Negotiated Grammar Transformation
Extreme Metamodel Evolution
Models & Evolution
Consistent Model Evolution – Facts and Myths

Alexander Egyed

Johannes Kepler University (JKU), Linz, Austria
http://www.sea.jku.at
Modeling Languages are Diverse
Modeling Languages are Diverse
Alternative Locations for Change Propagation:
1) rename message playPause
2) Change receiver of message playPause
3) add a new operation to the class Display
4) change the ownership of object display
6) rename operation select
7) rename operation play
8) rename operation draw
9) delete message playPause
Alternative Locations for Change Propagation:
1) rename message \textit{playPause}
2) Change receiver of message \textit{playPause}
3) add a new operation to the class \textit{Display}
4) change the ownership of object \textit{display}
6) rename operation \textit{select}
7) rename operation \textit{playPause}
8) rename operation \textit{draw}
9) delete message \textit{playPause}
Change 1
Inconsistencies: 4 cause
Change 2
Inconsistencies: no change
Change 3
Inconsistencies: 1 cause/1 repair
Change 4
Inconsistencies: 1 repair
Change 5
Inconsistencies: 1 repair

Selecting a Movie

- select()
- draw()
- play()
- stop()

Streamer
- connect()
- wait()
- stream()

Behavior Display
- playing
  - play
  - stop
- select
- wait
- stream
  - stream
  - draw
  - wait

User
- u:User
- d:Display

Display
- select
- draw
- play
- stop

© 2012 Alexander Egyed
Change 6
Inconsistencies: 2 repair
Grammar Transformation
Grammar Transformation
Not that metamodel evolution

What is Metamodel Evolution?

In practice; organic, trial and error, manual

Rarely written as a model transformation
Operator-based Model Migration Approaches:

Operator-based co-evolution

+ migration strategy is “free”
- tool lock in
- operator set completeness
- reverse engineering challenging

Transformation components
Transformation components

- known semantics, well-defined algorithm
- rename, fold, factor, inject, remove, …
Transformation components

Arguments

• what exactly to rename/factor/inject/…?
**Transformation components**

- **Input grammar**
  - determines applicability

![Diagram showing transformation process with nodes G, F, f(a), G', and a, and arrows indicating the flow.](image-url)
Transformation components

\[ \text{G} \xrightarrow{f(a)} \text{G'} \]

F

a
A transformation sequence

expr : ...;
atom : ID | INT | '(' expr ')';

abstractize

expr : ...;
atom : ID | INT | expr;

vertical

expr : ...;
atom : ID;
atom : INT;
atom : expr;

unite

expr : ...;
expr : INT;
expr : expr;

abridge

expr : ...;
expr : ID;
expr : INT;
expr : expr;

Adaptation
Adjustment

Adjustment
Adjustment
Adjustment
Tolerance
Tolerance
Tolerance
Examples
Rename a nonterminal

rename(expr, Expr)

ok
Rename a nonterminal

rename(expr, Expr)

no expr!
Rename a nonterminal

rename(expr,Expr)

no expr!

rename(exp,Exp)

ok
Vertical vs. horizontal

- A is B or C
- \text{vertical}
- A is C
- A is B
Vertical vs. horizontal

- A is B or C
- A is B
- A is C
Vertical vs. horizontal

- A is B or C

ok

vertical(A)
Vertical vs. horizontal

- A is B
- A is C

vertical(A)

A is vertical!
Vertical vs. horizontal

- A is B
- A is C

A is vertical.
Conclusion
To summarise

• Some metamodel transformation paradigms are too rigid.

• Adaptation through:
  • tolerance
  • adjustment

• Isolate applicability assertions

• Negotiate the outcome
  • the oracle
  • the script
  • the user
Credits

• Some slides by Alexander Egyed: Consistent Model Evolution

• Some slides by James Williams: Searching for Model Migration Strategies

• Tatjavanavark-machine.jpg (GFDL, CC-BY-SA)

• Torii kiyoshige bando hikosaburo ii.jpg (PD)

• Software Language Processing Suite (CC-BY-SA)

• See the poster on renarration!

• Ask questions offline and online!
Questions?

vadim@grammarware.net