UNIVERSITÉ DE NANTES

Report from Case Studies in Model-Driven Reverse Engineering

Modelsward 2019

Pascal André

LS2N lab, University of Nantes, France

Outline of the talk

- Context and Problem statement
- MDRE Case studies
- Discussion
- Conclusion

! This is not a systematic study
Share issues/principles raised from my own experience

Outline of the talk

- Context and Problem statement

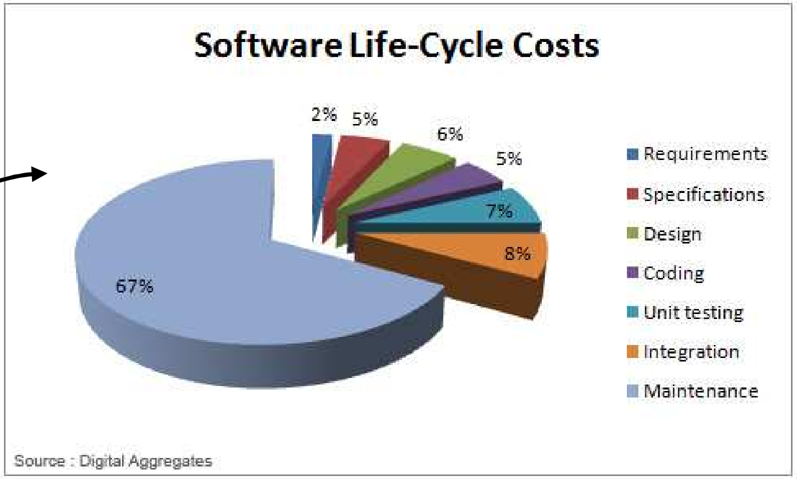
  MDRE for software maintenance
- MDRE Case studies
- Discussion
- Conclusion

Context and Problem Statement

software maintenance

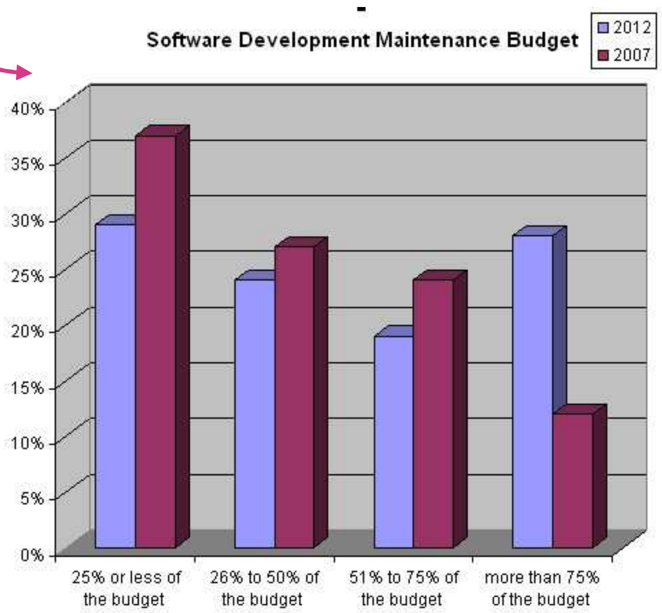
key values findings

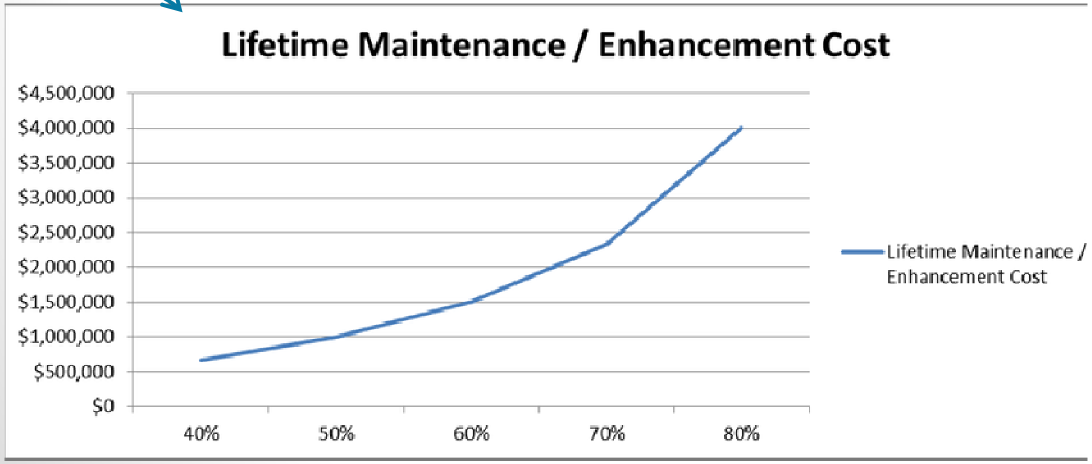
- ~ 70% of the software cost
- Increases with time
- Increases with complexity



http://thibautvs.com/blog/?tag=maintenance

old legacy applications have higher technical debt





https://dotnetsilverlightprism.wordpress.com/2012/02/19/the-relationship-between-software-structure-and-the-softwares-value-to-a-business/

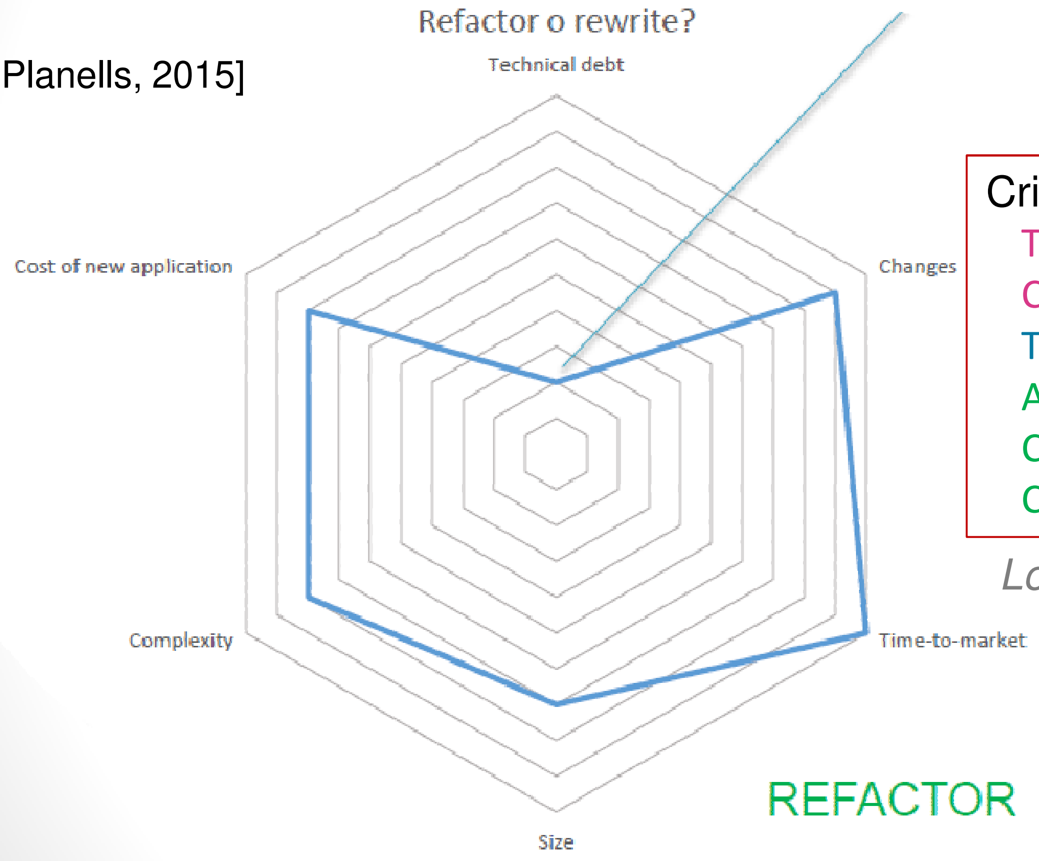
http://blog.martinig.ch/numbers/increased-software-development-maintenance-costs/

Context and Problem Statement

Software maintenance

Refactor or Rewrite??

[Planells, 2015]



REFACTOR

Criteria
- Time to market
- Continuous changes
- Technical debt
- Application size
- Complexity
- Cost of (new) application redesign

Lost Design decision traceability

https://www.slideshare.net/JorgeCapelPlanells/presentation-refactoring-large-legacy-applications

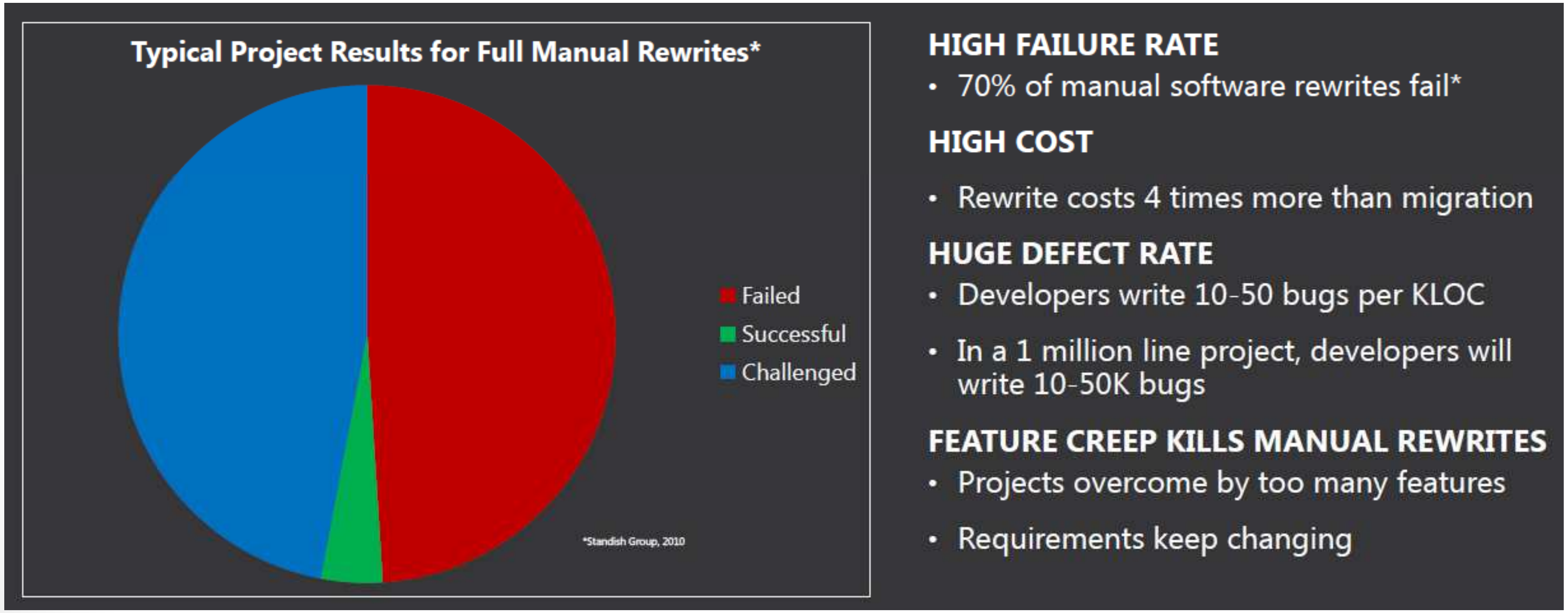
☞ Need abstractions (models)

Context and Problem Statement

Software maintenance

Automated or manual ??

The problem with manual rewrite…



HIGH FAILURE RATE

- 70% of manual software rewrites fail*

HIGH COST

- Rewrite costs 4 times more than migration

HUGE DEFECT RATE

- Developers write 10-50 bugs per KLOC
- In a 1 million line project, developers will write 10-50K bugs

FEATURE CREEP KILLS MANUAL REWRITES

- Projects overcome by too many features
- Requirements keep changing

https://www.slideshare.net/ddskier/calculating-the-cost-of-manual-rewrites

☞ Need automation (Model driven)

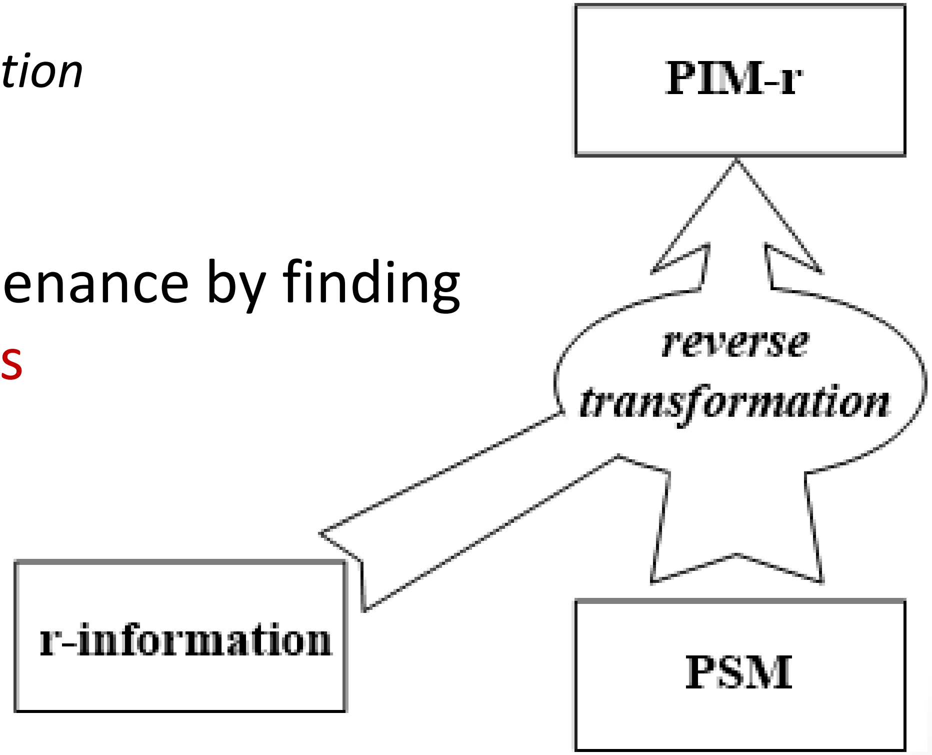
Motivations

In summary

Concepts are more resilient than implementations

- Model Driven
  enforce automation
- Reverse Engineering
  raise the level of abstraction

MDRE – assisting maintenance by finding the missing abstractions



Outline

- MDRE for software maintenance
- MDRE Case studies
- Discussion
- Conclusion

MDRE for software maintenance

3 situations

- Extract information of lost or deprecated design documentation.
- Understand an existing software solution with missing documents.
- Align business processes with legacy applications. ← 1
- Improve genericity by replacing hard coded information by configuration files.
- Extract software components to put on the shelf. ← 2
- Upgrade technical framework releases or updating technical components.
- Re-factor application to improve the quality or follow new coding standards. ← 3
- Modify the presentation layer or the persistence layer in n-tier web applications.
- Change the programming languages (e.g. from Cobol to Java).

MDRE for software maintenance

3 case studies

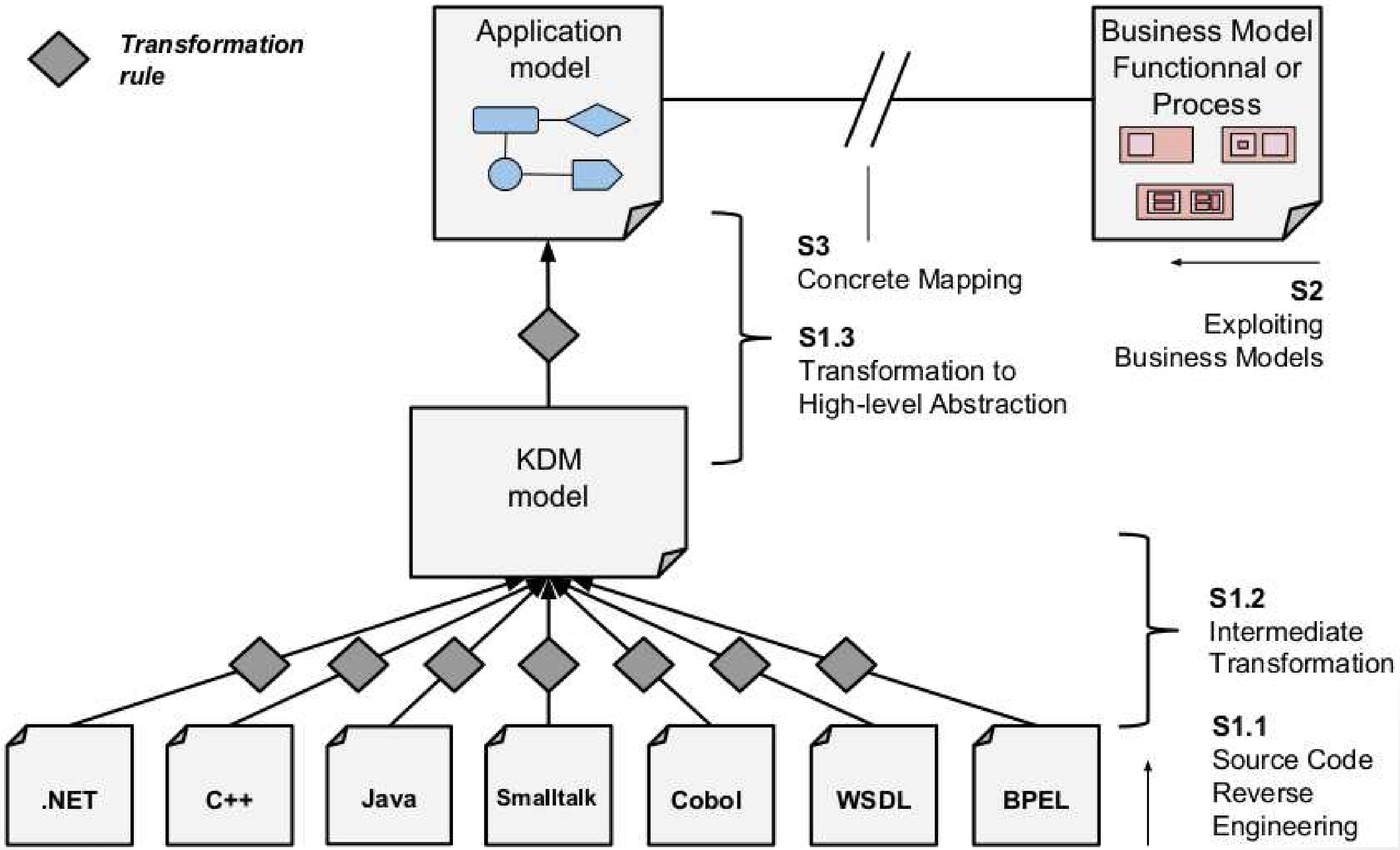
	1	2	3
	Information system	Component	Manufacturing
Domain	Information system	Software Architecture	Manufacturing
Maintenance operation	Application cartography	Verification, refactoring	refactoring / rewrite
	BITA alignment	Model extraction	MDE initialization
Context	Industrial	Research	Applied research
Models	KDE, EMF, App…	Sofa, Fractal, Kmelia…	UML like
MDRE	build the IT architecture view in order to compare to business models	Component extraction (structure, behaviour) for model verification	Third party discovering to refactor (code and process)
Tool / process	General purpose	Dedicated	General purpose
	Tool chain application	Tool box, iterative process	Two-way process (engineering/reverse)
Case studies	Insurance real case	CoCoME benchmark	SOFAL product line

Overview, few details

MDRE Case study 1

Align business processes with legacy applications.

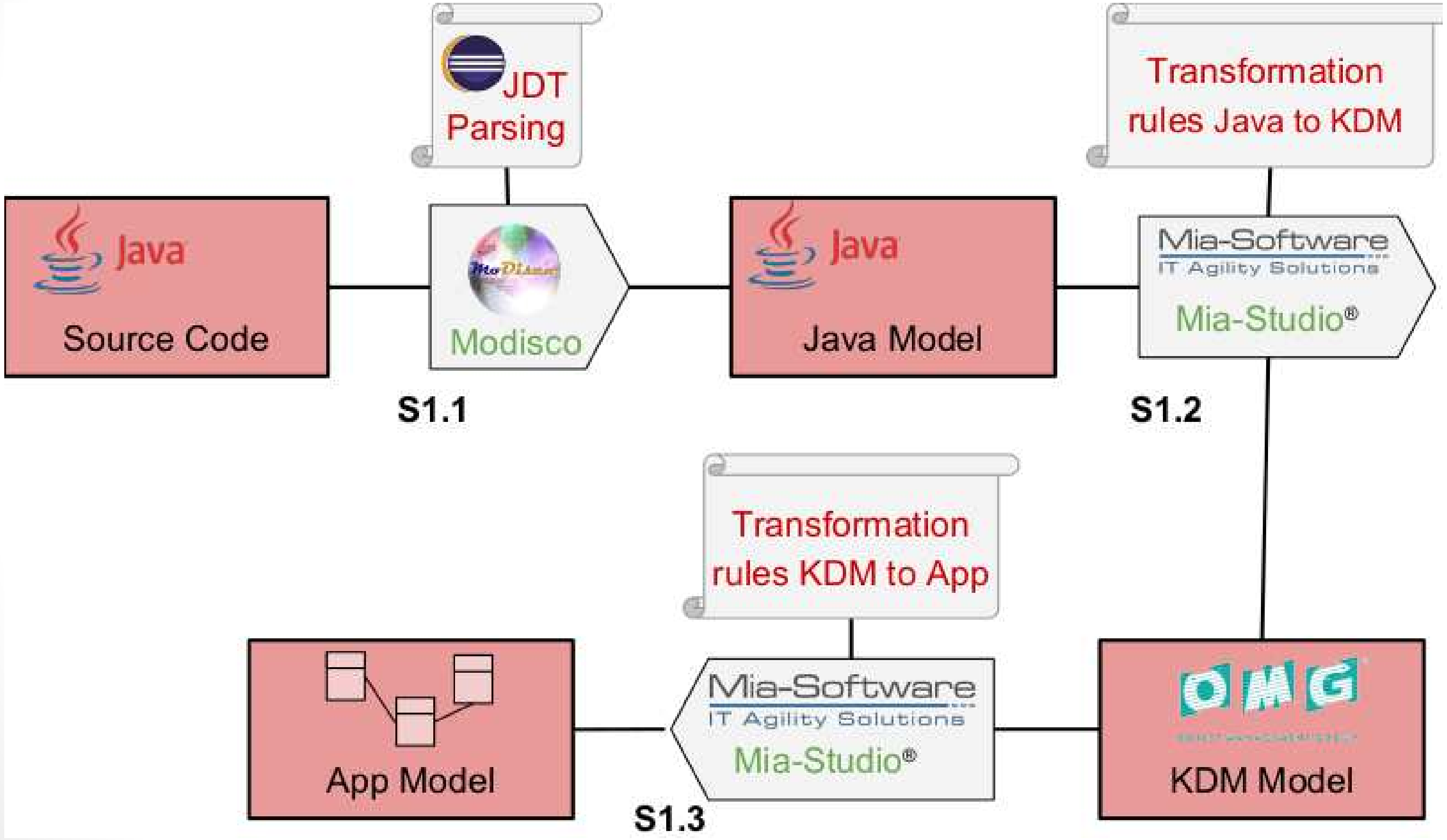
- Goal: reduce the Business-IT misalignment between the Information Technology (IT) and Business viewpoints
- MDRE: build the IT architecture view


Transformation rule
Application model
Business Model Functionnal or Process
S3
Concrete Mapping
S2
Exploiting Business Models
S1.3
Transformation to High-level Abstraction
KDM model
S1.2
Intermediate Transformation
S1.1
Source Code Reverse Engineering
.NET
C++
Java
Smalltalk
Cobol
WSDL
BPEL

MDRE Case study 1

Align business processes with legacy applications.

Tool

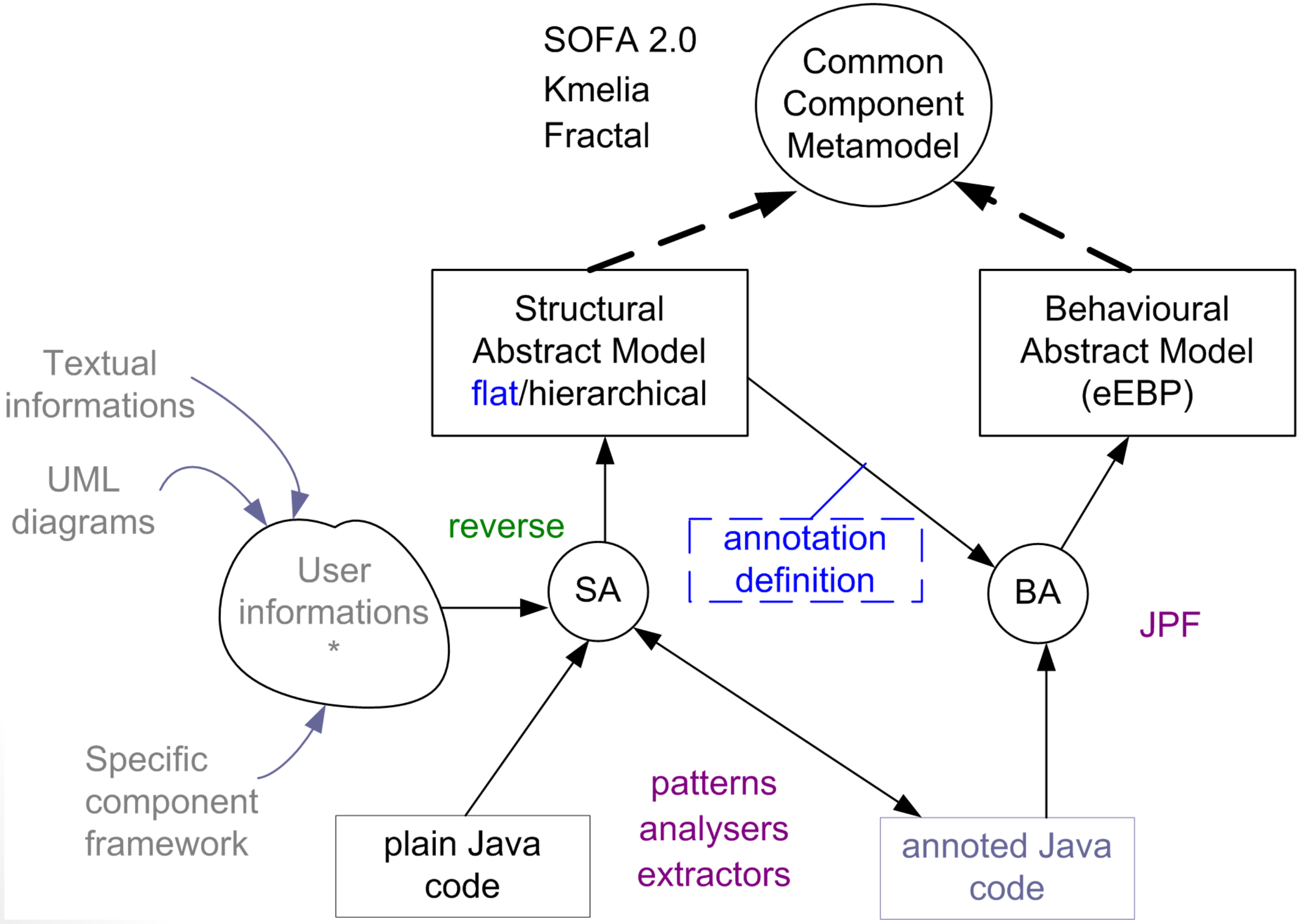


Reverse engineering process with general purpose tooling

MDRE Case study 2

Software Architecture Extraction

- Goal: reduce the architecture erosion, software verification
- MDRE: extract components (structure/behaviour)

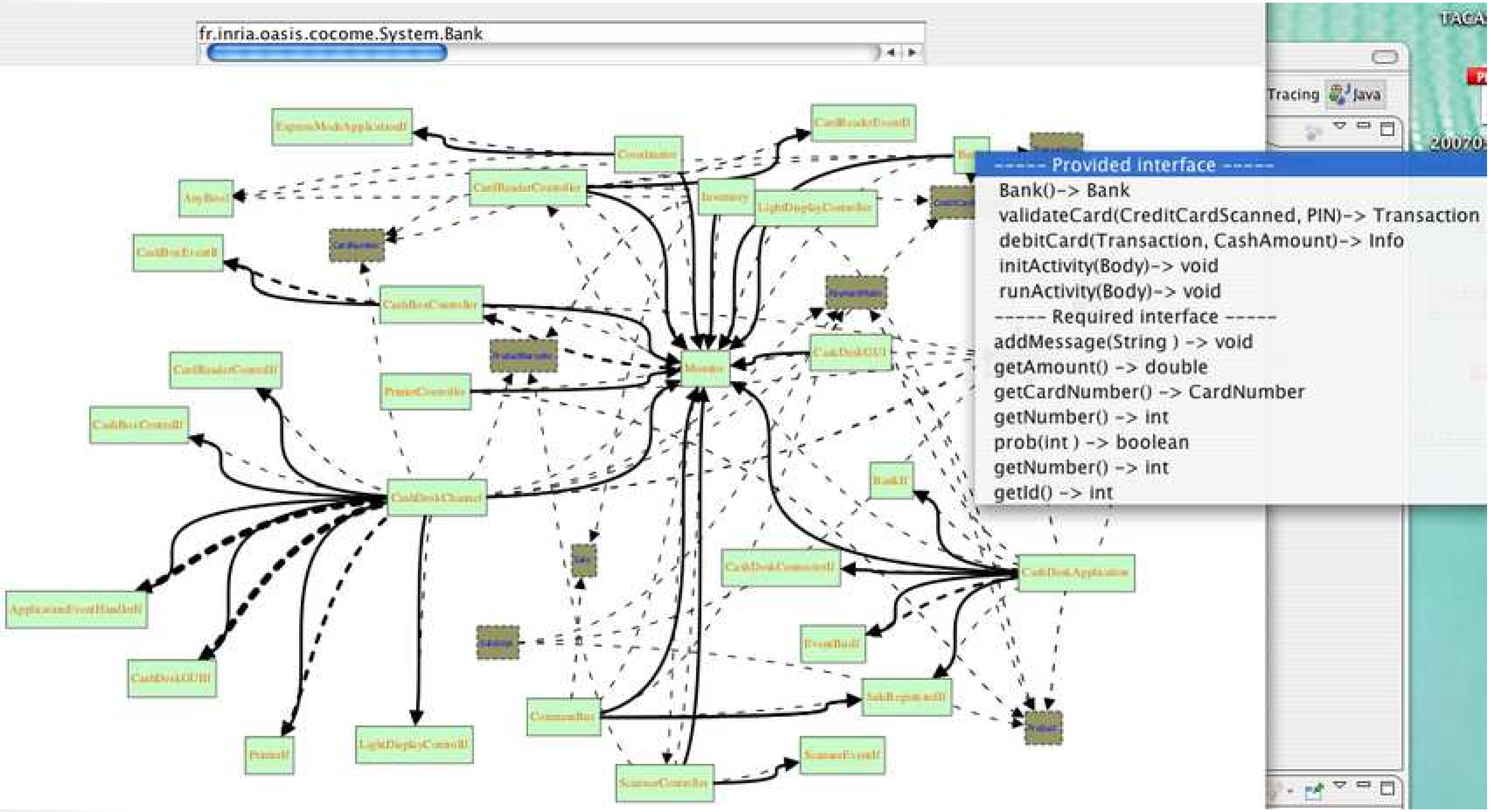


SOFA 2.0
Kmelia
Fractal
Common Component Metamodel
Structural Abstract Model flat/hierarchical
Behavioural Abstract Model (eEBP)
Textual informations
UML diagrams
User informations *
reverse
SA
annotation definition
BA
JPF
Specific component framework
plain Java code
patterns analysers extractors
annoted Java code

MDRE Case study 2

Software Architecture Extraction

Tool



```
----- Provided interface -----
Bank()-> Bank
validateCard(CreditCardScanned, PIN)-> Transaction
debitCard(Transaction, CashAmount)-> Info
initActivity(Body)-> void
runActivity(Body)-> void
----- Required interface -----
addMessage(String ) -> void
getAmount() -> double
getCardNumber() -> CardNumber
getNumber() -> int
prob(int ) -> boolean
getNumber() -> int
getId() -> int
```

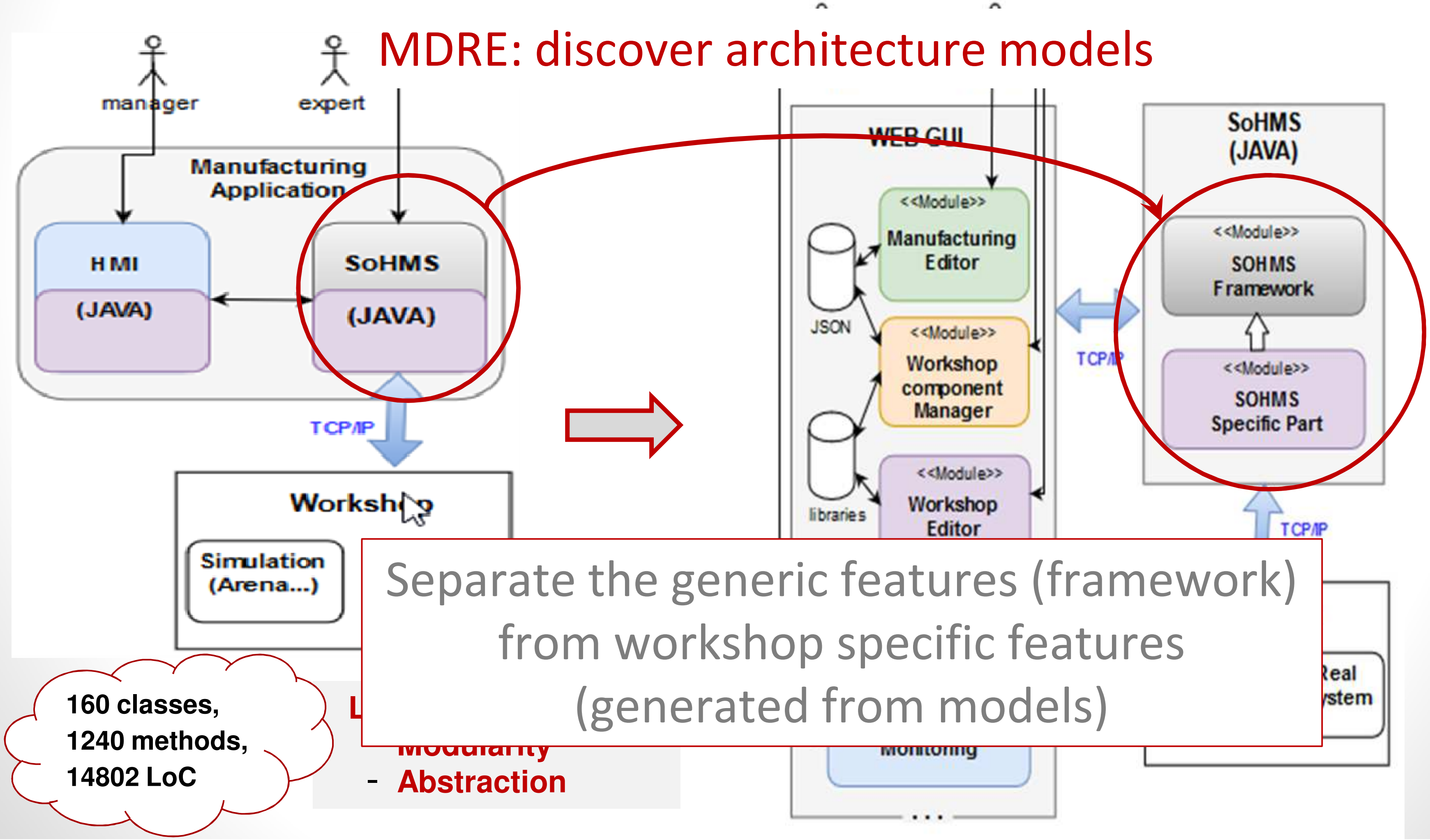
Javacompext : classify Java classes into Data types, components

Dependency / inheritance / communication

MDRE Case study 3

Re-engineering a Manufacturing application

- Goal: improve the software quality for better maintenance and verification, revisit the software process



MDRE: discover architecture models

Separate the generic features (framework) from workshop specific features (generated from models)

160 classes, 1240 methods, 14802 LoC

L

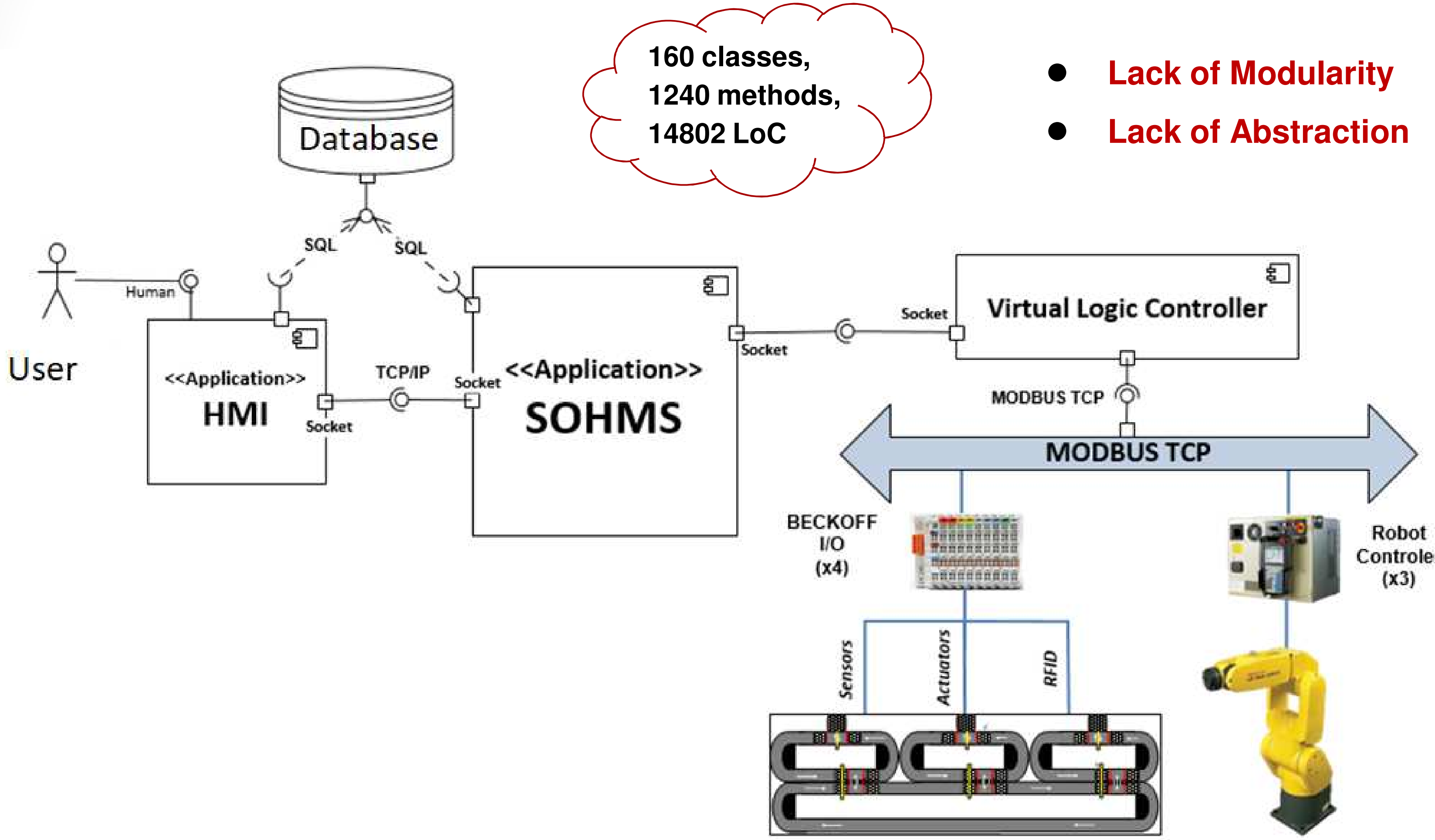
Modularity

- Abstraction

MDRE Case study 3

Re-engineering a Manufacturing application

- Current SOFAL application

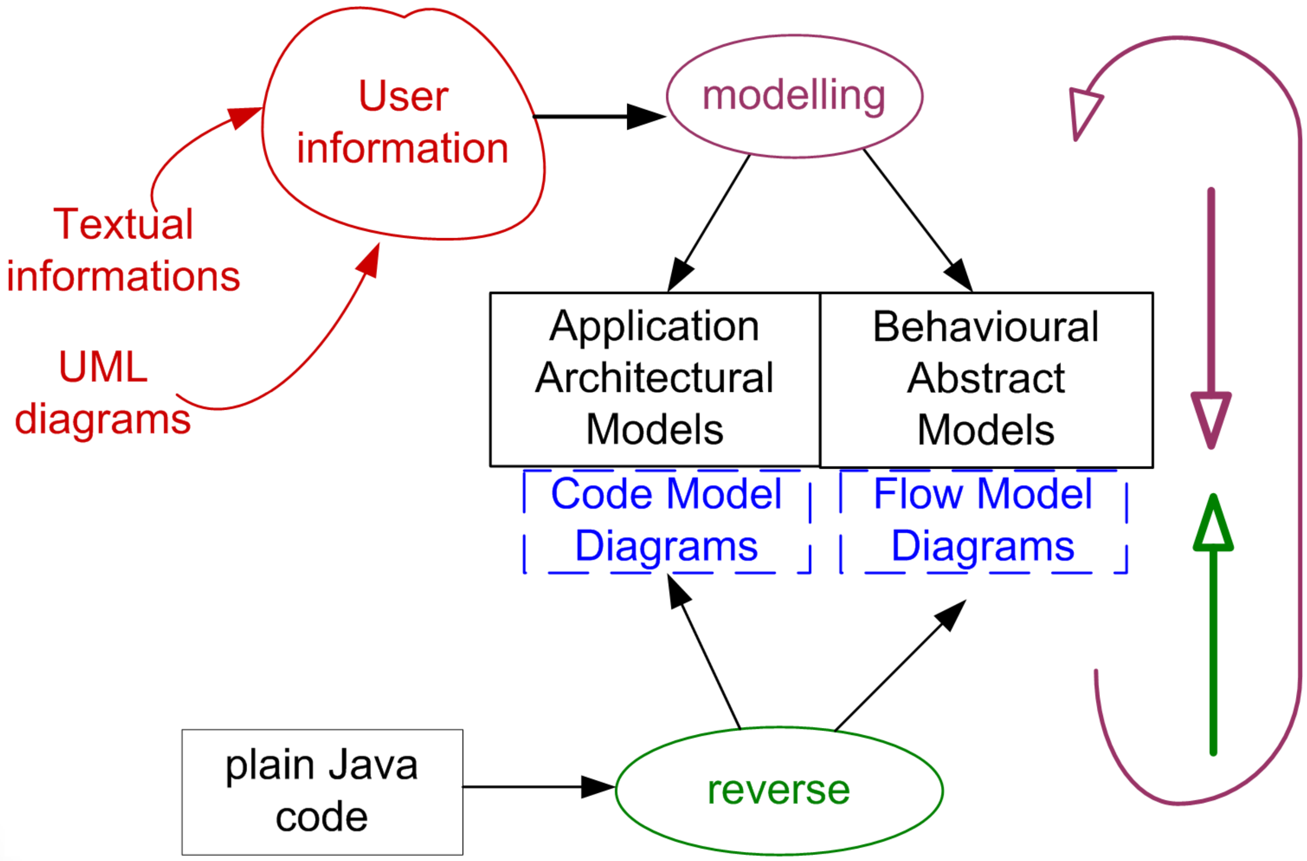


160 classes,
1240 methods,
14802 LoC

- **Lack of Modularity**
- **Lack of Abstraction**

MDRE Case study 3

Re-engineering a Manufacturing application



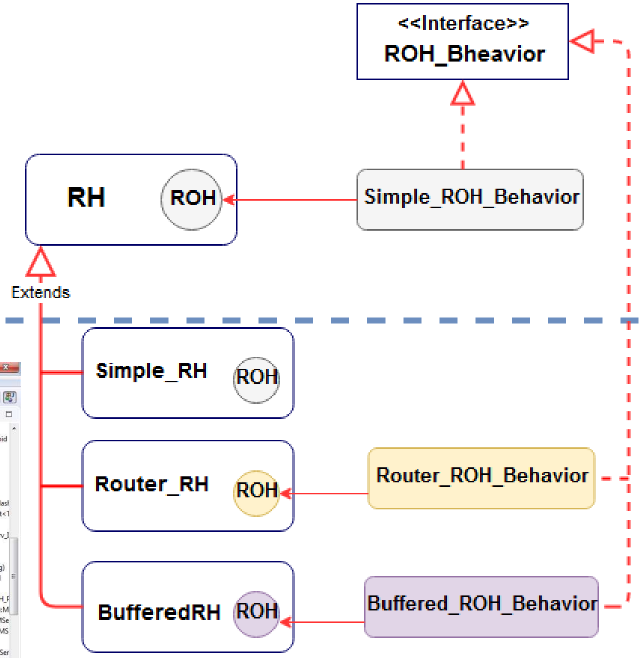
Two-way reverse engineering

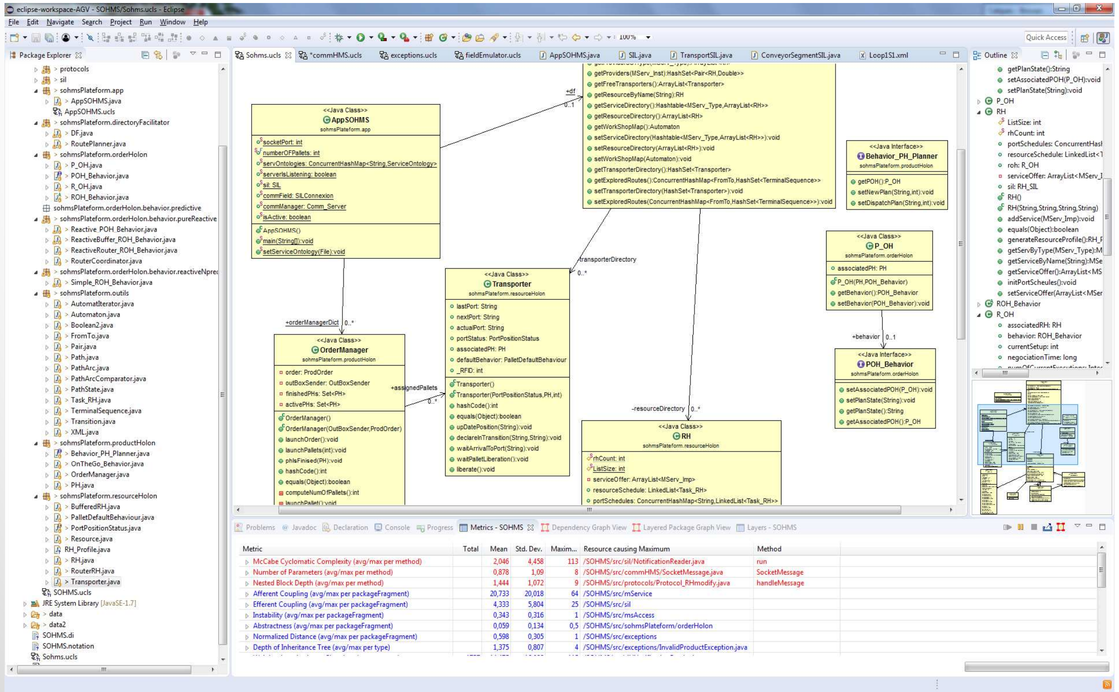
MDRE Case study 3

Re-engineering a Manufacturing application

MDRE: discover architecture models

- PHD thesis UML diagrams
- Java code (ObjectAid)





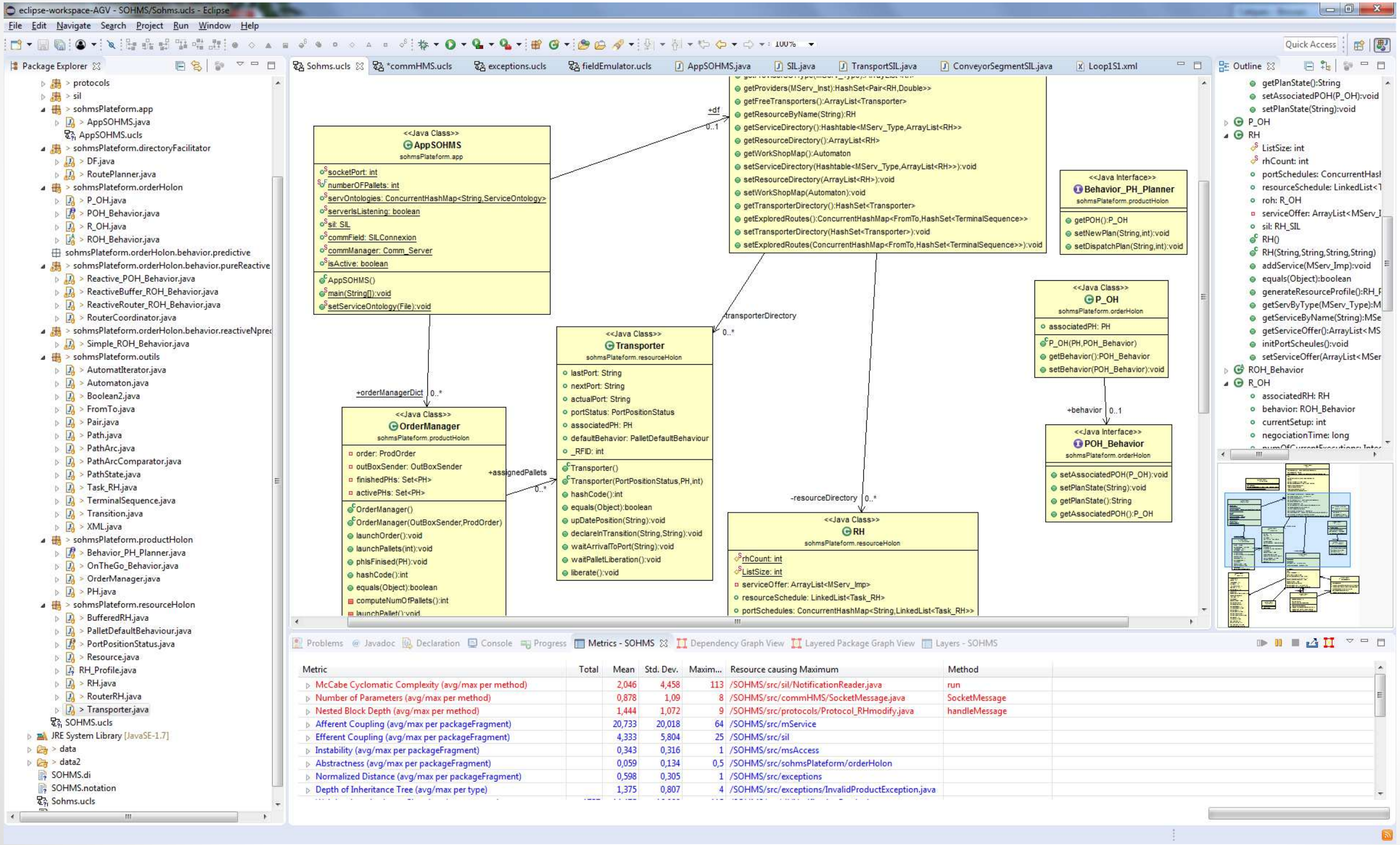
Metric	Total	Mean	Std. Dev.	Maxim...	Resource causing Maximum	Method
McCabe Cyclomatic Complexity (avg/max per method)		2,046	4,458	113	/SOHMS/src/sil/NotificationReader.java	run
Number of Parameters (avg/max per method)		0,878	1,09	8	/SOHMS/src/commHMS/SocketMessage.java	SocketMessage
Nested Block Depth (avg/max per method)		1,444	1,072	9	/SOHMS/src/protocols/Protocol_RHmodify.java	handleMessage
Afferent Coupling (avg/max per packageFragment)		20,733	20,018	64	/SOHMS/src/mService	
Efferent Coupling (avg/max per packageFragment)		4,333	5,804	25	/SOHMS/src/sil	
Instability (avg/max per packageFragment)		0,343	0,316	1	/SOHMS/src/msAccess	
Abstractness (avg/max per packageFragment)		0,059	0,134	0,5	/SOHMS/src/sohmsPlateform/orderHolon	
Normalized Distance (avg/max per packageFragment)		0,598	0,305	1	/SOHMS/src/exceptions	
Depth of Inheritance Tree (avg/max per type)		1,375	0,807	4	/SOHMS/src/exceptions/InvalidProductException.java	

Separate the generic/specific features

MDRE Case study 3

ObjectAid

Re-engineering a Manufacturing application



eclipse-workspace-AGV - SOHMS/Sohms.ucls - Eclipse
File Edit Navigate Search Project Run Window Help
Quick Access
Package Explorer
protocols
sil
sohmsPlateform.app
AppSOHMS.java
AppSOHMS.ucls
sohmsPlateform.directoryFacilitator
DF.java
RoutePlanner.java
sohmsPlateform.orderHolon
P_OH.java
POH_Behavior.java
R_OH.java
ROH_Behavior.java
sohmsPlateform.orderHolon.behavior.predictive
sohmsPlateform.orderHolon.behavior.pureReactive
Reactive_POH_Behavior.java
ReactiveBuffer_ROH_Behavior.java
ReactiveRouter_ROH_Behavior.java
RouterCoordinator.java
sohmsPlateform.orderHolon.behavior.reactiveNpred
Simple_ROH_Behavior.java
sohmsPlateform.outils
AutomatIterator.java
Automaton.java
Boolean2.java
FromTo.java
Pair.java
Path.java
PathArc.java
PathArcComparator.java
PathState.java
Task_RH.java
TerminalSequence.java
Transition.java
XML.java
sohmsPlateform.productHolon
Behavior_PH_Planner.java
OnTheGo_Behavior.java
OrderManager.java
PH.java
sohmsPlateform.resourceHolon
BufferedRH.java
PalletDefaultBehaviour.java
PortPositionStatus.java
Resource.java
RH_Profile.java
RH.java
RouterRH.java
Transporter.java
SOHMS.ucls
JRE System Library [JavaSE-1.7]
data
data2
SOHMS.di
SOHMS.notation
Sohms.ucls
Sohms.ucls
*commHMS.ucls
exceptions.ucls
fieldEmulator.ucls
AppSOHMS.java
SIL.java
TransportSIL.java
ConveyorSegmentSIL.java
Loop1S1.xml
<<Java Class>>
AppSOHMS
sohmsPlateform.app
socketPort: int
numberOFPallets: int
servOntologies: ConcurrentHashMap<String,ServiceOntology>
serverIsListening: boolean
sil: SIL
commField: SILConnexion
commManager: Comm_Server
isActive: boolean
AppSOHMS()
main(String[]):void
setServiceOntology(File):void
+df
0..1
getProviders(MServ_Inst):HashSet<Pair<RH,Double>>
getFreeTransporters():ArrayList<Transporter>
getResourceByName(String):RH
getServiceDirectory():Hashtable<MServ_Type,ArrayList<RH>>
getResourceDirectory():ArrayList<RH>
getWorkShopMap():Automaton
setServiceDirectory(Hashtable<MServ_Type,ArrayList<RH>>):void
setResourceDirectory(ArrayList<RH>):void
setWorkShopMap(Automaton):void
getTransporterDirectory():HashSet<Transporter>
getExploredRoutes():ConcurrentHashMap<FromTo,HashSet<TerminalSequence>>
setTransporterDirectory(HashSet<Transporter>):void
setExploredRoutes(ConcurrentHashMap<FromTo,HashSet<TerminalSequence>>):void
<<Java Interface>>
Behavior_PH_Planner
sohmsPlateform.productHolon
getPOH():P_OH
setNewPlan(String,int):void
setDispatchPlan(String,int):void
-transporterDirectory
0..*
+orderManagerDict
0..*
<<Java Class>>
Transporter
sohmsPlateform.resourceHolon
lastPort: String
nextPort: String
actualPort: String
portStatus: PortPositionStatus
associatedPH: PH
defaultBehavior: PalletDefaultBehaviour
_RFID: int
Transporter()
Transporter(PortPositionStatus,PH,int)
hashCode():int
equals(Object):boolean
updatePosition(String):void
declareInTransition(String,String):void
waitArrivalToPort(String):void
waitPalletLiberation():void
liberate():void
<<Java Class>>
P_OH
sohmsPlateform.orderHolon
associatedPH: PH
P_OH(PH,POH_Behavior)
getBehavior():POH_Behavior
setBehavior(POH_Behavior):void
+behavior
0..1
<<Java Class>>
OrderManager
sohmsPlateform.productHolon
order: ProdOrder
outBoxSender: OutBoxSender
finishedPHs: Set<PH>
activePHs: Set<PH>
OrderManager()
OrderManager(OutBoxSender,ProdOrder)
launchOrder():void
launchPallets(int):void
phIsFinised(PH):void
hashCode():int
equals(Object):boolean
computeNumOfPallets():int
launchPallet():void
+assignedPallets
0..*
-resourceDirectory
0..*
<<Java Class>>
RH
sohmsPlateform.resourceHolon
rhCount: int
ListSize: int
serviceOffer: ArrayList<MServ_Imp>
resourceSchedule: LinkedList<Task_RH>
portSchedules: ConcurrentHashMap<String,LinkedList<Task_RH>>
<<Java Interface>>
POH_Behavior
sohmsPlateform.orderHolon
setAssociatedPOH(P_OH):void
setPlanState(String):void
getPlanState():String
getAssociatedPOH():P_OH
Outline
getPlanState():String
setAssociatedPOH(P_OH):void
setPlanState(String):void
P_OH
RH
ListSize: int
rhCount: int
portSchedules: ConcurrentHash
resourceSchedule: LinkedList<T
roh: R_OH
serviceOffer: ArrayList<MServ_I
sil: RH_SIL
RH()
RH(String,String,String,String)
addService(MServ_Imp):void
equals(Object):boolean
generateResourceProfile():RH_P
getServByType(MServ_Type):M
getServiceByName(String):MSe
getServiceOffer():ArrayList<MS
initPortScheules():void
setServiceOffer(ArrayList<MSer
ROH_Behavior
R_OH
associatedRH: RH
behavior: ROH_Behavior
currentSetup: int
negociationTime: long
Problems
Javadoc
Declaration
Console
Progress
Metrics - SOHMS
Dependency Graph View
Layered Package Graph View
Layers - SOHMS

Metric	Total	Mean	Std. Dev.	Maxim...	Resource causing Maximum	Method
McCabe Cyclomatic Complexity (avg/max per method)		2,046	4,458	113	/SOHMS/src/sil/NotificationReader.java	run
Number of Parameters (avg/max per method)		0,878	1,09	8	/SOHMS/src/commHMS/SocketMessage.java	SocketMessage
Nested Block Depth (avg/max per method)		1,444	1,072	9	/SOHMS/src/protocols/Protocol_RHmodify.java	handleMessage
Afferent Coupling (avg/max per packageFragment)		20,733	20,018	64	/SOHMS/src/mService	
Efferent Coupling (avg/max per packageFragment)		4,333	5,804	25	/SOHMS/src/sil	
Instability (avg/max per packageFragment)		0,343	0,316	1	/SOHMS/src/msAccess	
Abstractness (avg/max per packageFragment)		0,059	0,134	0,5	/SOHMS/src/sohmsPlateform/orderHolon	
Normalized Distance (avg/max per packageFragment)		0,598	0,305	1	/SOHMS/src/exceptions	
Depth of Inheritance Tree (avg/max per type)		1,375	0,807	4	/SOHMS/src/exceptions/InvalidProductException.java	

Outline

- MDRE for software maintenance
- MDRE Case studies
- Discussion – return on experience
- Conclusion and perspectives

Discussion

General observations

MDRE tools provide convenient abstract views of the code

+ static representation of the code
- behavioural abstraction is complex to establish

More intelligent algorithms are required to raise in abstraction

1. Inject engineering information (design and coding rules)
2. Heuristics to separate components from or data types.
3. Iterative Roundrip for incremental discovering.

and lessons learnt from experience

Discussion

Findings and lessons

1. The process is guided by the objectives (what you look for) and the results will depend on them.
2. Automatic high-level reverse-engineering for general purpose languages (even OO-only) stay a myth .
3. One step reverse-engineering is impossible to raise in abstraction.
4. There are no universal process.
5. A reverse engineering technique, designed for a given goal (lesson 1) in a given context (lesson 4), will be improved by applying it to new case studies.
6. Discovering a model is much harder than comparing a model with an implementation.
7. MDE helps in MDRE.
8. Never ending process

Context sensitive

Semantic distance

Small steps

Custom

Learning

Repository

Reversibility /traceability

Roundtrip

Tracks for best practices

Outline

- MDRE for software maintenance
- MDRE Case studies
- Discussion
- Conclusion and vision

Conclusion

Evolution in maintenance

- Business and duties
- Technical debt

Missing traceability

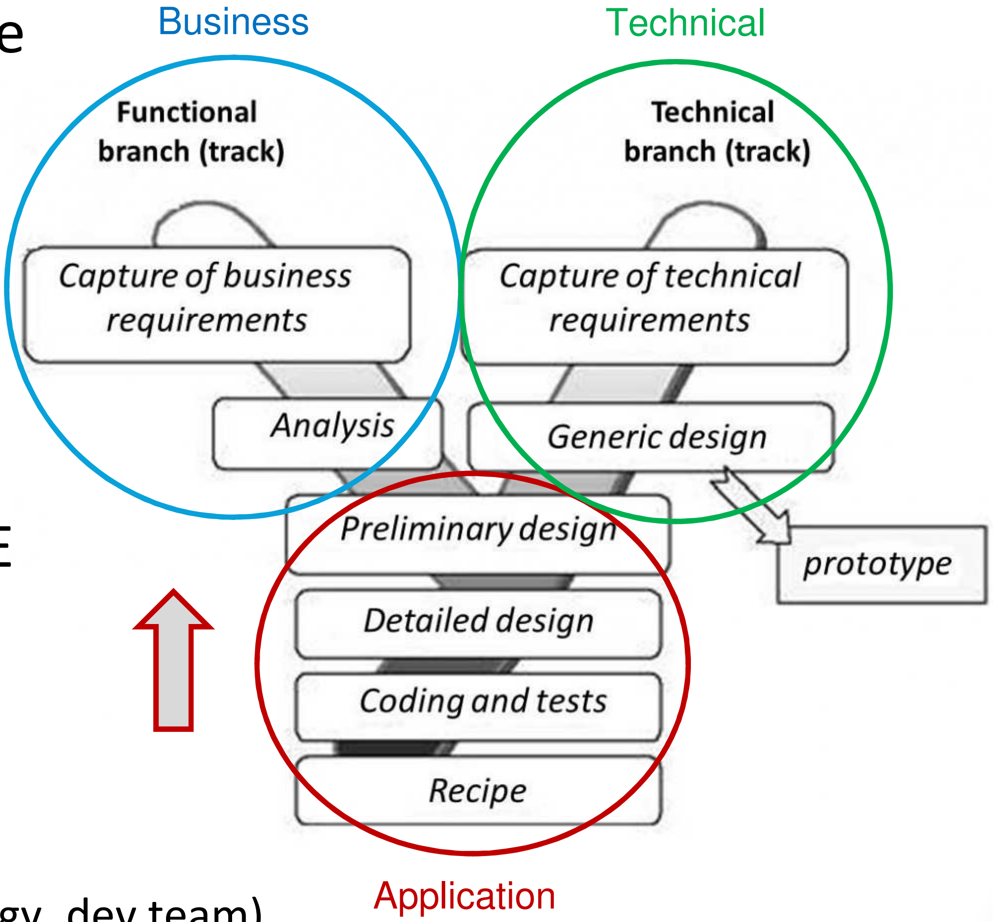
- Abstraction debt

MDRE helps in redo MDE

- Find abstractions
- Reconnect to models

Often context sensitive

- Rules and patterns (technology, dev team)
- Exceptions

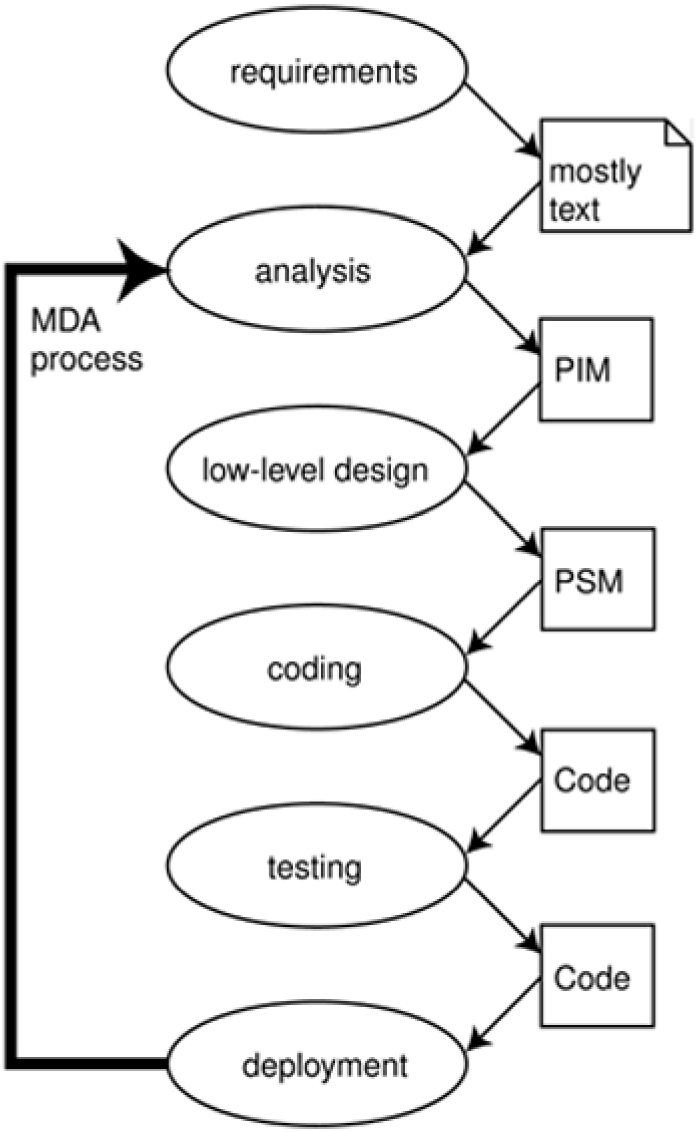


https://www.mysciencework.com/media/20151129/f72e6f12b01703ebd06f9157734b9fb6314fe11e.jpeg

Conclusion

Vision and Perspectives

1. Maintenance is a continuous [incremental] development

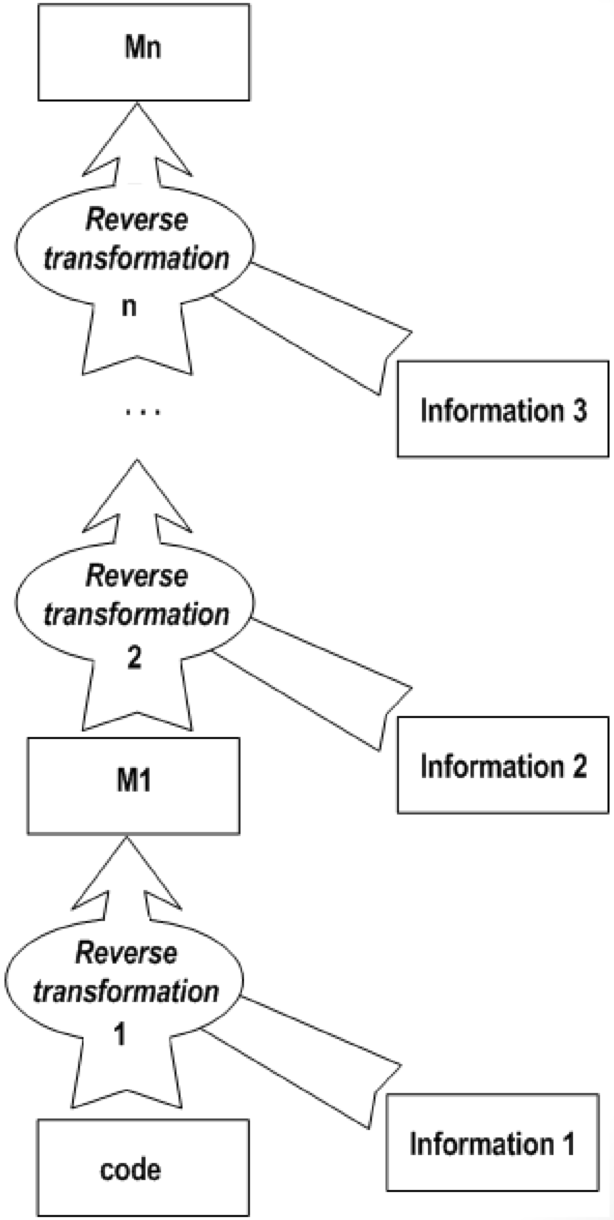


[Kleppe et al. 2003]

Conclusion

Vision and Perspectives

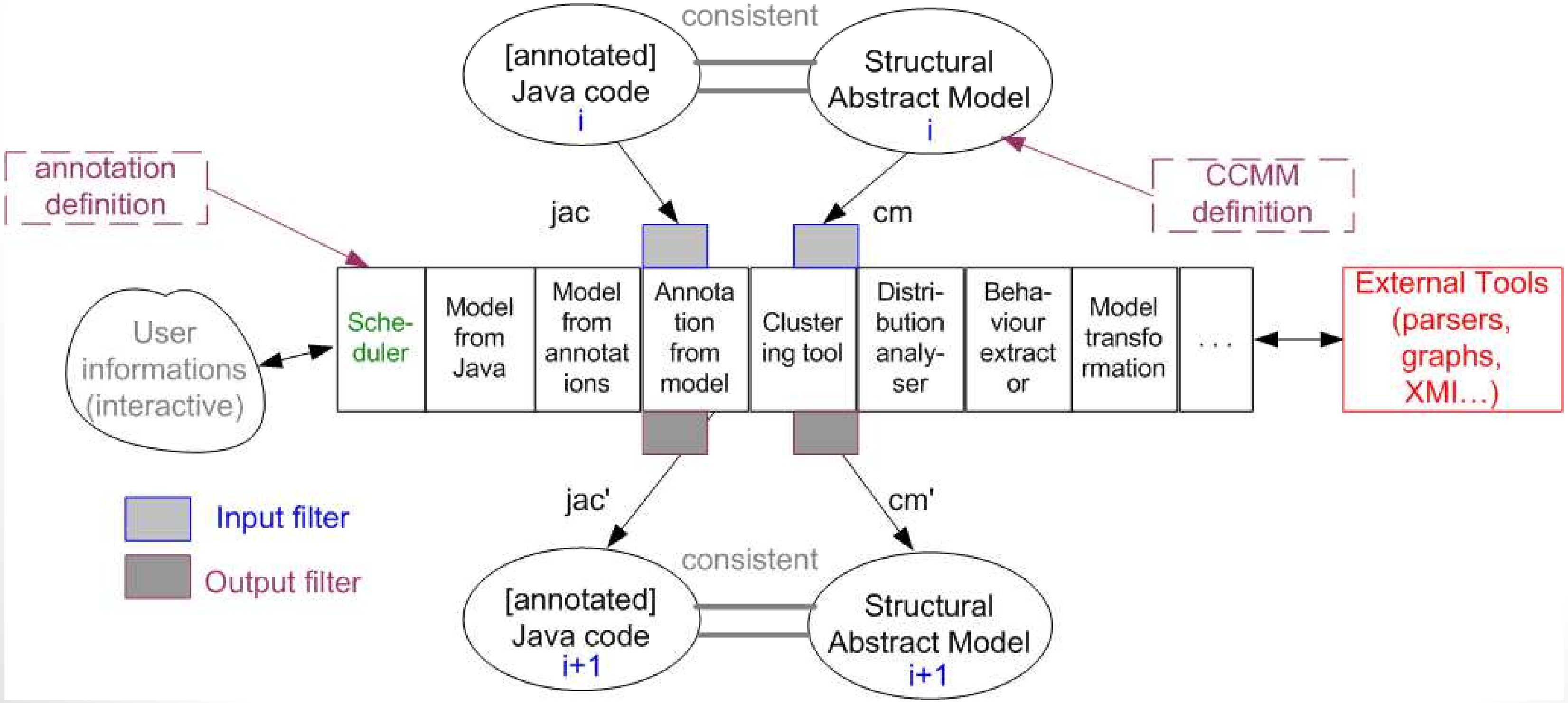
1. Maintenance is a continuous [incremental] development
2. MDRE transformation process with small step transformations



Conclusion

Vision and Perspectives

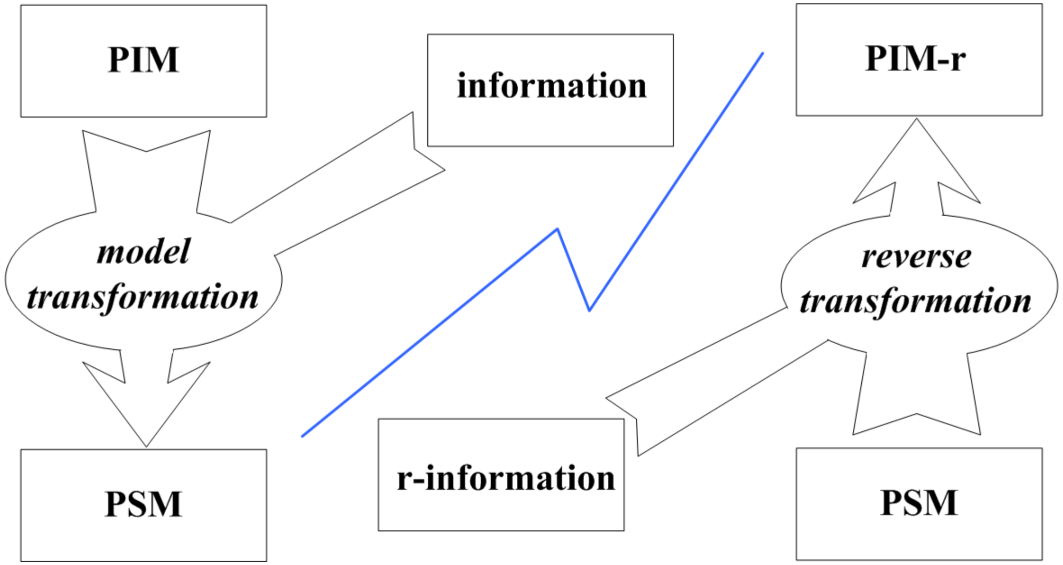
1. Maintenance is a continuous [incremental] development
2. MDRE transformation process with small step transformations
3. Each transformation picks from MDRE toobox - customization



Conclusion

Vision and Perspectives

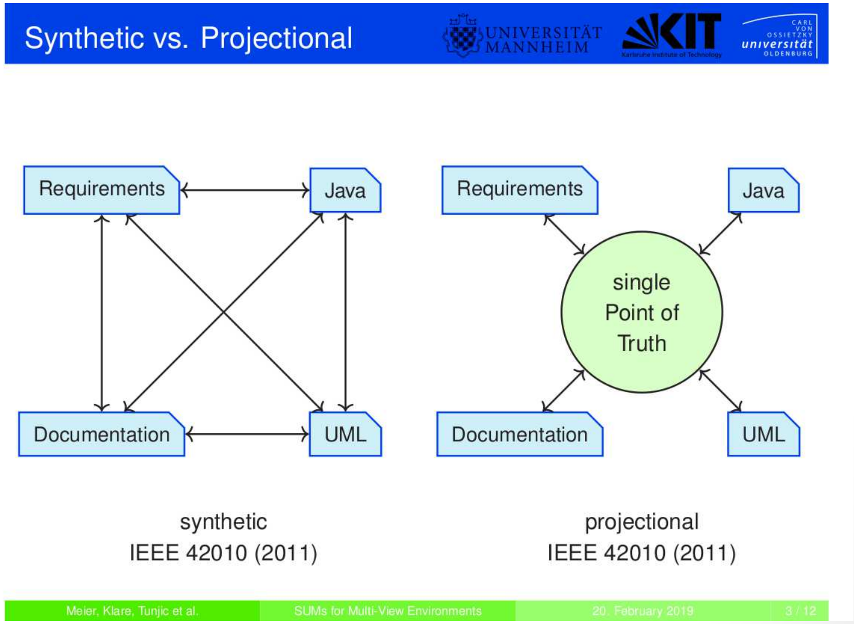
1. Maintenance is a continuous [incremental] development
2. MDRE transformation process with small step transformations
3. Each transformation picks from MDRE toobox - customization
4. MD(R)E is an iterative and round trip process = connect code to models



Conclusion

Vision and Perspectives

1. Maintenance is a continuous [incremental] development
2. MDRE transformation process with small step transformations
3. Each transformation picks from MDRE toobox - customization
4. MD(R)E is an iterative and round trip process = connect code to models
5. Consistency



Synthetic vs. Projectional

synthetic
IEEE 42010 (2011)

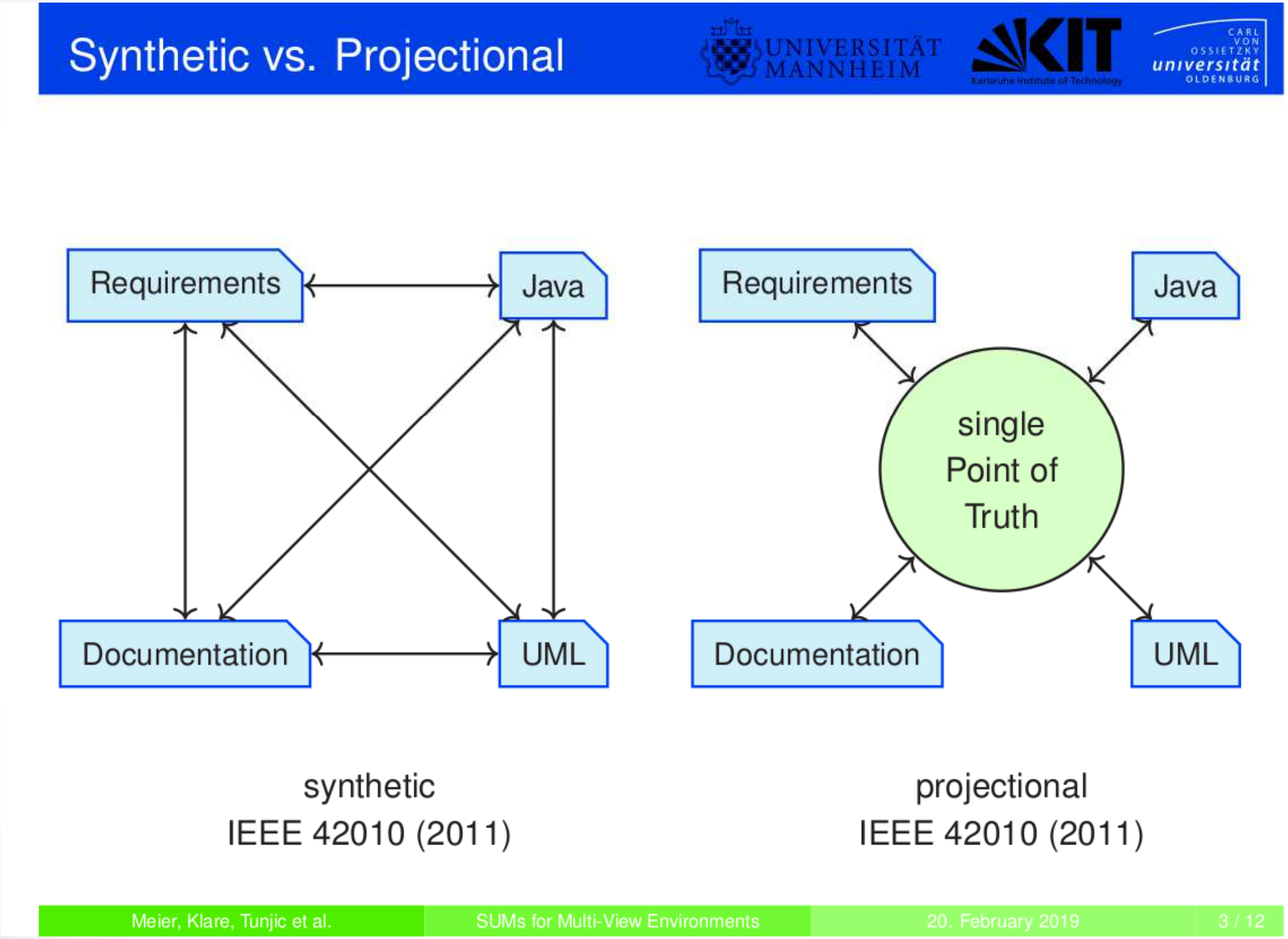
projectional
IEEE 42010 (2011)

Meier, Klare, Tunjic et al. SUMs for Multi-View Environments 20. February 2019 3 / 12

See presentation of Johannes Meier (MW2019)

Conclusion

Vision and Perspectives



Synthetic vs. Projectional

synthetic
IEEE 42010 (2011)

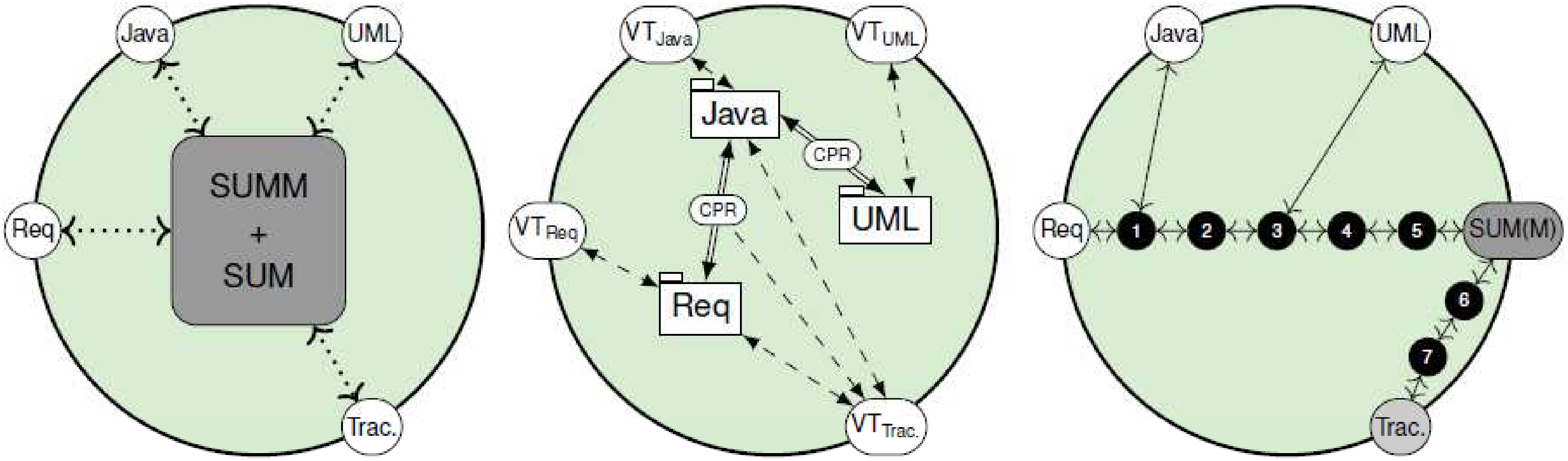
projectional
IEEE 42010 (2011)

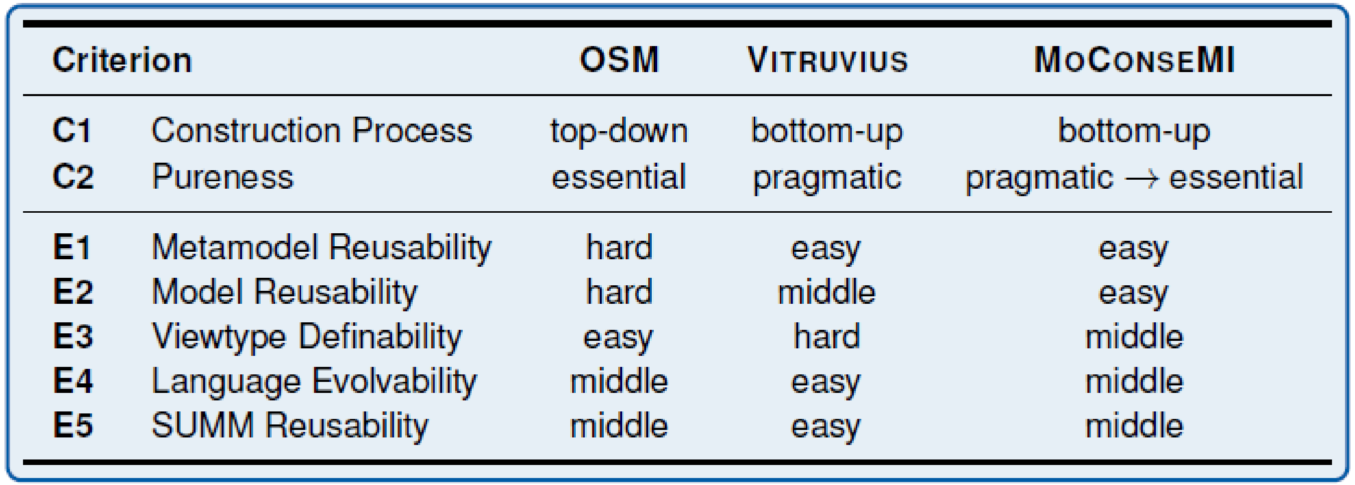
Meier, Klare, Tunjic et al. SUMs for Multi-View Environments 20. February 2019 3 / 12

See presentation of Johannes Meier (MW2019)

Conclusion

Vision and Perspectives





Criterion		OSM	Vitruvius	MoConseMI
C1	Construction Process	top-down	bottom-up	bottom-up
C2	Pureness	essential	pragmatic	pragmatic → essential
E1	Metamodel Reusability	hard	easy	easy
E2	Model Reusability	hard	middle	easy
E3	Viewtype Definability	easy	hard	middle
E4	Language Evolvability	middle	easy	middle
E5	SUMM Reusability	middle	easy	middle

See presentation of Johannes Meier (MW2019)

Thank you for your attention

Any Questions ?!

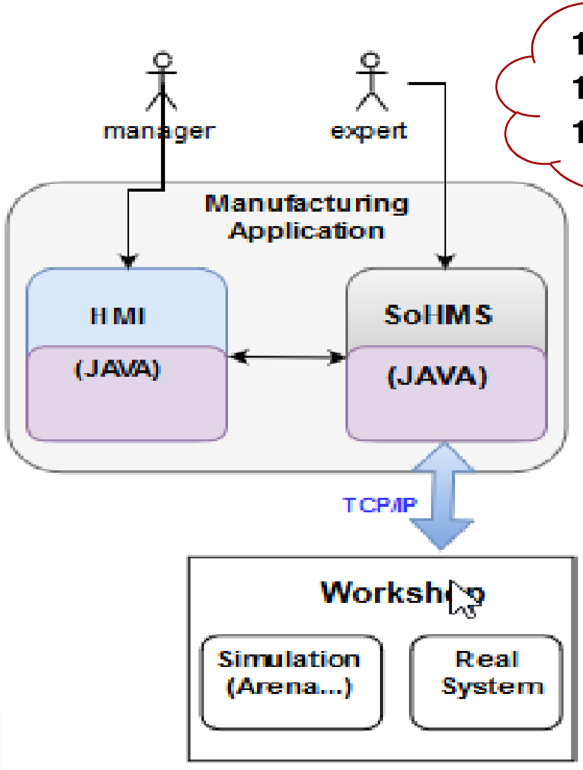
MDRE Case study 3

Re-engineering a Manufacturing application

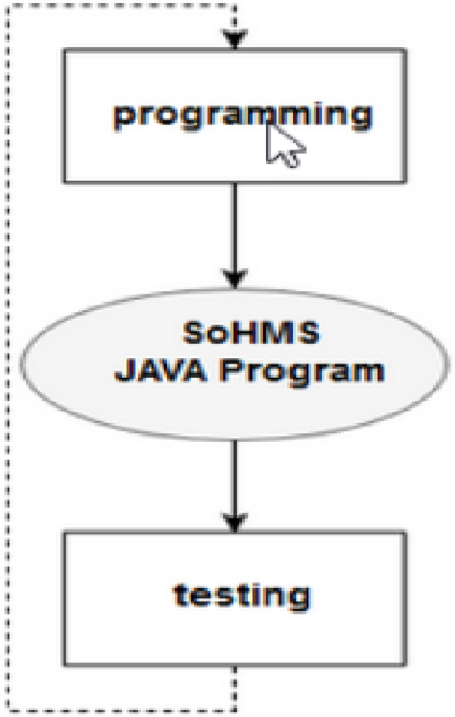
- Goal: improve the software quality for better maintenance and verification, revisit the software process
- MDRE: discover architecture models

160 classes,
1240 methods,
14802 LoC

- **Lack of Modularity**
- **Lack of Abstraction**



Application Architecture

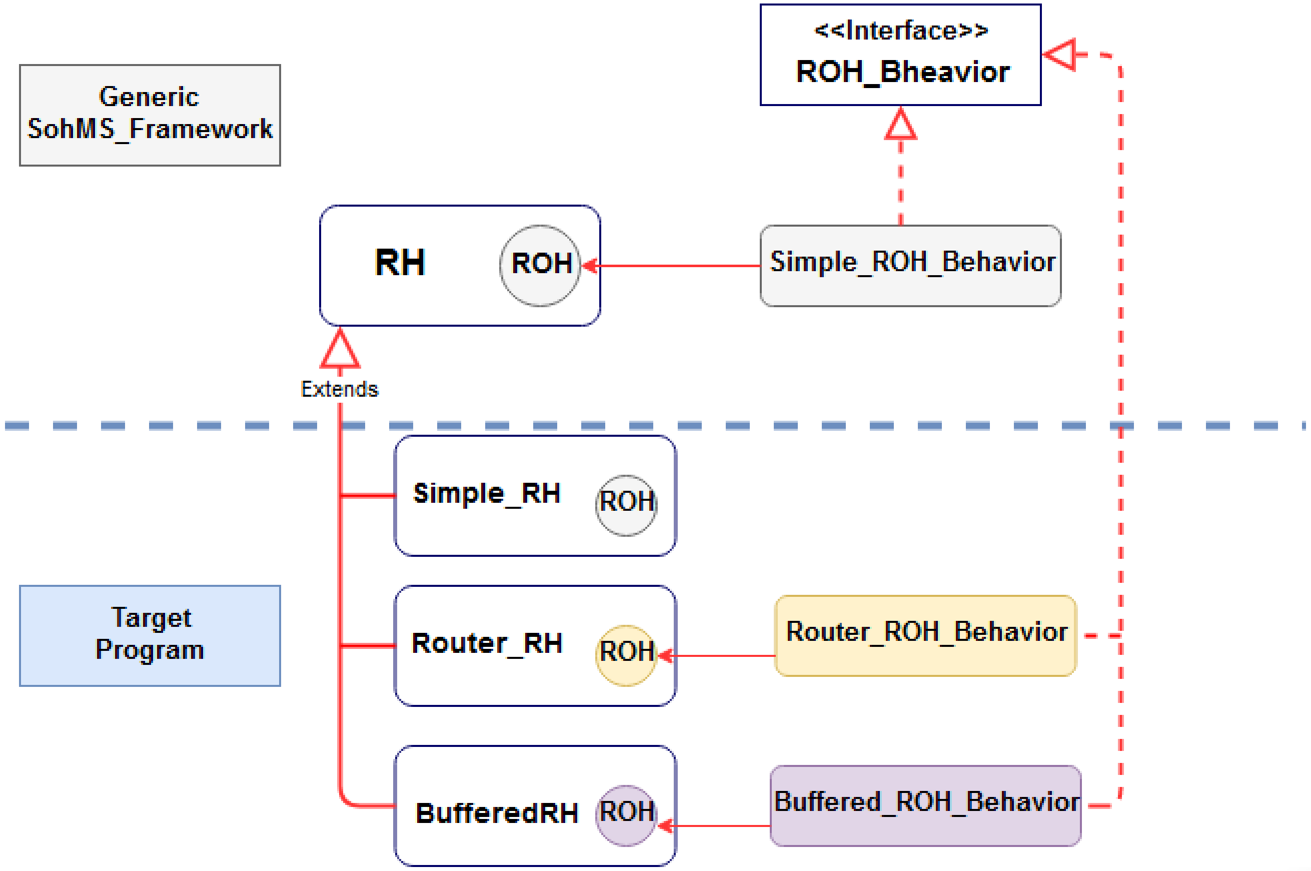


Software Construction Process

MDRE Case study 3

New Vision

Re-engineering a Manufacturing application



<<Interface>>
ROH_Bheavior
Generic
SohMS_Framework
RH
ROH
Simple_ROH_Behavior
Extends
Simple_RH
ROH
Target
Program
Router_RH
ROH
Router_ROH_Behavior
BufferedRH
ROH
Buffered_ROH_Behavior