A Brief Introduction to Rover

CMSC818G

• Spring 2010
• 04 February 2010

Presenter

• Christian Almazan
Objectives

- High level overview of Rover:
  - What is the purpose of Rover?
  - What does it do?
  - What does it not do?
  - How can you use Rover?

- Make sure you understand Rover enough to evaluate how to use it in your projects.
Background: Organizational Units

- Identify Similarities and Differences:
  - Employee Records
  - Schemas / Ontologies
  - Relationships
  - Services
  - Interactions
Overall Goals of Rover

- Managing and Mitigating Context
- Data Sources and Services
- Uniform Access
- Messaging
- Logging

04 February 2010  A Brief Introduction to Rover 4
What Rover Does Not Address

- Usability / Human-Computer Interaction
- Authorization
- Privacy Issues
- Dissonant Information
- Misinformation
- Methods of Reasoning
- Schema and Ontology Development
- Distributed Version of Rover
Rover Ecosystem and Entities

User Tier

Assistance Tier

Server Tier

Utility Tier

Users

Participating

Non-Participating

Rover Server

Services

Context Watchers

Context Providers
A Brief Introduction to Rover

04 February 2010
Authentication

- Credential-Based Authentication
  - Domain
  - Username
  - Password

- Each domain has an associated checker:
  - One checker could be used for multiple Rover ecosystems.
  - More than one checker allowed per server.
  - Checkers may be local or remote.
Authentication Process

A Brief Introduction to Rover

04 February 2010  A Brief Introduction to Rover  9
Context Representation

- Similar to an RDF triple (forms a graph):
  - \{subject, predicate, object\}
- Our context entries look as follows:

<table>
<thead>
<tr>
<th>identifier</th>
<th>Each entry on a Rover server has a unique ID.</th>
</tr>
</thead>
<tbody>
<tr>
<td>subject</td>
<td>Domain/Username – Who the entry describes.</td>
</tr>
<tr>
<td>predicate</td>
<td>Root/Leaf – Relationship between subject/object.</td>
</tr>
<tr>
<td>object</td>
<td>The value of the entry.</td>
</tr>
<tr>
<td>creator</td>
<td>Who created the entry.</td>
</tr>
<tr>
<td>evidence</td>
<td>Supporting data why the entry exists, if available.</td>
</tr>
<tr>
<td>timestamp</td>
<td>When the context arrives at the Rover server.</td>
</tr>
</tbody>
</table>
Context Entry Example

Entity

Root

Leaf

Leaf

Leaf

Leaf

Christian

Location

Physiological

GPS

Building

Hunger

Thirst
Context Methods

- Subscription to Context
  - Subscribe (*WatchContext*)
  - Unsubscribe (*UnwatchContext*)

- Querying Context
  - Field Query (*QueryContext*)
  - By Reference (*QueryContextByReference*)

- Updating Context
  - Your Own (*InformContext*)
  - Others (*SupplyContext*)
Quick Note on Active Context

- Context that can be query immediately will be stored as active context.
  - The Rover server logs all context updates.

- InformContext and SupplyContext has two important parameters:
  - pruneRoot – remove tree containing the root
  - removeLeaf – remove leaf if it is there

- Example – Location Update
Context Watchers and Providers

A Brief Introduction to Rover

[Diagram showing the process flow for context watchers and providers, with steps involving GPS chip, building number, and context updates.]
Services

- Entities which provide a service indicate their availability to the Rover server:
  - Publication (*PublishService*)
  - Removal (*UnpublishService*)

- Entities can use services:
  - List Services (*ListAllServices/ListRelevantServices*)
  - Call Services (*CallService*)

- Services may specify required and optional context, which will be forwarded.
Messaging

- An entity may send a message directly to another entity.
  - SendMessage

- An entity may send a message to other entities which satisfy certain criteria based on context.
  - BroadcastMessage
Historical Information

- The Rover server stores the following:
  - All context updates:
    - Query with `ObtainContextProvenance`
  - All calls to the server:
    - Query with `ObtainCallHistory`
    - Includes all request and response data.

- Use a combination of both to reconstruct a series of events or active context.
Perspective Broker (TCP and TLS)
- Python/Twisted (asynchronous)
- Java/TwistedJava Client (not asynchronous)

HTTP and HTTPS (JSON for objects)
• Mobile devices may disconnect frequently:
  ◦ Use a Rover proxy to store context subscriptions and messages for an entity.
• Published services cannot use a proxy.
V911 Example

User Tier

Assistance Tier

Server Tier

Utility Tier

Users

VLC

Livecast

Rover Server

Records Access

Dispatcher

Campus Status

Participating

Non-Participating

Services

Context Watcher

Context Provider

04 February 2010   A Brief Introduction to Rover
Questions?

- **Rover 5**
  - Class Implementation
  - Not directly backward compatible w/Rover 4

- **Rover 4**
  - Demo Implementation
  - If you want to extend V911, sample code will be made available.