An Experimental Evaluation of some Design Knowledge Compilation Mechanisms

D. C. Brown & M. B. Spillane

AIRG, CS Dept., WPI, Worcester, MA 01609, USA.
OUTLINE

DSPL - routine design.

BPEN - the domain.

COMPIlATION

- Inheritance.
- Absorption.
- Relaxation.
- Combined System.

EXPERIMENT

- Design.
- Results.
Routine Design

- Done many, many times before
- No innovation
- Designer knows how to solve the problem
- Knowledge is in the form of plans
- May have several alternate plans for solving the same subproblem
- Design attributes are known

DSPL

- Design Specialists and Plans Language
- Reflects the Designer’s decomposition of the problem
- Hierarchical structure of design agents
- Specialists, Plans, Tasks, Steps, Constraints
BPEN

1. Head
2. Refill
3. Spring
4. Body
5. Pushing Component
(CONSTRAINT
   (NAME TwiceMinThick<LgNotchHt)
   (USED-BY PushCompLgNotchHt)
   (SOFT-CONSTRAINT T)
   (CURRENT-PREFERENCE .0022)
   (RELAXATION-LIMIT (BodySizeOD -- BodySizeId))
   (COMMENT "Example of Soft Constraint")
   (FAILURE-MESSAGE "Large Notch Ht is too small")
   (FAILURE-SUGGESTION
      (SUGGEST (RELAX PushCompLgNotchHt by
                   FAILURE-AMOUNT))
   )
)
(BODY
   (KNOWN
      Factor 2
      BodySizeOD
      (KB-FETCH 'PBody 'BodySizeOD)
      BodySizeID
      (KB-FETCH 'PBody 'BodySizeID))
   (TEST
      ((Factor * (BodySizeOD - BodySizeID))
       = LgNotchHt))))
<table>
<thead>
<tr>
<th>Requirement</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Ridgedisttol</td>
<td>1</td>
</tr>
<tr>
<td>Lgnotchgaptol</td>
<td>1</td>
</tr>
<tr>
<td>Headotolerance</td>
<td>1</td>
</tr>
<tr>
<td>Refillridgelen</td>
<td>1.25</td>
</tr>
<tr>
<td>Refilltipod</td>
<td>0.002</td>
</tr>
<tr>
<td>Refillod</td>
<td>0.012</td>
</tr>
<tr>
<td>Refillbodylen</td>
<td>3.1</td>
</tr>
<tr>
<td>Heado shaftlen</td>
<td>2</td>
</tr>
<tr>
<td>Headtid</td>
<td>0.22</td>
</tr>
<tr>
<td>Refillbodymaterial</td>
<td>Flexibleplastic</td>
</tr>
<tr>
<td>Refilltipmaterial</td>
<td>Steel26</td>
</tr>
<tr>
<td>Pushcompmaterial</td>
<td>Rigidplastic</td>
</tr>
<tr>
<td>Bodymaterial</td>
<td>Rigidplastic</td>
</tr>
<tr>
<td>Headmaterial</td>
<td>Rigidplastic</td>
</tr>
<tr>
<td>Pushcompendlen</td>
<td>0.2</td>
</tr>
</tbody>
</table>
Deep Knowledge.
Surface Knowledge.

Process of improving the surface knowledge through experience in order to increase the efficiency of the problem-solving.

Deep to Surface       -- Formation.
Surface to Surface    -- Adjustment.

Absorption (& Relaxation) System:
Constraints disappear (or relax).

Inheritance System:
Constraints appear.
CONSTRANT INHERITANCE

Generic Object Knowledge Base (GOKB)

Knowledge Found

Constraint Inheritor

Examine

GOKB Reasoner

Expectation Violation

KB-Store

DSPL

Design Data Base

Inherit Constraint

Relationships

Shape

Expected Value

Expected Value
CONSTRAINT ABSORPTION

Detection of redundant constraining knowledge and its removal.

After absorption, the constraint need not be tested again.

Condition:
- always true.
- subsumed by prior knowledge.

Success-driven Knowledge Refinement.

Uses History to notice candidates.
CONSTRAINT RELAXATION

Hard Constraints:

- Normal DSPL.
- Must be true for success of design.

Soft Constraints:

- Allows range of valid values.
- Temporary relaxation when constraint fails and there is suggestion to try relaxing.
**PREDICTED EFFECTS**

Constraint Inheritance:

- Causes failures to be found earlier.
- Causes different paths to be used.
- Prevents incorrect designs

Constraint Absorption:

- Creates shorter paths.
- Execution time is reduced.

Constraint Relaxation:

- Removes barriers to completion.
- Reduces failures & backtracking.
- Execution time is reduced.
COMBINED SYSTEM

Constraint Inheritance System

Absorption and Relaxation System
THE EXPERIMENT

△ 3 phases:
- Pre-compilation
- Compilation
- Post-compilation

Compilation Set of Design Requirements

Time

Control Set of Design Requirements
EVALUATION CRITERIA

▲ System:
  - CPU time
  - Number of each agent used
  - Number of each agent failed
  - Total number of agents used
  - Total number of agents in the system
  - Ratio of failed agents to successful agents

▲ Compilation:
  - Amount of compilation
  - What type and where in the process it occurred

▲ Design object:
  - Ensured the final design met all constraints
## DURING COMPILATION

Non-Compiled & Compilation Phase Results

<table>
<thead>
<tr>
<th>Test Run</th>
<th>Avg. Time Non-Comp.</th>
<th>Avg. Time Compilation</th>
<th>Change in Time</th>
<th>% Change in Time</th>
<th>Change in Design Path</th>
</tr>
</thead>
<tbody>
<tr>
<td>Inheritance</td>
<td>510.4</td>
<td>710.8</td>
<td>+200.4</td>
<td>39.26% slow down</td>
<td>+35.8 agents</td>
</tr>
<tr>
<td>Relaxation &amp; Absorption</td>
<td>510.4</td>
<td>786.4</td>
<td>+276</td>
<td>54.07% slow down</td>
<td>+1.4 agents</td>
</tr>
<tr>
<td>Combined</td>
<td>510.4</td>
<td>1064.6</td>
<td>+554.2</td>
<td>108.58% slow down</td>
<td>+36.6 agents</td>
</tr>
</tbody>
</table>

On average it is ≈100% slower during compilation.
PRE vs POST COMPILATION

Pre & Post-Compilation Phase Results

<table>
<thead>
<tr>
<th>Test Run</th>
<th>Avg. Time Pre-Comp.</th>
<th>Avg. Time Post-Comp.</th>
<th>Change in Time</th>
<th>% Change in Time</th>
<th>Change in Design Path</th>
</tr>
</thead>
<tbody>
<tr>
<td>Inheritance</td>
<td>554.7</td>
<td>501.2</td>
<td>-53.5</td>
<td>9.64% speed up</td>
<td>+3.1 agents</td>
</tr>
<tr>
<td>Relaxation &amp; Absorption</td>
<td>554.7</td>
<td>471.7</td>
<td>-83.0</td>
<td>14.96% speed up</td>
<td>-2.9 agents</td>
</tr>
<tr>
<td>Combined</td>
<td>554.7</td>
<td>529.8</td>
<td>-24.9</td>
<td>4.49% speed up</td>
<td>-4.5 agents</td>
</tr>
</tbody>
</table>

\[ 3.1 - 2.9 \neq -4.5 \]

Constraints appear.

Constraints disappear.

Different paths used.