A Vision for Generic Concern-Oriented Requirements Reuse$^{RE@21}$

Gunter Mussbacher ♦ University of Ottawa
gunterm@eecs.uottawa.ca
Jörg Kienzle ♦ McGill University
Joerg.Kienzle@mcgill.ca

RE 2013 ♦ Rio de Janeiro, Brazil **July 17, 2013**

Table of Contents

Vision

some years from now… a software engineer is tasked to build a new application

broadly identifies domain-specific and generic concerns

virtual software engineering bookshelf filled with generic reusable concern units

- complete set of models from requirements to implementation
- interactive guidelines
  - composition instructions
  - variations + their implications on system goals

Concern Characteristics

encapsulate composable reusable models of a generic concern (i.e., not product-specific) with well-defined interfaces for all models (well-packaged)

fundamental principles of SoC, encapsulation, and information hiding

reach across all software development phases

SPL principles

coordinated composition

– use the most appropriate formalism to express model properties and composition rules relevant at current level of abstraction

1st key principle

Concern Characteristics

impact evaluation of variations

- modelled as generally as possible including all relevant variations
- guidance on how to choose among variations
- known impact on high-level system & stakeholder goals

2nd key principle

define model transformations

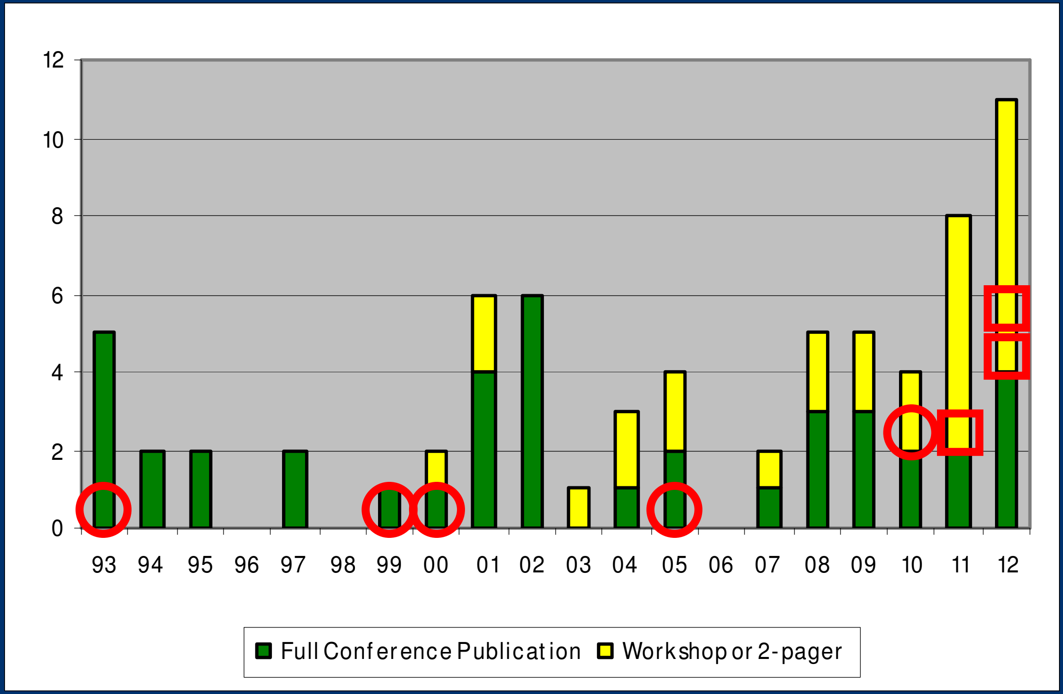
- link models/compositions from one phase to next
- avoid duplication of effort
- preserve properties
- minimize accidental model complexity

MDE principles

Survey of RE Publications

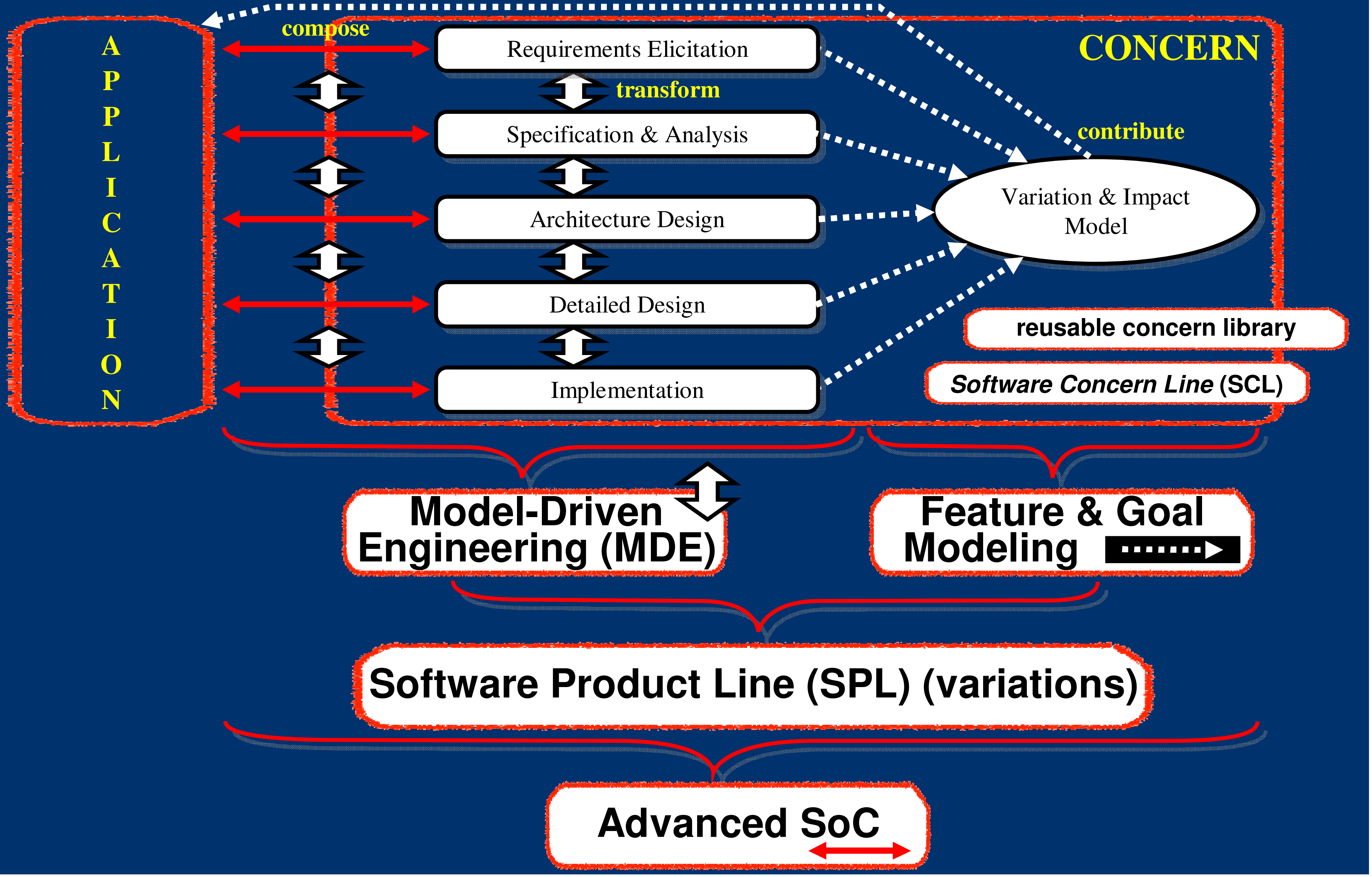
very few publications that are generic (G), model-based (M), and span several development phases (P)
- transformation / refinement (‘93/’99/’05)
- case-based approach (‘10)
- selection-based approach over design space (‘00)

improved requirements reuse



survey of RE Conference publications on requirements reuse
(SPL-related, pattern-based, matching against abstraction, impact, transformations, traceability)
(◯... G + M + P; □... G + M + P + our publication)

Building Blocks



The CORE Approach

one incarnation of
concern-driven development

requirements:
AoURN/SPL (Aspect-oriented User Requirements Notation with extensions for feature modeling)

design:
RAM (Reusable Aspect Models)

implementation:
Java or **AspectJ**

Research Agenda

constructing concern vs. composing concern

coordinated composition is at the heart of this vision

- theory of concern relationships
- partial generation of composition specifications from a preceding phase?
- examination of system properties based on concern compositions?
- concern interaction detection and handling

Research Agenda

analyzable impact model of selected variations is just as crucial

- need a better way to characterize and measure solution artifacts
- difficult to establish well-structured hierarchy of impacts
- patterns may emerge in these impact models that could help describe and reason about impact in more abstract terms

integration of modeling formalisms is also an issue

- domain-specific models of a concern have to be composable with those of other concerns which may use different formalisms

Research Agenda

tool integration is of paramount importance
- model composition, traceability, and impact analysis
- applying MDE principles to ensure that higher-level models are reliably transformed into lower-level models
- since compositions are now a primary application-specific artifact, shift of focus for traceability from structural entities to composition specifications?

more concrete examples and empirical studies are needed
- complete, end-to-end, composable, reusable concerns
- empirically evaluated and quantitatively assessed