Joshua Shinavier

Semantic Web agents for real-time applications

TetherlessWorld

Wednesday Nights in the Tetherless World (TWed)
December 14th, 2011
Outline

• motivation and concepts
  • what are agents?
    • conversational protocols
  • what is real time?
    • the human limits of real time
    • technical limits of reactivity and throughput
  • tools
    • TwitLogic, RDFAgents, others
Motivation and concepts
Imagine that...

• ...you’re walking down a busy sidewalk in NYC.
• A lady passes you in the opposite direction. At just that moment, a little voice in your pocket says “Medha Atre”.
• “Who’s Medha Atre?” you ask.
• “Medha Atre is a graduate student at the Tetherless World Constellation. You are also a graduate stu...”
• “OK, how do you know?”
• “Source is the Tetherless World Constellation home page.”
• You catch up with Medha and say “hi”.
Wouldn’t it be nice if...

- your devices could share information **proactively** with my devices, and vice versa
- like a **conversation** between peers
- they could do it very quickly
- over the shortest available network path
- using any available communication protocol
- they didn’t require the cooperation of some authoritative mega-peer
- information were shared in such a way as to
  - preserve info **provenance**
  - maintain **privacy** and info **accountability**
That sounds like Semantic Web agents

“The real power of the Semantic Web will be realized when people create many programs that collect Web content from diverse sources, process the information and exchange the results with other programs. The effectiveness of such software agents will increase exponentially as more machine-readable Web content and services becomes available.”

— Tim Berners-Lee et al., 2001
By “agent”, we mean...

- peer-to-peer communication
- proactive, event-driven information sharing
- support for a variety of interaction protocols
  - e.g. for query answering, subscriptions, contracts
- conventions for provenance tracking

(not that kind of agent.)
Agent communication is an alternative to RPC

- no distinguished client and server roles
- data is both **pushed** and **pulled**
- agent **conversations** resemble human speech acts
A basic conversation: query answering

- Agent $A \rightarrow$ Agent $B$: “Who is Josh?”
- Agent $B \rightarrow$ Agent $A$: “Josh is student in the Tetherless World.”
A basic conversation: subscription and data stream

- Agent A → Agent B: “Keep me up to date about the Tetherless World.”
- Agent B → Agent A: “Will do!”
- Agent B → Agent A: “Josh has just written this tweet about the Tetherless World: ...”
- Agent B → Agent A: “Xixi has just checked in to the Tetherless World (Winslow building) in Troy”
More complex conversations: query delegation with provenance

• Agent A → Agent B: “Who is Josh?”
  - Agent B → Agent C, Agent D: “Who is Josh?”
  - Agent C → Agent B: “Josh is a student in the Tetherless World.”
  - Agent D → Agent B: “Josh has the Twitter handle @joshsh and has written these tweets: ...”

• Agent B → Agent A: “Agent C says that Josh is a student in the Tetherless World. Agent D says that Josh has the Twitter handle @joshsh and has written these tweets: ...”
More complex conversations: syndication with provenance

- Agent A → Agent B: “Keep me up to date about the Tetherless World.”

- Agent B → Agent A: “Will do!”
  - Agent B → Agent C: “Keep me up to date about the Tetherless World.”
  - Agent C → Agent B: “Will do!”
  - Agent C → Agent B: “Josh has just written this tweet about the Tetherless World: ...”

- Agent B → Agent A: “Agent C says that Josh has just written this tweet about the Tetherless World: ...”
Segue: what is real time?

- on the **real-time Web**, data is pushed from producers to consumers
- **real-time computing** is subject to constraints of response time or **latency**
- real-time systems are highly **reactive**
- results which arrive too late are less useful, or even useless
- we’ll combine both definitions:
  - **real-time Semantic Web** is fast, event-driven RDF data flow
- how fast is fast enough?
The human limit of real time

- **simultaneity** ⇔ no perceivable delay
- anywhere from 2ms (intramodal, auditory) to **40ms** (cross-modal, e.g. sound and touch) [Levitin et al. 2000]
- central to human sensory integration
- with a human in the loop:
  - lower latency is overkill
  - higher latency is distracting
- a practical **real-time constraint**: simultaneous or faster
Spot the bottleneck: the speed of light

- convert some social data into RDF/XML: ~1ms
  - one tweet in TwitLogic
- ingest RDF data into a triple store: ~4ms
  - 20-triple transaction with batch commit
- compute a SPARQL query result: ~5ms
  - 3 triple patterns, 300 intermediate results
- send and receive a FIPA message: ~1ms
  - in RDFAgents, excluding network latency
- round-trip between NY and CA: ~85ms
  - ideally, in a vacuum: 28ms
What about throughput?

- a single connection between remote servers can carry
  - most of the Twitter Firehose
    - around 1,000 tweets/s (as of a study last year)
  - RDF streams of equivalent volume
    - around 20,000 triples/s
- existing triple stores (e.g. AllegroGraph) can ingest data at these rates
Data throughput using TCP + HTTP POST
Data throughput using UDP
The point is...

- human and technical limits of real-time are comparable
- existing software operates close to those limits
- the Semantic Web is ready for demanding real-time applications, but
  - peer-to-peer architectures are required which cut down on network latency
  - SemWeb needs to extend to **streams** and **conversations** as well as documents and service requests
Tools
• capture Twitter data
• translate it into RDF
• produce an RDF stream
• preserve provenance
• provide real-time, continuous search

http://twitlogic.fortytwo.net
@prefix sioc: <http://rdfs.org/sioc/ns#> .

[...] 

{ 
  <http://twitlogic.fortytwo.net/post/twitter/16810997455> 
  a sioc:MicroblogPost ;
  dc:created "2010-06-23T00:43:35.000Z"^^xsd:dateTime ;
  geo:location <http://twitlogic.fortytwo.net/location/twitter/d0fc0f618c1eb790> ;
  sioc:content "the Karate Kids: karate vs. kung fu..." ;
  sioc:has_creator <http://twitlogic.fortytwo.net/user/twitter/7083182> ;
  sioc:topic
    <http://twitlogic.fortytwo.net/topic/2121544629>,
    <http://twitlogic.fortytwo.net/topic/1462827592> ;
  sioc:embeds_knowledge <http://twitlogic.fortytwo.net/graph/twitter/16810997455> .

  <http://twitlogic.fortytwo.net/graph/twitter/16810997455> 
  a rdfg:Graph .

  <http://twitlogic.fortytwo.net/location/twitter/d0fc0f618c1eb790> 
  dc:title "Half Moon Bay, CA" ;
  geo:lat "32.528832"^^xsd:double ;
  geo:long "-124.482003"^^xsd:double ;
  rdfs:seeAlso <http://api.twitter.com/1/geo/id/d0fc0f618c1eb790.json> ;
  rdfs:label "Half Moon Bay" ;
}

<http://twitlogic.fortytwo.net/graph/twitter/16810997455> { 
  <http://twitlogic.fortytwo.net/topic/2121544629> 
  a dbo:Film ;
  dc:title "The Karate Kid" ;
  rdfs:label "The Karate Kid (1984)" ;
  foaf:page <http://www.rottentomatoes.com/m/karate_kid> ;
  owl:sameAs <http://dbpedia.org/resource/The_Karate_Kid> .

  <http://twitlogic.fortytwo.net/topic/1462827592> 
  a dbo:Film ;
  dc:title "The Karate Kid" ;
  rdfs:label "The Karate Kid (2010)" ;
  foaf:page <http://www.rottentomatoes.com/m/karate_kid_2010> ;
  owl:sameAs <http://dbpedia.org/resource/The_Karate_Kid_%282010_film%29> .
}
Use case: “stream me tweets about...
...places in developing countries”

PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>
PREFIX owl: <http://www.w3.org/2002/07/owl#>
PREFIX sioc: <http://rdfs.org/sioc/ns#>
PREFIX geonames: <http://www.geonames.org/ontology#>
SELECT DISTINCT ?post WHERE {
  GRAPH ?graph {
    ?place rdf:type geonames:Feature .
    ?place owl:sameAs ?geoplace .
  }.
  ?country owl:sameAs ?dbc .
  ?dbc rdf:type <http://dbpedia.org/class/yago/LeastDevelopedCountries> .
}
Use case: “stream me tweets about...
...English-language movies starring Chinese actors”

```
PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>
PREFIX owl: <http://www.w3.org/2002/07/owl#>
PREFIX sioc: <http://rdfs.org/sioc/ns#>
PREFIX dbo: <http://dbpedia.org/ontology/>
SELECT DISTINCT ?post WHERE {
    GRAPH ?graph {
        ?movie rdf:type dbo:Film .
        ?movie owl:sameAs ?dbmovie .
    } .
}
```
Use case: “stream me tweets about...
...songs by artists my friends like”

PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>
PREFIX owl: <http://www.w3.org/2002/07/owl#>
PREFIX sioc: <http://rdfs.org/sioc/ns#>
PREFIX foaf: <http://xmlns.com/foaf/0.1/>
SELECT DISTINCT ?post WHERE {
  GRAPH ?graph {
    ?song rdf:type mo:Track .
    ?song owl:sameAs ?mbtrack .
  } .
}
RDFAgents = FIPA + data streams

- **RDFAgents** specification
  - extends **FIPA** agent communication standards
  - enables query answering and real-time data streams *with provenance*

- includes:
  - RDF content languages
  - two interaction protocols: **Query** and **Pub-Sub**

- see
  - [https://github.com/joshsh/rdfagents](https://github.com/joshsh/rdfagents)
RDFAgents queries

consumer

pose query

provider

refuse to answer
(with explanation)

agree to answer
(optional)

inform of query result
(RDF content)
RDFAgents subscriptions

subscribe

refuse subscription (with explanation)

agree to subscription

inform of update (RDF content)

0..n

cancel subscription

confirm cancellation
An RDFAgents message

(inform-ref
  :sender (agent-identifier
    :name http://example.org/twitlogic
    :addresses (sequence xmpp:twitlogic@example.org))
  :receiver (set (agent-identifier
    :name http://example.org/consumer
    :addresses (sequence xmpp:consumer@example.org)))
  :protocol fipa-subscribe
  :conversation-id c976b710a5
  :language rdf-trig
  :content "
@prefix dc: <http://purl.org/dc/terms/> .
@prefix ex: <http://example.org/> .
@prefix sioc: <http://rdfs.org/sioc/ns#> .
@prefix sioc: <http://rdfs.org/sioc/types#> .
@prefix xsd: <http://www.w3.org/2001/XMLSchema#> .

{ ex:post1785782813 a sioc:MicroblogPost ;
  dc:title "#SemTech begins again!" ;
  dc:created "2011-06-06T19:43:35.000Z"^^xsd:dateTime ;
  sioc:topic <http://twitlogic.fortytwo.net/hashtag/semtech> .
}")
Other useful tools and frameworks

- **C-SPARQL**: SPARQL for continuous queries
- **SPARQLPuSH**: creates RSS feeds and PubSubHubbub (PuSH) hubs for continuous SPARQL results
- **SMOB** semantic microblogging framework and SMOB 2, including Sharing Spaces
- Android Semantic Web Core Library
- RDF on the Go
- many other mobile Semantic Web clients and publishing tools, including DBpedia Mobile, mSpace Mobile, OntoWiki Mobile
Some useful vocabularies

- **SWP** (Semantic Web Publishing) vocabulary for provenance in publishing and syndication
- **SIOC** ontology for online community artifacts (e.g. blog and microblog posts)
- **FOAF** (Friend of a Friend) vocabulary for linking people and information
- **OPO** (Online Presence Ontology) for describing user presence and visibility
• **Open Sound Control** (OSC) is a messaging protocol for multimedia software

• an OSC message contains:
  • an address pattern: identifies the recipient (device)
  • a tuple of typed arguments

• SPARQL-OSC translates SPARQL result tuples into OSC messages

• lets multimedia devices react to Semantic Web data streams

• [https://github.com/joshsh/sparql-osc](https://github.com/joshsh/sparql-osc)
Even more tools: SesameTools

- miscellaneous components for use with the OpenRDF toolkit, incl.
  - Sesame transaction library (for producing streams)
  - transaction logging and replay layers
  - URI mapping layer
  - N-Quads, RDF/JSON, and JSON-LD libraries
- [https://github.com/joshsh/sesametools](https://github.com/joshsh/sesametools)
Even more tools:

- Ripple is a Turing-complete **RDF path language** in which every program is an RDF list
- can be embedded in Linked Data or RDF streams
- follows a functional “**pipes and filters**” pattern
  - for push-based RDF data flow
- [http://ripple.fortytwo.net](http://ripple.fortytwo.net)
Even more tools:
TinkerPop’s Semantic Web suite

- TinkerPop: pluggable components for high-performance graph databases
- there are SemWeb components to:
  - make a graph database into a triple store
  - map between RDF and graph data models
  - apply fast graph algorithms to RDF data
- [http://tinkerpop.com](http://tinkerpop.com)
Thanks!

- Tetherless World Constellation: [http://tw.rpi.edu](http://tw.rpi.edu)
- Institute of Automation: [http://english.ieia.cas.cn](http://english.ieia.cas.cn)
- Franz Inc: [http://www.franz.com](http://www.franz.com)
- TinkerPop: [http://tinkerpop.com](http://tinkerpop.com)
- Contact:
  - josh@fortytwo.net, @joshsh
forty-two.