Automatic Knowledge Revision in a Generalisation System

ACI Workshop 2007

Patrick TAILLANDIER

PhD Supervisor : Alexis Drogoul
Co-supervisor : Cécile Duchêne
Outline

1. Introduction
   - Generalisation automation
   - Procedural knowledge revision

2. The AGENT model
   - General presentation
   - The constraints
   - The actions cycle
   - Procedural knowledge of the model

3. Our knowledge revision approach
   - General approach
   - Procedural knowledge revision methods

4. Case study: the housing estates
   - Context
   - Results

5. Conclusion
1. Introduction:

Generalisation automation

- Lot of existing approaches
- Agent based approach:
  - Geographical objects are modelled by agents
  - Agents generalise themselves by applying generalisation algorithms (actions) on themselves
  - Much knowledge is included in the process
1. Introduction:

Procedural knowledge revision

- Knowledge is good but not perfect
- Knowledge improvement => Generalisation system could be:
  - More effective (cartographic quality of the result)
  - More efficient (to get the result faster)
- Knowledge evolution is complex

Objective: Propose a procedural knowledge revision approach, without experts interventions
2.1 The AGENT model:

*General presentation*

- Conceptual model developed by Anne Ruas (1999), then improved during the European project AGENT

- Geographical Objects = geographical agents
2.1 The AGENT model:  
*General presentation*

- Two kinds of agents:
  - *Micro*: individual geographical object
  - *Meso*: set of organised objects
2.2 The AGENT model:

The constraints

**Size constraint:**
Your surface must be big enough
*Proposed action:* scaling

**Shape constraint:**
Your shape must be preserved

**Granularity constraint:**
The details too small must be eliminated
*Proposed action:* elimination of the details which are too small

**Orientation constraint:**
Your orientation must be preserved
2.2 The AGENT model:

The constraints

- Each constraint has:
  - **A goal value**: value to reach. Computed according to the users specifications
  - **A current value**: value which represents the current state
  - **A satisfaction**: computed from the current value and the goal value for a given state. Marked between 1 and 10.
  - **A priority**: the constraint treatment urgency. Marked between 1 and 5.
  - **An importance**: importance to satisfy the constraint for the cartographic result. Marked between 1 and 5.

**Size constraint**

Goal value: > 300 m²  Current value: 240 m²  Satisfaction: 8
Priority: 3  Importance: 4
2.3 The AGENT model: the actions cycle

1. Is the state perfect?
   - Yes
   - No, go to 2.

2. Is the state valid?
   - No, go to 3.
   - Yes, go to 4.

3. Backtrack to the previous state.

4. Ask actions from its constraints and order them.

5. Are there still actions to test?
   - No, go to 7.
   - Yes, go to 6.

6. Apply the action which has priority and store the new state.

7. Is the state the initial state?
   - Yes, go to 10.
   - No, go to 8.

8. Backtrack to the previous state.


10. Stop the cycle.

2 constraints:
- **Size** (S): priority = 1, importance = 4
- **Granularity** (G): priority = 3, importance = 2
2.4 The AGENT model:

Procedural knowledge of the model

- Procedural knowledge:
  - Application order for the actions proposed by the constraints
  - State validity criterion
  - Actions cycle stopping criterion
3.1 Our knowledge revision approach: 

*General approach*

- Approach based on 3 steps:
  - Log the system when generalising
  - Acquire new knowledge from the logs
  - Integrate new knowledge in the system
3.2 Our knowledge revision approach:

*Constraints priority rules learning*

- **Objective:** Learn for each constraint, a base of priority rules

- **Rule example:**
  
  - **If** the size constraint satisfaction > 5 and the granularity constraint satisfaction < 4
  
  **Then** the size constraint has priority with a confidence rate of 80%
3.2 Our knowledge revision approach:

*Constraints priority rules learning*

- Example: 2 constraints: a size constraint (SC) and a granularity constraint (GC)
  - Learnt rules:
    - Rules for SC: **if** SC’s satisfaction (SSC) > 4 and GC’s satisfaction (SGC) < 3 **then** SC has priority with a confidence rate: CR = 80%
    - Rules for GC: **if** SGC < 10 **then** GC has priority with CR = 75%
  - States example:
    - SSC = 7, SGC = 8: GC has priority with CR = 75%
      → GC’s actions will be tested in priority
    - SSC = 8, SGC = 2: SC has priority avec CR = 80% and GC has priority with CR = 75%
      → SC’s actions will be tested in priority
3.2 Our knowledge revision approach:

*Constraints priority rules learning*

### Learning set:

**For SC:**
- State 1, Priority
- State 2, Non priority

**Pour GC:**
- State 1, Non priority
- State 2, Priority
3.3 Our knowledge revision approach: 
*States validity rules learning*

- **Objective:** Learn **rules** allowing the system to determine the **validity** of a given state

- **Rule example:**
  - **If** the size constraint satisfaction > 9 and the granularity constraint satisfaction < 2
  - **then** *the state is invalid*
3.3 Our knowledge revision approach: 
*States validity rules learning*

- **Learning set:**
  - State 1, Valid
  - State 2, Valid
  - State 3, Invalid
  - State 4, Valid
  - State 5, Invalid
3.4 Our knowledge revision approach:

*Actions cycle stopping rules learning*

- **Objective**: Learn **rules** allowing the system to determine if the **actions cycle** must be **stopped** or not for given state

- **Rule example**:
  - **If** the size constraint satisfaction $> 9$ and the granularity constraint satisfaction $> 7$
  - **then** the **actions cycle must be stopped**
3.4 Our knowledge revision approach: 
*Actions cycle stopping rules learning*

Learning set:
- State 1, Continue
- State 2, Continue
- State 4, Stop
4.1 Case study: the housing estates

**Context**

**Constraints:**

- **Buildings satisfaction**
  - **Action**: buildings generalisation
  - Priority = 5, Importance = 5

- **Proximity between buildings**
  - **Action**: buildings aggregation
  - Priority = 5, Importance = 4

- **Proximity between buildings and between roads and buildings**
  - **Action**: displacement and removal of buildings
  - Priority = 5, Importance = 4

- **Preservation of a buildings minimum number**
  - **No action**
  - Priority = 5, Importance = 4
4.1 Case study: the housing estates

Context

- Learning set: built from the generalisation of 40 pieces of housing estates
- Machine learning algorithm: c4.5
- Test phase: system tested on 40 pieces of housing estates
4.2 Case study: the housing estates

Results
4.2 Case study: the housing estates

Results
5. Conclusion

- Summary
  - Development of a procedural knowledge revision approach which does not need experts interventions
  - Validation of the approach by a case study on the housing estates
5. Conclusion

Perspectives:

- Actions choice revision
- Development of methods to better take into account the system initial knowledge
THANK YOU FOR YOUR ATTENTION