Project AGENT

Overview and results of an European R&D projects in Map Generalisation

Preparation: A. Ruas, C. Duchêne
Presentation: A. Ruas
Automated Generalization New Technology

Esprit Project

\[ \text{3 years: Dec 97 -> Nov 2000} \]
\[ \text{21 man years} \]

\[ \text{Expected results:} \]
- Prototype implementation
- which generalises topographic data
- based on the Agent paradigm
Partners

Uni. Zurich
Robert Weibel,
Mathieu Barrault,
Geof. Dutton,
Mats Bader

Laser-Scan
Mike Jackson,
Paul Hardy,
Kelvin Haire,
Richard Horn
Daniel Ormsby

COGIT
Sylvie Lamy,
Anne Ruas,
Cécile Duchêne..

Generalisation

SIG
OODB

AGENT

Uni. Edinburgh
William Mackaness,
Nicolas Regnauld,
Alistair Edwards

INPG
Yves Demazeau,
Christof Baeijs

Agent
Create a SIG package which contains:
- a large set of algorithms
- measures
- mechanisms

to automate generalisation process
What is an Agent?

- **Entity**
  - with Behaviours
  - with **Autonomy**
    - Can use itself its own behaviours
    - To reach its own **GOAL**
  - with Perception
    - can ‘see’ other entities
  - which communicates
    - can exchange with other entities
A building agent
AGENT is a principle

Many implementations are possible
How we have implemented Agent

**Object Oriented**
- class = one type of geographical object (including groups)
- with Behaviours (= methods):
  - generalisation algorithms

**with Autonomy**
- conception of specific engine
- use knowledge, based on object type

**with control to activate agents**
Behaviours:
- Simplification
- Dilation
- Squaring
- Change-to-rectangle

Autonomy: rules & engine

If I AM I USE

- smaller than \( \lambda \)
  - Change-to-rect
- too small
  - Dilation
- too detailed
  - Simplification
- not squared
  - Squaring
How does it work?

- Generalisation knowledge is located at the class level
  - If XXX then use XXX or XXX
- An action is applied if certain conditions are fulfilled
- Conditions are constraints violation / user needs
Characters and Constraints

**Characters:**
- size
- granularity
- shape
  - elongation
  - squareness
- position
- orientation

**Constraints:**
- **visual**
  - Size > X1
  - granularity > X2
  - squareness MAX
- **Maintenance**
  - elon-fin ≅ elon-ini
  - pos-fin ≅ pos-ini
  - ori-ini ≅ ori-ini

**Characters:**
- density
- type
- proximity

**Constraints:**
- **visual**
  - density < X3
  - proximity > X4
- **Maintenance**
  - type-fin ≅ type-ini
Building modelling

Severty = ‘distance’ btw current value and goal_value
Priority = which constraint should be solved first
Happiness = ‘Σ’ of constraint severity
AGENT Engine

- An agent can act only when it is active.
- It consults its characters and its constraint violations.
- It tries processes to improve itself & to reach a ‘good’ state:
  - according to procedural knowledge
  - if ... then
  - controlled by the evolution of its state.
The engine of one agent

- Characterise & Evaluate
- Propose plans
- Choose best plan & trigger
- Re-evaluate
  - valid
  - invalid
- Climb
  - No plan left at a node
- Store & compare
- Perfect
- Passive

better
worse
Who activates an Agent?

Another Agent which has a more global view (concept of meso agent)

- a building is activated by its urban block,
- a urban block is activated by its town

A Map-Agent for upper level agents
Automated Generalization New Technology

Application field

Road-network generalisation
- Road selection & displacement
  - Each road generalisation
    - with recursive line segmentation

Town generalisation
- Street removal
  - Each urban block generalisation
    - Buildings removal & displacement
      - Each building generalisation
        - Dilation, shape improvement
Dynamic Urban block

Automated Generalization New

Co-ordination

Legislation

Select the best

Analyze and change the goal

Congestion

Side effect management

Building gene

Control

No more

Create Urban block Meso agent

Characterise & Evaluate

Propose plans

Choose best plan & trigger

Re-evaluate

valid

invalid

Better

Worse

Bldg removal
Bldg displacement
MST Typification
access typification

Carry on or backtrack

Perfect

Passive
Some examples of results (1)
Some examples of results (2)
Convergence towards a solution

The convergence depends on the **completeness** and the **quality** of:

- the algorithms
- the measures to qualify object’s characters
- the procedural knowledge
Possibility to introduce
  • Knowledge & user needs

Proof of the AGENT paradigm
  • Convergence towards solution

Usable in production line

Conclusions

http://agent.ign.fr
Next Steps ...

Research

• Include results coming from machine learning

• Introduce \textit{negotiation mechanisms} to improve objects choice / context

• Enrich side effect management
  • new algorithms?

Production

• tune / each generalisation
HOW TO IMPROVE COMMUNICATION?