


# Constraints



# Project 2 ideas



- Window system

- <http://www.berlin-consortium.org/>

- BeOS ([www.be.com](http://www.be.com))

- Toolkits

- Subarctic ([www.cs.cmu.edu/~hudson](http://www.cs.cmu.edu/~hudson))

- GIMP


- Motif, OpenLook (somewhat boring)

# Multi-way constraints



- Allows info to flow in both directions
  - $A = B + C$

More powerful,  
Less understandable (sometimes)



- May be over constrained
- May be under constrained

# Constraint Hierarchies (Constraint Strengths?)



- Blue/DeltaBlue use them
- Over-constrain
  - Use strengths
  - Implied "weakest" "stay" constraint

# Intuitive Solution to Constraint Hierarchies



# More Formal Solution



- Assign strength levels to constraints
  - $C_0, C_1, \dots, C_n$
- Solution: mapping of values onto variables
- "Admissible" solution

"Best Admissible" solution



DeltaBlue Comparator:  
"local-predicate-better"



# Solving constraint hierarchies



- Each constraint has multiple methods

# Two part solution



- Change constraints: planning
- Change values: update using plan

# Graph notation for plans



# An incremental planning algorithm (DeltaBlue)

---

- Start: current plan +  
change to the constraints
- Goal: incrementally find new plan
- For removed constraint

# Adding constraints



Key to adding constraints:  
Walkabout Strength

A thick, horizontal yellow brushstroke underline that spans the width of the text above it, with a slightly textured, hand-painted appearance.

# Add Constraints using Walkabout Strength

