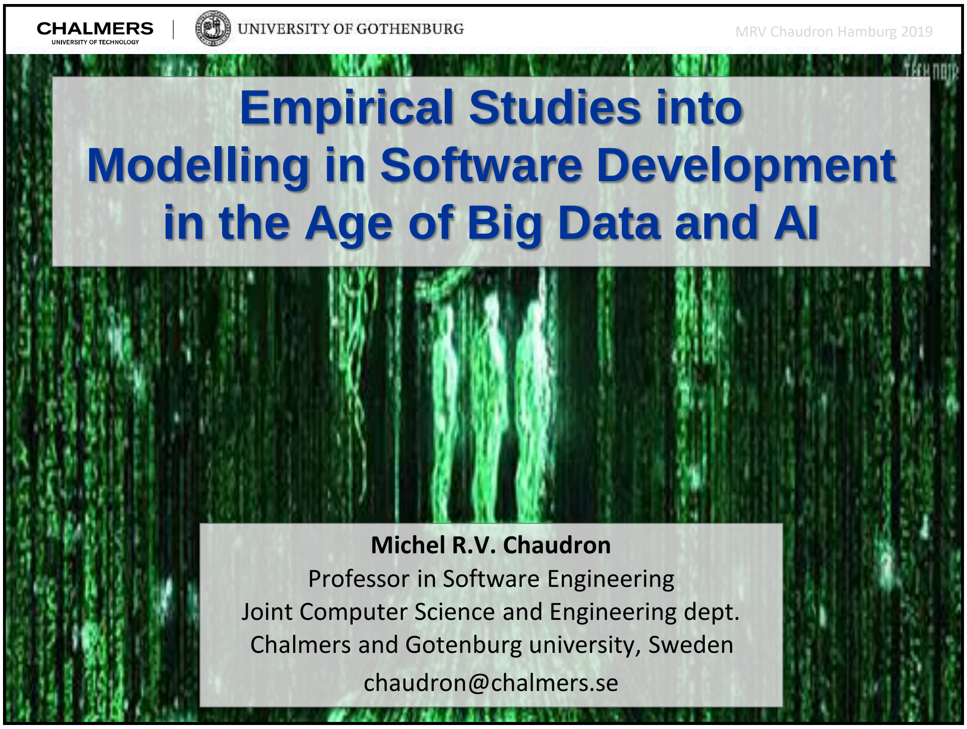
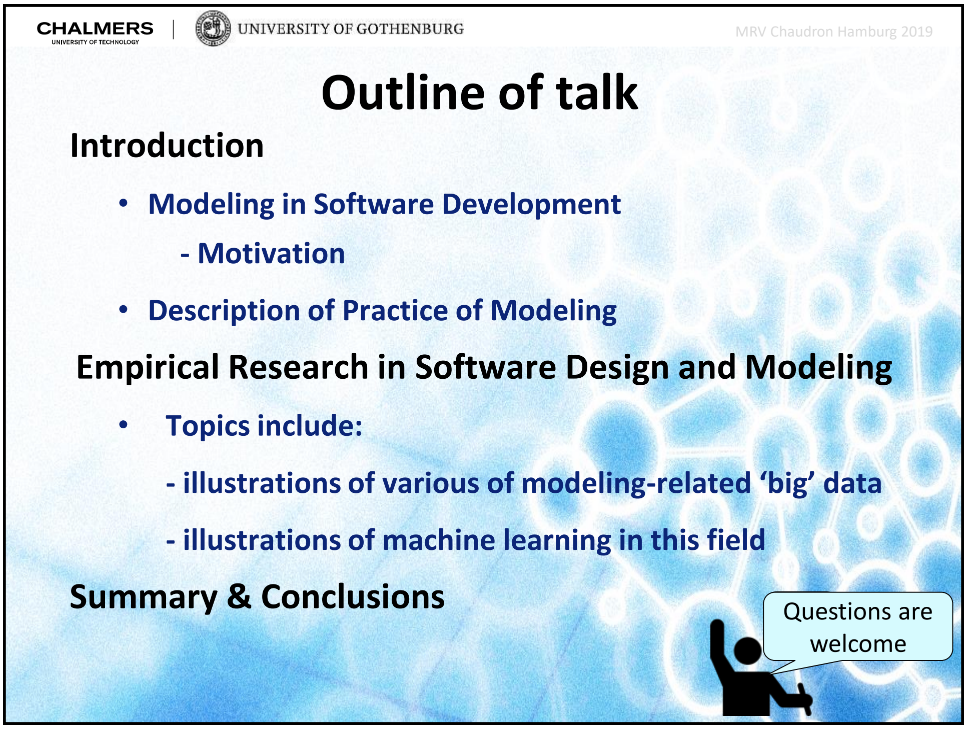


Empirical Studies into Modelling in Software Development in the Age of Big Data and AI

Michel R.V. Chaudron
Professor in Software Engineering
Joint Computer Science and Engineering dept.
Chalmers and Gotenburg university, Sweden
chaudron@chalmers.se



Outline of talk

Introduction

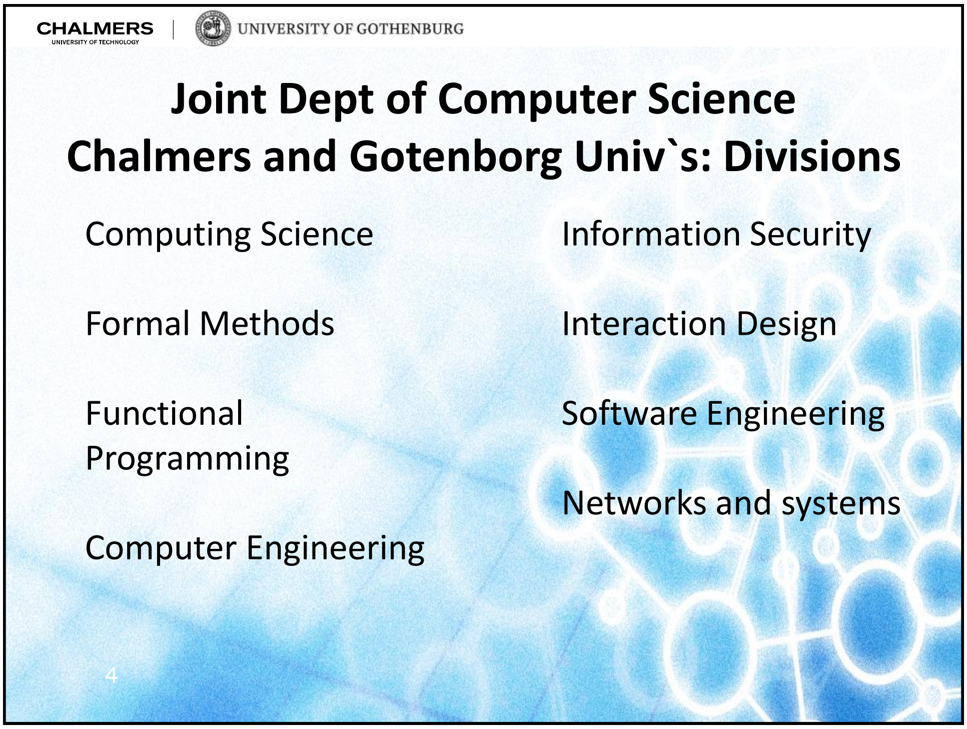
- Modeling in Software Development
  - Motivation
- Description of Practice of Modeling

Empirical Research in Software Design and Modeling

- Topics include:
  - illustrations of various of modeling-related 'big' data
  - illustrations of machine learning in this field

Summary & Conclusions

Questions are welcome



CHALMERS UNIVERSITY OF TECHNOLOGY | UNIVERSITY OF GOTHENBURG

Joint Dept of Computer Science Chalmers and Gotenborg Univ`s: Divisions

Computing Science

Formal Methods

Functional Programming

Computer Engineering

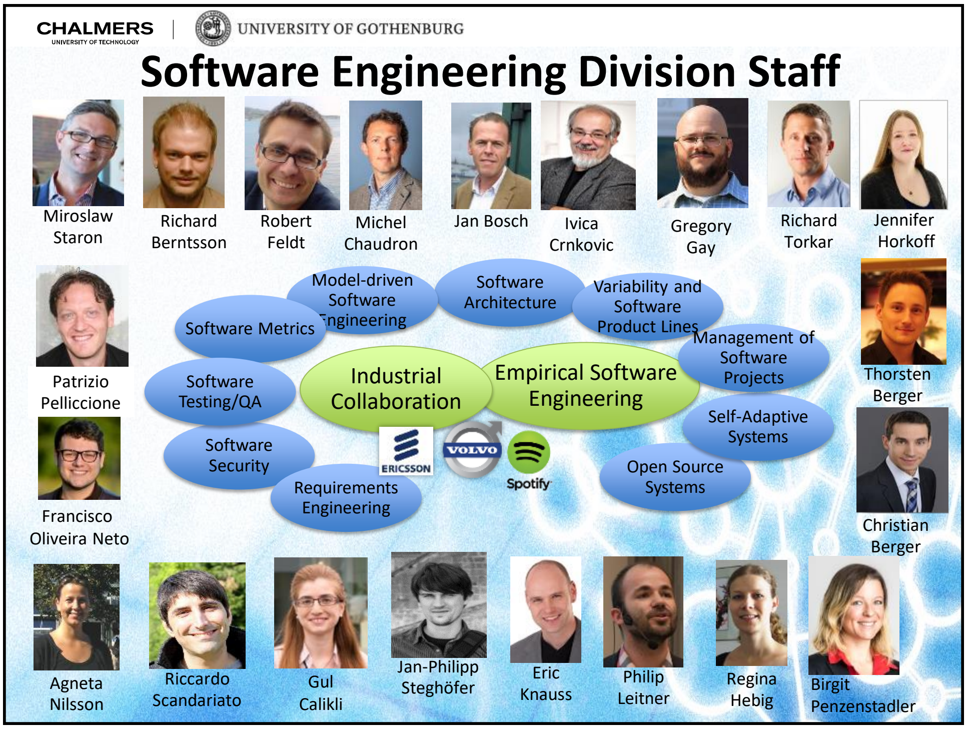
Information Security

Interaction Design

Software Engineering

Networks and systems

4



CHALMERS UNIVERSITY OF TECHNOLOGY | UNIVERSITY OF GOTHENBURG

Software Engineering Division Staff

Miroslaw Staron

Richard Berntsson

Robert Feldt

Michel Chaudron

Jan Bosch

Ivica Crnkovic

Gregory Gay

Richard Torkar

Jennifer Horkoff

Patrizio Pelliccione

Francisco Oliveira Neto

Thorsten Berger

Christian Berger

Model-driven Software Engineering

Software Architecture

Variability and Software Product Lines

Software Metrics

Management of Software Projects

Software Testing/QA

Industrial Collaboration

Empirical Software Engineering

Self-Adaptive Systems

Software Security

Requirements Engineering

Open Source Systems

ERICSSON

VOLVO

Spotify

Agneta Nilsson

Riccardo Scandariato

Gul Calikli

Jan-Philipp Steghöfer

Eric Knauss

Philip Leitner

Regina Hebig

Birgit Penzenstadler

CHALMERS UNIVERSITY OF TECHNOLOGY | UNIVERSITY OF GOTHENBURG

SOFTWARE ENGINEERING CONFERENCES IN GOTHENBURG

Open Source 2016
REFSQ 2016
ICSA 2017 http://icsa-conferencesorg/2017/
Mensura 2017 http://www.iwsm-mensura.org/2017
IFIPTM 2017 Int. Conf. on Trust Management
http://wp.portal.chalmers.se/ifiptm2017/

ICSA 2017
April 3-7 Gothenburg, Sweden
ICSE 40TH INTERNATIONAL CONFERENCE ON SOFTWARE ENGINEERING
SPLC Gothenburg 2018
22nd Systems and Software Product Line Conference

+ yearly Lindholmen Software Development day (600+)

CHALMERS UNIVERSITY OF TECHNOLOGY | UNIVERSITY OF GOTHENBURG

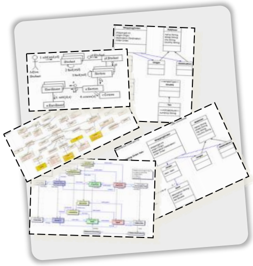
Software Engineering in Africa (SEiA)

- Software Engineering in Africa workshop at ICSE
- SIDA BRIGHT project (2015-2010)
  - Supervise 10 lecturers in Uganda towards their PhD degree.
  - Yearly summerschools in East-Africa

- Tentative plans:
  African Summerschool on Software Engineering 2020

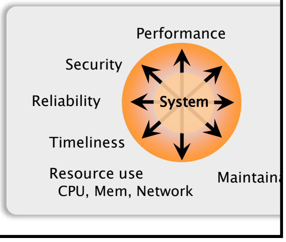
Introduction: Research Interests

- **What are the pay-offs of investing in architecture/design/modeling?**
  Fewer defects?
  Cheaper maintenance? …



- **Extract design knowledge from software repositories** (models, text, …)

- **Analysis and Reasoning about Quality Properties of System Architectures**
  - Automating Architecture Design Optimization by Genetic Algorithms
  - Maintainability, Technical Debt



Performance

Security

Reliability

System

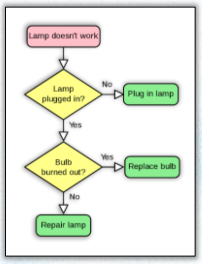
Timeliness

Resource use
CPU, Mem, Network

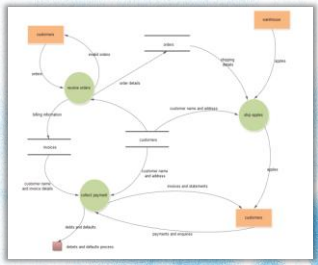
Maintaina

Modeling in Software Engineering

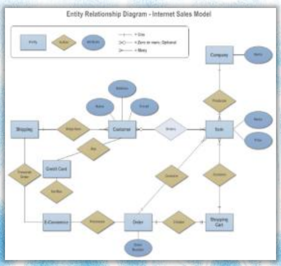
Long history of modeling in software engineering:



Flow Chart (dd 1921!)



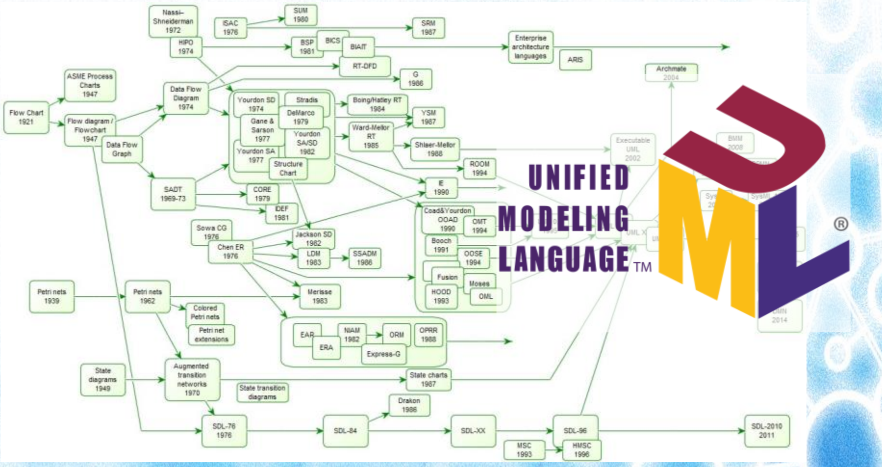
Data Flow Diagrams (dd 1970)



Entity-Relation Diagram (1976)

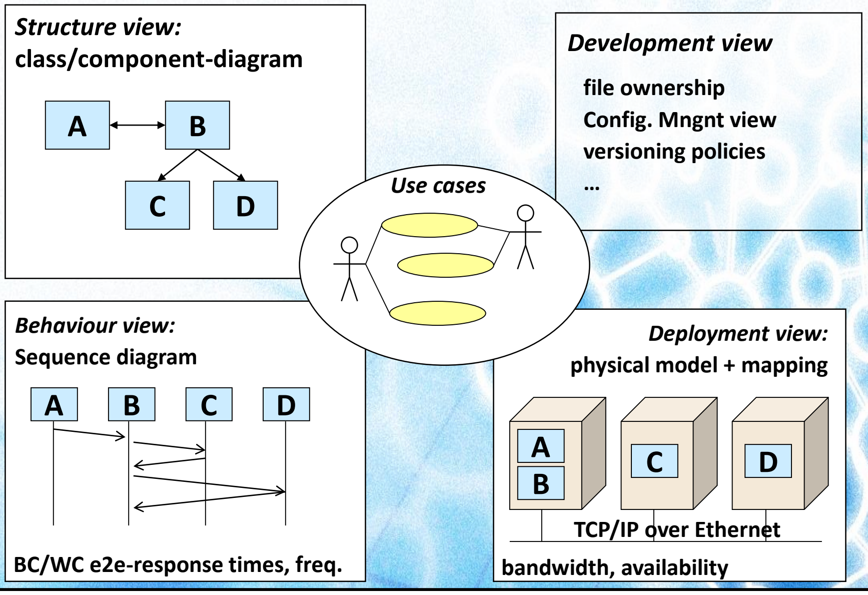
Models of software capture structure and behaviour

Modelling History



Juha-Pekka Tolvanen, CEO Meta-Case, Keynote at Code Generation 2014:
The business cases of modeling and generators

Example 4+1 Views model



Structure view:
class/component-diagram

Development view

file ownership
Config. Mngnt view
versioning policies
…

Use cases

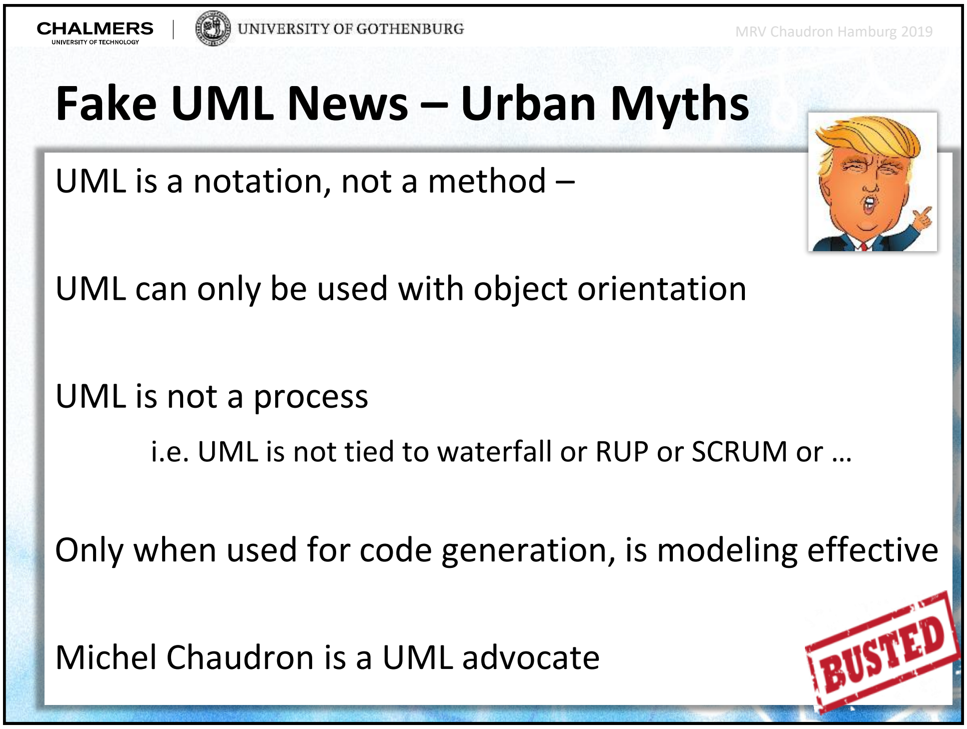
Behaviour view:
Sequence diagram

BC/WC e2e-response times, freq.

Deployment view:
physical model + mapping

TCP/IP over Ethernet

bandwidth, availability



Fake UML News – Urban Myths

UML is a notation, not a method –

UML can only be used with object orientation

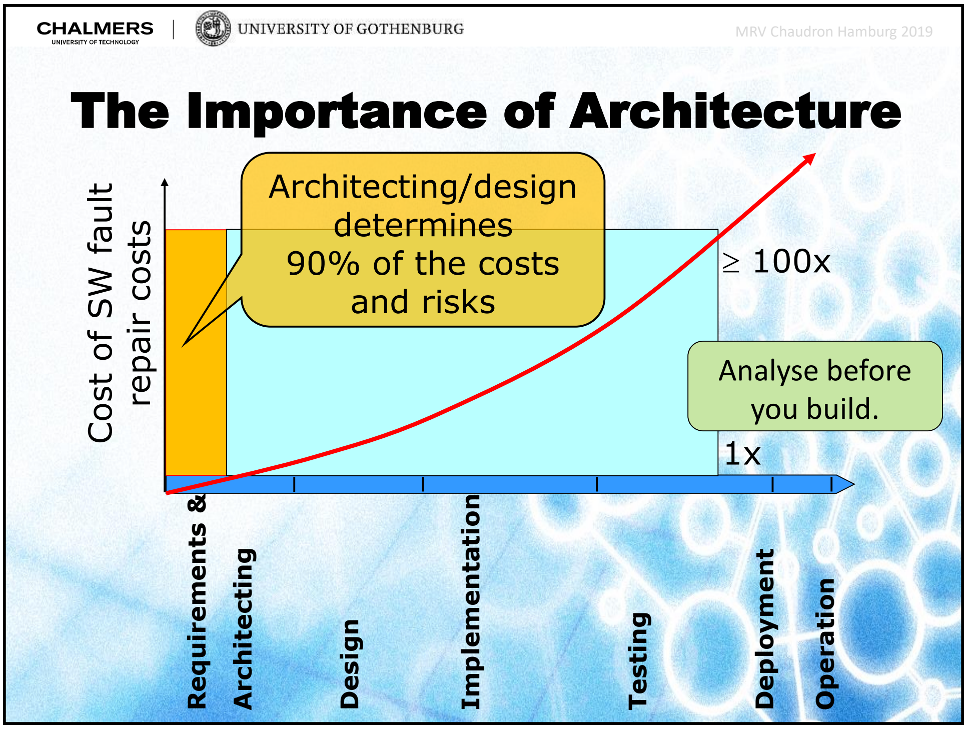
UML is not a process

i.e. UML is not tied to waterfall or RUP or SCRUM or …

Only when used for code generation, is modeling effective

Michel Chaudron is a UML advocate

BUSTED

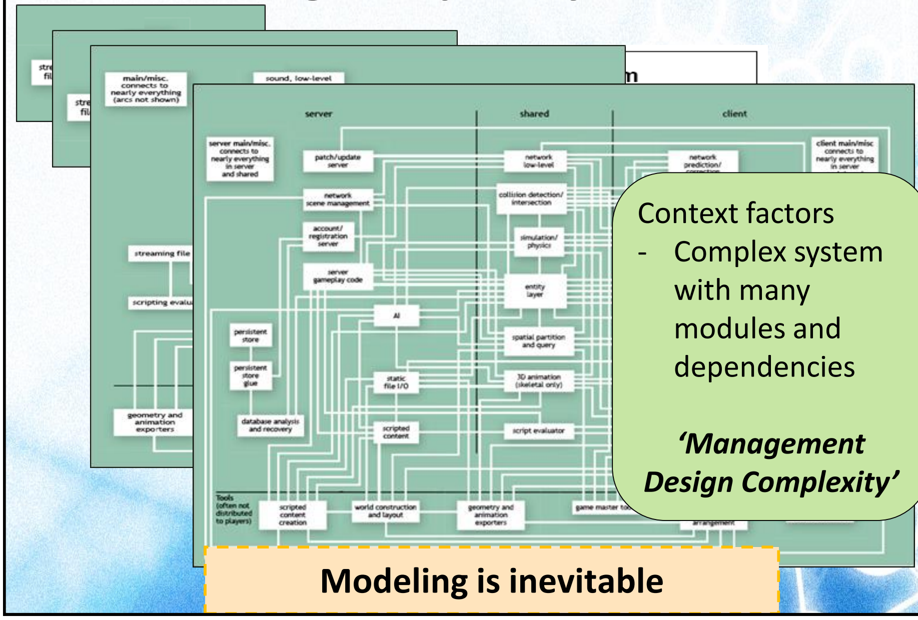


The Importance of Architecture

Architecting/design determines 90% of the costs and risks

Analyse before you build.

Increasing Complexity of Software


main/misc. connects to nearly everything (arcs not shown)
sound, low-level
streaming file
scripting evalu
geometry and animation exporters
server
shared
client
server main/misc. connects to nearly everything in server and shared
patch/update server
network low-level
network prediction/correction
client main/misc connects to nearly everything in server
network scene management
collision detection/intersection
account/registration server
simulation/physics
server gameplay code
entity layer
AI
persistent store
spatial partition and query
persistent store glue
static file I/O
3D animation (skeletal only)
database analysis and recovery
scripted content
script evaluator
Tools (often not distributed to players)
scripted content creation
world construction and layout
geometry and animation exporters
game master too
arrangement

Context factors
- Complex system with many modules and dependencies

'Management Design Complexity'

Modeling is inevitable

Blow, J. "Game Development: Harder Than You Think." *ACM Queue*

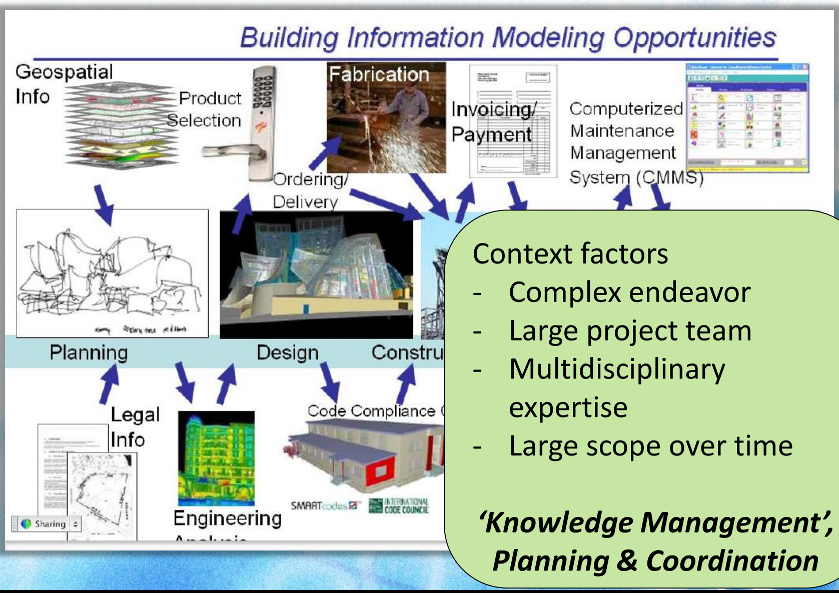
Motivation

Why should we care about modeling software design?

Pro-Modeling-camp:
"I can not imagine we could develop our systems without any modeling."

Anti-Modeling-camp:
"Models are useless. Code is the only truth."

Modeling & Design in Civil Engineering



Context factors
- Complex endeavor
- Large project team
- Multidisciplinary expertise
- Large scope over time

‘Knowledge Management’, Planning & Coordination

Summary of Arguments in favour of Modelling in Software Development

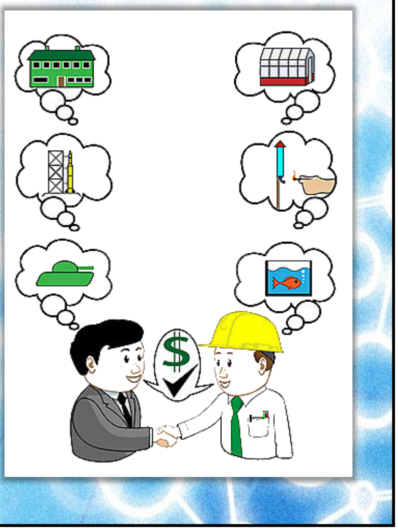
+ Communicating / Coordination

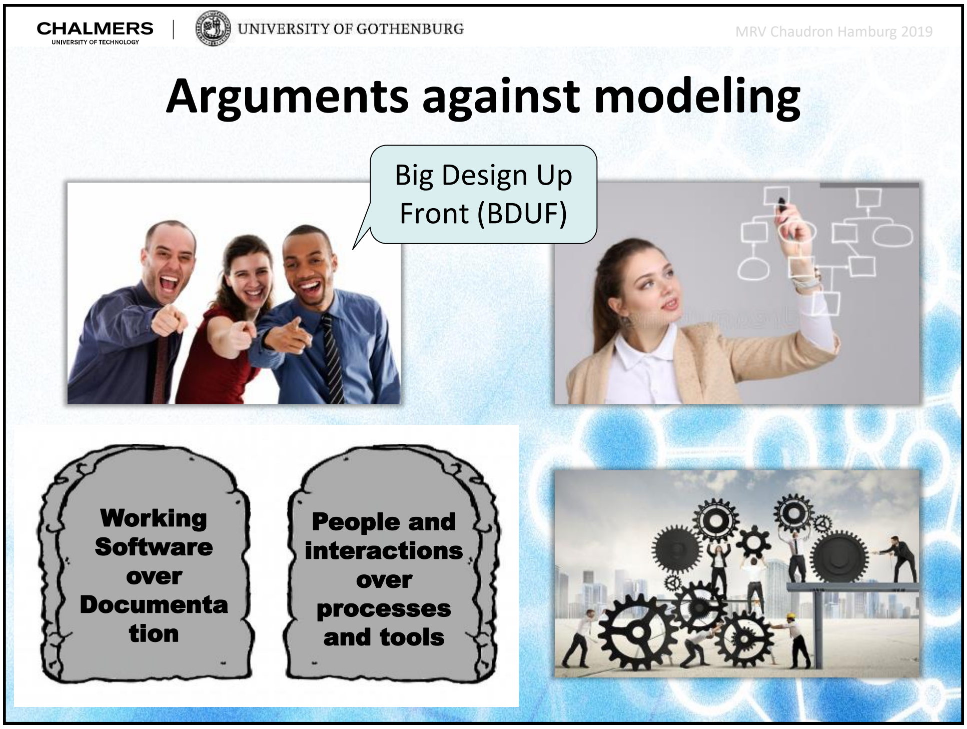
+ Analysing / Predicting

+ Understanding / Structuring

+ Guiding Construction

+ Blueprint for Production





CHALMERS UNIVERSITY OF TECHNOLOGY | UNIVERSITY OF GOTHENBURG

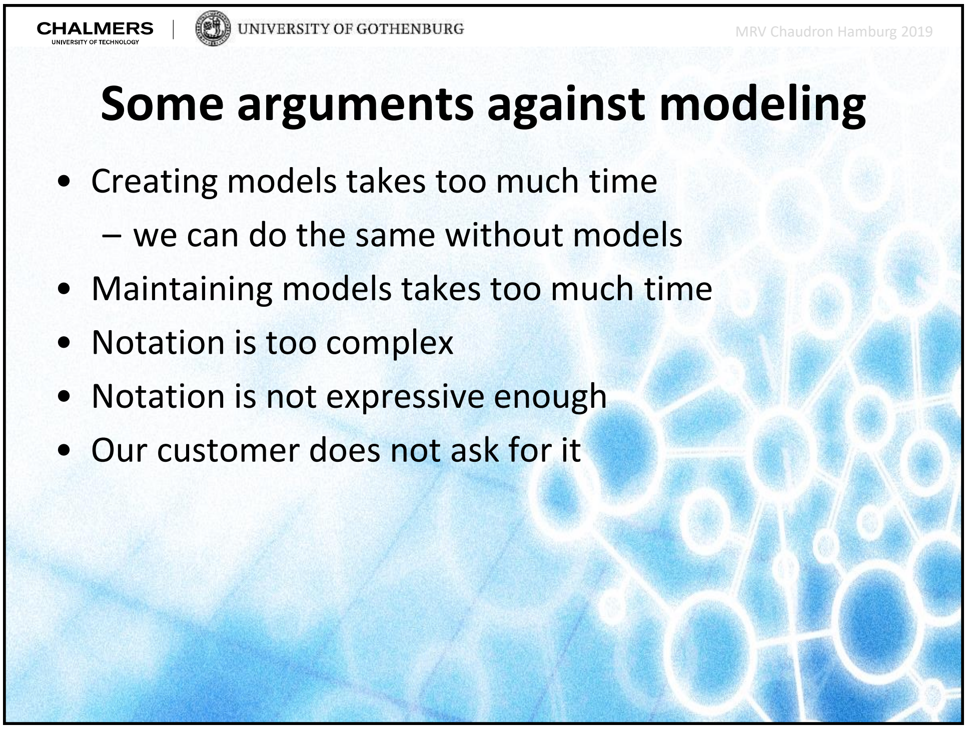
MRV Chaudron Hamburg 2019

Arguments against modeling

Big Design Up Front (BDUF)

Working Software over Documentation

People and interactions over processes and tools



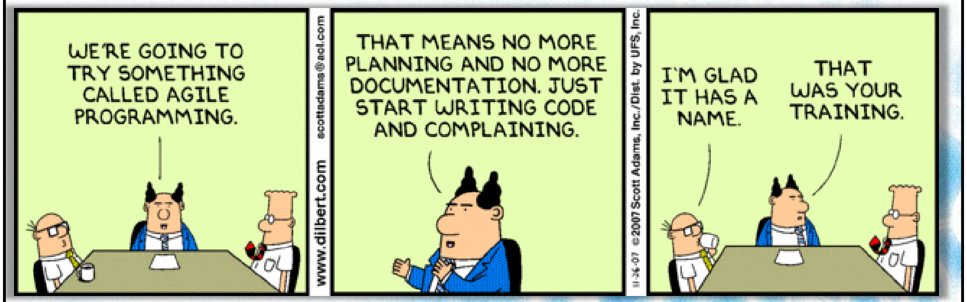
CHALMERS UNIVERSITY OF TECHNOLOGY | UNIVERSITY OF GOTHENBURG

MRV Chaudron Hamburg 2019

Some arguments against modeling

- Creating models takes too much time
  – we can do the same without models
- Maintaining models takes too much time
- Notation is too complex
- Notation is not expressive enough
- Our customer does not ask for it

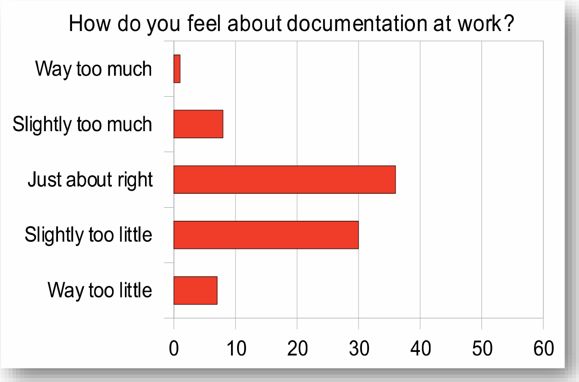
Not always rational …


WE'RE GOING TO TRY SOMETHING CALLED AGILE PROGRAMMING.
scottadams@aol.com
www.dilbert.com
THAT MEANS NO MORE PLANNING AND NO MORE DOCUMENTATION. JUST START WRITING CODE AND COMPLAINING.
11-26-07 ©2007 Scott Adams, Inc./Dist. by UFS, Inc.
I'M GLAD IT HAS A NAME.
THAT WAS YOUR TRAINING.

Modeling & Documentation in Agile Development

- Agile principles:

  working software over comprehensive documentation



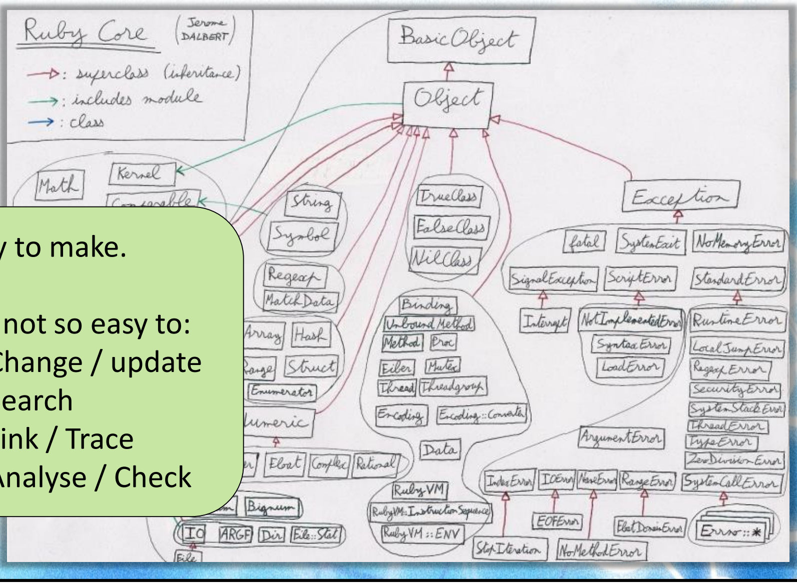
Survey under 75+ agile developers

Tending towards: we need a bit more

Modeling is compatible with agile development

Christoph J. Stettina and Werner Heijstek, Necessary and Neglected? An Empirical Study of Internal Documentation in Agile Software Development Teams 29th ACM Int. Conf. on Design of Communication , Pisa, Italy

Informal (whiteboard) drawing

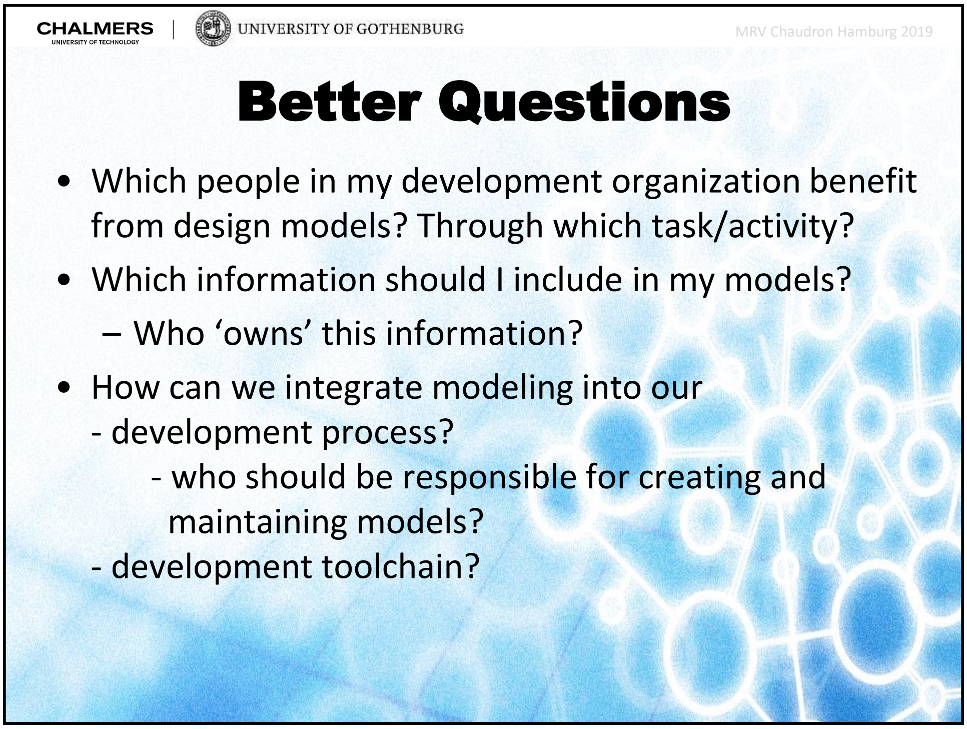


Easy to make.

But not so easy to:
- Change / update
- Search
- Link / Trace
- Analyse / Check

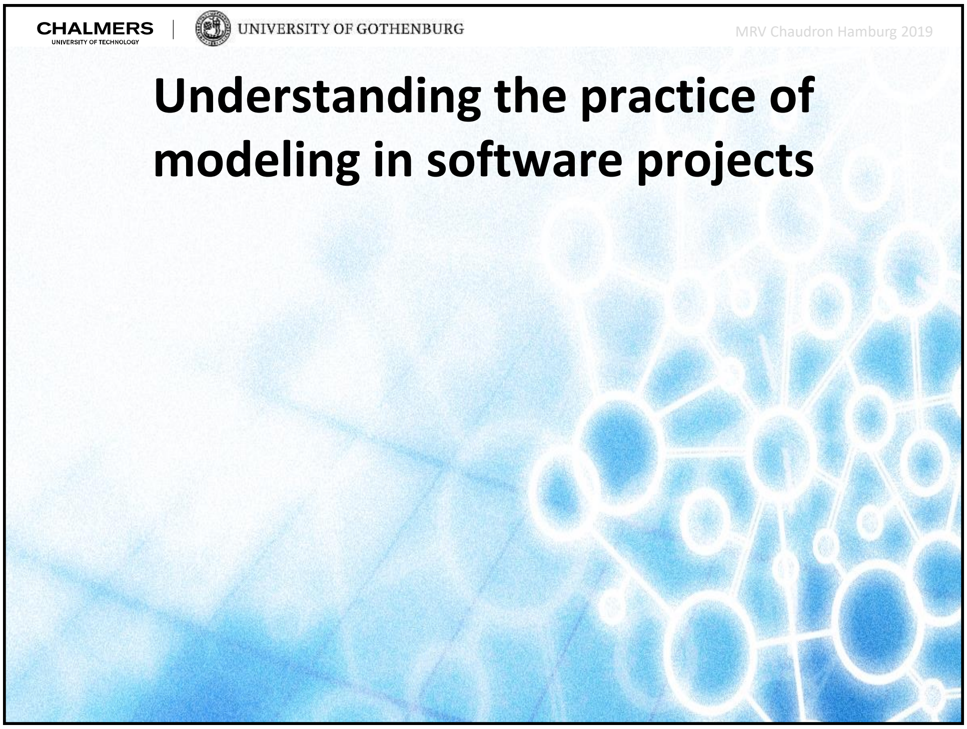
To Model or Not to Model?

That is NOT the question



Better Questions

- Which people in my development organization benefit from design models? Through which task/activity?
- Which information should I include in my models?
  - Who ‘owns’ this information?
- How can we integrate modeling into our
  - development process?
    - who should be responsible for creating and maintaining models?
  - development toolchain?



Understanding the practice of modeling in software projects

The Model-based SE Spectrum

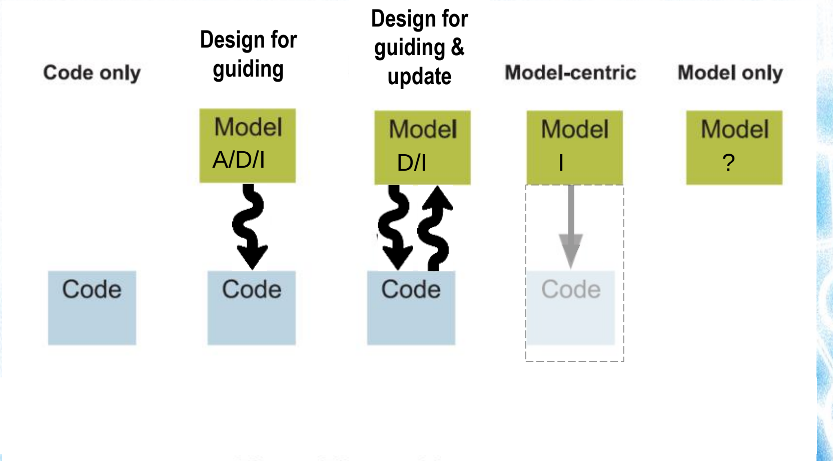
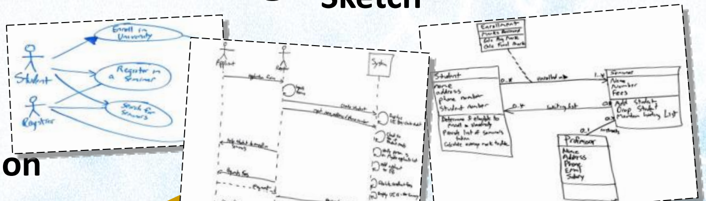


Fig. 1. The modeling spectrum

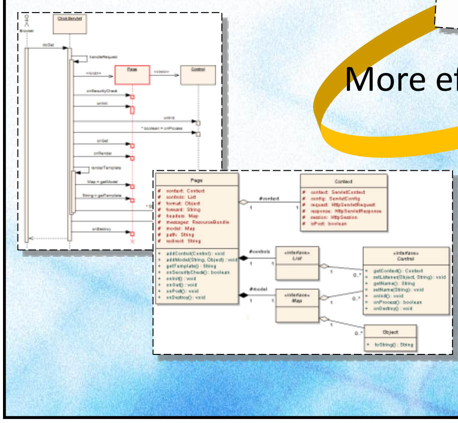
Model driven architecture: Principles and practice, Brown, A.W., Software and Systems Modeling, 2004

Styles of UML modeling

Sketch

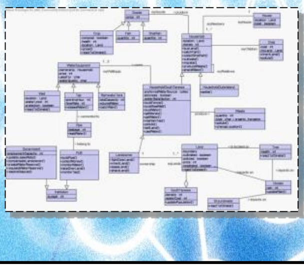


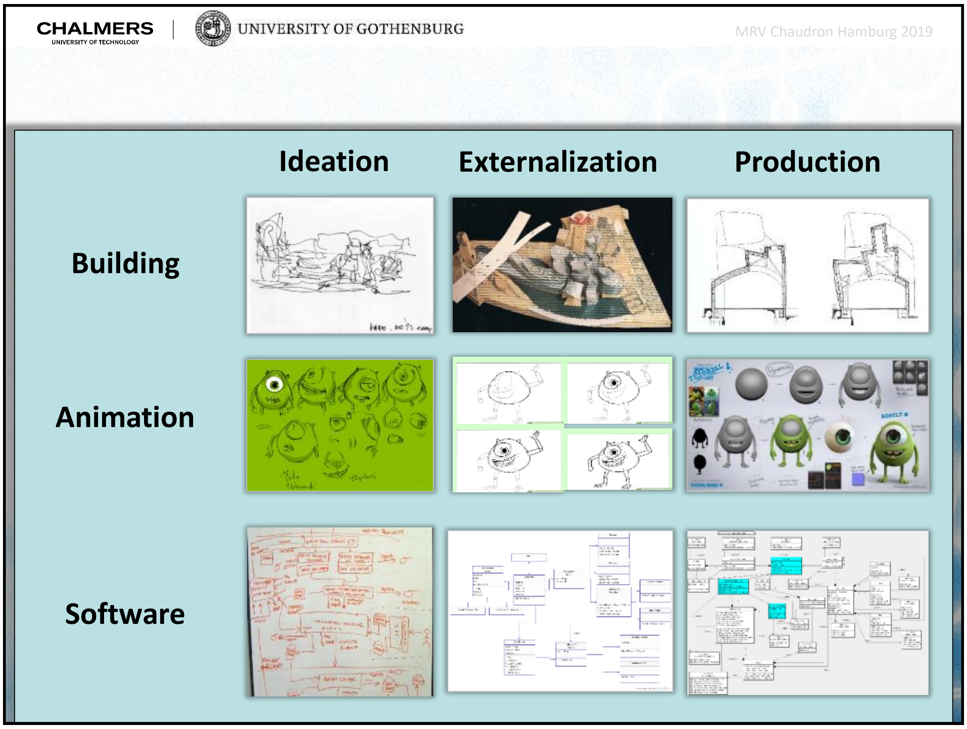
Communication



More effort ⇨ More expensive

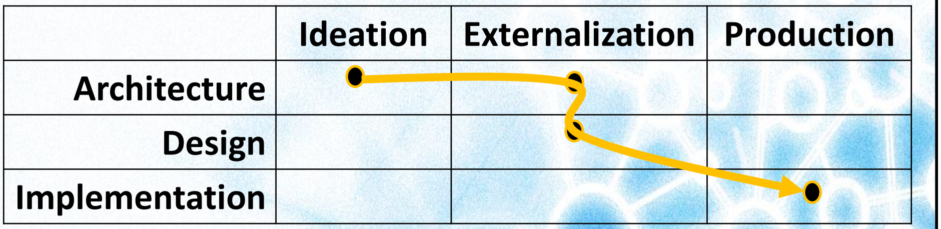
Recipe for construction





	Ideation	Externalization	Production
Building			
Animation			
Software			

Evolution of Models during a project

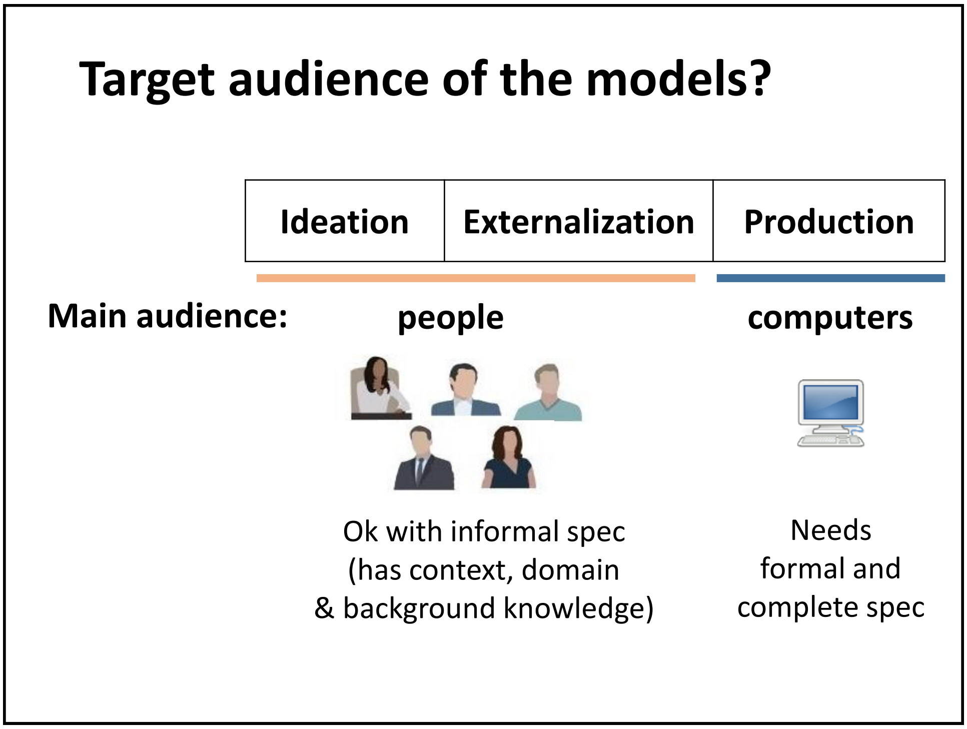


	Ideation	Externalization	Production
Architecture			
Design			
Implementation			

Different stages have different needs on formality

Yet tasks are connected:

'smooth' handover over design representation is needed



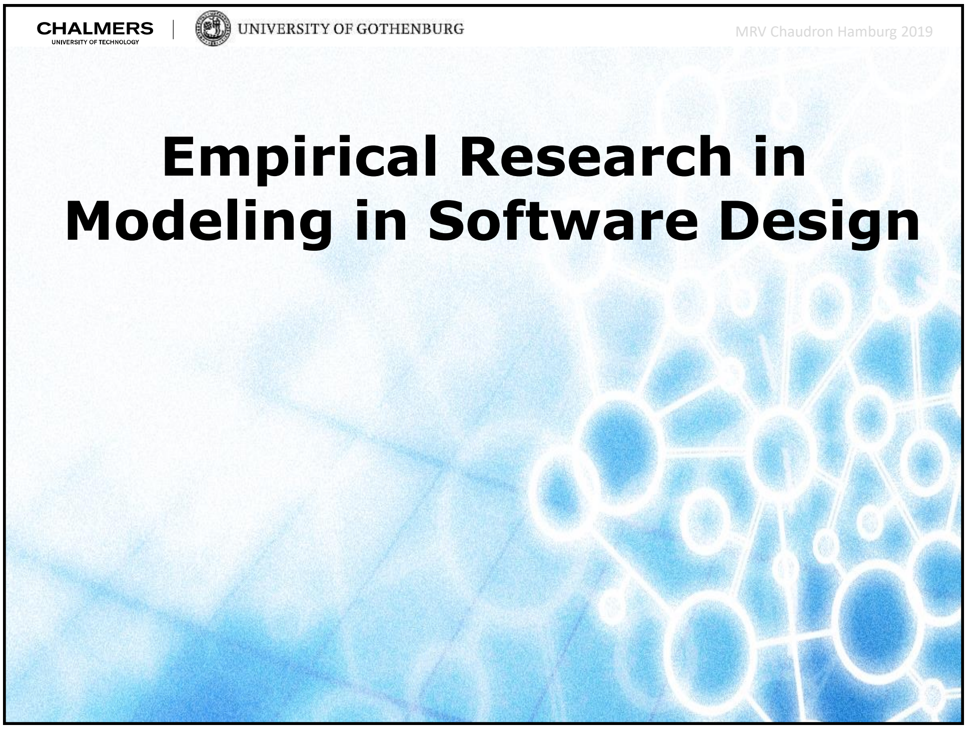
Target audience of the models?

Ideation	Externalization	Production

Main audience: people computers

people: Ok with informal spec (has context, domain & background knowledge)

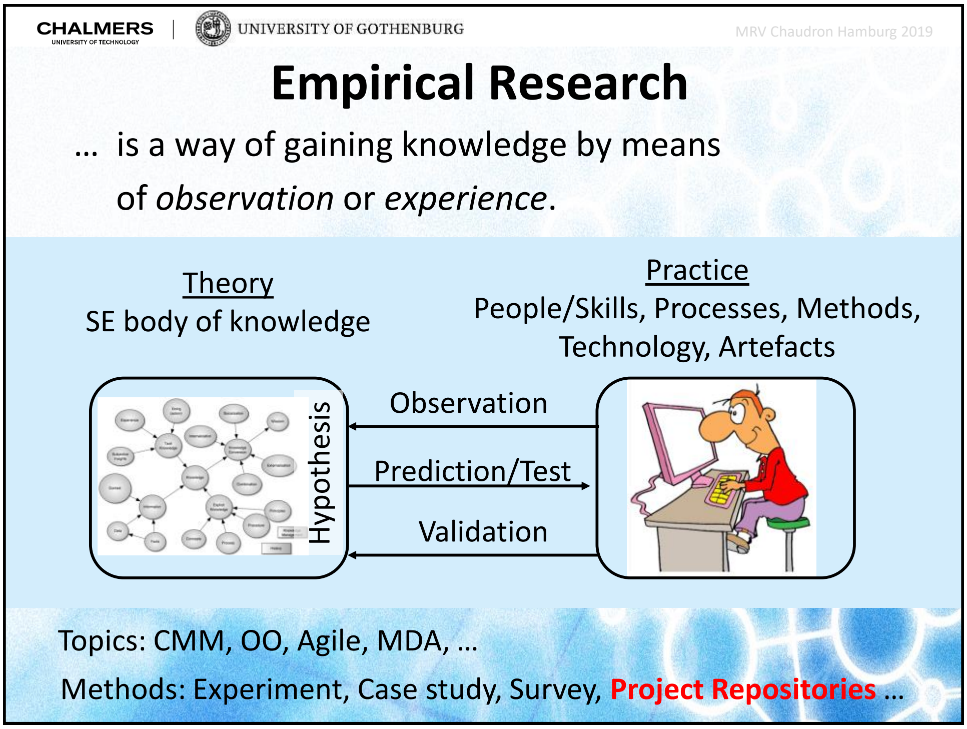
computers: Needs formal and complete spec



CHALMERS UNIVERSITY OF TECHNOLOGY | UNIVERSITY OF GOTHENBURG

MRV Chaudron Hamburg 2019

Empirical Research in Modeling in Software Design

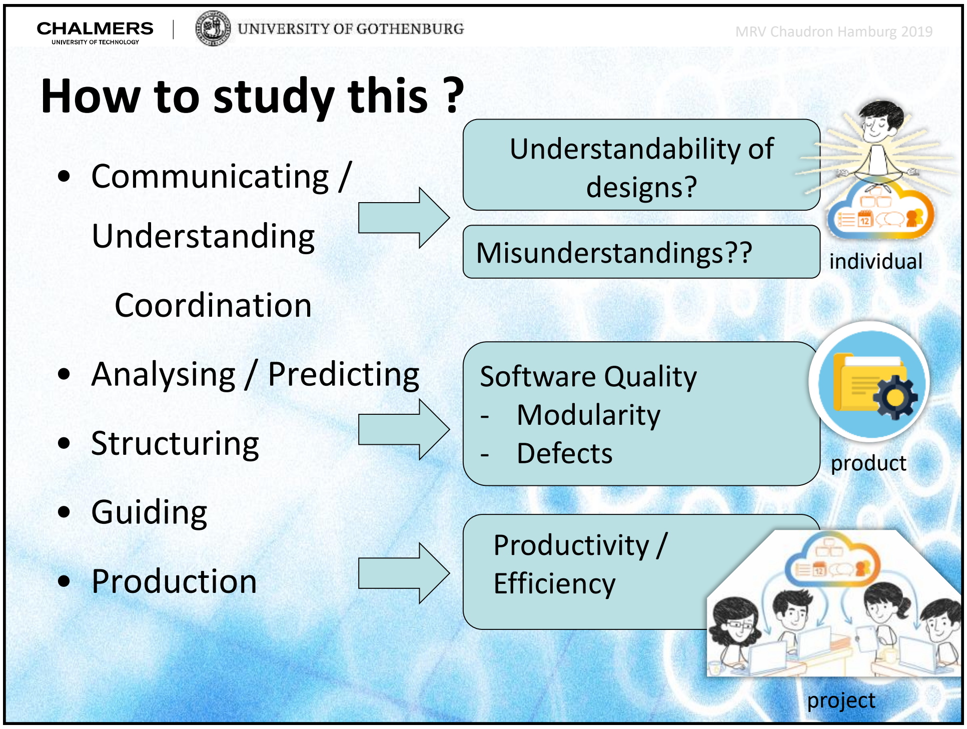


Empirical Research

… is a way of gaining knowledge by means of *observation* or *experience*.

Topics: CMM, OO, Agile, MDA, …

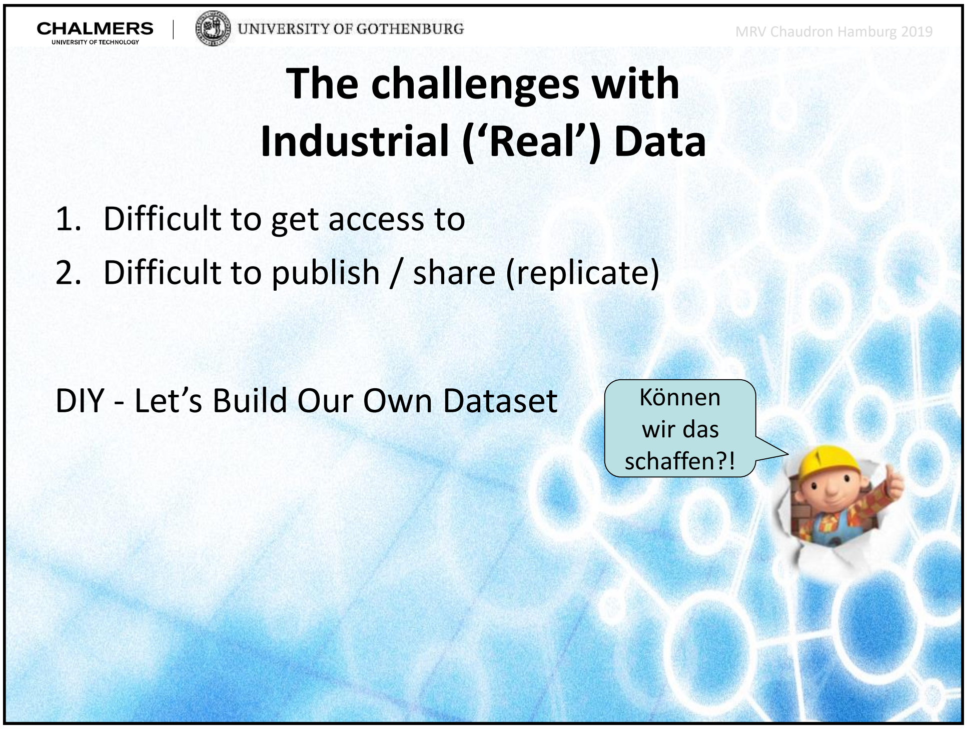
Methods: Experiment, Case study, Survey, **Project Repositories** …



How to study this ?

- Communicating / Understanding

  Coordination
- Analysing / Predicting
- Structuring
- Guiding
- Production



CHALMERS UNIVERSITY OF TECHNOLOGY | UNIVERSITY OF GOTHENBURG

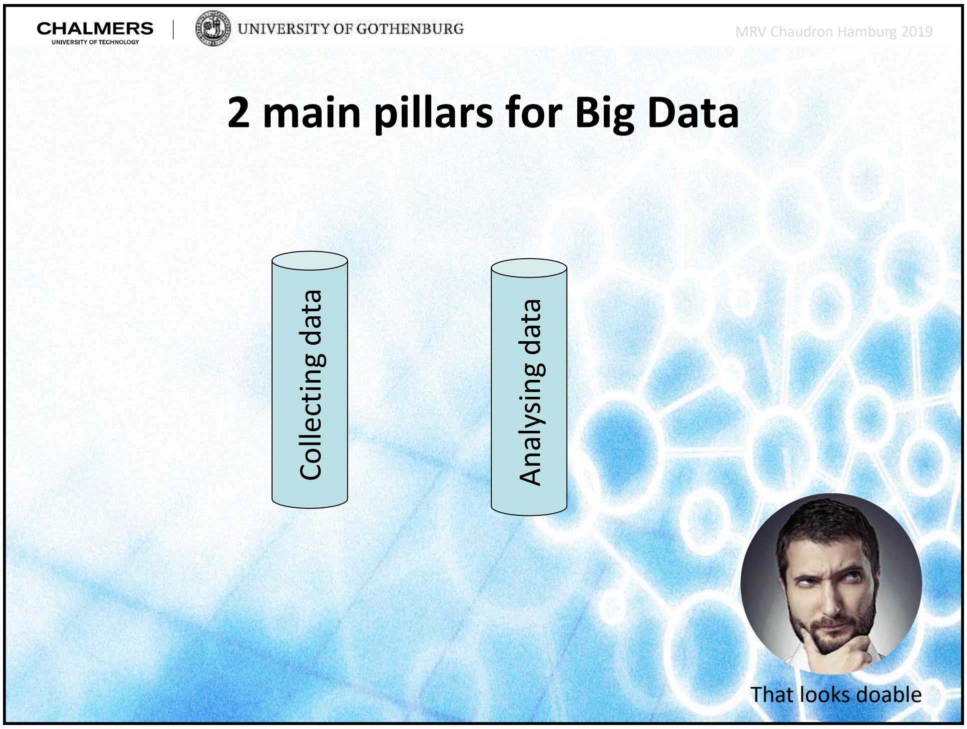
MRV Chaudron Hamburg 2019

The challenges with Industrial ('Real') Data

1. Difficult to get access to
2. Difficult to publish / share (replicate)

DIY - Let's Build Our Own Dataset

Können wir das schaffen?!



CHALMERS UNIVERSITY OF TECHNOLOGY | UNIVERSITY OF GOTHENBURG

MRV Chaudron Hamburg 2019

2 main pillars for Big Data

Collecting data

Analysing data

That looks doable

Collecting Big Data

Bilal Karasneh

Idea: This is a ‘Search’-problem.

Probably Google can do this for us.

Attempt 0.1 & 0.2:

1. Search for (filetype) .xmi using Google

   Unsuccessful: incredibly much that is not about ‘XMI’ for UML

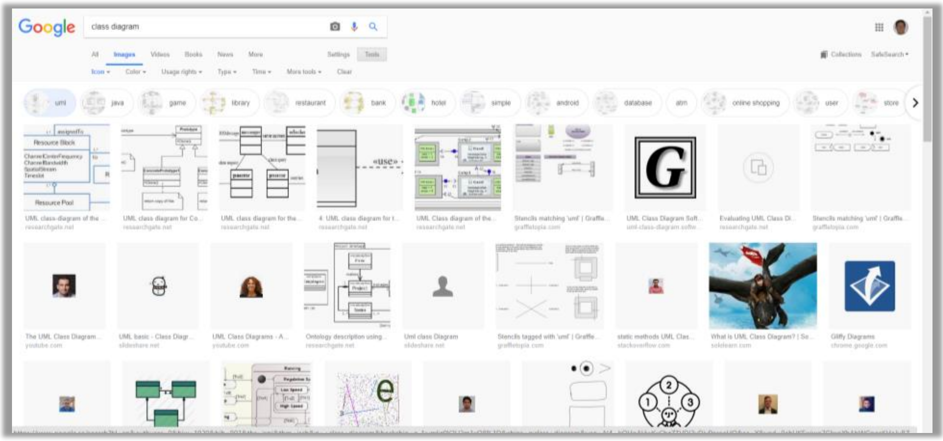
   Manual filtering is very time-consuming

2. Search for ‘class diagram’ via Google Image Search

   + : many images can be found, not all are class diagrams

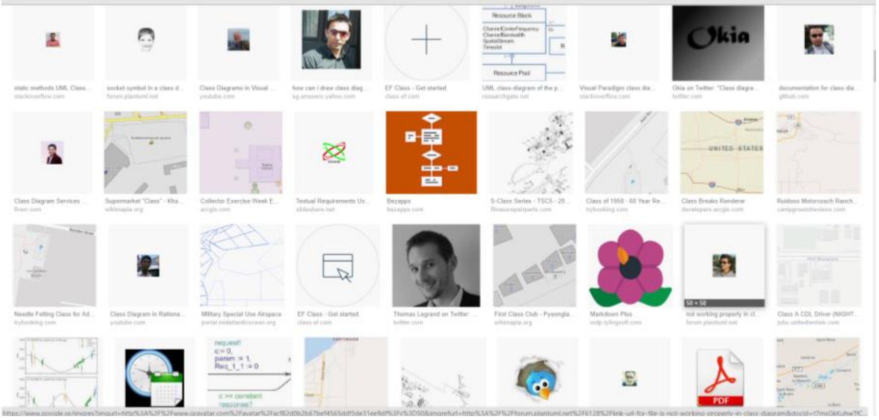
   - : very high variation in quality
     tutorial/lecture-slides (on notation), student-projects, …

Google Image Results – page 1



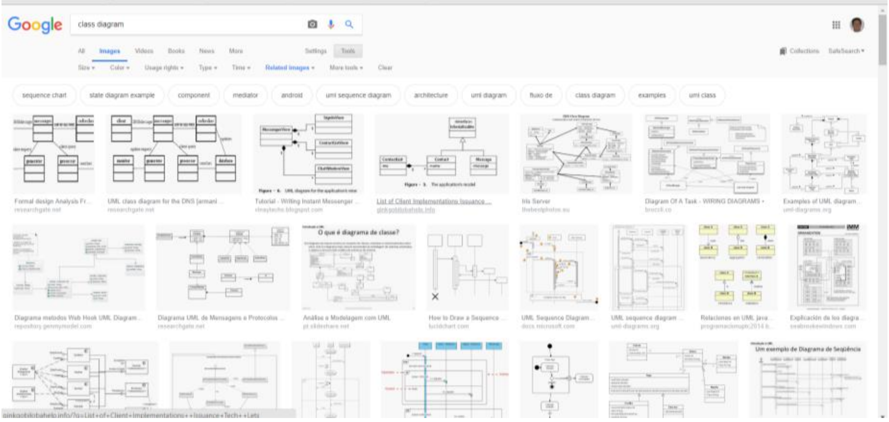
class diagram

Google Image Results – **page 2**



Many many false positives.

Google Image Results – **page 3**



class diagram

Still many false positives.

After much manual filtering: collection of about 1000 diagrams.

Classifier for recognizing UML Class diagrams

With my students:
- Ingimar Samúelsson
- Jóel Hjaltason

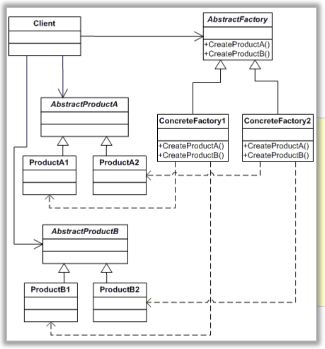
- Input:
  arbitrary image (.jpeg)
- Output:
  Yes/No (=UML Class Diagram)
- Initial filtering done by rules
  – Size (no icons), colours
- 1300 Diagrams, of which 650 UML CD and 650 other (incl. Seq.D, ER, charts, …)

Features include:
Contours & Shapes
Horizontal and Vertical Lines
Rectangle-H-W-ratio
Connected rectangles
Rectangles with 'divider-line'
NON-Hor/Ver lines
% of area used

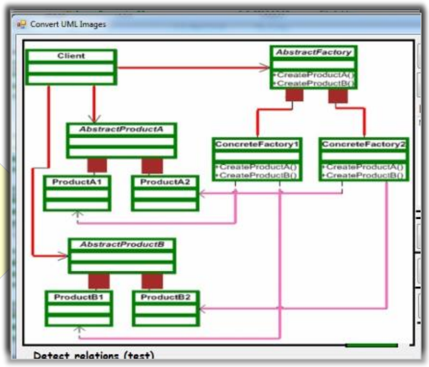
Algorithm	Precision
Random Forrest	0.95

Automated Extraction of Models from Images

Bilal Karasneh



Recognizing:
- Rectangles
- Lines
- Text,
- arrow-heads



Input: .jpeg, .png

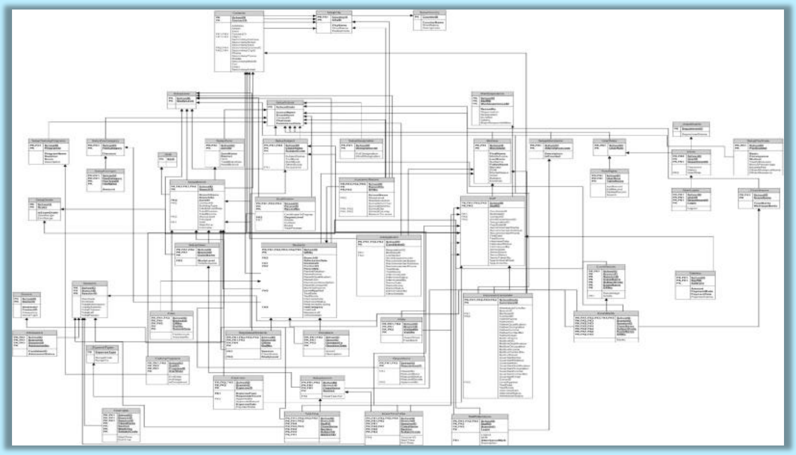
Output: XMI for the class model

Milestone: IMG2UML Tool

Is Reverse Engineering a solution?

- Automatically generated from source code, hence always up-to-date!
- But, …

Reverse Engineering of a small system



Clearly different from forward designed UML designs
(o.a. in size, layout, detail, naming, ….)

CHALMERS UNIVERSITY OF TECHNOLOGY | UNIVERSITY OF GOTHENBURG

Classifier for UML Class Diagram Styles using Machine Learning

- Forward vs Reverse Engineered Diagram Classifier
- 16 Features
  - mostly related to 'parameters' of methods in diagram
- Sample of 999 class diagrams
  - Reverse : 806
  - Forward : 193

Algorithm	Accuracy	Precision	Recall	F-measure
Random Forrest	90.74	0.95	0.93	0.94

CHALMERS UNIVERSITY OF TECHNOLOGY | UNIVERSITY OF GOTHENBURG

Collecting data

The Quest for UML in Open Source Projects

-- Mining GitHub --

Regina Hebig *

Truong Ho-Quang *

Miguel-Angel Fernandez +

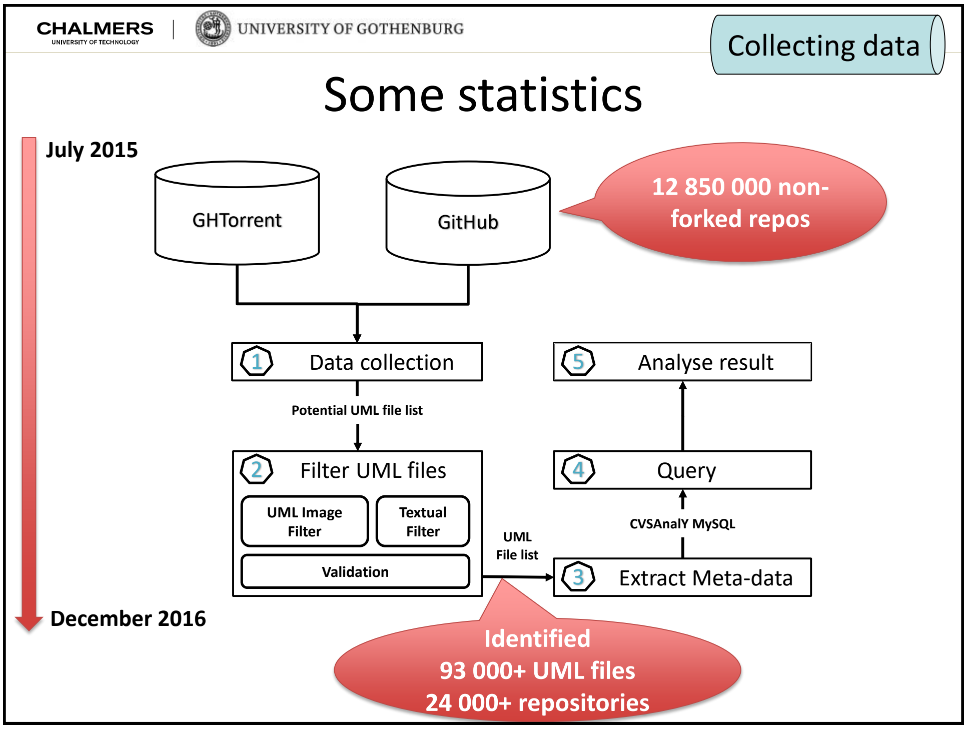
Gregorio Robles +

Michel R.V. Chaudron *

(*) Gothenburg and Chalmers University

(+) Universidad Rey Juan Carlos

Hebig, Regina, et al. "The quest for open source projects that use UML: mining GitHub." *Proceedings of MODELS* 2016.



Collecting data

Some statistics

July 2015

GHTorrent

GitHub

12 850 000 non-forked repos

1 Data collection

Potential UML file list

2 Filter UML files

UML Image Filter

Textual Filter

Validation

UML File list

3 Extract Meta-data

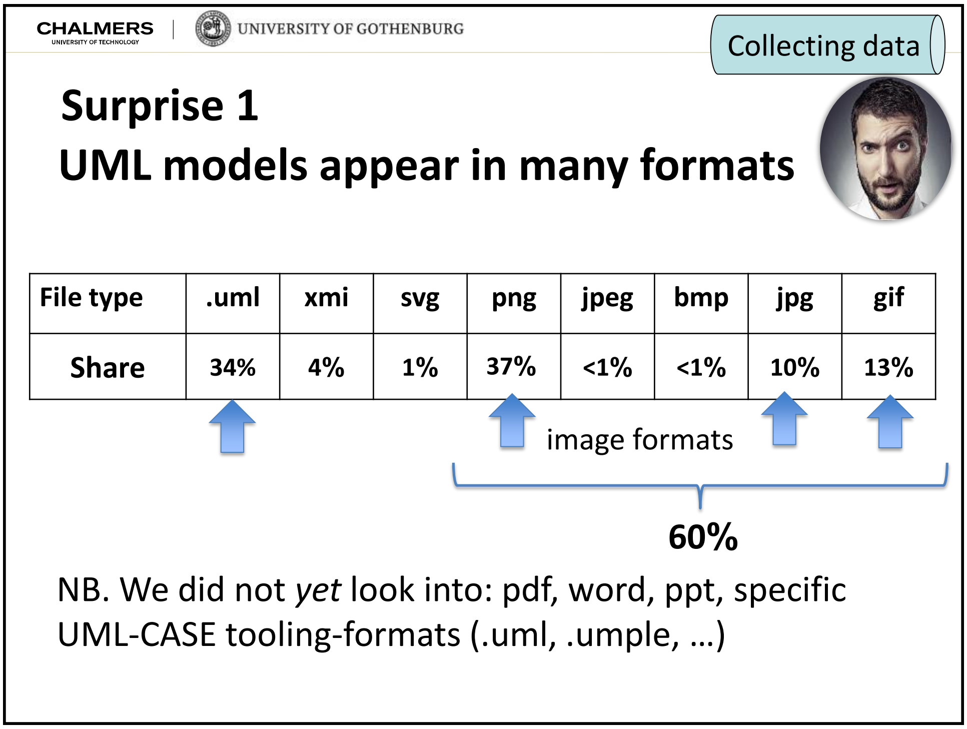
CVSAnalY MySQL

4 Query

5 Analyse result

December 2016

Identified
93 000+ UML files
24 000+ repositories



Collecting data

Surprise 1
UML models appear in many formats

File type	.uml	xmi	svg	png	jpeg	bmp	jpg	gif
Share	34%	4%	1%	37%	<1%	<1%	10%	13%

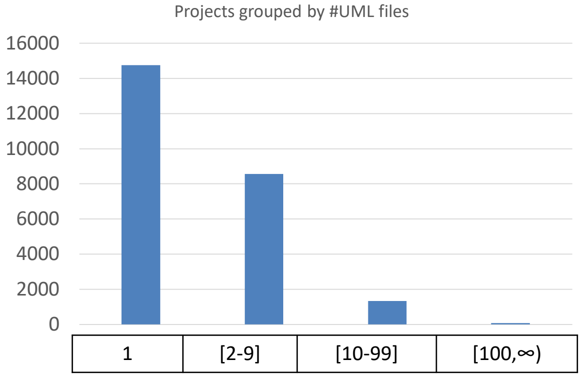
image formats

60%

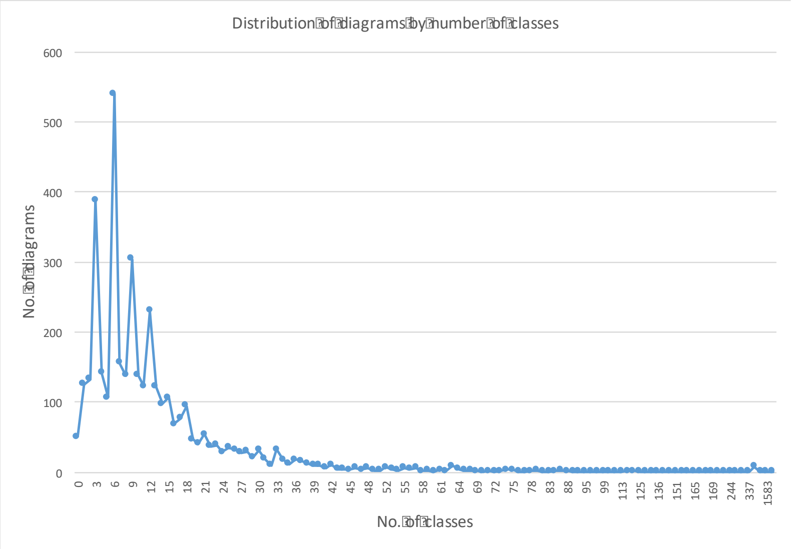
NB. We did not *yet* look into: pdf, word, ppt, specific UML-CASE tooling-formats (.uml, .umple, …)

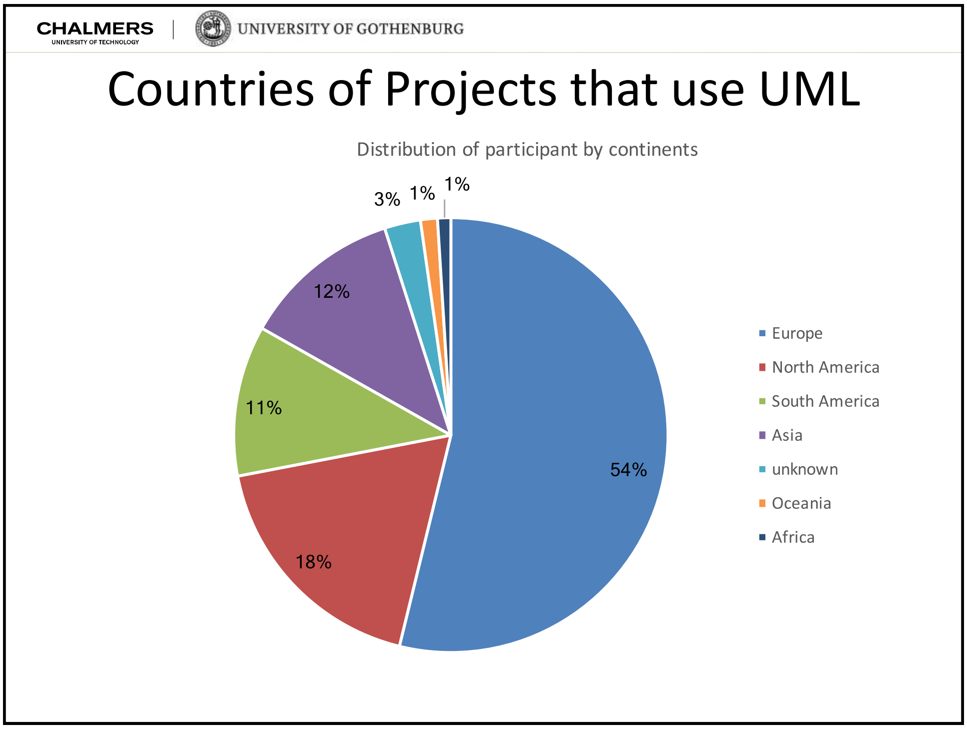
Projects grouped by number of UML files

#UML files	1	[2-9]	[10-99]	[100,∞)
#Repositories	14 749	8 567	1 333	68



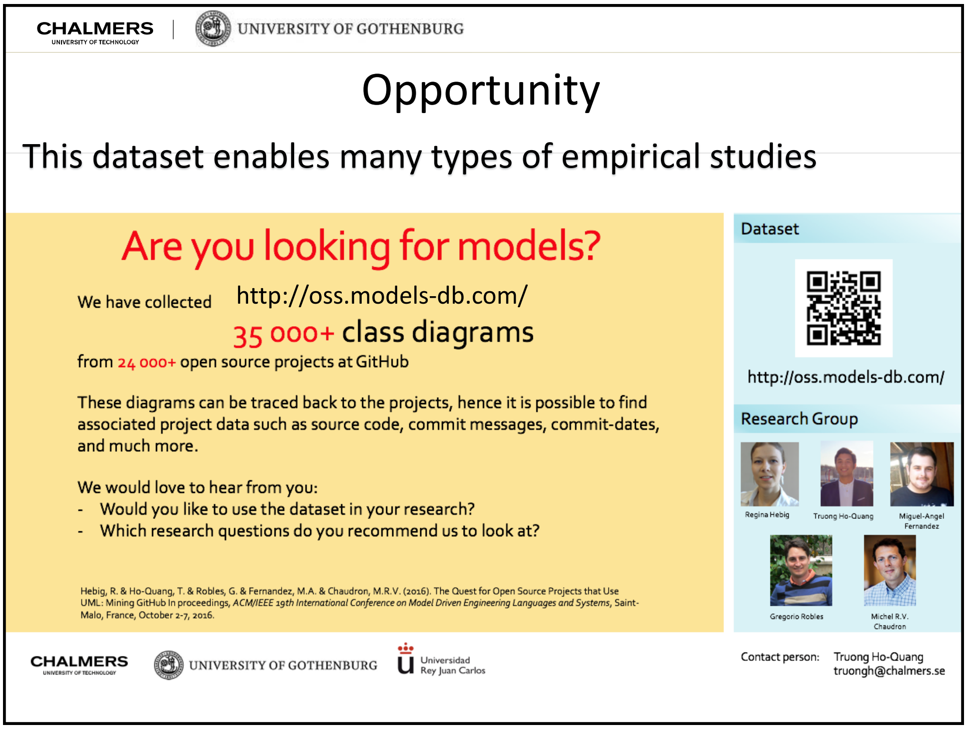
Example analysis: size of class diagrams





CHALMERS UNIVERSITY OF TECHNOLOGY | UNIVERSITY OF GOTHENBURG

Countries of Projects that use UML



CHALMERS UNIVERSITY OF TECHNOLOGY | UNIVERSITY OF GOTHENBURG

Opportunity

This dataset enables many types of empirical studies

Are you looking for models?

We have collected http://oss.models-db.com/

35 000+ class diagrams

from 24 000+ open source projects at GitHub

These diagrams can be traced back to the projects, hence it is possible to find associated project data such as source code, commit messages, commit-dates, and much more.

We would love to hear from you:

- Would you like to use the dataset in your research?
- Which research questions do you recommend us to look at?

Hebig, R. & Ho-Quang, T. & Robles, G. & Fernandez, M.A. & Chaudron, M.R.V. (2016). The Quest for Open Source Projects that Use UML: Mining GitHub In proceedings, *ACM/IEEE 19th International Conference on Model Driven Engineering Languages and Systems*, Saint-Malo, France, October 2-7, 2016.

Dataset

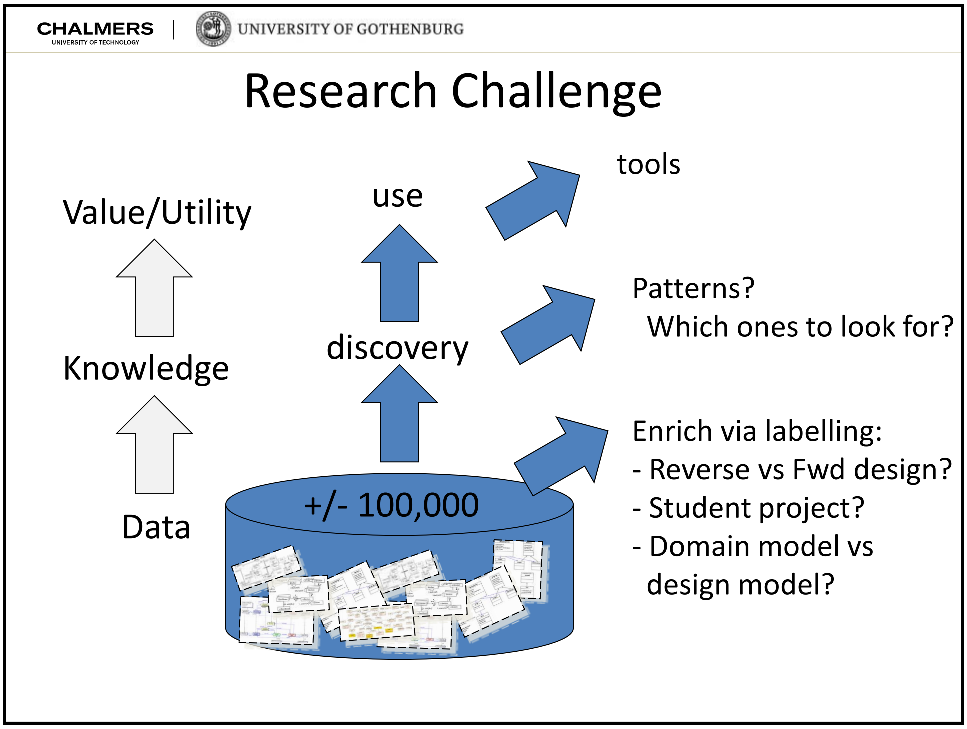
http://oss.models-db.com/

Research Group

Regina Hebig
Truong Ho-Quang
Miguel-Angel Fernandez
Gregorio Robles
Michel R.V. Chaudron

CHALMERS UNIVERSITY OF TECHNOLOGY | UNIVERSITY OF GOTHENBURG | Universidad Rey Juan Carlos

Contact person: Truong Ho-Quang
truongh@chalmers.se



Research Challenge

Value/Utility

Knowledge

Data

use

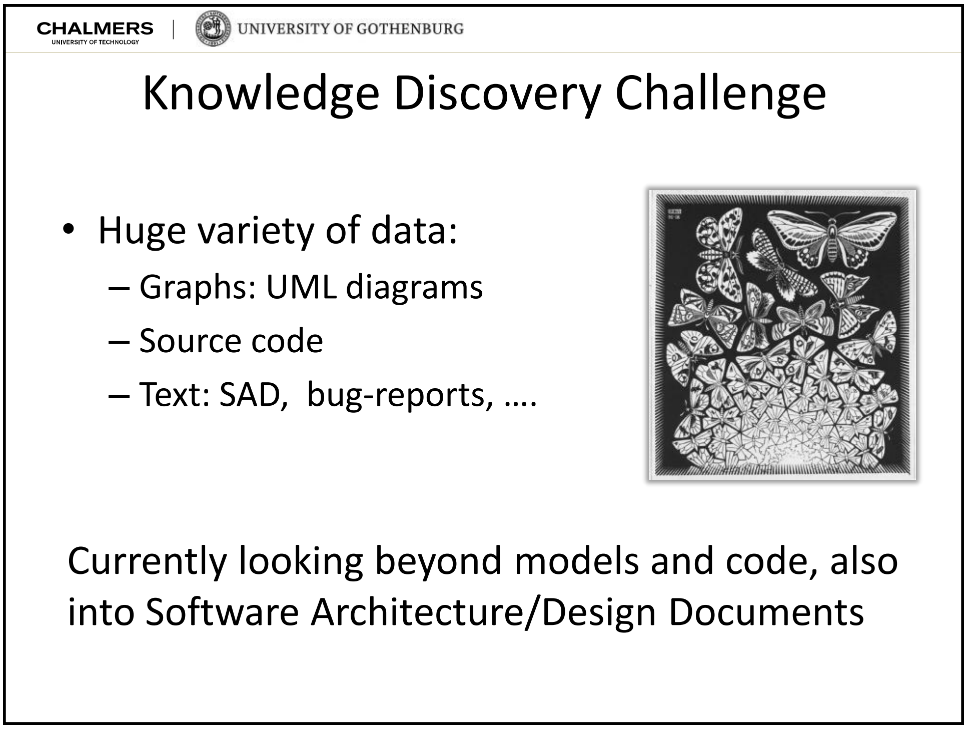
discovery

+/- 100,000

tools

Patterns?
Which ones to look for?

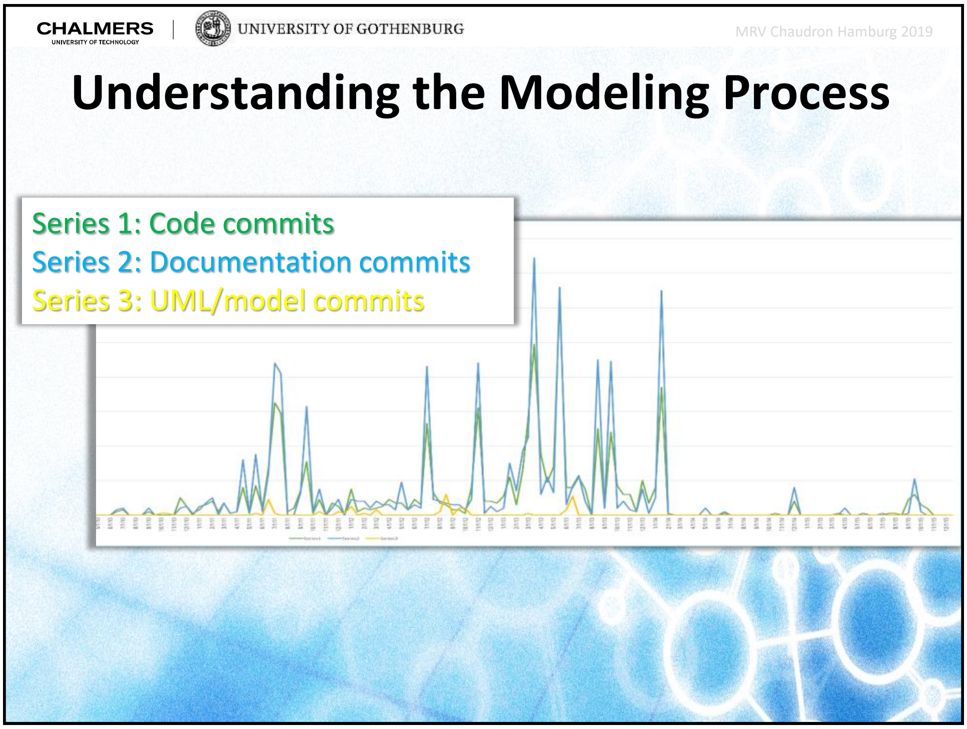
Enrich via labelling:
- Reverse vs Fwd design?
- Student project?
- Domain model vs design model?



Knowledge Discovery Challenge

- Huge variety of data:
  - Graphs: UML diagrams
  - Source code
  - Text: SAD, bug-reports, ….

Currently looking beyond models and code, also into Software Architecture/Design Documents

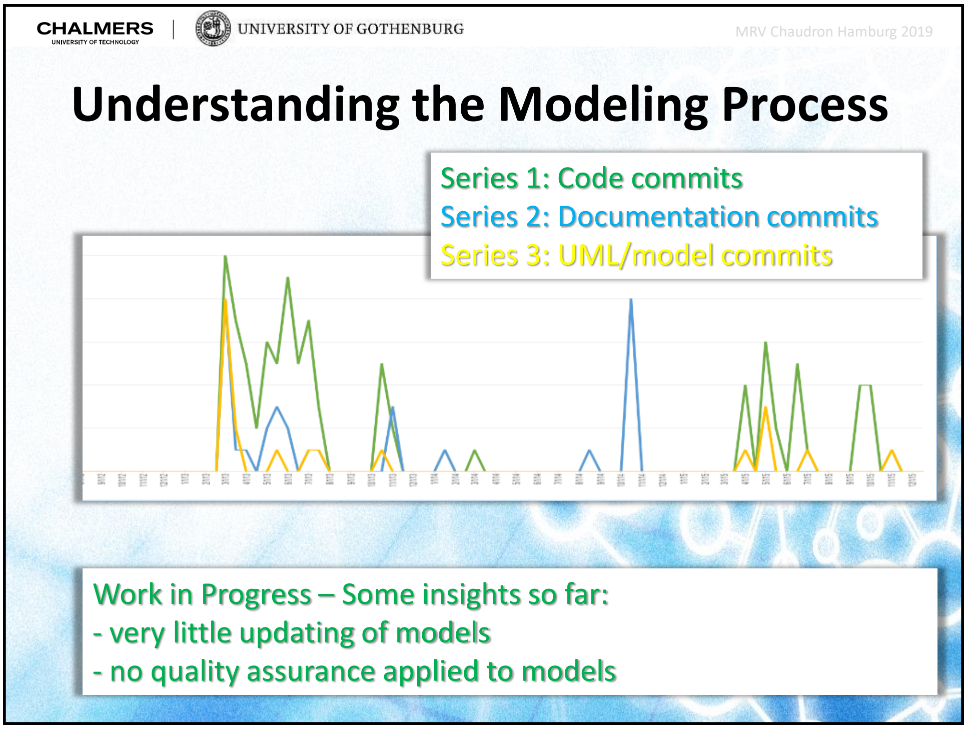


CHALMERS UNIVERSITY OF TECHNOLOGY | UNIVERSITY OF GOTHENBURG

MRV Chaudron Hamburg 2019

Understanding the Modeling Process

Series 1: Code commits
Series 2: Documentation commits
Series 3: UML/model commits



CHALMERS UNIVERSITY OF TECHNOLOGY | UNIVERSITY OF GOTHENBURG

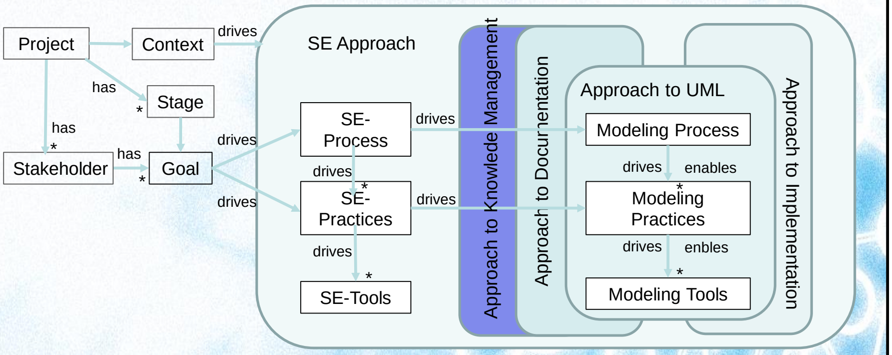
MRV Chaudron Hamburg 2019

Understanding the Modeling Process

Series 1: Code commits
Series 2: Documentation commits
Series 3: UML/model commits

Work in Progress – Some insights so far:
- very little updating of models
- no quality assurance applied to models

Approach to Modeling is contextual

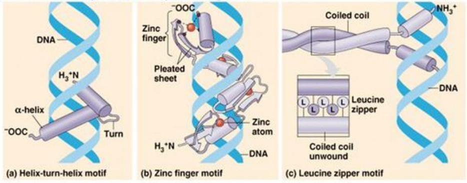


Is there a DNA for Software Design?

"Sequence motifs are short recurring patterns in DNA that are presumed to have biological function."

D'haeseleer, P. *Nature Biotechnology* **24**, 423 - 425 (2006).

protein motifs that help bind to DNA major & minor grooves



(a) Helix-turn-helix motif (b) Zinc finger motif (c) Leucine zipper motif

Image taken from bio.miami.edu

Challenge: Uncovering the hidden structure of designs

Rebecca Wirfs-Brock: Software designs are built from building blocks that have stereotypical responsibility-roles

Controller

Coordinator

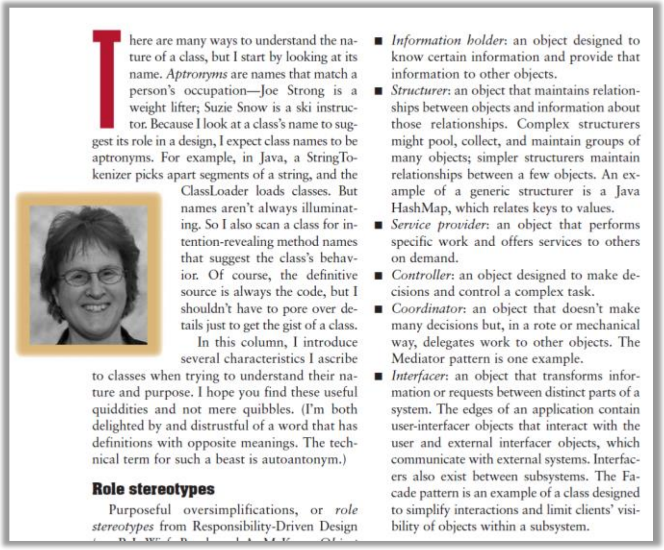
Interfacer

Structurer

Service Provider

Information Holder

Definition of Role Stereotypes



There are many ways to understand the nature of a class, but I start by looking at its name. *Aptronyms* are names that match a person's occupation—Joe Strong is a weight lifter; Suzie Snow is a ski instructor. Because I look at a class's name to suggest its role in a design, I expect class names to be aptronyms. For example, in Java, a StringTokenizer picks apart segments of a string, and the ClassLoader loads classes. But names aren't always illuminating. So I also scan a class for intention-revealing method names that suggest the class's behavior. Of course, the definitive source is always the code, but I shouldn't have to pore over details just to get the gist of a class.

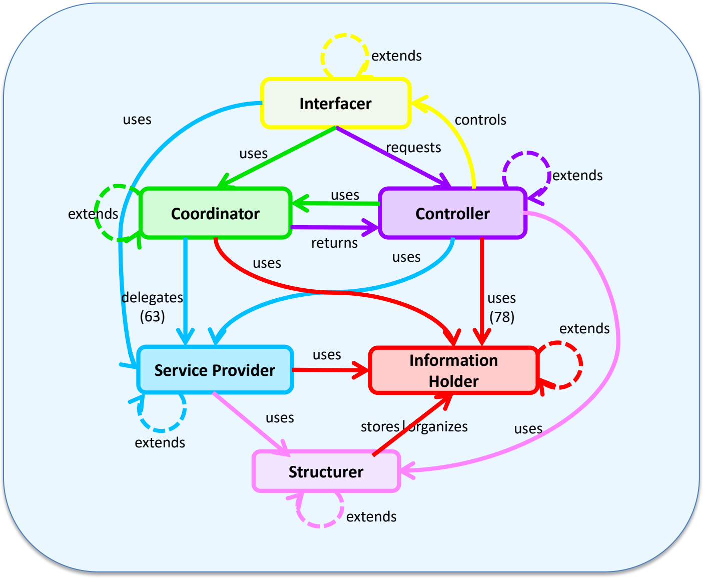
In this column, I introduce several characteristics I ascribe to classes when trying to understand their nature and purpose. I hope you find these useful quiddities and not mere quibbles. (I'm both delighted by and distrustful of a word that has definitions with opposite meanings. The technical term for such a beast is autoantonym.)

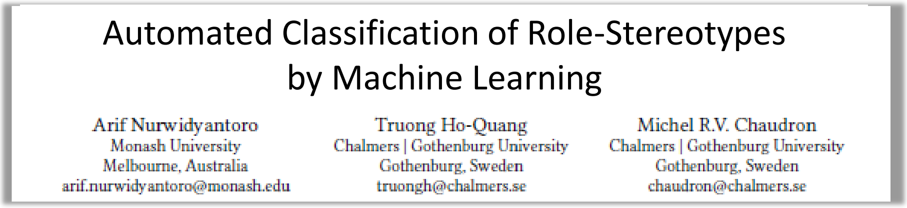
Role stereotypes

Purposeful oversimplifications, or *role stereotypes* from Responsibility-Driven Design

- *Information holder*: an object designed to know certain information and provide that information to other objects.
- *Structurer*: an object that maintains relationships between objects and information about those relationships. Complex structurers might pool, collect, and maintain groups of many objects; simpler structurers maintain relationships between a few objects. An example of a generic structurer is a Java HashMap, which relates keys to values.
- *Service provider*: an object that performs specific work and offers services to others on demand.
- *Controller*: an object designed to make decisions and control a complex task.
- *Coordinator*: an object that doesn't make many decisions but, in a rote or mechanical way, delegates work to other objects. The Mediator pattern is one example.
- *Interfacer*: an object that transforms information or requests between distinct parts of a system. The edges of an application contain user-interfacer objects that interact with the user and external interfacer objects, which communicate with external systems. Interfacers also exist between subsystems. The Facade pattern is an example of a class designed to simplify interactions and limit clients' visibility of objects within a subsystem.

Relationships between role stereotypes




Automated Classification of Role-Stereotypes by Machine Learning

Arif Nurwidyantoro
Monash University
Melbourne, Australia
arif.nurwidyantoro@monash.edu

Truong Ho-Quang
Chalmers | Gothenburg University
Gothenburg, Sweden
truongh@chalmers.se

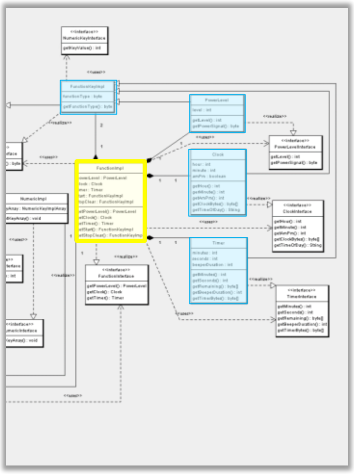
Michel R.V. Chaudron
Chalmers | Gothenburg University
Gothenburg, Sweden
chaudron@chalmers.se

Accepted for publication in EASE2019

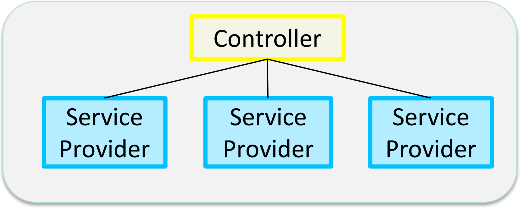
- Build a ground truth for 3 open source projects
  - K9, BitCoinWallet, Home3D
- Features used:
  - source code (e.g. names), design metrics, network-metrics

CHALMERS UNIVERSITY OF TECHNOLOGY | UNIVERSITY OF GOTHENBURG

Mining graph-patterns in Software Designs



Through labelling of roles, we find recurring patterns in the design



Controller

Service Provider

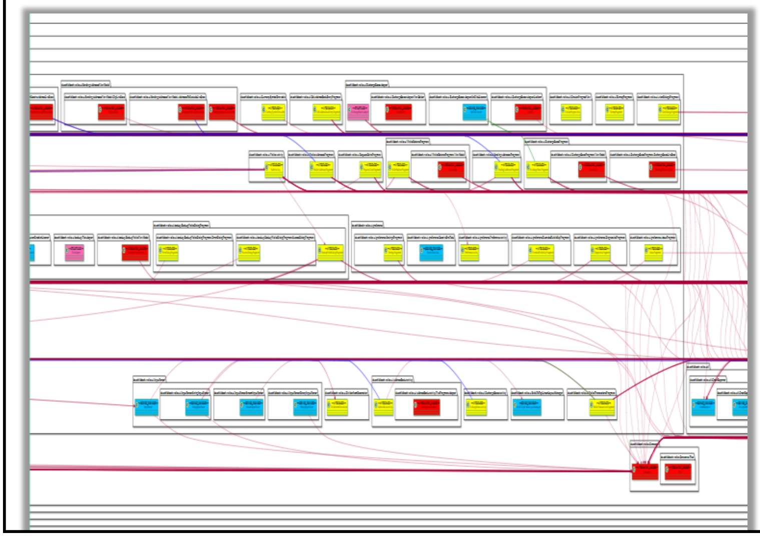
Service Provider

Service Provider

These patterns represent typical collaborations between responsibility-stereotypes

CHALMERS UNIVERSITY OF TECHNOLOGY | UNIVERSITY OF GOTHENBURG

Role-stereotypes in software design

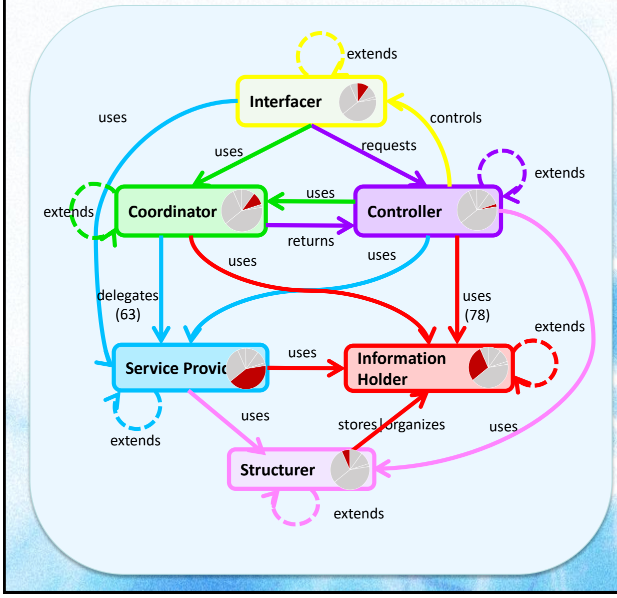


- Interfacer
- Controller
- Coordinator
- Service Provider
- Information Holder
- Structurer

CHALMERS UNIVERSITY OF TECHNOLOGY | UNIVERSITY OF GOTHENBURG

MRV Chaudron Hamburg 2019

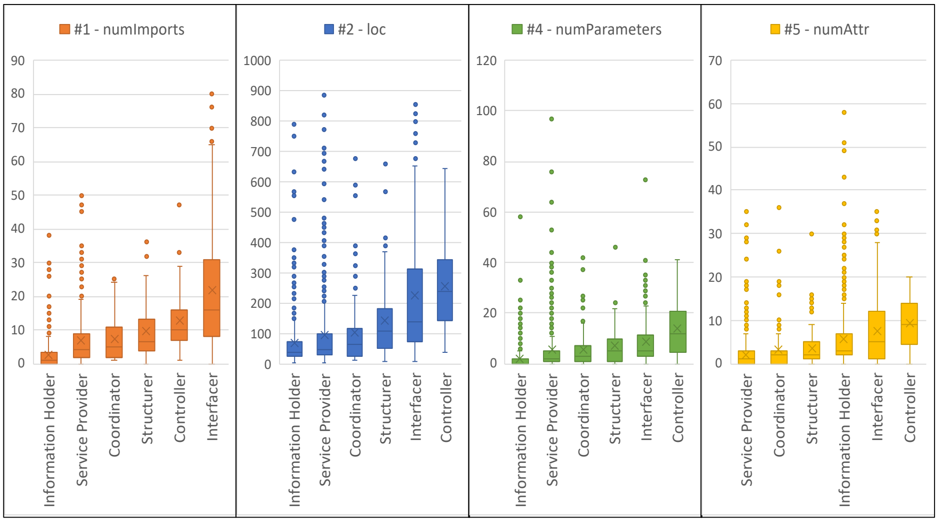
Distribution of Role-Stereotypes



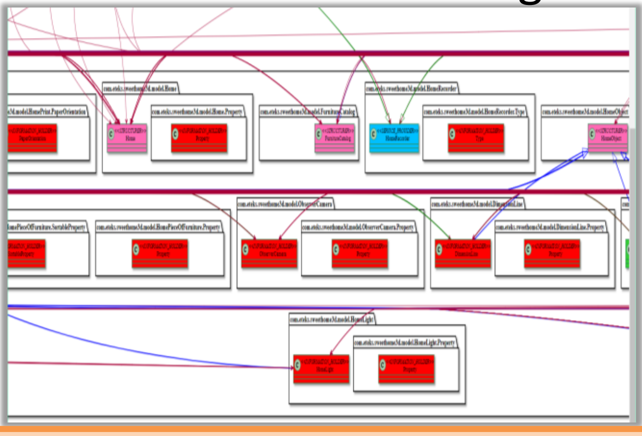
Role	Freq.	%
INTF	77	10
COORD	79	10
CNTRL	20	2
SRVC	323	42
IHLDR	231	30
STRCTR	49	6
	779	

CHALMERS UNIVERSITY OF TECHNOLOGY | UNIVERSITY OF GOTHENBURG

Role-Stereotypes Differ Significantly in their Design Metrics



Role-stereotypes-based visualization of software design



Interfacer

Controller

Coordinator

Service Provider

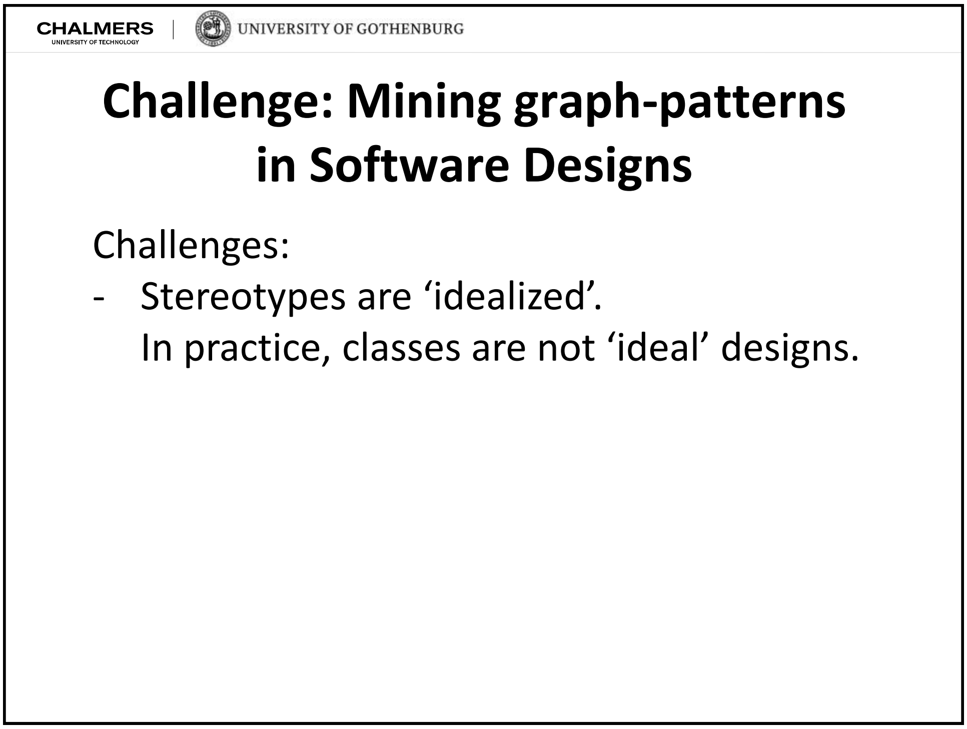
Information Holder

Structurer

Ratio of different role-stereotypes within a component gives information about its main type of responsibility

Possible uses of Role-Stereotypes

Role-Stereotypes aid in:
- Program understanding
- Design summarization
- QA for Architecture/Design
- Tailoring test-generation/coverage/…



CHALMERS UNIVERSITY OF TECHNOLOGY | UNIVERSITY OF GOTHENBURG

Challenge: Mining graph-patterns in Software Designs

Challenges:

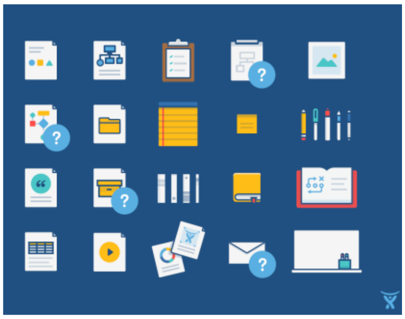
- Stereotypes are ‘idealized’.
  In practice, classes are not ‘ideal’ designs.

CHALMERS UNIVERSITY OF TECHNOLOGY | UNIVERSITY OF GOTHENBURG

MRV Chaudron Hamburg 2019

Knowledge Management in Software Development & Maintenance

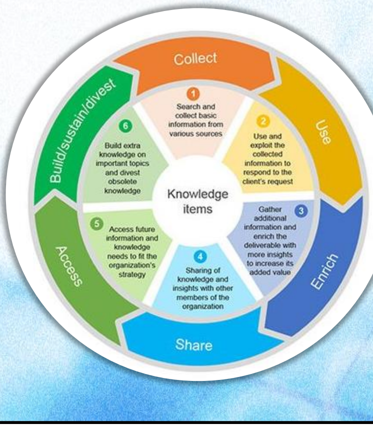
Documentation Challenge: Find & Navigate



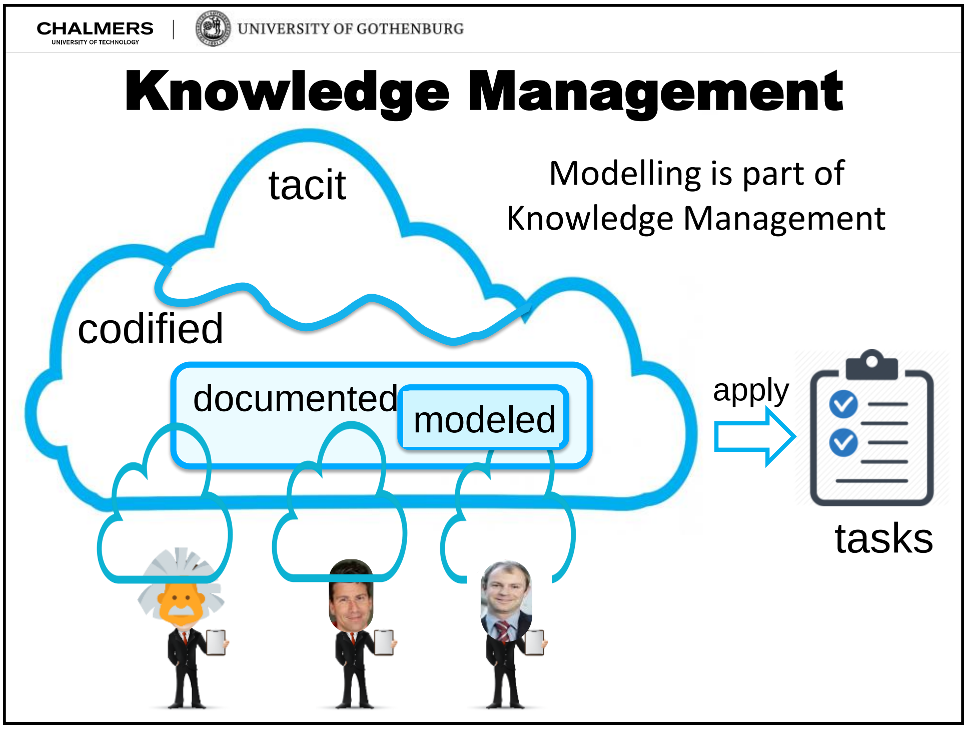
- Where is the information that I need?
- Many ‘places’ *GitHub, Project-PC, …*
- Poor searchability
- Poor navigability
- Is it up-to-date?

Knowledge Management in Software Development & Maintenance

Reduces load on creating & maintaining documentation


Collect
1 Search and collect basic information from various sources
Use
2 Use and exploit the collected information to respond to the client's request
Enrich
3 Gather additional information and enrich the deliverable with more insights to increase its added value
Share
4 Sharing of knowledge and insights with other members of the organization
Access
5 Access future information and knowledge needs to fit the organization's strategy
Build/sustain/divest
6 Build extra knowledge on important topics and divest obsolete knowledge
Knowledge items

- Extraction
  - NLP, image processing
- Linking
  - ontologies
- Up-dating
- Navigate / Present



Knowledge Management

Modelling is part of Knowledge Management

tacit

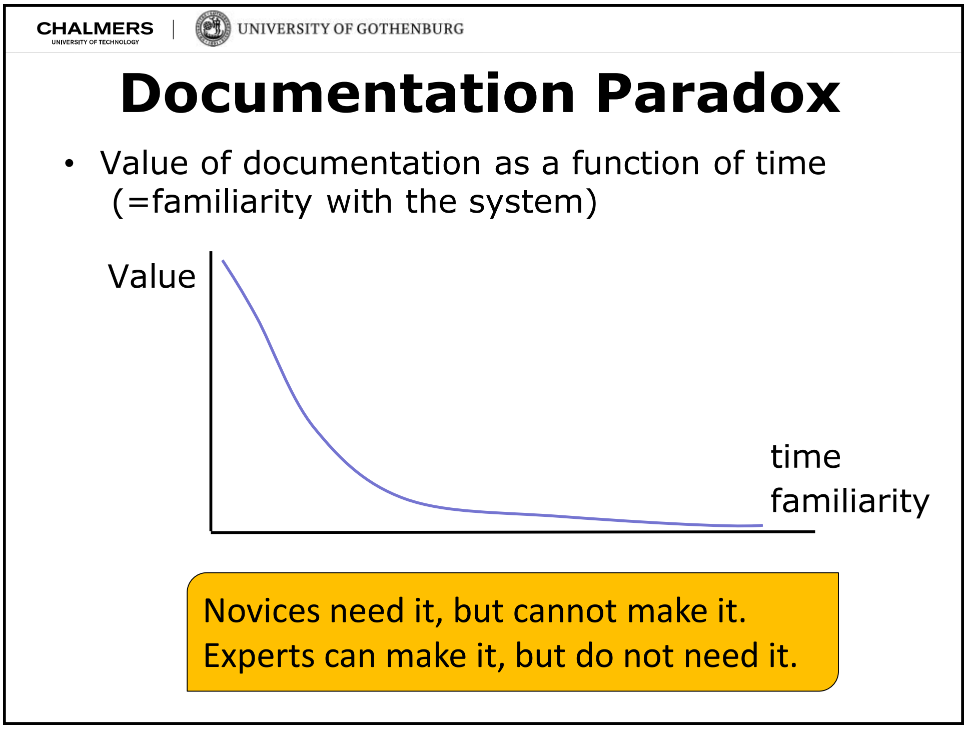
codified

documented

modeled

apply

tasks



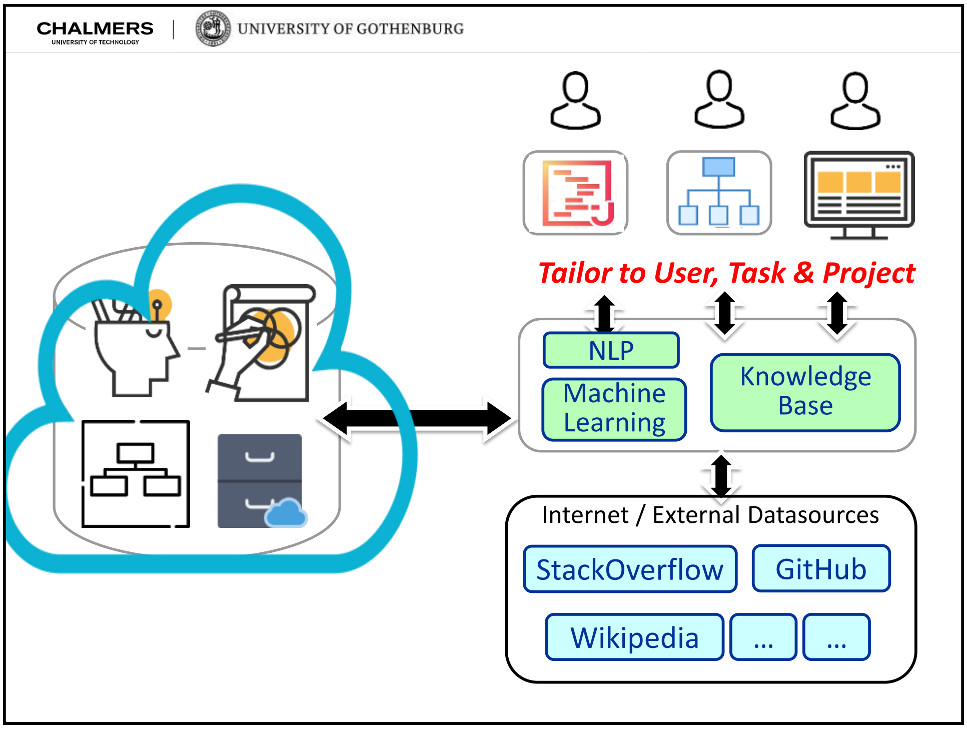
Documentation Paradox

- Value of documentation as a function of time (=familiarity with the system)

Value

time
familiarity

Novices need it, but cannot make it.
Experts can make it, but do not need it.



CHALMERS UNIVERSITY OF TECHNOLOGY | UNIVERSITY OF GOTHENBURG

Tailor to User, Task & Project

NLP

Machine Learning

Knowledge Base

Internet / External Datasources

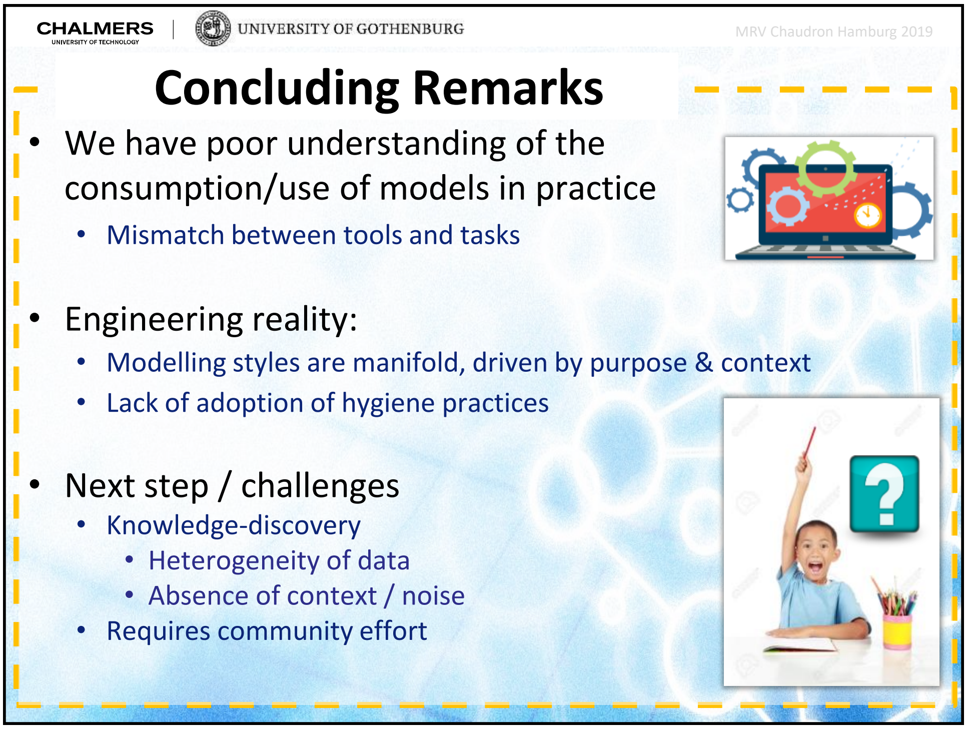
StackOverflow

GitHub

Wikipedia

…

…

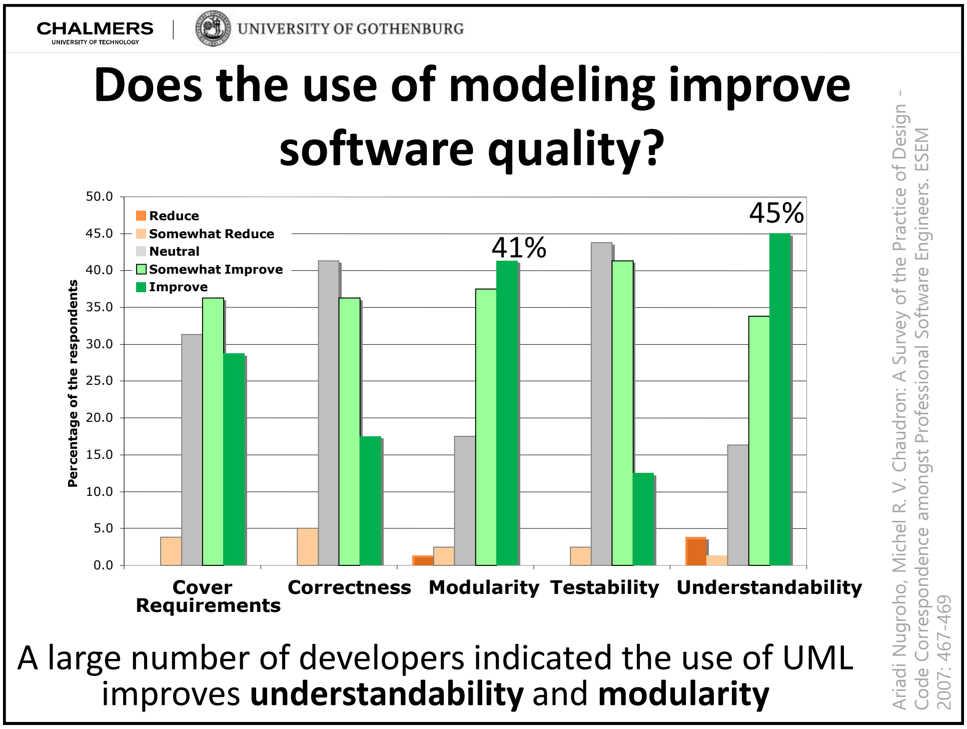


CHALMERS UNIVERSITY OF TECHNOLOGY | UNIVERSITY OF GOTHENBURG

MRV Chaudron Hamburg 2019

Concluding Remarks

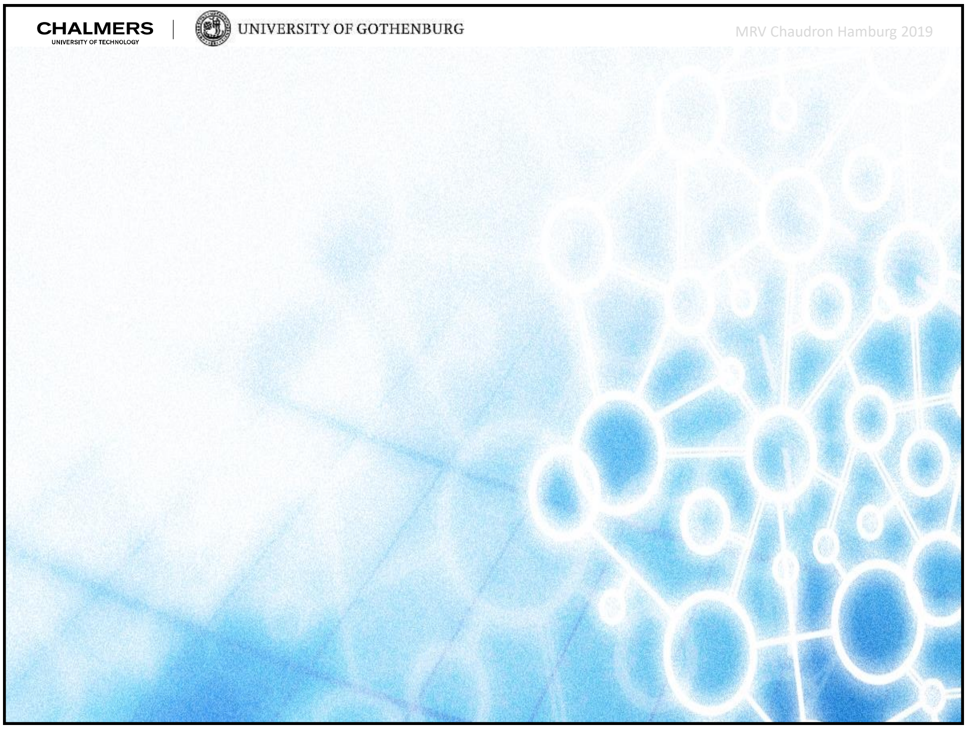
- We have poor understanding of the consumption/use of models in practice
  - Mismatch between tools and tasks
- Engineering reality:
  - Modelling styles are manifold, driven by purpose & context
  - Lack of adoption of hygiene practices
- Next step / challenges
  - Knowledge-discovery
    - Heterogeneity of data
    - Absence of context / noise
  - Requires community effort



Does the use of modeling improve software quality?

A large number of developers indicated the use of UML improves **understandability** and **modularity**

Ariadi Nugroho, Michel R. V. Chaudron: A Survey of the Practice of Design - Code Correspondence amongst Professional Software Engineers. ESEM 2007: 467-469



Summary of Evidence on Modeling

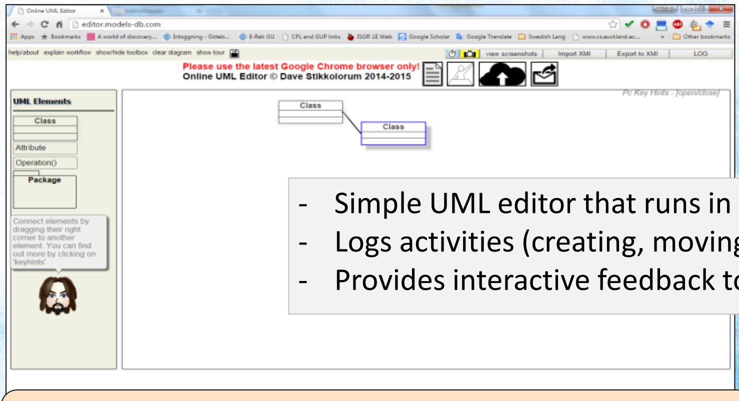
Benefit	Status
Communication	Confirmed
Analysis	No evidence
Structuring	No evidence
Guiding	Confirmed

Main hurdles:

- migrating existing documentation
- Integrating modeling tools in toolchain
- Integrating modeling into process
- Keeping models & documentation up to date

Project : Towards an online learning environment for software design

with Dave Stikkolorum



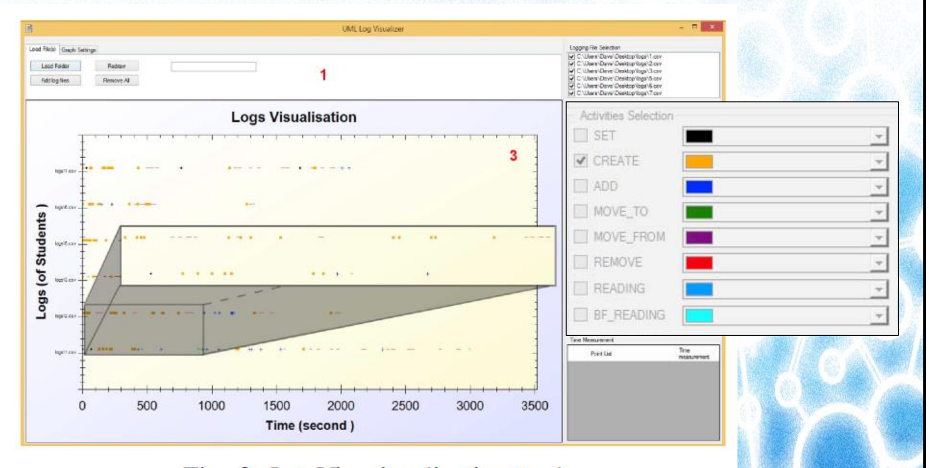
Please use the latest Google Chrome browser only!
Online UML Editor © Dave Stikkolorum 2014-2015

Connect elements by dragging their right corner to another element. You can find out more by clicking on 'keyhints'

- Simple UML editor that runs in a browser
- Logs activities (creating, moving, renaming)
- Provides interactive feedback to students

Goal is to integrate ‘doing design’ in online learning courses (UML/Analysis & Design/Softw architecture)

Visualization of log of modeling actvities



Logging File Selection

- ☑ C:\Users\Dave\Desktop\logs\1.csv
- ☑ C:\Users\Dave\Desktop\logs\2.csv
- ☑ C:\Users\Dave\Desktop\logs\3.csv
- ☑ C:\Users\Dave\Desktop\logs\5.csv
- ☑ C:\Users\Dave\Desktop\logs\6.csv
- ☑ C:\Users\Dave\Desktop\logs\7.csv

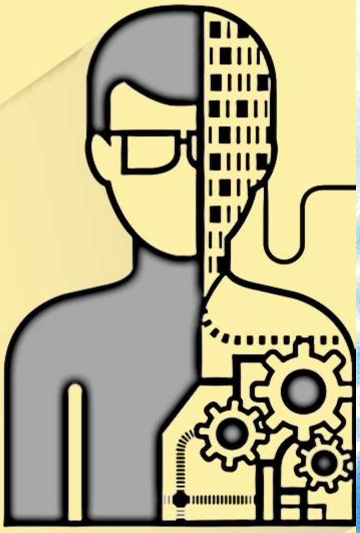
Activities Selection

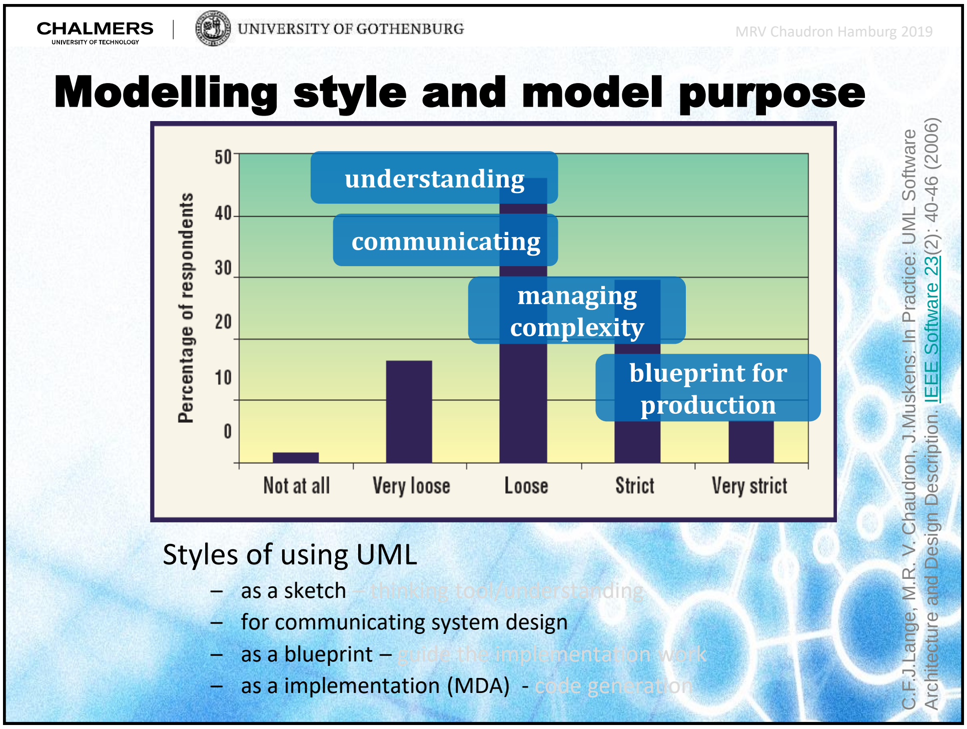
- □ SET ⬛
- ☑ CREATE 🟧
- □ ADD 🟦
- □ MOVE_TO 🟩
- □ MOVE_FROM 🟪
- □ REMOVE 🟥
- □ READING 🟦
- □ BF_READING 🟦

Fig. 2: LogViz visualisation tool

Summary:

AI will augment Human Intelligence in Software Engineering



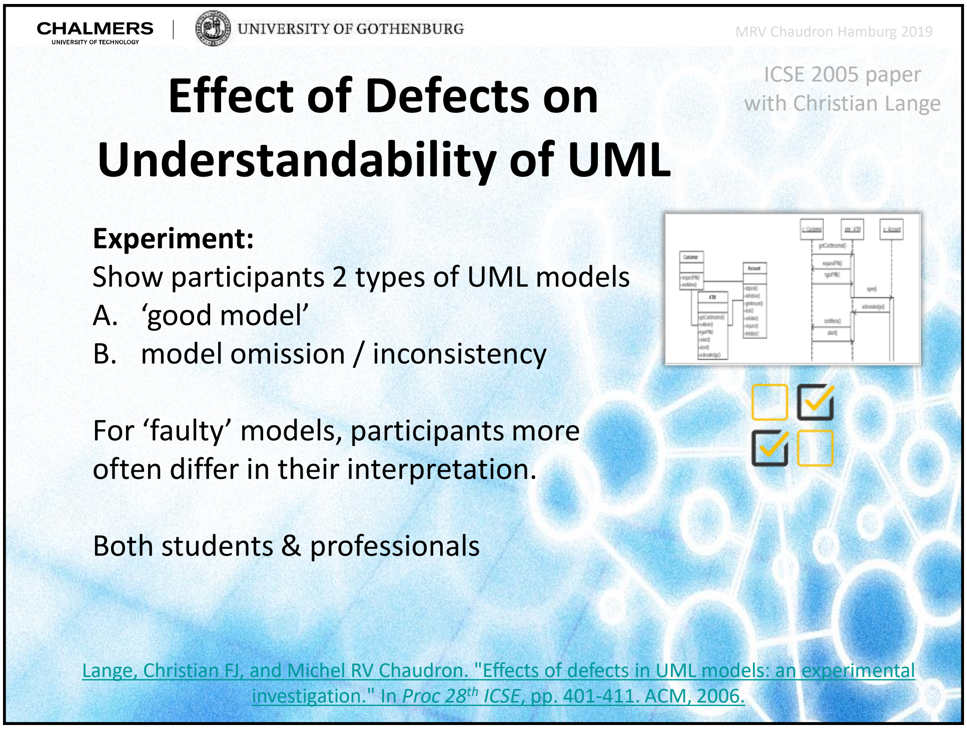


Modelling style and model purpose

Styles of using UML

- as a sketch – thinking tool/understanding
- for communicating system design
- as a blueprint – guide the implementation work
- as a implementation (MDA) - code generation

C.F.J.Lange, M.R. V. Chaudron, J.Muskens: In Practice: UML Software Architecture and Design Description. IEEE Software 23(2): 40-46 (2006)



CHALMERS
UNIVERSITY OF TECHNOLOGY
UNIVERSITY OF GOTHENBURG
MRV Chaudron Hamburg 2019

Effect of Defects on Understandability of UML

ICSE 2005 paper
with Christian Lange

Experiment:
Show participants 2 types of UML models

A. ‘good model’
B. model omission / inconsistency

For ‘faulty’ models, participants more often differ in their interpretation.

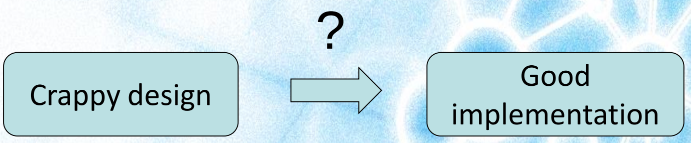
Both students & professionals

☐ ☑
☑ ☐

Lange, Christian FJ, and Michel RV Chaudron. "Effects of defects in UML models: an experimental investigation." In *Proc 28th ICSE*, pp. 401-411. ACM, 2006.

Challenge 1

Can we automatically assess the quality of a software design?



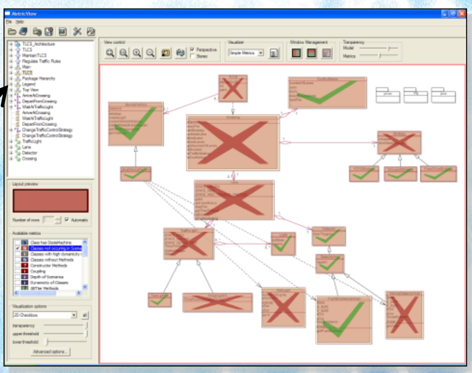
Project: Quality Assessment of Software Design - the MetricView tool

When is a design ‘good’?
Absence of ‘bad things’?

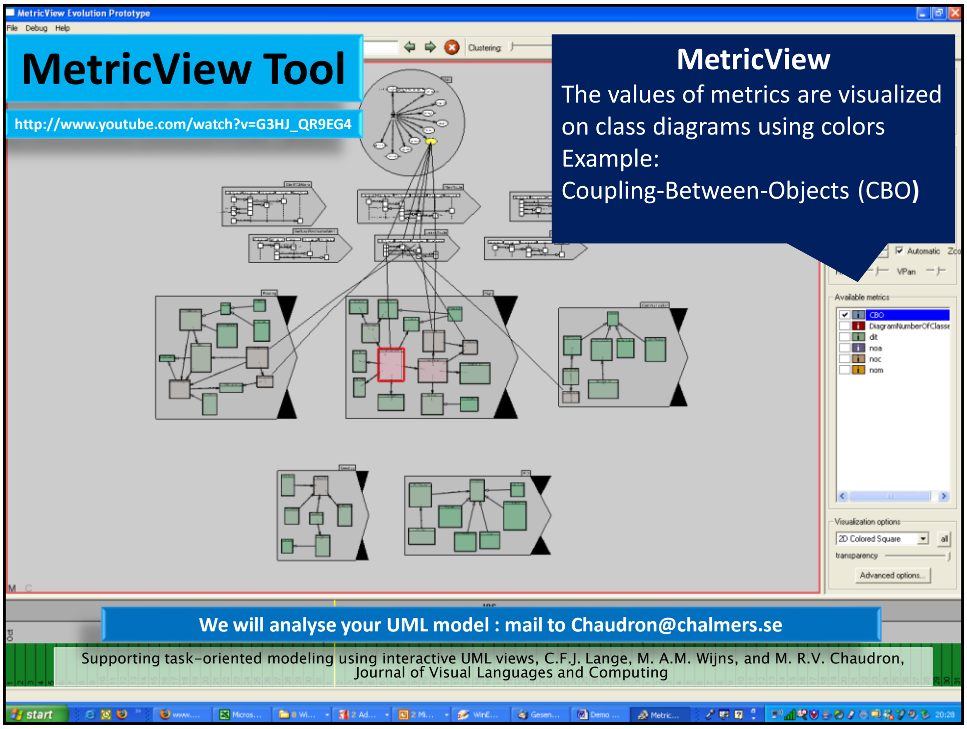
with Christian Lange
& Johan Muskens

Quality Metrics capture design principles

- Coupling
- Cohesion
- … <extensible>



✗ = violation ✓ = ok



MetricView Tool

http://www.youtube.com/watch?v=G3HJ_QR9EG4

MetricView
The values of metrics are visualized on class diagrams using colors
Example:
Coupling-Between-Objects (CBO)

We will analyse your UML model : mail to Chaudron@chalmers.se

Supporting task-oriented modeling using interactive UML views, C.F.J. Lange, M. A.M. Wijns, and M. R.V. Chaudron, Journal of Visual Languages and Computing

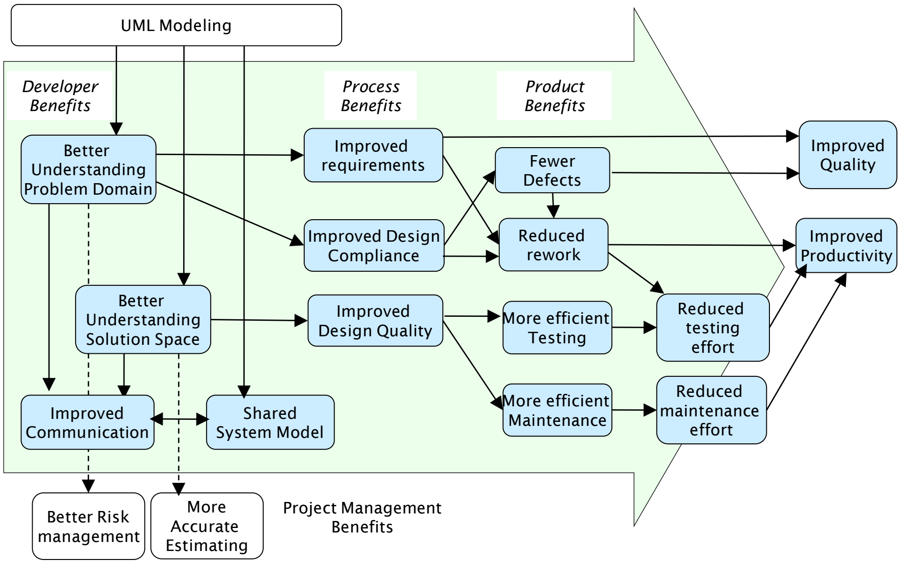
Industrial Experience with MetricView as Quality Assurance Tool

- Based on 15+ industrial projects
- If there are weak spots in the design, then these are indicated as ‘suspects’ by MetricView
- About 90% of the weak spots indicated by MetricView do not require improvement according to project architects

Syntactic checks (like metrics) are not sufficient to identify important areas in the design!

- The later MetricView is applied, the fewer ‘weak spots’ are removed from the design → process issue

Theory of Benefits of Modeling



UML Modeling

Developer Benefits

Process Benefits

Product Benefits

Better Understanding Problem Domain

Improved requirements

Improved Quality

Fewer Defects

Improved Design Compliance

Reduced rework

Improved Productivity

Better Understanding Solution Space

Improved Design Quality

More efficient Testing

Reduced testing effort

Improved Communication

Shared System Model

More efficient Maintenance

Reduced maintenance effort

Better Risk management

More Accurate Estimating

Project Management Benefits

Integrate Modeling into Process and Tooling

Low Hanging Fruit
(c) Erin Witzel, 2010

Naming and layout-conventions

Version Management

Many tools around (e.g. CVS, SVN, …)

Reviews & Inspections

Guidelines by e.g. Shull et.al., Biffl,

Easy QA-tool: SDMetrics http://www.sdmetrics.com/

Process

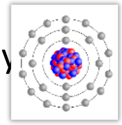
Integrate into process:

Who? When? How? Update a model in documentation

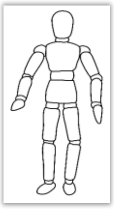
‘Definition of done’

Definitions

Model (noun) = an abstract representation
of a thing/sy



often *systematic* representation



Models abstract: they focus on the
essential features and leave out others.

Modeling = the process of creating a model
(verb) i.e. choosing what to represent
how to represent it

Design (verb, noun)

Definition

Design (v) = the process of making decisions
about something
that is to be built
or created**:**

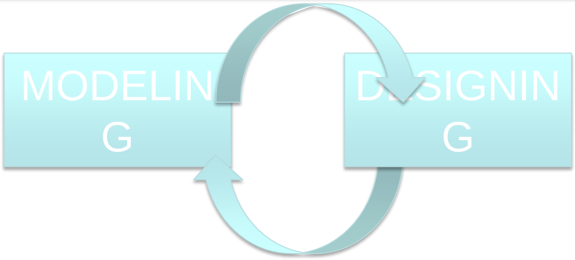
Design (n) = the plans, drawings, etc., that
show how
something can be
made

Pitfall : ‘design’ & ‘model’ can be a verb and a noun

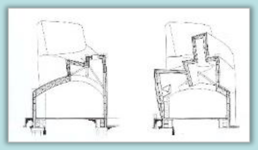
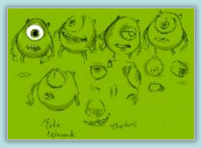
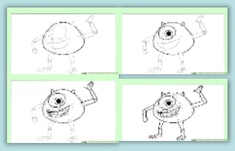
How do ‘modeling’ and ‘design’ relate?

Modeling and designing often go hand-in-hand:

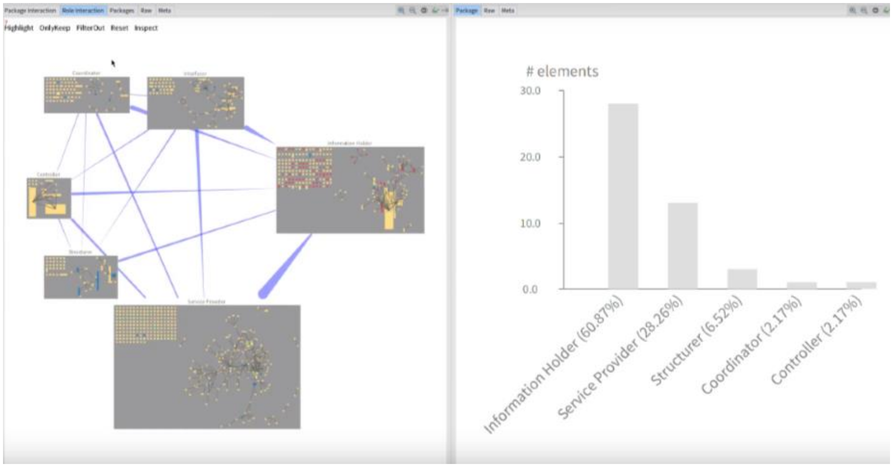
A model is used to understand, reason, analyse, break-down, which leads to adaptation and refinement of the design.



Stages of Design & Modeling

	Ideation	Externalization	Production
Building			
Animation			
	Develop idea	Explain to others	Blueprint for Production

Challenge: Visualisation

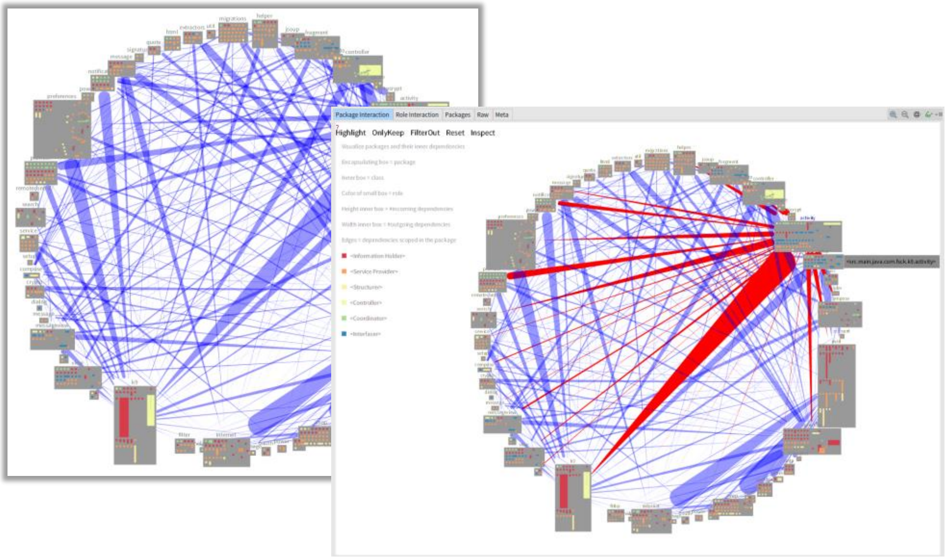


Visualisation Challenge

- Scale
- Multiple uses
- Multiple abstraction levels
  - Models & Source Code

CHALMERS UNIVERSITY OF TECHNOLOGY | UNIVERSITY OF GOTHENBURG

How to layout packages of classes?



Visualize packages and their inner dependencies

Encapsulating box = package

Inner box = class

Color of small box = role

Height inner box = #incoming dependencies

Width inner box = #outgoing dependencies

Edges = dependencies scoped in the package

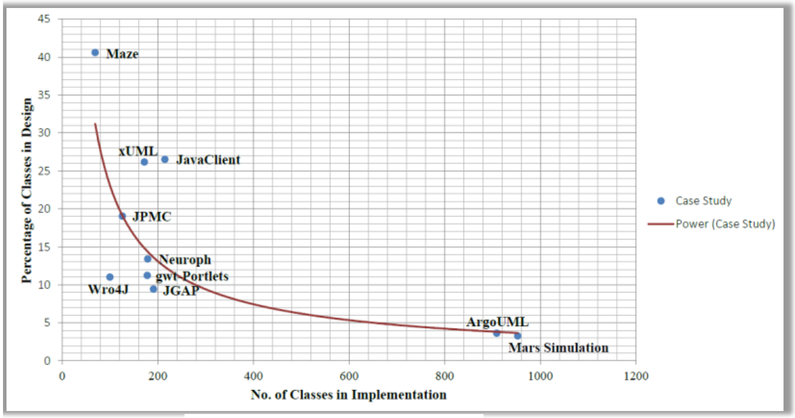
- 🟥 <Information Holder>
- 🟧 <Service Provider>
- 🟨 <Structurer>
- 🟨 <Controller>
- 🟩 <Coordinator>
- 🟦 <Interfacer>

CHALMERS UNIVERSITY OF TECHNOLOGY | UNIVERSITY OF GOTHENBURG | Universidad Rey Juan Carlos

MODELS 2016

Economy of Modeling

% of system covered by model



Size of the system

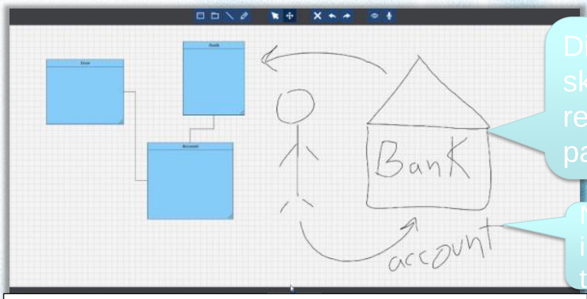
Integration of Tools in Design Flow and Development Process

Are the tools the problem?

Project : OctoUML

with Rodi Jolak

- Merge **whiteboard** (informal, free-form) and **CASE-tool** (formal) modeling



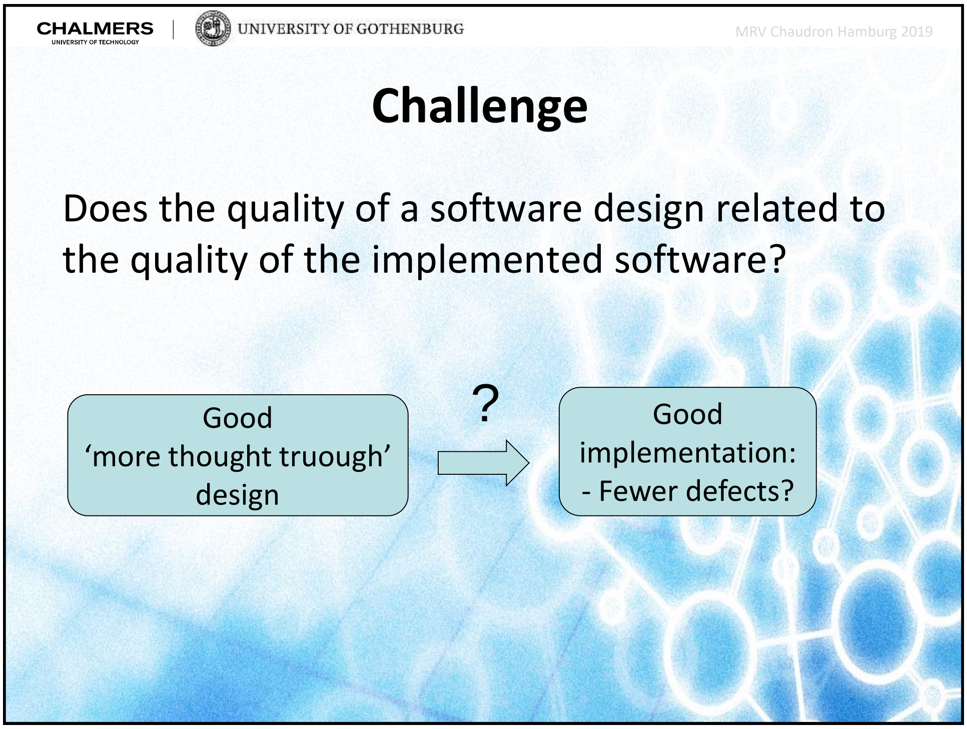
Digital editing for sketching: undo, resize, move, pan, zoom

Novel UI interaction models: touch, voice

https://www.youtube.com/watch?v=fsN3rfEAYHw&t=74s

Goal: Tools should support the ***development process*** (not only the modeling task)

of Gothenburg – OctoUML



CHALMERS UNIVERSITY OF TECHNOLOGY | UNIVERSITY OF GOTHENBURG

MRV Chaudron Hamburg 2019

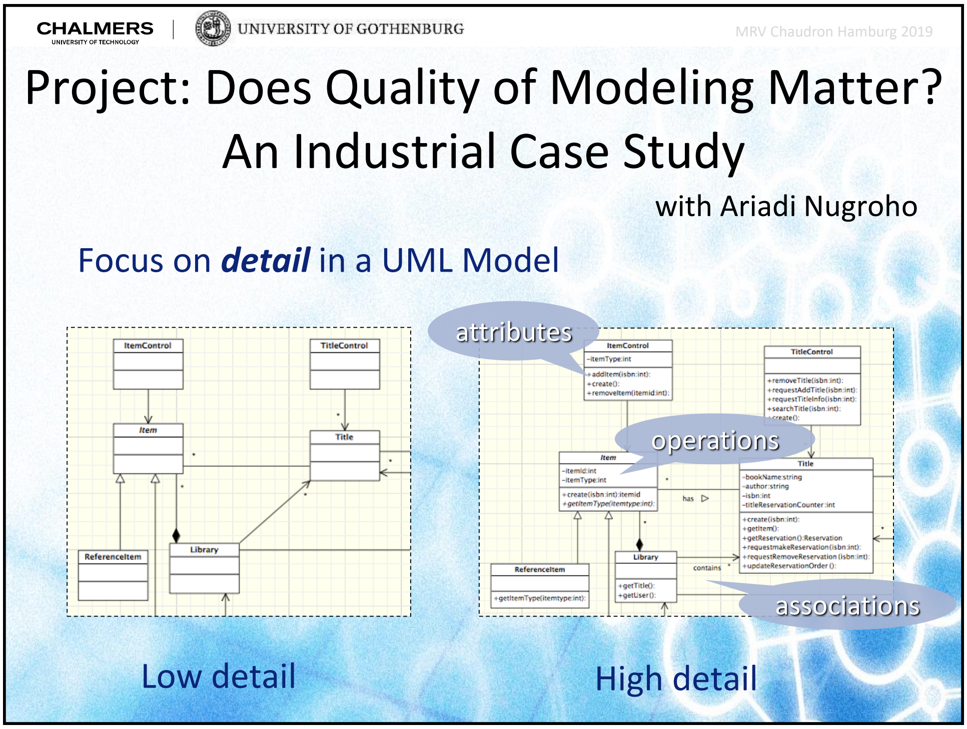
Challenge

Does the quality of a software design related to the quality of the implemented software?

Good ‘more thought truough’ design

?

Good implementation: - Fewer defects?



CHALMERS UNIVERSITY OF TECHNOLOGY | UNIVERSITY OF GOTHENBURG

MRV Chaudron Hamburg 2019

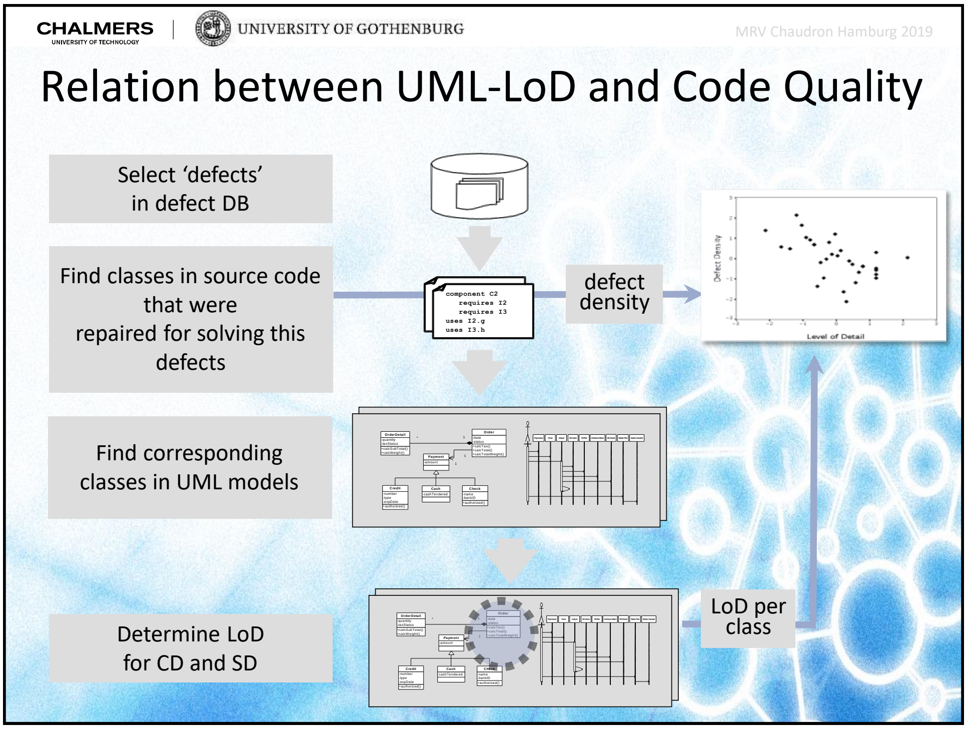
Project: Does Quality of Modeling Matter? An Industrial Case Study

with Ariadi Nugroho

Focus on ***detail*** in a UML Model

Low detail

High detail



CHALMERS UNIVERSITY OF TECHNOLOGY | UNIVERSITY OF GOTHENBURG

MRV Chaudron Hamburg 2019

Relation between UML-LoD and Code Quality

Select ‘defects’ in defect DB

Find classes in source code that were repaired for solving this defects

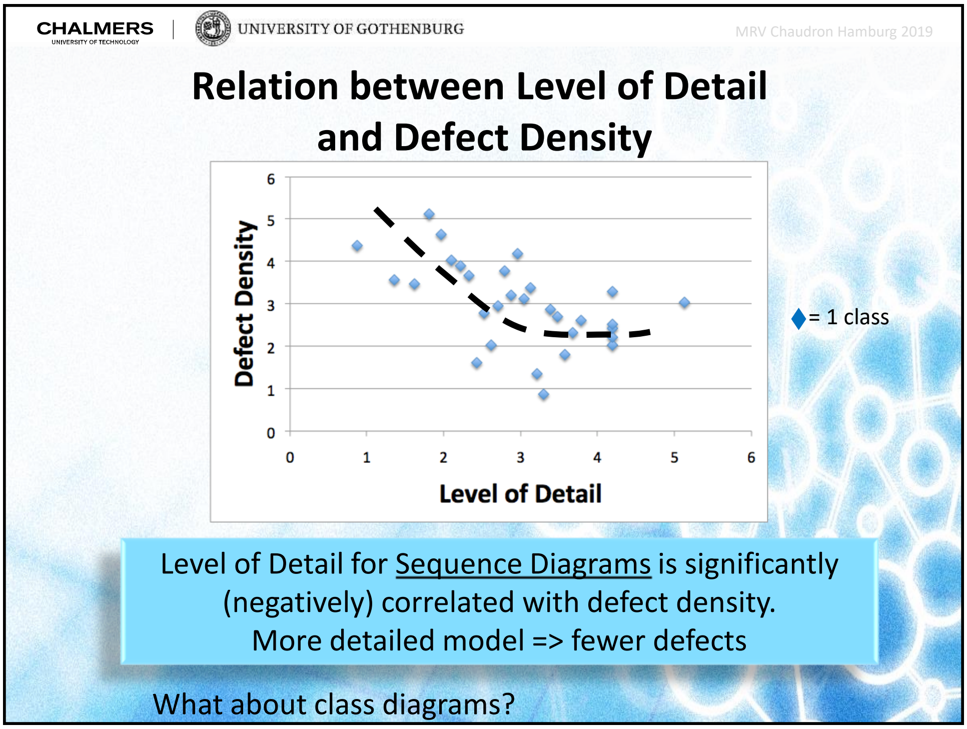
Find corresponding classes in UML models

Determine LoD for CD and SD

```
component C2
   requires I2
   requires I3
uses I2.g
uses I3.h
```

defect density

LoD per class



CHALMERS UNIVERSITY OF TECHNOLOGY | UNIVERSITY OF GOTHENBURG

MRV Chaudron Hamburg 2019

Relation between Level of Detail and Defect Density

Level of Detail for Sequence Diagrams is significantly (negatively) correlated with defect density.
More detailed model => fewer defects

What about class diagrams?