Conception d'applications web avec WAMM 1.0

Modélisation et implantation

Nathanaël Cottin
www.ncottin.net

version 0.1.3 – 2007

Avant propos : prérequis

- **Modélisation :**
  - Diagrammes UML : cas d'utilisation, états, classes, séquences

- **Implantation :**
  - JSP
  - JavaBeans
  - Connaissances (X)HTML
  - Notions de CSS (annexes)

Qu'est-ce que WAMM ?

WAMM (« Web Applications Modeling Method ») permet de modéliser et mettre en œuvre des applications web indépendamment des technologies employées selon les recommandations du PIM de l'OMG

Il s'agit d'un ensemble de stéréotypes, de nouvelles notations et d'intégration de modèles

WAMM est basée sur le standard UML 2

WAMM s'inspire des travaux menés par Pascal Roques et relatifs à l'adaptation d'UML aux besoins spécifiques des architectures web

Étude de cas prise pour exemple

- **Élection d'un individu dans un ensemble prédéfini lors d'un concours**
- **Il peut s'agir d'animaux, d'humains ou d'objets quelconques identifiables**
- **L'application à réaliser doit proposer :**
  - Le vote pour un candidat (le vote blanc n'est pas pris en compte)
  - La consultation du nombre de votes (statistiques) des candidats, accessible :
    - Avant de procéder à un vote (factultatif)
    - Nécessairement après avoir voté
- **Est déclaré gagnant le candidat ayant comptabilisé le plus grand nombre de votes**
- **Aucun gagnant si plusieurs ex-æquo à l'issue de l'élection**

Plan général

Partie 1 : Conception préliminaire

Partie 2 : Conception détaillée

Partie 3 : Mise en œuvre avec les JSP

Annexes

Partie 1

Conception préliminaire

- Utilisation du standard UML 2 comme base de modélisation
- Présentation de diagrammes d'analyse
- Définition de stéréotypes adaptés à la modélisation web

Conception préliminaire : étapes de réalisation

1. **Expression des besoins :**
   - Dictionnaire de données
   - Diagramme de contexte statique
   - Cas d'utilisations de contexte
2. **Cas d'utilisation d'implantation**
3. **Modélisation contextuelle de navigation**
4. **Prédéfinition du visuel :**
   - Charte graphique, images, etc.
   - Techniques de navigation
5. **Visuel sous forme de maquette**

Dictionnaire de données

- **Lister les mots-clés**
- **Donner une définition non ambiguë**
- **Les définitions peuvent faire référence à d'autres mots-clés**

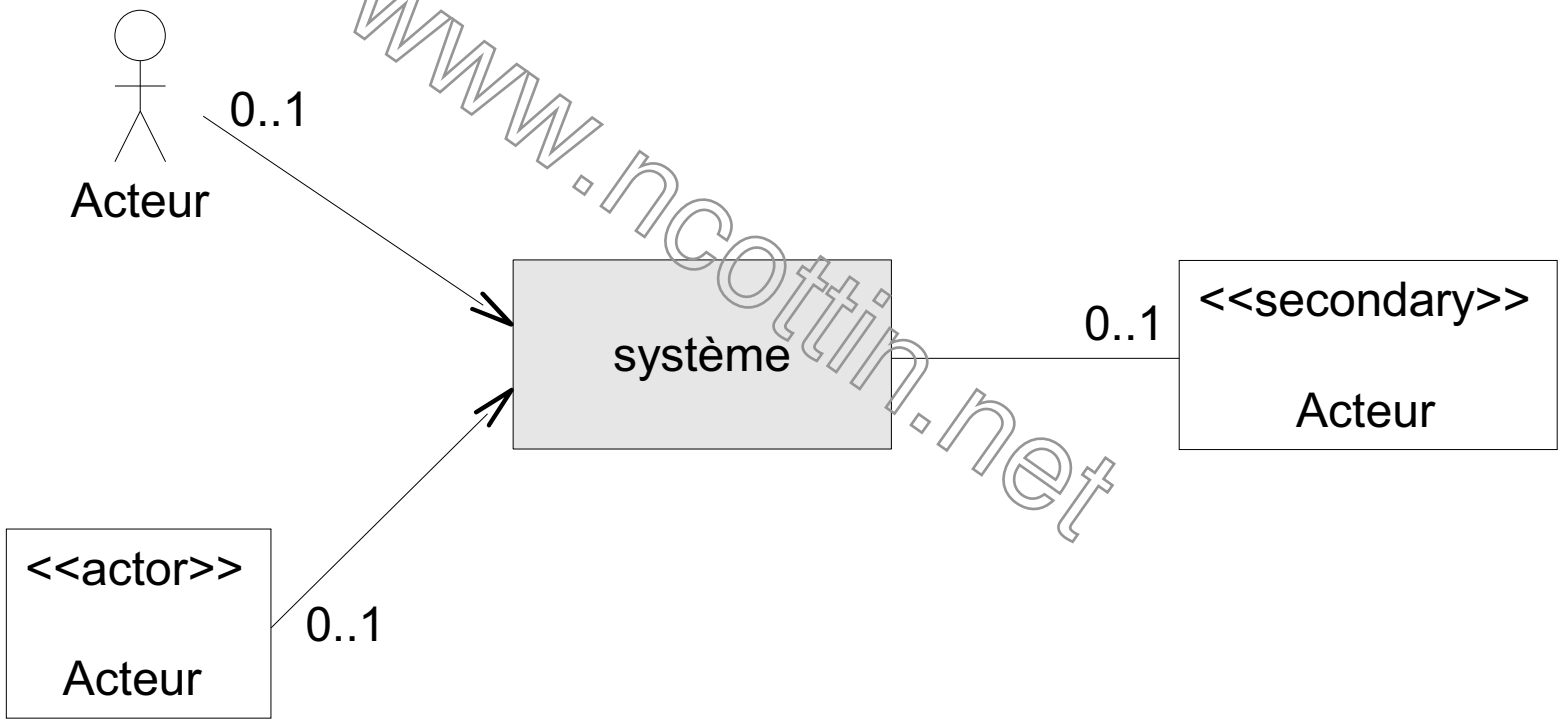
+ Dictionnaire de correspondances au cas où les termes employés par le client et le développeur diffèrent

A éviter...

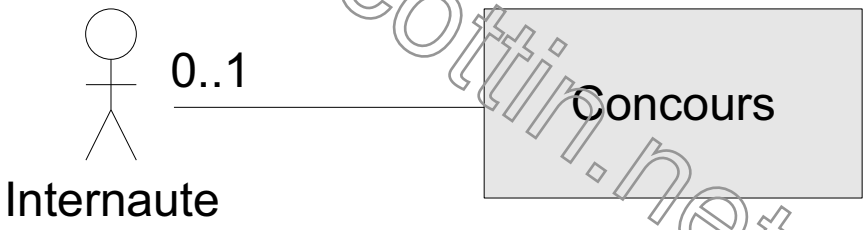
Étude de cas : dictionnaire de données

- **Internaute :** personne physique prenant part au vote
- **Individu :** cible de vote possible qu'un internaute peut choisir lors de son vote
- **Candidat :** *syn.* individu
- **Vote :** processus de sélection d'un individu par un internaute, accessible pendant une période donnée
- **Vote blanc :** également appelé vote nul, permet à un internaute d'indiquer qu'aucun individu ne lui convient
- **Statistique :** ratio du nombre de votes pour un individu (ou votes blancs) par rapport au nombre total de votes
- **Vainqueur :** individu dont la statistique est la plus élevée à la clôture des votes
- **Gagnant :** *syn.* vainqueur

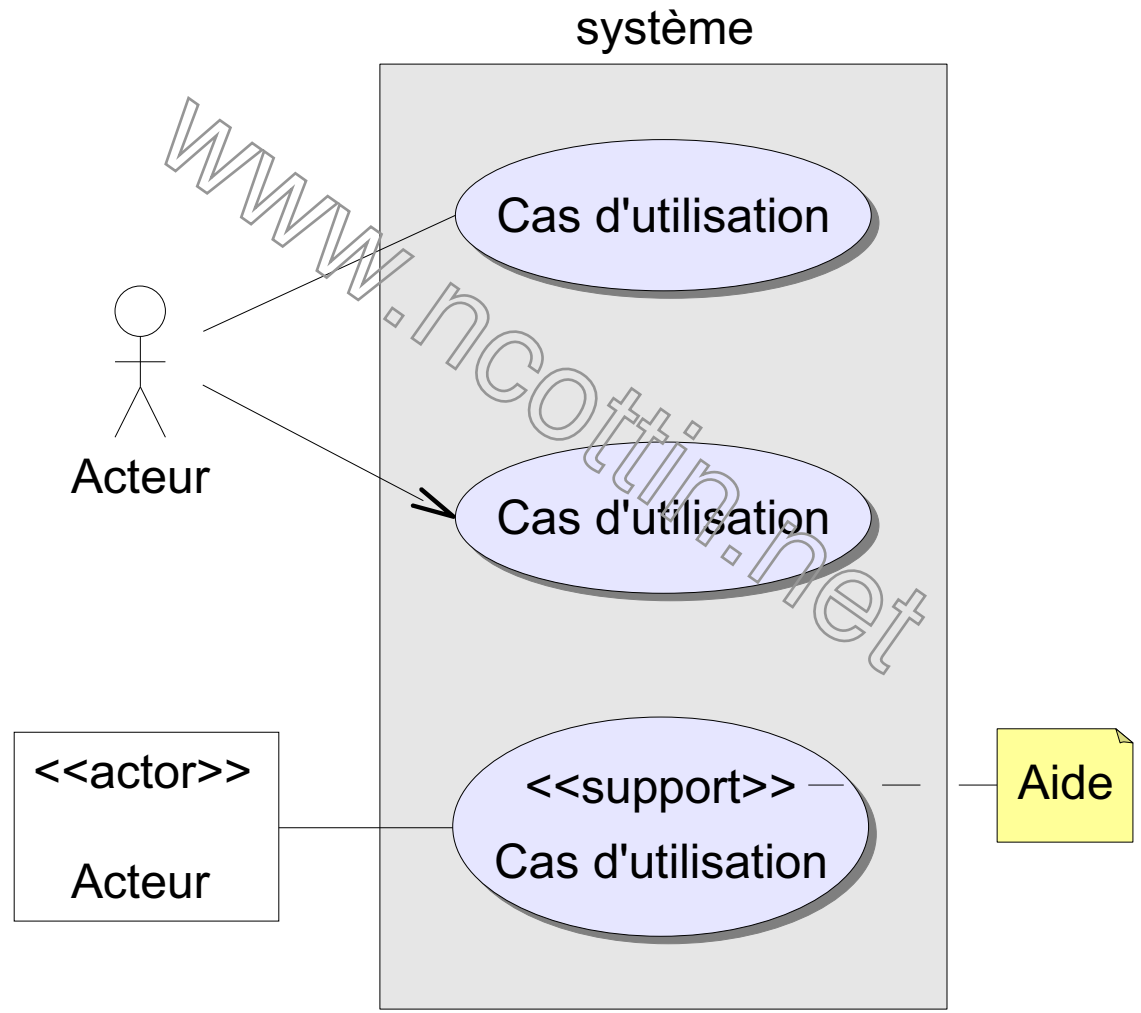
Diagramme de contexte statique : vue d'ensemble



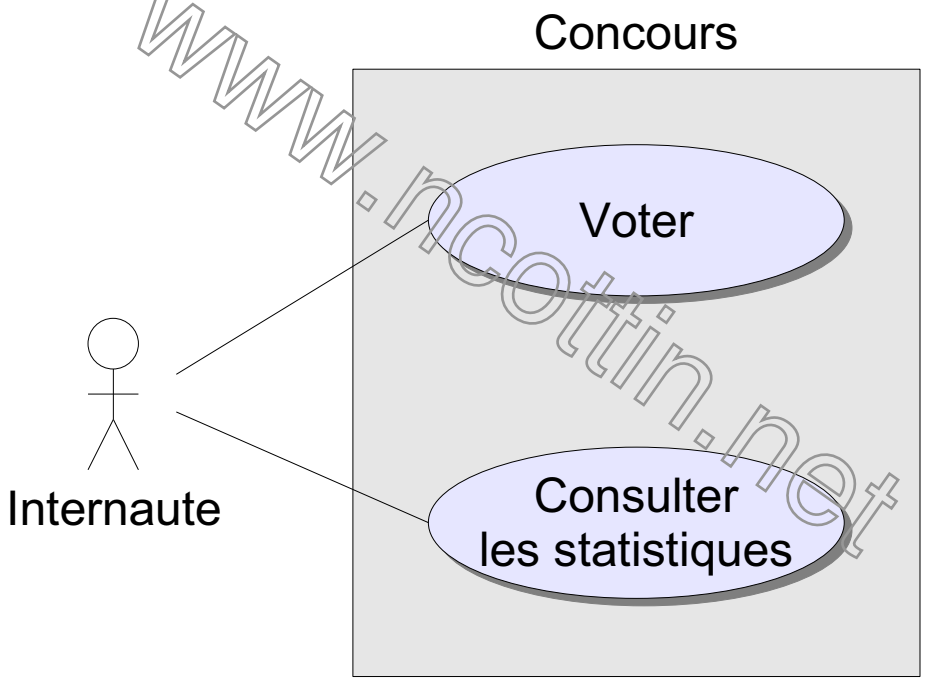
Étude de cas : élection d'un individu lors d'un concours



Cas d'utilisation de contexte : déclaration



Étude de cas : élection d'un individu lors d'un concours



Cas d'utilisation de contexte : description

- **Pour chaque cas d'utilisation identifié, préciser :**
  - Titre
  - Description
  - Acteur(s)
  - Préconditions
  - Postconditions
  - Version
  - Auteur
  - Date de mise à jour

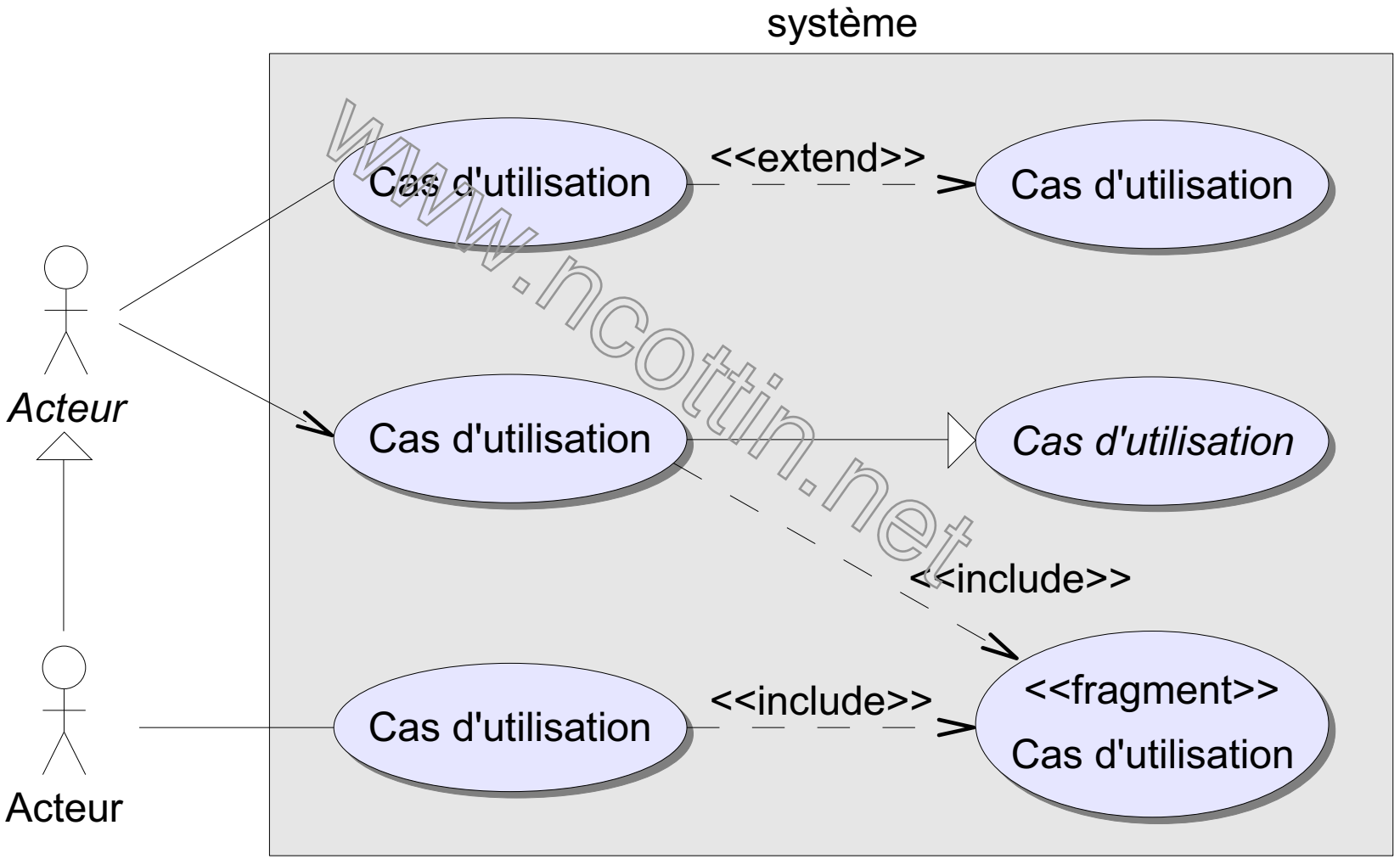
Facultatifs

Étude de cas : description des cas d'utilisation de contexte

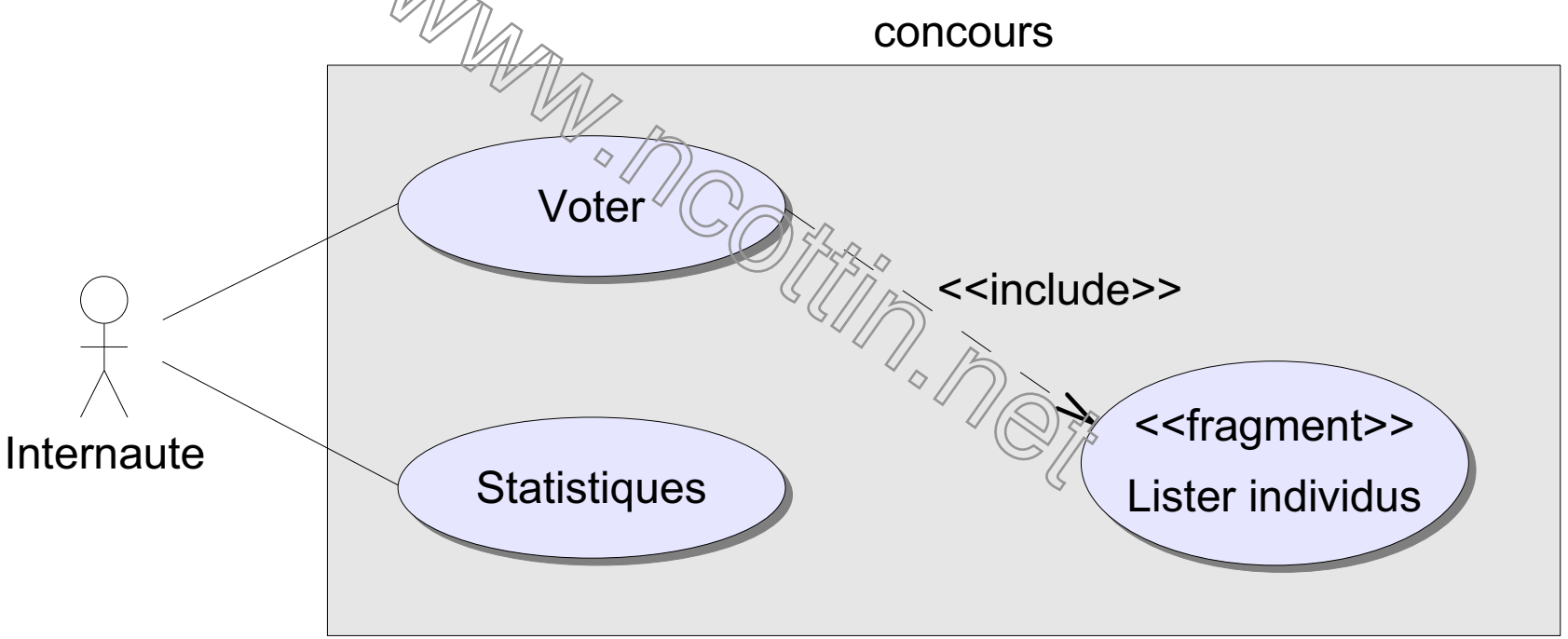
- **Cas d'utilisation :**
  - Voter
  - Propose à l'internaute de choisir un individu dans une liste et de valider son choix
  - Acteur humain : internaute
  - ...

- **Cas d'utilisation :**
  - Consulter les statistiques
  - Affiche pour chaque individu le pourcentage de votes
  - Acteur humain : internaute
  - ...

Cas d'utilisation d'implantation



Étude de cas : cas d'utilisation d'implantation



Cas d'utilisation d'implantation : description

- **Pour chaque cas d'utilisation non encore identifié, préciser :**
  - Titre
  - Description
  - Acteur(s)
  - Préconditions
  - Postconditions
  - Version
  - Auteur
  - Date de mise à jour

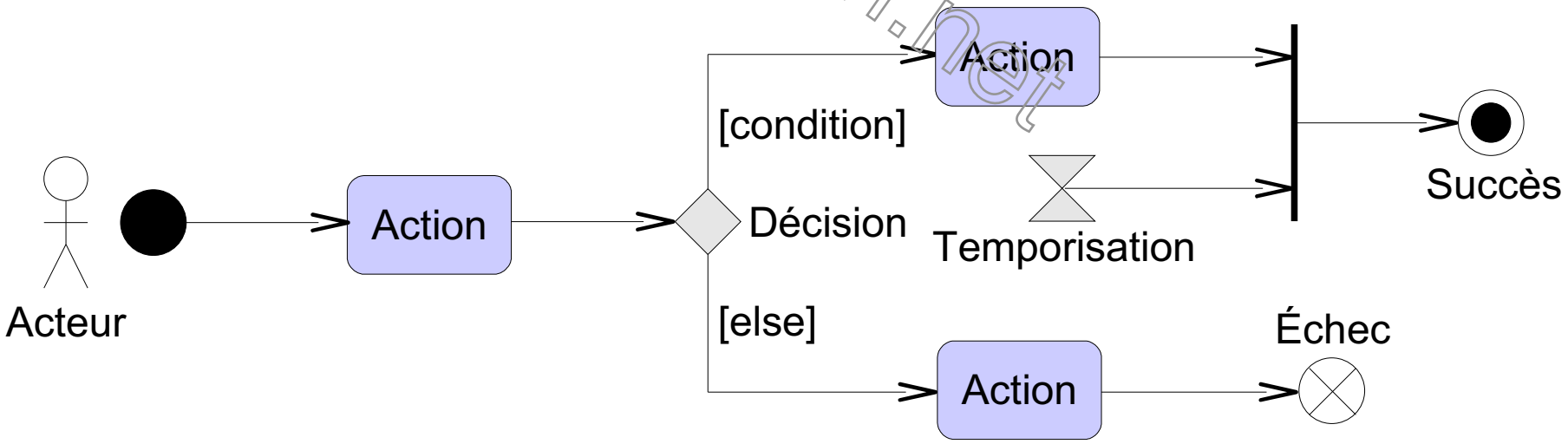
Facultatifs mais recommandés

Étude de cas : description des nouveaux cas d'utilisation

- **Cas d'utilisation :**
  - Lister individus
  - Propose l'ensemble des données d'identité afférentes à chaque individu
  - Aucun acteur (fragment)
  - Préconditions : la liste des individus est initialisée (non vide)
  - Postconditions : néant
  - Version 1.0.0
  - Nathanaël Cottin
  - 9 avril 2007

Déclaration des scénarii

- **Principaux**
- **Alternatifs**
- **Emploi de diagrammes d'activité**



Définition des chemins critiques

- **Établir des relations de précédence entre cas d'utilisation**
- **Notée <**
- **Typiquement, une inclusion se traduit par une précédence**

<<include>> **<**

Étude de cas : définition des chemins critiques

- **La liste des individus doit être disponible *avant* d'offrir la possibilité de voter :**

Lister individus < Voter

Attribution de priorités et de risques aux cas d'utilisation

- **Priorité*** :
  - Très faible (TF)
  - Faible (F)
  - Moyennement faible (MF)
  - Moyennement élevée (ME)
  - Élevée (E)
  - Critique (C)

- **Risque*** :
  - Bas (B)
  - Moyen (M)
  - Haut (H)

Cas d'utilisation	Priorité	Risque
Cas d'utilisation 1	MF	M
Cas d'utilisation 2	C	H
...	...	...

* Décompositions recommandées

Étude de cas : priorités et risques liés aux cas d'utilisation

- **Priorité*** :
  - Très faible (TF)
  - Faible (F)
  - Moyennement faible (MF)
  - Moyennement élevée (ME)
  - Élevée (E)
  - Critique (C)

Cas d'utilisation	Priorité	Risque
Voter	E	H
Consulter les statistiques	MF	B

Ces valeurs permettent un découpage du projet en itérations (processus unifié)

Attention à bien hiérarchiser priorités et risques

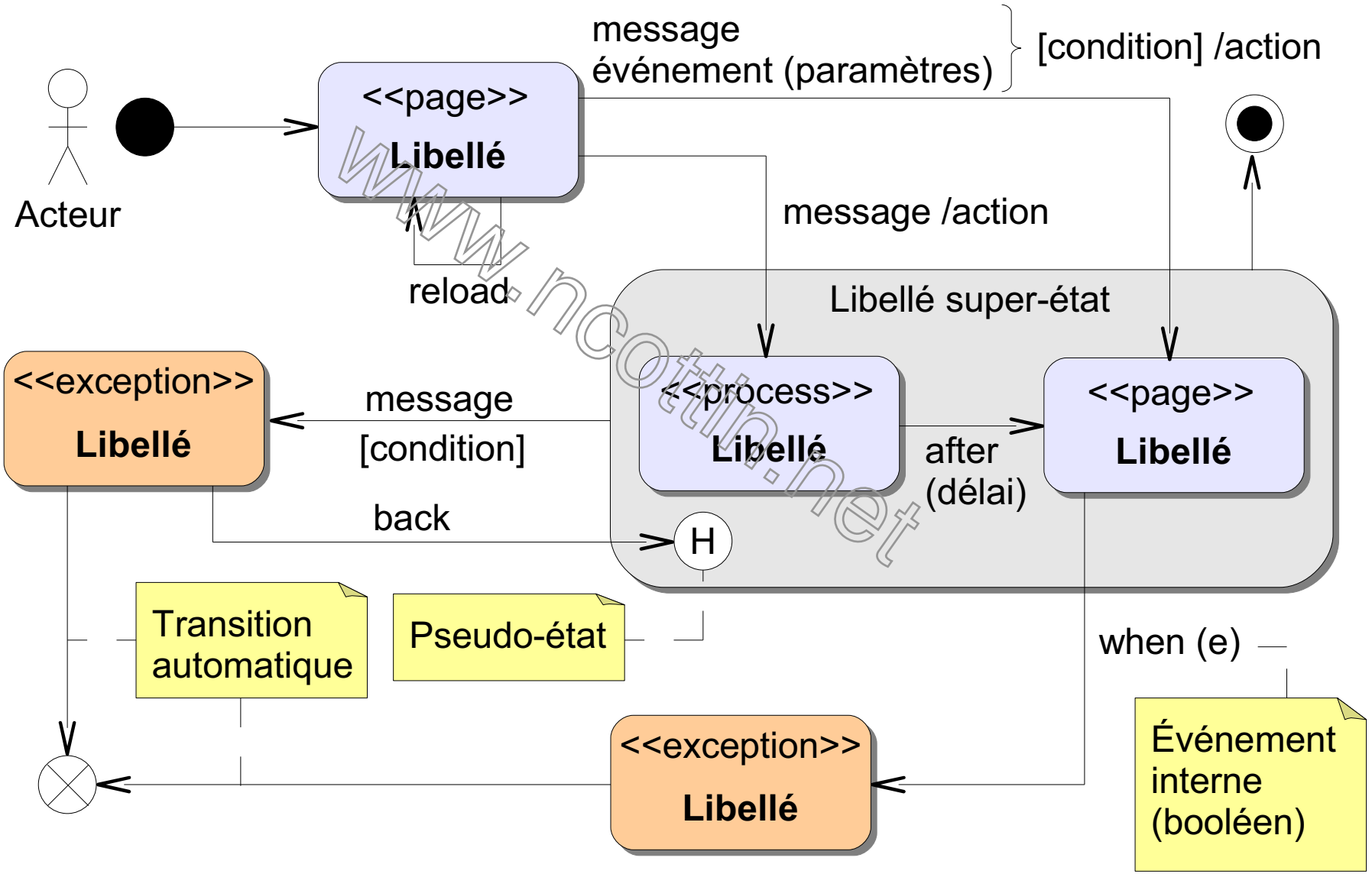
- **Risque*** :
  - Bas (B)
  - Moyen (M)
  - Haut (H)

Éléments de modélisation contextuelle de navigation

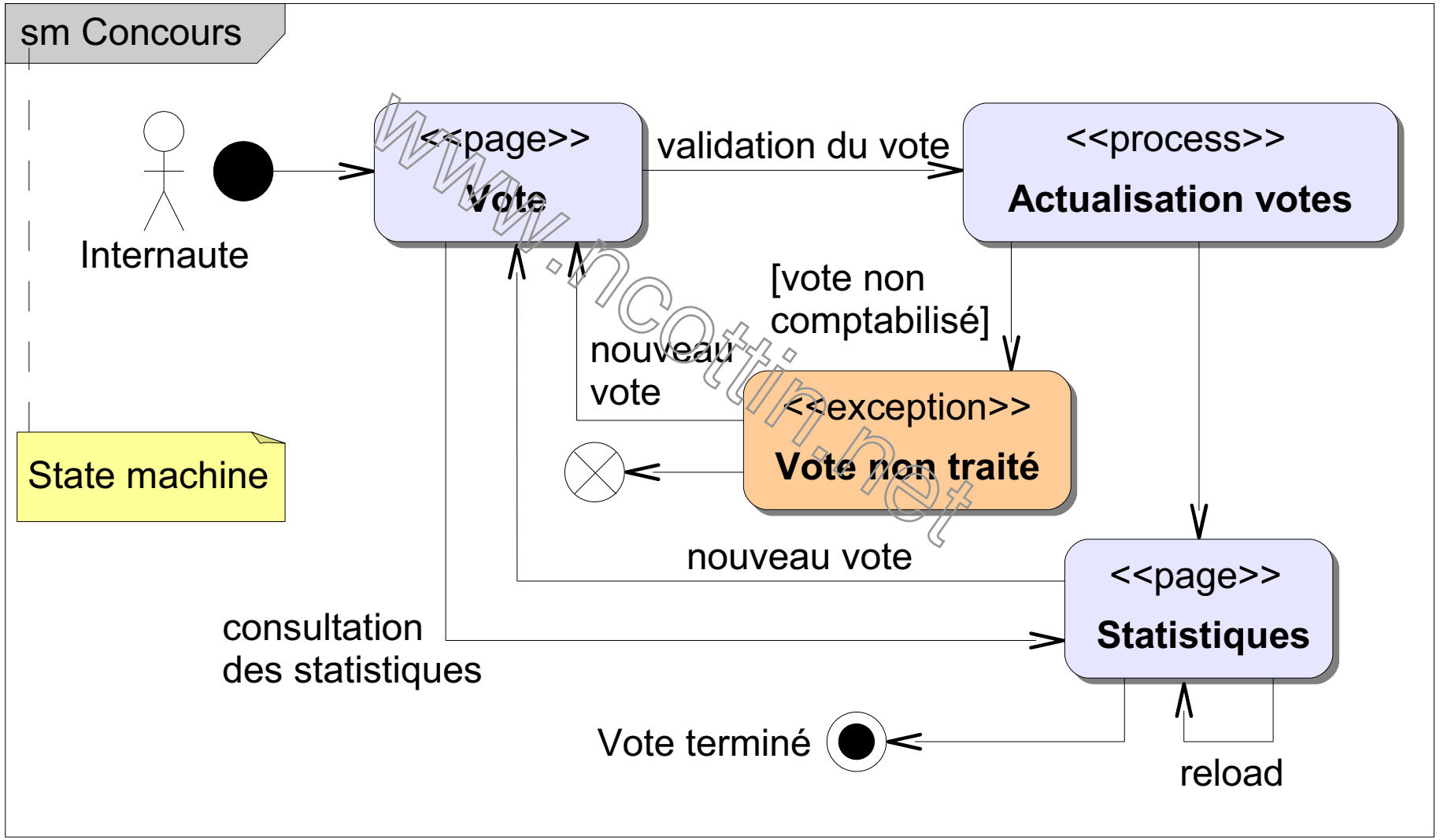
- **Sous forme de diagrammes d'états – transitions :**
  - **Etat** : situation accessible au cours du cycle de vie d'un objet
  - **Transition** : passage d'un état à un autre en réponse à un événement (when, after, ...) ou un message (back, reload, ...)

- **Etats initiaux**
- **Etats finaux** **succès** **échec**

Principes de modélisation contextuelle de navigation



Étude de cas : navigation générale



Prédéfinition de la charte graphique

- **Styles, couleurs et textes alternatifs**

- **Images :**
  - Logos
  - Boutons
  - Autres images

Prédéfinition des techniques de navigation

- **Techniques de navigation :**
  - Menus déroulants
  - Hyperliens

- **Canevas de page :**
  - Homogénéité
  - Conformité aux standards (W3C notamment) : XHTML (Strict), CSS

Réalisation de la maquette

- **Permet au client de valider :**
  - Les informations des pages
  - Les enchaînements de pages
- **Aucun traitement (coquilles vides)**
- **Mise en page sommaire utilisant ou non la charte graphique**

Partie 2

Conception détaillée

- Dérivée des diagrammes UML de conception
- Niveau de détail permettant de générer du code source

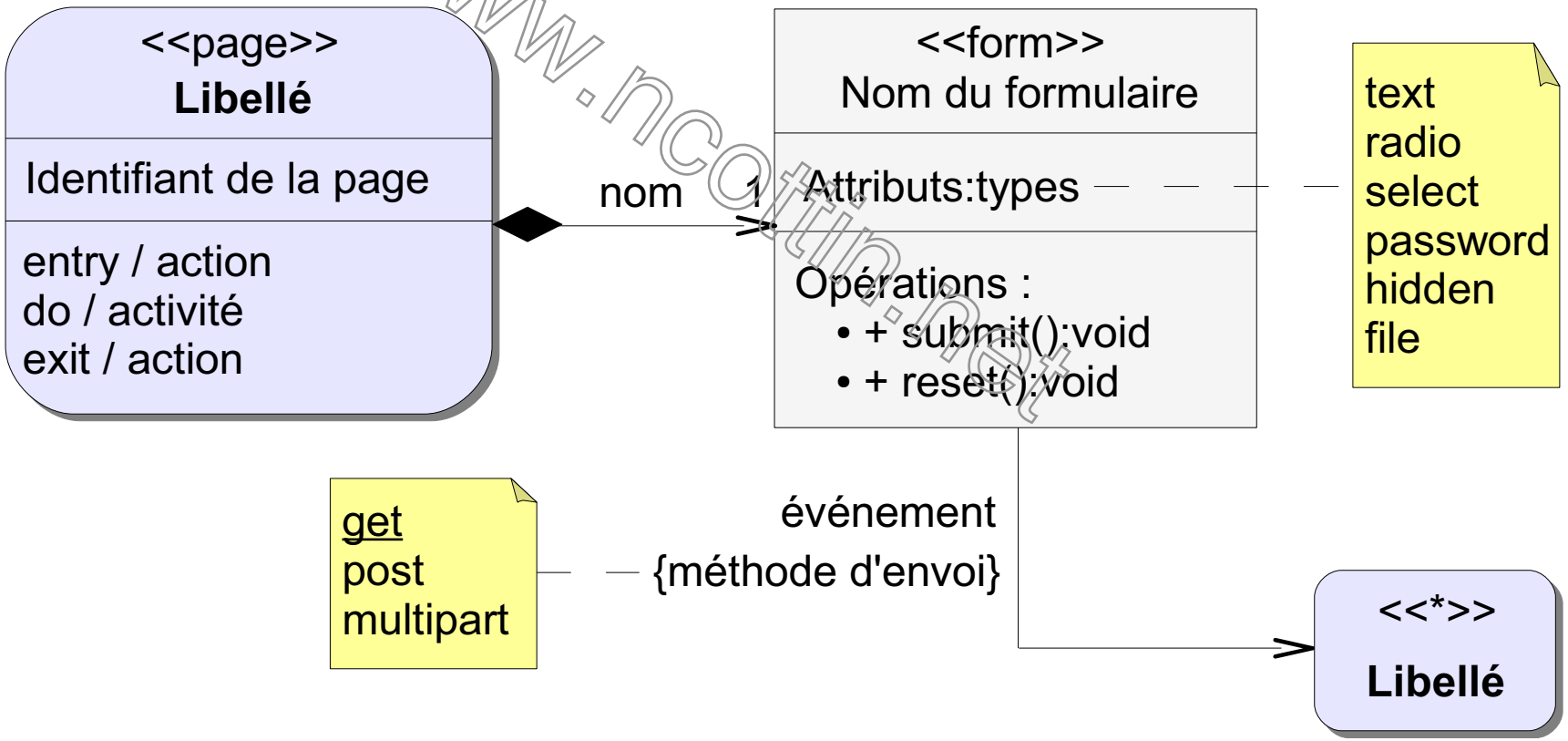
Conception détaillée : étapes de réalisation

1. **Modélisation détaillée de navigation**
2. **Modélisation détaillée de conception**
3. **Modélisation comportementale**
4. **Modélisation des données (tables, requêtes)**
5. **Modification du visuel**
6. **Mise en œuvre**

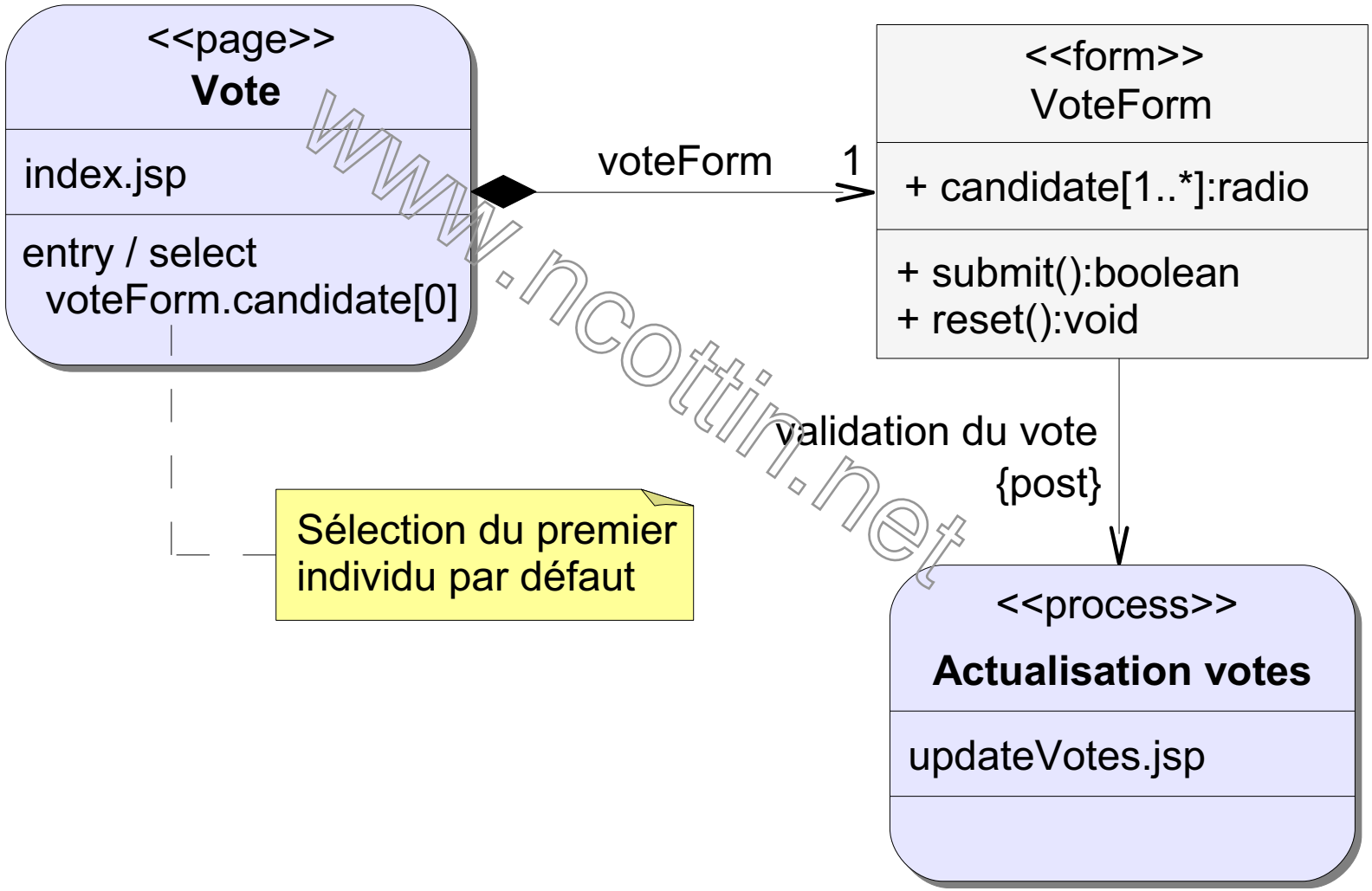
Processus incrémental et itératif

Modélisation détaillée de navigation : déclaration d'un formulaire

- **Met en valeur la transmission d'informations**
- **Utilisation du stéréotype <<form>>**

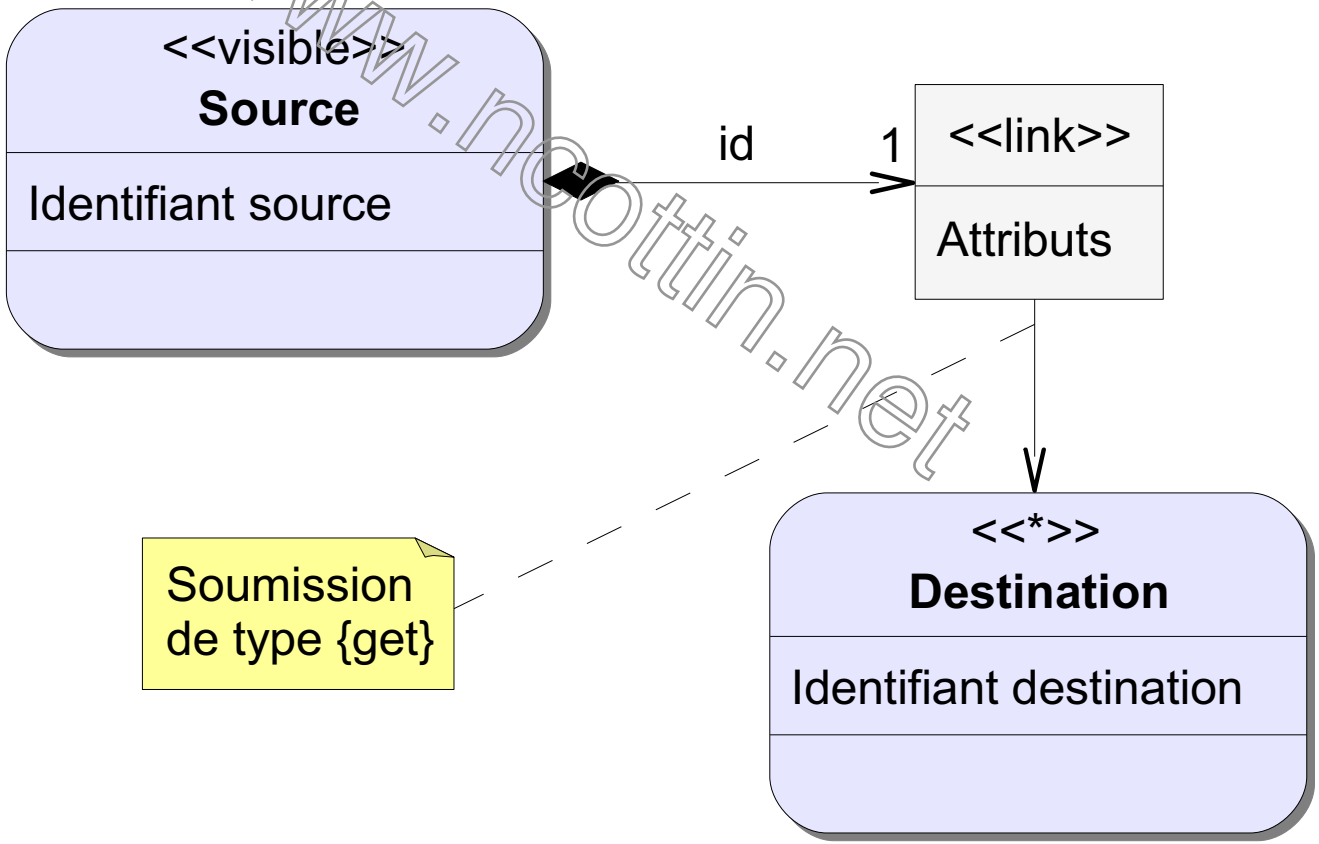


Étude de cas : déclaration du formulaire de vote

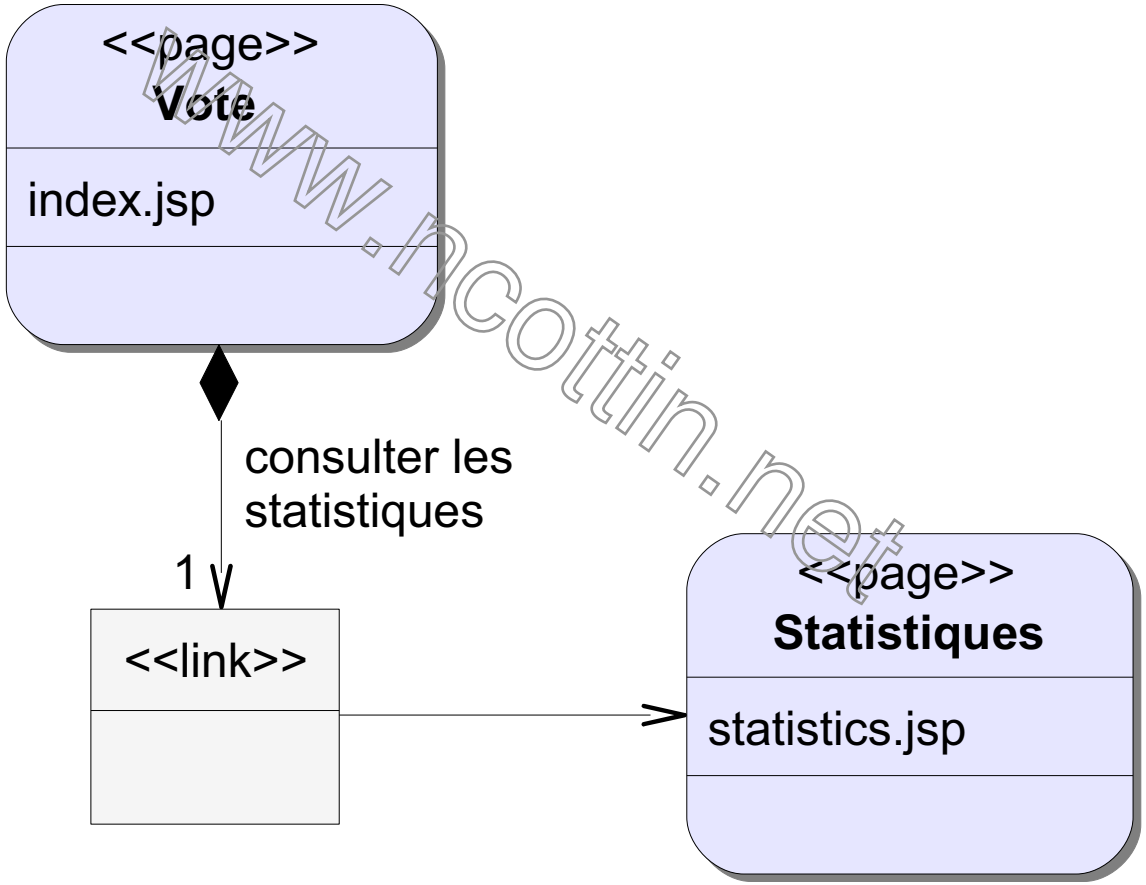


Modélisation détaillée de navigation : intégration d'hyperliens

- **Modélisation des liens pouvant contenir des paramètres**
- **Utilisation du stéréotype <<link>>**

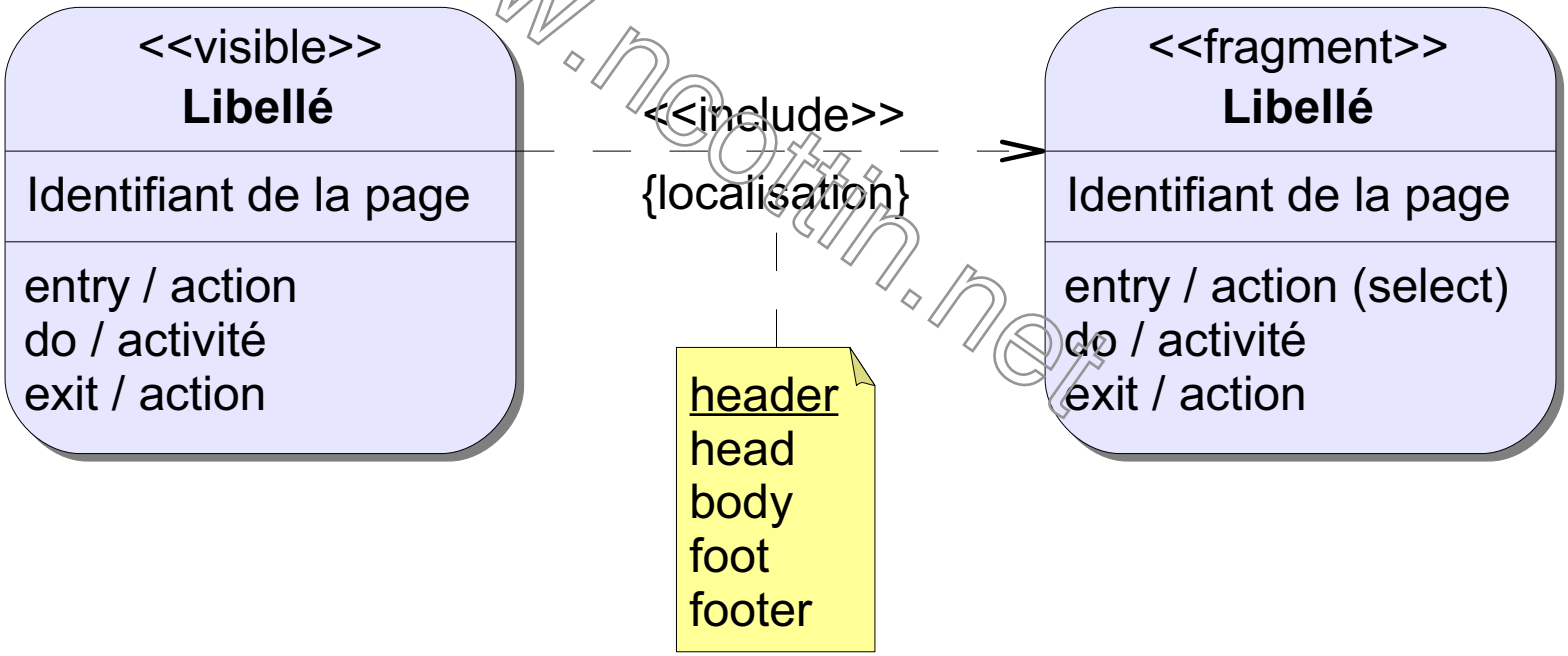


Étude de cas : consultation des statistiques depuis la page de vote

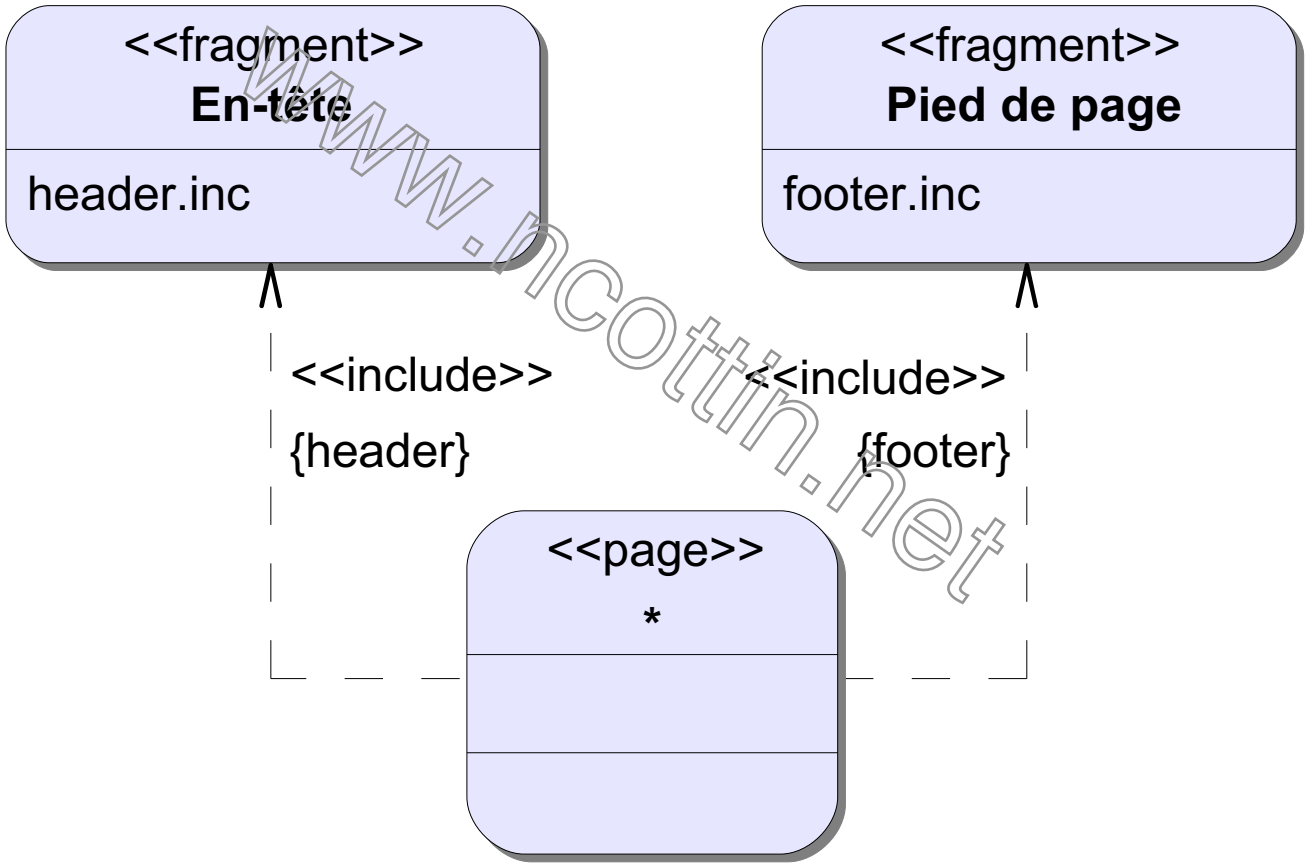


Modélisation détaillée de conception : inclusion de parties

- **Homogénéité des pages (frames, etc.), traitements communs**
- **Utilisation du stéréotype <<fragment>>**

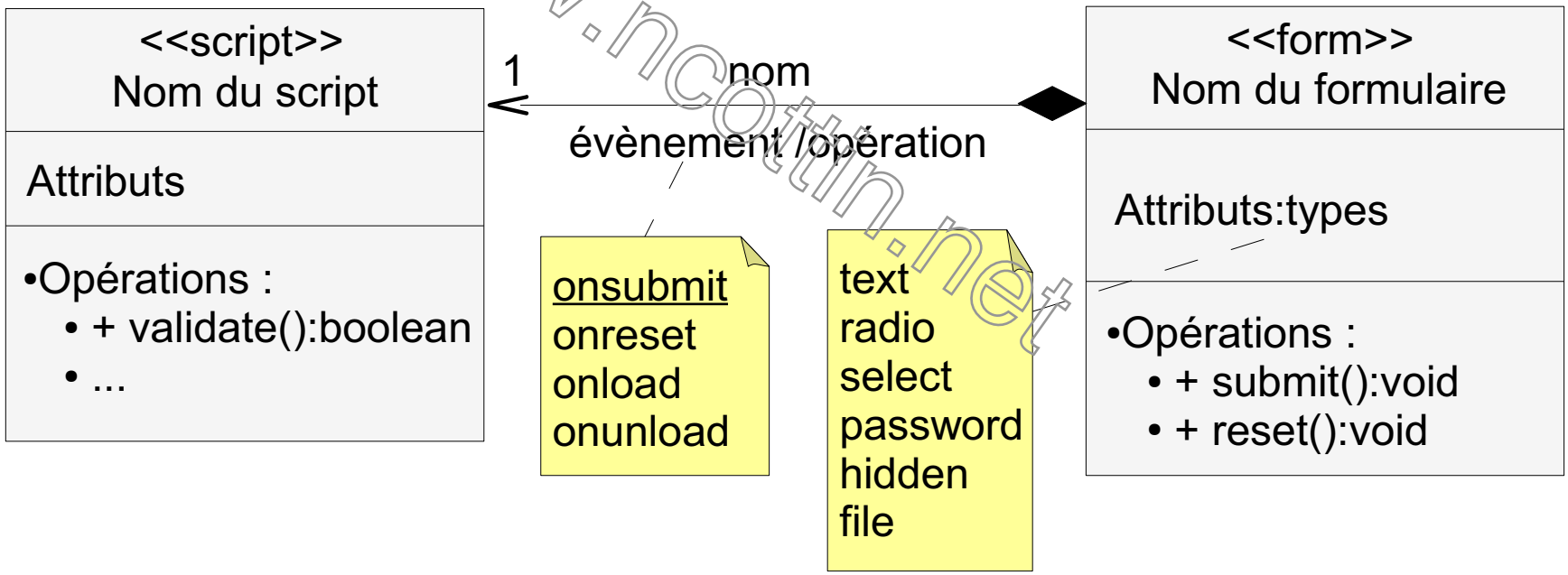


Étude de cas : inclusion d'en-tête et pied de page



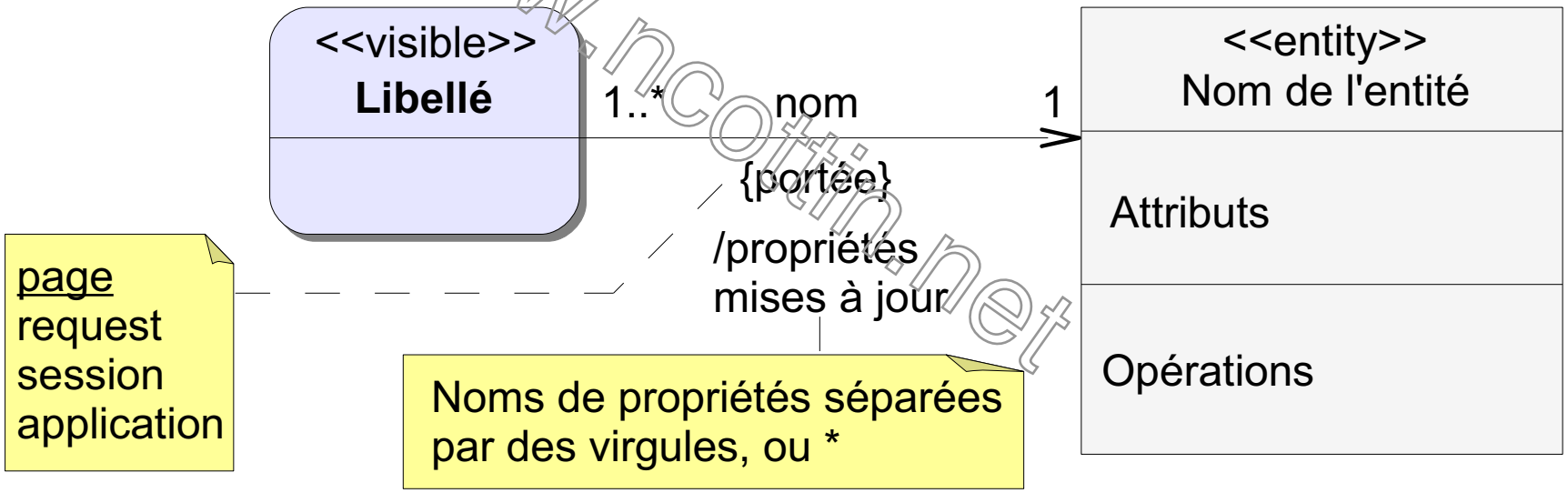
Modélisation détaillée de conception : appel d'un script

- **Un script peut être appelé par un formulaire, une page, etc.**
- **Utilisation du stéréotype <<script>>**



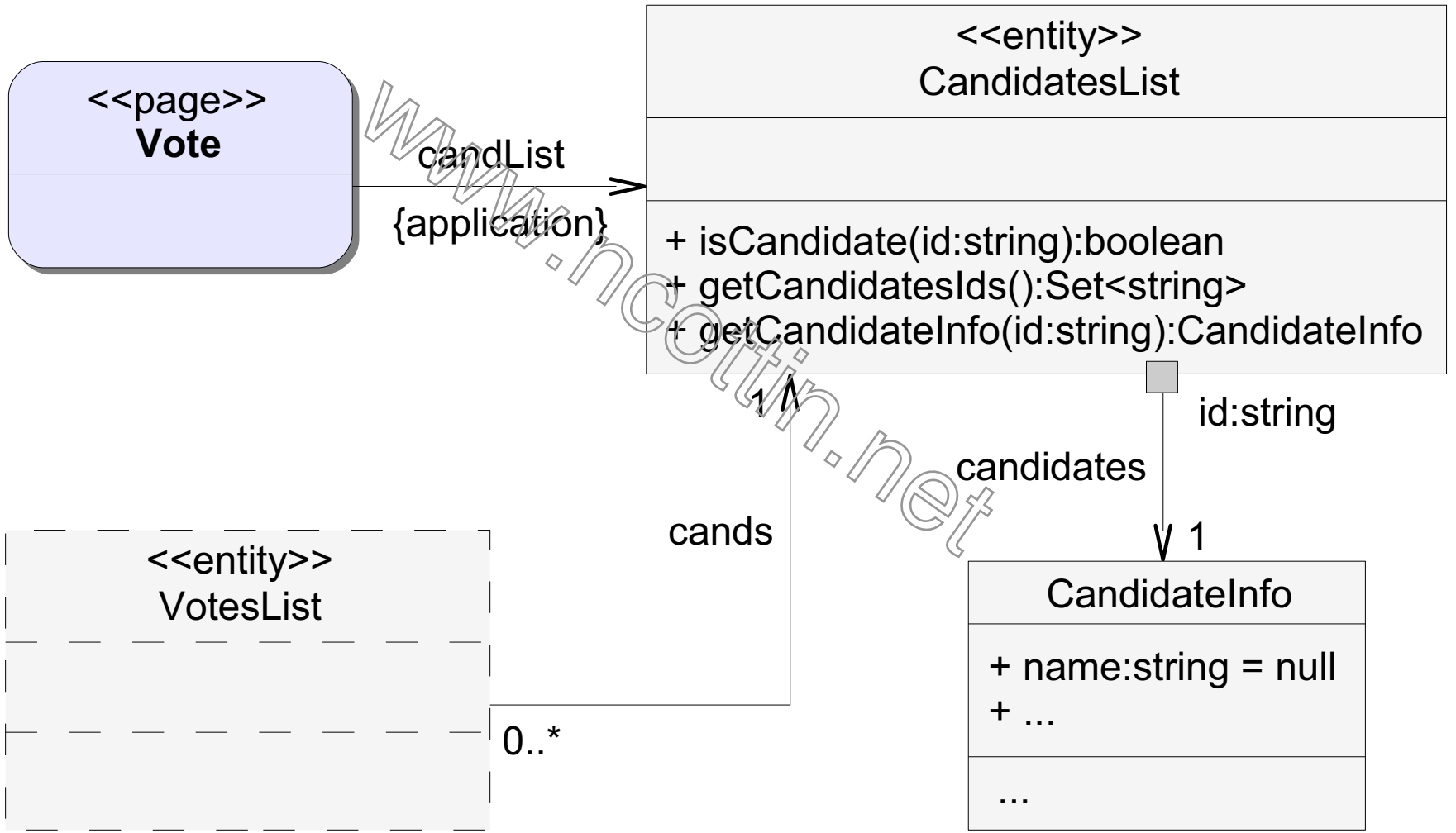
Modélisation détaillée de conception : intégration d'entités

- **Interactions entre pages et entités de programmation**
- **Fait intervenir les diagrammes de classes**

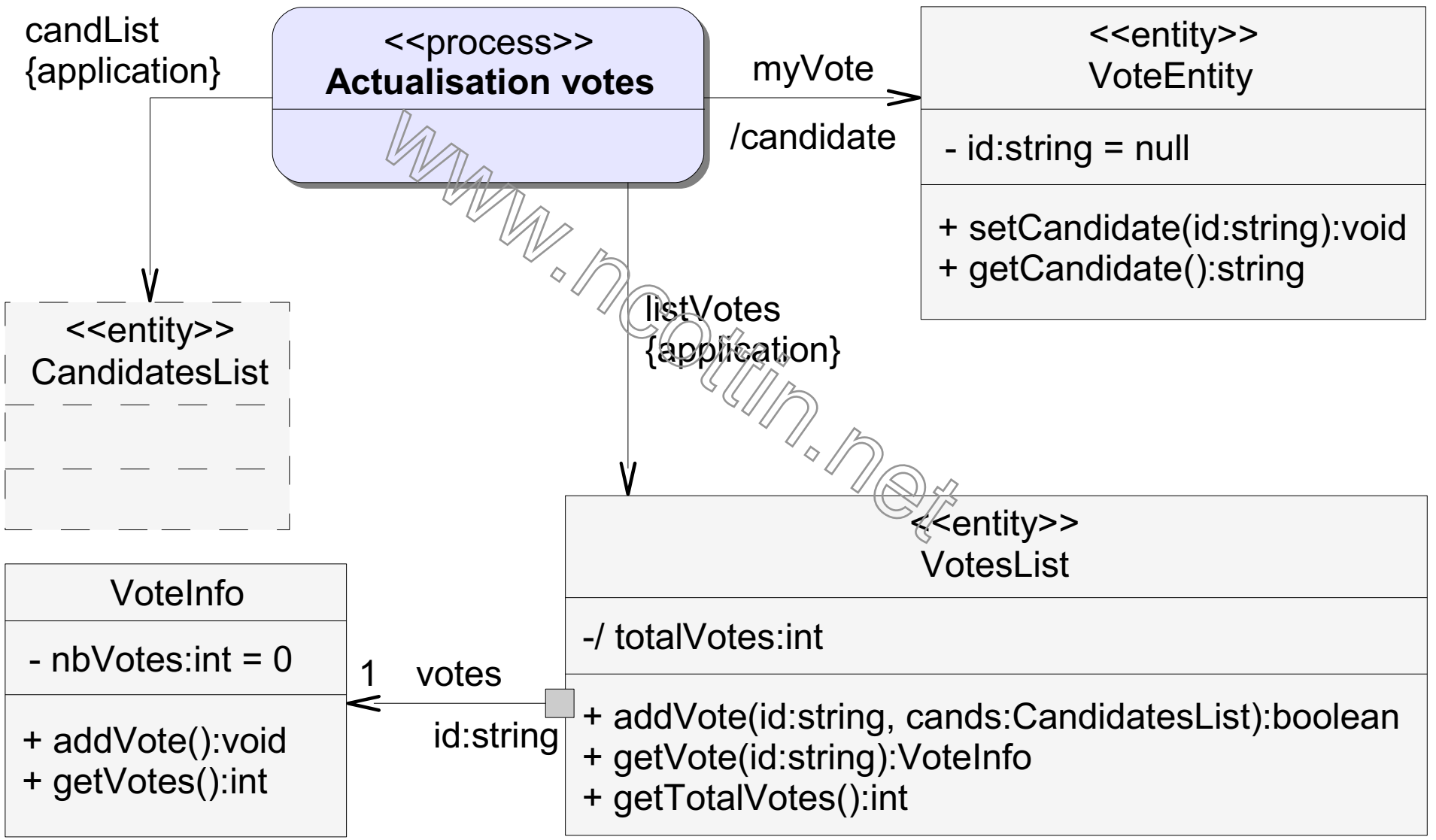


- **Séparation présentation – traitements (paradigme MVC)**

Étude de cas : vote

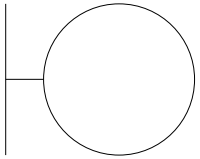


Étude de cas : comptabilisation d'un vote



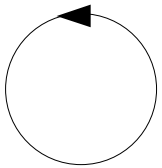
Modélisation comportementale : objets manipulés

- **Modélisation séquentielle des interactions**



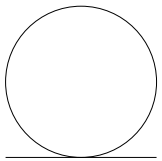
<<page|frame|exception|fragment>>

Interactions entre système et acteurs



<<process>>

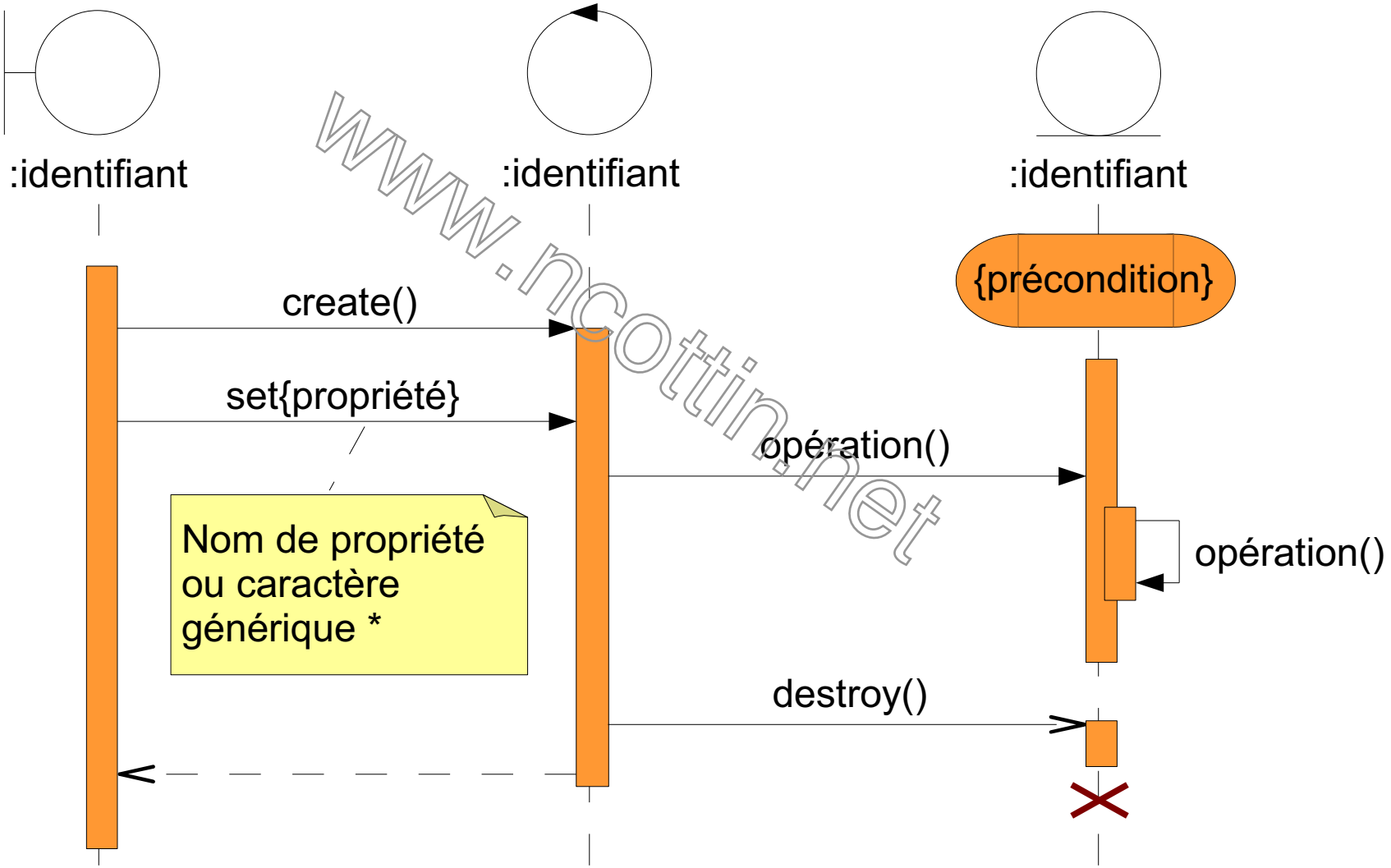
Traitements sans interface avec les acteurs : lien entre pages et entités
A rapprocher de l'élément <<control>> du MVC



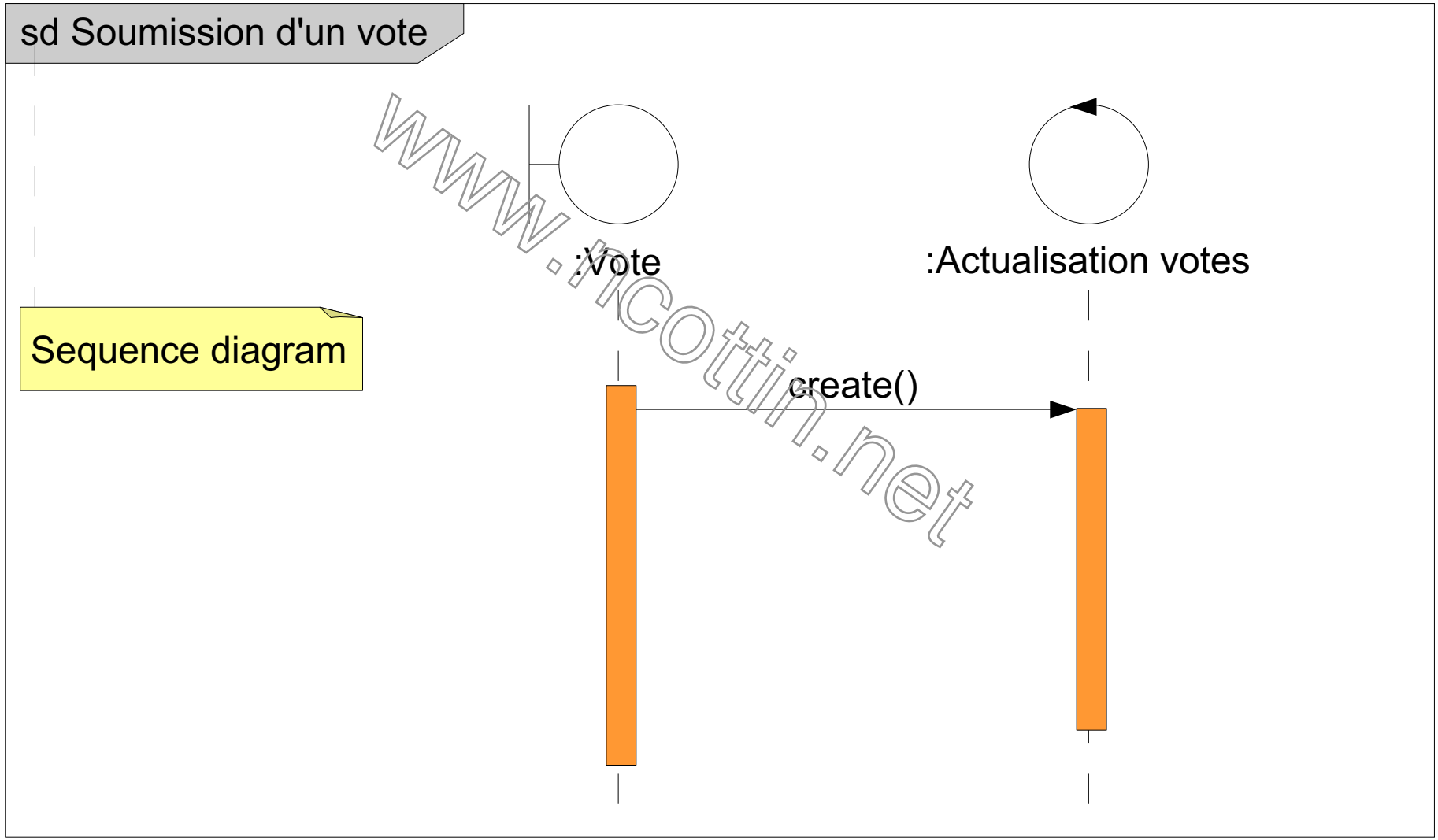
<<entity>>

Unités de traitement (générales ou orientées métier)

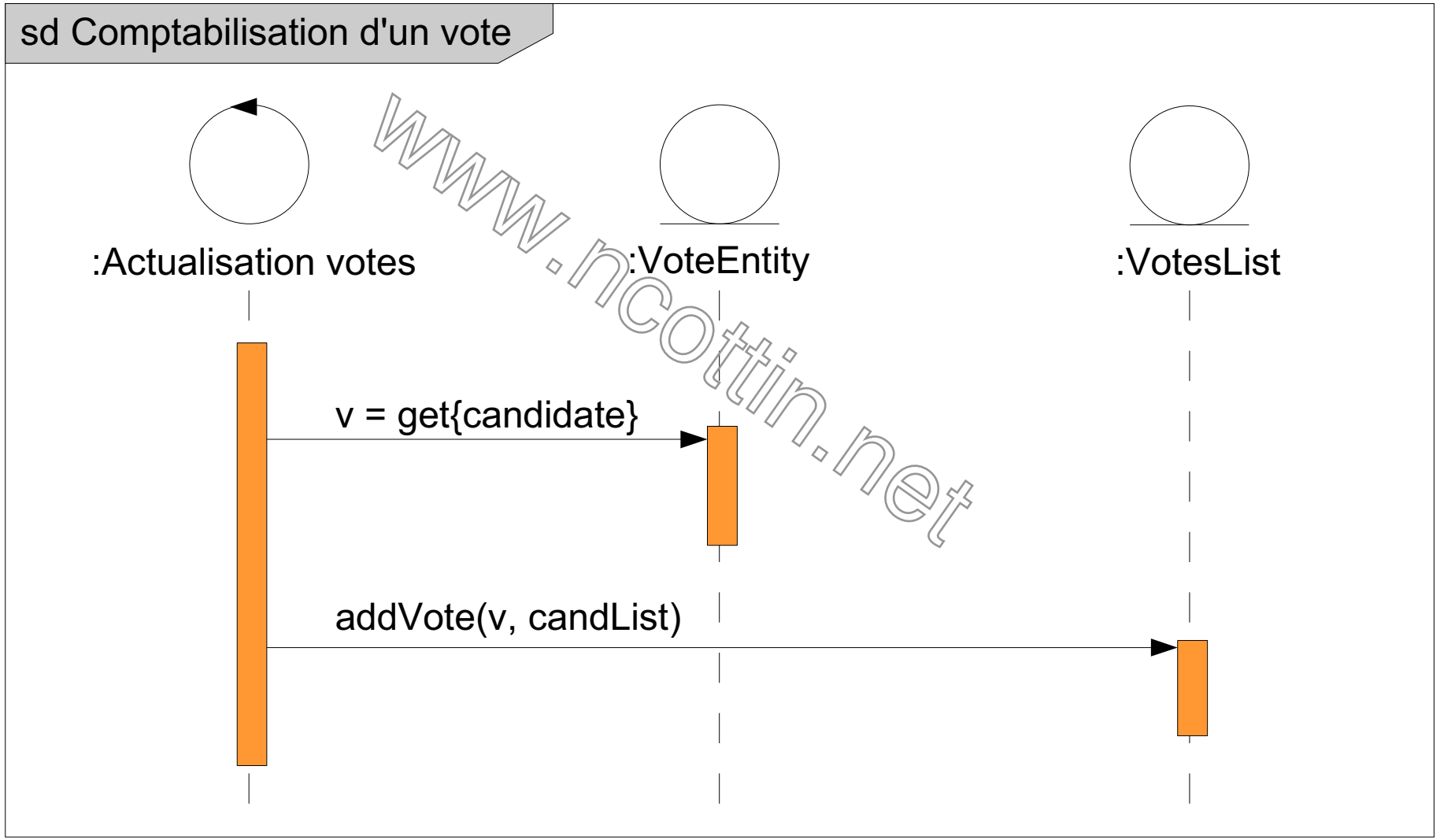
Modélisation comportementale : vue d'ensemble



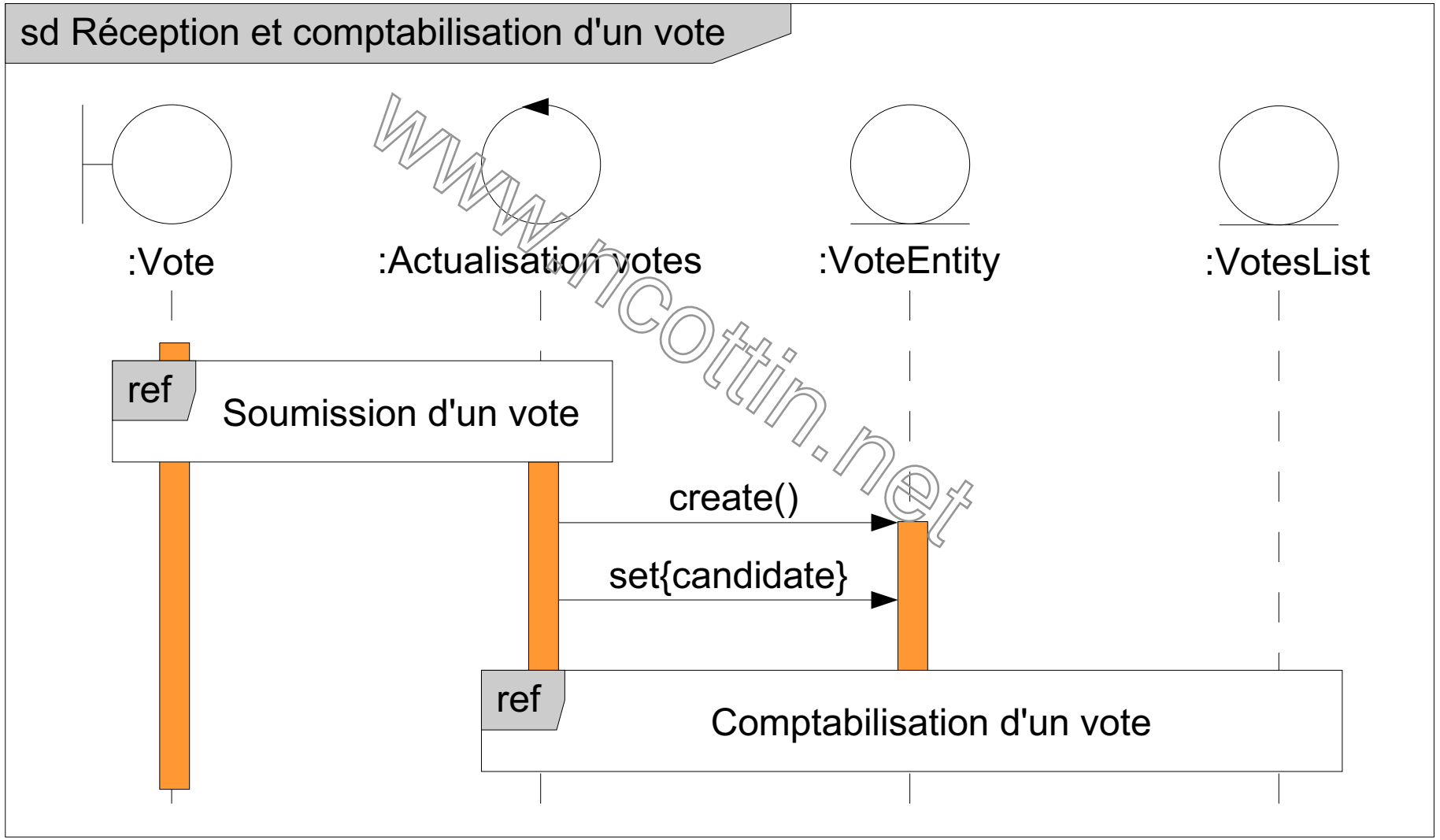
Étude de cas : soumission d'un vote



Étude de cas : comptabilisation d'un nouveau vote



Étude de cas : processus de vote (soumission et comptabilisation)

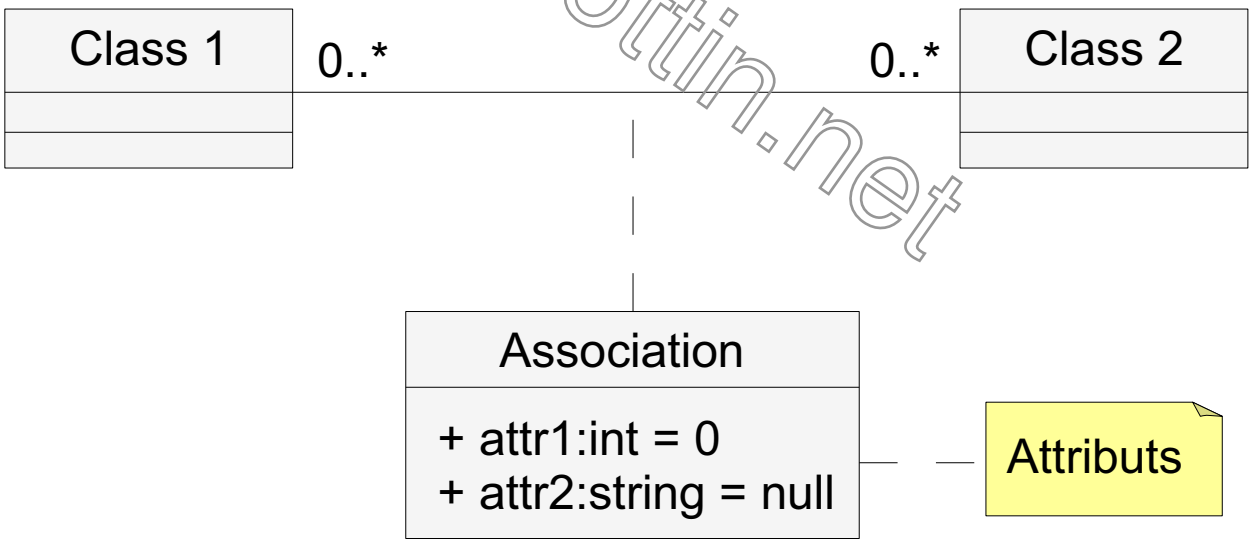


Modélisation des données

- **A l'aide de Merise (déconseillé)**

ou

- **A l'aide de diagrammes de classes utilisant les classes d'associations**



Modification du visuel

- **Nouveaux styles et mises en pages**
- **Images**
- **Prise en compte des nouveaux besoins :**
  - Intégration directe
  - Préparation à l'intégration
- **En fonction des limitations (X)HTML et de la conformité des navigateurs cibles**

Recommandations de mise en œuvre

- **Respecter les priorités établies (cas d'utilisation)**
- **Garder à l'esprit que « le mieux est l'ennemi du bien » (Voltaire)**
- **Débuter si possible par la partie d'administration :**
  - Permet de repérer rapidement les erreurs de modélisation
  - Utile lors des autres développements (reconstruire une base de données, gérer les comptes utilisateurs, etc.)
- **Séparer les rôles :**
  - Administration des serveurs web
  - Développement web
  - Infographie

Partie 3 :

Mise en œuvre avec les JSP

- Exemple de réalisation : étude de cas
- Prérequis : manipulation de JSP et JavaBeans
- Note : beans présentés en annexe

Page de vote : « index.jsp » 1/2


```
<%@page
  language="java"
  contentType="text/html;charset=ISO-8859-1"
  import="java.util.*" %>
<jsp:useBean
  id="candList"
  class="concours.CandidatesList"
  scope="application" />
<%@include file="header.inc" %>
<h1>Votez pour votre favori</h1>
<form name="voteForm" method="post" action="updateVotes.jsp">
<%
Set<String> ids = candList.getCandidatesIds();
for (String id : ids) {
%>
  <p><input type="radio" name="candidate" value="<%= id %>"/>
  <%= candList.getCandidateInfo(id).getName() %></p>
<%
}
%>
```

Page de vote : « index.jsp » 2/2

```
  <p class="button"><input type="submit" value="Envoyer"/>
  <input type="reset" value="Initialiser"/></p>
</form>

<div id="navig">
  <ul>
    <li><a href="statistics.jsp">Consulter les statistiques
                                 actuelles</a></li>
  </ul>
</div>

<%@include file="footer.inc" %>
```

Actualisation des votes : « updateVotes.jsp » 1/2


```
<%@page
  language="java"
  errorPage="error.jsp"
  contentType="text/html;charset=ISO-8859-1" %>

<jsp:useBean
  id="candList"
  class="concours.CandidatesList"
  scope="application" />
<jsp:useBean
  id="listVotes"
  class="concours.VotesList"
  scope="application" />
<jsp:useBean
  id="myVote"
  class="concours.VoteEntity" />
<jsp:setProperty
  name="myVote"
  property="candidate" />
```

Actualisation des votes : « updateVotes.jsp » 2/2

```
<%
listVotes.setCandidatesList(candList);
if (!listVotes.addVote(myVote.getCandidate())) {
  throw new Exception("Vote non traité");
}
%>

<jsp:forward page="statistics.jsp" />
```

Page de statistiques : « statistics.jsp » 1/3


```
<%@page
  language="java"
  contentType="text/html;charset=ISO-8859-1"
  import="java.util.*,java.text.DecimalFormat,concours.*" %>
<jsp:useBean
  id="candList"
  class="concours.CandidatesList"
  scope="application" />
<jsp:useBean
  id="listVotes"
  class="concours.VotesList"
  scope="application" />
<%@include file="header.inc"%>
<%
int total = listVotes.getTotalVotes();
double ratio = (total == 0) ? 0 : (double)100 / total;
Set<String> candIds = candList.getCandidatesIds();
DecimalFormat df = new DecimalFormat("##0.00");
```

Page de statistiques : « statistics.jsp » 2/3

```
CandidateInfo cInfo;
VoteInfo voteInfo;
%>
<h1>Statistiques</h1>
<table>
<%
for (String id : candIds) {
  cInfo = candList.getCandidateInfo(id);
  voteInfo = listVotes.getVoteInfo(id);
%>
  <tr>
    <td><%= cInfo.getName() %></td>
    <td class="vote"><%= df.format(voteInfo.getVotes() * ratio) %></td>
    <td>%</td>
  </tr>
<%
}
%>
```

Page de statistiques : « statistics.jsp » 3/3

```
  <tr class="more">
    <td colspan="3">Total : <span class="vote"><%= total %></span>
vote<%
if (total > 1) {
  out.print("s");
}
%></td>
  </tr>
</table>

<div id="navig">
  <ul>
    <li><a href="statistics.jsp">Mise à jour des statistiques</a></li>
    <li><a href="index.jsp">Nouveau vote</a></li>
  </ul>
</div>
<%@include file="footer.inc"%>
```

Page d'erreur : « error.jsp »


```
<%@page
  language="java"
  isErrorPage="true"
  contentType="text/html;charset=ISO-8859-1" %>

<%@include file="header.inc" %>
<h1 class="error">Erreur</h1>
<p><%= exception.getMessage() %></p>

<div id="navig">
  <ul>
    <li><a href="index.jsp">Nouveau vote</a></li>
  </ul>
</div>
<%@include file="footer.inc" %>
```

Conclusion générale 1/2

- **Attention à la synchronisation des traitements dans les entités et processus**
- **Persistence des entités (sérialisation)**
- **Contrôle des données soumises (en Java) avec « util »**
- **Penser à respecter la distinction entre présentation et traitements (MVC)**
- **Utiliser les « design patterns » si possible**

Différents modèles peuvent répondre favorablement à un même problème

Conclusion générale

- **WAMM permet une génération automatique des pages et entités modélisées ainsi que de la rétroconception**
- **Futures versions :**
  - Prise en compte des informations de page (charset, mime, XML d'en-tête, etc.)
  - Intégration de balises de mise en page (couches DIV, etc.)
  - Ajout de nouveaux stéréotypes en fonction des besoins (<<embed>>, etc.)
- **Objectif : génération automatique > 80% du code**

Références utilisées

[ROQ 07] P. Roques, *UML 2 – Modéliser une application web*, Les cahiers du Programmeur, Eyrolles, 3ème édition, 2007

[COT 07b] N. Cottin, *Servlets et JSP par la pratique*, 2007

[COT 07a] N. Cottin, librairie Java « util », http://util.ncottin.net

[ROQ 06] P. Roques, *UML 2 par la pratique – Etudes de cas et exercices corrigés*, Eyrolles, 5ème édition, 2006

[DEL 05] J.-L. Deleage, *JSP et servlets efficaces*, Dunod, 2005

[JAC 99] I. Jacobson et al., *The Unified Software Development Process*, Addison Wesley, 1999

Partie 4 :

Annexes

- En-tête et pied de page inclus au sein des pages JSP
- Fichier de styles
- JavaBeans utilisés

Fichier d'en-tête : « header.inc »


```
<?xml version="1.0" encoding="ISO-8859-1"?>
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Strict//EN"
   "http://www.w3.org/TR/xhtml1/DTD/xhtml1-strict.dtd"><%!
private static final String titre = "Election lors d'un concours"; %>
<html xmlns="http://www.w3.org/1999/xhtml" xml:lang="fr">
<head>
  <title><%= titre %></title>
  <link rel="stylesheet" title="Default style" type="text/css"
        href="default.css"/>
</head>
<body>
<div id="title">
  <h1><%= titre %></h1>
</div>
<div id="info">
  <p>Informations diverses...</p>
</div>
<div id="data">
```

Fichier de pied de page : « footer.inc »

```
    </div>
  </body>
</html>
```

Bean «VoteEntity »


```
package concours;

public final class VoteEntity {

  private String value = null;

  public void setCandidate(String id) {
    value = id;
  }

  public String getCandidate() {
    return value;
  }
}
```

Classe « CandidateInfo »


```
package concours;

public final class CandidateInfo {

  private String name = null;

  public void setName(String value) {
    name = value;
  }

  public String getName() {
    return name;
  }
}
```

Bean «CandidatesList » 1/2


```
package concours;

import java.util.HashMap;
import java.util.Map;
import java.util.Set;

public final class CandidatesList {

  private Map<String, CandidateInfo> candidates =
    new HashMap<String, CandidateInfo>();

  public CandidatesList() {
    CandidateInfo info;
    for (int id=1; id<10; id++) {
      info = new CandidateInfo();
      info.setName("Candidat " + id);
      candidates.put("cand" + id, info);
    }
  }
```

Bean «CandidatesList » 2/2

```
    public boolean isCandidate(String id) {
      return (id != null) && candidates.containsKey(id);
    }

    public Set<String> getCandidatesIds() {
      return candidates.keySet();
    }

    public CandidateInfo getCandidateInfo(String id) {
      return (id != null) && candidates.containsKey(id)
        ? candidates.get(id)
        : null;
    }
}
```

Classe « VoteInfo »


```
package concours;

public final class VoteInfo {

  private int nbVotes = 0;

  public void addVote() {
    nbVotes++;
  }

  public int getVotes() {
    return nbVotes;
  }
}
```

Bean «VotesList » 1/3


```
package concours;

import java.util.HashMap;
import java.util.Map;

public final class VotesList {

  private Map<String, VoteInfo> votes = new HashMap<String, VoteInfo>();
  private int totalVotes = 0;
```

Bean «VotesList » 2/3


```
public synchronized boolean addVote(String id, CandidatesList cands) {
  if ((cands == null) || !cands.isCandidate(id)) {
    return false;
  }

  VoteInfo info;
  if (votes.containsKey(id)) {
    info = votes.get(id);
  }
  else {
    info = new VoteInfo();
  }

  info.addVote();
  votes.put(id, info);
  totalVotes++;
  return true;
}
```

Bean «VotesList » 3/3


```
  public synchronized VoteInfo getVoteInfo(String id) {
    if (id == null) {
      return null;
    }

    return votes.containsKey(id)
      ? votes.get(id)
      : new VoteInfo();
  }

  public synchronized int getTotalVotes() {
    return totalVotes;
  }
}
```

CSS employé : « default.css » 1/7


```
body {
  margin: 0;
  padding: 0 4em 0 4em; /* top right bottom left */
  color: black;
  font: 90% "Trebuchet MS", helvetica, sans-serif;
}

div#title {
  padding: 2em 0 2em 0;
  clear: both;
}

div#title > h1 {
  font-size: 1.5em;
  text-align: center;
  font-weight: bold;
  border-top: 1px solid #CCCCCC;
  border-bottom: 1px solid #CCCCCC;
  margin-top: .7em;
}
```

CSS employé : « default.css » 2/7


```
h1 {
  font-size: 1.3em;
  font-weight: bold;
}

h1.error {
  color: Red;
}

div#info {
  float: right;
  top: 7em;
  border: 1px solid #CCCCCC;
  background: #F5F5F5;
  padding: 1em;
  width: 35%;
}
```

CSS employé : « default.css » 3/7

```
div#info p.caution {
  color: Red;
  font-variant: small-caps;
}

div#data {
  width: 60%;
}

div#navig {
  clear: both;
  margin-top: 4em;
  border-top: 1px solid #CCCCCC;
}
```

CSS employé : « default.css » 4/7


```
div#navig ul {
  padding-bottom: .5em;
  padding-bottom: .5em;
  float: right;
  margin: 0;
  list-style-type: none;
}

div#navig ul li {
  float: left;
  text-align: right;
  padding-top: 0;
  padding-bottom: 0;
  padding-right: 2em;
}
```

CSS employé : « default.css » 5/7


```
div#navig a {
  color: Blue;
  text-decoration: none;
  outline: none;
}

div#navig a:hover {
  color: Green;
}

table td,
table td.vote {
  padding: .2em;
}
```

CSS employé : « default.css » 6/7


```
span.vote,
table td.vote {
  font-family: "Courier New";
  font-weight: bold;
}

table td.vote {
  text-align: right;
}

table tr.more td {
  padding-top: 1em;
}

input {
  margin-right: .5em;
}
```

CSS employé : « default.css » 7/7

```
p.selection {
  margin: 0;
  margin-top: .5em;
}

p.button input {
  margin-left: 1em;
  margin-right: 1em;
  padding: .5em;
}
```