JBoss innovation AWARD WINNER 2006

JBoss World 2006 LAS VEGAS

LEXICON

Using JEMS to deploy truly mission-critical applications

A seamless migration

Pre-flight Check

- Innovation Award Track
- Seam, jBPM

Overview

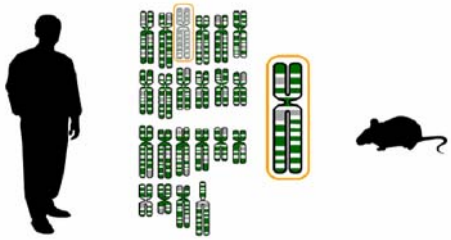
- Company Overview
  - Lexicon Genetics
  - TIGM
- Project Objectives
- Application Overview
- Lessons Learned

Lexicon Genetics: A Research-driven, Integrated Drug Discovery Company

- Founded in 1995
- Headquarters in The Woodlands, TX
- Medicinal chemistry operations in Princeton, NJ
- 770 employees
  - 265 with Ph.D.s, M.D.s or other advanced degrees
  - 102 employees at Princeton medicinal chemistry facility
- Financing:
  - More than $300 million in public and private equity financing
  - More than $350 million in partnership funding

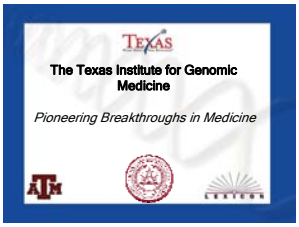
Lexicon's Mouse Gene Knockout Technology

99% of human genes have a mouse counterpart



"The laboratory mouse is hailed as holding the experimental key to the human genome. Working on mouse models allows the manipulation of each and every gene to determine their functions, and will give us detailed insights into many aspects of human diseases as well as basic human biology." *--Nature, Vol. 420. December 5, 2002*

The Texas Institute for Genomic Medicine



- Lead the next stage of the human genome project by analyzing the function of each gene
- Build the world's largest library of 350,000 mouse gene knockout embryonic stem cell lines

•Create a knockout for each gene

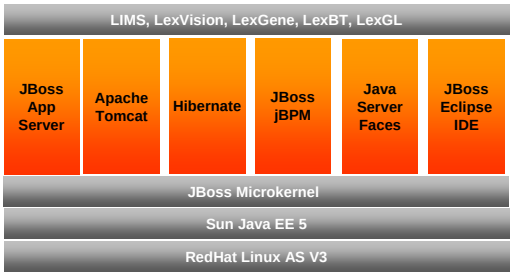
•Accelerate the pace of medical discoveries

Project Objectives

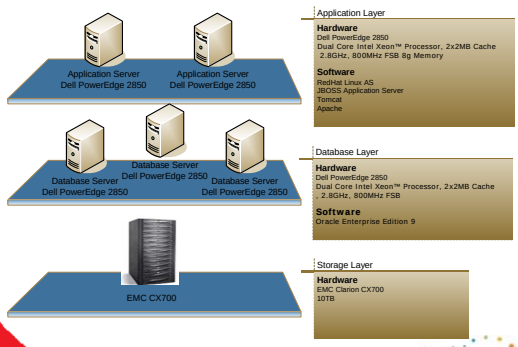
Deliver a suite of applications to TIGM based on Lexicon's existing software platform

- Lab Workflow
- Data Collection
- Inventory Management
- Data Analysis

Software Stack



Mice are mission critical



Legacy Software

- PHP/MySQL animal management software written in 1998
- Ported to Oracle database in 2001

Migration Challenge

- Scientific processes are difficult to model in the early discovery stages
- Legacy application changes would require extensive refactoring

Solution – Rewrite the application

- ERP implementation in 2000 introduced BPM, workflow.
- Evaluated solutions for workflow
- jBPM needs another "layer" necessary to glue together – commercial solutions integrated but very heavy

Development Bottlenecks

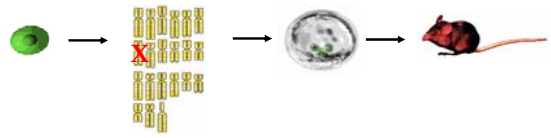
- JSF/JSP
- Database round tripping
- Hibernate lazy initialization (dirty touch)
- Backing Beans
- Model process changes
- Codegen

Addressing Bottlenecks – adding to the stack

- Seam
- Facelets
- jBPM

Application Overview

Gene knockouts mimic potential drug action



Isolation and culture of mouse embryonic stem cells (ES)

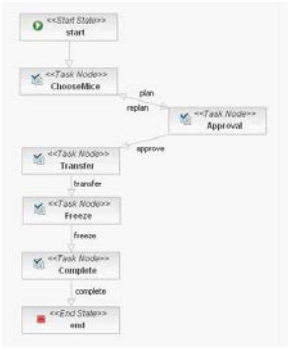
Each ES cell has a specific gene deleted

Re-inject ES cells back into mouse embryos

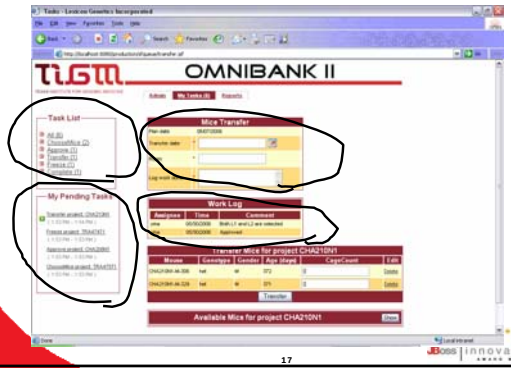
Generate mice missing the gene and analyze the effects (phenotype)

Human version of genes isolated for drug development

Starting with a process



Workspace



jBPM in a Seam Component

```
Stateful
@Name("sfQueueRequest")
@Scope(ScopeType.SESSION)
public class RequestMiceAction implements RequestMiceInf, Serializable {


private void createProcess(Project project) {
        JbpmContext jbpmContext = ManagedJbpmContext.instance();

        ProcessDefinition pd = jbpmContext.getGraphSession()
                                .findLatestProcessDefinition("sfqueue");

        ProcessInstance process = pd.createProcessInstance();
        process.getContextInstance().setVariable("projectId", project.getId());
        process.getContextInstance().setVariable("projectName", project.getName());
        jbpmContext.save(process);
```

Dynamic Task list

```
<ul id="taskList">
<c:forEach var="taskName" items="#{taskBean.taskNameList}">
  <li><s:link  view="/sfqueue/taskList.jsf" value="#{taskName}
     (#{fn:length(allTaskInstanceListForType[taskName])})"
     linkStyle="link">
  <f:param name="taskName" value="#{fn:trim(taskName)}"/>
  </s:link></li>
</c:forEach>
</ul>
```

Dynamic worklog of the processInstance

```
<h:dataTable value="#{taskInstComments}" var="comment">
      <h:column>
      <table width="100%"><tr>
      <td><h:outputText value="#{comment.actorId}"
            styleClass="timeSpan"/></td>
      <td align="right"><h:outputText value="#{comment.time}"
            styleClass="timeSpan">
      <f:convertDateTime locale="en-US" pattern="MM/dd/yyyy"/>
      </h:outputText></td></tr>
      <tr>
      <td colspan="2"><h:outputText value="#{comment.message}" style="font-
            family : Arial, Helvetica, sans-serif;font-size: 9pt;"/></td>
      </tr>
      <tr class="tableRowSeperator"><td colspan="2"></td></tr>
      </table>
      </h:column>
</h:dataTable>
```

Bookmarking

Restful model

```
<s:link  view="/sfqueue/taskList.jsf" styleClass="#{tasksTabStyle}"
      propagation="none"
      value="My Tasks (#{fn:length(allTaskInstanceListForType['All'])})"
      linkStyle="link" />


@Name("allTaskInstanceListForType")
@Scope(EVENT)
@Interceptors(SeamInterceptor.class)
public class TaskInstanceListForType {

      @In(create = true)
      private List<TaskInstance> pooledTaskInstanceList;

      @Unwrap
      public Map<String, List<TaskInstance>> getTaskInstanceList() {
                              .
                              .
                              .
      }
```


UI Messages

Messages & Validation

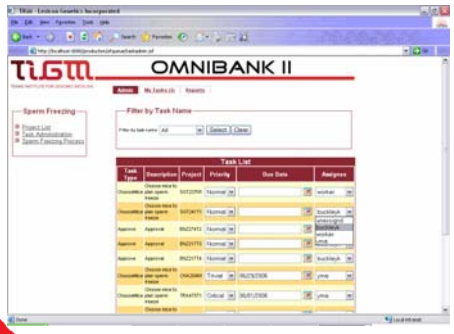
```
      @Valid
      private Date planDate;


      @NotNull
      @Future
      public Date getPlanDate() {
            return planDate;
      }
```

Task list - concurrency

Conversation in a Seam Component

```
@Name("taskDispatcher")
@Scope(ScopeType.SESSION)
public class TaskDispatcher implements Serializable {


private void mapTaskConversation(String taskName,String projectName){
        Conversation conversation = Conversation.instance();
        conversation.setTimeout(72000000);
        conversation
          .setViewId("/sfqueue/"+taskName.substring(0,1)
          .toLowerCase()+taskName.substring(1)+".xhtml");
        conversation.setDescription(taskName+" project: "+projectName);
        conversation.begin();
        taskConversationMap.put(taskId,conversation.getId());
}
```

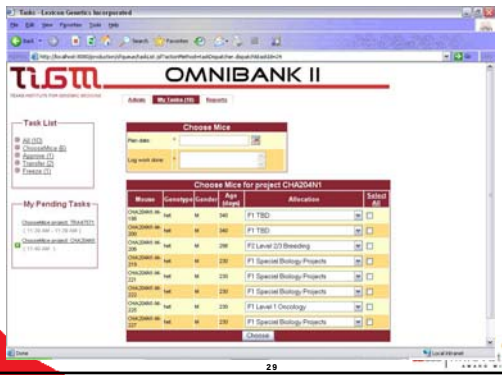

Task Assignment



jBPM Roles

```
-- CONFIG FILE --
<policy>
<application-policy name="sfqueue">
 <authentication>
<login-module code="org.jboss.security.auth.spi.LdapLoginModule" flag="required">
  <module-option name="principalDNPrefix">uid=</module-option>
 <module-option name="principalDNSuffix">,ou=People,dc=lexgen,dc=com,ou=system</module-option>
 <module-option name="rolesCtxDN">ou=sfqueue,ou=Apps,dc=lexgen,dc=com,ou=system</module-option>
  <module-option name="uidAttributeID">uniqueMember</module-option>
  <module-option name="matchOnUserDN">true</module-option>
  <module-option name="roleAttributeID">cn</module-option>
  </login-module>
  </authentication>
  </application-policy>
  </policy>

-- JPDL TASK --
        <task-node name="Approval">
                <task name="Approve" description="Approval">
                        <assignment pooled-actors="mpmanager" />
                </task>
          <transition name="approve" to="Transfer"></transition>
          <transition name="replan" to="ChooseMice"></transition>
        </task-node>
```


Task List Detail



Dynamic Search – UI widget

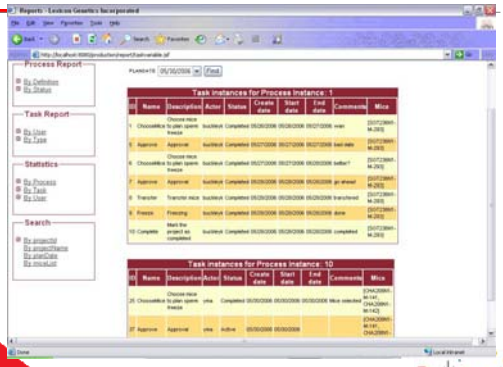
```
<ul>
   <li>
   <c:forEach items="#{reportFilter.allVariables}"
      var="jbpmVar">
   <s:link  view="/report/taskvariable.xhtml"
      value="By #{jbpmVar}" linkStyle="link" >
    <f:param name="variable"
      value="#{jbpmVar}"/>
     </s:link>
   <br/>
   </c:forEach>
   </li>
 </ul>
```

Dynamic Search Seam Component

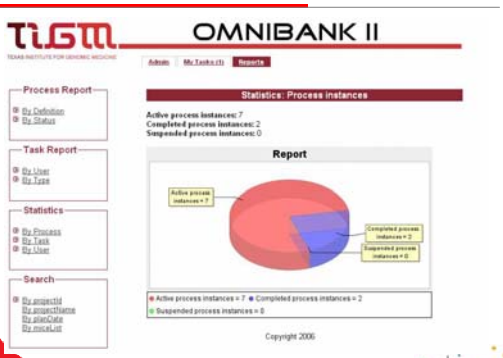
```
@Stateless
@Scope(EVENT)
@Name("reportFilter")
@Interceptors(SeamInterceptor.class)
public class ReportFilter implements ReportFilterInf, Serializable {

...

@In (create=true)
private JbpmContext jbpmContext;

public List<String> getAllVariables() {
Session session = jbpmContext.getSession();
Query query = session.getNamedQuery("ReportFilterBean.getDistinctVariables");
List<String> variables = new ArrayList<String>(query.list());
return variables;
}

<query name="ReportFilterBean.getDistinctVariables">
<![CDATA[
    select distinct vi.name
    from org.jbpm.context.exe.VariableInstance as vi
  ]]>
</query>
```


Reports

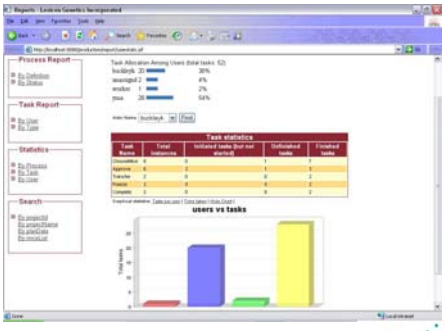
Reporting



Reports



Reports



Gene knockouts mimic potential drug action

Drug Target	Drug Name (utility)	2001 Sales ($MM)	Mouse phenotype
Unknown Target	Depakote (epilepsy) Neurontin	$908.0 $1751.0	Not available
GABA Receptor	Ambien (insomnia) Stilnox	$902.0 $704.0	Hyperactive, hyper-responsive
Histamine H2 Receptor	Gaster (gastroesophageal reflux disease) Zantac	$896.0 $727.0	Induction of gastric acid secretion by histamine or gastrin is completely abolished
Mu Opioid Receptor	Duragesic (pain) Ultram	$875.0 $601.0	Increased sensitivity to pain
Serotonin 5HT3 Receptor	Zofran (nausea/vomiting)	$865.0	Not available
Prostanoid Receptors	Xalatan (glaucoma)	$818.0	Intraocular pressure not tested in prostanoid receptor knockouts
TNF Alpha	Enbrel (arthritis) Remicade	$762.0 $721.0	Decreased contact hypersensitivity, IgG and IgE

Sales Info Source: *PharmaLive (Engel Publishing)

Lessons Learned

- Training
- Comment for future fixes - prepare for breakage
- Separate entities from process
- Expect to refactor
- Get to know jira.jboss.com

37

Lesson #1 Bring in the experts!

That's not a framework!
This is a framework!

38

Lesson #2 – It's not always perfect

```
//check if the conversation has expired, if it is, create a new conversation
//otherwise return error message
if(Manager.instance().getConversationIdEntryMap().get(conversationId)==null)
{
                mapTaskConversation(taskName,projectName);
}else {
        //why it does not work?
        //return Manager.instance().getConversationIdEntryMap().get(conversationId).select();
        facesMessages.add("You are currently working on this task.");
        return "tasks";
}
```

39

Lesson #3 – Separate entities

```
@Entity
@Table(name = "MOUSE_TABLE")
@Inheritance(strategy=InheritanceType.JOINED)
public class Mouse implements java.io.Serializable {
```

```
@Stateful
@Name("chooseMice")
@Scope(ScopeType.CONVERSATION)
@Conversational(ifNotBegunOutcome = "chooseMiceTasks")
public class ChooseMiceAction extends SFBaseAction implements ChooseMiceInf, Serializable {

…

        taskInstance.addComment(new Comment(actor.getId(), getComment()));
        taskInstance.setVariable("planDate", planDate);
        taskInstance.setVariable("miceList", mice);
```

40

Thanks!

www.lexicon-genetics.com
www.tigm.org

41