Introduction to Model-Driven System Development

Ákos Horváth

Dept. of Measurement and Information Systems
Song writing methods of Simon and Garfunkel
Paul Simon’s technique #1

1. Create music first

Bridge Over Troubled Water

When you're weary
Feeling small
When tears are in your eyes
I will dry them all

I'm on your side
When times get rough
And friends just can't be found
Like a bridge over troubled water
I will lay me down
Like a bridge over troubled water
I will lay me down

2. Write lyrics accordingly
Paul Simon’s technique #2

The Boxer
I am just a poor boy
Though my story’s seldom told
I have squandered my resistance
For a pocket full of mumbles such as promises
All lies and jests
Still a man hears what he wants to hear
And disregards the rest
When I left my home and my family
I was no more than a boy
In the company of strangers
In the quiet of the railway station running scared
Laying low, seeking out the poorer quarters
Where the ragged people go
Looking for the places only they would know

1. Write lyrics first

2. Compose music accordingly
A Combined Technique...

**Scarborough Fair (Folk Song)**
Tell her to find me an acre of land,
Parsley, sage, rosemary and thyme;
Between the salt water and the sea strand,
Then she'll be a true love of mine.

**The Side of a Hill (P. Simon)**
On the side of a hill, a little cloud weeps
And waters the grave with its silent tears
While a soldier cleans and polishes a gun

**Canticle (rearranged by A. Garfunkel)**
On the side of a hill a sprinkling of leaves
Washes the grave with silvery tears
A soldier cleans and polishes a gun
Music Driven Song Development (MDSD)

1. Create music first

Bridge Over Troubled Water
Words & Music by Paul Simon

Moderately, like a spiritual

Bridge Over Troubled Water

When you’re weary
Feeling small
When tears are in your eyes
I will dry them all

I’m on your side
When times get rough
And friends just can’t be found
Like a bridge over troubled water
I will lay me down
Like a bridge over troubled water
I will lay me down

2. Write lyrics accordingly

The Boxer
Words & Music by Paul Simon

Modern song

The Boxer

I am just a poor boy
Though my story’s seldom told
I have squandered my resistance
For a pocket full of mumbles such are promises
All lies and jests
Still a man hears what he wants to hear
And disregards the rest
When I left my home and my family
I was no more than a boy
In the company of strangers
In the quiet of the railway station running scared
Laying low, seeking out the poorer quarters
Where the ragged people go
Looking for the places only they would know

Lyrics Driven Song Development

1. Write lyrics first

Bridge Over Troubled Water

When you’re weary
Feeling small
When tears are in your eyes
I will dry them all

I’m on your side
When times get rough
And friends just can’t be found
Like a bridge over troubled water
I will lay me down
Like a bridge over troubled water
I will lay me down

2. Compose music accordingly

The Boxer

I am just a poor boy
Though my story’s seldom told
I have squandered my resistance
For a pocket full of mumbles such are promises
All lies and jests
Still a man hears what he wants to hear
And disregards the rest
When I left my home and my family
I was no more than a boy
In the company of strangers
In the quiet of the railway station running scared
Laying low, seeking out the poorer quarters
Where the ragged people go
Looking for the places only they would know
Applying the Principle to Software Systems

MDSD = Model Driven Software Development / Engineering

Model

Bridge Over Troubled Water

Consumer 1

Time Out [1 week]

Collect Votes

Join Time?

Post Result on Web-Site

E-mail Results of vote

Did Enough Members and Accurate Vote

Deadline announcement

Assess Number of Votes

End if Enough Members have voted

Post Results

Model

Music

Bridge Over Troubled Water

When you're weary

Feeling small

When tears are in your eyes

I will dry them all

I'm on your side

When times get rough

And friends just can't be found

Like a bridge over troubled water

I will lay me down

Lyrics

Code

Bridge Over Troubled Water

public class AboutDialog extends JDialog

protected CardLayout mLayout;

protected JButton mCredits;

protected JPanel mMainPanel;

public AboutDialog(JFrame owner) {

super(owner);

setModal(true);

setUndecorated(true);

}
Model Driven System Engineering
Terminology

- **MDSE** = Model Driven System Engineering
- **MDSD** = Model Driven System Development
- **MDD** = Model Driven Development
- **MDE** = Model Driven Engineering
- **MBSE** = Model Based Systems (Software) Engineering

MDSE \(\approx\) MDE \(>\) MDSD \(\approx\) MDD \(>\) MBSE

- **MDA** = Model Driven Architecture
  - Design methodology proposed by OMG (Object Management Group)
  - A specific realization of Model Driven Software Engineering

- Related concepts
  - **MDT** (Model Driven Testing) \(\approx\) **MBT** (Model Based Testing)
Model Driven Architecture – Example

Platform Independent Model

Platform Specific Model

Code

Application

```
package hu.bme.mit.entity;

import java.beans.EJBLocalObject;

public interface BookLocal extends javax.ejb.EJBLocalObject {

    public java.lang.String getTitle();

    public void setTitle(java.lang.String newTitle);

    public String getAuthor();

    public void setAuthor(String author);      
```

Please enter author information

First Name: 

Last Name: 

Submit  |  Cancel
Why to model?

```java
public class SomeThing {

    int s = 0;

    public void process(E e) {
        if (s==0) {
            if (e == E.N) s = 1;
        } else if (s==1) {
            if (e == E.S) s = 2;
            else if (e == E.M) s = 3;
        } else if (s==2) {
            if (e == E.I) s = 0;
            else if (e == E.F) s = 4;
        } else if (s==3) {
            if (e == E.I) s = 0;
        }
    }
}
```
Why to model?

Better documentation
Better understanding
public class SomeThing2 {

private static final int INITIALIZATION = 0;
private static final int NORMAL = 1;
private static final int SUSPICIOUS = 2;
private static final int MAINTENANCE = 3;
private static final int FAULTY = 4;

int state = 0;

public void process(E event) {
    if (state == INITIALIZATION) {
        if (event == E.N) state = NORMAL;
    } else if (state == NORMAL) {
        if (event == E.S) state = SUSPICIOUS;
        else if (event == E.M) state = MAINTENANCE;
    } else if (state == SUSPICIOUS) {
        if (event == E.I) state = INITIALIZATION;
        else if (event == E.F) state = FAULTY;
    } else if (state == MAINTENANCE) {
        if (event == E.I) state = INITIALIZATION;
    }
}
}
Why to model?

Can I return to *Normal operation* from *Faulty* state?

Better analyzability
Why to model?

- Generate tests
  - state coverage
    - test1: N, S, F
    - test2: N, M
  - transition coverage
    - N,S,I,N,M,I,N,S,F
  - path coverage
    - from all states to all other states
    - e.g. from *Initialization* to *Faulty*:
      - NSF,
      - NSI NSF,
      - NMI NSF,
      - NSI NMI NSF,
      - **NMI NSI** NSF
What is a model?

- **Mapping** → the model is based on the original system
- **Reduction** → it reflects only an important/relevant segment
- **Replaceable** → can be used in place of the original system for a limited feature
What is a model?

- Structured representation of the information

A railway station is composed of signals, tracks, and points.

Tracks can be connected to each other.

Signals are located at both ends of the track.
What is a model?

Terminology / Metamodel / Modeling Language

Graphical syntax

Instance model – diagram

Instance model – abstract syntax

Points p1

Signal s1

Track t1

Signal s2

Track t2

Point p2

Signal s3

Signal s4
Model vs. Diagram

- **Diagram**
  - A view of the model
  - Important aspects from a given viewpoint are shown

- **Model**
  - All the elements and their relations
How to process a model?

- Define interesting model parts (patterns)

- Find in the model (pattern matching)
  - Result: match sets
    - s11,s12,t1
    - s21,s22,t2

- Apply some operation
  - modify model
  - create other model
  - generate artefact

```java
pattern p1 (s1,s2,t) {
    Signal(s1);
    Signal(s2);
    Track(t);
    locatedAt(s1,t);
    locatedAt(s1,t);
    protectedBy(t,s1);
    protectedBy(t,s2);
}
```
What is needed for MDSE?
The Three Pillars

- **Modeling Language**
  - Defines elements and their relationship
  - Defines syntax and semantics
  - *What type of elements can be used during modeling?*
  - E.g. SysML

- **Development Methodology**
  - Defines the steps of analyzing and designing the system
  - Defines the usage of the model elements and diagrams
  - *How shall the model be built?*
  - E.g. SYSMOD (SYstem MODeling) by Tim Weilkiens, OOSEM, Rational Harmony

- **Proper! Tool**
  - E.g. MagicDraw, Enterprise Architect, IBM Rational Rhapsody
What is needed for MDSE?

+ Two extra things

- **Domain knowledge**
  - Should know what to model

- **Distinguished team**
  - Should have people who have the ability and experience to create good models
What types of models can be used?

- **Static models**
  - Defines the static aspects of the system including data, design and architecture.
  - e.g., E-R model

- **Dynamic models**
  - Defines or describes the dynamic behavior of the system.
    - Usually demonstrates execution.
  - e.g., State machine
What types of models can be used?

Typically we use these levels

Levels of verification algorithms

Low level mathematical formalisms: Automatons, SAT, ...

Higher level formalisms: Dataflow, Petri-net, ...

Engineering models

Model Transformations

UML, SysML, Domain Specific Models
What types of modeling languages can be used?

- **General Purpose Modeling** languages
  - Languages that can be applied to any domain for modeling purpose
  - e.g., State machines, Petri-nets, SAL, UML, SysML

- **Domains Specific Modeling** languages
  - Languages that are specifically designed for a certain domain
  - e.g., AutoSAR, Mathematica, Logo, AADL, etc
What can be done with MDSE?

- **DOCUMENTATION**
  - Support requirements specification
    - Textual based \(\rightarrow\) extended with models (diagrams)
  - Support system design
    - Mind based \(\rightarrow\) documented

- **VERIFICATION**
  - Check consistency, completeness, well-formedness

- **ANALYSIS**
  - Analyze / simulate parts of the system to determine or derive properties
    - Applied separately for a selected component
      - E.g. for a communication protocol: consensus is always reached
    - Integrated into the model based development process

- **SYNTHESIS**
  - Synthetize some parts of the system
    - design, application or other artefacts
Analysis in Model Driven System Engineering

System Design

- Engineering model (e.g. SysML)

Formal verification

- Automated Model Generation
  - Formal model
    - Backtracing results
      - Analysis
### Synthesis in Model Driven System Engineering

**MDSE framework**

- **Platform independent model (PIM)**
- **Platform model**
- **Platform specific model (PSM)**

**Possibilities**
- Platform independent $\rightarrow$ platform specific
- Configuration generation
- Code generation
- Monitor generation
- Test Case generation
- Documentation generation

**Tool Support**
- Workflow Based Execution

---

- **Configuration**
- **Monitor program**
- **Program code**
- **Test cases**

**Application / HW**

- Configures
- Monitors

**Tests**
Model-Driven Engineering of Critical Systems

Traditional V-Model

Model-Driven Engineering

Main ideas of MDE
- early validation of system models
- automatic source code generation
  ➔ quality++ tools ++ development cost

- DO-178B/C: Software Considerations in Airborne Systems and Equipment Certification (RTCA, EUROCAE)
- Steven P. Miller: Certification Issues in Model Based Development (Rockwell Collins)
How to apply Model Driven Engineering?

act Possibilities for applying Model Driven Development

- Create Model Based Requirements Specification
- Verify Requirements Specification
- Create Model Based System Design
- Extend Model Based System Design with Variation Modeling
- Simulate System Level Behavior
- Determine System Properties, Perform Variation Analysis
- Analyze System
- Model Configuration
- Generate Configuration
- Simulate Software Behavior
- Analyze Software
- Generate Software
- Create Test Model
- Generate Test Cases
- Model Software Components
Summary

- **What is a model?**
  - Abstraction of the real world
  - Built from elements defined by the modeling language (metamodel)

- **Why to model?**
  - Document, verify, analyze, synthesize

- **What are needed for modeling?**
  - *Language + methodology + tool + domain knowledge + expertise*

- **What types of models to use?**
  - General purpose vs. domain specific
  - Engineering vs. mathematical (~ semi-formal vs. formal)
  - For modeling behavior and structure

- **How to process models?**
  - By means of model queries and transformations