DIAGRAM SOFTWARE

UNIVERSITAT POLITÈCNICA DE VALÈNCIA

PROGRAMACIÓN DE OPENERP/ODOO

1.4 HERENCIAS

José Zambudio Bernabeu
jose.zambudio@diagram.net

NubeaERP global cloud solutions

DIAGRAM SOFTWARE

UNIVERSITAT POLITÈCNICA DE VALÈNCIA

ÍNDICE

- Model Inheritance
- View Inheritance

MODEL INHERITANCE

- Extender un modelo ya existente.
  - Creamos una nueva clase en la que indicamos el modelo que vamos a extender/modificar, con:
    - _inherit = 'module.model_name'
  - **¡Atención!** el valor de _inherit puede ser tanto un string como una lista de strings.
    - Esto nos nos concatenaría varios modelos en uno mismo.
  - **¡Atención!** El orden de invocación del método padre mediante super() viene indicado por el orden de la lista de dependencias del manifest file (__openerp__.py)

MODEL INHERITANCE

- Extender un modelo ya existente.
  - 3 tipos de herencia según el valor de _name y _inherit:
    - **_name == _inherit || !_name && _inherit**
      (Class inheritance)
      - Extendemos el modelo del que heredamos añadiendo/modificando métodos, parámetros, fields…etc.
      - Los registros de este nuevo modelo son visibles por la vistas del modelo padre.
      - Ejemplo en openacademy…
    - **_name && !_inherit**
      - En caso de especificar el mismo _name de otro modelo y no indicar _inherit estaríamos redefiniendo este.

MODEL INHERITANCE

- Extender un modelo ya existente.
  - Diferentes tipos de herencia según el valor de _name y _inherit:
    - **_name != _inherit**
      (inheritance by prototyping)
      - Creamos otro modelo con nombre del valor de _name el cuál tendrá todos los fields y métodos del modelo padre junto con las nuevas funcionalidades.
      - Los registros de este nuevo modelo no son visibles por las vistas del modelo padre.

MODEL INHERITANCE

- Extender un modelo ya existente.
  - **_name && _inherits={**
    'model_a': 'field_model_a_id',
    'model_b': 'field_model_b_id',
    'model_c': 'field_model_c_id',
    }
    (polymorphic)
    - Este tipo de herencia añade a nuestro nuevo modelo todos los fields de los modelos padre.
    - Los registros nuevos de nuestro nuevo modelo introducirán en los modelos padres los fields correspondientes, y los fields del nuevo modelo en una tabla nueva.

VIEW INHERITANCE

- Realizamos un registro de una vista normal:
  - name: Por convención mismo valor que id, substituyendo '_' por '.'
  - model: Modelo de la vista que queremos heredar.
  - **inherit_id**: id de la vista a modificar. Si esta es de otro módulo indicamos el nombre del módulo seguido del un '.' y el id
    - <field name="inherit_id" ref="modulo.id_vista_a_heredar"/>
- Dentro del field "arch" introduciremos aquello que queramos modificar/añadir/eliminar. El engine buscará el elemento en la vista padre y realizará la modificación que le estamos indicando mediante el atributo "position":
  - position="inside" (default)
    - Inserta contenido dentro del tag.
  - position="after"
    - Inserta contenido después del tag.
  - position="before"
    - Inserta contenido antes del tag.
  - position="replace"
    - Reemplazamos el tag por el valor. En caso de no especificar valor, estaríamos borrando dicho elemento.
  - position="attributes"
    - Modificamos los atributos del tag indicado.

VIEW INHERITANCE

- position="inside" (default)
  - addons/product/partner_view.xml

```
<record id="view_partner_property_form" model="ir.ui.view">
    <field name="name">res.partner.product.property.form.inherit</field>
    <field name="model">res.partner</field>
    <field name="inherit_id" ref="base.view_partner_form"/>
    <field name="arch" type="xml">
        <page string="Sales &amp; Purchases" position="inside">
            <group>
                <group name="pricelists" attrs="{'invisible': [('is_company','=',
                False),('parent_id','!=',False)]}">
                    <field name="property_product_pricelist" groups="product.group_sale_pricelist"/>
                </group>
                <div name="parent_pricelists" groups="product.group_sale_pricelist" attrs="
                {'invisible': ['|',('is_company','=',True),('parent_id','=',False)]}">
                    <p>Pricelists are managed on <button name="open_commercial_entity" type="object"
                    string="the parent company" class="oe_link"/></p>
                </div>
            </group>
        </page>
    </field>
</record>
```

VIEW INHERITANCE

- XPATH
  - A veces la selección del campo a modificar es más complicado debido a la complejidad de la vista. Por ejemplo, cuando un mismo campo es utilizado en varios lugares.
  - En estos casos utilizamos xpath para identificar donde se deben situar los cambios.
  - Formato:
    - expr="//TAG[@SELECTOR="VALUE"]
      - expr="//field[@name='name']"
    - position="" <- Mismos valores que la transparencia anterior.

VIEW INHERITANCE

- XPATH

```
<record id="view_bank_statement_form_journal_items" model="ir.ui.view">
    <field name="name">account.bank.statement.journal.items.form.inherit</field>
    <field name="model">account.bank.statement</field>
    <field name="inherit_id" ref="view_bank_statement_form"/>
    <field name="arch" type="xml">
        <xpath expr="//div[@name='import_buttons']" position="inside">
            <button name="button_journal_entries"
                    string="Journal Items" type="object"
                    attrs="{'invisible':[('state','!=','confirm')]}"/>
        </xpath>
    </field>
</record>
```