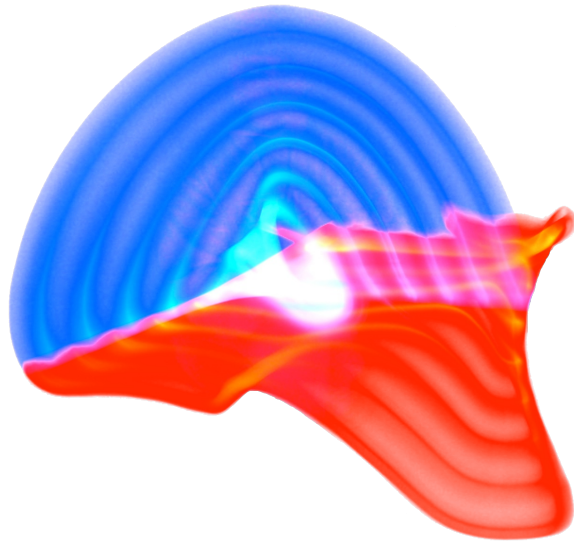


Lightning Talk auf der DZUG-Tagung 2008 Saarbrücken



devilstick project

Jens Klein <jens@bluedynamics.com>

Geschichtliches

- **ArchGenXML** – Code Generator
stabil, alt, fokussiert Plone/Archetypes,
monolythisch, schwer erweiterbar.
- **Genesis** – Model-Transformation,
zu komplex, schwer verständlich, zu
wissenschaftlicher Ansatz, nicht pragmatisch.
- **Devilstick** – Model-Runtime
neu, nicht stabil (lauffähiger Prototyp), einfache
Erweiterbarkeit, leicht einzusetzen, basiert auf
bekannten Pattern.

Haupt-Teile

- Model-Runtime
- Model-Import/Export
- DAO-API (Data-Access-Objects)
- automatische Formular-Generierung für die Datenstruktur.

Model Rulez

- Modell als Strukturdefinition nur Daten, kein Code,
- Modell kann zur Laufzeit verändert werden,
- Meta-Modell definiert Modell,
- Meta-Meta-Modelling (cornerstone.model) erlaubt verschiedene Meta-Sprachen,
- Devilstick stellt ein spezielles Meta-Modell zur Verfügung und eine Runtime zum Datenmanagement.

```
<?xml version="1.0"?>
<dao xmlns="http://devilstickproject.net/ns/dao"
      xmlns:widget="http://devilstickproject.net/ns/widget"
      xmlns:permission="http://devilstickproject.net/ns/permission"
      xmlns:validation="http://devilstickproject.net/ns/validation">
  < cage id="foo">
    <storage>zodb</storage>
    <permission:view>View</permission:view>
    <permission:add>Add portal content</permission:add>
    <permission:modify>Modify portal content</permission:modify>
    <permission:delete>Delete objects</permission:delete>
    <widget:label>My Cage</widget:label>
    <atom id="body">
      <type>devilstick.Text</type>
      <widget:label>Body-Text</widget:label>
      <widget:description>Enter a long text</widget:description>
      <widget:type>richtext</widget:type>
    </atom>
    <molecule id="comments">
      <cardinality for="reply" max="*" />
      <molecule id="reply">
        <cardinality for="reply" max="*" />
        <atom id="title">
          <type>devilstick.String</type>
        </atom>
        <atom id="replytext">
          <type>devilstick.Text</type>
          <validation:required name="notempty" />
          <validation:sufficient name="notempty" />
        </atom>
      </molecule>
    </molecule>
  </cage>
</dao>
```

DAO-API

Data-Access-Objects

- Baum-Struktur,
- Cage (Root-Container, sonst wie Molecule),
- Molecule (Container für Molecule und Atome),
- Atom (Data-Unit), via Zope-Schema definiert.
- generische API an Zope-Container API angelehnt,

Beispiel Modelle

- Modelle werden importiert (in die interne Modell-Sprache),
- XML als Modell-Input, direkte Abbildung der internen Sprache,
- LDAP-Schemas werden dynamisch importiert,
- XMI (UML-Austauschformat der OMG) wird importiert,
- Python-Code als Modell-Definition auch möglich (z.B. aus dicts, lists, schemas).