Hiding expressed using Relation Algebra with Multi-relations
- oblique lifting and lowering for unbalanced systems -

Reinder J. Bril, Loe M.G. Feijs, André Glas, René L. Krikhaar and Thijs Winter

Overview
• Context
• Relation algebra
• Contributions
• Conclusions

Application domain
• Telecommunications domain
• Philips’ family of PBXs (called Sopho):
  – 100 - 1M telephony lines
  – origin dating back to early 1980’s
  – maintenance obligations >= 10 years
  – successful => asset
    => careful to maintain this legacy

Application domain
• Characteristics (module architecture):
  – 5 K files, 2.5 MLOC in C++;
  – 8 K architectural entities, organised in an unbalanced tree, depth 5 - 12;
  – 35 K include statements;
  – layered system.
Application domain

- Characteristics (execution architecture):
  - 100 - 300 static processes;
  - up to 3000 dynamic processes (telephone calls).

Need for architectural support

- Complexity of legacy systems
- Vital role of software architecture
- Architectural Maintenance:
  - recovery and analysis
  - control and verification
  - (improvements)

Need for architectural support

- Lack of COTS tools;
- Development of URSA:
  - Understanding and Recovery of the Sopho Architecture;
  - program understanding & complexity control;
- Forthcoming paper in JSM.

Module architecture notions

- Module diagram
- Decomposition tree
- Lifting
- Lowering
- Weights
- Hiding

Module diagram

- Visualises a system's architecture
- "Boxes-in-boxes" representation
- Boxes represent entities
- Arrows represent dependencies (uses relation)

System S is not layered

Decomposition tree

- Arrows represent containment (part-of relation)

System S is balanced
Lifting (1)

Uses relation corresponds with a level (tree-cut)

Lifting (1)

Transform a relation to a higher level

Lifting (2)

Transform a relation to a higher level

Lifting (2)

Transform a relation to a higher level
Lifting (2) →

Lifting (2) →

Lifting (2) →

Lowering

Lowering

Both cases a complete graph.

Application: architectural verification, e.g. layering

Weights

Which value to be associated?
Weights

4: number of uses relations

size-oriented weight

Fisheye view of the

size-oriented weight

Weights

3: number of used entities (A₁, A₂, and A₃)

fan-in-oriented weight

2: number of using entities (B₁ and B₂)

fan-out-oriented weight

Each weight has its merits during architectural analysis

Hiding

Hiding the decomposition structure of both A and B
Overview

- Context
- Relation algebra
  - Usage
  - Overview
  - Application
- Contributions
- Conclusions

Usage

- Visualisation and view calculations
  - reverse architecting purposes
- Relational calculus
  - software architecture analysis
- Architectural rules
  - software architecture verification

Application

- Applied for balanced systems
- However:
  - Sopho is an unbalanced system
  - No obvious generalisation of lifting and lowering to unbalanced systems
  - Transformation towards a balanced system yields awkward representations

Decomposition tree

System S is balanced
Decomposition tree

System S is unbalanced

Lifting ...

System S is unbalanced

Lifting ...

No tree-cut

Lifting ...

No tree-cut

Lifting ...

Transformation

Let's make things better

PHILIPS
Transformation

Transformation: balanced tree
But what about a depth 5 - 12?

Overview

• Context
• Relation algebra
• Contributions
  – Summary
  – Lifting revisited
  – Hiding
  – Remarks
• Conclusions

Summary

• Generalisation for unbalanced systems
  – oblique lifting and lowering:
  – for both relations and multi-relations;
  – 4 definitions of weights
• Enhancement based on
  – (reflexive) transitive closure

Lifting revisited

Lifting: \( U = P^\top ; U ; P (= U \uparrow P) \)
where \( P \) is a part-of relation, i.e., functional and a-cyclic

Hiding the structure of A

Degenerated case: tree-cuts

Hiding the structure of A

Degenerated case: tree-cuts
Hiding the structure of $A$

Degenerated case: tree-cuts

Definition: $M = P^* | \text{dom } T | \text{ran } T$

where $P^*$ is the reflexive transitive closure of $P$

Remarks

- Reflexive elements in $M$:
  $M = \{<B_1, B_1>, <B_2, B_2>, <A_1, A>, <A_2, A>, <A_3, A>, <A_4, A>\}$

  - $M$ is functional, but
  - $M$ is not a-cyclic, hence
  - $M$ not a part-of relation
- liberal notion of lifting $\Rightarrow$ oblique lifting
- applicable for weights distinguished
- similar for lowering $\Rightarrow$ oblique lowering

"Collapsing" inter-subsystem uses

Hiding $A$: see paper

No tree-cut
Conclusions

• Oblique lifting and lowering for unbalanced systems
• Oblique lifting as formal basis for MAB;
  – MAB supports all weights
• Main usage:
  – architectural analysis, and
  – architectural verification

Acknowledgements

• Rob C. van Ommering
• André Postma
• Marc A.L. Stroucken
• Gerard J.J.M. van de Ven