Usability Extensions for the Worklet Service

Michael Adams

Queensland University of Technology
Brisbane, Australia
mj.adams@qut.edu.au

YAWL Symposium, Sankt-Augustin, Germany, 7 June 2013
The YAWL Worklet Service

- Part of standard deployment since Dec. 2005
- Two main components:
  - Selection: runtime flexibility
  - Exception: runtime exception handling
- Each uses:
  - An extensible repertoire of self-contained processes
  - An extensible Ripple-Down-Rule set (RDR)
Service Architecture

From: Modern Business Process Automation, Figure 11.2, p. 296
Usability Issues

- Platform constraint of the Rules Editor
- Dependencies of the Rules Editor
- Some conceptual complexity
- Offered functionality an optional extra
Service Architecture

From: Modern Business Process Automation, Figure 11.2, p. 296
Worklet Service API

• Two main functionalities:
  – Create, access, update and evaluate RDR sets
  – Provide notifications to event listeners

• RDR sets can now be created to:
  – Support YAWL process executions
  – Any other purpose!
    • Offers RDR support to external applications
API Methods

- **Get** a rule node, tree or entire set
- **Add** a new rule node
- **Evaluate** a given dataset against an RDR set
  - Returning the result (if a rule is satisfied)
- **Process** a given dataset against an RDR set
  - Triggering an exception handler (if a rule is satisfied)

- Rules can be expressed using:
  - the usual numeric, comparison and logical operators
  - XQuery
  - pluggable user-defined functions
Event Listeners

• Listeners implement the `WorkletEventListener` interface and register with the Worklet Service via the API

• Events:
  – `caseLevelExceptionEvent`
  – `itemLevelExceptionEvent`
  – `selectionEvent`
  – `constraintSuccessEvent`
  – `shutdown`

www.yawlfoundation.org
Rules Editor

• A replacement Rules Editor will be added as a plugin to the YAWL Editor
  – Platform independence
  – Less complexity
  – Linked to selected specification or task

• API used for rules management
  – Smaller ‘conceptual overhead’
Conclusion

• Users can now:
  – More easily integrate the Worklet Service into projects.
  – Easily add, manage and evaluate rules.
  – Process (trigger) exceptions externally.
  – Avoid the constraints and learning curve of the existing Rules Editor.
  – Be notified of service events
    • Support exception-handling service chains
Conclusion

• For a novel application of the API, see: