A Weighted-Tree Similarity Algorithm for Multi-Agent Systems in e-Business Environments

Virendra C.Bhavsar*
Harold Boley**
Lu Yang*

*Faculty of Computer Science, Univ. of New Brunswick, Fredericton

**Institute for Information Technology – e-Business, NRC, Fredericton

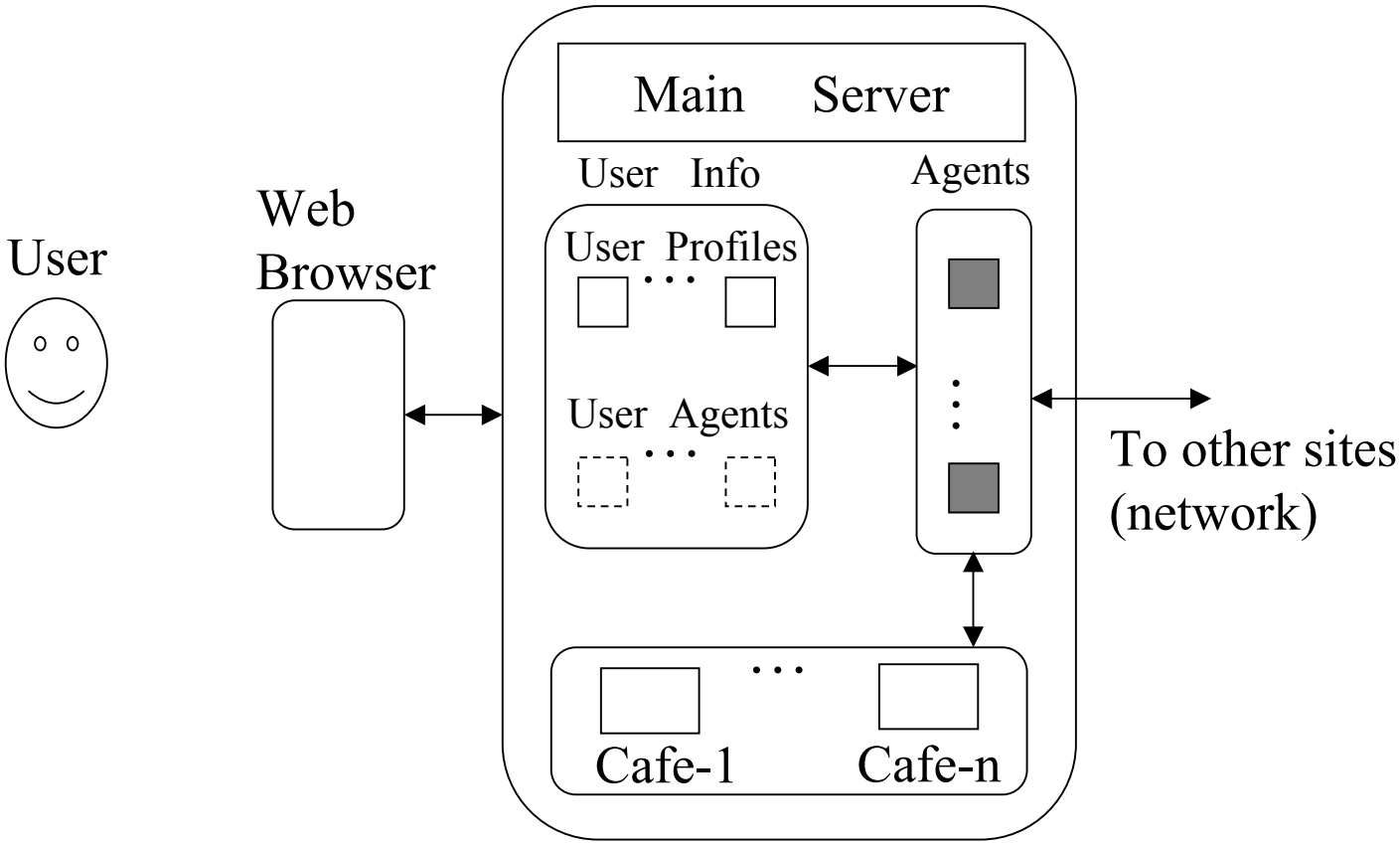
Outline

- Introduction
- Multi-agent system architecture
- Tree representation
- Similarity of trees
- Experimental results
- Conclusion

Introduction

- e-business, e-learning.
- Semantic Web and Web Services.
  - Virtual marketplace.
  - Buyer-seller message exchange.
- Semantic match-making in multiagent systems.
  - Keywords/keyphrases.

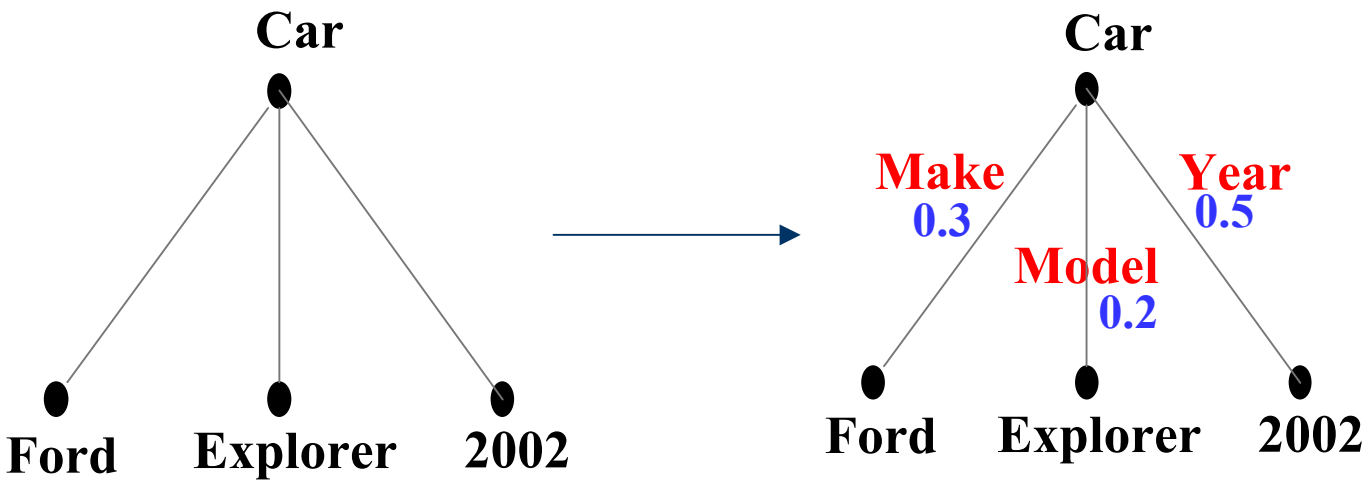
Multi-agent system architecture



Agent-based Community Oriented Routing Network (ACORN)

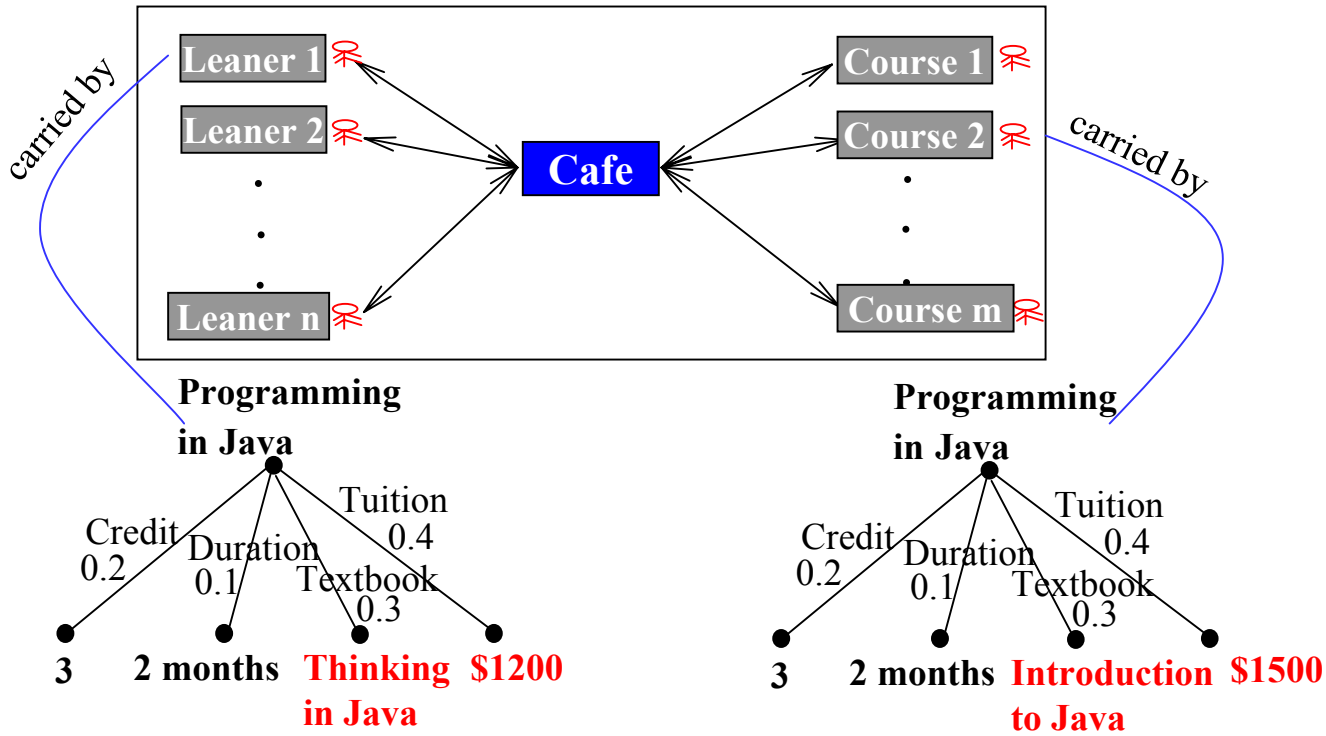
Tree representation

- Why tree representation?
  - Flexibly represent complex structures.
  - Why *arc-labelled*, *arc-weighted* tree?



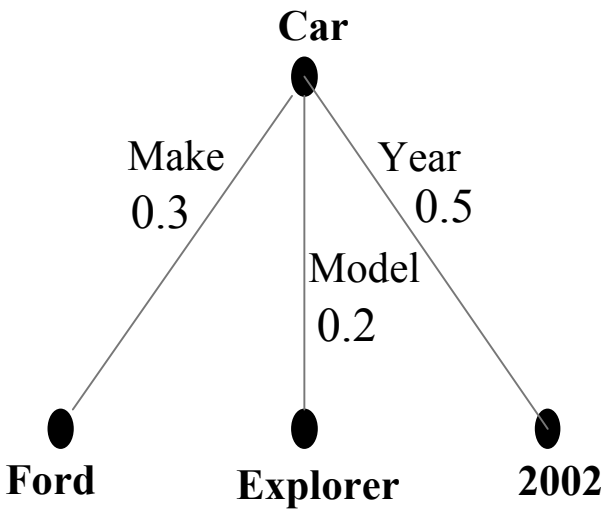
Matchmaking of agents

- Match-making in the Cafe.

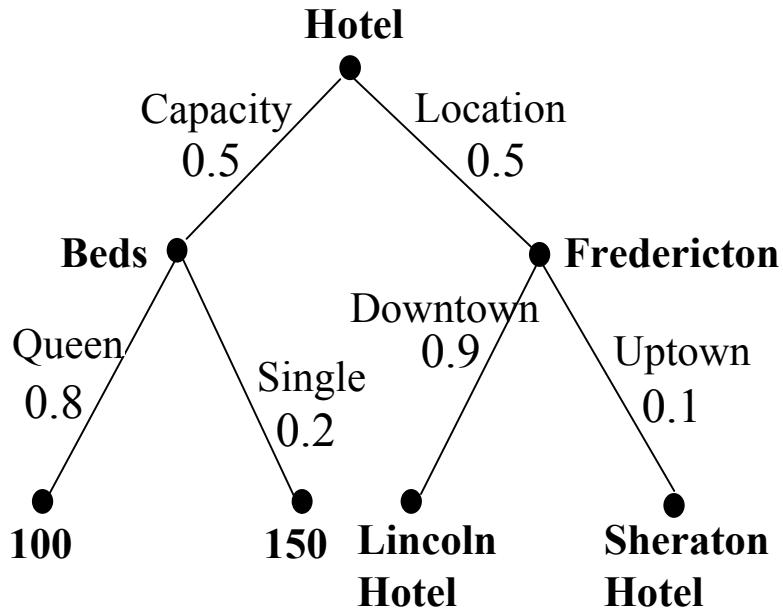


Tree representation - lexicographic order

- The arcs are labelled in lexicographic (alphabetical) order.



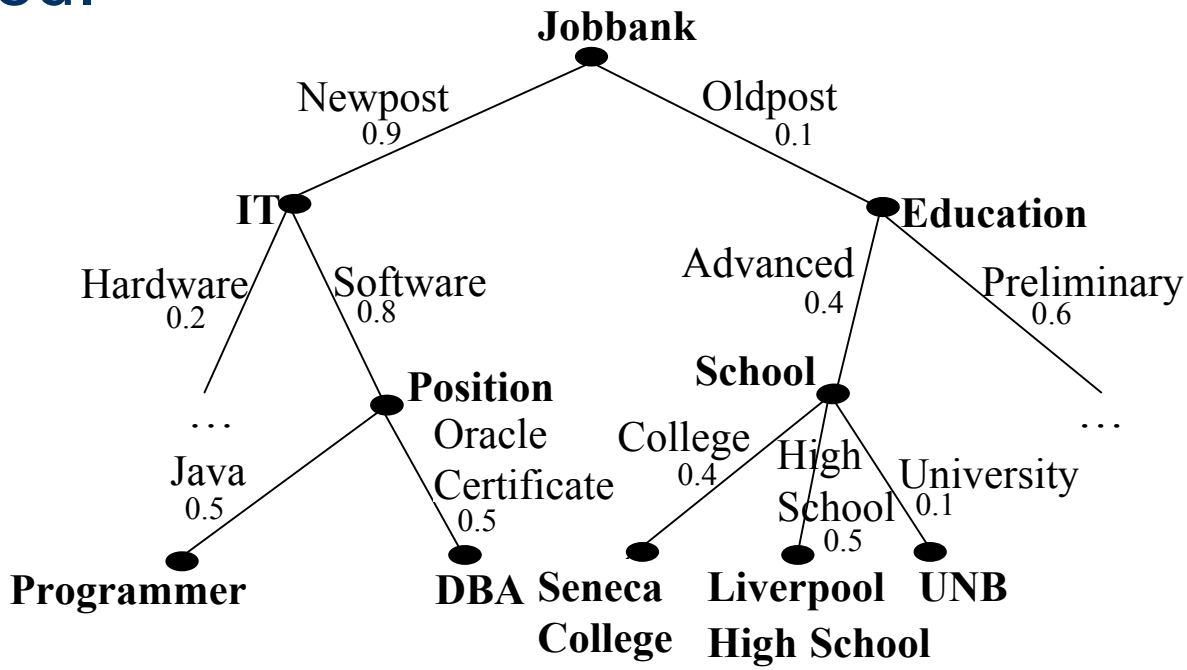
A tree describing "Car".



A tree describing "Hotel".

Tree representation - depth and breadth

- The depth and breadth of any subtree are not limited.



A tree that describes "Jobbank".

Serialization of trees

- Weighted Object-Oriented RuleML.
- XML attributes for arc labels and weights.

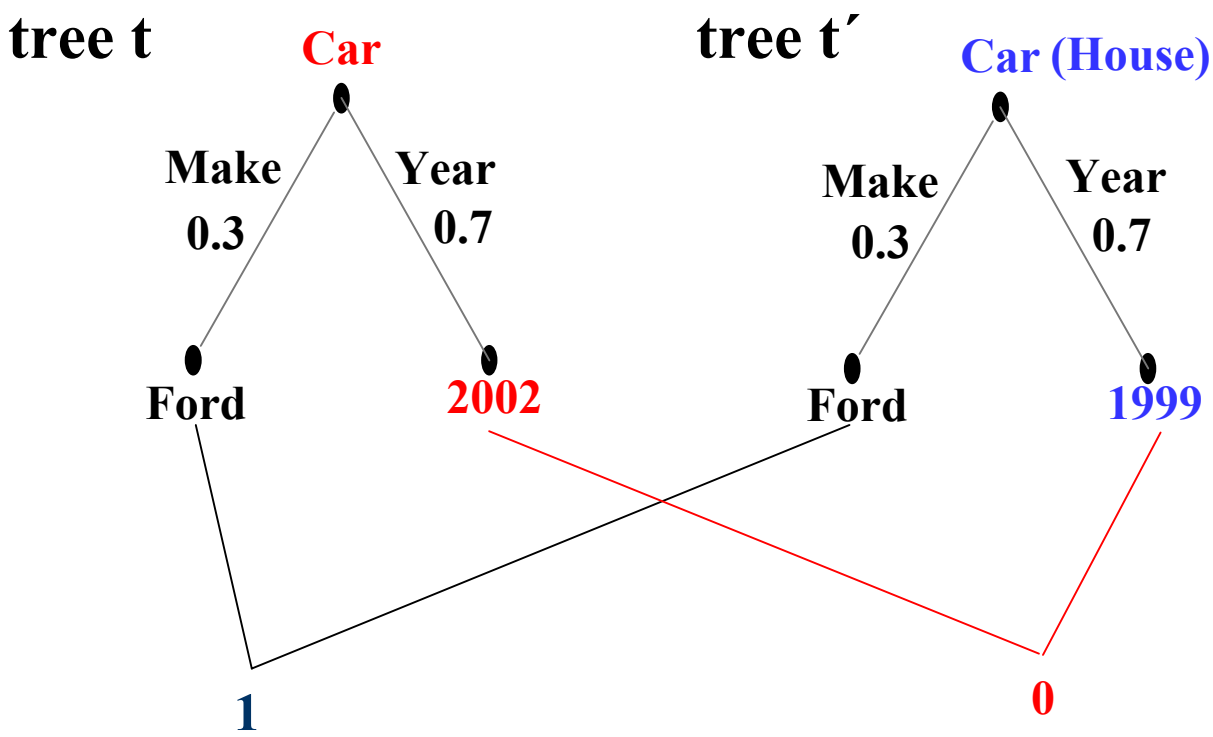
```
<cterm>
  <_opc><ctor>Car</ctor></_opc>
  <_r n="Make" w="0.3"><ind>Ford</ind></_r>
  <_r n="Model" w="0.2"><ind>Explorer</ind></_r>
  <_r n="Year" w="0.5"><ind>1999</ind></_r>
</cterm>
```

Tree serialization in OO RuleML.

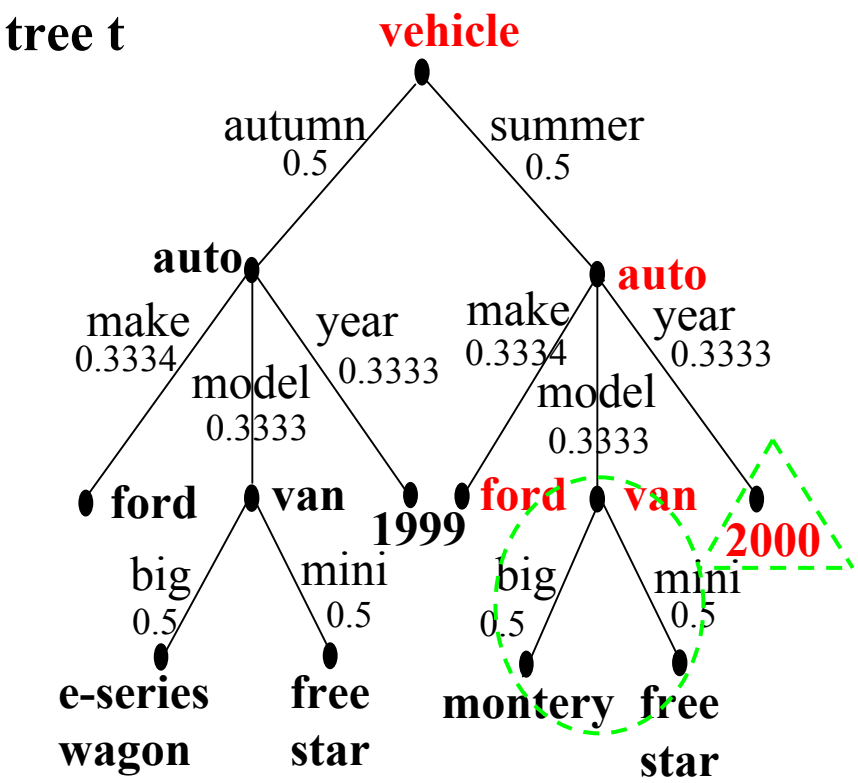
```
cterm[ -opc[ctor[car]],
       -r[n[make],w[0.3]][ind[ford]],
       -r[n[model],w[0.2]][ind[explorer],
       -r[n[year],w[0.5]][ind[1999]]
      ]
```

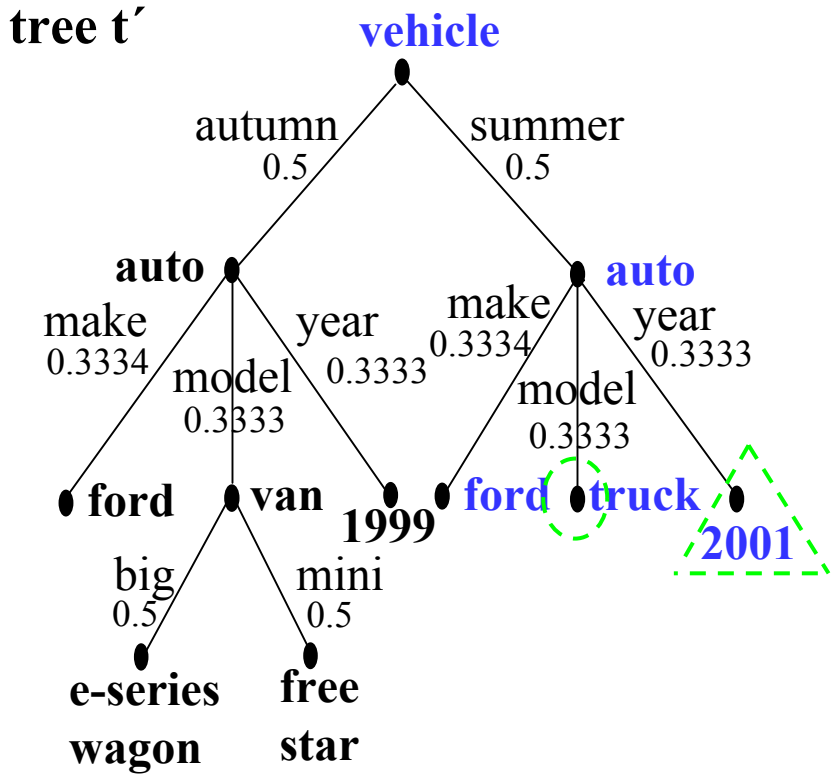
Tree representation in Relfun.

Similarity of trees – simple trees



Similarity of trees – complex trees





$\sum s_i(w_i + w'_i)/2 \quad \rightarrow \quad \sum A(s_i)(w_i + w'_i)/2$

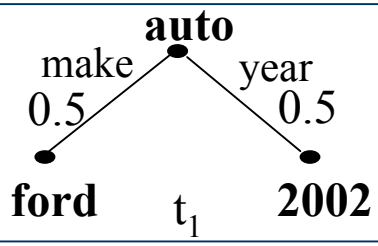
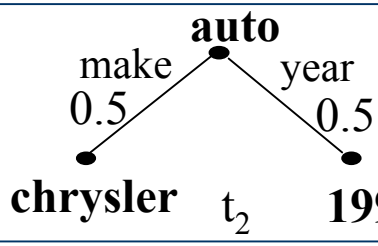
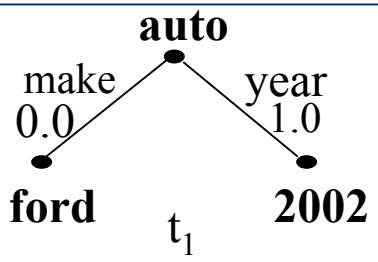
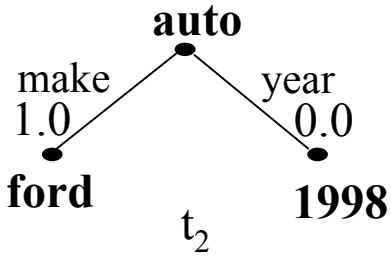
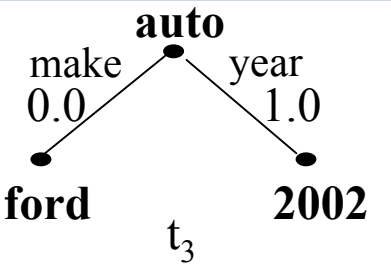
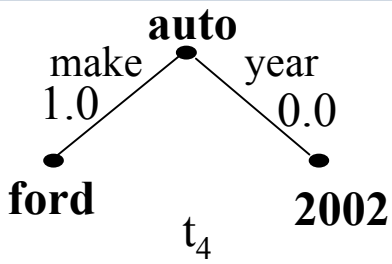
$A(s_i) = s_i$

$A(s_i) = \sqrt{s_i}$.

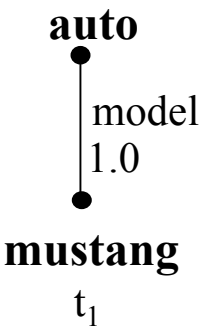
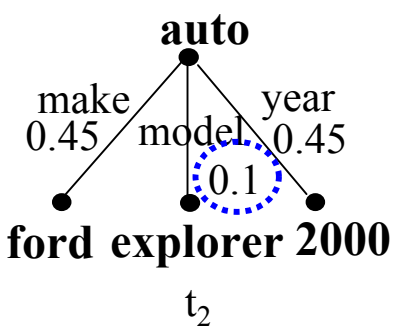
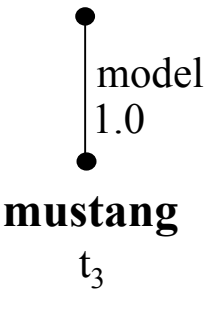
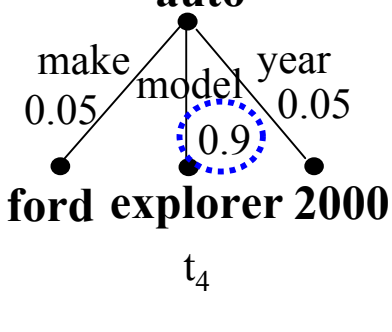
Algorithm for tree similarity

- Three main recursive functions of the algorithm.
  - Treesim(t,t'): Recursively compares any (unordered) pair of trees.
  - Treemap(l,l'): Co-recursively maps two lists, l and l', of labelled and weighted arcs: descends into identical–labelled subtrees.
  - Treeplicity(i,t): Decreases the similarity with decreasing simplicity.

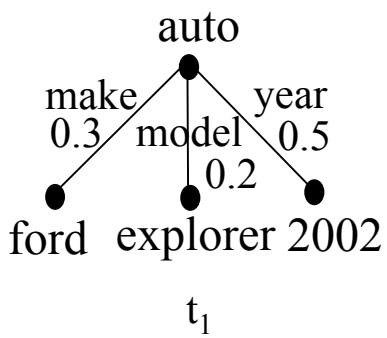
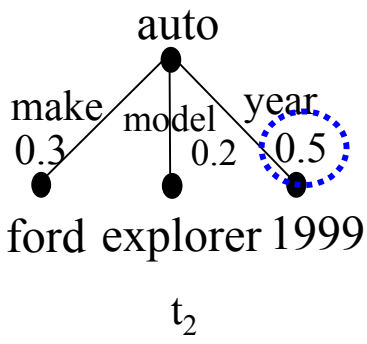
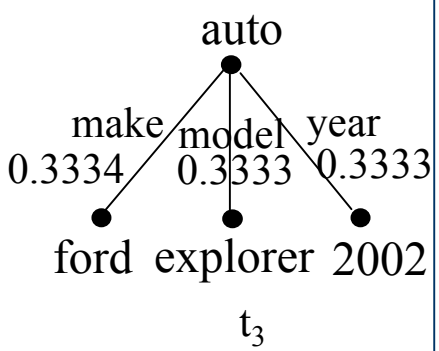
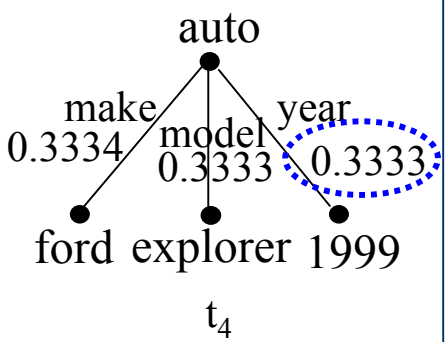
Experimental results –simple trees

Experiments	Tree	Tree	Results
1			0.1
2			0.5
			1.0

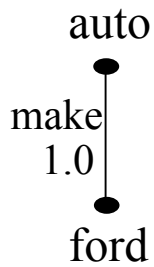
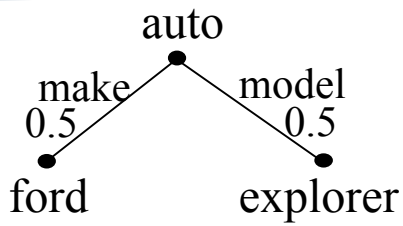
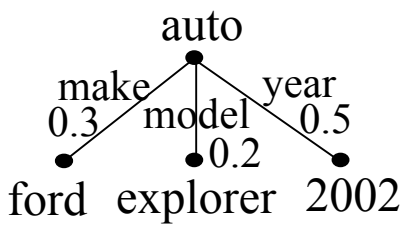
Experimental results – simple trees (cont'd)

Experiments	Tree	Tree	Results
3	 t_1	 t_2	**0.2823**
	 t_3	 t_4	**0.1203**

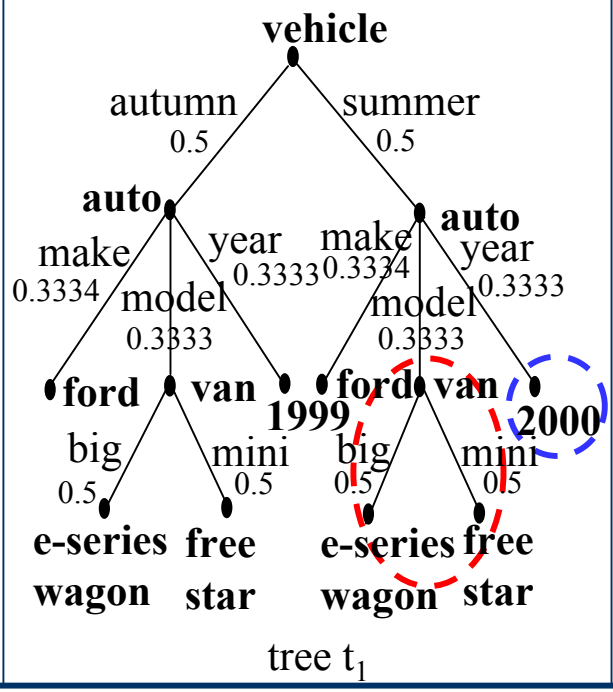
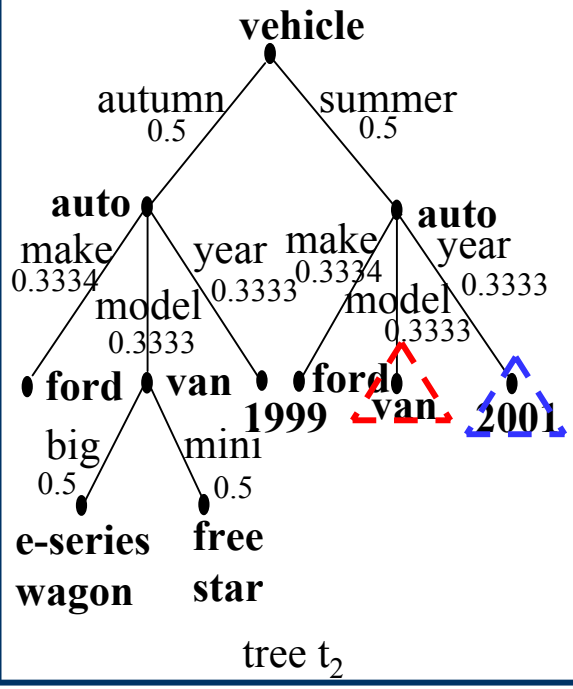
Experimental results – identical tree structures

Experiments	Tree	Tree	Results
4	 t_1	 t_2	**0.55**
	 t_3	 t_4	**0.7000**

Experimental results – progressively complex trees

Experiments	Tree	Tree	Results
5	auto • t_1	t_2  	0.3025
		t_3  	0.3025
		t_4  	0.3025

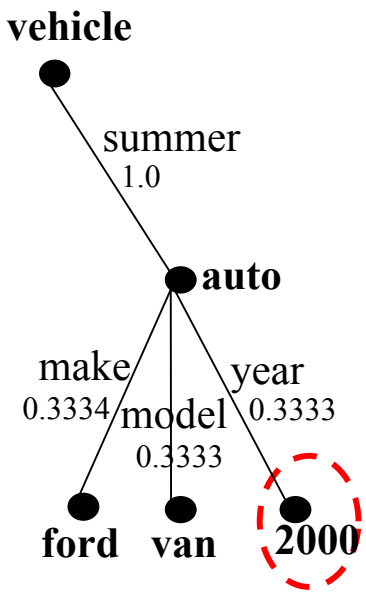
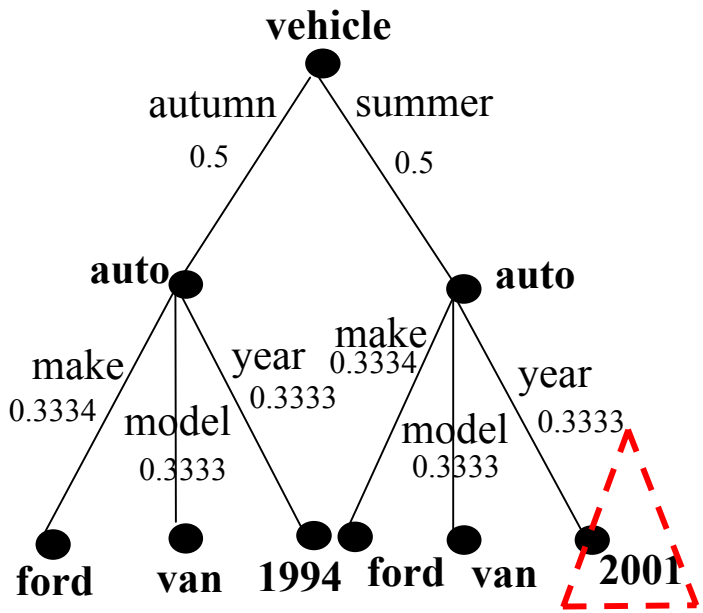
Experimental results – complex trees

Experiments	Tree	Tree	S_i / $\sqrt{s_i}$
6			0.5950 0.7611

tree t_1

tree t_2

Experimental results – complex trees (cont'd)

Experiments	Tree	Tree	S_i / $\sqrt{s_i}$
7			0.5894 0.6816

Conclusion

- Tree representations – useful for e-Business, e-Learning.
- Matchmaking in multiagent systems – a versatile tree similarity algorithm is proposed.
- Executable specification available in functional-logic language: Relfun.
  - A Java implementation is in progress.
- Future work - Clustering of agents.

Thank you!
?