

Breaking the Software Development Roadblock: Continuous Software Enhancement By Design Maintenance

Ira Baxter

Dec 13, 2001

**Workshop on New Visions for Software Design and Productivity:
Research and Applications**



***Design* must be THE development product**

- **Successful Software Systems have Long Lives**
 - Fundamental Issue: how to *enhance* them, not to build them!
 - Key Idea: Must not forget DESIGN: specification, architecture, rationale
 - ... and yet, we always do ... --> reverse engineering business
- **Require design update to be on path to obtain code**
 - Costs must not seriously hinder engineering
 - Must provide significant value during development/maintenance
- **Must be mechanically processable**
 - Enables Automation in Validation and Generation
- **Issue: What *is* a Design?**
 - Structure that can answer most questions about product
What? [purpose] How? [implemented] Why? [that method]

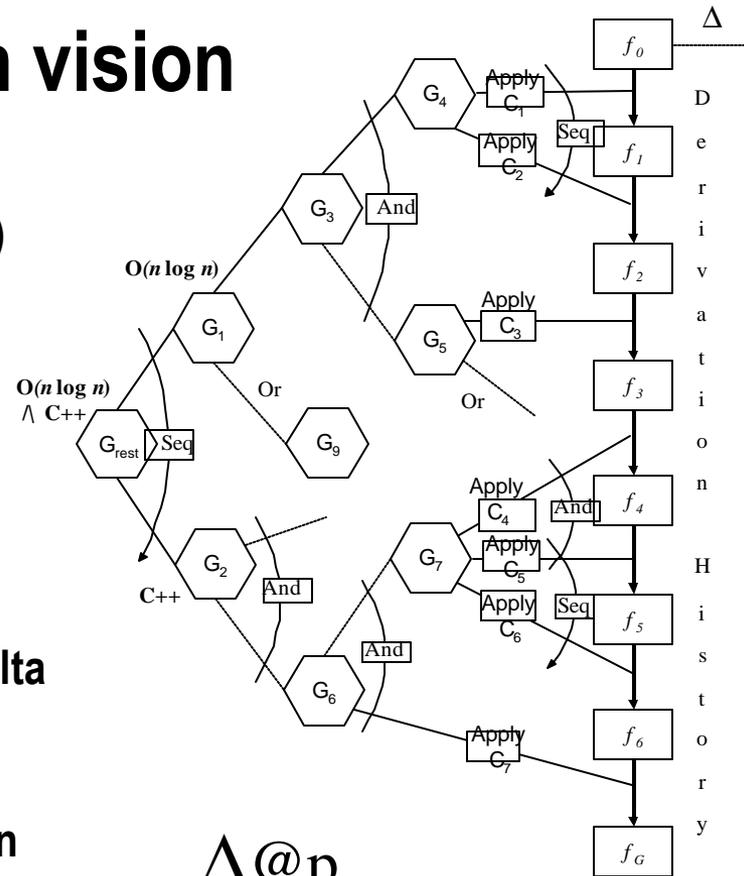
Research Required

(and in varying degrees of progress at SD)

- Concrete answers to "What is a design"?
- Specification Methods: Problem domains, technologies
 - How to specify performance: *how well*
- Combining Specifications of different styles
 - Different domains, different semantics/logics
- Validating Partially implemented Specifications
- Design capture *while* engineering
 - For individuals and concurrent teams
- Means for specifying *changes*
- Automated support for Design revision driven by change
 - Modularity: via interfaces or algebraic properties
- Making this scale to large software systems!

The Design Maintenance System vision

- Transformational Designs
 - Functionality Spec (f_0) + Performance Spec (G_{rest}) + Derivation + Justification + Alternatives
- Scale
 - Metaprogram driven automation
 - Incremental Updates
 - Specification & Technology Δ s
 - Δ s drive design revision: retain transforms that commute with delta
 - Domain-based specification/implementation
 - Simplify expression of problem
 - Store implementation knowledge with domain
 - PARLANSE: Parallel foundation of DMS



$$\Delta @ p (C_i @ q (f_i)) = C_i @ q' (\Delta' @ p' (f_i))$$

