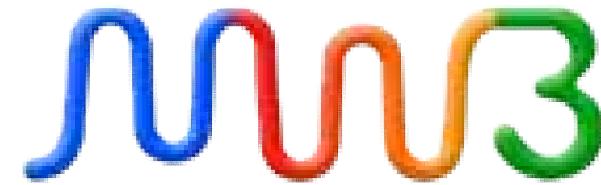


웹기반 AJAX개발을 위한 프레임워크 – 메타웍스3

유엔진 오픈소스 BPM 프로젝트
장진영
jyjang@uengine.org



Concurrent Web-based Applications :

The Asana homepage features a large blue "asana:" logo at the top. Below it, there's a banner with a smartphone and a laptop. The main content area is divided into several sections:

- Multiple workspaces:** Shows how to invite up to 30 members to a workspace.
- Activity feed for every task:** Displays a feed of tasks from team members.
- People views:** Shows a list of team members with their profiles.
- Real-time: no browser refreshing:** Shows a "no refresh" icon.
- Tag Items across projects:** Shows a list of tags and projects.
- Search everything:** Shows a search bar with a magnifying glass icon.
- Daily reminder emails:** Shows a reminder for "Hi Marty".
- SSL all the time:** Shows a lock icon.
- Drag-and-drop:** Shows a list of tasks with a reorder icon.

The Cloud9 IDE interface includes a large blue cloud icon with the number "9" in the center. The main window contains:

- Code Editor:** Displays a file named "a.js" with some JavaScript code.
- Terminal:** Shows a log of ticks from 1001 to 1018.
- Variables:** A table showing variables like "no" (number), "fn" (object), and "repeat" (number).

The Facebook news feed interface shows:

- A profile picture of Jane Smith with the text "View My Profile".
- A post from "MGM" with a link to "https://asana.com/awesome-co".
- A post from "MGM GRAND" with a movie poster.
- A post from "MGM Mondo del Vino" with a wine bottle icon.
- Navigation links: News Feed, Messages (5), Events, Photos, Friends (3), Applications, Games, Trips.
- Text at the bottom: "See More Results for mgm" and "Displaying Top 4 Results".

You are the mother, How about your Object ?

Mother cares All the things?



Do that by himself ?



Introducing Metaworks3 –

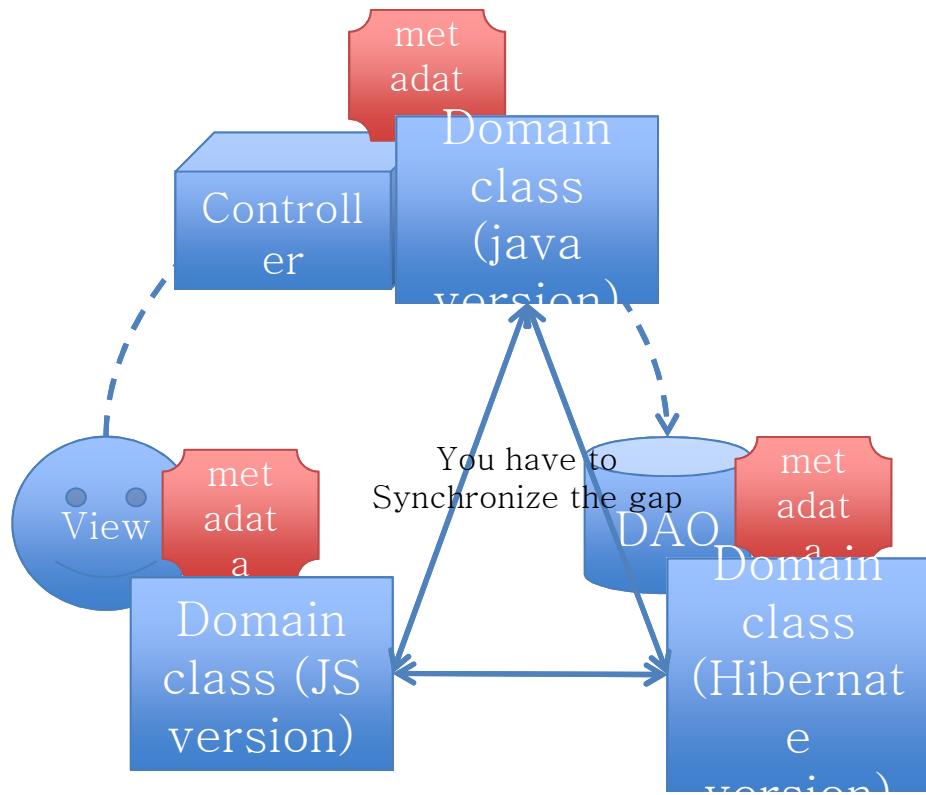


- Metaworks is ‘metadata-oriented framework’
- Concurrent Web-based (and mobile) Applications are degrading the Object’s own property – Cohesion of metadata.
- For example, you need to keep the properties’ type as same value in DHTML scripts, Server Object, Data Object(Hibernate), and the DDL script as well.
- Centralizing metadata and behaviors – it’s Object-Oriented Manner itself!
- Metaworks automatically synchronizes(or hides the operation) ...
 - Stubs for calling Javascript to the server function (using DWR)
 - DAO object and relational query (not completely, SQL is still good)
 - Keeps intuitive and Sophisticated Programming Model... **Don't be a dog!**
- **By managing the metadata**

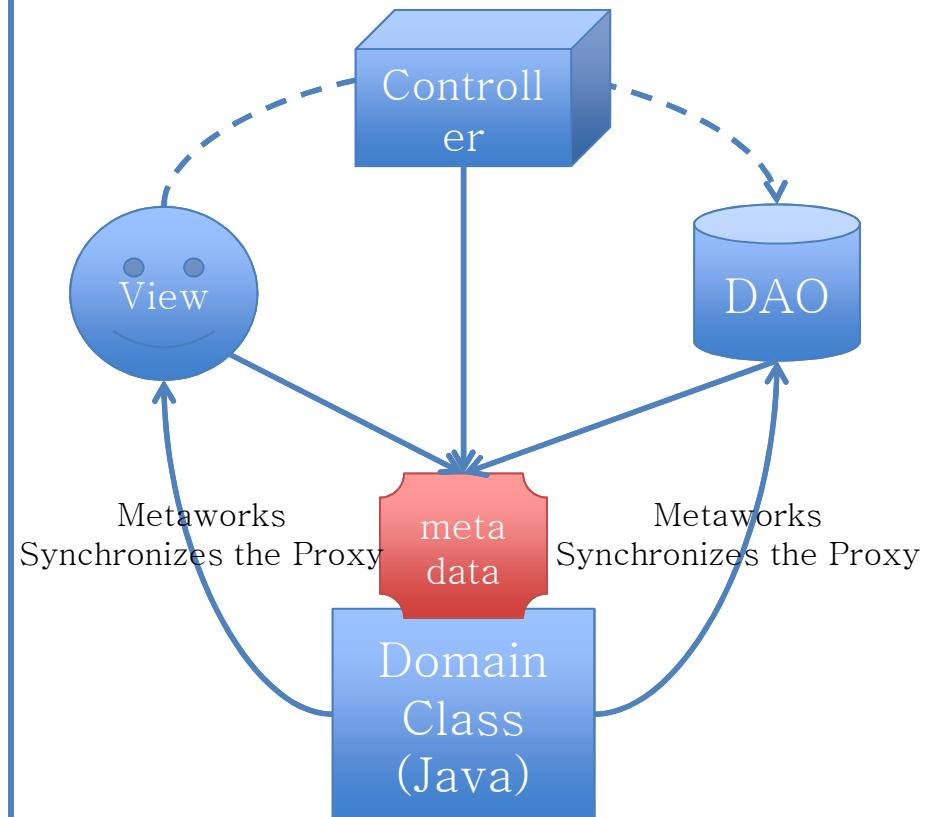
문제:

- 흩어진 메타데이터

General Approach: Spring MVC, JSON, jQuery, Hibernate



Using Metaworks3:



No more words, Seeing is believing:

#Domain Class (Posting.java)

```
@Table(name = "fb_posting")
public class Posting extends MetaworksObject<IPosting>
implements IPosting{

    String document;
    @Id
    public String getDocument() {
        return document;
    }
    public void setDocument(String document) {
        this.document = document;
    }
    @ServiceMethod
    public void share() throws Exception{
        createDatabaseMe();
    }
}
```

#Application (application.html)

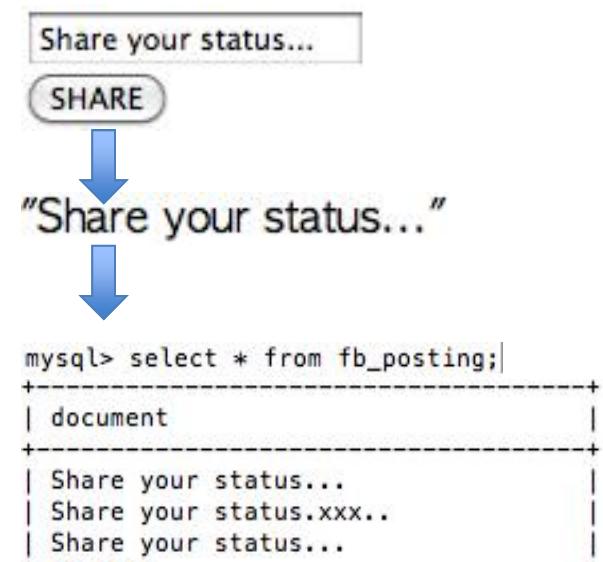
```
<script>
    mw3.setContextWhen(mw3.WHEN_EDIT);
    mw3.locateObject(
        { document:"",
          __className:
        "org.metaworks.example.Posting"
    },
    null,
    'newEntry'
);
</script>
...
<div id='newEntry'></div>
```

#Presentation (Posting.ejs)

```
<%=fields.document.here() %>

<%
    if(mw3.when == mw3.WHEN_EDIT){
%>
    <%=methods.share.here()%>
<%
    }
%>
```

#Result



FaceBook example

Steps:

1. Recognizing what is (was) ‘Object’ in your application
 2. Modeling the Objects
 3. Face making and Context-awareing for the Objects
 4. Creating Application with highly-abstracted programming Model – almost natural language!
- Humanity-Recovery !

Step 1.

Recognizing what is (was)
'Object' in your application



1. Recognizing Objects in facebook

The screenshot shows a Facebook group page for "유엔진 일일 전략 및 회고". Several objects are highlighted with colored boxes:

- Person object**: A blue box highlights the profile picture of a user named 장진영.
- Posting object**: A red box highlights the post by koramco titled "글 쓰기".
- Doc object**: A green box highlights the post by Hojun-Harry Sung titled "Hojun-Harry Sung".
- Group object**: A yellow box highlights the group "uEngine.org".
- Group object**: Another yellow box highlights the group "유엔진 CXOs".
- Group object**: A third yellow box highlights the group "유엔진 일일 전략 및 회고".
- Group object**: A fourth yellow box highlights the group "Social BPM and KM - Pro...".
- Group object**: A fifth yellow box highlights the group "도크스포럼".
- Group object**: A sixth yellow box highlights the group "Talk IT (Intelligenc...).
- Group object**: A seventh yellow box highlights the group "ICE 학교".
- Group object**: An eighth yellow box highlights the group "HiOSS Media Group".
- Group object**: A ninth yellow box highlights the group "유엔진-진잔해주세요".
- Group object**: A tenth yellow box highlights the group "클라우드 컴퓨팅 구...".
- Group object**: A eleventh yellow box highlights the group "uEngineSolutions".
- Group object**: A twelfth yellow box highlights the group "CSS 프로세스코디".
- Group object**: A thirteenth yellow box highlights the group "데 보기".
- Group object**: A fourteenth yellow box highlights the group "체험 가능한 친구".
- Group object**: A fifteenth yellow box highlights the group "조신원".
- Group object**: A sixteenth yellow box highlights the group "1. Smart e-Learning App".
- Group object**: A green box highlights the "멤비(28)" section.
- Group object**: A green box highlights the "문서(21)" section.
- Group object**: A green box highlights the "스폰서" section.
- Group object**: A green box highlights the "광고 만들기" section.
- Group object**: A green box highlights the "SKTsmartsupporters" section.
- Group object**: A green box highlights the "유엔진 일일 전략 및 회고" section.

□ Person object

□ Posting object

□ Doc object

□ Group object
www.metaworks3.org

1. Recognizing Objects in facebook

1.1. Person objects in different ‘Context’

The screenshot shows a Facebook group page for "유엔진 일일 전략 및 회고" (Yeonjin Daily Strategy and Review). The left sidebar shows various groups like "uEngine.org", "유엔진CXOs", and "Social BPM and KM - Pro...". The main content area displays posts from members like "류승호" and "Hojun-Harry Sung". A legend at the bottom indicates three types of person object placement:

- Where: NORMAL**: Represented by a blue-bordered square around the user profile picture of "장진영" in the sidebar.
- Where: MINIMISED**: Represented by a blue-bordered square around the user profile picture of "류승호" in the post list.
- Where: MEDIUM**: Represented by a blue-bordered square around the user profile picture of "Hojun-Harry Sung" in the post list.



Where: NORMAL



Where: MINIMISED



Where: MEDIUM
www.metaworks3.org

1. Recognizing Objects in facebook

1.2. Posting objects in different ‘Context’

The screenshot shows a Facebook group page titled "유엔진 일일 전략 및 회고" (UNJIN Daily Strategy and Review). The left sidebar shows the user's profile picture and name, along with links for news feed, messages, events, friends, groups, and pages. The main content area displays two posts:

Post 1 (NORMAL EDIT Context): Posted by "류승호" on 2011-10-06. The post content is:
koranico 뷰조 보수
1. 이종글리 방지 퍼치 (formValidation 루직도 같이 채크)
2. 메일필터 이전업무 담당자 찾는로직 수정
3. 플로우 차트, 업무이력 택등의 특정사용자에게만 권한을 주도록 적용
=> 100% Hard Coding
좋아요 · 댓글 달기 · 게시를 받아보기 · 12분 전

Post 2 (NORMAL VIEW Context): Posted by "Hojun-Harry Sung" on 2011-10-06. The post content is:
2011-10-06
1.스마트워크 플레오 스 구축시나리오 협문 번역
-지인관리, 통합커뮤니케이션 및 통합검색 부분
2. OSS협회 팬넬이미지 송부 & Danang 대응하기
3.웹사이트 헬스체크 – Enterprise Solution Fair 2011, 8월 22일 컨퍼런스 과제보고서
좋아요 · 댓글 달기 · 게시를 받아보기 · 44분 전

Post 3 (MINIMISED Context): Posted by "Hojun-Harry Sung" on 4. 지난주 전시회 방문객 – 삼정터빈 큐경문팀 장과 통화
- 아직까지 전체 프로세스를 다 정리하지 못했음.
- 전산부서 외에, 전사 부서 미팅을 통해 프로세스 정리하여 10월 중에 TO-BE 모습을 요청하도록 하겠음
- 핸디 BPM 도입을 하였지만, 잘 안되었음
- 10월 18일 Enterprise Solution Fair 2011 컨퍼런스에 참석예정

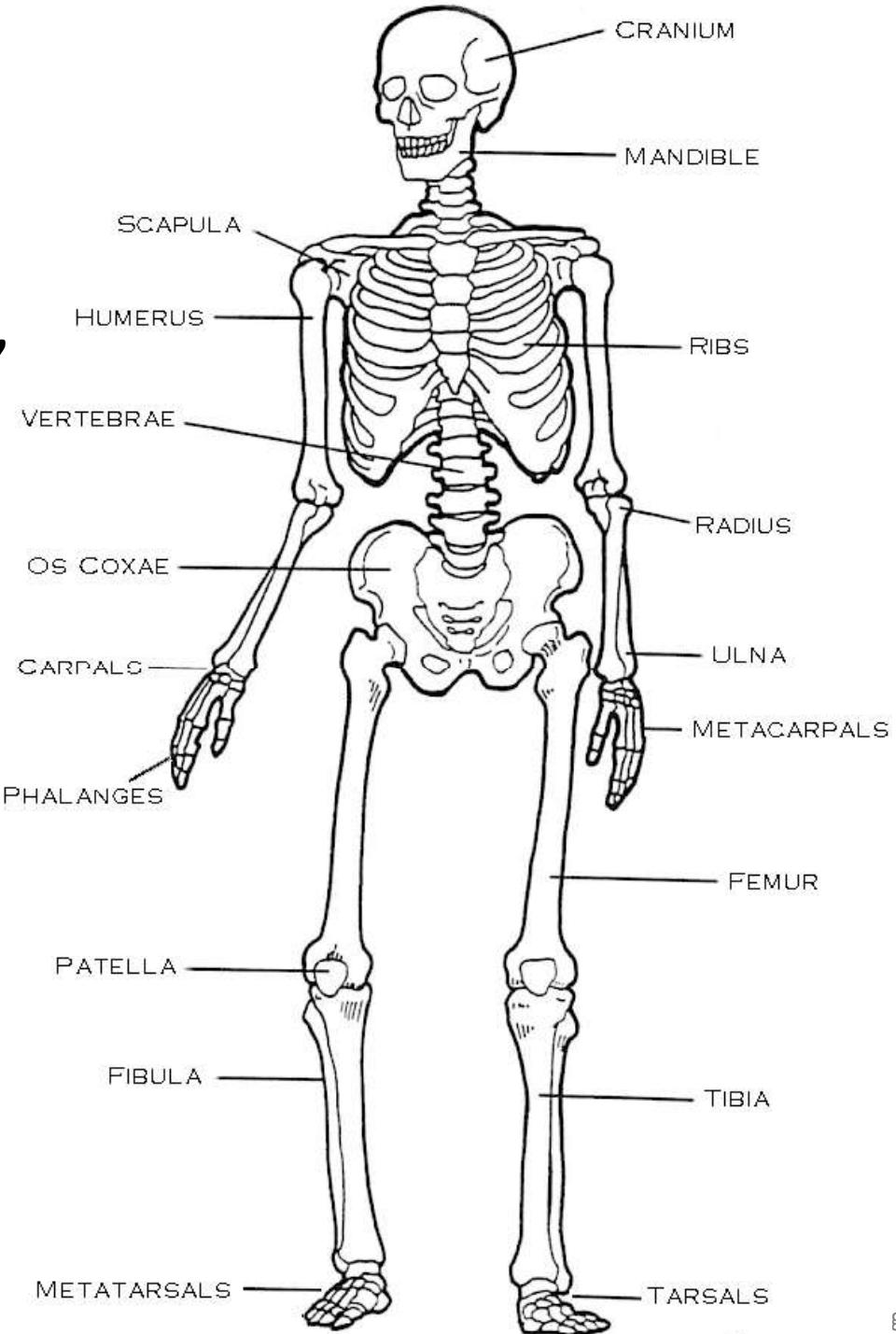
Where: NORMAL
When: EDIT

Where: NORMAL
When: VIEW

Where: MINIMISED
When: EDIT

Step 2.

Modeling the ‘Cohesive’ Objects



2. Modeling Objects - Posting object Design –

2.1. create interface

```
import org.metaworks.dao.IDAO;

public interface IPosting extends IDAO{

    public Person getWriter(); // 작성자
    public void setWriter(Person writer);

    public String getDocument(); // 글
    public void setDocument(String document);

    public boolean isLikeIt(); // 좋아요 여부
    public void setLikeIt(boolean likeIt);

}
```

2. Modeling Objects - Posting object Design –

2.2. add metadata

```
import org.metaworks.dao.IDAO;

@Table(name="Posting")
public interface IPosting extends IDAO{

    public Person getWriter();
    public void setWriter(Person writer);

    @Id
    public String getDocument();
    public void setDocument(String document);

    public boolean isLikelt();
    public void setLikelt(boolean likelt);

    @ServiceMethod
    public void post() throws Exception;

    @ServiceMethod
    public void like() throws Exception;
}
```

2. Modeling Objects – Posting object

2.3. create implementation object

```
public class Posting extends Database<IPosting> implements IPosting{  
  
    Person writer;  
    public Person getWriter() {  
        return writer;  
    }  
    public void setWriter(Person writer) {  
        this.writer = writer;  
    }  
  
    String document;  
    public String getDocument() {  
        return document;  
    }  
    public void setDocument(String document) {  
        this.document = document;  
    }  
  
    boolean likelt;  
    public boolean isLikelt() {  
        return likelt;  
    }  
    public void setLikelt(boolean likelt) {  
        this.likelt = likelt;  
    }  
}
```

2. Modeling Objects - Posting object

2.4. add behaviors

```
public class Posting extends Database<IPosting> implements IPosting{

    Person writer;
    public Person getWriter() {
        return writer;
    }
    public void setWriter(Person writer) {
        this.writer = writer;
    }

    String document;
    public String getDocument() {
        return document;
    }
    public void setDocument(String document) {
        this.document = document;
    }

    boolean likelt;
    public boolean isLikelt() {
        return likelt;
    }
    public void setLikelt(boolean likelt) {
        this.likelt = likelt;
    }

    public void post() throws Exception{
        createDatabaseMe();
    }
    public void like() throws Exception{
        databaseMe().setLikelt(true); //will automatically synchronize the value
    }
}
```

A close-up photograph of a woman's face. She has dark hair and is wearing vibrant, multi-colored eyeshadow. A decorative headband with a pattern of small, colorful sequins is visible on her forehead. Her lips are painted with a glossy, reddish-pink shade. The lighting is soft, highlighting the texture of her skin and the reflective surfaces of the makeup and headband.

Step 3

Face making and Context-awareing for
the Objects

3. Face Making

3.1. creating 'Face' – Posting.ejs

```
<table>

    <tr>
        <td><%=fields.writer.here() %></td>
        <td><%=fields.document.here() %>

        <%
            if(value.likeIt){
        %>
        You like this
        <% }else{%
        <a onclick=<%=methods.like.caller()%>>좋아요</a>
        <%}>
        </td>

    </tr>
</table>
```

3. Face Making

3.1. considering the Contexts – WHEN_EDIT

```
<%
    if(mw3.when == mw3.WHEN_EDIT){
        value.document = 'Share your status...';
    }
%>
<table>
    <tr>
        <td><%=fields.writer.here()%></td>
        <td><%=fields.document.here()%>
<%
    if(mw3.when == mw3.WHEN_EDIT){
%>
        <%=methods.post.here()%>
<%
    }
    if(value.likeIt){
%>
        You like this
<% }else{%
        <%=methods.like.here()%>
<%}%
<%
    if(mw3.when == mw3.WHEN_EDIT){
%>
        <input type=button value="save">
<%
    }else{
%>
        <input type=button onclick=<%=editFunction%> value="edit" a>
<%
    }
%>
        </td>
    </tr>
</table>
```

Next, Do 1~3 for Person object – Person.java & Person.ejs

#Person.java

```
public interface Person extends IDAO{

    @org.metaworks.Metadata(isKey = true)
    public String getName();
    public void setName(String name);

    public int getAge();
    public void setAge(int age);

    @org.metaworks.Metadata(face="faces/image.ejs")
    public String getPortraitURL();
    public void setPortraitURL(String portraitURL);

    public boolean isMyFried();
    public void setMyFried(boolean isMyFried);
}
```

#Person.ejs

```
<%if (!value){%>
Writer is not specified
<%}else{%
<br>
<b><%=value.name%></b>
<%}>
```

Step 4

Creating Application with
highly-abstracted
programming Model –

Write a Novel or Poet, not
the alien language



4. Now, create the Application

```
<script type="text/javascript">

    lastEntryId = new MetaworksObject(
        {
            document:"",
            writer: loginUser,
            __className: "org.metaworks.example.Posting"
        },
        'newEntry'

    );
</script>

<div id="newEntry"></div>
```

Result

http://localhost:8080/mw3/index.html

jinyoung jang You like this edit

 "Share your status..."
jinyoung jang You like this edit

 "Share your status...dfsdfsdf"
jinyoung jang You like this edit

 "Share your status..."
jinyoung jang You like this edit

 "Share your status..."
jinyoung jang You like this edit

 "Share your status..."
jinyoung jang You like this edit

 "Share your statuddfsdfsdfsds..."
jinyoung jang You like this edit

 "Share your stdfsfdsstatus..."
jinyoung jang You like this edit

 "Share your statusssdfsdfsd..."
jinyoung jang like edit

 "Share your statussssdad..."
jinyoung jang You like this edit

 Share your statusxxxx
joe bloggs SHARE like save

[add to friend]

EJS TEMPLATE DEBUG INFO :
DWR DEBUG INFO :

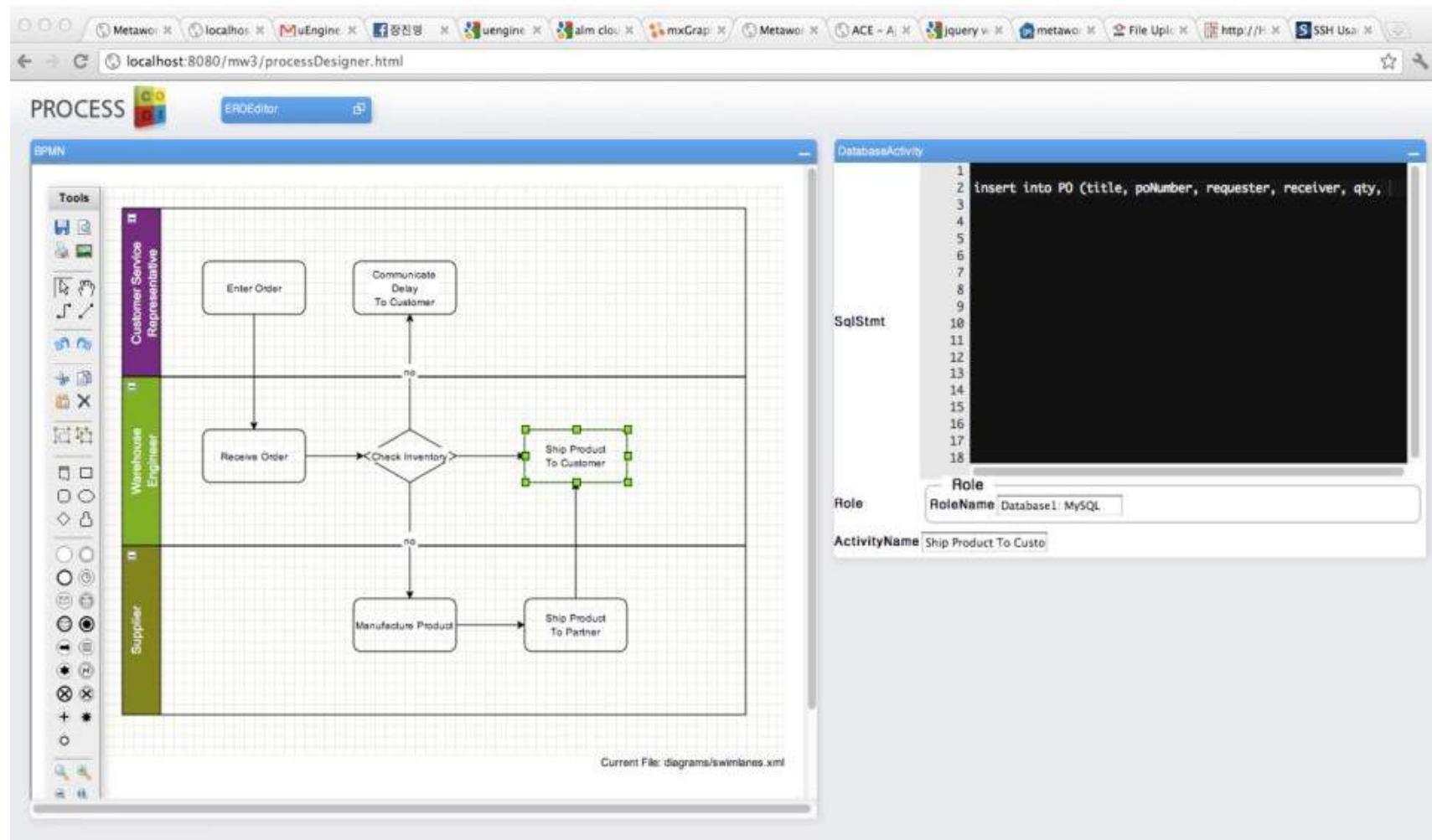
http://localhost:8080/mw3/index.html

Applications

The screenshot shows a web browser window with multiple tabs open. The main content area displays a 'FeedbackPanel' section with three comments from a user named '장진영'. The first comment says, "Hi, I'm the administrator of metaworks3 project! Enjoy very simple model-driven AJAX applications with Metaworks3!" The second comment says, "You may leave any comment to all the pages after logging in facebook account." The third comment says, "Have good time!" Below the comments is a 'Login with Facebook' button. To the left of the feedback panel, there is a 'Sample Application' section featuring a video player embedded in a code editor interface. The video player shows a progress bar at 39:16 and a 'vimeo' logo. The overall layout includes a navigation menu on the left and several search results at the top.

www.metaworks3.org website Social CMS

Applications



ProcessCodi Cloud IDE

Applications

The screenshot displays the ProcessCodi Enterprise 2.0 Platform interface. At the top, there is a navigation bar with tabs for 'Process' and 'Home'. Below the navigation bar, the main area is divided into several sections:

- Left Sidebar:** Contains sections for 'Personal Center' (내가 할 일, 전체, 요청한 건), 'Task Center' (전체 중심), 'Job Center' (조직 중심), and 'Ability Center' (기술 중심). It also includes a 'Contact' section with a search bar and a list of contacts.
- Central Task List:** A 'Personal Center - 전체 검색' (Personal Center - Full Search) window shows a list of tasks. One task is highlighted in red:

참여자	상태	제목	시작일	중요
[User Icons]	[Status Icon]	검색이 잘 되지 않는 것 2011년 8월 4일... 아니면...	2011년 8월 4일	[Star Icon]
[User Icons]	[Status Icon]	검색 때문에 문제가 발생하는 경우는 생기지 않겠군요...	2011년 8월 4일	[Star Icon]
[User Icons]	COMPLETED	[Briefcase Icon] [1월성과]재무_하나_90	2011년 7월 28일	[Star Icon]
[User Icons]		gksrmf	2011년 7월 27일	[Star Icon]
- Right Side:** A 'Human work(Initiator : Jinyoung Jang)' window is open, showing two messages from Jinyoung Jang. The first message is 'Human work(Initiator : Jinyoung Jang)' dated 2011-10-07 (COMPLETED). The second message is another message from Jinyoung Jang. There are buttons for '완료처리' (Complete) and '저장' (Save).

ProcessCodi Enterprise 2.0 Platform

Conclusion

- No duplicated metadata mapping
 - e.g. JSON object mapping in Spring Controller and DAO mapping for Hibernate.
- No errors from duplicated metadata source.
- Cohesive User Interface management by EJS-object mapping.
 - Intuitive Programming Model – model and metadata-driven, not implementation-driven.

Only thing you have to do is... just forgetting the old manners!

All the Source Code available here:

www.metaworks3.org

it's very early stage now! You are welcomed to participate this Open Source Project.