Title
Web Tasking Interface within The Extensible Sensing System

Permalink
https://escholarship.org/uc/item/7fz099f2

Authors
Poonam Jolly
Ben Huang
Andrew Wu
et al.

Publication Date
2003
Web Tasking Interface of the Extensible Sensing System

Poonam Jolly, Ben Huang, Eric Osterweil, Andrew Wu, Deborah Estrin
Laboratory for Embedded Collaborative Studies – http://lecs.cs.ucla.edu

Introduction: The structure and implementation of the Extensible Sensing System (ESS)

- **ESS Description**
  - Designed as a testbed for sensors and user interfaces to be used in environment sensing.
  - Deployed at the James Reserve to facilitate microclimate analysis.
  - Data is communicated through a Publish-Subscribe Bus where the clients are databases, GUIs, and other information sinks.

- **Goals**
  - A means of data flow and collection
  - Allows remote sensor tasking
  - Monitoring of the sensor network state
  - Allow the end-user to interact with the network
  - Provide an external user with these services over the World Wide Web

Problem Description: Lack of a well-defined user interface

- This GUI is designed with the intent of making it easily accessible to any user.
- Reasons for a Web Tasking Interface
  - The need for remote sensor tasking and network monitoring.
  - Means of allowing the end-user to make data requests to the sensor network.
  - To allow scientists to configure and access experiments and data.
  - Simplicity of using a web client allows the average user to access the GUI.

Proposed Solution: Designing an extensible interface to facilitate user interaction

- **User Requests**
  - Several webpages are created to allow users to reprogram and monitor various aspects of the network.
    - Type of data collected
    - Number of nodes deployed
    - Data collection specifications

- **Services**
  - The user’s data request is parsed and input to a storage class through a CGI program.
  - This user input is then serialized and sent over a Publish-Subscribe bus.
  - The message is received by the ESS Handler.

- **Tasks**
  - The Handler then passes the message into the tasking class corresponding with the service request being received.
  - The tasking class unpacks data from the message.
  - The tasking class provides for the appropriate execution of the tasks specified.
  - This data is then analyzed within the network and the results are sent back to the user over the pub-sub bus.

Acknowledgements:

- Jamie Burke
- Thanos Stathopoulos
- Alberto Cerpa
- Yan Yu
- Tom Schoellhammer
- LECS Lab