Research Overview

- **Formal Methods**
  - Formal verification of programs
  - Investigation of a specification language
  - Formalizations in MultiMedia

- **Consistency in UML**
  - Series of workshops on MoDELS
  - Classification framework
    - Consistency Issues in Modelling

- **Didactics**
  - Educators Symposium on MoDELS
    - *Best Practices for Teaching UML Based Software: MoDELS 2004*

- **Software Development and Empirical Methods**
  - Validation of stereotypes with experiments
    - Empirical assessment of using stereotypes to improve comprehension of UML models: A set of experiments
  - Validation using survey
    - Empirical extension of a classification framework for addressing consistency in model based development

Research Overview

- **Formal Methods**
  - Formal verification of programs
  - Hiding deadlocks in Lotos
  - Formalizations in MultiMedia

- **Consistency in UML**
  - Series of workshops on MoDELS
  - Classification framework
    - Consistency Issues in Modelling

- **Didactics**
  - Educators Symposium on MoDELS
    - *Best Practices for Teaching UML Based Software: MoDELS 2004*

- **Software Development and Empirical Methods**
  - Validation of stereotypes with experiments
    - Empirical assessment of using stereotypes to improve comprehension of UML models: A set of experiments
  - Validation with survey
    - Empirical extension of a classification framework for addressing consistency in model based development
Teaching Overview

- WUT
  - OO Software Development,
  - Formal methods in SE
- BIT
  - OO Software Development – 1st y BSc level
  - Product Line Architecture – 2nd y MSc level
  - Master Thesis course – 2nd y MSc level
- Double Diploma BIT - WUT
  - Research Methodology
- NUR – National University of Rwanda
  - OO Software Development
  - Product Line Architecture
  - Master Thesis
- NU – Newcastle University
  - Introduction to UML

OMG’s MDA

Object Management Group (OMG)
  an open membership, not-for-profit consortium
  that produces and maintains computer industry specifications for interoperable enterprise applications.

- Bad news
  There will never be a single OS, Pl, Network Architecture that replaces all that have passed
- Good news
  You can still manage to build systems economically in this environment
- Remedy
  A specific approach to software development – Model Driven Architecture (MDA)
OMG’s MDA

- Provides open, vendor neutral approach to the challenge of technology change.
- Is a broad conceptual framework that describes an overall approach to software development.
- Is the OMG implementation of MDD approach to software development by standards and a set of tools that can be used within MDD.
- Based on OMG standards separates application logic from underlying platform technology.
- Does not define a particular software architecture or an architectural style.
- Raises the level of abstraction in software development.

MDA Core Standards - Technology Space

- MOF - Meta-Object Facility
  - an abstract language and a framework for specifying, constructing, and managing technology neutral metamodels => languages
- UML - Unified Modeling Language
  - a graphical language for specifying, constructing, visualizing & documenting the artifacts of distributed object oriented systems
- XMI - XML Metadata Interchange
  - technology mappings from MOF metamodels conforming to XML DTDs and XML documents.
- Others still coming ....

MDA: Development Process Perspective

- MDA (Model-Driven Architecture)
  is a type of
  MDD (Model-Driven Development)
- Not a Software Architecture design
  - “Architecture” refers to a framework of concepts, tools etc.
- But a Development Paradigm
  Approach to developing software
MDA: Development Process Perspective

- **MDA = approach + tools** for:
  - **Specifying**
    - platform-independent system – application platforms
  - **Choosing**
    - a platform for the system
  - **Transforming**
    - the platform-independent specification into a platform-specific one

- **Concepts**
  - Application: the functionality being developed
  - Platform: technology that provides functionality through interfaces and usage patterns (generic, standard, manufacturer)
  - System: application(s) supported by platform(s)

### Categories of Models in MDA

- **Computation Independent Model - CIM**
  - Does not show details of system structure
  - Independent of how the system will be implemented
  - “domain model” or “business model”
  - Bridges the gap between domain experts and design/development experts

- **Platform Independent Model - PIM**
  - Structure, functionality and behavior of the software system built using OMG standards

- **Platform Specific Model - PSM**
  - Realization of the PIM on a given platform/technology

Models Hierarchy: Four Layers Architecture

- **Language for defining languages** - MOF
- **Modelling Language** - UML

MDA Tool:

- **Modelling components**
  - To build and maintain PIMs
- **Code generation component**
  - To perform a series of transformations that map PIM elements to elements in PSMs

MDA Benefits:

- Instead of writing PS-code, focus on developing models that are specific to application domain but independent of the platform
- A tool that implements MDA concept allows to:
  1. produce models of the application and business logic, and
  2. generate code for a target platform by means of transformations
MDA Model Transformation in General

MDA-based SD: Research Areas

T1: Model generation
- constructing UML models based on non-UML artefacts
T2: Model transformation
- constructing UML models based on UML artefacts
T3: Code generation
- construction of the software based on UML artefacts

Empirical Methods in Engineering

- **Confirmation** of (more or less accepted) hypotheses.
  For example: object-orientation is good for reuse.
- **Evaluation** of methods, models, languages and tools.
  For example: whether Java produces higher quality code than C++
- **Identification** of relationships.
  For example: find relationship between fault-prone components and design concepts.
- **Validation** of models or measures.
  For example: to validate a specific cost estimation model.
- **Understanding** of methods, techniques and models.
  For example: to understand relationship between inspections and test
- **Guidance** to help in management.
  For example: for migration from one technology to another.
Experiment

Basic characteristic
- carefully planned and fully controlled,
- should be replicable.

Experiment Model

Experiment Process

Experiment Design - Instrumentation

<table>
<thead>
<tr>
<th>Experiment schema</th>
<th>Round 1</th>
<th>Round 2</th>
</tr>
</thead>
<tbody>
<tr>
<td>Group 1</td>
<td>Set A – S</td>
<td>Set B – N</td>
</tr>
<tr>
<td>Group 2</td>
<td>Set A – N</td>
<td>Set B – S</td>
</tr>
</tbody>
</table>

- Four set of artefacts
  - Set A-S: stereotyped model A and description of stereotypes used
  - Set B-N: non-stereotyped model B,
  - Set A-N: non-stereotyped model A,
  - Set B-S: stereotyped model B and description of stereotypes.

- Artifact set A-x describes a domain of radio transmissions.
  - A class diagram describing different types of existing objects
    (radio station, retransmission station, different types of antennas, etc)
  - A corresponding object diagram describing one of possible situations (like sending a news program across a country).

- Artifact set B-x describes a domain of GSM telephony.
  - A class diagram describing different types of existing objects
    (mobile phone, BTS station, connection to conventional telephone network, etc)
  - A corresponding object diagram describing one of possible situations of using the network (like making phone calls in a given time).

Sample Experiment Design

- Idea Behind Empirical Research
  *Investigate how stereotypes influence understanding of UML encoded artefacts*

- Method
  - Experiment
  - Design
    - Pair comparison on design artefacts
    - Subjects: SE students of different background
    - Input / Objects: (4 sets of) design artefacts (with 6 stereotypes)
    - Instruments (initial): questionnaires (on 4 types) of diagrams
      - Component Model & Collaboration Diagram
      - non stereotypes and stereotyped
  - Output:
    - time
    - level of understanding

Sample artefacts

<table>
<thead>
<tr>
<th>Name</th>
<th>Icon</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>&lt;&lt;sender&gt;&gt;</td>
<td><img src="sender.png" alt="Icon" /></td>
<td>makes the class capable of sending messages to classes stereotyped &lt;&lt;receiver&gt;&gt; or &lt;&lt;transmitter&gt;&gt;, allows the class to send messages without receiving them first, prevents the class from receiving any message,</td>
</tr>
<tr>
<td>&lt;&lt;receiver&gt;&gt;</td>
<td><img src="receiver.png" alt="Icon" /></td>
<td>makes the class capable of receiving messages from classes stereotyped &lt;&lt;sender&gt;&gt; or &lt;&lt;transmitter&gt;&gt;, enables the class to receive a message without sending it further, prevents the class from sending any messages to other classes,</td>
</tr>
<tr>
<td>&lt;&lt;transmitter&gt;&gt;</td>
<td><img src="transmitter.png" alt="Icon" /></td>
<td>makes the class capable of receiving messages from classes stereotyped &lt;&lt;sender&gt;&gt; or &lt;&lt;transmitter&gt;&gt;, and sending the received messages to classes stereotyped &lt;&lt;receiver&gt;&gt; or &lt;&lt;transmitter&gt;&gt;, any message that is sent from such a class must be preceded by receiving of the message by this class</td>
</tr>
</tbody>
</table>
• NRESP – number of correct responses

• Time spent

**Best Practices**

• Tailoring of Development Process
  - Defined Artefacts and Creation Procedures
  - Effective Usage of Models and Modelling – MDA conformance

• Consistency Awareness and Management

• Research elements
  - Participation - Conducting Experiments During the Course
    - Passive and Active
  - Transfer of Research Results
    - proper and effective usage of advanced UML elements – for instance where and how introduce stereotypes, how they can help, what benefits can be obtained
    - usage of capabilities included in modelling tools

• Industrial and Professional Relevance

• **Constant Feedback from Participants**

**Follow Up**

• Replication
• Categorization
• Teaching

**Overall improvement**

- NRESP improves; TSEC the same
- TSEC improves; NRESP the same
- NRESP improves; TSEC deteriorates
- TSEC improves; NRESP deteriorates
- Both deteriorate