

MVC et applications Web

Servlets/JSP et modèle MVC

Model2, Struts

Pierre-Yves Gibello - pierreyves.gibello@experlog.com

Mise à jour : 30 Septembre 2004

Ce document est couvert par la licence Creative Commons Attribution-ShareAlike.

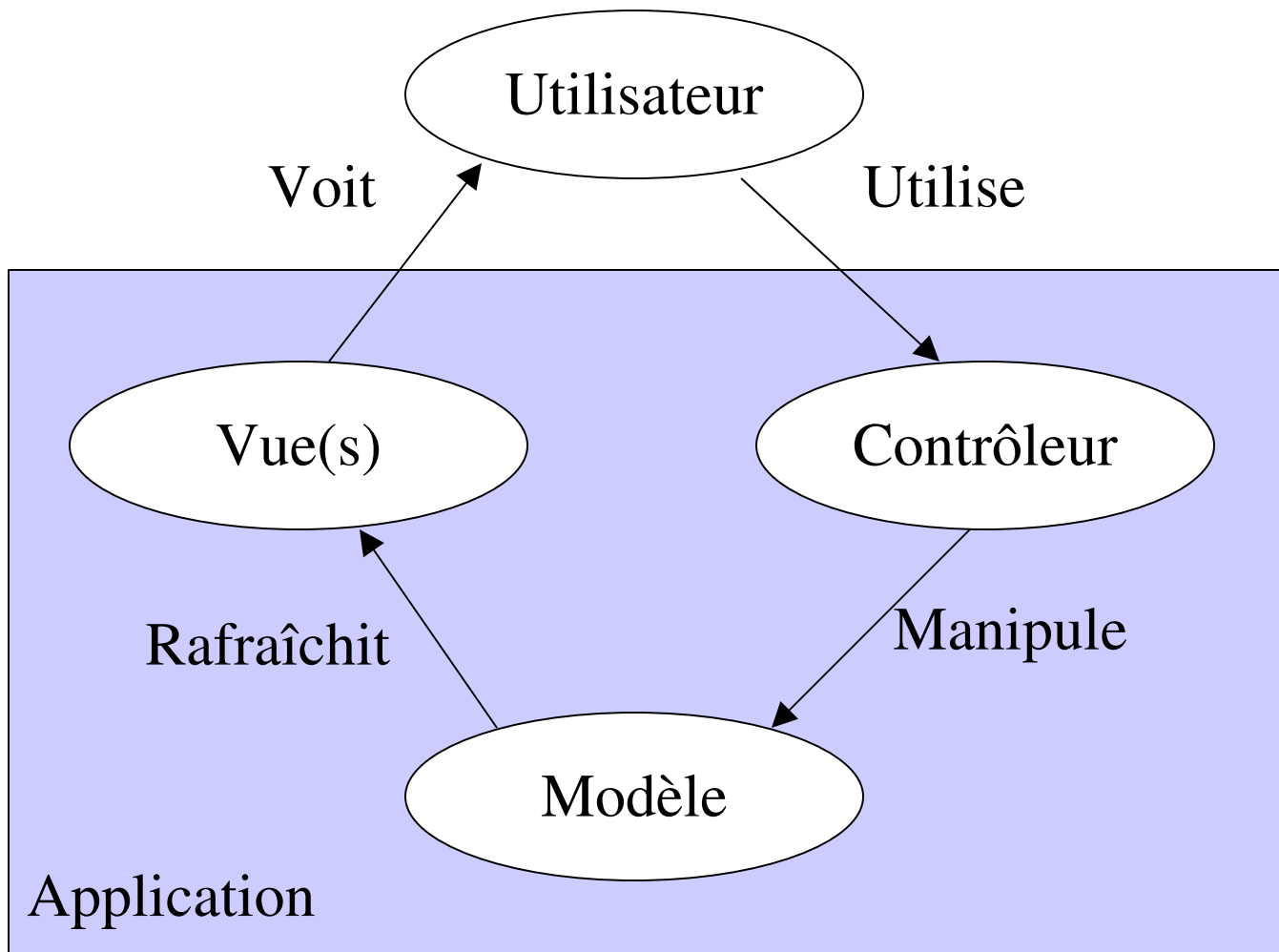
This work is licensed under the Creative Commons Attribution-ShareAlike License.

Modèle MVC

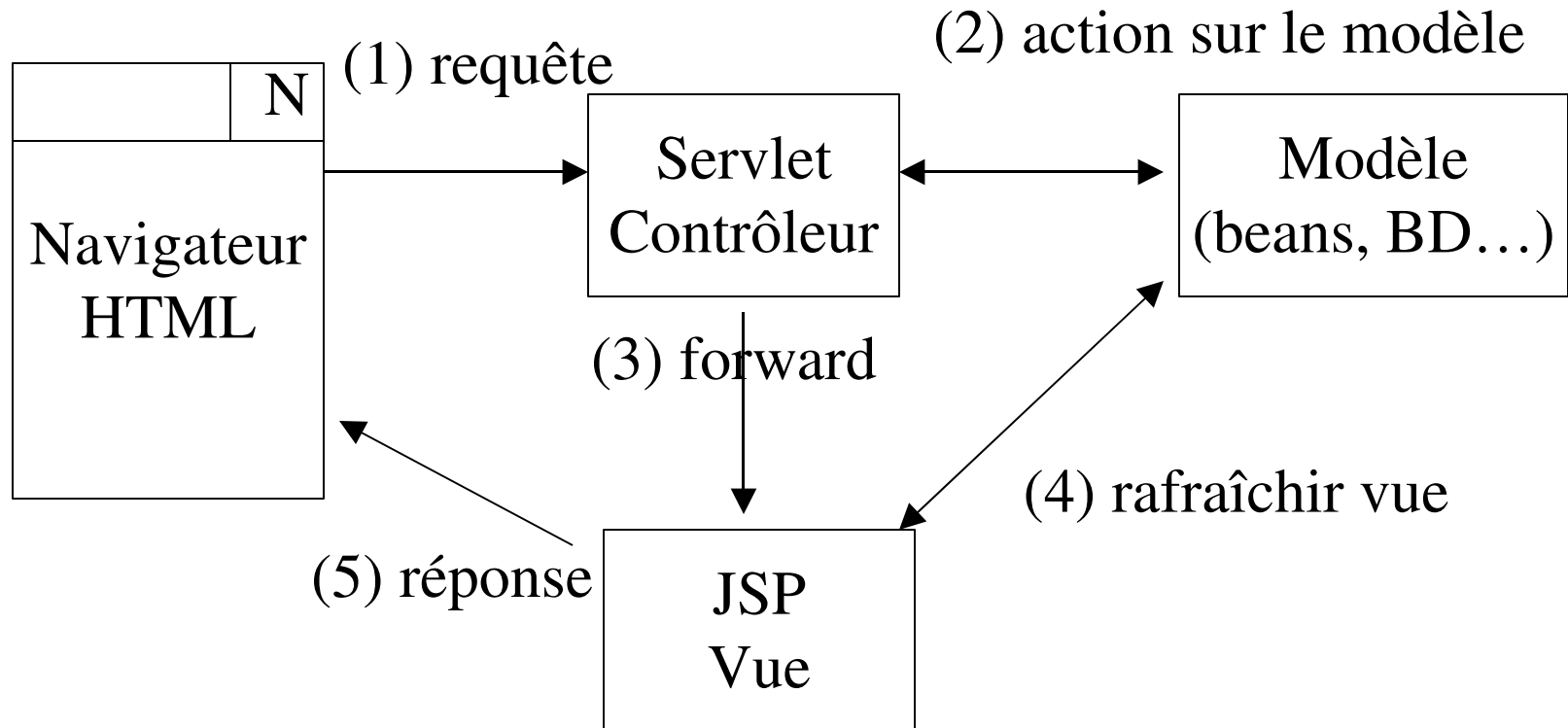
- Model / View / Controller
 - Modèle : l 'application métier, les données... ce qui est manipulé par l 'application.
 - Vue : présentation de l 'état du modèle (généralement à un utilisateur)
 - Contrôleur : interface d 'action sur le modèle (par exemple, pour les mises à jour)
- L 'interface utilisateur, composée de vues, interagit avec le contrôleur.

Le Pattern MVC

Une « boucle de rétroaction »...



Servlets/JSP « Model2 »



Exemple : Servlet Contrôleur

```
public class ControllerServlet extends HttpServlet
{

    public void doGet (HttpServletRequest req,
    HttpServletResponse res)
    throws ServletException, IOException {

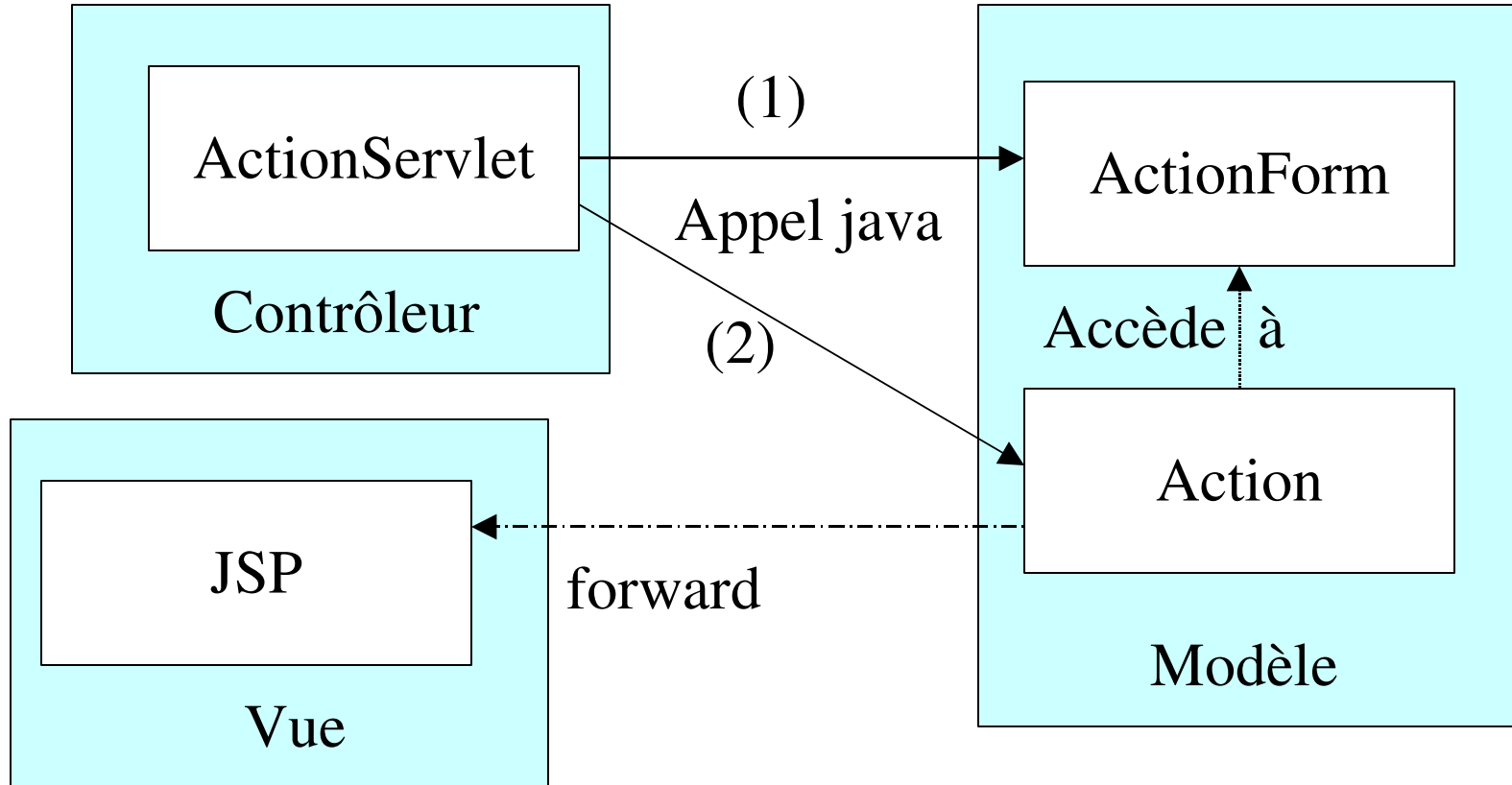
        String action = req.getParameter("action");

        if("consulter".equals(action)) {
            String nom = req.getParameter("nom");
            if(nom == null) forward("error.jsp", req,
res);
            else {
                String tel = annuaire.chercherContact(nom);
                req.setAttribute("tel", tel);
                forward("contact.jsp", req, res);
            }
        }
    }
}
```

Struts

- Servlet Contrôleur générique
 - org.apache.struts.action.ActionServlet
 - servlet-mapping sur « *.do »
- Interaction avec le modèle
 - ActionForm et Action à implémenter
- WEB-INF/struts-config.xml
 - Association URL(.do)/ActionForm/Action/View JSP
- Taglibs utilitaires
 - struts-html, struts-logic, struts-bean

Struts : architecture



ActionForm

- Java Bean
- getter et setter pour chaque champ de formulaire
- gestion d 'erreur (validate() pour validation de formulaire)
- associé à une URL « .do », une Action et une vue JSP dans struts-config.xml

Action

- Classe qui interagit avec l'objet métier et l'ActionForm
 - Etend org.apache.struts.action.Action
 - public ActionForward perform(ActionMapping mapping, ActionForm form, HttpServletRequest request, HttpServletResponse response)
- Lien avec l'ActionForm, l'URL et les vues (forward) dans struts-config.xml
 - ActionMapping pour accéder à l'information, et pour le forward(vue)

struts-config.xml

```
<form-beans>
```

```
  <form-bean name=«myForm »  
    type=« mypackage.MyForm »/>
```

Implém. de
1 'ActionForm

```
</form-beans>
```

```
<!-- association URL.do/ActionForm/Action/vue -->
```

```
<action-mappings>
```

```
  <action path=«/myform »  
    type=«mypackage.MyAction »  
    name=«myForm »  
    input=«/saisie.jsp »  
    scope=«request »>
```

Signifie «/myform.do»

Implém. de 1 'Action

JSP si erreur saisie

```
    <forward name="success" path="/ok.jsp"/>
```

```
    <forward name="failure" path="/erreur.jsp"/>
```

```
  </action>
```

```
</action-mappings>
```

Formulaire de saisie

```
<!-- saisie.jsp -->
```

```
<%@ taglib uri=« /WEB-INF/struts-html.tld » prefix=« html »>
```

```
<!-- traitements erreur (attribut «input» de «action-mapping») -->
```

```
<html:errors/>
```

```
<!-- formulaire utilisant la taglib struts-html -->
```

```
<!-- «myform»(.do) = attribut «path» de «action-mapping» -->
```

```
<html:form action=« myform.do »>
```

```
  Nom : <html:text property=« nom »/>
```

```
  <html:submit />
```

```
</html:form>
```

ActionForm

```
package mypackage;  
  
public final class MyForm extends ActionForm {  
  
    private String nom_ = null;  
  
    public void setNom(String n) { nom_ = n; }  
    public String getNom() { return nom_; }  
}
```

Action

```
package mypackage;
```

```
public final class MyAction extends Action {  
    public ActionForward perform(  
        ActionMapping mapping, ActionForm form,  
        HttpServletRequest request, HttpServletResponse response) {  
  
        MyForm f = (MyForm)form;  
        String nom = f.getNom();  
        // ... traitements ...  
        request.setAttribute(« leNom », nom);  
        return mapping.findForward(« success »);  
    }  
}
```

Vue JSP

```
<!-- ok.jsp -->  
<%@ taglib uri=« /WEB-INF/struts-logic.tld » prefix=«logic»>  
  
<logic:present name=« leNom » scope=« request »>  
  Bonjour <logic:write name=« leNom » scope=« request »/>  
</logic:present>
```

Struts : installer

- Appli d 'exemple : struts-blank.war
 - Déployer dans webapps
 - Modifier la configuration, et partir de l 'exemple
- Documentation : struts-documentation.war
 - Déployer dans webapps