qiWorkbench™ – an extensible open-source platform for seismic interpretation

Gil Hansen
Michael Glinsky

What is the qiWorkbench?

- Component-based application
  - qSefCommanders = 2D, 3D, well log
  - qCommanders = Amplitude Extraction, Wavelet Decomposition, XMLeditor
  - qServices = text IO, seismic IO, job, file chooser dialog, error dialog
- Message-based architecture - message framework supports component communication via messages
- State saving, state restoring capability
- Web-based = install workbench on any client machine from server deployed to
  - Major client = majority of functionality and processing on client machine
- Platform independent – written in pure Java; runs on Windows, Linux, MacOS
- Extensible – add core or external components (application components)
- Open Source; under GPL license
- Component API allows 3rd party vendors to build commercial components; under BSD license
- New messages registered to avoid duplication and conflicts
- Patent pending

Architecture

- Components interact via message
  - CID of producer (CID is a unique system-wide ID for component)
  - CID of consumer
  - Kind of Message – command with arguments or data
  - Content type
  - Content – arbitrary object
  - Status – processing status of a request
  - Message Dispatcher – routes client-side request and response messages
  - Dispatcher Connector – communication interface between the Message Dispatcher and the Serviet Dispatcher
  - Serviet Dispatcher – routes server-side request and response messages
  - Messages transmitted over the network are serialized
  - Message hander per component – manages message queue (enqueue, dequeue)

Message Framework

Messaging Manager

- Communication interface between a component and the message dispatcher
- Messaging manager per component – communicates via component’s message handler
- Artifact of licensing
- Functionality provided
  - Manage component’s message queue via component’s message handler
  - High-level message methods – route message to message dispatcher
  - Register and unregister component with message dispatcher
  - Retrieve information maintained by message dispatcher (e.g., component descriptor of a registered component)
  - Extract parts of a message

qiComponents™

- qCommanders – send requests, receive responses; Ex: Amplitude Extraction
- qSefCommanders – send and receive requests and responses; Ex: viewers (2D, 3D, well log)
- System qiComponents
  - Workbench Manager – manages workbench GUI and canvas
  - State Manager – save and restore state of active component and workbench upon request
  - qServices – common utilities; Ex: read/write text or seismic data, execute job, file chooser dialog, error dialog

Software Installation

- qiWorkbench requires Java 1.5.0_06 Runtime Environment (JRE) or above be installed on client machine
- qiWorkbench downloaded and installed by Java Web Start from Tomcat server where deployed
- WebStart automatically updates to newer version of qiWorkbench each time application launched

Software Prerequisites

- Java 1.5.0_06 Runtime Environment (JRE) or above with WebStart
- Seismic UnX (optional)
- BHP-SU (optional)
- Both available from Colorado School of Mines at http://www.cwp.mines.edu/cwpcodes/

Development Status

- Available end of June 2006 at qiworkbench.org
- No eCommerce mechanism (yet) to install and update commercial components; currently manual install
- System can recognize, load and launch commercial components
- Message framework and messaging manager in place
- qiComponents: 2D viewer, Amplitude Extraction
  - Local and remote text IO, job services, file chooser dialog, error dialog
- Save and restore state
  - Local and remote seismic IO services being implemented
    - Read/write any variation of segy format (rev 0)
    - Read/write Landmark data
- Implementing Wavelet Decomposition component
- qiWorkbench.org Website for developing qiComponents:
  - Wiki, bug tracking, register new commands, Subversion
- Contact: info@qiworkbench.org

Workbench Screenshot