VoIP Application Success

See how the JBPM and JSF drive the big success of VoIP (Voice over IP) Order Management application for a telecommunication company

Peter Wang
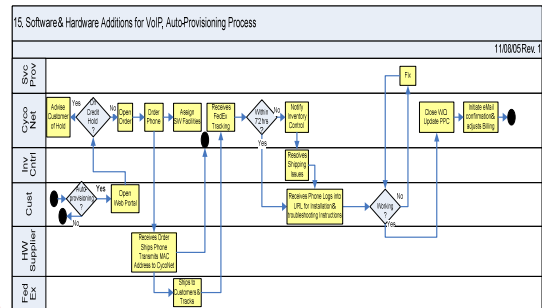**Google House LLC**
June 14th, 2006

Our Problems

- Not clearly defined business process
- Disconnected data sharing and application access
- Lack of IT infrastructure support
- High cost of existing commercial BPM tool

JBPM Winning Points

- Simple process definition language and vocabulary
- Highly pluggable architecture for easy integration
- Flexible GUI integration
- No cost tool and IDE
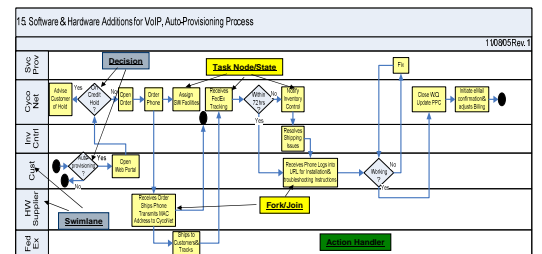- Broader user community support and activities
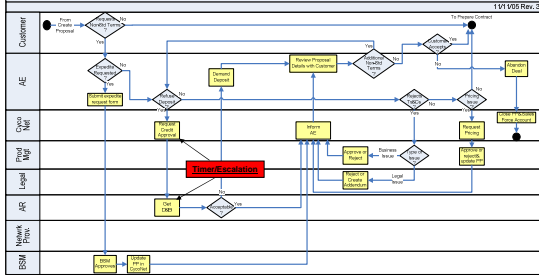

Business Process Swimlane

15. Software & Hardware Additions for VoIP, Auto-Provisioning Process

---

How to Model

- Task Node/State
- Fork/Join
- Swimlane
- Decision
- Action handler
- Timer/Scheduler

How to Model

15. Software & Hardware Additions for VoIP, Auto-Provisioning Process

How to Model

4. New Selling Process – 4.O Negotiate

How to Model

- **Task Node/State**

```
<task-node name="Choose Terms">
    <task swimlane="ae">
        <controller>
            <variable name="pp" />
        </controller>
    </task>
    <event type="node-leave">
        <action name="create pp action"
class="net.googlehouse.jbpm.ordermanagement.UpdateProvisionProfile"></action>
    </event>
    <transition to="Choose Terms Decision" />
</task-node>
```

- **Fork/Join**

```
<fork name="Non Standard Fork">
    <transition name="legalterm" to="Legal Term and Conditions Decision" />
    <transition name="businessterm" to="Business Terms and Conditions Decision" />
    <transition name="dupoiterefuse" to="Get Credit Score Decision" />
    <transition name="expedite" to="Fill Expedite Request Fields Decision" />
    <transition name="priceissues" to="Pricing Issues Decision" />
</fork>
```

- **Swimlane**

```
<swimlane name="ae">
    <assignment class="net.googlehouse.jbpm.BaseRoleAssignmentHandler">
        <role>OM_AE,OM_ADMIN</role>
    </assignment>
</swimlane>
```

How to Model

- **Decision**

```
<decision name="Choose Terms Decision" >
    <task swimlane="ae">
        <controller>
            <variable name="pp" />
        </controller>
    </task>
    <handler class="net.googlehouse.jbpm.ordermanagement.ChooseTermsDecisionHandler"/>
    <transition name="nonstandard" to="Non Standard Fork"></transition>
    <transition name="standard" to="Prepare Contract"></transition>
</decision>
```

- **Action handler**

How to Model

- **Timer/Scheduler**

```
<task-node name="Business Terms and Conditions">
    <task swimlane="pm">
        <timer name='business terms and condition escalation'
            duedate='3 minutes'
            repeat='true'
        >
            <action class='net.googlehouse.jbpm.BaseEscalationRoleAssignmentHandler'>
                <role>OM_BSM</role>
                <swimlane>pm</swimlane>
            </action>
        </timer>
        <controller>
            <variable name="pp" />
        </controller>
    </task>
    <event type="node-leave">
        <action name="create pp action"
class="net.googlehouse.jbpm.ordermanagement.UpdateProvisionProfile"></action>
    </event>
    <transition to="No Standard Join" />
</task-node>
```
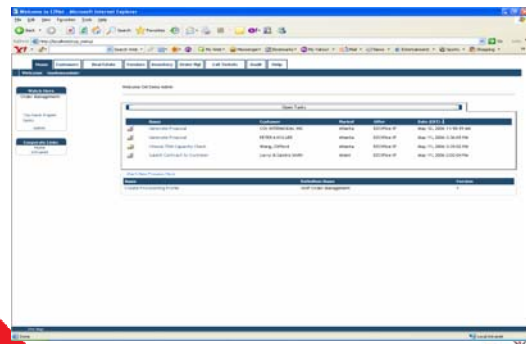
-

Role and Group

- Actor and Pool Actors
- Private View -> Actor
- Role/Group View -> Pool Actors
- Design time assignment handler->Action class in swimlane
- Runtime assignment handler-> action class in task node events

---

Runtime Role Assignment Code

```
private String role = null;
private String swimlane = null;

private String dueDate = null;

private static BusinessCalendar businessCalendar = new BusinessCalendar();

public void execute(ExecutionContext executionContext) throws Exception {

        Duration duration = new Duration(dueDate);
        ProvisioningProfile pp = (ProvisioningProfile)getVariable(executionContext,"pp");
    Date dueDate = businessCalendar.add( pp.getModifiedDate(), duration );

        if (isDue(dueDate)) // time is up , escalate it to the new role
assign(executionContext.getTaskMgmtInstance().getSwimlaneInstance(this.getSwimlane()),ex
ecutionContext);
}

public void assign(Assignable assignable, ExecutionContext executionContext)
            throws Exception {
        // TODO Auto-generated method stub
        assignable.setActorId(null); // back to the pool first
        assignable.setPooledActors(parseRoles(role));
        log.debug("successfully assigned to poolactor "+ role + " for instance "
+executionContext.getTaskInstance().getName());

}
```

JAAS Integration

- Integration with JBoss JAAS framework
- Customized JAAS database
- Leverage existing user and group management

13

JBOSS JAAS Integration

- Add standard J2EE security constrains and roles in Web.xml
- Add application policy in the login-config.xml in JBoss
- Add JBoss-web.xml to WAR to map the application policy entry in the login-config.xml
- Add user information by looking up by user id in the web authentication filter

14

Put the Face on

- Choice of GUI –Java Server Face
- Rich components set provided by JSF and My Faces open source tool
- Nature integration with JBPM variables and JSF managed bean model
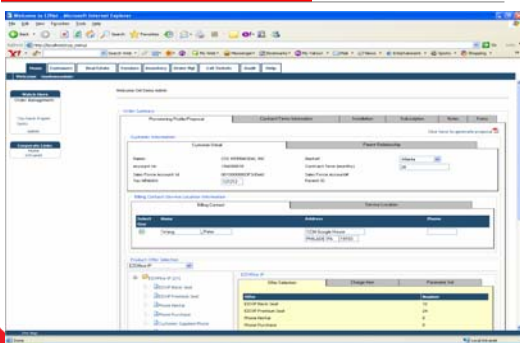- Easy Portal and portability support for flexible GUI
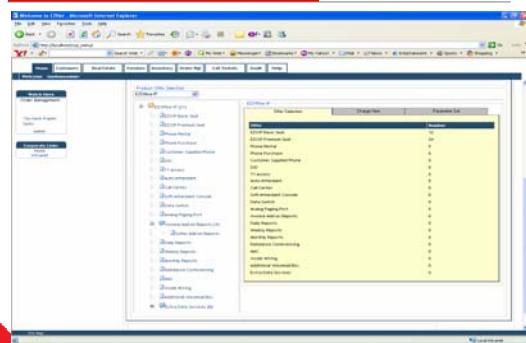
15

Put the Face on

16

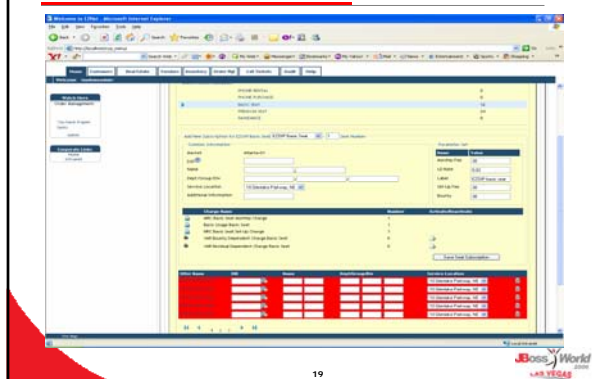## Put the Face On

17

Put the Face on

18

Put the Face On

19

Conclusions

- JBPM provides the simple tool sets that can solve the complex telecommunication process
- Open source JSF/MyFaces offers rich components and build-in portal integration
- Open source rocks -> JBPM,JSF/MyFaces, JBoss and Eclipse
- A complete development/deployment suite

20

Q & A

Any Questions?
- **Thank You!**

- *Visit us at http://www.googlehouse.com*
  - *pwang@googlehouse.com*
    - *678-923-2719*

21