

## Zope sovelluskehittäjän näkökulmasta

Arto Teräs (ajt@iki.fi)  
12. huhtikuuta 2002

Perustuu suurelta osin Zope Developer's Guiden ja www-sivujen <http://www.zope.org> materiaaliin. Sisältöä saa levittää muokkaamattomana tai muokattuna Open Publication License v. 1.0 -lisenssin mukaisesti, ilman valinnaisia rajoituksia (License Options). Lisätietoja: <http://opencontent.org/openpub/>

Zope — Arto Teräs 12.4.2002 – p.1/16

## Zope, yleistä

- Web-sivustojen kehitystyökalu ja sovelluspalvelin
- Kilpailijoita Midgard, (Open)ACS, Java-pohjaiset sovelluspalvelimet, PHP/Perl/ASP
- Integroitu kokonaisuus: oma www-palvelin, oliotietokanta ja hallintaliittymä
- Rakentuu Python-kielen ympärille, suorituskyvyn kannalta kriittisimmät osat toteutettu C:llä
- Paljon käyttäjien tekemiä laajennuksia
- <http://www.zope.org/>

Zope — Arto Teräs 12.4.2002 – p.2/16

## Hiukan historiaa

- Jim Fulton / Digital Creations 1996
- Julkinen (open source) versio 11/1998
- Zope 2.0 9/1999
- Digital Creations ⇒ Zope Corporation 7/2001
- Lisenssin vaihto omasta copyleft-mallista BSD-tyyppiseksi 11/2001
- Uusin vakaa versio 2.5.0 1/2002
- Tällä hetkellä noin 70 aktiivista kehittäjää ja yli 20 000 käyttäjää (Zope Corporationin arvio)

Zope — Arto Teräs 12.4.2002 – p.3/16

## Liiketoiminta ja kehitysmalli

- Zope Corporation rakentaa www-sivustoja asiakkaille ja koordinoi edelleen ydinkehitystä, mutta ei tee läheskään kaikkea itse
- Kehitysmalli sekoitus katedraalia ja basaaria — uudet ominaisuudet pitää perustella ja niistä tehdään mm. vaatimusmäärittelyt ja suunnitelmat, mutta kaikki on koko ajan avointa
- Kolmansien osapuolien laajennusosia (Zope Products) useita satoja
- Myös suurin osa dokumentaatiosta vapaan lisenssin alla

Zope — Arto Teräs 12.4.2002 – p.4/16

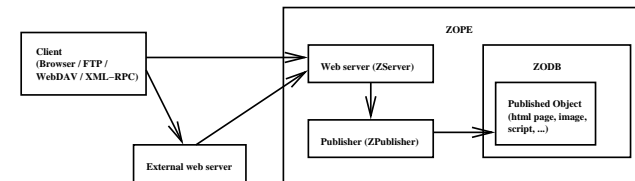
## Python

- Oliopohjainen korkean tason skriptikieli
- Jo yli 10 vuotta vanha, mutta tullut suosituksi vasta muutaman viime vuoden aikana
- Toimii lähes kaikissa ympäristöissä (Unix, Windows, Mac, Amiga...)
- Pyrkimys eleganttiin syntaksiin, kontrollirakenteiden alku ja loppu määritellään indentaatiolla
- Kilpailee erityisesti Perl'n kanssa, mutta eri filosofialla: *“There should be one — and preferably only one — obvious way to do it.”*

Zope — Arto Teräs 12.4.2002 — p.5/16

## Rakenne

- Kaikki elementit olioita, oma oliotietokanta ZODB
- Mihin tahansa voidaan viitata URLilla
- Liittymät ulkoisiin tietokantoihin



Zope — Arto Teräs 12.4.2002 — p.6/16

## Asennus ja vaatimukset

- Yleisimmät Unixit ja Windows-ympäristö tuettuja
- Binääri- tai sorsapaketti, kääntäminen nopeaa (AMD Duron 650 MHz alle 2 min)
- Serverin muistinkulutus n. 10 MB yhdellä käyttäjällä
- Suorituskyky riippuu huomattavasti sivuston suunnittelusta, tyyppillisesti selvästi hitaampi kuin esim. Apache staattisilla sivuilla mutta nopeampi kuin CGI-skriptit

Zope — Arto Teräs 12.4.2002 — p.7/16

## Sivujen suunnittelu

- Hierarkinen puumalli, käyttää perintää
- Oma selainpohjainen käyttöliittymä, mutta editointi onnistuu myös WebDAV:n tai FTP:n kautta (demo)
- Useita osin päällekkäin meneviä tasoja: template-pohjat, DTML-kieli, “hiekkalaatikossa” ajettavat skriptit, Python-pohjaiset moduulit (Zope Products)
- Eri käyttäjien oikeudet yksityiskohtaisesti säädettävissä
- Sisältöä ja toiminnallisuutta ei kunnolla erotettu toisistaan

Zope — Arto Teräs 12.4.2002 — p.8/16

## Zope Products, 1(2)

- Laajennusosia, vastaavat Zope-maailmassa Perlin CPAN-moduulikokoelmaa
- Verkosta haettavissa useita satoja erilaisia, suurin osa vapaita, laajuus ja laatu vaihtelee
- Tyypillisesti lisäävät uudentyyppisiä elementtejä sivukehittäjän käyttöön, esim. valokuva-albumi-moduuli (demo)
- Kirjoitetaan Pythonilla, Zopen API tarjoaa erilaisia palveluja ja asettaa joitakin vaatimuksia

Zope — Arto Teräs 12.4.2002 — p.9/16

## Zope Products, 2(2)

- Zope-yleistoiminnallisuus (cut'n'paste, tallennus tietokantaan) saadaan perinnällä perusluokista
- Käyttäjän näkymän lisäksi toteutettava hallintaliittymä
- Olioiden instanssit yleensä pysyviä (persistent), tila tallentuu automaattisesti ZODB:hen
- Moduulin päivitys vaikuttaa myös jo luotuihin instansseihin — rajapintojen taaksepäin yhteensopivuus tärkeää
- ZODB toimii transaktiopohjaisesti, jonkun mennessä pieleen riittää että heittää poikkeuksen eikä tarvitse siivota

Zope — Arto Teräs 12.4.2002 — p.10/16

## Debuggaus

- Pythonin mukana tuleva debuggeri muistuttaa gdb:tä

Erilaisia vaihtoehtoja:

- Kutsujen (URL-viittausten parametreineen) simulointi komentoriviltä
- Debuggerin käynnistyskutsujen asettaminen keskelle Python-koodia ja Zopen ajaminen debug-tilassa — sivustoa voi käyttää selaimella ja vaihtaa terminaaliiin kun suoritus pysähtyy breakpointiin (demo)
- ZEO (Zope Enterprise Objects) mahdollistaa yhden säikeen debuggauksen samalla kun sivusto palvelee muita käyttäjiä

Zope — Arto Teräs 12.4.2002 — p.11/16

## Koodin laajuus ja jako

- Noin 45 000 riviä C- ja 53 000 riviä Python-koodia (Zope 2.5.1b1 + mukana toimitettavat perusobjektit)
- Python suunniteltu helposti liitettäväksi C:hen, suorituskyyvyn kannalta kriittisimmät tietorakenteet toteutettu C:llä
- Mahdollisuus hakea uusien kehitysversio cvs:stä
- Koodin hakemistopuu rakenteeltaan hiukan sekava
- Mukana testikoodia (unit tests), josta osa ei kuitenkaan toimi

Zope — Arto Teräs 12.4.2002 — p.12/16

## Dokumentaatio

- Käyttö ja laajennusosien kehittämisessä tarvittavat API:t hyvin dokumentoitu (The Zope Book ja Developer's Guide), kehittynyt huomattavasti viimeisen vuoden aikana
- Lisäksi runsaasti sekalaisia HOWTO-tyyppisiä artikkeleita, joista kuitenkin osa vanhentuneita
- Postituslistat
- Koodi ja sisäiset API:t varsin huonosti dokumentoitu, ei luokkakaavioita

Zope — Arto Teräs 12.4.2002 – p.13/16

## Koodiesimerkki: ZODB/DB.py (open)

```
if type(version) is not StringType:
    raise POSException.Unimplemented, 'temporary versions'

self._a()
try:

    if transaction is not None:
        connections=transaction._connections
        if connections:
            if connections.has_key(version) and not temporary:
                return connections[version]
        else:
            transaction._connections=connections={}
    transaction=transaction._connections
```

Zope — Arto Teräs 12.4.2002 – p.14/16

## Koodiesimerkki: Photo/PhotoFolder.py

```
security.declareProtected('Manage properties', 'manage_addDisplay')
def manage_addDisplay(self, id, width, height, REQUEST=None):
    """Add display."""
    d = self._displays
    d[id] = (width, height)
    self._displays = d
    # Update contained Photo objects
    for photo in self.objectValues(['Photo']):
        try: photo.manage_addDisplay(id, width, height)
        except: pass
    if REQUEST is not None:
        return self.manage_editDisplaysForm(REQUEST,
            manage_tabs_message='Display added.')
```

Zope — Arto Teräs 12.4.2002 – p.15/16

## Tulevaisuus

- Zope 3.0:ssa kirjoitettu suuria osia uudestaan
- Tavoitteena erottaa logiikka ja sisältö paremmin toisistaan
- Toisena tavoitteena helpottaa edelleen laajennusosien (Zope Products kirjoittamista)
- Laaja Content Management Framework -laajennusosa (<http://cmf.zope.org>) antaa esimakua tulevasta

Zope — Arto Teräs 12.4.2002 – p.16/16