OSM-in-a-box –
A Ready-Made Highly Configurable Map Server

Prof. Stefan Keller
Computer Science Department
University of Applied Sciences Rapperswil (CH)
www.gis.hsr.ch
Contents

- Motivation
- The osm2gis converter tool
- The Schema Mapping Problem
- DB Schema example and tests
- GeoServer configuration (pre-configured too)
- Showcases and Conclusion
Motivation

- Map server for mobile apps.
- Web mapping site for an organisation with
  - Maps with own design
  - Reliable web services
- There are many OSM APIs and tools available...
  - e.g. XAPI returns XML format for a given region of the globe, can also filter - So let’s install it...
Some DB Software in OSM

- PostgreSQL dbms
  - Main OSM db

- osmosis
  - Enhancements for PostGIS

- osm2pgsql
  - Mapnik specific
The Schema Mapping Problem

- What we’d expect: A DB/GIS related information model
  - incl. geom. types
  - Point
  - Polyline
  - Polygon

- What we have: The 'famous' OSM information model
  - Street
    + name
    + geom
    + type
  - Nodes (Point)
  - Ways (Polyline)
  - Relations
  - Tags

...
Approach / Solution

- Write own converter/updater
  - to map schema from OSM to GIS
  - to synchronize with OSM main DB

- osm2gis Tool
  - Configurable with consistency check
  - Initial data import
  - Keep local database up-to-date by continuously importing the differential update files

- Use GeoServer version 2.0
Overview

GIS Community
- Tool (WFS)
- Website (OpenLayers)

OSM-in-a-Box
- WFS/WMS
- GeoServer
- GeoWebCache
- PostgreSQL/PostGIS Database
- osm2gis

OSM Community
- OSM API
- OSM XAPI
- Mapnik Tiles

Planet File
Differential File
The osm2gis Tool

- Creates target DB, checks config. consistency and syncs data from OSM server to own DB
The osm2gis Tool

- Pure Java

- Syncs OSM data: Handles the 3 diff. file types:
  - Minutely (Replicate)
  - Hourly (Replicate)
  - Daily

- Update scheduler
  - Uses Quartz library for scheduling jobs
  - jobs download differential files (xml)
The osm2gis Tool

- Parses config. XML
- Generates DDL
- Optional: updates DDL if needed and does checks
- Parses OSM xml data files (planet, diff) and stores it in DB
  - Maps tables, attributes and attribute types
  - Special handling of POLYGON type:
    - 'closed ways'
    - Also converts relations to polygons (type=multipolygon and type=boundary > polygon)
- Handles relation objects according to their tags
- A buffer holds the parsed OSM entities and passes them to the database layer once it’s full
osm2gis Configuration

Structure of Schema Mapping File

- Definition of destination schema
- Tables
- Join tables
- Views (optional)
- Indexes
- User defined data (optional, i.e. more optimization)
- Source DB Schema to Destination Schema Mapping

Example:
<dst_schema_def>…</dst_schema_def>
<src_to_dst_mappings>…</src_to_dst_mappings>
<mapping type="point">
  <and_ed_conditions>
    <tag k="amenity" v="motel" />
  </and_ed_conditions>
  <dst_table name="poi" />
  <dst_columns>
    <column name="osm_id" value="%attribute_id%" />
    <column name="lastchange" value="%attribute_timestamp%" />
    <column name="type" value="motel" />
    <column name="name" value="%tag_name%" />
    <column name="keyvalue" value="%tags_all%" />
    <column name="geom" value="%geom%" />
  </dst_columns>
</mapping>
Ex. command: % osm2gis --xml2ddl

```xml
<dst_table_def name="gisentity">
    <dst_column name="osm_id" type="bigint" not-null="true" />
    <dst_column name="lastchange" type="TIMESTAMP" not-null="false" />
    <dst_column name="name" type="VARCHAR(255)" not-null="false" />
    <dst_column name="keyvalue" type="hstore" />
</dst_table_def>

<dst_table_def name="indoor" inherits="gisentity">
    <dst_column name="id" type="serial" primary-key="true" />
    <dst_column name="geom" type="geometry(4326,'POINT',2)" />
    <dst_column name="description" type="text" />
    <dst_column name="level" type="VARCHAR(20)" />
    <dst_column name="website" type="VARCHAR(255)" />
    <dst_column name="wikipedia" type="VARCHAR(255)" />
    <dst_column name="image_url" type="VARCHAR(255)" />
    <dst_column name="video_url" type="VARCHAR(255)" />
    <dst_column name="audio_url" type="VARCHAR(255)" />
</dst_table_def>

CREATE TABLE gisentity (  
osm_id bigint NOT NULL,  
lastchange TIMESTAMP,  
type VARCHAR(255),  
name VARCHAR(255),  
keyvalue hstore  
);

CREATE TABLE indoor (  
id serial,  
description text,  
level VARCHAR(20),  
website VARCHAR(255),  
wikipedia VARCHAR(255),  
image_url VARCHAR(255),  
video_url VARCHAR(255),  
audio_url VARCHAR(255),  
PRIMARY KEY (id)
) INHERITS (gisentity);
```
Diff Update Specialities

- Missing data is downloaded via the OSM API
- Simple changes in the diff file can have heavy changes on the database
  - E.g. Node that was previously a police station and its tags are changed to a place needs to be deleted from the "poi" and inserted into the "place" table
osm2gis Configuration

- Consistency checks
  - Schema to DB
  - Schema to Mapping
  - Mapping to Styles
Default Config. and Tests

- **Target DB Schema configured for topogr. base map:**
  - (based on “OpenStreetMap Data in Standard GIS Formats Draft 0.2”, Jochen Topf, Geofabrik 2008-09-04)
  - Places (inc. indoor places), POIs, Railway Stations
  - Boundaries, Roads, Railways, Waterways
  - Landuse, Water

- **Tests:**
  - Software Eng.: Continuous integration tests (of course)
  - Data : OSM import of Switzerland, Li. (->[->]), Germany(!)
  - Apps.:
    - Showcase/Demo Web Site (see below)
    - Augmented Reality for indoor navigation (Android, WFS)
GeoServer 2.0

- Web Services

(De) Service Capabilities

WCS
- 1.0.0
- 1.1.1

WFS
- 1.0.0
- 1.1.0

WMS
- 1.1.1

http://sinv-56029.edu.hsr.ch/geoserver/ows?
service=wms&version=1.1.1&request=GetCapabilities
Showcase

Switzerland delivered by ...
Showcase

Map compare
Deliveries of Version 1 (http://dev.ifs.hsr.ch/osminabox/)

- Osm2gis software
- Pre-configured PostGIS schema (base map)
- Pre-configured GeoServer 2.0 styles (*.SLD)
- Installation script

⇒ „Ready made“ ...
  - Base map
  - Standardized APIs (WMS, WFS, WCS, ...)
⇒ ... „and Highly Customizable“
  - Own ‚real‘ GIS DB schema and own map design
⇒ A „mirror“reliable (as far as you can) and even offline
Conclusions

- Experiences
  - Don‘t underestimate relation handling and diff update
  - Troubles to find techn. information about OSM
  - There‘s an URGENT need for a POLYGON type!

- Future work
  - (Even) better map styles
  - Can be used as a tool for generating benchmark data (see also „HSR Texas Geo Database Benchmark“)
  - Usage as GeoServer tool?
Questions?

http://dev.ifs.hsr.ch/osminabox
Stefan Keller, sfkeller@hsr.ch