

**Unité d'Enseignement en Informatique
Année 2013-2014**

**Master M1 EFREI – ASI – ISI
Devoir Écrit de ERP – Première Session du 9 Avril 2013
(durée 2h00, aucun document autorisé)**

Consigne: Réponses en Français ou en Anglais, terminologie Anglophone autorisée dans les deux cas.

Exercice 1: (4 points)

- 1) Que signifie le sigle ERP ?
- 2) Dans une entreprise, à quoi sert un logiciel ERP ?
- 3) Que signifie le sigle EAI ?
- 4) Quelles sont les différences d'architecture entre les approches ERP et EAI ?
- 5) Donnez 2 avantages des ERP comparé aux EAI.
- 6) Donnez 1 avantage des EAI comparé aux ERP.

Exercice 2: (4 points)

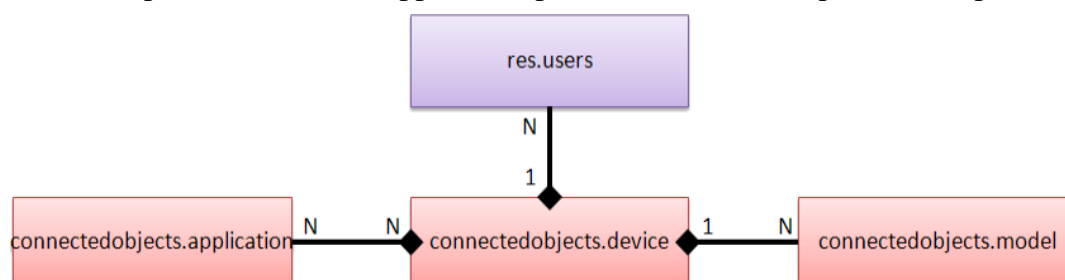
Trois solutions de logiciels pour votre entreprise sont présentées comme des ERP par leur vendeur. Une analyse de ces trois ERP, montrant quelles applications sont intégrées (I) au composant central de l'ERP et lesquelles sont installables en tant que module (M), est présentée dans le tableau suivant :

Application	ERP1	ERP2	ERP3
Sales	I	M	I
Purchasing		M	I
Warehouse Management	I	M	I
CRM		M	I
MRP	I	M	I
Accounting		M	M
Human Resources		M	M
CMMS	I	M	

- 1) Dans quelle grande catégorie d'ERP classez-vous chacun de ces 3 ERP ?
- 2) Lors d'une réunion de la DSI de votre entreprise, les personnes présentes s'interrogent sur l'installation de chacune de ces 3 configurations. On vous demande votre avis. Quels avantages et inconvénients donneriez-vous pour chacun des 3 ERP ?

Exercice 3: (12 points)

De nombreux objets connectés ont été déployés au sein d'une entreprise. Pour aider à la gestion de ce vaste parc, nous souhaitons gérer l'affectation des objets connectés et les applications installées dessus en développant un nouveau module OpenERP. Le modèle de donnée de ce nouveau module est décrit ci-dessous. Un dispositif est possédé par un utilisateur et un utilisateur peut posséder plusieurs dispositifs. Différents modèles de dispositifs existent. Chaque dispositif appartient à un modèle et plusieurs dispositifs peuvent être du même modèle. Plusieurs applications peuvent être installées sur un dispositif et la même application peut être installée sur plusieurs dispositifs.



L'écriture du code source de `connectedobject.model` a déjà commencé. Voir les trois fichiers `connectedobject.py`, `connectedobject_data.xml` and `connectedobject_view.xml` ci-après. Afin de poursuivre ce développement, vous devez répondre aux questions suivantes.

- 1) Créer trois nouveaux filtres pour `connectedobjects.model` afin de filtrer les modèles avec :
 - a) WiFi et 3G
 - b) WiFi seulement (ni 3G, ni 4G)
 - c) 3G ou 4G
- 2) Définir l'objet métier `connectedobjects.application` avec 2 colonnes définissant le nom et la version du logiciel. Puis, écrire les vues `tree` et `form` correspondantes.
- 3) Définir l'objet métier `connectedobjects.device` avec 4 colonnes, définissant notamment les trois relations avec les autres objets métiers.
- 4) Dans `connectedobjects.model`, est-ce une bonne solution d'avoir plusieurs booléens pour décrire les différentes connexions possibles ? Pourquoi ? Changeriez-vous pour une autre solution ? Si oui, alors décrivez comment ?
- 5) Que permettent les services web (ou "webservices") ? Dans le cas d'OpenERP, citez 5 fonctions de l'O.R.M. qui sont accessibles via les services web. Quel protocole est utilisé par OpenERP pour ses services web ? Que crée le script appelé `create.php` ci-après ?
- 6) Donnez les modifications de code à apporter au script `create.php` afin de créer un nouvel enregistrement de `connectedobjects.application` pour une version du logiciel Safari.

Documentation :

- a) Exemple de test d'une valeur booléenne dans un filtre : `[('active', '=', True)]`
- b) Le domaine d'un filtre peut aussi évaluer plusieurs conditions en utilisant les opérateurs booléens en notation préfixée (aussi connue comme notation Polonaise). Par exemple, filtrer si le champ `country_code` vaut celui de la Belgique OU de la France serait fait comme suit :

```
[ '|', ('country_code', '=', 'be'), ('country_ code', '=', 'fr') ]
```

connectedobjects.py	
1	<code>from openerp.osv import osv</code>
2	<code>from openerp.osv import fields</code>
3	<code>from openerp.tools.translate import _</code>
4	<code>import time</code>
5	
6	<code>listOS = [('win', 'Windows'), ('and', 'Android'), ('ios', 'iOS'), \</code>
7	<code> ('blb', 'BlackBerry'), ('sym', ' Symbian,')]]</code>
8	
9	<code>class connectedobjects_model(osv.osv):</code>
10	<code> """ The model of a connected objects """</code>
11	<code> _name = "connectedobjects.model"</code>
12	<code> _description = "The model of a connected objects"</code>
13	<code> _columns = {</code>
14	<code> 'name': fields.char('Model of connected objects', size=64, required=True),</code>
15	<code> 'operating_system': fields.selection(listOS, 'Operating system', required=True),</code>
16	<code> 'bluetooth': fields.boolean('Bluetooth connection', required=True),</code>
17	<code> 'wifi': fields.boolean('WiFi connection', required=True),</code>
18	<code> '3G': fields.boolean('3G connection', required=True),</code>
19	<code> '4G': fields.boolean('4G connection', required=True),</code>
20	<code> }</code>
21	<code> _sql_constraints = [</code>
22	<code> ('name', 'unique(name)', 'The name of a model must be unique')</code>
23	<code>]</code>
24	<code> _order = 'name asc'</code>
25	

connectedobjects_data.xml

```

1 <?xml version="1.0"?>
2 <openerp>
3   <data>
4     <record model="connectedobjects.model" id="connectedobjects_model_GTN8000">
5       <field name="name">Galaxy Note 10.1 (GT-N8000)</field>
6       <field name="operating_system">and</field>
7       <field name="bluetooth" eval="True"/>
8       <field name="wifi" eval="True"/>
9       <field name="3G" eval="False"/>
10      <field name="4G" eval="False"/>
11    </record>
12    <record model="connectedobjects.model" id="connectedobjects_model_GTN8010">
13      <field name="name">Galaxy Note 10.1 (GT-N8010)</field>
14      <field name="operating_system">and</field>
15      <field name="bluetooth" eval="True"/>
16      <field name="wifi" eval="True"/>
17      <field name="3G" eval="True"/>
18      <field name="4G" eval="False"/>
19    </record>
20    <record id="base.user_demo" model="res.users">
21      <field name="groups_id" eval="[ (4,ref('base.group_user'))]"/>
22    </record>
23  </data>
24 </openerp>

```

connectedobjects_view.xml

```

1 <?xml version="1.0"?>
2 <openerp>
3   <data>
4     <!-- ConnectedObjects Model: Form View -->
5     <record model="ir.ui.view" id="view_connectedobjects_model_form">
6       <field name="name">connectedobjects.model.form</field>
7       <field name="model">connectedobjects.model</field>
8       <field name="arch" type="xml">
9         <form string="Models of devices" version="7.0">
10          <label for="name"/><field name="name"/>
11          <label for="operating_system"/><field name="operating_system"/>
12          <label for="bluetooth"/><field name="bluetooth"/>
13          <label for="wifi"/><field name="wifi"/>
14          <label for="3G"/><field name="3G"/>
15          <label for="4G"/><field name="4G"/>
16        </form>
17      </field>
18    </record>
19
20     <!-- ConnectedObjects Model: Tree View -->
21     <record model="ir.ui.view" id="view_connectedobjects_model_tree">
22       <field name="name">connectedobjects.model.tree</field>
23       <field name="model">connectedobjects.model</field>
24       <field name="field_parent"></field>
25       <field name="arch" type="xml">
26         <tree string="Models of devices">
27           <field name="name"/>
28           <field name="operating_system"/>
29           <field name="bluetooth"/>
30           <field name="wifi"/>
31           <field name="3G"/>
32           <field name="4G"/>
33         </tree>
34       </field>
35     </record>
36
37     <!-- ConnectedObjects Model: Search Idea -->
38     <record model="ir.ui.view" id="view_connectedobjects_model_search">
39       <field name="name">connectedobjects.model.search</field>
40       <field name="model">connectedobjects.model</field>
41       <field name="arch" type="xml">
42         <search string="Models of devices">
43           <field name="name" string="Models"/>
44           <filter string="Windows OS" domain="[('operating_system','=', 'win'))" help="With
Windows OS"/>
45           <filter string="Not Windows OS" domain="[('operating_system','!=', 'win'))"
help="Without Windows OS"/>
46           <group expand="0" string="Group By...">
47             <filter string="Operating System" help="Operating System"
context="{ 'group_by': 'operating_system' }"/>
48         </search>

```

```

49     </search>
50     </field>
51 </record>
52
53 <!-- ConnectedObjects Model: Action -->
54 <record model="ir.actions.act_window" id="action_connectedobjects_model">
55     <field name="name">Models</field>
56     <field name="res_model">connectedobjects.model</field>
57     <field name="view_type">form</field>
58     <field name="view_mode">tree,form</field>
59     <field name="search_view_id" ref="view_connectedobjects_model_search"/>
60 </record>
61
62 <!-- Top menu item -->
63 <menuitem name="Connected Objects" id="base.menu_connectedobjects_root" sequence="120"
64 groups="base.group_user"/>
65
66 <!-- Menus sections -->
67 <menuitem name="Configuration" id="menu_connectedobjects_configuration"
68 parent="base.menu_connectedobjects_root" sequence="2"/>
69
70 <!-- Menus items -->
71 <menuitem name="Models" id="menu_model" parent="menu_connectedobjects_configuration"
72 action="action_connectedobjects_model" sequence="1"/>
73 </data>
74 </openerp>

```

create.php

```

1 <?php
2 require_once('XML/RPC.php'); // Include PEAR library for XML-RPC
3 require_once('login.inc.php'); // Include login function for connexion
4 require_once('vars.inc.php'); // Define connexion variables
5
6 $uid = login ($HOST, $PORT, $DB, $USER, $PASS, $DEBUG) ;
7
8 $client = new XML_RPC_Client('/xmlrpc/object', "http://$HOST:$PORT");
9 $client->setDebug($DEBUG);
10
11 $structVal = array( // Values to give to the fields of the new record
12     'name' => new XML_RPC_Value('iPad Air 4G', 'string'),
13     'operating_system' => new XML_RPC_Value('ios', 'string'),
14     'bluetooth' => new XML_RPC_Value(true, 'boolean'),
15     'wifi' => new XML_RPC_Value(true, 'boolean'),
16     '3G' => new XML_RPC_Value(false, 'boolean'),
17     '4G' => new XML_RPC_Value(true, 'boolean'),
18 );
19
20 $msg = new XML_RPC_Message('execute');
21 $msg->addParam(new XML_RPC_Value($DB, 'string'));
22 $msg->addParam(new XML_RPC_Value($uid, 'int'));
23 $msg->addParam(new XML_RPC_Value($PASS, 'string'));
24 $msg->addParam(new XML_RPC_Value('connectedobjects.model', 'string')); // Name of the relation
25 $msg->addParam(new XML_RPC_Value('create', 'string')); // The ORM method
26 $msg->addParam(new XML_RPC_Value($structVal, 'struct')); // Values of the new record
27
28 $resp = $client->send($msg);
29
30 if (!$resp)
31     die('Communication error: ' . $client->errstr);
32 else if ($resp->faultCode()) { // FaultCode seems to be never set by OpenERP 7.0
33     echo 'Fault Code: ' . $resp->faultCode() . "\r\n";
34     echo 'Fault Reason: ' . $resp->faultString() . "\r\n";
35     exit(1);
36 }
37
38 $val = $resp->value() ;
39
40 if ($val->kindOf() == 'struct') {
41     $struct = XML_RPC_decode($val) ;
42     if (isset($struct['faultCode'])) // Trick to catch fault code
43         echo '<p>FAILURE RESPONSE: ' . $struct['faultCode'] . "</p>\r\n" ;
44 }
45 else { // kindOf == 'scalar'
46     // Show id of the new record
47     $id = $val->scalarval() ;
48     echo "<p>New element created (id:$id)</p>\r\n";
49 }
50 ?>

```