Una piattaforma per il map sharing ed i mediascape

PyCon Italia Due

Argomenti del Talk/1

- Sistemi per la gestione della cartografia
  - Binding e applicazioni disponibili per Python
- Limiti dei GIS nella costruzione di sistemi documentali con annotazioni cartografiche
- Presentazione di una piattaforma SOA *Python based* per la gestione di documenti cartografici e la condivisione di mappe

Argomenti del Talk/2

- La contestualizzazione delle aree archeologiche: sperimentazione con Python su dispositivi Windows Mobile, per la distribuzione di mappe e contenuti multimediali
  - Problematiche connesse al linguaggio e la piattaforma
  - GUI
  - Aspetti ‘energetici’ e peculiarità dell’Hardware

Python e la cartografia

- Python è ‘presente’ nel settore della cartografia sia tra i sistemi proprietari sia tra quelli Open Source
  - ESRI ha rilasciato il binding per consentire lo sviluppo di script per la sua piattaforma ArcGIS
  - Mapscript è il binding Python al celebre motore di Web GIS
- Esistono altri progetti in cui Python è di riferimento nel settore dei GIS
  - *TileCache*: WMS-C scritto in Python con l’ausilio di *memorycache*
  - *Mapnik*: piattaforma di Web GIS C++/Python
  - *PrimaGIS*: integrazione in Plone, ma fermo

Gestione docmentale e GIS

- La maggior parte dei sistemi GIS sono realizzati nell'ottica della visualizzazione a video di entità geometriche
- Il paradigma "dati geometrici + dati" che il GIS persegue può ritenersi abbondantemente superato
  - Con l'affermazione del web, come unica piattaforma distribuita per l'accesso e la condivisione di dati, continuare a ragionare in termini di tabelle è anacronistico
- Oggi il dato non è più solo 'valore' ma è anche struttura e semantica
  - Semantic Web
  - *Microformats* o tagging

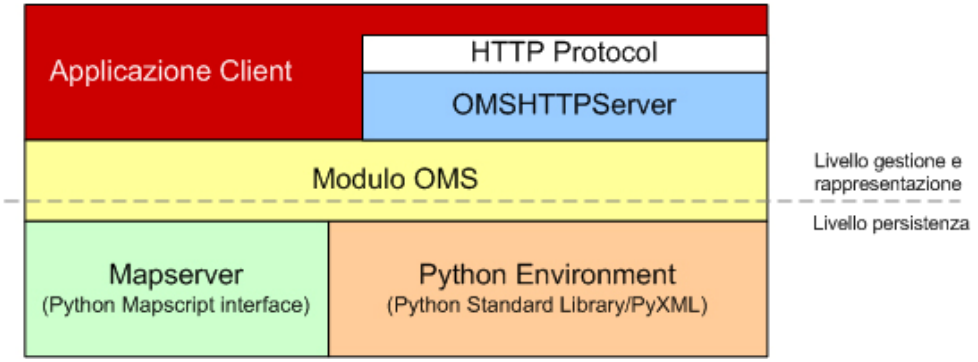
Continua….

Gestione docmentale e GIS

- Il passaggio successivo si ottiene nel considerare una mappa un documento dotato di una sua rappresentazione, semantica ed interpretazione che è eventualmente associato ad un altro documento
  - Si parla di *mashup* di servizi e documenti
    - *Lascia che siano altri a trattare quello che sanno fare descrivendo i tuoi dati*
    - *Dicci qual è il problema, non la soluzione*
  - In generale è necessario avere architetture a servizi (ma che siano TRUE)
- Esempi: Google Map (KML)
  - Ma orientato alla cartografia generalista
- Yahoo Pipes

Il modulo Octapy MapServer

- Modulo realizzato in Python per la rappresentazione strutturata di mappe
  - Formati multipli di interscambio (XML, RDF)
  - Integrazione diretta con la piattaforma Octapy (CMS Plone)
  - Architettura a servizi



`from __future__ import openmapper`

- Progetto per la costruzione di una piattaforma di Web GIS realizzata in Python
  - SOA Oriented
  - 100% Open Source
  - Gestione documentale cartografica
  - Shared Map Server
  - KML compatibile

openmapper

La contestualizzazione delle aree archeologiche

- Le aree archeologiche sono la testimonianza diretta del nostro passato
  - Ma ‘decontestualizzate’ perché dalle strutture *in situ* sono state rimossi tutti gli oggetti rinvenuti
- La maggior parte degli oggetti rinvenuti nelle aree archeologiche sono oggi disponibili nei Musei più disparati o nei depositi delle soprintendenze
- La cartografia rappresenta un eccezionale strumento di contestualizzazione e correlazione di informazioni

Il progetto Rione Terra in Rete

- Progetto di contestualizzazione dei reperti rinvenuti nell'area archeologica del Rione Terra di Pozzuoli (Campi Flegrei)
  - Estrapolazione del posizionamento di decine di reperti a partire da cartografie tecniche georiferite
  - Generazione di cartografie tematiche a partire dai dati di scavo e dai dati di fruizione
  - Contestualizzazione nei rispettivi Musei di esposizione

Dispositivi mobile e cartografia

- La gran parte dei palmari e cellulari di ultima generazione integra dispositivi GPS
- Tuttavia, i software cartografici a corredo mal si prestano alla contestualizzazione
  - Cartografia ‘precaricata’ e specializzata
  - API dei software di base fortemente dedicata
- Grazie alla diffusione delle reti WI-FI a basso costo di gestione è possibile ipotizzare una futura remotizzazione della cartografia
  - Google Maps per i dispositivi mobile
  - Nokia Maps
    - Ma ancora una volta cartografie specializzate e/o a scale di dettagli eccessivamente alte

I *mediascape* di HP

- Progetto per la costruzione di una piattaforma open per la distribuzione di contenuti multimediali connessi con il posizionamento geografico
- La piattaforma consta di due parti principali
  - Un tool di authoring che genera il mediascape annotandolo con una coordinata
  - Un driver su Windows Mobile che esegue il contenuto multimediale quando il dispositivo si trova nell'area prefissata
  - Testato solo su alcuni dispositivi HP

Python su Windows Mobile

- Esiste un porting relativamente supportato di CPython per la piattaforma Windows Mobile
  - Windows Mobile 2003, ma compatibile con 2005 e 6.x
  - Implementazione pressocché completa di tutto il core di Python (Standard Library compresa)
  - `http://pythonce.sf.net`

GUI toolkit disponibili/1

- VensterCE: porting su CE del wrapper *ctypes* di Venster
  - Pro: performante, accesso all'API Win32 diretta, migliore integrazione nella piattaforma (finestre *fullscreen*, migliore interazione con i device di input)
  - Contro: molto ma molto poco pythonica, API estremamente prolissa, progetto interrotto da molto tempo, pochissima documentazione (anche se soccorre l'SDK Microsoft)

GUI toolkit disponibili/2

- PocketPyGui: dall'autore di Venster CE un wrapper molto più pythonico della GDI
  - Pro: codice drasticamente ridotto, libreria molto completa e ben scritta sin dalla primissima versione, molto performante, controllo HTML, l'installer visuale costituisce un ottimo esempio di come crare un'applicazione redistribuibile
  - Contro: continua ad essere necessario conoscere la Win32 API, immagini gestite direttamente tramite l'API GDI e supporto solo al BMP

Confronto Venster / PPyGUI

```
from venster.ce import *

class MyWindow(CeMainWindow):
    _window_title_ = u"Benvenuti al PyCon Due"

    @msg_handler(WM_PAINT)
    def OnPaint(self, event):
        ps = PAINTSTRUCT()
        hdc = self.BeginPaint(ps)
        rc = self.clientRect
        msg = u"Ciao a tutti"
        DrawText(hdc, msg, len(msg), byref(rc), 1|4)
        self.EndPaint(ps)

def main():
    w = MyWindow()
    app = Application()
    app.Run()
```

Continua….

Confronto Venster / PPyGUI

```
import ppygui as gui

class MainFrame(gui.CeFrame):
    def __init__(self):
        gui.CeFrame.__init__(self, title="Benvenuti \
                                          al PyCon Due")
        self.label = gui.Label(self, text="Ciao a tutti")

if __name__ == '__main__':
   app = gui.Application(MainFrame())
   app.run()
```

GUI toolkit disponibili/3

- wxPython per PythonCE: porting del binding Python al framework wxWidgets
  - Pro: molto completa, possibilità di porting su più piattaforme della stessa applicazione (vantaggio teorico), libreria ampiamente documentata e ricca di esempi, porting delle PIL
  - Contro: disponibile precompilata solo per Python CE 2.4.x su architettura ARM, purtroppo non più supportata dall'autore (il motivo sarà chiaro fra poco…..)

GUI toolkit disponibili/4

- Tkinter per PythonCE: nel porting di Python è stato effettuato anche quello di Tk con un supporto completo alla libreria
  - Pro: pochissime modifiche al codice rispetto a Windows standard (ottimo per la fase di sviluppo), ampiamente documentata, disponibilità di widget custom
  - Contro: scarsa integrazione con le peculiarità del sistema (impossibile avere finestre fullscreen, disabilitare la virtual keyboard), Tk o si ama o si odia, prestazioni generalmente inferiori rispetto a PPyGui

I *limit* dei dispositivi mobile

- Uno dei problemi con cui ci si scontra sin da subito è la dimensione estremamente ridotta della RAM e l'assenza di memoria virtuale
  - Questo è il motivo per cui il porting di wxPython è stato dismesso
- Forti personalizzazioni dei singoli *vendor* possono rendere le applicazioni incompatibili
- Non esiste uno standard consolidato per l'accesso ad alcune risorse hardware, come stack BT e dispositivi GPS di bordo
  - Fortunatamente, le ultime versioni dello Stack BT Broadcom permetto l'accesso via seriale
  - Molti dispositivi GPS sono rimappati via seriale

Continua….

I *limit* dei dispositivi mobile

- Il modulo `ceserial` permette di gestire la porta seriale da PythonCE
  - API conforme al modulo `pyserial`
  - Con qualche aggiustamento minimo si riesce a farlo funzionare con le ultime versioni dello stack Broadcom (2.x o superiori)
  - `http://www.problemboard.com/dl/ceserial.zip`
- Alcuni dispositivi GPS di bordo sono configurati per l'uso 'automobilistico', anche nel caso di telefonini
  - Insensibilità a velocità < 5km/h o spostamenti al disotto dei 20mt
  - La gran parte dei chip SiRF III consente l'abilitazione della modalità *pedestrians*, per mezzo di codici di controllo da scrivere sulla seriale all'atto dell'apertura
  - HP ha rilasciato un *service pack* per i suoi dispsitivi della serie rx57xx o rx59xx

Tips ‘n tricks

- Usare gli *xda tools* per l’accesso al dispositivo via RAPI
  - `http://www.xs4all.nl/~itsme/projects/xda/tools.html`
- Fare sempre uso di un task manager avanzato se la semantica del bottone “X” non è di chiusura (`tMan`, `HandySwitcher`)
- Usare l’estensione `.pyw` per non visualizzare la consolle di Python
  - Ma attenzione che l’installer sbaglia ad aggiungere key nel registro:
    - `/nopcshell`
- Con TK spesso le immagini GIF risultano visualizzate in maniera corrotta: generare un immagine con un ‘padding maggiore’ e mostrarne solo una ‘sotto area’

Continua….

Tips ‘n tricks

- Ottimizzare sempre il codice dal punto di vista dell’allocazione della memoria e ove possibile ricorrere a codice ‘a mano’ più ottimizzato
  - Ad esempio, SAX in luogo di DOM per documenti XML molto complessi e di grosse dimensioni
- Nell’eseguire operazioni in background cicliche (ad esempio, un thread per la lettura dei dati dal GPS) adottare politiche di sleeping per aumentare la durata della batteria
  - Nel caso di applicazioni multimediali connesse con lo spostamento a piedi è perfettamente inutile leggere costantemente dal GPS, ed è conveniente intervallare il codice con delle chiamate alla funzione time.sleep()
- Se si sviluppa codice di wrapper per accedere a particolari risorse hardware e non si fa ricorso ai *ctypes*, ricordarsi di rilasciare il GIL

Demo

http://www.lab32.org/blogais/?page_id=31