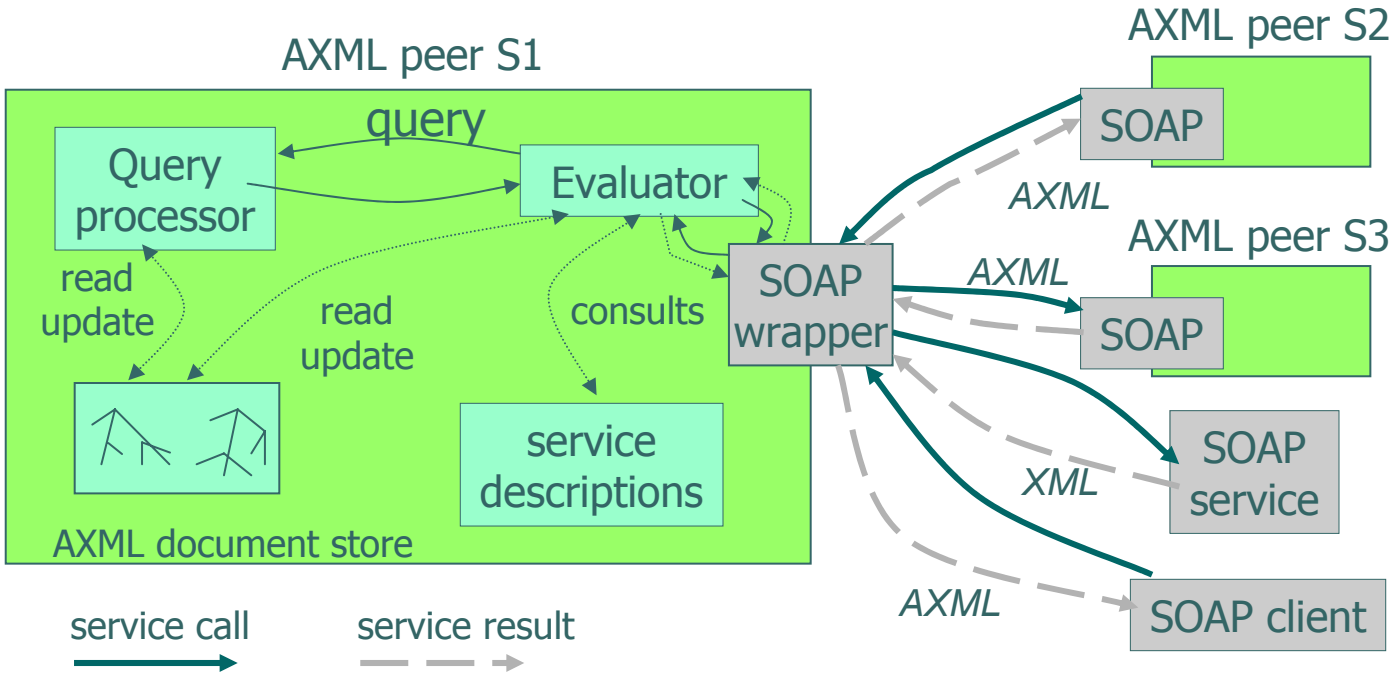
⇨ Basic AXML services are compatible with current standards for web services invocation.

… which allows for new, powerful features.

- Intentional parameters and results: AXML documents (containing service calls) can be exchanged.
- Continuous services send back a stream of answers (SOAP messages) to the caller.

⇨ Used in AXML documents, AXML services are powerful tools for data integration.

AXML Architecture



Technical environment:

- SUN's Java SDK 1.4 (includes XML parser, XPath processor, XSLT engine)
- Apache Tomcat 4.0 servlet engine
- Apache Axis SOAP toolkit 1.0 beta 3
- X-OQL query processor, persistant DOM repository
- JSP-based user interface, using JSTL 1.0 standard tag library

Communications

- Peer to peer:
  - Peers communicate together only through service calls, using SOAP.
  - They don't directly access other peers' documents.

- User interface:
  - Users interact with each peer through an HTML-based interface.
  - AXML Documents are displayed using XSLT stylesheets.
  - Transformations are done server-side using JSP pages.
  - Forms allow to invoke locally defined services, which update the documents.

⇨ Specific distributed applications can be easily built as AXML documents and services, and XSLT stylesheets.

Demonstration highlights

Each peer provides some auctions:

- The document myAuctions.xml contains the peer's items and their bids
- Services offered to other peers:
  - getAuctions(),
  - getHighestBid(auctionId),
  - bid(auctionId, amount)

Each peer knows about some peers' auctions:

- knowAuctions.xml is an AXML document, containing calls to other peers that transitively retrieve their know auctions.
- Offered service: getKnownAuctions()

Each peer can bid on any auction:

- myBids.xml keeps track of the peer's bids
- Manually, using the Bid(auctionId, amount)
- Automatically, using the local service bidUpTo(peer, auctionId, increment, limit)

When an auction closes, the winner is notified.

⇨ Functional bidding system, without a centralized server