Clarity: A New Environment For Generalisation Using Agents, Java, XML And Topology

Paul Hardy, Principal Consultant (paul.hardy@laser-scan.com), Melanie Hayles, Senior Software Engineer, Patrick Revell, Senior Software Engineer

http://www.laser-scan.com
April 2003
Contents

- Introduction to Clarity
- Agents & active objects
- Java & XML
- Summary

- Read the online paper for more details!
  - Http://www.???
What is Laser-Scan?

- British company, based in Cambridge
  - Founded 1969, digital mapping since 1975
  - Spatial databases and object-oriented mapping
- Customers government & commercial
  - National Mapping Agencies - OSGB, UKHO, NOAA, NIMA, LINZ, INEGI, IGN-F, IGN-B, KMS, + AA, Philips…
- Now a member of the Yeoman Group
  - British Group listed on AIM stock market
  - Specialists in mobile navigation - sea and land
  - First commercial turn-by-turn car navigation service to the mobile phone - TravelM8
Scenario, Problems, & Goal

• Mapping organisations have captured framework data at detailed scale, but customers want different:
  – mapping at smaller scale (covering bigger area per sheet)
  – mapping to suit current purpose, with clutter removed
  – simplified mapping for location-based services (LBS)

• Or maintain several map series at different scales:
  – very costly in manpower
  – derived maps often out of date

• Goal is capture once - use many times
  – previously impossible, but new technology changes that
  – considerable cost savings over manual generalisation
Technologies to provide solution

- Active objects
- Intelligent Agents
- Java & XML
- Spatial databases
- Explicit topology (adjacency, connectivity, sharing)
- Spatial data re-engineering
Good Generalisation Requires:

- **Contextual analysis** -
  - you can't generalise one map feature at a time in isolation. You have to consider groups of objects as a whole.

- **Adaptive processing** -
  - you can't apply a single algorithm to all features (even of a single class). You have to choose appropriate algorithms according to the circumstances of that feature.

- **Backtracking** -
  - you can't get it right first time every time. You have to be prepared to assess whether an operation has made things better or not, and be prepared to undo it and try something else.
Agent Lifecycle

1. Evaluate Constraints using Measures
2. Formulate Plans List
3. Choose Next Best Plan
4. Execute Algorithm of best plan
5. Re-evaluate Constraints

- Good Enough?
  - No: Backtrack State
  - Yes: Done

- Better?
  - No: Backtrack State
  - Yes: Backtrack State
Agent-based generalisation of farms
• Mapping
• Agencies
• Generalisation
• NETwork

A co-ordinated group of National and Regional Mapping Agencies working with Laser-Scan to consolidate and extend current generalisation capabilities in line with their business requirements
New Agent Core Design

AGENT Project Core

JADE Agent Parameters

JADE Generalisation Parameters

JADE Process Sequencing

Debug Tools

Combined Algorithms

Build, Test and Issue

Docs

Training Courses

Examples

Ongoing

JAADML

Text Label Placement

Related Projects 11/02/02 PGH

Generalisation - Work to create generic product

Existing

Old Generaliser Interface

Old Generaliser Sequencing

Old Generaliser Processes

AGENT Project Algorithms

Carto2001 Algorithms

KMS Algorithms

Academic Research

Roads (Beams)

German AdV Processes

New Agent Core Design

New Agent Core

Clarity
Gothic OODB
(active object cache)

Client
Operator or Batch
Import / Export Data
FME

Flowline Developer
Setup Generalisation
JADE plug-in

Operator or Batch
Instigate Generalisation
JADE plug-in

Operator or Developer
View/Edit map
JADE plug-in

Gothic Plug-in

JADE Java Desktop

CORBA
JNI

Gothic CORBA API
Gothic Java API

Gothic Spatial Toolkit
+ Object Access

Generaliser support
AGENT core etc

Server

Gothic OODDB

Data Model Behaviours
Lull/Java

Roads
Buildings
Rivers
Etc…

Operator or
Developer

View/Edit
map
JADE plug-in

Operator or Batch
Instigate
Generalisation
JADE plug-in

Flowline
Developer
Setup
Generalisation
JADE plug-in

Operator or Batch
Import / Export Data
FME

Gothic Plug-in

JADE Java Desktop

CORBA
JNI

Gothic CORBA API
Gothic Java API

Gothic Spatial Toolkit
+ Object Access

Generaliser support
AGENT core etc

Server

Gothic OODDB

(active object cache)
package gothic.user;

/**
 * An example of how to write a Java class that defines Java methods for use as Gothic method behaviours.
 */

import gothic.descriptor.Plot;
import gothic.descriptor.Geometry;
import gothic.descriptor.Version;
import gothic.main.GothicException;
import gothic.main.UnexpectedNullException;
import gothic.main.UnexpectedTypeException;

public class Area extends gothic.main.GothicObject {
    /**
     * Our display method. This assumes that the Gothic "Area" class using this display method inherits from "graphic"
     * or one of its relations, which provide a "Default" display method.
     */
    public boolean Default(Plot plotId, int overlayIndex, int colourModel, boolean draw)
        throws GothicException, UnexpectedNullException, UnexpectedTypeException
    {
        int result;
        Geometry geomId;
        boolean success = false;

        geomId = (Geometry) getValue("geometry");

        try {
            // Plot in a colour according to our number of vertices
            plotId.areaFilled(overlayIndex, colourModel, draw, geomId, getValue("number_of_vertices"));
            success = true;
        } finally {
            geomId.destroy();
        }

        return new Boolean(success);
    }
}
Java interface to Agent Constraints
Java Interface to Map Spec Parameters
XML for Map Specification definition, including Constraints

```xml
<xml version="1.0" encoding="UTF-8" ?>
  <mapSpecification>
    <targetScale>30000.000000</targetScale>
    <description>XML map specification</description>
    <version>1</version>
    <happinessData>
      <happiness>
        <happinessClass>agent_building</happinessClass>
        <happinessValue>2.000000</happinessValue>
      </happiness>
    </happinessData>
    <constraintData>
      <agentClass>
        <agentClassName>agent_maj_road</agentClassName>
        <subgroup>
          <constraint>
            <constraintName>road_building_dist_constraint</constraintName>
            <type>4</type>
            <priority>2</priority>
            <importance>3</importance>
            <goalValues>
              <goal>
                <type>5</type>
                <value>11.250000</value>
              </goal>
            </goalValues>
            <proposingCounters>
              <counter>
                <action>accordion</action>
                <restriction>1</restriction>
              </counter>
              <proposingCounters>
              </constraint>
            </proposingCounters>
          </subgroup>
        </agentClass>
      </agentClass>
    </constraintData>
  </mapSpecification>
</xml>
```
XSD schema for Process Sequences

```xml
<?xml version='1.0' encoding='UTF-8' ?>
<!-- Generated by Gothic Developer LULL code -->
<xs:schema targetNamespace='http://laser-scan.com/schema'
    xmlns:ps='http://www.w3.org/2001/XMLSchema' elementFormDefault='qualified'>
    <xs:element name='process-sequence'>
        <xs:annotation>
            <xs:documentation>Gothic process sequence</xs:documentation>
        </xs:annotation>
        <xs:complexType>
            <xs:sequence maxOccurs='unbounded'>
                <xs:element name='Process' type='ps:process_specify' />
            </xs:sequence>
        </xs:complexType>
    </xs:element>

    <xs:element name='View' abstract='true' />
    <xs:element name='Markup' abstract='true' />
    <xs:element name='ProcessInvocation' abstract='true' />

    <xs:complexType name='view_specify'>
        <xs:sequence>
            <xs:element name='view' ref='ps:View' />
        </xs:sequence>
    </xs:complexType>

    <xs:complexType name='markup_specify'>
        <xs:sequence>
            <xs:element name='markup' ref='ps:Markup' />
        </xs:sequence>
    </xs:complexType>

    <xs:complexType name='process_specify'>
        <xs:sequence>
            <xs:element name='process' ref='ps:ProcessInvocation' />
        </xs:sequence>
    </xs:complexType>

    <xs:complexType name='process-invocation'>
        <xs:sequence>
            <xs:element name='Name' type='xs:string' />
            <xs:element name='Process-Type' type='ps:process_type' minOccurs='1' />
            <xs:element name='markup-output-with' type='ps:markup_specify' minOccurs='1' />
            <xs:element name='Abort-on' minOccurs='1' maxOccurs='unbounded'>
                <xs:simpleType>
                    <xs:restriction base='xs:string'>
                        <xs:enumeration value='Invalid' />  
                        <xs:enumeration value='Not Known' />  
                        <xs:enumeration value='Processing' />  
                    </xs:restriction>
                </xs:simpleType>
            </xs:element>
        </xs:sequence>
    </xs:complexType>
</xs:schema>
```
Incremental Generalisation

One Gothic Data Set - Two Schemas

External Data Base
Base-DLM
Update Base-DLM
Update Base-DLM
Update Base-DLM
Update Base-DLM
DLM 50
DLM 50
DLM 50
DLM 50

Initial Derivation

Generalisation of Upgrades

Automatic Incremental Generalisation
Explicit topology knowledge is vital to good generalisation.
Radius - Topology in Oracle

MapInfo → GeoMedia → Other clients → Web clients:
Clients

Radius Topology Engine → Oracle 9i → Oracle Spatial
Server

Radius Triggers → Geometry → Business data
Database

Nodes → Edges → Faces

Lines → Areas
Summary

• Generalisation needs:
  – Contextual analysis
  – Adaptive processing
  – Backtracking
• Active objects and agent lifecycle provide framework
• Java & XML provide underlying tools & interoperability
• Explicit topology provides spatial knowledge
• MAGNET consortium steers development
• Rapidly evolving product can already give big savings
Generalisation with Agents and active objects

Is the Way Ahead!

Paul Hardy, Melanie Hayles, Patrick Revell
(paul.hardy@laser-scan.com)
http://www.laser-scan.com