AN APPLICATION SPECIFIC TECHNIQUE FOR RETRIEVAL AND ADAPTATION OF TRUSTED COMPONENTS

Benny Thomas
Master of Computer Science
Supervised by Dr. David Hemer

SOFTWARE DEVELOPMENT COSTS!!

From AGARD Advisory Report 292:-
- Concept Definition 5%
- System Design 12%
- Software Requirements 14%
- Software Design 15%
- Coding 10%
- Software Unit Test 12%
- Integration Test 7%
- System Test 10%
- Documentation 15%
SOFTWARE DEVELOPMENT COSTS!!
Continued…

From NASA Manager’s Handbook for Software Development:-

- Requirement Analysis 6%
- Preliminary Design 8%
- Detailed Design 16%
- Implementation 45%
- System Testing 20%
- Acceptance Testing 5%

COMPONENT BASED SOFTWARE ENGINEERING (CBSE) TO THE RESCUE!

- 40% of costs are directed towards the implementation and testing phases
- CBSE saves time and money by advocating the reuse of software
- BUT effort of using components must be less than effort of developing from scratch
- Technical challenges include
  - Locating components
  - Adapting components to suit the requirements of the specification
HOW DOES CBSE WORK?

- Previously, traditional keyword-based retrieval strategies were used
  - Ambiguous
  - Imprecise
- More recently formal specifications used to define the components and their interfaces
- Retrieval done using theorem provers and unification algorithms
- This itself is not enough. Component adaptation needs to be addressed too.

RECENT EFFORTS

- Parameterized templates used to define adaptation strategies
- Specification matching introduced to semi-automate component retrieval and adaptation
- BUT still early days
  - Need more primitive templates
  - Need more gluing templates
  - Need to test ideas on bigger examples
AIMS OF THE PROJECT

- Develop new library components
  - Primitive components
  - Gluing components
- Apply new templates to case studies

LIBRARY POPULATION

- Generally populating the library requires human presence
- Main focus will be on modifying existing templates to accommodate other types
- REVERSE ENGINEERING
  - Systematic breakdown of programs
  - Primitive and essential components
  - Helps the adaptation process too
CASE STUDIES

- Work on 2 or 3 non-trivial examples
  - Develop from scratch
  - Reverse Engineer implementation
  - Represent as library templates
  - Redevelop using library templates
- Develop application specific search tactics based on existing ones
- Semi-automate process using developed search tactics

QUESTIONS???