CARPENTRY
COMPILER

Chenming Wu (Tsinghua, University of Washington)
Haisen Zhao (Speaker, University of Washington)
Chandrakana Nandi (University of Washington)
Jeffrey I. Lipton (University of Washington)
Zachary Tatlock (University of Washington)
Adriana Schulz (University of Washington)
Manufacturing Advances
A design completely determines a fabrication process.
From Design to Fabrication Plan

Design

Band saw
Jig saw
Chop saw
Table saw

Least Fab Error
Least Fab Time
Least Material Cost
Fabrication and Design

Fabrication and design are inherently related.

Design

Fabrication

- Defines the space of what can be physically realized

- Affects the fabrication performance

Optimization: physical behavior

Optimization: production cost
Insight: Designs and Fabrication Plans are Programs!

Design

A sequence of geometric construction operations *(Code)*

Fabrication

A sequence of physical instructions *(Code)*

This insight inspires us to draw ideas from computer systems!
Inspiration: Instruction Set Architectures (ISA)

- An interface between software and hardware
- Enable the independent development

Carpentry Design and Fabrication

- Vast application scope
- Appropriate level of complexity for initiating research
Goal of our system:

- Ensure design is driven by available fabrication processes
- Decouple design and fabrication
HELM: Hardware Extensible Languages for Manufacturing

- Inspiration: CAD languages

High-Level Code:

```plaintext
1. Box001 = Make_Stock(457.2, 38.1, 88.9);
2. MyLine000 = Line(457.2, 0, 435.203, 38.1);
3. Sketch = Make_Sketch(
   Query_Face_By_Closest_Point(Box001, 228.6, 19.05, 88.9),
   Geometry(MyLine000),
   Constraint(Coincident(Start(MyLine000), End(
      Query_Edge_By_Closest_Point(Box001, 228.6, 0, 88.9)));
   PointOnObject(End(MyLine000), Query_Edge_By_Closest_Point(
      Box001, 228.6, 38.1, 88.9), Angle(Start(
      Query_Edge_By_Closest_Point(Box001, 457.2, 19.05, 88.9), Start(
      MyLine000), 30)));
4. Cut = Make_Cut(Box001, Sketch, 0);
5. MyLine001 = Line(0, 38.1, 21.997, 0);
6. Sketch001 = Make_Sketch(
   Query_Face_By_Closest_Point(Cut, 228.6, 19.05, 88.9),
   Geometry(MyLine001),
   Constraint(Coincident(Start(MyLine001), End(
      Query_Edge_By_Closest_Point(Cut, 0, 19.05, 88.9)));
   PointOnObject(End(MyLine001), Query_Edge_By_Closest_Point(
      Cut, 228.6, 0, 88.9), Angle(End(Query_Edge_By_Closest_Point(
      Cut, 0, 19.05, 88.9)), Start(MyLine001), 30));
```
HELM: Hardware Extensible Languages for Manufacturing

- Inspiration: CAD languages
- Subtractive: map to woodworking
- Parametric: design optimization

```
Make_Stock (6, 2, 4)
Make_Cut (6, 30°)
```

```
Make_Stock (6, 2, 4)
Make_Cut (6, 45°)
```
HELM: Hardware Extensible Languages for Manufacturing

- Inspiration: CAD languages
- Subtractive: map to woodworking
- Parametric: design optimization
- Verifier: ensures manufacturability
HELPM: Hardware Extensible Languages for Manufacturing

- Process specific: be followed to generate one design
- Extensible to more fabrication hardware

2x4_lumber 30°

Setup_Chopsaw (30°, 0, 6)
Chopsaw (2x4_lumber)

Setup_Jigsaw (30°, 0, 6)
Jigsaw (2x4_lumber)
HELM: Hardware Extensible Languages for Manufacturing

- Design validation
- Fabrication optimization

**Make_Stock** (6, 2, 4)
**Make_Cut** (6, 30°)

**Setup_Chopsaw** (30°, 0, 6)
**Chopsaw** (2x4_lumber)

Design
High-Level HELM

Compiler

Fabrication
Low-Level HELM
Challenges of HELM Compiler

1. Long sequence of interdependent steps
   - need a data structure to represent such combinatorial space

Choose material stocks

Pack the parts

Define Cuts Tools/Orders on Stock
Challenges of HELM Compiler

1. Long sequence of interdependent steps
   - need a data structure
2. Multiple (conflicting) fabrication costs
   - multi-objective optimization

- Material cost: 2.3
  - Fabrication time: 8.5

- Material cost: 2.95
  - Fabrication time: 5
Challenges of HELM Compiler

1. Long sequence of interdependent steps
   - need a data structure
2. Multiple (conflicting) fabrication costs
   - multi-objective optimization
Learning from Programming Languages

- E-graphs [Joshi et al. 2002; Tate et al. 2009]
  - **Compactly** and **efficiently** represent large scale equivalence programs
  - Define a data-structure that make this search **tractable**

\[
3x + \frac{1}{3}y - \frac{(x+2(x+y))}{3}
\]

simplify

\[
2x - \frac{1}{3}y
\]

\[
a + b = b + a
\]

\[
(a + b)/c = b/c + a/c
\]

.....

Rewrite rules
E-Graphs for Carpentry Compiler

Equivalence classes (e-classes)

design

programs
E-Graphs for Carpentry Compiler

#Fabrication plans: 8830

((20*20+40)*20)+30
Challenges of E-Graphs for Carpentry Compiler

1. Linearity: reuse-ability of variables

X = 10;
Y = X + 5;
Z = X + Y;

No need linearity constraints

Fabrication needs linearity constraints

Common E-Graphs Applications

Carpentry Compiler
Challenges of E-Graphs for Carpentry Compiler

1. Linearity: reus-ability of variables
2. Rewrite rules

\[
\begin{align*}
  a + b &= b + a \\
  (a + b)/c &= a/c + b/c \\
  a - b &= a + (-b) \\
  a \times b &= b \times a
\end{align*}
\]

.....

Syntactic rewrite rules

Common E-Graphs Applications

Carpentry Compiler
Challenges of E-Graphs for Carpentry Compiler

1. Linearity: reuse-ability of variables
2. Rewrite rules
   • A geometric method to populate and modify the e-graph

1: Pack pieces onto stocks
2: Define Cuts Tools/Orders on Stock
Challenges of E-Graphs for Carpentry Compiler

1. Linearity: reuse-ability of variables
2. Rewrite rules
   - A geometric method to populate and modify the e-graph
Challenges of HELM Compiler

1. Long sequence of interdependent steps
   - need a data structure
     - Apply **E-Graphs** to this carpentry domain.

2. Multiple (conflicting) fabrication costs
   - multi-objective optimization
   - extract the optimal solution from E-graph
Extract the optimal solution from E-graph

Single objective:
- Running time or Memory cost

Multiple objectives:
- Material cost
- Fabrication time
- Precision

Common E-Graphs Applications

Carpentry Compiler

Least Fab Time

Least Fab Error

Least Material Cost
Extract the optimal solution from E-graph

- Genetic algorithm for multi-objective optimization in the E-Graphs

Cross-over
Extract the optimal solution from E-graph

- Genetic algorithm for multi-objective optimization in the E-Graphs

**Mutation**

Randomly expand e-graphs
Challenges of HELM Compiler

1. Long sequence of interdependent steps
   - need a data structure
   ❖ Apply **E-Graphs** to this carpentry domain. ✓

2. Multiple (conflicting) fabrication costs
   - multi-objective optimization
     - extract the optimal solution from E-graph
   ❖ **Genetic algorithm** for multi-objective optimization ✓

Results
Design Results by Domain Experts
Optimization Results – Expert Comparison

9.C: Bookcase

Material Cost: 3.0
Precision: 1.0
Fabrication Time: 9.5

Experts

Material Cost: 2.3
Precision: 1.0
Fabrication Time: 8.5

Our solution

Material Cost: 1.3
Precision: 1.03
Fabrication Time: 22.5

9.I: Flowerpot

Material: 1.3
Precision: 1.06
Fabrication Time: 23.0

Experts

Material: 1.3
Precision: 1.03
Fabrication Time: 22.5

Our solution

Material Cost: 1.3
Precision: 1.03
Fabrication Time: 22.5

Experts

Material Cost: 2.3
Precision: 1.0
Fabrication Time: 8.5

Our solution

Material Cost: 1.3
Precision: 1.03
Fabrication Time: 22.5

Experts

Material Cost: 3.0
Precision: 1.0
Fabrication Time: 9.5

Our solution

Material Cost: 1.3
Precision: 1.03
Fabrication Time: 22.5

Experts

Material Cost: 2.3
Precision: 1.0
Fabrication Time: 8.5

Our solution

Material Cost: 1.3
Precision: 1.03
Fabrication Time: 22.5

Experts

Material Cost: 3.0
Precision: 1.0
Fabrication Time: 9.5

Our solution

Material Cost: 1.3
Precision: 1.03
Fabrication Time: 22.5
Optimization Results – Expert Comparison

- A tool that can be used for **non-experts**
- Help **expert carpenters** search this large combinatorial space

<table>
<thead>
<tr>
<th>Experts</th>
<th>Mat. Cost</th>
<th>Precision</th>
<th>Fab. Time</th>
<th>Mat. Cost</th>
<th>Precision</th>
<th>Fab. Time</th>
</tr>
</thead>
<tbody>
<tr>
<td>Adirondack chair</td>
<td>4.0</td>
<td>1.05</td>
<td>6.5</td>
<td>3.85</td>
<td>1.02</td>
<td>4.5</td>
</tr>
<tr>
<td>Drafting Table</td>
<td>9.8</td>
<td>1.29</td>
<td>16</td>
<td>9.05</td>
<td>1.17</td>
<td>16</td>
</tr>
<tr>
<td>Book case</td>
<td>3.0</td>
<td>1.0</td>
<td>9.5</td>
<td>2.3</td>
<td>1.0</td>
<td>8.5</td>
</tr>
<tr>
<td>Bird house</td>
<td>2.7</td>
<td>1.57</td>
<td>32</td>
<td>2.7</td>
<td>1.57</td>
<td>31.5</td>
</tr>
<tr>
<td>Dining room chair</td>
<td>4.3</td>
<td>1.06</td>
<td>13.5</td>
<td>3.75</td>
<td>1.02</td>
<td>10.5</td>
</tr>
<tr>
<td>Bench</td>
<td>8.0</td>
<td>1.06</td>
<td>7</td>
<td>5.3</td>
<td>1.05</td>
<td>10</td>
</tr>
<tr>
<td>Coffee table</td>
<td>10.48</td>
<td>5.72</td>
<td>69.5</td>
<td>10.43</td>
<td>5.06</td>
<td>54.5</td>
</tr>
<tr>
<td>Flower pot</td>
<td>1.3</td>
<td>1.06</td>
<td>23</td>
<td>1.3</td>
<td>1.03</td>
<td>22.5</td>
</tr>
<tr>
<td>Z-table</td>
<td>9.15</td>
<td>1.14</td>
<td>17</td>
<td>9.05</td>
<td>1.09</td>
<td>16.5</td>
</tr>
</tbody>
</table>
Trade-off Exploration

- Return **multiple** solutions with different **trade-offs**
- Allow users to **interactive explore** the solution space
Physical Models

Bird house

Toy car

Book case
Validation

Additional results

Statistics

Large scale tests
Conclusion

HELM: Hardware Extensible Language for Manufacturing

- Key insight: fabrication plans are programs
- New DSLs for HL-HELM and LL-HELM
- Applied and extended compiler technique
- Multi-objective optimization
Future work

• Carpentry Extensions

Grain orientation

Fabrication uncertainty
Future work

- Carpentry Extensions
- Other processes

Additive Manufacturing  Subtractive Manufacturing  Assembly
Future work

- Carpentry Extensions
- Other processes
- Design Optimization
Future work

- Carpentry Extensions
- Other processes
- Design Optimization
- Scheduling for future workshops

Thank you!

Acknowledgement:
Anonymous reviewers, Dan Grossman, Benjamin T. Jones, Colin Topper, Victor Wu, Mary Mattsen, Guoxin Fang, Liang He, Max Willsey and Yuxuan Mei

National Science Foundation: 1813166 and 1644558

https://grail.cs.washington.edu/projects/carpentrycompiler/
Hardware Extensible Language for Manufacturing
1. Key insight: fabrication plans are programs
2. New DSLs for HL-HELM and LL-HELM
3. Applied and extended compiler technique
4. Multi-objective optimization

Future work:
1. Carpentry Extensions
2. Other processes
3. Design Optimization
4. Scheduling for future workshops