The Future of Component-Based Development is Generation, not Retrieval

Hans de Bruin, Hans van Vliet
{hansdb,hans}@cs.vu.nl
Vrije Universiteit, Amsterdam
CBD as a Reuse Technology ...

• … has not lived up to its expectations, yet

• Problems
  – Finding the right components is difficult due to incomplete specifications
  – Architectural mismatch
  – Domain-specific components are typically too rigid to be adapted to (slightly) different settings

• For these reasons, CBD is not longer seen as a reuse technology
CBD for Constructing Flexible Software

• The main driver is change

• Unfortunately, CBD is not the final answer
  – underlying assumption: stable software architecture
  – some changes are not localized, but have a
crosscutting effect on multiple components

• Our conclusion: CBD does not deliver its
promises of reuse and flexibility
• What alternatives can be offered?
Component Generation

• We propose a generation technique to generate systems from functional as well as non-functional requirements

• It is based on two pillars
  – Feature-Solution (FS) graphs
    • Features are connected to solutions that realize these features
  – Top-down composition technique

• Iterative approach
  – Starting point: reference architecture addressing functionality
  – Reference architecture is stepwise refined to cater for non-functional requirements
FS Graph Example
Concluding Remarks

- We are putting our approach in practice: the QuaSAr (Quality-driven Software Architecture) project
- The goal is to generate component-based systems in real application domains
- Components are identified through a domain engineering process
- In our approach, components are gray-boxes in order to refine them recursively
- CBD future: domain engineering and versatile component generation/adaptation techniques