A Multi-Agent System for Building Dynamic Ontologies

*Kévin Ottens*, Marie-Pierre Gleizes & Pierre Glize

Institut de Recherche en Informatique de Toulouse (IRIT)
SMAC team

AAMAS 2007 – May 14–18 2007, Honolulu, Hawai’i, USA.
Plan

1. Introduction
2. Introducing the Dynamo System
3. Distributed Clustering Algorithm
4. Multi-Criteria Hierarchy
5. Discussion & Perspectives
Plan

1. Introduction

2. Introducing the Dynamo System

3. Distributed Clustering Algorithm

4. Multi-Criteria Hierarchy

5. Discussion & Perspectives
Introduction

Current situation

- Text analysis makes ontology building easier
- NLP analysis examination is a difficult and slow process
- Emerging technics based on machine learning

Our proposal

- Keep the user in the production loop
- Allow the ”Living Design” of ontologies
- Reorganization following user modifications
Plan

1. Introduction
2. Introducing the Dynamo System
3. Distributed Clustering Algorithm
4. Multi-Criteria Hierarchy
5. Discussion & Perspectives
Overview

Architecture

Multi-Agent System

Ontologist Interface

Term network

Concept Agent

Term

Terms Extraction Tool
Overview

Term Network
- Produced by Syntex
- "Head-Expansion" graph
  - knowledge engineering from text
  - knowledge engineering
- Term contexts used to compute similarity

Multi-Agent System
- Each agent represents a concept of the taxonomy
- Each agent tries to position itself
- Based on a condition/action rule set
Plan

1. Introduction

2. Introducing the Dynamo System

3. Distributed Clustering Algorithm

4. Multi-Criteria Hierarchy

5. Discussion & Perspectives
Distributed Clustering Algorithm

Local view

Steps

1. Evaluating similarity and "votes"
2. Partitioning and intermediate layer creation
3. Parent change
Distributed Clustering Algorithm

**Local view**

**Steps**

1. Evaluating similarity and "votes"
2. Partitioning and intermediate layer creation
3. Parent change
**Steps**

1. Evaluating similarity and "votes"
2. Partitioning and intermediate layer creation
3. Parent change
Experimental Complexity Results

- Average complexity: $O(n^2 \log(n))$
- Maximum variance: around 5%
Qualitative Point of View

**Automated run**
- Permanent view on the built hierarchy
- Allow to obtain a "first draft"

**User modification**
- No algorithm adjustment required
- Dynamicity, revision of the structure
Plan

1. Introduction
2. Introducing the Dynamo System
3. Distributed Clustering Algorithm
4. Multi-Criteria Hierarchy
5. Discussion & Perspectives
Observations

- Similarity can’t be always computed for term pairs
- Humans have specific heuristics for low-level structuring

Goal

- Take care of those terms
- Implement a similar heuristic

Parent Adequacy Function

- The best parent for $C$ is the $P$ agent that maximizes $a(P, C)$.
- When an agent $C$ is unsatisfied by its parent $P$, it evaluates $a(B_i, C)$ with all its brothers (noted $B_i$) the one maximizing $a(B_i, C)$ is then chosen as the new parent.
Managing Several Criteria

Guidelines

How?

- Keeping it simple
  - Local criteria
  - Nominal values for those criteria
- Use cooperation heuristic

Cooperation

- Minimizing non-cooperation
- Priority system
  - Determine the current problems
  - Find the most urgent one
  - Try to fix it
### Minimize non cooperation

- $\mu_H(A)$: "head coverage" non cooperation degree of $A$
- $\mu_B(A)$: "brotherhood" non cooperation degree of $A$
- $\mu_M(A)$: "message" non cooperation degree of $A$
- $\mu(A) = \max(\mu_H(A), \mu_B(A), \mu_M(A))$

### Take care of the worst problem first

- $\mu(A) = \mu_H(A) \rightarrow$ Try to find a better parent
- $\mu(A) = \mu_B(A) \rightarrow$ Improve structuring through clustering
- $\mu(A) = \mu_M(A) \rightarrow$ Process other agent message
Experimental Complexity Revisited

- Average complexity: $O(n^3)$
- Maximum variance: around 0.6%
Discussion

Advantages of our approach

- Easier system/ontologist coupling
- Possible distribution on a network

Current limitations

- Results tend to depend on the add order
- Tend to produce binary trees only (except on leaves)
Perspectives

Concerning knowledge engineering
- Get closer to a taxonomy tree
- Find non taxonomic relations

Concerning multi-agent systems
- Improve the clustering algorithm
  - Remove the dependency on add order
  - Optimize
- Test this algorithm in other domains
In progress...

- Taxonomy production
  - Tree pruning
  - Not only binary tree
- Evaluate the system on more corpora

Conclusion

- Evolving structure is possible in this field
- Performances are acceptable
- More efforts needed...
Questions?