Knowledge Sharing in CBR
Data and Information

Data: tokens/symbols
- BLUE
- 1941

Information: data + context
- a relational database
- a story
- a semantic network
Knowledge and Sharing

Agent applies knowledge to increase utility

Knowledge has two components:
• Procedures that agent can execute
• Classifiers that recognize environment

To share knowledge...
• Conversing agents must have similar utility metrics
Summary of Previous Work

- Translated attribute of database with coded semantic
- Agent applied action based upon perception of the state of the attribute of interest
- Information translation, not knowledge
- Limited by information/knowledge available to agents

- Contribution: incorporation of utility allows for highly autonomous reasoning.
Motivation

- CBR (case-based **retrieval**, more precisely) has become common in knowledge management environments.

- Sharing of knowledge is an economical way for CBR systems to increase effectiveness

- Dell / Microsoft Example
Case-Based Reasoning: Overview

- “similar problems have similar solutions”
- Agent keeps history of experience
  - Problem
  - Applied solution
  - Outcome
- A solution to a new problem starts by searching case base for most similar past experience
Case-Based Reasoning: “4Rs”

The 4 Rs of case-based reasoning:
• Retrieve
  – Find most similar case to current problem
• Reuse
  – Adapt solution of prior experience to current problem
  – Apply solution
• Revise
  – Observe outcome, repair solution if necessary
• Retain
  – Add new case to case-base
Case-Based Reasoning: Knowledge

Knowledge in Case-Based Reasoning is distributed across three components:

• Indexed properties of the problem
• Solution associated to past experience of problem
• Similarity metric used in retrieval
  – Derived from domain knowledge, approximates similarity of solutions between problems.
Description of Problem

- Two agents Al and Ar are CBR with attribute-indexed homogeneous cases
- Al and Ar have different indexing schemes and retrieval metrics
- Create a translation that maps the cases of Ar into cases usable by Al
**Goals**

Agents are social, but not necessarily cooperative.

- Maximize agent's knowledge base
- Minimize work required by remote agent
- Maximize autonomy of agents
Related...

On Case-based Knowledge Sharing in the Semantic Web

- H. Chen and Z. Wu
- + Introduce RDF markup and ontological-based framework for case-base sharing in open systems
- + Meta description valid for attribute-indexed, structured, and other case representations
- - Use uberlingua for translation
Related...

Integrating Case-based Reasoning and Decision Theory

- C. Tsatsoulis, Q. Cheng, and H. Wei

+ Novel approach to reasoning about cases without completely defined problem spaces
+ Application of decision/utility theory to case similarity
- No knowledge sharing/acquisition
Formulation: Schema

- Schema is a set $S = \{a_1, \ldots, a_n\}$
- For each attribute $a$ in $S$, $\text{domainOf}(S, a)$ is a finite, discrete set of the values which an instance of $a$ can take.
Formulation: Problem Instance

- A problem instance $P$ is a set of ordered pairs of the form $\{<a_1,v_1>,...,<a_m,v_m>\}$ where each $a$ in schema $S$ and $v$ in domainOf($S,a$)

- $\text{schemaOf}(P)$ is the schema over which $P$ is defined
- $\text{valueOf}(P, a)$ is the value bound to the given attribute.
- A complete problem defines values for all $a$ in $S$. 
Formulation: Agent

An agent is defined by the following functions...

- `schemaOf(A)` the schema used by A
- `actionsOf(A)` a set \( \{k_1, \ldots, k_n\} \) of actions
- `casesOf(A)` the set of cases known by A.
- `distance(A, P_1, P_2)` the distance metric
- `reward(A, k, P)` the utility gained by A by applying action k to problem P.
- `retrieve(C, d, p)` retrieves the most similar case in C, with respect to problem p and metric d.
Formulation: Case

A case $c$ is defined by...

- $\text{idOf}(c)$ a unique (random) identifying string
- $\text{agentOf}(c)$ the agent that recorded the case
- $\text{problemOf}(c)$ the problem as defined in the case, $c$.
- $\text{solutionOf}(c)$ the solution associated to $c$.

Note that $\text{solutionOf}(c)$ in $\text{actionsOf}(\text{agentOf}(c))$
Formulation: Attribute Translator

An attribute translator, $T_a(S_1, a_1, S_2, a_2, v_1)$, maps the value $v_1$ in $\text{domainOf}(S_1, a_1)$ into a new value in $\text{domainOf}(S_2, a_2)$.

- Let $M$ be $|\text{domainOf}(S_1, a_1)|$.
- Let $N$ be $|\text{domainOf}(S_2, a_2)|$.
- There are $N^M$ possible attribute translators for $(a_1, a_2)$. 
**Formulation: Schema Translator**

Schema translators are constructed from sets of attribute translators.

- A schema translator $T_s(S_1, S_2, P_1)$ maps the problem $P_1$ defined over $S_1$ schema into a problem $P_2$ defined over $S_2$.
- An attribute $P_1$ of may only map into one attribute of $P_2$.
- $\text{Permute}(M,N)$ possible schema translators, where $M = |S_1|$ and $N = |S_2|$.
Problem Statement

- Two agents $A_l$ and $A_r$
- $\text{schemaOf}(A_l) \neq \text{schemaOf}(A_r)$
- $\text{actionsOf}(A_l) == \text{actionsOf}(A_r)$
- Find schema translator $T_s$ such that...

Sum for all $c$ in $\text{casesOf}(A_r)$ $[\text{utility}(A_l, \text{solutionOf}(c), T_s(\text{schemaOf}(A_r), \text{schemaOf}(A_l), \text{problemOf}(c)))] > 0$
Solution Overview

- Generate Ts
- Apply Ts to each case c in casesOf(AI) to produce xCases
- For each case test in xCases:
  - Find best match for test in (xCases – test) according to Ar's distance metric
  - Compute utility gain by applying match case solution to original problem.
- Accept Ts if sum of utilities is > 0
Future: Better Solution Method

- Solution method requires brute force search of large space.
- While intractable for pathological cases, likely, heuristics will make average case feasible.
Future: Feedback Mechanism

- Remote cases should remain in state of partial trust until local agent has made use of a case and evaluated its outcome according to its own perception of utility.

- As remote cases are applied trust in the translation should be modified accordingly.

- Allow agent to preserve translation and re-evaluate translation as more evidence is collected.
Future: More Realistic Knowledge

• The omniscient perspective where each agent knows perfectly its utility function is very unrealistic.

• The formulation must evolve to reflect conditions where the agent knows the utility of problem/solution pairs only its case-base.
Future: Graph-based Cases

• Graph-based cases (semantic networks) have been proposed as a richer means of case representation

• Study feasibility of applying utility-oriented translation techniques to graphical cases
Conclusions

- Database utility-oriented attribute-entity translation scheme has been modified/extended for use in CBR.

- Completion of method will enable CBR agents to operate in open systems with high autonomy.