A Methodology for Knowledge Acquisition by Heterogeneous STRIPS-Based Planners

Sujata Ramchandran

Dept. of Electrical & Computer Engineering
University of Arizona
Outline

• Introduction
• Problem Formulation
• Definitions
• Learning Methodology
• Detailed Example
• Related Work
• Conclusion
Introduction

• We can achieve communication in a network of heterogeneous planning systems through translation
• If a problem cannot be solved,
  – it indicates a lack of knowledge within the planner
  – results in the need to send out a query to other planners
• To avoid future queries and solve other similar problems, the planner should acquire knowledge from the solutions that it receives from other planners
• The challenges
  – Many different ways to learn
  – Knowledge representation may be at different levels of abstraction
  – Preserve the correctness of the knowledge base
Problem Formulation

Components of the System

- *Planner*$_i$ contains
  - Knowledge Base - $KB_i$ with
    - Predicates - $Pred_i$
    - Actions – $Actions_i$
      - $Pre(a^k_i)$, $Add(a^k_i)$, $Del(a^k_i)$, $Con(a^k_i)$
    - Translators – $Trans_i$
  - Domain Truth Table
    - Sets of predicates that can never exist together and those that cannot exist separately in the domain theory of $Planner_i$
    - Predicates that are always true or always false with respect to the common ontology
- Common Ontology - the knowledge representation which used by all planners to communicate with each other.
Problem Formulation

Communication between Planners

• Communication is through
  • Queries containing Initial Conditions, Goals and Constraints
    \( \text{Query}_i = <I_i, G_i, C_i> \)
  • Plans containing Actions and Orderings
    \( \text{Plan}_i = <A_i, O_i> \)

• Communication occurs with the following steps
  In Planner\(_i\):
    • Create \( \text{Query}_i = <I_i, G_i, C_i> \)
    • Translate and send as \( \text{Query}_{co} = <I_{co}, G_{co}, C_{co}> \)
  In Planner\(_j\):
    • Receive \( \text{Query}_{co} \) and translate to \( \text{Query}_j = <I_j, G_j, C_j> \)
    • Create the solution, \( \text{Plan}_j = <A_j, O_j> \)
    • Translate and send \( \text{Plan}_{co} = <A_{co}, O_{co}> \) as the reply
  In Planner\(_i\):
    • Receive the solution and translate to \( \text{Plan}_i = <A_i, O_i> \)
Problem Formulation

Communication between Planners

• Translation implies that equivalent knowledge is present.

• If this is not true, the solution cannot be represented in the domain theory of $Planner_i$. Hence, the solution cannot be used.

• Objective – To present a methodology that will achieve translation through learning while preserving the correctness of knowledge in the domain theory of the planner.
Problem Formulation

Formal Objective

- Given Plan$_j$ = <$A_j,O_j$> in the domain theory of Planner$_j$ to add Pred$_i^{new}$, Actions$_i^{new}$ and Trans$_i^{new}$ to KB$_i$ = <$Pred_i,Actions_i,Trans_i$>, in the domain theory of Planner$_i$, such that
  - Plan$_j$ = <$A_j,O_j$> can be translated to Plan$_i$ = <$A_i,O_i$> with the following properties
    - Plan$_i$ ≡ Plan$_j$
    - Plan$_i$ is a correct plan
    - Plan$_i$ is a solution for Query$_i$ = <$I_i,G_i,C_i$>
  - Constraints on this solution
    - Actions$_i^{new}$ is minimal subject to the individual actions being the most similar to Actions$_i$ when compared to any other Actions$_i^{new'}$ that could have been added to KB$_i$
    - KB$_i'$ = <$Pred_i$ $\cup$ Pred$_i^{new}$, Actions$_i$ $\cup$ Actions$_i^{new}$, Trans$_i$ $\cup$ Trans$_i^{new}$ > is consistent and all its elements are correct
Problem Formulation
Why is this Non-Trivial?

- Creating an algorithm for a heuristic process
- Given Plan$_j$ with $m$ actions, maximum number of plans $N$ considered for learning is in the range

$$2^{m-1} < N < B_m$$

where $B_m$ is a Bell number corresponding to the number of ways to partition a set into non-empty subsets.

$$B_{m+1} = \sum_{k=0}^{m} B_k \binom{m}{k}, \text{ with } B_0 = 1$$

e.g. with $m = 8$ actions, $128 < N < 4140$
Methodology

Definitions - Equivalence of Knowledge

• The subset of predicates, $Pred_i^p$ from $Planner_i$, is equivalent to the subset of predicates, $Pred_j^q$ in $Planner_j$, if their aggregate semantics are the same. In this case we write $Pred_i^p \equiv Pred_j^q$

  $Pred_i^p = Tower_i(A,B,C)$
  $Pred_j^q = On_j(A,B), On_j(B,C), On_j(C, Table), Clear_j(A)$

• Two plans, $Plan_i = \langle A_i, O_i \rangle$ and $Plan_j = \langle A_j, O_j \rangle$ are equivalent ($Plan_i \equiv Plan_j$) if $Pre(Plan_i) \equiv Pre(Plan_j)$, $Add(Plan_i) \equiv Add(Plan_j)$, $Del(Plan_i) \equiv Del(Plan_j)$ and $Con(Plan_i) \equiv Con(Plan_j)$.

  $Plan_i$
  $A_i = 2\_Tower\_to\_Table_i(D,E), 3\_Tower\_to\_Tower_i(A,B,D)$
  $O_i = before(2\_Tower\_to\_Table_i(D,E), 3\_Tower\_to\_Tower_i(A,B,D))$

  $Plan_j$
  $A_j = Unstack_j(D,E), Putdown_j(D), Unstack_j(A,B), Stack_j(A,D)$
  $O_j = before(Unstack_j(D,E), Putdown_j(D)), before(Putdown_j(D), Unstack_j(A,B)), before(Unstack_j(A,B), Stack_j(A,D))$
Methodology

Definitions - Correctness of Knowledge

• An action $a_i^k$ is correct in $Planner_i$ if
  – $Del(a_i^k)$ is a subset of $Pre(a_i^k)$.
  – $\exists P_i^k \subset Pre(a_i^k)$ or $\exists P_i^k \subset Add(a_i^k)$ or $\exists P_i^k \subset Con(a_i^k)$, such that $P_i^k$ is listed as always false in the Domain Truth Table of Planner$_i$.
  – $\exists P_i^k \subset Del(a_i^k)$, such that $P_i^k$ is listed as always true in the Domain Truth Table of Planner$_i$.

• $Plan_i = <A_i, O_i>$, produced by $Planner_i$ is correct if
  – The orderings between the actions in $Plan_i$ are acyclic.
  – All actions in $A_i$ are correct with respect to the domain theory of $Planner_i$.

• $Plan_i = <A_i, O_i>$ is a solution to $Query_i = <I_i, G_i, C_i>$ if
  – $Plan_i$ is correct plan.
  – $Pre(Plan_i)$ is a subset of $I_i$.
  – $G_i$ is a subset of $Add(Plan_i)$.
  – $C_i$ is a subset of $Con(Plan_i)$.
Methodology

Definitions – Similarity of Knowledge

• Similarity Index (SI) - the ratio of the number of common predicates between actions to the total number of predicates in both actions. Ranges from 1 (actions are equal) to 0 (actions have no common predicates)

\[ SI(a^p_i, a^q_i) = \frac{|\text{Common}_\text{ pred}(a^p_i, a^q_i)|}{|\text{Not}_\text{ common}_\text{ pred}(a^p_i, a^q_i)| + |\text{Common}_\text{ pred}(a^p_i, a^q_i)|} \]

\[ SI(a^p_i, a^q_i) = 0, \text{ if } \text{Common}_\text{ pred}(a^p_i, a^q_i) = \emptyset \]

where

\[ p^k_i \in \text{Common}_\text{ pred}(a^p_i, a^q_i) \text{ if } (p^k_i \in \text{Pre}(a^p_i) \land p^k_i \in \text{Pre}(a^q_i)) \]

\[ \lor (p^k_i \in \text{Add}(a^p_i) \land p^k_i \in \text{Add}(a^q_i)) \lor (p^k_i \in \text{Con}(a^p_i) \land p^k_i \in \text{Con}(a^q_i)) \]

\[ \text{otherwise } p^k_i \in \text{Not}_\text{ common}_\text{ pred}(a^p_i, a^q_i) \]

• If \( SI(a^p_i, a^k_i) > SI(a^q_i, a^k_i) \), then action \( a^p_i \) is more similar than action \( a^q_i \) when compared to action \( a^k_i \).
**Methodology**

**Definitions – Similarity of Knowledge**

- The Similarity Index of two sets of actions, $A_i^p$ and $A_i^q$, is defined as

$$SI(A_i^p, A_i^q) = \frac{\left|\sum_{a_i^r, a_i^s} SI(a_i^r, a_i^s)\right|}{|A_i^p|}$$

where

$$\sum_{a_i^r, a_i^s} SI(a_i^r, a_i^s), \ \forall \ a_i^r \in A_i^p \text{ and } a_i^s \in A_i^q \text{ such that, }$$

$$SI(a_i^r, a_i^s) > SI(a_i^r, a_i^m) \ \forall \ a_i^m \in A_i^q, \ m \neq s.$$  

- If $SI(A_i^p, A_i^k) > SI(A_i^q, A_i^k)$, then the set of actions $A_i^p$ is more similar than the set $A_i^q$ when compared to the set of actions $A_i^k$. 

Methodology

Learning Algorithm

1. For $Plan_{co} = <A_{co}, O_{co}>$, generate all possible partitions of actions in $A_{co}$ and corresponding orderings.
   
   $Partition^{n}_{co} = \{<A_{co}^{n-1}, O_{co}^{n-1}> <A_{co}^{n-2}, O_{co}^{n-2}> ... <A_{co}^{n-n}, O_{co}^{n-n}> <O_{co}^{n-m}>\}$

2. Eliminate all partitions, $Partition^{k}_{co}$, that will result in plans which are not acyclic.
   
   $Partition^{k}_{co}$ is eliminated if, $\exists before(a_{co}^{p}, a_{co}^{q}) \in O_{co}^{k-m}$ and $\exists before(a_{co}^{r}, a_{co}^{s}) \in O_{co}^{k-m}$, $p \neq q \neq r \neq s$, $a_{co}^{p}, a_{co}^{s} \in A_{co}^{k-x}$ and $a_{co}^{q}, a_{co}^{r} \in A_{co}^{k-y}$, $x \neq y$

3. Eliminate the partitions that will result in plans containing incorrect actions.
   
   From the remaining partitions, select those which contain the most number of elements $<A_{co}^{k-l}, O_{co}^{nk-l}>$ which can be translated to the domain theory of $Planner_i$. 
Methodology

Learning Algorithm

4. For every Partition \( \text{co}^k \) selected above, form a set of actions \( A^k_i \) such that, \( \forall <A_{\text{co}^k}^k, O_{\text{co}^k}^k> \in \text{Partition}_{\text{co}^k} \) that cannot be translated by Planner \( i \), \( \exists a^l_i \in A^k_i, a^l_i \equiv <A_{\text{co}^k}^l, O_{\text{co}^k}^l> \).
   a. Each action \( a^l_i \) in \( A^k_i \) is formed as follows.
   b. Partition \( \text{Pre}(<A_{\text{co}^k}^k, O_{\text{co}^k}^k>) \) in to \( \{P_{\text{co}^1}, P_{\text{co}^2}, ..., P_{\text{co}^n}, P_{\text{co}^m}\} \) such that, \( \forall P_{\text{co}^k}, k=1,...,n, \exists t^k_i \in \text{Trans}_i \) that represents the mapping

\[
P^k_{\text{co}} \leftarrow t^k_i \rightarrow P^k_i
\]

and \( \forall p_{\text{co}}^r \in \text{Pre}(<A_{\text{co}^k}^k, O_{\text{co}^k}^k>) \land p_{\text{co}}^r \notin P^k_{\text{co}}, k=1,...,n, p_{\text{co}}^r \in P^m_{\text{co}} \)
Similarly, partition \( \text{Add}(<A_{\text{co}^k}^k, O_{\text{co}^k}^k>) \), \( \text{Del}(<A_{\text{co}^k}^k, O_{\text{co}^k}^k>) \) and \( \text{Con}(<A_{\text{co}^k}^k, O_{\text{co}^k}^k>) \).
   c. Form action \( a^l_i \) with \( \text{Pre}(a^l_i) = P_i^1 \cup P_i^2 \cup \ldots \cup P_i^n \cup P^m_{\text{co}} \) where \( P^k_{\text{co}} \) is the translation of \( P_{\text{co}^k} \), i.e.

\[
P^k_{\text{co}} \leftarrow t^k_i \rightarrow P^k_i
\]

In a similar manner, \( \text{Add}(a^l_i), \text{Del}(a^l_i) \) and \( \text{Con}(a^l_i) \) are obtained.
   d. Replace each constant in \( \text{Pre}(a^l_i), \text{Add}(a^l_i), \text{Del}(a^l_i) \) and \( \text{Con}(a^l_i) \) with a unique variable, to form the final action, \( a^l_i \).
Methodology

Learning Algorithm

5. For each set $A_i^k$, calculate the Similarity Index, $SI(A_i^k, Actions_i)$.

6. If $Actions_i^{new} = A_i^p$ such that $SI(A_i^p, Actions_i) > SI(A_i^q, Actions_i)$, $\forall q \neq p$. Then, $Partition_{co}^p$ is the final choice for learning.

7. $\forall a_i^k \in Actions_i^{new}$, if $\exists p_{co}^k \in Pre(a_i^k)$ or $\exists p_{co}^k \in Add(a_i^k)$ or $\exists p_{co}^k \in Con(a_i^k)$, such that
   a. $p_{co}^k \notin KB_i$, then let $p_i^k = p_{co}^k$ and $Pred_i^{new} = Pred_i^{new} \cup p_i^k$
      Formulate a translator $t_i^k$ that represents the mapping
      $p_i^k \leftarrow t_i^k \rightarrow p_{co}^k$
   b. If $p_{co}^k = p_i^k \in KB_i$, and $\nexists t_i^k \in Trans_i$ such that $p_i^k \leftarrow t_i^k \rightarrow p_{co}^k$
      then create the translator, $t_i^k$.

Let $Trans_i^{new} = Trans_i^{new} \cup t_i^k$. 
Methodology

Learning Algorithm

8. \( \forall a_i^k \in \text{Actions}_i^{new} \),
   1. If \( a_i^k \notin \text{Actions}_i \), generate a new name that is the concatenation of the action names in the common ontology. The parameters of \( a_i^k \) consist of the list of all the variables that are a part of the action definition of \( a_i^k \).
      Form the translator \( t_i^k \) which represents the mapping
      \[
      a_i^k \xrightarrow{t_i^k} < A_{co}^{p-k}, O_{co}^{p-k} >
      \]
      If \( a_i^k = a_i^l \in \text{Actions}_i, l \neq k \) and \( \nexists \ t_i^l \in \text{Trans}_i \) such that
      \[
      a_i^k \xrightarrow{t_i^k} < A_{co}^{p-k}, O_{co}^{p-k} >
      \]
      then formulate the translator.
      Let \( \text{Trans}_i^{new} = \text{Trans}_i^{new} \cup t_i^l \)

9. Update \( \text{KB}_i \) with the new information
   \[
   \text{Pred}_i = \text{Pred}_i \cup \text{Pred}_i^{new}
   \]
   \[
   \text{Actions}_i = \text{Actions}_i \cup \text{Actions}_i^{new}
   \]
   \[
   \text{Trans}_i = \text{Trans}_i \cup \text{Trans}_i^{new}
   \]
Example

Planner₃ — Toy Car Assembly

\[ I₃ = \text{Clear}_3(\text{Axle\_end1}), \text{Clear}_3(\text{Axle\_end2}), \text{Clear}_3(\text{Axle\_end3}), \text{Clear}_3(\text{Axle\_end4}), \text{Clear}_3(\text{Wheel1}), \text{Clear}_3(\text{Wheel2}), \text{Clear}_3(\text{Wheel3}), \text{Clear}_3(\text{Wheel4}), \text{Clear}_3(\text{Eng}), \text{Clear}_3(\text{Batt}), \text{Clear}_3(\text{Eng\_slot}), \text{Clear}_3(\text{Batt\_slot}), \text{Clear}_3(\text{Top}), \text{Clear}_3(\text{Base}) \]

\[ G₃ = \text{Car}_2₃ \]

\[ C₃ = \text{Wheel}_3(\text{Wheel1}), \text{Wheel}_3(\text{Wheel2}), \text{Wheel}_3(\text{Wheel3}), \text{Wheel}_3(\text{Wheel4}), \text{Car\_top}_3(\text{Top}), \text{Car\_base}_3(\text{Base}), \text{Axle\_end}_3(\text{Axle\_end1}), \text{Axle\_end}_3(\text{Axle\_end2}), \text{Axle\_end}_3(\text{Axle\_end3}), \text{Axle\_end}_3(\text{Axle\_end4}), \text{Engine}_3(\text{Eng}), \text{Engine\_slot}_3(\text{Eng\_slot}), \text{Battery}_3(\text{Batt}), \text{Battery\_slot}_3(\text{Batt\_Slot}) \]
## Example

### Planner$_3$ – Action Descriptions

<table>
<thead>
<tr>
<th>Action</th>
<th>Pre</th>
<th>Add</th>
<th>Del</th>
<th>Con</th>
</tr>
</thead>
<tbody>
<tr>
<td>Fix_together$_3$ (?x,?y)</td>
<td>Clear$_3$(?x)</td>
<td>On$_3$(?x,?y)</td>
<td>Clear(?y)</td>
<td>Wheel$_3$(?x), Axle_end$_3$(?y)</td>
</tr>
<tr>
<td>Insert_wheel$_3$(?y,?x)</td>
<td>Clear$_3$(?x)</td>
<td>Inserted_wheel$_3$(?y,?x)</td>
<td>Clear$_3$(?y)</td>
<td>Wheel$_3$(?w1), Wheel$_3$(?w2)</td>
</tr>
<tr>
<td>Base_assembly$_3$ (?bs,?w1,?w2,?w3,?w4,?ae1,?ae2,?ae3,?ae4)</td>
<td>Inserted_wheel$_3$ (?ae1,?w1)</td>
<td>Assembled_base_car$_1$</td>
<td>Wheel$_3$(?w3), Wheel$_3$(?w4) Axle_end$_3$(?ae1) Axle_end$_3$(?ae2) Axle_end$_3$(?ae3) Axle_end$_3$(?ae4) Car_base$_3$(?bs)</td>
<td></td>
</tr>
<tr>
<td>Body_assembly$_3$ (?tp,?bs)</td>
<td>Fastened$_3$(?tp,?bs)</td>
<td>Assembled_body$_3$</td>
<td>Car_top$_3$(?tp)</td>
<td>Car_base$_3$</td>
</tr>
<tr>
<td>Make_car$_1$$_3$</td>
<td>Assembled_base_car$_1$</td>
<td>Car$_1$$_3$</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Make_car$_2$$_3$</td>
<td>Assembled_base_car$_2$</td>
<td>Car$_2$$_3$</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Example

**Planner$_3$ — Mappings**

### Action Mappings in KB$_3$

<table>
<thead>
<tr>
<th>Planner$_3$</th>
<th>Common Ontology</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Fix$_{together}_3$(?x, ?y)</strong></td>
<td>Pickup$<em>{co}(?x) &gt; Align$</em>{co}(?x,?y) &gt; $</td>
</tr>
<tr>
<td></td>
<td>Place$_{co-{co}}(?x,?y) &gt; $</td>
</tr>
<tr>
<td></td>
<td>Attach$<em>{on$</em>{co}}(?x,?y)$</td>
</tr>
<tr>
<td><strong>Insert$_{wheel}_3(?x, ?y)</strong></td>
<td>Pickup$<em>{co-{co}}(?x) &gt; Align$</em>{co}(?x,?y) &gt; $</td>
</tr>
<tr>
<td></td>
<td>Insert$<em>{wheel}</em>{co}(?x,?y)$</td>
</tr>
<tr>
<td><strong>Base$_{assembly}_car1$_3(?bs)</strong></td>
<td>Wheel$<em>{assembly}</em>{co} &gt; $</td>
</tr>
<tr>
<td></td>
<td>Base$<em>{assembly}</em>{co}(?bs,Car1)$</td>
</tr>
<tr>
<td><strong>Body$_{assembly}_3(?cr,tp,?bs)</strong></td>
<td>Body$<em>{assembly}</em>{co}(?cr,tp,?bs)$</td>
</tr>
<tr>
<td><strong>Make$_{car}1$_3</strong></td>
<td>Assemble$_{co}(Car1)$</td>
</tr>
<tr>
<td><strong>Make$_{car}2$_3</strong></td>
<td>Assemble$_{co}(Car2)$</td>
</tr>
</tbody>
</table>

### Domain Truth Table of Planner$_3$

<table>
<thead>
<tr>
<th>Always True</th>
<th>Always False</th>
</tr>
</thead>
<tbody>
<tr>
<td>Handempty$_{co}$</td>
<td>Holding$_{co}$</td>
</tr>
</tbody>
</table>

### Predicate Mappings in KB$_3$

<table>
<thead>
<tr>
<th>Planner$_3$</th>
<th>Common Ontology</th>
</tr>
</thead>
<tbody>
<tr>
<td>On$_3(?x,?y)$</td>
<td>On$_{co}(?x,?y)$</td>
</tr>
<tr>
<td>Inserted$_{wheel}_3(?x,?y)$</td>
<td>In$<em>{wheel}</em>{co}(?x,?y)$</td>
</tr>
<tr>
<td>Clear$_3(?x)$</td>
<td>Clear$_{co}(?x)$</td>
</tr>
<tr>
<td>Fastened$_3(?x,?y)$</td>
<td>Attached$_{co}(?x,?y)$</td>
</tr>
<tr>
<td>Connected$_3(?x,?y)$</td>
<td>Wired$_{co}(?x,?y)$</td>
</tr>
<tr>
<td>Wheel$_3(?x)$</td>
<td>Wheel$_{co}(?x)$</td>
</tr>
<tr>
<td>Axle$_{end}_3(?x)$</td>
<td>Axle$<em>{end}</em>{co}(?x)$</td>
</tr>
<tr>
<td>Engine$_3(?x)$</td>
<td>Engine$_{co}(?x)$</td>
</tr>
<tr>
<td>Battery$_3(?x)$</td>
<td>Battery$_{co}(?x)$</td>
</tr>
<tr>
<td>Engine$_{slot}_3(?x)$</td>
<td>Engine$<em>{slot}</em>{co}(?x)$</td>
</tr>
<tr>
<td>Battery$_{slot}_3(?x)$</td>
<td>Battery$<em>{slot}</em>{co}(?x)$</td>
</tr>
<tr>
<td>Car$_{top}_3(?x)$</td>
<td>Top$_{co}(?x)$</td>
</tr>
<tr>
<td>Car$_{base}_3(?x)$</td>
<td>Base$_{co}(?x)$</td>
</tr>
<tr>
<td>Car$_3(?x)$</td>
<td>Car$_{co}(?x)$</td>
</tr>
<tr>
<td>Assembled$_{eng}_batt$_3</td>
<td>Assembled$_{eng}<em>batt</em>{co}$</td>
</tr>
<tr>
<td>Assembled$_{base}_car1$_3</td>
<td>Assembled$<em>{base}</em>{co}(Car1)$</td>
</tr>
<tr>
<td>Assembled$_{base}_car2$_3</td>
<td>Assembled$<em>{base}</em>{co}(Car2)$</td>
</tr>
<tr>
<td>Assembled$_{body}_3</td>
<td>Assembled$<em>{body}</em>{co}$</td>
</tr>
<tr>
<td>Car$_1$_3</td>
<td>Complete$_{co}(Car1)$</td>
</tr>
<tr>
<td>Car$_2$_3</td>
<td>Complete$_{co}(Car2)$</td>
</tr>
</tbody>
</table>
Example

**Planner3 - Query**

**Query\textsubscript{3}**

\[ I_3 = \text{Clear}_3(Axle\_end1), \text{Clear}_3(Axle\_end2), \text{Clear}_3(Axle\_end3), \text{Clear}_3(Axle\_end4), \]
\[ \quad \text{Clear}_3(Wheel1), \text{Clear}_3(Wheel2), \text{Clear}_3(Wheel3), \text{Clear}_3(Wheel4), \text{Clear}_3(Eng), \]
\[ \quad \text{Clear}_3(Batt), \text{Clear}_3(Eng\_slot), \text{Clear}_3(Batt\_slot), \text{Clear}_3(Top), \text{Clear}_3(Base), \]
\[ G_3 = \text{Assembled\_base\_car}_2 \]
\[ C_3 = \text{Wheel}_3(Wheel1), \text{Wheel}_3(Wheel2), \text{Wheel}_3(Wheel3), \text{Wheel}_3(Wheel4), \text{Car\_top}_3(Top), \]
\[ \quad \text{Car\_base}_3(Base), \text{Axle\_end}_3(Axle\_end1), \text{Axle\_end}_3(Axle\_end2), \]
\[ \quad \text{Axle\_end}_3(Axle\_end3), \text{Axle\_end}_3(Axle\_end4), \text{Engine}_3(Eng), \text{Engine\_slot}_3(Eng\_slot), \]
\[ \quad \text{Battery}_3(Batt), \text{Battery\_slot}_3(Batt\_Slot) \]

**Query\textsubscript{co}**

\[ I_{co} = \text{Clear}_{co}(Axle\_end1), \text{Clear}_{co}(Axle\_end2), \text{Clear}_{co}(Axle\_end3), \text{Clear}_{co}(Axle\_end4), \]
\[ \quad \text{Clear}_{co}(Wheel1), \text{Clear}_{co}(Wheel2), \text{Clear}_{co}(Wheel3), \text{Clear}_{co}(Wheel4), \text{Clear}_{co}(Eng), \]
\[ \quad \text{Clear}_{co}(Batt), \text{Clear}_{co}(Eng\_slot), \text{Clear}_{co}(Batt\_slot), \text{Clear}_{co}(Top), \text{Clear}_{co}(Base) \]
\[ G_{co} = \text{Assembled\_base\_co}(Car2) \]
\[ C_{co} = \text{Wheel}_{co}(Wheel1), \text{Wheel}_{co}(Wheel2), \text{Wheel}_{co}(Wheel3), \text{Wheel}_{co}(Wheel4), \]
\[ \quad \text{Top}_{co}(Top), \text{Base}_{co}(Base), \text{Axle\_end}_{co}(Axle\_end1), \text{Axle\_end}_{co}(Axle\_end2), \]
\[ \quad \text{Axle\_end}_{co}(Axle\_end3), \text{Axle\_end}_{co}(Axle\_end4), \text{Engine}_{co}(Eng), \]
\[ \quad \text{Engine\_slot}_{co}(Eng\_slot), \text{Battery}_{co}(Batt), \text{Battery\_slot}_{co}(Batt\_Slot), \text{Handempty}_{co} \]
Example

Common Ontology – Solution Plan
Example

Common Ontology – Solution Plan

\[
\begin{align*}
A_{co} &= a_{co}^1 = \text{Pickup}_{co}(Wheel1), \\
       &= a_{co}^2 = \text{Align}_{co}(Wheel1, Axle\_end1), \\
       &= a_{co}^3 = \text{Insert\_wheel}_{co}(Wheel1, Axle\_end1), \\
       &= a_{co}^4 = \text{Pickup}_{co}(Wheel2), \\
       &= a_{co}^5 = \text{Align}_{co}(Wheel, Axle\_end2), \\
       &= a_{co}^6 = \text{Insert\_wheel}_{co}(Wheel2, Axle\_end2), \\
       &= a_{co}^7 = \text{Pickup}_{co}(Wheel3), \\
       &= a_{co}^8 = \text{Align}_{co}(Wheel3, Axle\_end3), \\
       &= a_{co}^9 = \text{Insert\_wheel}_{co}(Wheel3, Axle\_end3), \\
       &= a_{co}^{10} = \text{Pickup}_{co}(Wheel4), \\
       &= a_{co}^{11} = \text{Align}_{co}(Wheel4, Axle\_end4), \\
       &= a_{co}^{12} = \text{Insert\_wheel}_{co}(Wheel4, Axle\_end4), \\
       &= a_{co}^{13} = \text{Wheel\_assembly}(Wheel1, Wheel2, Wheel3, Wheel4, Axle\_end1, Axle\_end2, Axle\_end3, Axle\_end4) \\
       &\quad a_{co}^{14} = \text{Pickup}_{co}(Eng), \\
       &\quad a_{co}^{15} = \text{Align}_{co}(Eng, Eng\_slot), \\
       &\quad a_{co}^{16} = \text{Insert}_{co}(Eng, Eng\_slot), \\
       &\quad a_{co}^{17} = \text{Attach\_in}_{co}(Eng, Eng\_slot), \\
       &\quad a_{co}^{18} = \text{Pickup}_{co}(Batt), \\
       &\quad a_{co}^{19} = \text{Align}_{co}(Batt, Batt\_slot), \\
       &\quad a_{co}^{20} = \text{Insert}_{co}(Batt, Batt\_slot), \\
       &\quad a_{co}^{21} = \text{Attach\_in}_{co}(Batt, Batt\_slot), \\
       &\quad a_{co}^{22} = \text{Connect\_wires}_{co}(Eng, Batt, Eng\_slot, Batt\_slot), \\
       &\quad a_{co}^{23} = \text{Eng\_batt\_assembly}_{co}(Eng, Batt, Eng\_slot, Batt\_slot), \\
       &\quad a_{co}^{24} = \text{Base\_assembly}_{co}(Base, Car2)
\end{align*}
\]

\[
O_{co} = \text{before}(a_{co}^1, a_{co}^2), \text{before}(a_{co}^2, a_{co}^3), \text{before}(a_{co}^4, a_{co}^5), \text{before}(a_{co}^7, a_{co}^6), \text{before}(a_{co}^7, a_{co}^8), \\
\text{before}(a_{co}^8, a_{co}^9), \text{before}(a_{co}^{10}, a_{co}^{11}), \text{before}(a_{co}^{11}, a_{co}^{12}), \text{before}(a_{co}^3, a_{co}^{13}), \text{before}(a_{co}^6, a_{co}^{13}), \text{before}(a_{co}^9, a_{co}^{13}), \\
\text{before}(a_{co}^{12}, a_{co}^{13}), \text{before}(a_{co}^{14}, a_{co}^{15}), \text{before}(a_{co}^{15}, a_{co}^{16}), \text{before}(a_{co}^{16}, a_{co}^{17}), \text{before}(a_{co}^{18}, a_{co}^{19}), \text{before}(a_{co}^{19}, a_{co}^{20}), \\
\text{before}(a_{co}^{20}, a_{co}^{21}), \text{before}(a_{co}^{17}, a_{co}^{22}), \text{before}(a_{co}^{21}, a_{co}^{22}), \text{before}(a_{co}^3, a_{co}^{23}), \text{before}(a_{co}^{21}, a_{co}^{23}), \text{before}(a_{co}^{22}, a_{co}^{23}), \\
\text{before}(a_{co}^{13}, a_{co}^{24}), \text{before}(a_{co}^{22}, a_{co}^{24})
\]
Example
Solution Plan
- Partitions
Example

Solution Plan - Partition

\[ A_{co}^{1-1} = \{a_{co}^1, a_{co}^2, a_{co}^3\} \]
\[ A_{co}^{1-2} = \{a_{co}^4, a_{co}^5, a_{co}^6\} \]
\[ A_{co}^{1-3} = \{a_{co}^7, a_{co}^8, a_{co}^9\} \]
\[ A_{co}^{1-4} = \{a_{co}^{10}, a_{co}^{11}, a_{co}^{12}\} \]
\[ A_{co}^{1-5} = \{a_{co}^{14}, a_{co}^{15}, a_{co}^{16}\} \]
\[ A_{co}^{1-6} = \{a_{co}^{17}\} \]
\[ A_{co}^{1-7} = \{a_{co}^{18}, a_{co}^{19}, a_{co}^{20}\} \]
\[ A_{co}^{1-8} = \{a_{co}^{21}\} \]
\[ A_{co}^{1-9} = \{a_{co}^{22}\} \]
\[ A_{co}^{1-10} = \{a_{co}^{23}\} \]
\[ A_{co}^{1-11} = \{a_{co}^{13}\} \]
\[ A_{co}^{1-12} = \{a_{co}^{24}\} \]

\[ O_{co}^{1-1} = \{(a_{co}^1 > a_{co}^2), (a_{co}^2 > a_{co}^3)\} \]
\[ O_{co}^{1-2} = \{(a_{co}^4 > a_{co}^5), (a_{co}^5 > a_{co}^6)\} \]
\[ O_{co}^{1-3} = \{(a_{co}^7 > a_{co}^8), (a_{co}^8 > a_{co}^9)\} \]
\[ O_{co}^{1-4} = \{(a_{co}^{10} > a_{co}^{11}), (a_{co}^{11} > a_{co}^{12})\} \]
\[ O_{co}^{1-5} = \{(a_{co}^{14} > a_{co}^{15}), (a_{co}^{15} > a_{co}^{16})\} \]
\[ O_{co}^{1-6} = \{\} \]
\[ O_{co}^{1-7} = \{(a_{co}^{18} > a_{co}^{19}), (a_{co}^{19} > a_{co}^{20})\} \]
\[ O_{co}^{1-8} = \{\} \]
\[ O_{co}^{1-9} = \{\} \]
\[ O_{co}^{1-10} = \{\} \]
\[ O_{co}^{1-11} = \{\} \]
\[ O_{co}^{1-12} = \{\} \]

\[ O_{co}^{1-13} = \{(a_{co}^{16} > a_{co}^{17}), (a_{co}^{17} > a_{co}^{22}), (a_{co}^{17} > a_{co}^{23}), (a_{co}^{20} > a_{co}^{21}), (a_{co}^{21} > a_{co}^{22}), (a_{co}^{22} > a_{co}^{23}), (a_{co}^{22} > a_{co}^{23}), (a_{co}^{23} > a_{co}^{24}), (a_{co}^{13} > a_{co}^{24})\} \]

1/31/2003

Knowledge Acquisition by Heterogeneous Planners

24
Example

Solution Plan - Partition $\text{co}^1$

- Groups
  $\langle A_{\text{co}}^{1-1}, O_{\text{co}}^{1-1} \rangle$, $\langle A_{\text{co}}^{1-2}, O_{\text{co}}^{1-2} \rangle$, $\langle A_{\text{co}}^{1-3}, O_{\text{co}}^{1-3} \rangle$, $\langle A_{\text{co}}^{1-4}, O_{\text{co}}^{1-4} \rangle$

  all translate to the action
  $\text{Insert\_wheel}_3(?x,?y)$
Example

**Partition**\textsubscript{co} \textsuperscript{1} – New Actions

- Group \(<A_{co}^{1-5},O_{co}^{1-5}>\) cannot be translated
- It has the following Preconditions, Add list, Delete List & Constraints
  
  \[
  \text{Pre} (\langle A_{co}^{1-5},O_{co}^{1-5} \rangle) = \text{Clear}_{co} (\text{Eng}), \text{Clear}_{co} (\text{Eng}\_slot)
  \]
  \[
  \text{Add}(\langle A_{co}^{1-5},O_{co}^{1-5} \rangle) = \text{In}_{co} (\text{Eng}, \text{Eng}\_slot)
  \]
  \[
  \text{Del}(\langle A_{co}^{1-5},O_{co}^{1-5} \rangle) = \text{Clear}_{co} (\text{Eng}\_slot)
  \]
  \[
  \text{Con}(\langle A_{co}^{1-5},O_{co}^{1-5} \rangle) = --
  \]
- The new action this would form is
  \[
  a_{3}^{1}(?x,?y)
  \]
  \[
  \text{Pre}(a_{3}^{1}) = \text{Clear}_{3} (?x), \text{Clear}_{3} (?y)
  \]
  \[
  \text{Add}(a_{3}^{1}) = \text{In}_{co}(?x,?y)
  \]
  \[
  \text{Del}(a_{3}^{1}) = \text{Clear}_{3} (?y)
  \]
  \[
  \text{Con}(a_{3}^{1}) = -
  \]
- The similarity index is \(SI(a_{3}^{1},\text{Fix}\_\text{together}_{3}) = 0.4\)
- The element \(<A_{co}^{1-7},O_{co}^{1-7}>\) forms a new action, \(a_{3}^{3}\), that is equal to \(a_{3}^{1}\). Therefore, \(a_{3}^{3} = a_{3}^{1}\) and \(SI(a_{3}^{3},\text{Fix}\_\text{together}_{3}) = 0.4\)
Example

Partition\textsubscript{co} \textsuperscript{1} – New Actions

• Similarly, <A\textsubscript{co} \textsuperscript{1-6}, O\textsubscript{co} \textsuperscript{1-6}> and <A\textsubscript{co} \textsuperscript{1-8}, O\textsubscript{co} \textsuperscript{1-8}> form actions a\textsubscript{3} \textsuperscript{2} and a\textsubscript{3} \textsuperscript{4} which are equal and their similarity indices are
  \[ \text{SI}(a\textsubscript{3} \textsuperscript{2}, \text{Fix}\textunderscore{together}\textsubscript{3}) = 0.2 \]
  \[ \text{SI}(a\textsubscript{3} \textsuperscript{4}, \text{Fix}\textunderscore{together}\textsubscript{3}) = 0.2 \]

• Finally, <A\textsubscript{co} \textsuperscript{1-9}, O\textsubscript{co} \textsuperscript{1-9}>, <A\textsubscript{co} \textsuperscript{1-10}, O\textsubscript{co} \textsuperscript{1-10}>, <A\textsubscript{co} \textsuperscript{1-11}, O\textsubscript{co} \textsuperscript{1-11}>, <A\textsubscript{co} \textsuperscript{1-12}, O\textsubscript{co} \textsuperscript{1-12}> form the new actions a\textsubscript{3} \textsuperscript{5}(?w,?x,?y,?z), a\textsubscript{3} \textsuperscript{6}(?en,?bt,?es,?bs), a\textsubscript{3} \textsuperscript{7}(?en,?bt,?es,?bs) and a\textsubscript{3} \textsuperscript{8}(?w,?x,?y,?z) whose similarity indices are as below.
  \[ \text{SI}(a\textsubscript{3} \textsuperscript{5}, \text{Fix}\textunderscore{together}\textsubscript{3}) = 0.125 \]
  \[ \text{SI}(a\textsubscript{3} \textsuperscript{6}, \text{Body}\textunderscore{assembly}\textsubscript{3}) = 0.067 \]
  \[ \text{SI}(a\textsubscript{3} \textsuperscript{7}, \text{Base}\textunderscore{assembly}\textsubscript{3}) = 0.75 \]
  \[ \text{SI}(a\textsubscript{3} \textsuperscript{8}, \text{Base}\textunderscore{assembly}\textsubscript{3}) = 0.105 \]

• The similarity index of the entire set is
  \[ \text{SI}(A\textsubscript{3} \textsuperscript{1}, \text{Actions}\textsubscript{3}) = 0.281 \]
Example

Solution Plan – Calculation Results

- Below are a few of the highest similarity indices for the remaining partitions.

\[
\begin{align*}
SI(A_3^2, \text{Actions}_3) &= 0.374 \\
SI(A_3^3, \text{Actions}_3) &= 0.348 \\
SI(A_3^4, \text{Actions}_3) &= 0.523 \\
SI(A_3^5, \text{Actions}_3) &= 0.344 \\
SI(A_3^6, \text{Actions}_3) &= 0.515 \\
SI(A_3^7, \text{Actions}_3) &= 0.348 \\
SI(A_3^8, \text{Actions}_3) &= 0.579 \\
SI(A_3^9, \text{Actions}_3) &= 0.436 \\
SI(A_3^{10}, \text{Actions}_3) &= 0.467
\end{align*}
\]

- The set \(A_3^8\) is most similar to the existing knowledge in Planner_3.
Example

Planner3 - New Actions

A₃³₈= {a₁³, a₂³, a₃³}, contains 3 new actions described as follows.

a₁³(x,y) = a₂³(x,y)

Pre(a₁³) = Pre(a₂³) = Clear₃(x), Clear₃(y)

Add(a₁³) = Add(a₂³) = Inco₃(x,y), Fastened₃(x,y)

Del(a₁³) = Del(a₂³) = Clear₃(y)

Con(a₁³) = Con(a₂³) = -

a₃³(en, bt, es, bs, bs₁, cr)

Pre(a₃³) = Inserted_wheel₃(w₁, ae₁),
Inserted_wheel₃(w₂, ae₂), Inserted_wheel₃(w₃, ae₃),
Inserted_wheel₃(w₄, ae₄), Fastened₃(en, es),
Fastened₃(bt, bs), Inco₃(en, es), Inco₃(bt, bs),
Wheel₃(w₁), Wheel₃(w₂),

Add(a₃³) = Connected₃(en, bt), Assembled_base_car₂₃

Del(a₃³) = --

Con(a₃³) = Wheel₃(w₃), Wheel₃(w₄), Axle_end₃(ae₁),
Axle\ end₃(ae₂), Axle\ end₃(ae₃), Axle\ end₃(ae₄),
Engine₃(en), Battery₃(bt), Engine\ slot₃(es),
Battery\ slot₃(bt), Base₃(bs₁), Car₃(co(Car₂))
### Example

**Planner}_3 - New Mappings**

<table>
<thead>
<tr>
<th>Planner}_3</th>
<th>Common Ontology</th>
</tr>
</thead>
<tbody>
<tr>
<td>In}_3(x,y)</td>
<td>In}_co(x,y)</td>
</tr>
<tr>
<td>Car}_3(x)</td>
<td>Car}_co(x)</td>
</tr>
<tr>
<td>Pickup_align_insert_attach_in}_3(x,y)</td>
<td>Pickup}_co(x) &gt; Align}_co(x,y) &gt; Insert}_co(x,y) &gt; Attach}_in}_co(x,y)</td>
</tr>
<tr>
<td>Connect_wires_eng_battAssembly_wheelAssembly_baseAssembly}_3(x,w1,w2,w3,w4,ae1,ae2,ae3,ae4,en,bt,es,bs,bs1,cr)</td>
<td>Connect_wires}_co(en,bt) &gt; Eng_battAssembly}_co(en,bt,es,bs) &gt; WheelAssembly}_co(w1,w2,w3,w4,ae1,ae2,ae3,ae4) &gt; BaseAssembly}_co(bs1,cr)</td>
</tr>
</tbody>
</table>
Example

Planner$^3$ - Final Solution

Plan$_3$

$A_3 =$

$a^1_3 = \text{Insert} \_ \text{wheel}_3(\text{Wheel1}, \text{Axle} \_ \text{end}1),$

$a^2_3 = \text{Insert} \_ \text{wheel}_3(\text{Wheel2}, \text{Axle} \_ \text{end}2),$

$a^3_3 = \text{Insert} \_ \text{wheel}_3(\text{Wheel3}, \text{Axle} \_ \text{end}3),$

$a^4_3 = \text{Insert} \_ \text{wheel}_3(\text{Wheel4}, \text{Axle} \_ \text{end}4),$

$a^5_3 = \text{Pickup} \_ \text{align} \_ \text{insert} \_ \text{attach}_3(\text{Eng}, \text{Eng} \_ \text{slot}),$

$a^6_3 = \text{Pickup} \_ \text{align} \_ \text{insert} \_ \text{attach}_3(\text{Batt}, \text{Batt} \_ \text{slot}),$

$a^7_3 = \text{Connect} \_ \text{wires} \_ \text{eng} \_ \text{batt} \_ \text{assembly} \_ \text{wheel} \_ \text{assembly} \_ \text{base} \_ \text{assembly}_3(\text{Wheel1}, \text{Wheel2}, \text{Wheel3}, \text{Wheel4},$

$\text{Axle} \_ \text{end}1, \text{Axle} \_ \text{end}2, \text{Axle} \_ \text{end}3, \text{Axle} \_ \text{end}4, \text{Eng}, \text{Batt},$

$\text{Eng} \_ \text{slot}, \text{Batt} \_ \text{slot}, \text{Base}, \text{Car}2)$

$O_3 = \text{before}(a^1_3, a^7_3), \text{before}(a^2_3, a^7_3), \text{before}(a^3_3, a^7_3), \text{before}(a^4_3, a^7_3),$

$\text{before}(a^5_3, a^7_3), \text{before}(a^6_3, a^7_3)$
Related Work

Knowledge Interchange Interface - A formal framework for knowledge sharing between planners.

- Given Plan_{co}, the KII learns new knowledge from it as follows.
  - If action a^k_{co} contains predicates that are always true or false, it is combined with one or more actions to eliminate these predicates. The resulting composite action is added to Planner_i.
  - All other actions are added without any change to Planner_i.
  - All new predicates in the actions are added to Pred_i.
  - Translators are formulated for the actions and predicates and added to KB_i.

- Drawbacks
  - A subplan in Plan_{co} that have an equivalent action in Planner_i, but no translator get mapped to new actions, causing duplication of knowledge in Planner_i.
  - Does not handle combinations of subplans in a flexible manner.
  - Adds the same knowledge to planners, regardless of the level of abstraction.
Related Work

Learning through Exploration and Experimentation - improve the models for planning by using failure-driven experimentation with the environment

- Construction by direct analogy - creates new operators based on operators that are already defined in the system. This is done by direct analogy through the types of objects that the operators are applied to.
- Creation of micro-operators – creates a new operator that performs some subaction of an existing operator.
  - Experimentation is used to determine which preconditions and effects can be discarded in the new operator.
  - A existing action is broken into a sequence of subactions.
- Drawbacks
  - Refines existing knowledge in the system.
  - Does not learn anything new from external sources.
Related Work

Explanation-Based Learning - generalize a set of given examples into descriptions of concepts.

- Learning composite rules - alters the search space that is explored by combining existing rules to form new, possibly redundant but more efficient, rules. Improves the performance of a system without the introduction of new knowledge.
- Learning search control knowledge - altering the method in which the search is conducted. Speeds up the search of the concept hypothesis space by formulating heuristics that index the operators in an appropriate order.

Plan Mining - used to discover significant knowledge from a large database of plans such as airline schedules etc.

- Generalization-based divide-and-conquer - used to find high-level, highly regular sequential patterns along more than one dimension.
Conclusion

• The knowledge acquisition methodology allows a planner in a collection of networked planners to update its domain theory using solution plans that is receives from its peers.

• The information (predicates and actions) contained in a solution plan that cannot be translated is used to extend the knowledge base of the receiving planner.

• The addition of the necessary translators for the new knowledge means that it can be used to solve problem and communicate queries and solution plans in the future.

• The methodology guarantees that this information is correct in the domain of the planner and it is the most similar to the existing knowledge in the planner.