


JOHANNES KEPLER UNIVERSITY LINZ | JKU



## Detecting and Fixing Inconsistencies during Model-Driven Software Development

**Univ.-Prof. Dr. Alexander Egyed**  
 Johannes Kepler Universität, Linz (JKU)  
<http://www.sea.jku.at>

1

JOHANNES KEPLER UNIVERSITY LINZ | JKU

### Who am I?

Current Affiliations:

- Professor at **Johannes Kepler University**, Austria 2008
- Head of **Institute for Systems Engineering and Automation** (11 Staff Members)

Doctorate Degree:

- University of Southern California**, USA 2000 (under Dr. Barry Boehm)

Past Affiliations:

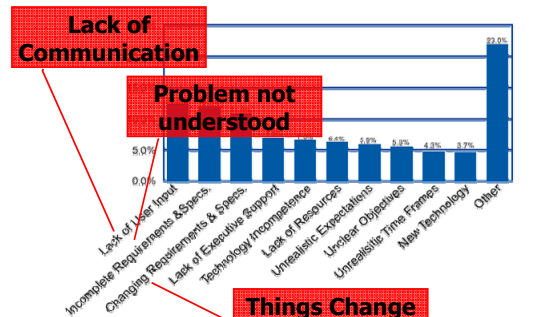
- Research Fellow at **University College London**, UK 2007
- Research Scientist at **Teknowledge Corporation**, USA 2000

SEA © 2009 Alexander Egyed

2

JOHANNES KEPLER UNIVERSITY LINZ | JKU

### Why Software Projects Fail



Reason	Percentage
Lack of Communication	23.5%
Problem not understood	13.8%
Lack of User Input	12.3%
Incomplete Requirements & Specs	11.8%
Changing Requirements & Specs	7.5%
Lack of Executive Support	7.0%
Technology Incompatibility	6.5%
Lack of Resources	5.8%
Unrealistic Expectations	4.8%
Unclear Objectives	4.8%
Unrealistic Time Frames	3.7%
New Technology	3.7%
Other	3.7%

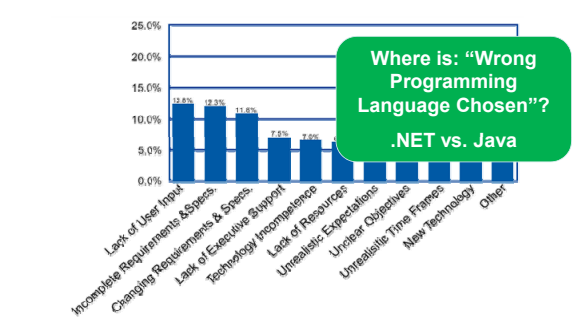
352 companies - 8,000 software projects. Source: The Standish Group, 1995

SEA © 2009 Alexander Egyed

3

JOHANNES KEPLER UNIVERSITY LINZ | JKU

### Why Software Projects Fail



Reason	Percentage
Lack of User Input	13.8%
Incomplete Requirements & Specs	12.3%
Changing Requirements & Specs	11.8%
Lack of Executive Support	7.5%
Technology Incompatibility	7.0%
Lack of Resources	6.5%
Unrealistic Expectations	5.8%
Unclear Objectives	4.8%
Unrealistic Time Frames	4.8%
New Technology	3.7%
Other	3.7%

352 companies - 8,000 software projects. Source: The Standish Group, 1995

SEA © 2009 Alexander Egyed

4

JOHANNES KEPLER UNIVERSITY LINZ | JKU

### Software Defects "enjoy" high Visibility

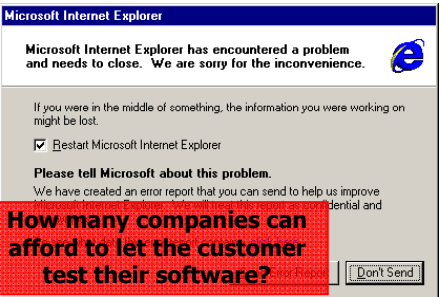
- London Ambulance System (LAS)-1992 [Finkelstein96]
- Mars Climate Orbiter-1998 [Breitman99]
- Therac 25, Medical Linear Accelerator-1985 [Leveson93]
- Siemens: Possible Hearing Damage in Some Cell Phones-2004 [Siemens05]
- Mercedes A-Class capsizing in corners [Mercedes97]
- Ariane 5: code reuse and the dimensions of variables
- Denver Airport Baggage Handling: Reset lost data

SEA © 2009 Alexander Egyed

5

JOHANNES KEPLER UNIVERSITY LINZ | JKU

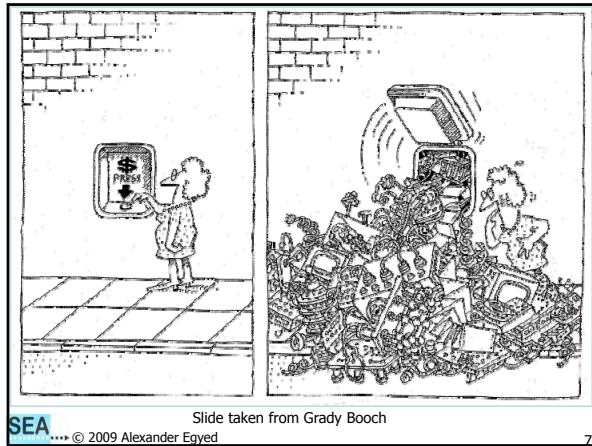
### Embarrassing Software Defects



**How many companies can afford to let the customer test their software?**

SEA © 2009 Alexander Egyed

6

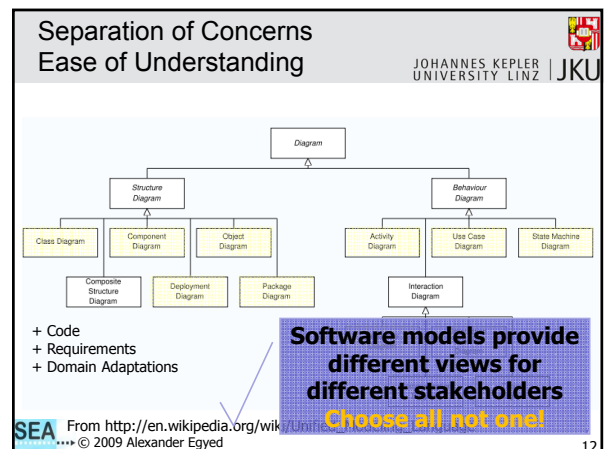
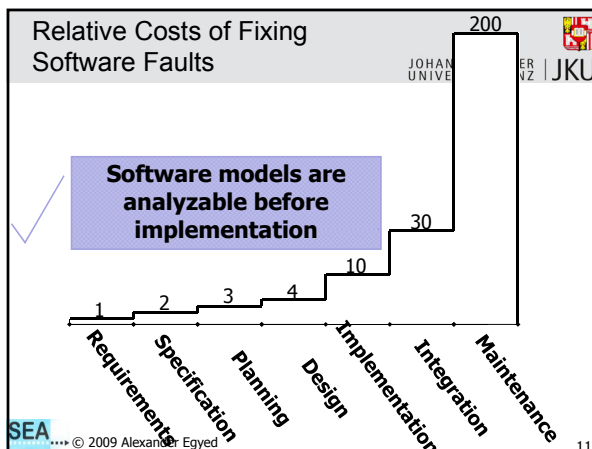
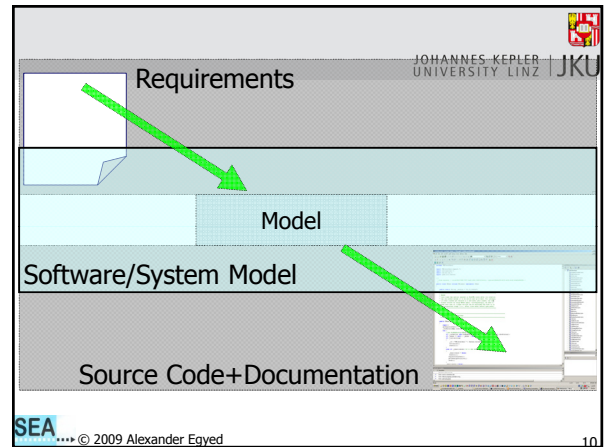
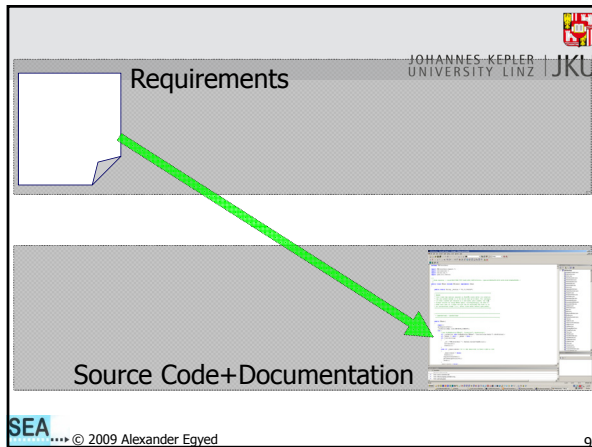


### Models take on Difficulties

JOHANNES KEPLER UNIVERSITY LINZ | JKU

- Complexity ✓ **Software models abstract and simplify**
- Conformity ✓ **Software models show context**
- Changeability ✓ **Software models separate concerns**
- Invisibility ✓ **Software models depict what is otherwise hidden**

SEA © 2009 Alexander Egyed 8



## The Benefits of Modeling

JOHANNES KEPLER UNIVERSITY LINZ | JKU

- Overcome the requirements/code gap
- Reduce the Cost of Fixing Errors
- Exploring the Problem
- Separation of Concerns
- Ease of Understanding
- Maintenance
- To "hide" implementation details
- To "emphasize" limitations and assumptions
- To think the problem through
- Assess Requirement Changes
- Correct Implementation guarantees
- Analyzable
- Integrate disciplines

SEA © 2009 Alexander Egyed 13

## BUT!

JOHANNES KEPLER UNIVERSITY LINZ | JKU

**Modeling tools are like a hammer**

- Can do good but, if used wrong, they are useless!
- Require training, guidance, and automation

Have witnessed many companies "introducing" modeling only to fail because they do not understand **model-driven software development**

SEA © 2009 Alexander Egyed 14

Requirements

**If software models no longer reflect the code or requirements...**

Model

Software/System Model

Source Code+Documentation

SEA © 2009 Alexander Egyed

## Model-Driven Software Development

JOHANNES KEPLER UNIVERSITY LINZ | JKU

Change in paradigm

- => Modeling is the "creative" part and answers questions
- => Coding is filling in details not in models
- => multi-view "programming"

Living Models

- Not documentation but path to solution
- Used/Useful throughout the software life cycle

**analysis**

**generation**

**consistency**

**why** **how** **where** **...**

SEA © 2009 Alexander Egyed 16

## Change, Inconsistencies, and their Impact

JOHANNES KEPLER UNIVERSITY LINZ | JKU

Source Code

Diagram

Structure Diagram

Behaviour Diagram

Class Diagram

Component Diagram

Object Diagram

Activity Diagram

Use Case Diagram

State Machine Diagram

Composite Structure Diagram

Deployment Diagram

Package Diagram

Interaction Diagram

Sequence Diagram

Interaction Overview Diagram

**Changes cause Inconsistencies and we can help the developer detect and fix them!**

SEA © 2009 Alexander Egyed 17

## What Engineers Need

JOHANNES KEPLER UNIVERSITY LINZ | JKU

- Identify Inconsistencies Quickly [ICSE 2006]
- Fixing Inconsistencies Quickly [ASE 2008][ICSE 2007]

**Demo of Model Analyzer developed at the Johannes Kepler University Linz**

SEA © 2009 Alexander Egyed Anthony Finkelstein 18

## Model Analyzer Approach

JOHANNES KEPLER UNIVERSITY LINZ | JKU

**Demo**

**Model and Inconsistencies**

SEA © 2009 Alexander Egyed 19

## Some Consistency Rules

JOHANNES KEPLER UNIVERSITY LINZ | JKU

Rule 1: Name of message must match an operation in receiver's class  
`operations=message.receiver.base.operations`  
`return (operations->name->contains(message.name))`

Rule 2: Sequence of object messages must correspond to events

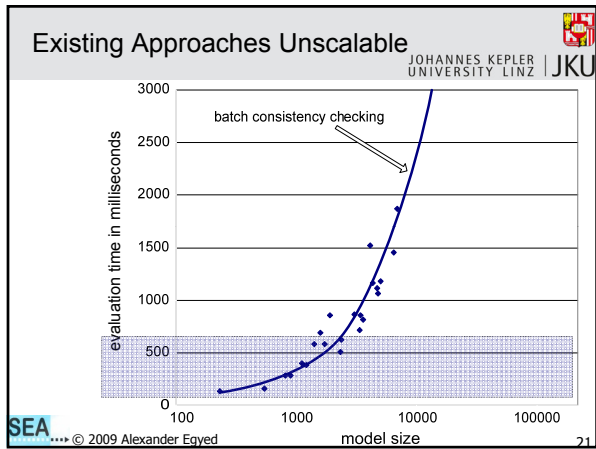
Rule 3: Calling direction of association must match calling direction of messages

...

Rule 100+

**Many rules – impossible to maintain consistency manually**

SEA © 2009 Alexander Egyed 20



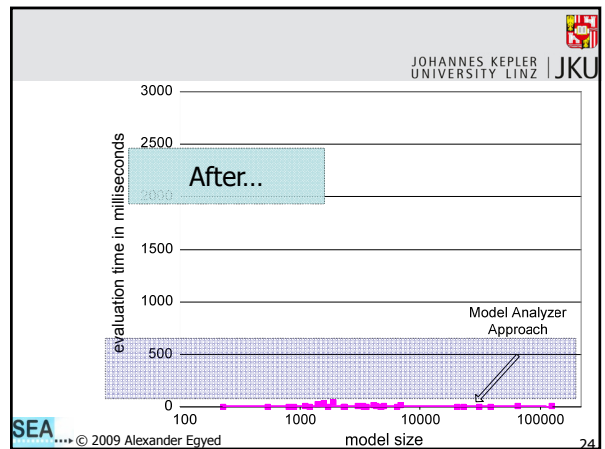
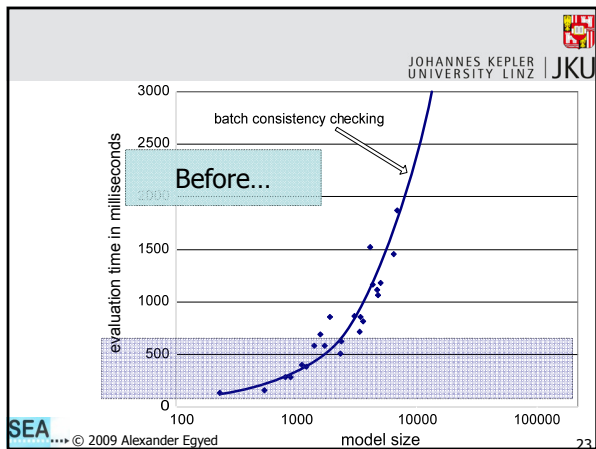
## Model Analyzer Approach

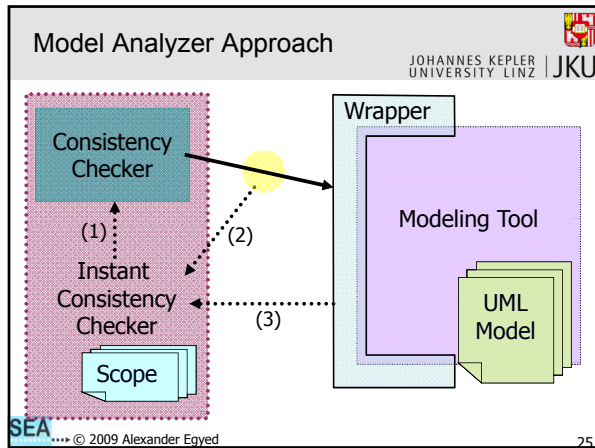
JOHANNES KEPLER UNIVERSITY LINZ | JKU

**Demo**

**Changes and Consistency Checking**

SEA © 2009 Alexander Egyed 22





- ### No Special Language Required
- JOHANNES KEPLER UNIVERSITY LINZ | JKU
- 1) Must use an obscure (and limited) rule language (who understands that language?)
  - 2) Must annotate constraints (who has the time and experience to do that?)
- SEA © 2009 Alexander Egyed 26

### Model Analyzer Approach

JOHANNES KEPLER UNIVERSITY LINZ | JKU

Modeling - Model:Main - Rational® Software Modeler™

File Edit Diagram Navigate Search Project Model Analyzer Run Modeling Window Help

Project Exp 12 Main Selecting a Movie

Miscellaneous Models  
Various Models  
VOD (VOD)  
Diagrams  
Models  
Model \* 51 12/

State/Activity Model  
connect stopped stream playing

Display  
select ()  
stop ()  
play ()  
draw ()

Streamer  
connect ()  
wait ()  
stream ()

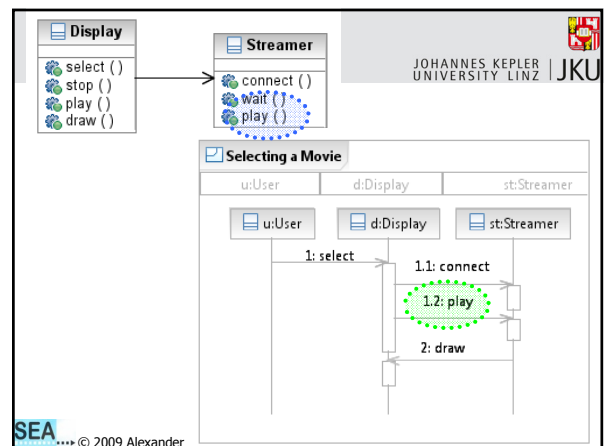
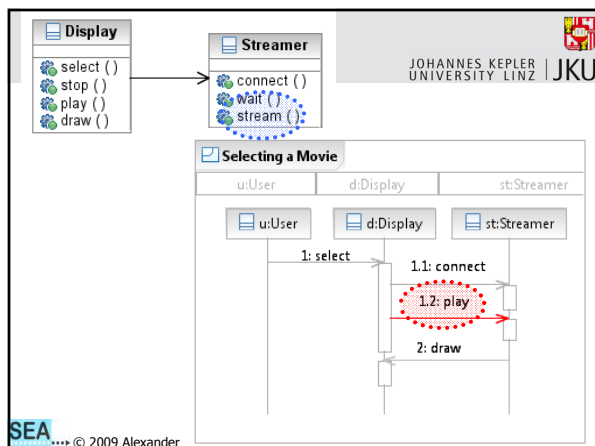
Selecting a Movie  
u:User d:Display st:Streamer  
u:User d:Display st:Streamer  
1: select 1.1: connect 1.2: play 2: draw

**Demo**

New Consistency Rules easily definable

SEA © 2009 Alexander Egyed 27

- ### What Engineers Need
- JOHANNES KEPLER UNIVERSITY LINZ | JKU
- (1) Identify Inconsistencies Quickly [ICSE 2006]
  - (2) Fixing Inconsistencies Quickly [ASE 2008][ICSE 2007]
- Or, how can you help an engineer resolve inconsistencies if arbitrary constraints can be defined
- SEA © 2009 Alexander Egyed Anthony Finkelstein 28



# Model Analyzer Approach

JOHANNES KEPLER UNIVERSITY LINZ | JKU

## Demo

### Finding Locations where to Fix Inconsistencies

SEA © 2009 Alexander Egyed 31

**Locations for fixing this inconsistency:**

- 1) rename message *play*
- 2) Change receiver of message *play*
- 3) add a new method to the class *Streamer*
- 4) change the ownership of object *st*
- 6) rename method *connect*
- 7) rename method *stream*
- 8) rename method *wait*
- 9) delete message *play* (makes rule obsolete)

SEA © 2009 Alexander 32

**Choices for fixing this inconsistency:**

- 1) rename message *play* to *stream*
- 2) Change receiver of message *play* to object *d*
- 3) add a new method *play* to the class *Streamer*
- 4) change the ownership of object *st* to *Display*
- 6) rename method *connect* to *play*
- 7) rename method *stream* to *play*
- 8) rename method *wait* to *play*
- 9) delete message *play* (makes rule obsolete)

SEA © 2009 Alexander 33

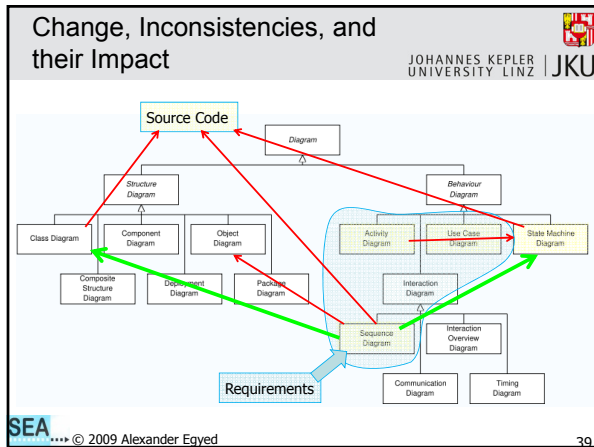
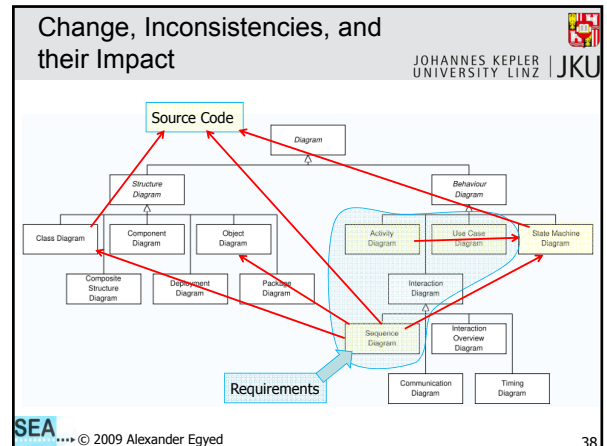
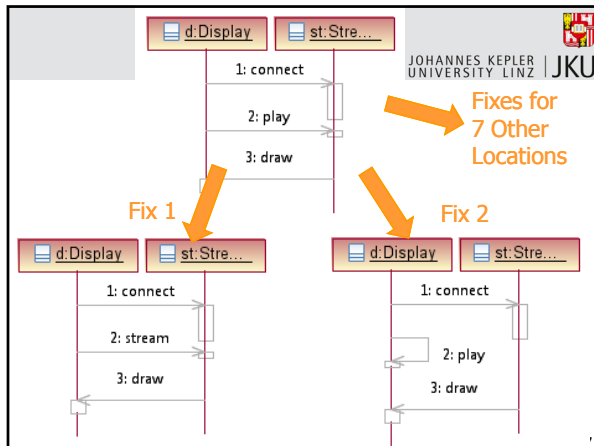
## Demo

**Problem: Wrong Choices! Why is 'stream' the only valid choice?**

SEA © 2009 Alexander Egyed 34

SEA © 2009 Alexander 35

SEA © 2009 Alexander 36



JOHANNES KEPLER UNIVERSITY LINZ | JKU

**Contact Me to Download and Test the Model Analyzer**

**Interested in other Technologies?**

- Traceability (Requirements/Code)
- Optimizing User Input
- Product Line Modeling
- Model Analyses and Generation

**Univ.-Prof. Dr. Alexander Egyed**  
alexander.egyed@jku.at

SEA © 2009 Alexander Egyed 40