ORC Layout: Adaptive GUI Layout with OR-Constraints

Yue Jiang  
University of Maryland, College Park

Ruofei Du  
Google, San Francisco

Christof Lutteroth  
University of Bath, Bath, United Kingdom

Wolfgang Stuerzlinger  
SIAT, Simon Fraser University, Vancouver, Canada
Motivation

Need to design different GUI layouts for different screen sizes, orientations, and aspect ratios.
Limitation: cannot restrict positions and relative sizes
Constraint-based Layout

Constraints:
- Same size
  \[ \text{Size(\text{Red1})} = \text{Size(\text{Red2})} = \ldots \]
- Same height as above
  \[ \text{Height(\text{Blue})} = \text{Height(\text{Red})} \]
- Double width as above
  \[ \text{Width(\text{Blue})} = \text{Width(\text{Red})} \times 2 \]

Limitations:
1. Widgets cannot move relative to other ones.
2. Device diversity a long-term challenge.
Goal: Unify constraint-based and flow layouts

Approach: OR-constraints

Input:
1. A set of constraints
2. Widget min/pref/max sizes
3. Window size

Output:
1. Widget sizes
2. Widget positions
OR-Constraints

Constraint1  OR Constraint2  OR Constraint3 …

Soft        Soft        Soft

Hard

• **Hard Constraints** must be satisfied.
• **Soft Constraints** are satisfied if possible. Their importance depends on weights.
OR-Constraints

to the right of the previous widget
(larger weight)
OR
at the beginning of the next row
(smaller weight)
Z3 Solver

OR Constraints $\rightarrow$ more branches

Microsoft Z3 Solver:

• Can solve OR-constraints
• Support incremental solving (fast if #widget not too large)
ORC Patterns

Low-level constraints tedious and error prone

Better approach:
Designers → choose a template & modify parameters
System → automatically maintain low-level constraints
Pattern #1: Connected Layout Pattern

Top toolbar widgets → left toolbar
Pattern #1: Connected Layout Pattern

Left toolbar widgets → top toolbar
Pattern #2: Balanced Flow Layout Pattern

6 widgets → Each row has 1 OR 2 OR 3 OR 6 widgets in the toolbar
Pattern #3: Alternative Positions Pattern

Top Toolbar
(Weights depend on which one you prefer)

OR

Left Toolbar
Pattern #4: Flowing Widgets around a Fixed Area
Pattern #5: Optional Layout Pattern

Less important
Pattern #6: Alternative Widget Layout Pattern
Limitations

- Patterns restrict what designers can create.
- Non-interactive solving time for larger number of widgets
Conclusion

• ORC Layouts
  • Introduce OR-constraints
  • Unify flow & constraint-based layouts
  • Enrich GUI layout design space
Co-authors

Ruofei Du
Google

Christof Lutteroth

Wolfgang Stuerzlinger
Contributions:

• Add OR-constraints to standard hard/soft constraint systems.
• Adapt layouts to screens with different screen sizes, orientations, and aspect ratios with only a single specification.
• Unify flow & constraint-based layouts.

Yue Jiang: yuejiang@cs.umd.edu
https://cs.umd.edu/~yuejiang

Thank you!