Context aware factors in rearchitecting two-level models into multilevel models

Mira Balaban, Igal Khitron, Azzam Maraeel
Ben-Gurion University
Israel
Essential MLM rearchitecting

The type-Instance pattern and its MLM rearchitecture:

Improved **accidental complexity** in the MLM:

- reduce instance **size**
- *Book* is upgraded from a dynamic instance into a **conceptual model element**
Essential MLM rearchitecting with context

- Direct clients
- Indirect associations
- Related classes
- Class hierarchy: Ancestors, descendants
Context “peculiarities”

- Association cycle with ProductType and Product
- ProductType has descendants
- Mixed class hierarchy with ProductType and Product
Context → levels? How?

- Association cycle classification
- Mixed class hierarchy
- Descendants

Difficult:

*What is a (the) best context aware MLM rearchitecture?*
Context aware MLM rearchitecture

- Need to decide how/where to place context elements
- Decisions affect/impact on quality of MLMs

No silver bullet – no best context aware MLM
Accidental complexity is affected by multiple factors

Intention:
- Single out factors that affect quality
  - determine accidental complexity of the MLM
- Suggest quantitative measures for the factors
  - measure accidental complexity of the MLM

Observation:
- Factors can be inter-related
  - Overlapping
  - Conflicting
- Modelers need to state ideals
  - Select factors that maximize the modeler’s ideals
**Factor redundancy**: Measure duplication

- level 2: `OrderItem`, `Bundle` join `ProductType`
- level 1: `ProductType` has 4 member subtypes
- Level 0: Some objects

![Diagram showing relationships between `OrderItem`, `Bundle`, and `ProductType` subtypes]
Factor **redundancy**: Measure **duplication**

- **OrderItem, Bundle** – no subtype classification

  ![Diagram showing redundancy and duplication](image)

  - Singletons
  - **Duplication** = 10

  - **Orders** duplicated by 4 associations
  - **Contains** duplicated by 4 associations
**Factor refinement**: dual to redundancy

Refinement is the "alter ego" of redundancy:

An unnecessary duplication can turn into a desirable refinement.

**Additional constraint**:

A bundle cannot include both a PCStan and PCDel.
Factor **Upward Level-Coupling (ULC)**: A level is coupled with multiple higher levels

- A level might be affected by changes in multiple higher levels
- Reminds the divergent change software smell

Level 0 is affected by levels 1 and 2:
- **Bundle**
- **Or1**
- **Oi1**
- **PCDel**

Suggested ULC measures:

\[
\text{ULC} = 1
\]
**More factors and measures for MLM rearchitecture**

- **Downward Level Coupling (DLC):** A level coupled with multiple lower levels
  - Dual to ULC
  - Reminds the *shotgun surgery* software smell
  - DLC measures:
    - \( A@\text{leap n} > 1 \)

- **Level instability:** A level of types is changed following state changes in a data level
  - Types are relatively stable
  - Objects are frequently changed – snapshots
  - Measure:
    - \( @n > 1 \)
      - \( A@i<n \)
More factors and measures for MLM rearchitecture

- **Level instability**: A level of types is changed following state changes in a data level
  - Types are relatively stable
  - Objects are frequently changed
  - Measure:

    - Add a bundle
More factors and measures for MLM rearchitecture

- **Conceptualization**: make conceptual elements first-class citizens in the model
  - Type-Instance relationship
  - Attribute inheritance control
  - Association inheritance control
  - Dynamic type creation
  - Lose of visibility
    - Addition of operation
    - Duplicate client request

- **Compositionality**: a model is composed from component models
  - Compositional management
  - Reuse
  - Measure: like ULC and DLC
More factors and measures for MLM rearchitecture

- **Direct mapping**: directly reflect intended abstractions
  - Level incohesion: caused by mixture of objects and types

- **Understandability**: a multilevel model is more difficult to understand than a 2-level model
  - Number of levels
  - Number of constraints
    - Implicit
    - explicit
  - Inter-level constraints
    - Explicit
    - Inter-level associations
    - Leap-potency
Factor conflicts

- **MLM** might
  - increase duplication
  - Reduce visibility (conceptualization)

- **Direct mapping** might increase
  - upward level coupling,
  - downward level-coupling,
  - level incohesion

- **Direct mapping and conceptualization** might reduce
  - compositionality and understandability

Modelers need to state **ideals**

- Select factors that maximize the modeler’s ideals
MLM transformation, guided by modeling ideals

The directives (ideals): Direct mapping; Understandability

Backward inter-level association

Accidental complexity = 7
Conclusion and future work

• **MLM needs**
  – In-depth study of management methods
  – Development methodology
  – User study

• **Careful analysis** of quality
  – Modeling elements in an MLM environment
  – E.g., use backward inter-level association, to preserve client visibility

• **Automation techniques**
  – E.g., modeling-ideals-based MLM transformations
Thank you