

**JKU**

**JOHANNES KEPLER  
UNIVERSITY LINZ**

# EFFECTIVELY USING AND REUSING ENGINEERING KNOWLEDGE IN A COLLABORATIVE ENGINEERING ENVIRONMENT



**Alexander Egyed**

[alexander.egyed@jku.at](mailto:alexander.egyed@jku.at), <http://www.isse.jku.at>



Institute for  
**Software Systems Engineering**  
Technologies for building better software

# WHO AM I?

Currently at **Johannes Kepler University**

- Professor for **Software-Intensive Systems**
- Vice Rector for **Research**
- Head of **Institute for Software Systems Engineering**

<http://www.isse.jku.at>

Past Affiliations:

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

Doctorate Degree:

- **University of Southern California**, USA 2000



## Johannes Kepler University

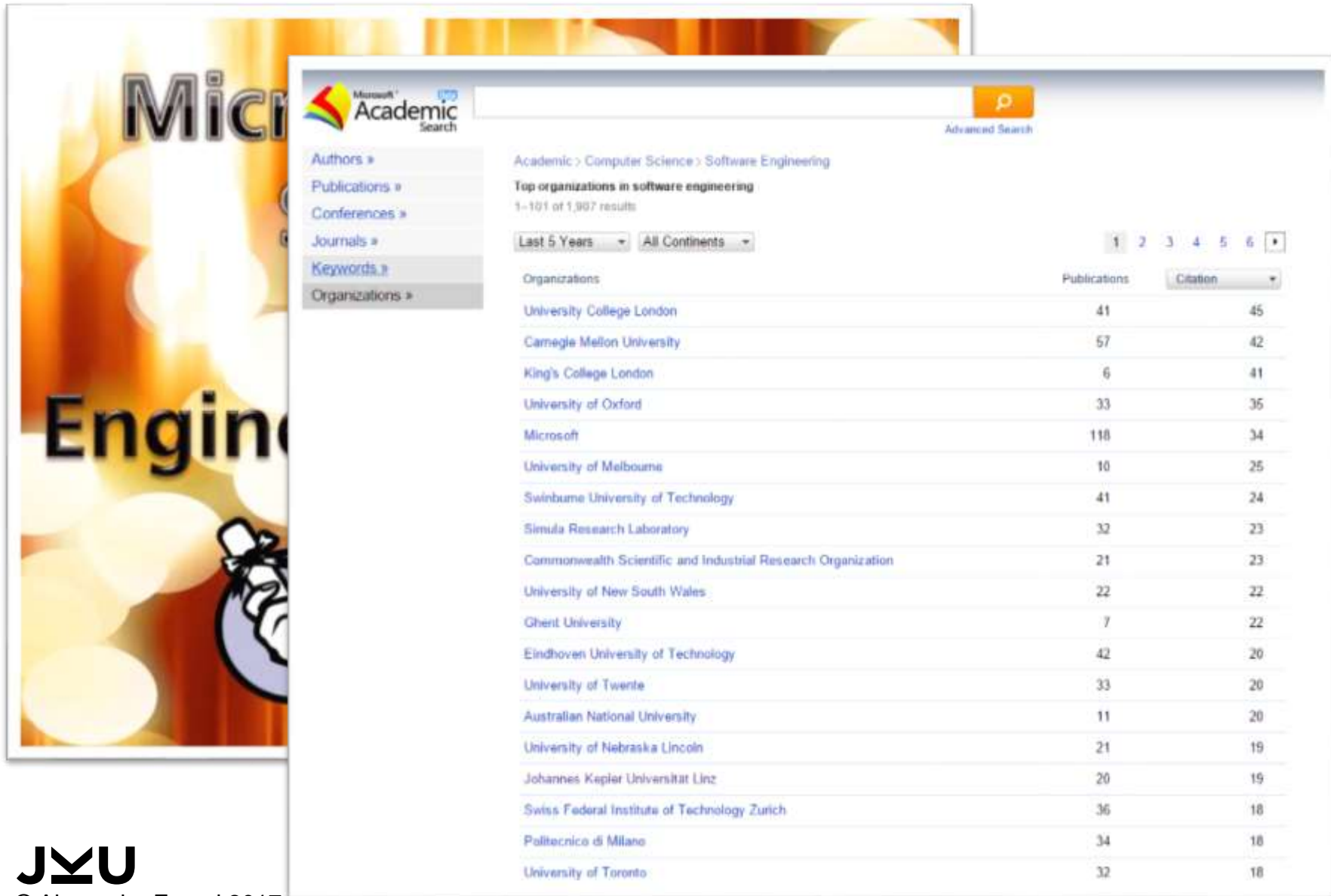
- 20.000 Students, 2.000 Staff
- Technology, Natural Sciences, Social Sciences, Business Administration, Law, Education, and Medicine)
- Largest Research and Teaching Institution in **Upper Austria**







# A FEW RANKINGS



The screenshot shows the Microsoft Academic Search interface. The search results are for 'Software Engineering' in the 'Computer Science' category. The top organizations are ranked by the number of publications and citations. The table below lists the top 20 organizations.

Organizations	Publications	Citation
University College London	41	45
Carnegie Mellon University	57	42
King's College London	6	41
University of Oxford	33	35
Microsoft	118	34
University of Melbourne	10	25
Swinburne University of Technology	41	24
Simula Research Laboratory	32	23
Commonwealth Scientific and Industrial Research Organization	21	23
University of New South Wales	22	22
Ghent University	7	22
Eindhoven University of Technology	42	20
University of Twente	33	20
Australian National University	11	20
University of Nebraska Lincoln	21	19
Johannes Kepler Universität Linz	20	19
Swiss Federal Institute of Technology Zurich	36	18
Politecnico di Milano	34	18
University of Toronto	32	18

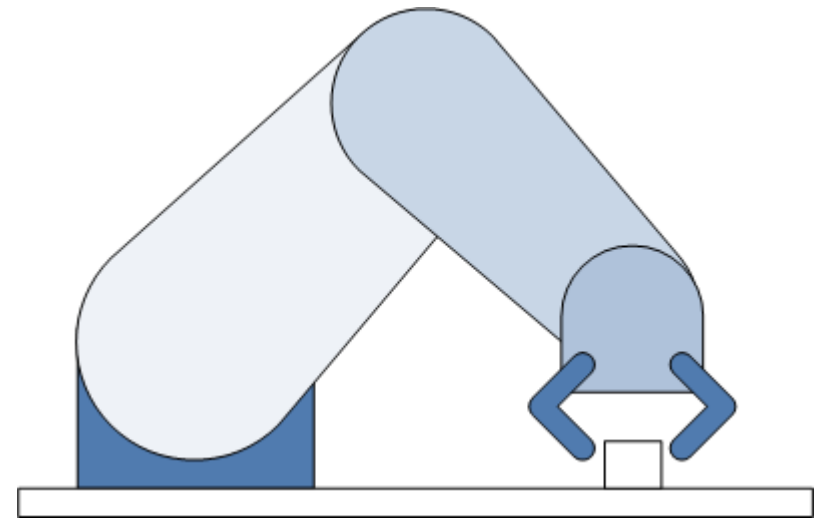
# Collaborative Engineering

# TEAM WORK

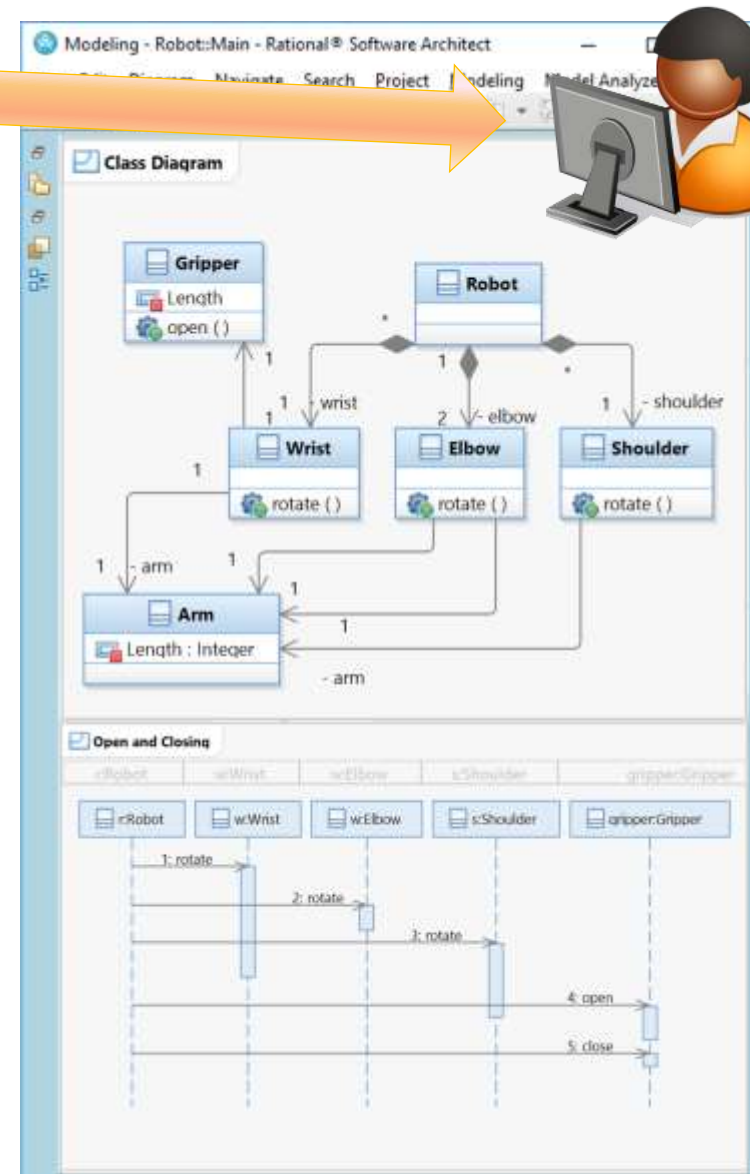
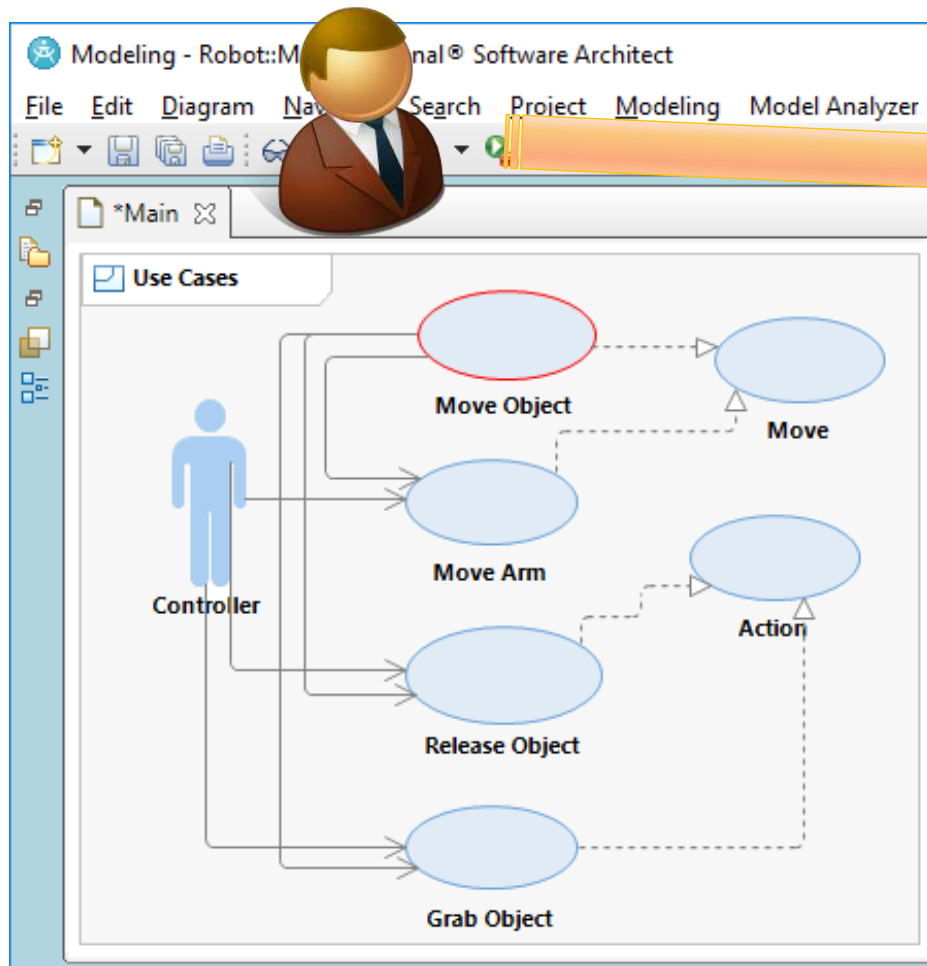


# BUILDING A ROBOT

- The robot has specific requirements, perhaps with a specific customer in mind



# BUILDING THE FIRST ROBOT

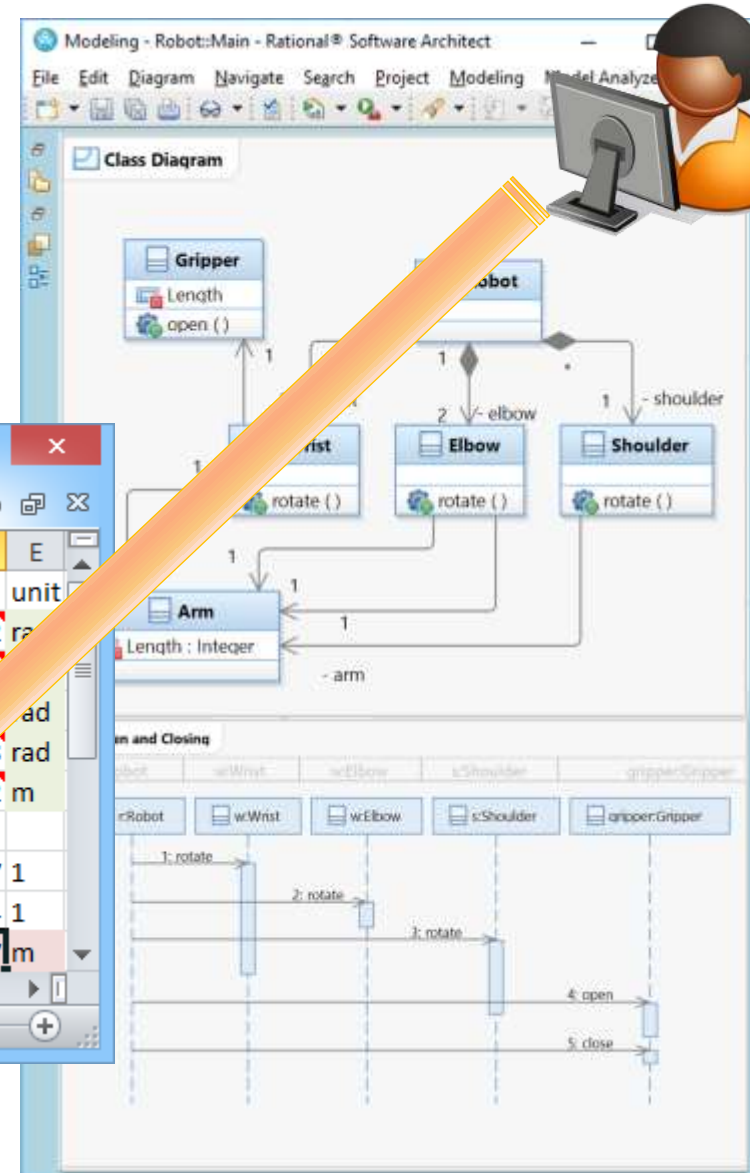


# BUILDING THE FIRST ROBOT

DistanceAngles2Length.xlsx - Microsoft Excel

	A	B	C	D	E
1	name	sign	formula	value	unit
2	alphaMin	amin		0.2 rad	
3	alphaMax	amax		1 rad	
4	betaMin	bmin		0.8 rad	
5	betaMax	bmax		2 m	
6	desired GraspingDistance	ddes			
7					
8	maxNormedRadius	r1	$(\cos(\alpha_{min}) / (\alpha_{min} + \beta_{max}))$	1.97005907	1
9	minNormedRadius	r2	$(\cos(\alpha_{max}) / (\alpha_{max} + \beta_{min}))$	0.39716924	1
10	minArmLength	x	$ddes / (r1 - r2)$	1.27154487	m

LengthCalculation



# BUILDING THE FIRST ROBOT

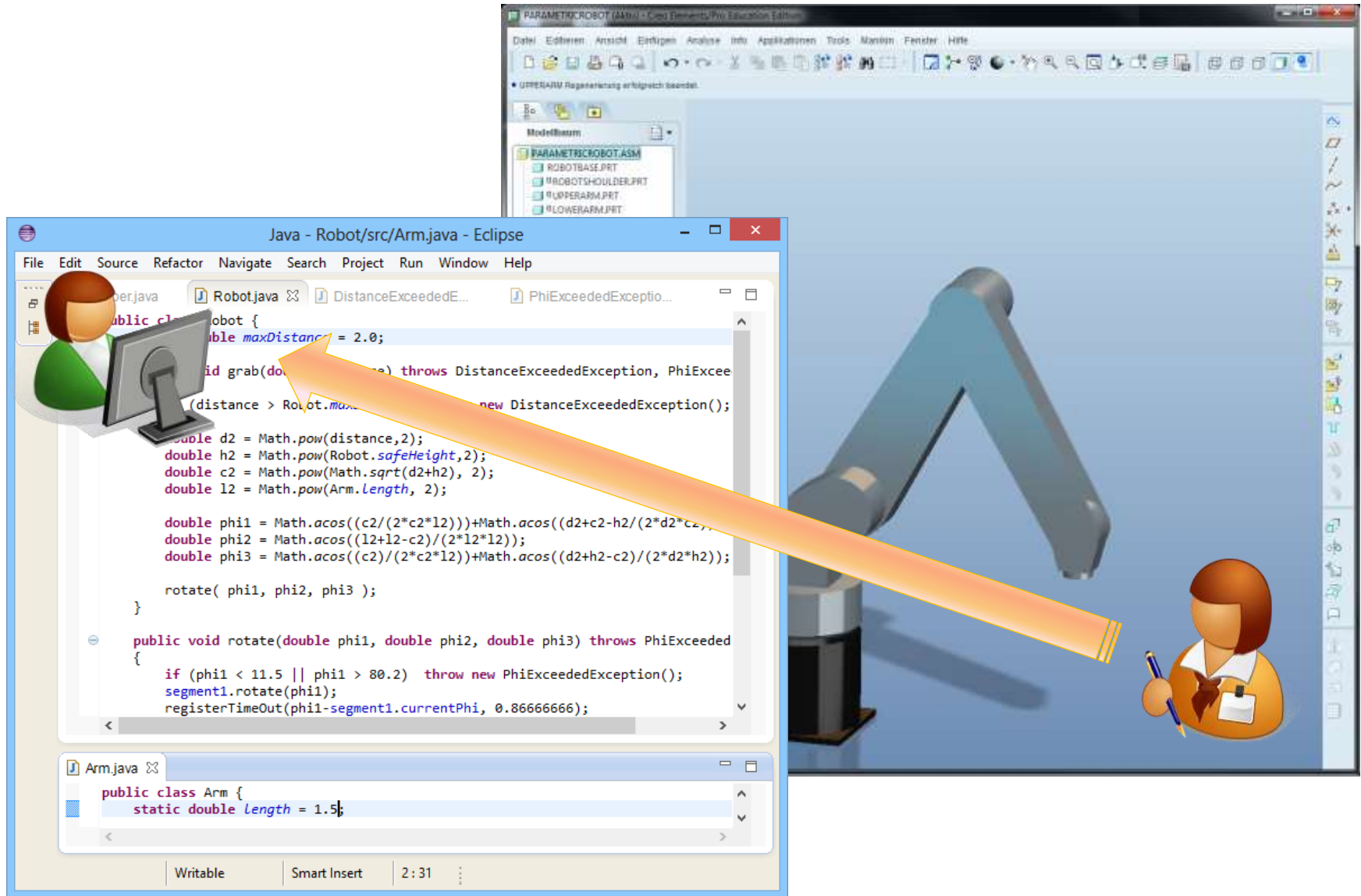
The image displays a CAD software window titled "PARAMETRICROBOT (ASM) - Creo Elements/Pro, Education Edition". The main view shows a 3D model of a robotic arm with a black base and grey joints. The left-hand "ModelBrowser" pane lists the assembly components: PARAMETRICROBOT.ASM, ROBOTBASE.PRT, #ROBOTSHOULDER.PRT, #UPPERARM.PRT, #LOWERARM.PRT, and #WRIST.PRT. A status bar at the top indicates "UPPERARM Regenerierung erfolgreich beendet".

In the foreground, a spreadsheet window titled "DistanceAngles2Length.xlsx" is open. It contains a table with the following data:

	A	B	
1	name	sign	formula
2	alphaMin	amin	
3	alphaMax	amax	
4	betaMin	bmin	
5	betaMax	bmax	
6	desired GraspingDistance	ddes	
7			
8	maxNormedRadius	r1	(COS(am
9	minNormedRadius	r2	(COS(am
10	minArmLength	x	ddes/(r1-

The spreadsheet also shows a formula bar with the value "154487 m" and a status bar with "LengthCalculation". An orange arrow points from the spreadsheet to the robot model. A small cartoon character of a person in a blue suit is sitting at a computer monitor in the bottom left corner.

# BUILDING THE FIRST ROBOT



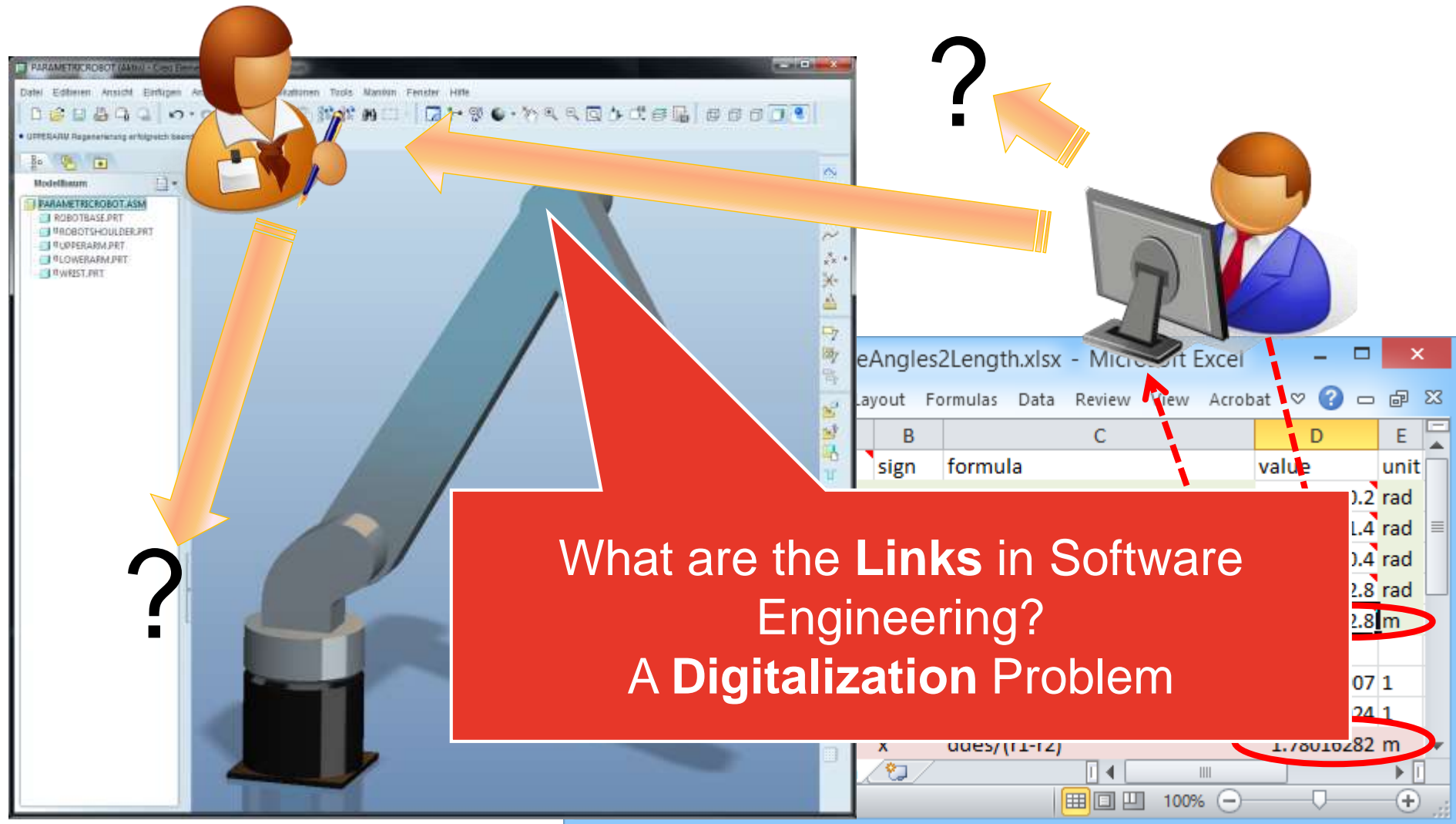




The WHY, The HOW, The WHAT, The WHAT FOR...



# IMPACT OF A CHANGE

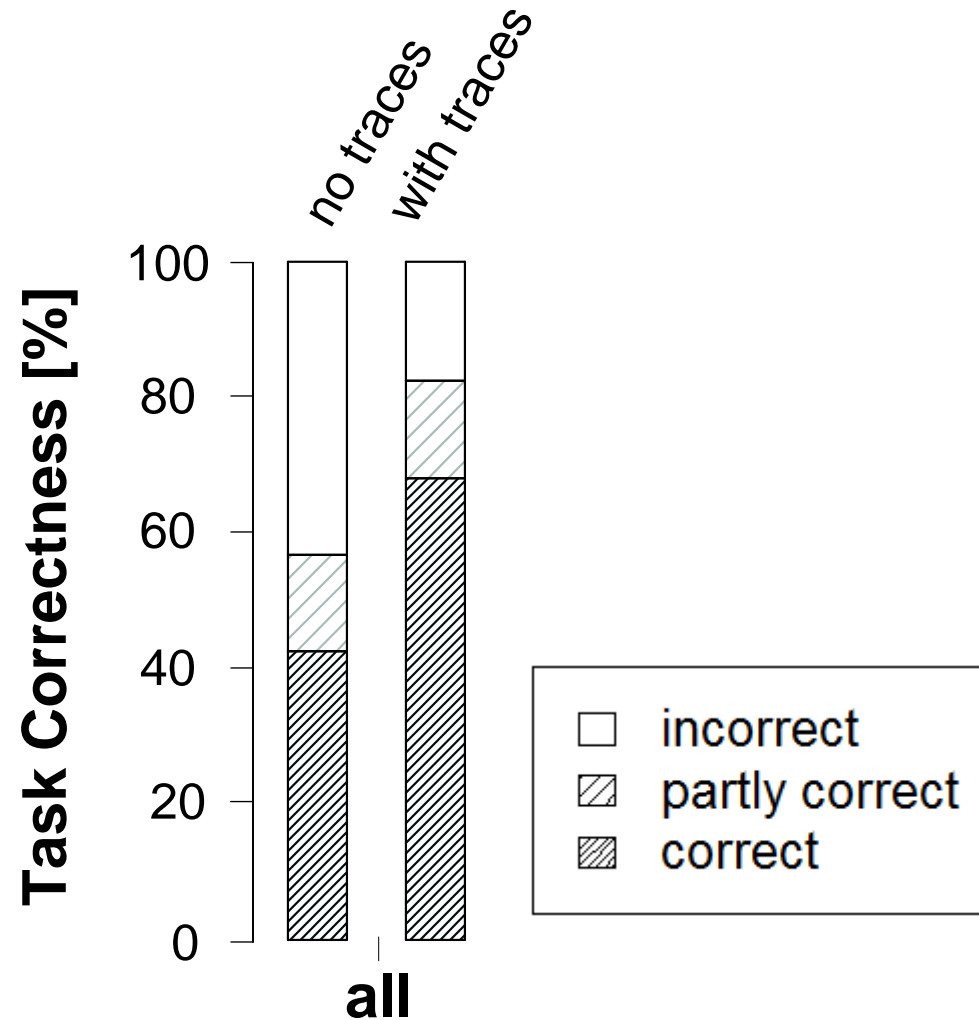
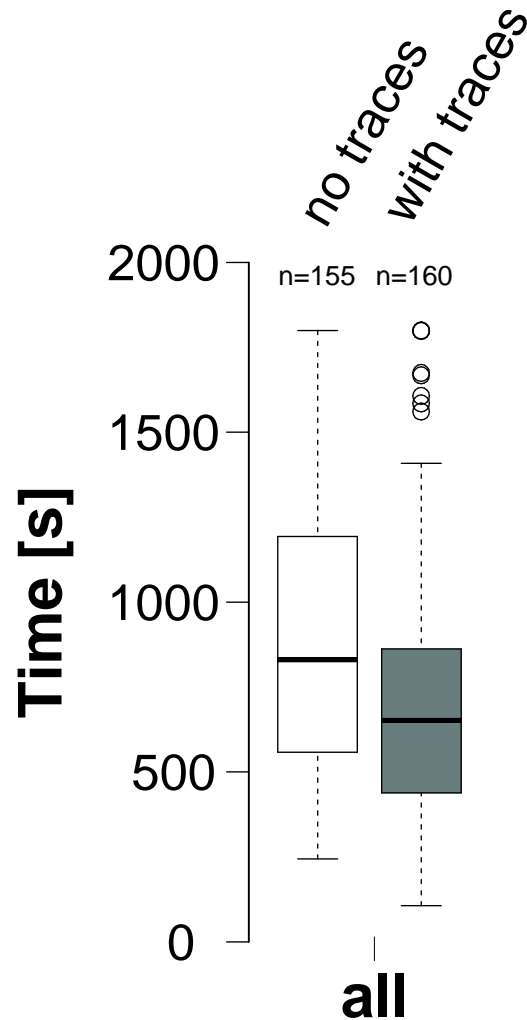


# Are Links Useful?

# EXPERIMENT: ARE WE SURE TRACE LINKS ARE USEFUL?

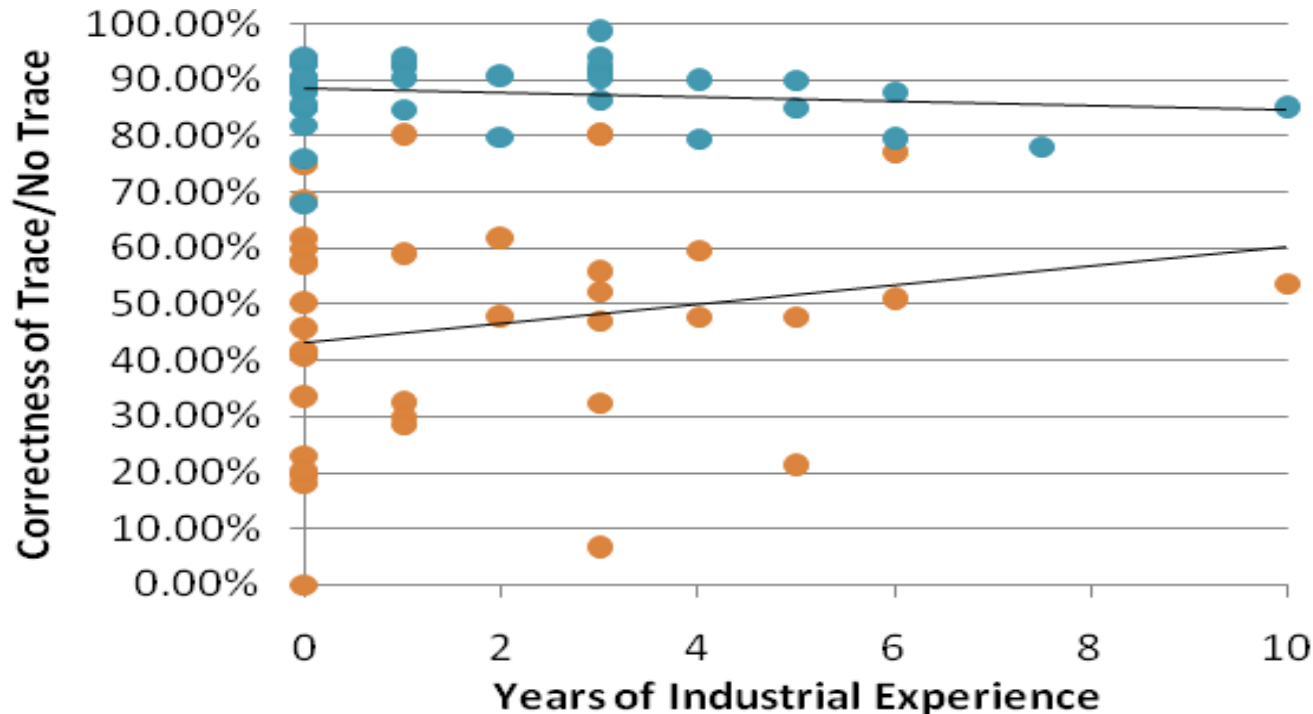
- Performing real tasks on real systems with/without real traces
- 52 students of computer science at the JKU
  - Average industry experience: 1 year
  - Development experience: 5.3 years
  - Development with JAVA: 3.4 years
- 18 practitioners from Upper Austria

# RESULTS: BOTH PROJECTS



# REQUIREMENTS-TO-CODE TRACES ARE ERROR PRONE IF DONE LATER

- Manual trace capture is error prone [Egyed et al. 2010]



- Manual trace validation improves low-quality traces only but worsens high-quality traces [Kong et al. 2011]

# IT'S ALL ABOUT DOCUMENTATION

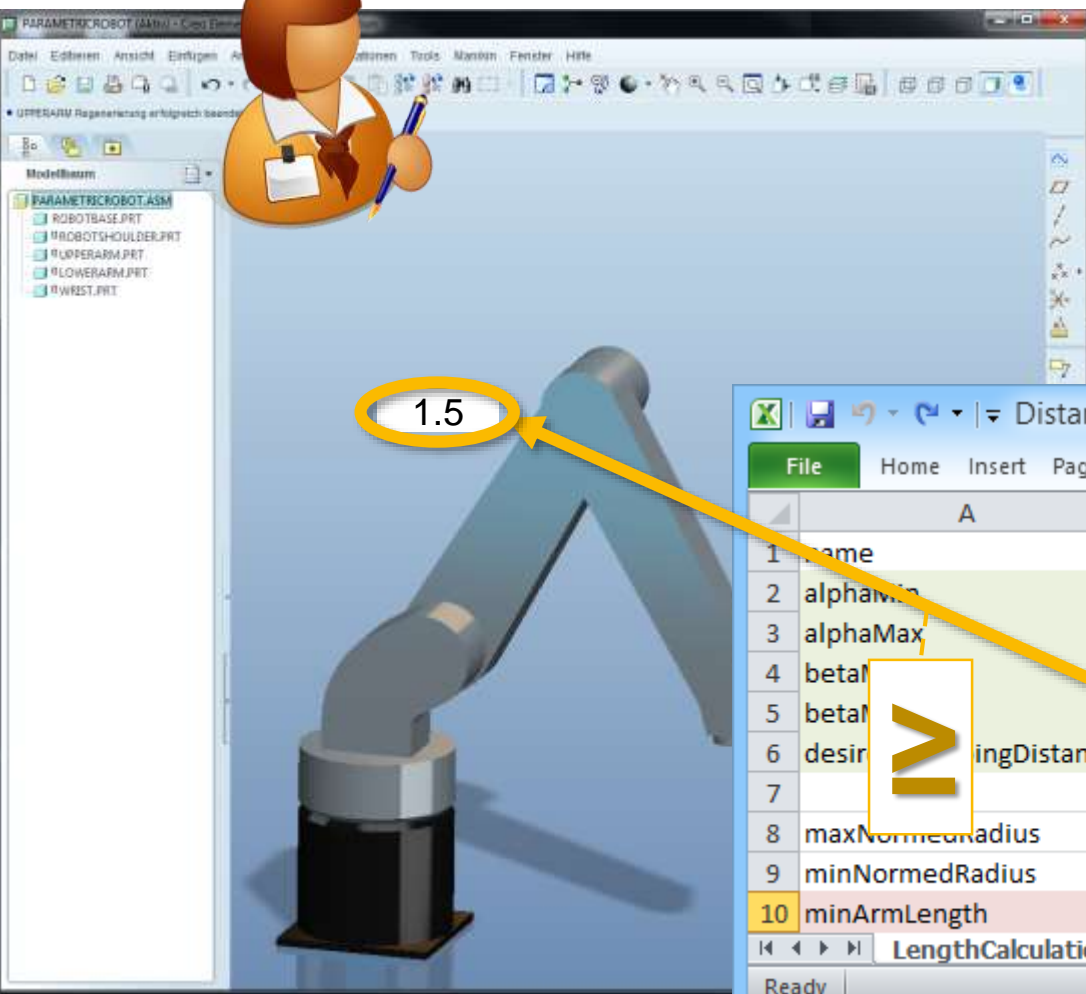
We need to capture right away

- What do we need to capture?
- How do we store it in a manner where we can find it when we need it?
- How do we maintain it to ensure it stays up-to-date?



# Linking and Meaning

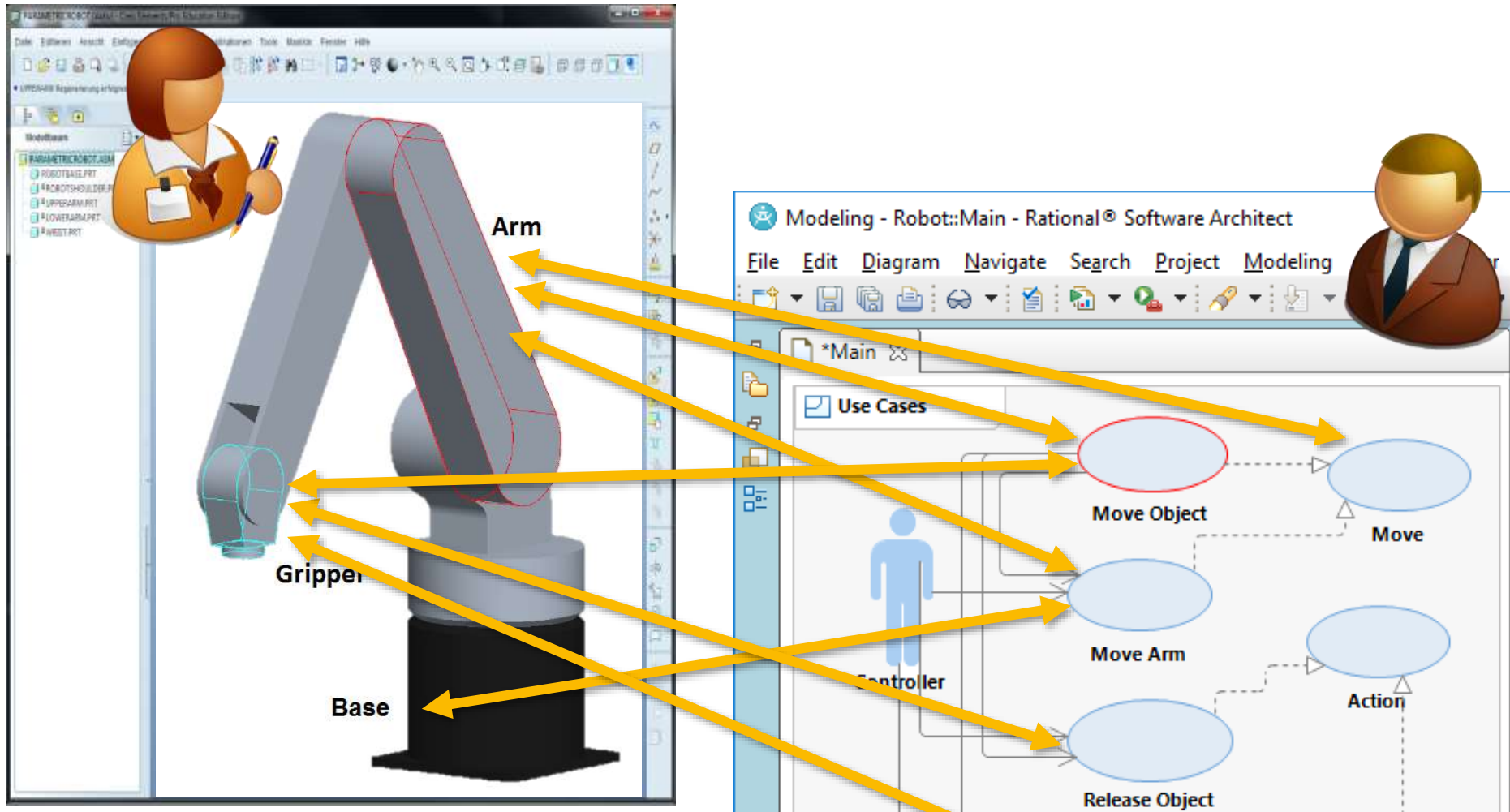
# A LINK EXAMPLE



DistanceAngles2Length.xlsx - Microsoft Excel

	A	B	C	D	E
1	name	sign	formula	value	unit
2	alphaMin	amin		0.2	rad
3	alphaMax	amax		1.4	rad
4	betaMin	bmin		0.4	rad
5	betaMax	bmax		2.8	rad
6	desiredReachingDistance	ddes		2	m
7					
8	maxNormedRadius	r1	$(\cos(\alpha_{\min}) - \cos(\alpha_{\min} + \beta_{\max}))$	1.97005907	1
9	minNormedRadius	r2	$(\cos(\alpha_{\max}) - \cos(\alpha_{\max} + \beta_{\min}))$	0.39716924	1
10	minArmLength	x	$ddes / (r1 - r2)$	1.27154487	m

# MORE LINK EXAMPLES



Rule: `CADComponent.links->exists(|l|.target->exists(c|c.type(UseCase)))`

# TRACE MATRICES TO THE RESCUE

Naming Conventions is a Dangerous Thing

		Hardware (CAD)		
		Grasper	Arm	Platform
Use Case	<i>Move</i>			
	<i>Action</i>			
	Move Object	X	X	
	Move Arm		X	X
	Release Object	X		
	Grab Object	X		

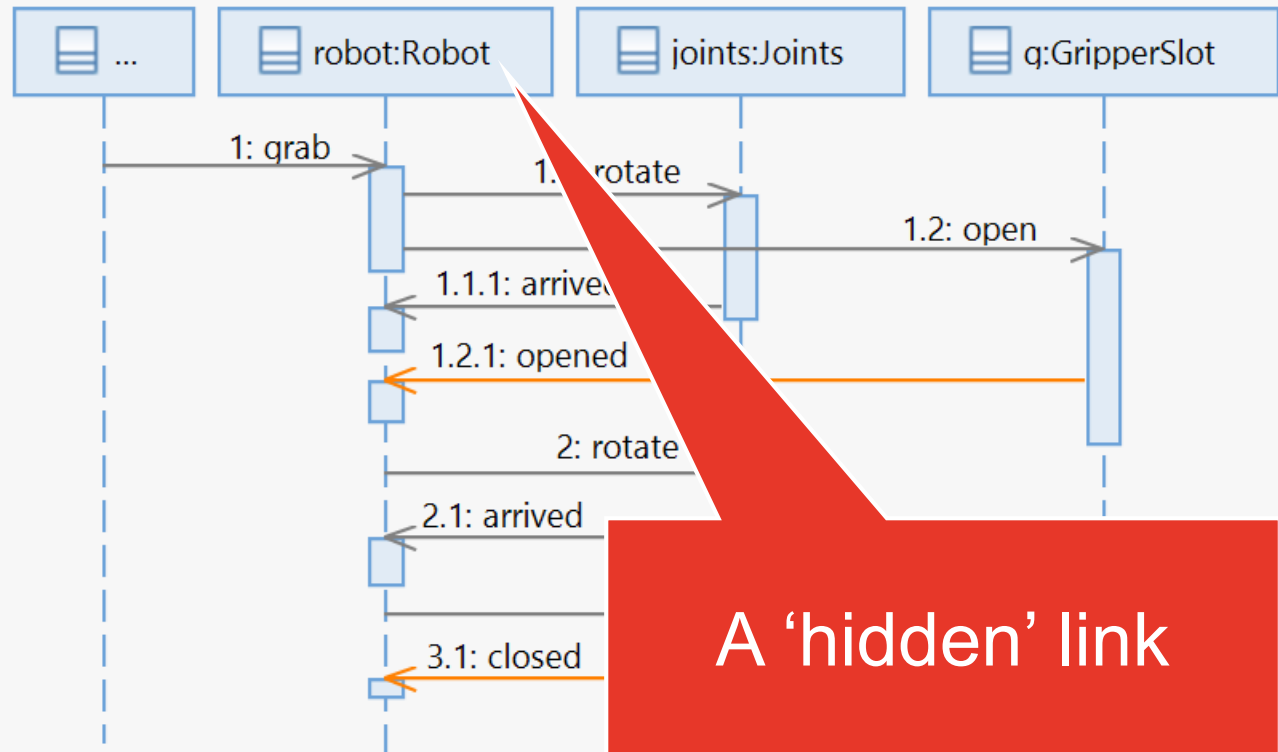
# LIVE FEEDBACK

Rule: `self.receiveEvent.covered->forAll(represents.type.ownedOperation->exists(name=self.name))`



## Grab an Object

control:Co...    robot:Robot    joints:Joints    g:GripperSlot



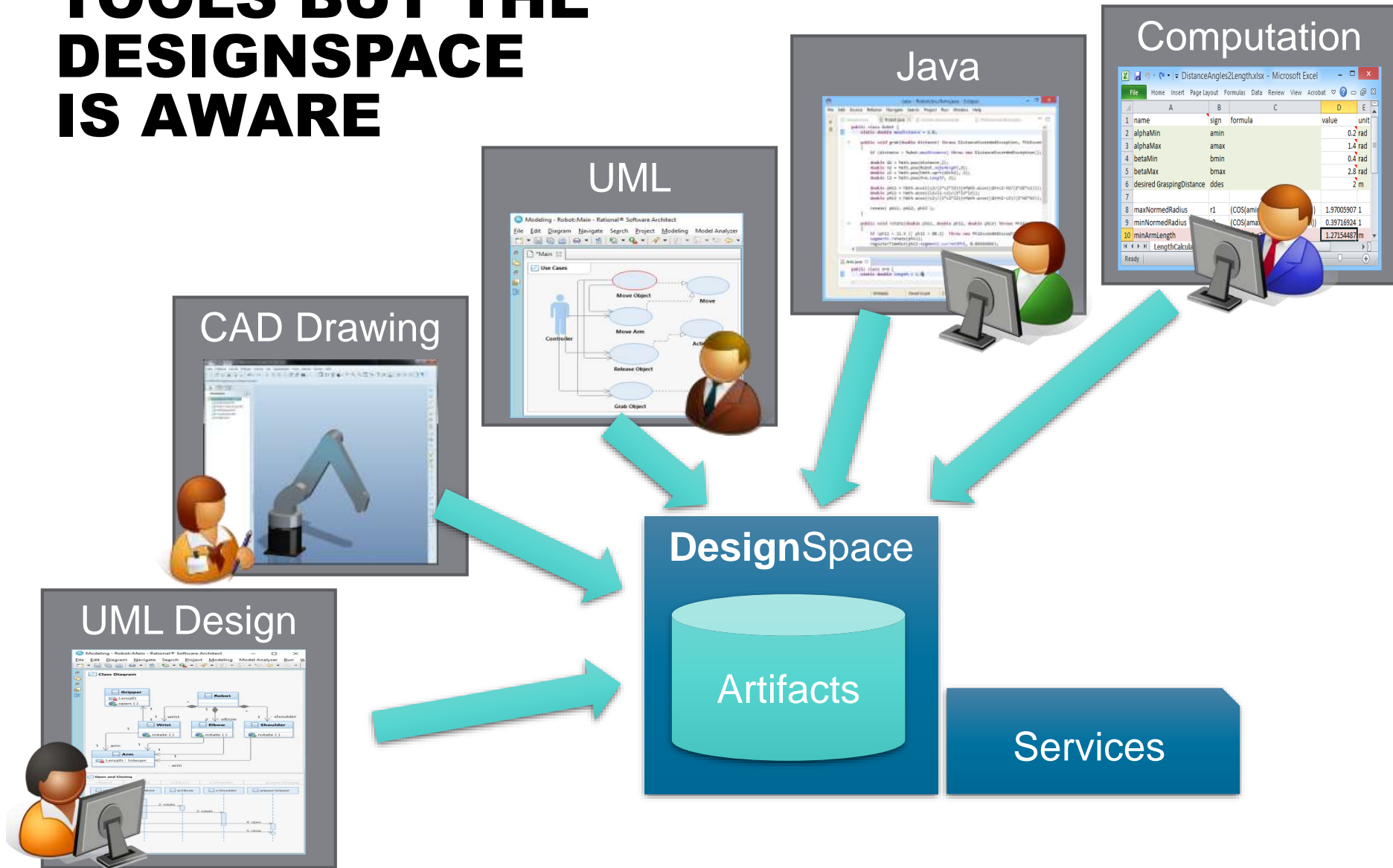
A 'hidden' link

# *DesignSpace*

**a cloud infrastructure for  
integrating cross-tool  
engineering knowledge**



# ENGINEERS CONTINUE TO USE TOOLS BUT THE DESIGNSPACE IS AWARE



# ARTIFACT: REFLECTION ON TOOL

## ARTIFACT

### INSTRUMENTATION AND ANNOTATION

	A	B	C	D	E
1	name	sign	formula	value	unit
2	alphaMin	amin	\$var=AlphaMin.In \$meta=Data	0.2	rad
3	alphaMax	amax		1.4	
4	betaMin	bmin		0.4	
5	betaMax	bmax		2.8	
6	desired GraspingDistance	ddes	\$var=GraspingDistance.In \$meta=Data	2	m
7					
8	maxNormedRadius	r1	(COS(min(COS(amin), COS(amax))))	1.97005907	1
9	minNormedRadius	r2	(COS(max(COS(amin), COS(amax))))	0.39716924	1
10	minArmLength	x	ddes/(r1-r2)	1.27154487	m

## Artifact

Tool=Excel  
 Engineer=PSMITH  
 ID=<unique>  
 alphaMin = 0.2  
 alphaMax = 1.4  
 betaMin = 0.4  
 betaMax = 2.8  
 GraspingDistance=2.8  
 minArmLength = 1.27

# CONSTRAINTS BETWEEN TWO TOOLS IN CLOUD

DesignSpace

Tool Artifacts  
(Fragments)

Artifact

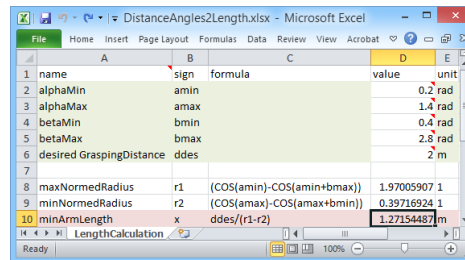
alphaMin=0.2  
alphaMax=1.4  
...  
minArmLength=1.27

Artifact

upper.armlength=1.5  
lower.armlength=1.5  
...

Sync

Sync



	A	B	C	D	E
1	name	sign	formula	value	unit
2	alphaMin	amin		0.2	rad
3	alphaMax	amax		1.4	rad
4	betaMin	bmin		0.4	rad
5	betaMax	bmax		2.8	rad
6	desired GraspingDistance	d-des		2	m
7					
8	maxNormedRadius	r1	(COS(amin)-COS(amin+bmax))	1.97005907	1
9	minNormedRadius	r2	(COS(amax)-COS(amax+bmin))	0.39716924	1
10	minArmLength	x	d-des/(r1-r2)	1.27154487	m



Tools

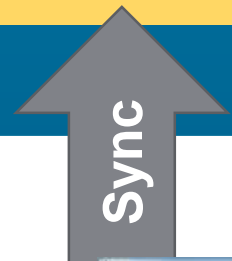
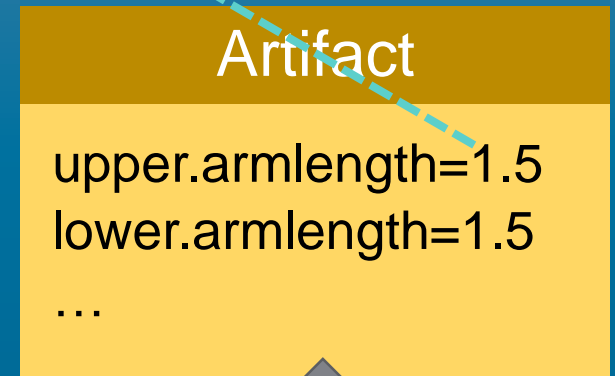
JYU

© Alexander Egyed 2017

# CONSTRAINTS BETWEEN TWO TOOLS IN CLOUD

DesignSpace

Tool Artifacts  
(Fragments)



	A	B	C	D	E
1	name	sign	formula	value	unit
2	alphaMin	amin		0.2	rad
3	alphaMax	amax		1.4	rad
4	betaMin	bmin		0.4	rad
5	betaMax	bmax		2.8	rad
6	desired GraspingDistance	ddes		2.8	m
7					
8	maxNormedRadius	r1	(COS(amin)-COS(amin+bmax))	1.97005907	1
9	minNormedRadius	r2	(COS(amax)-COS(amax+bmin))	0.39716924	1
10	minArmLength	x	ddes/(r1-r2)	1.78016282	m



# CONSTRAINTS ON TOOL MODEL

DesignSpace

Tool Artifacts  
(Fragments)

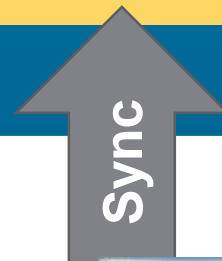
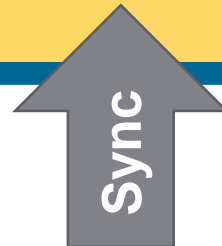


Artifact

alphaMin=0.2  
alphaMax=1.4  
...  
minArmLength=1.78

Artifact

upper.armlength=1.5  
lower.armlength=1.5  
...



	A	B	C	D	E
1	name	sign	formula	value	unit
2	alphaMin	amin		0.2	rad
3	alphaMax	amax		1.4	rad
4	betaMin	bmin		0.4	rad
5	betaMax	bmax		2.8	rad
6	desired GraspingDistance	d-des		2.8	m
7					
8	maxNormedRadius	r1	(COS(amin)-COS(amin+bmax))	1.97005907	1
9	minNormedRadius	r2	(COS(amax)-COS(amax+bmin))	0.39716924	1
10	minArmLength	x	d-des/(r1-r2)	1.78016282	m



Tools

JYU

© Alexander Egyed 2017

# LIVE AND FINE GRANULAR

```
Java - Robot/src/Arm.java - Eclipse
File Edit Source Refactor Navigate Search Project Run Window Help
Gripper.java Robot.java DistanceExceededE... PhiExceededExceptio...
public class Robot {
    static double maxDistance = 2.0;

    public void grab(double distance) throws DistanceExceededException, PhiExcee
    {
        if (distance > Robot.maxDistance) throw new DistanceExceededException();

        double d2 = Math.pow(distance,2);
        double h2 = Math.pow(Robot.safeHeight,2);
        double c2 = Math.pow(Math.sqrt(d2+h2), 2);
        double l2 = Math.pow(Arm.Length, 2);

        double phi1 = Math.acos((c2/(2*c2*l2))+Math.acos((d2+c2-h2)/(2*d2*c2)));
        double phi2 = Math.acos((l2+l2-c2)/(2*l2*l2));
        double phi3 = Math.acos((c2)/(2*c2*l2))+Math.acos((d2+h2-c2)/(2*d2*h2));

        rotate( phi1, phi2, phi3 );
    }

    public void rotate(double phi1, double phi2, double phi3) throws DistanceExceeded
    {
        if (phi1 < 11.5 || phi1 > 80.2) throw new PhiExceededException(phi1);
        segment1.rotate(phi1);
        registerTimeout(phi1-segment1.currentPhi, 0.86666666);
    }
}

Arm.java
public class Arm {
    static double length = 1.5;
}
```

Sync

Artifact

Robot

Arm

grab  
rotate ...

**Java.Class**  
- name: String  
- methods : Set  
- Fields : Set

**Java.Method...**

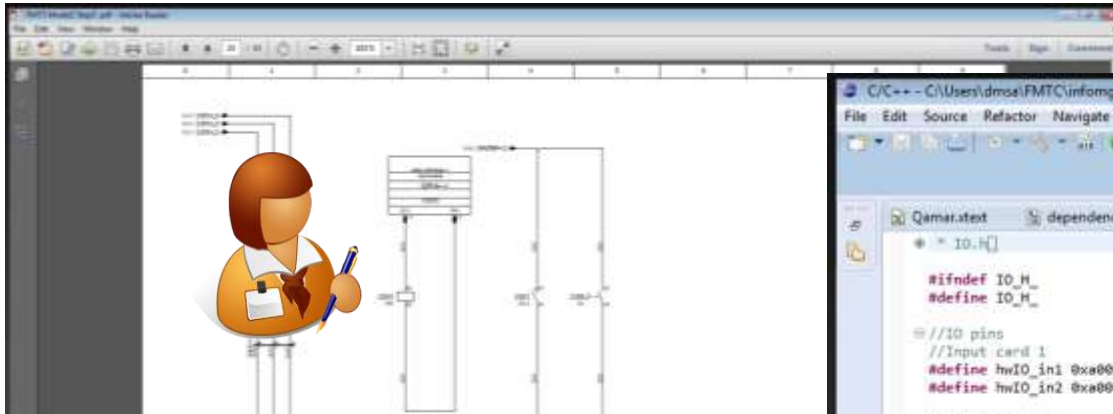
**Java Field**  
- name: String  
- value : Any  
- type : Class

Well-defined Data Structures with Types and Instances (similar to EMF)

# Case Study: VHA

# THREE ENGINEERING TOOLS

[Demuth et al. 2016]



```
context EplanCodeLink
```

```
if self.define.map.source = self.define then
    self.define.map.target.eplan.function.link.in = self.function
    and self.define.map.target.eplan.function.link.out <> self.function
endif
```

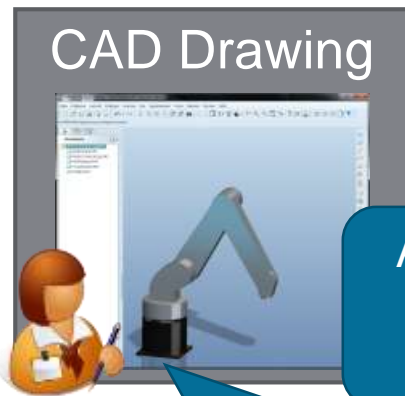
Risk judgement and evaluation (EN ISO 12100:2010)									
Project : Consistency use case									
Nr.	Danger	Consequence	Likelihood	frequency of exposure	DPH (Damage)	NP (#Personen)	HRN (risk)	Risc level	Remarks/ comments
1	Pinched in between conveyor belt	hand injury	1) before 2) after	Probably	Daily	fracture	1-2 persons	low, simple	conveyor motor
			1) before 2) after	nearly impossible	Daily	bruise	1-2 personen	negligible	



# *DesignSpace*

Re-thinking much of software  
engineering research

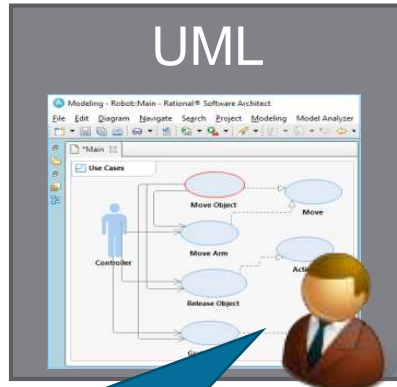
# *Collaborative Linking*



## CAD Drawing



Alice was asked to implement Feature x



## UML



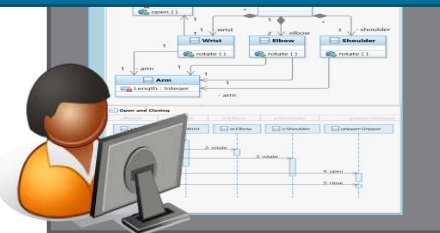
I implemented Feature x here but Bob had to make further changes also.



## Java

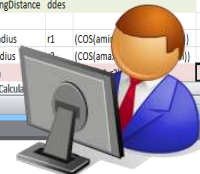


I used the arm length computation here and also let Emma know for the design



My arm length computation was used by Carol.

name	sign	formula	value	unit
1 alphaMin	amin		0.2	rad
2 alphaMax	amax		1.4	rad
3 betaMin	bmin		0.4	rad
4 betaMax	bmax		2.8	rad
5 desired GraspingDistance	dDes		2	m
6				
7				
8 maxNormedRadius	r1	(COS(amin	1.97005907	1
9 minNormedRadius	r2	(COS(ama	0.39716924	1
10 minArmLength			1.2754467	m
11 4 * r1 * LengthCalcul				



# LINKING

- Complementary Understandings
  - David knows that Alice implemented the use case
  - Alice knows where she implemented it
- Uncertainties [Ghabi-Egyed 2015]
  - The feature is implemented here but Bob made further changes
- Transitive Meanings [Kuang 2015, Ghabi-Egyed 2012]
  - The use case was modeled in this sequence diagram. This sequence is implemented in that code

Research on Language and Reasoning Engine for Complementary, Uncertain, Transitive Statements

# *Collaborative Awareness (The Changes of Others)*

# MULTI-USER ENGINEERING

DesignSpace

Public

Private

Artifact

Tool=Excel  
Engineer=PSMITH  
alphaMin=0.2  
alphaMax=1.4  
...  
minArmLength=1.27

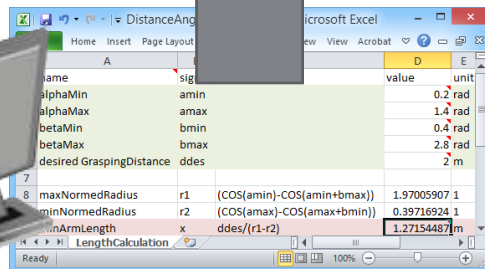
$t \leq s$

Artifact

Tool=ProEngineer  
Engineer=BANDER  
upper.armlength=1.5  
lower.armlength=1.5  
...

commit

Sync





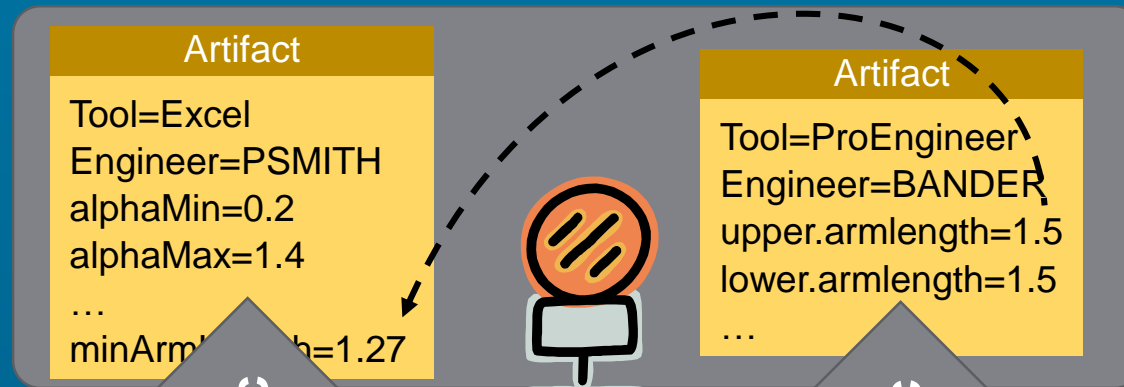
# MULTI-USER ENGINEERING

DesignSpace

Public

Private

$t \leq s$



Sync

name	sig	value	unit
alphaMin	amin	0.2	rad
alphaMax	amax	1.4	rad
betaMin	bmin	0.4	rad
betaMax	bmax	2.3	rad
desired GraspingDistance	ddes	2	m
maxNormedRadius	r1	(COS(amin)-COS(amin+bmax))	1.97005907 1
minNormedRadius	r2	(COS(amax)-COS(amax+bmin))	0.39716924 1
minArmLength	x	ddes/(r1-r2)	1.27154487 m

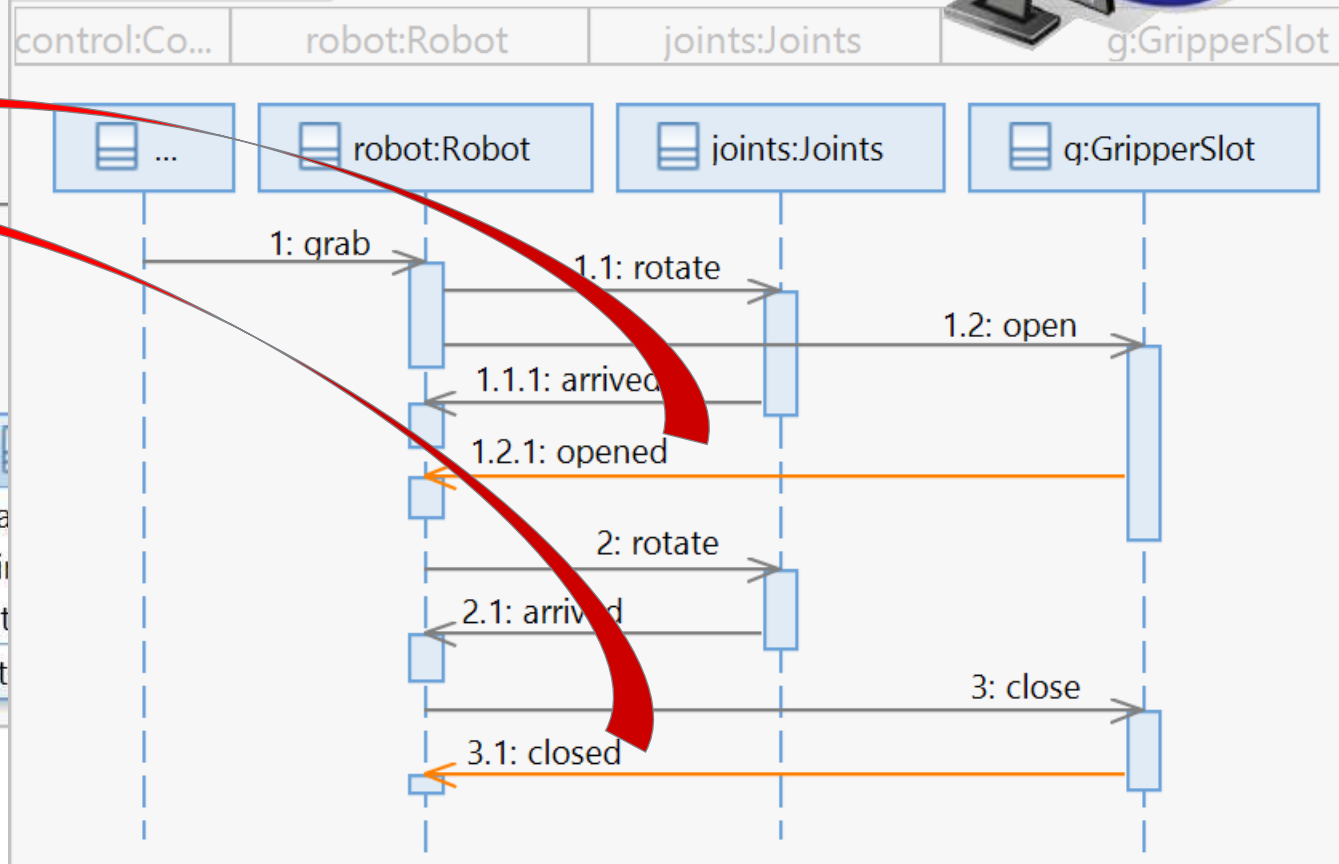
Microsoft Excel



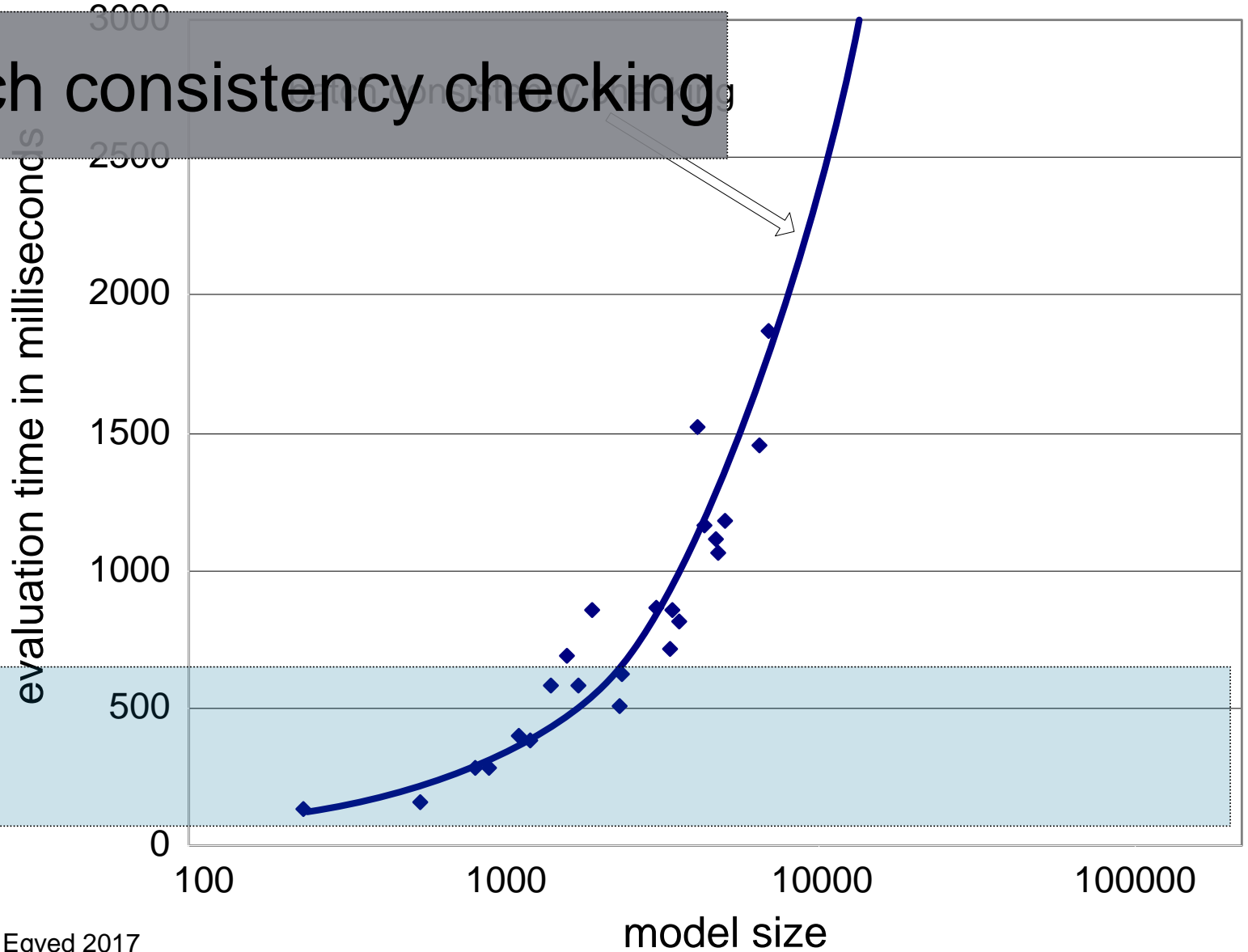
# *Collaborative Consistency*

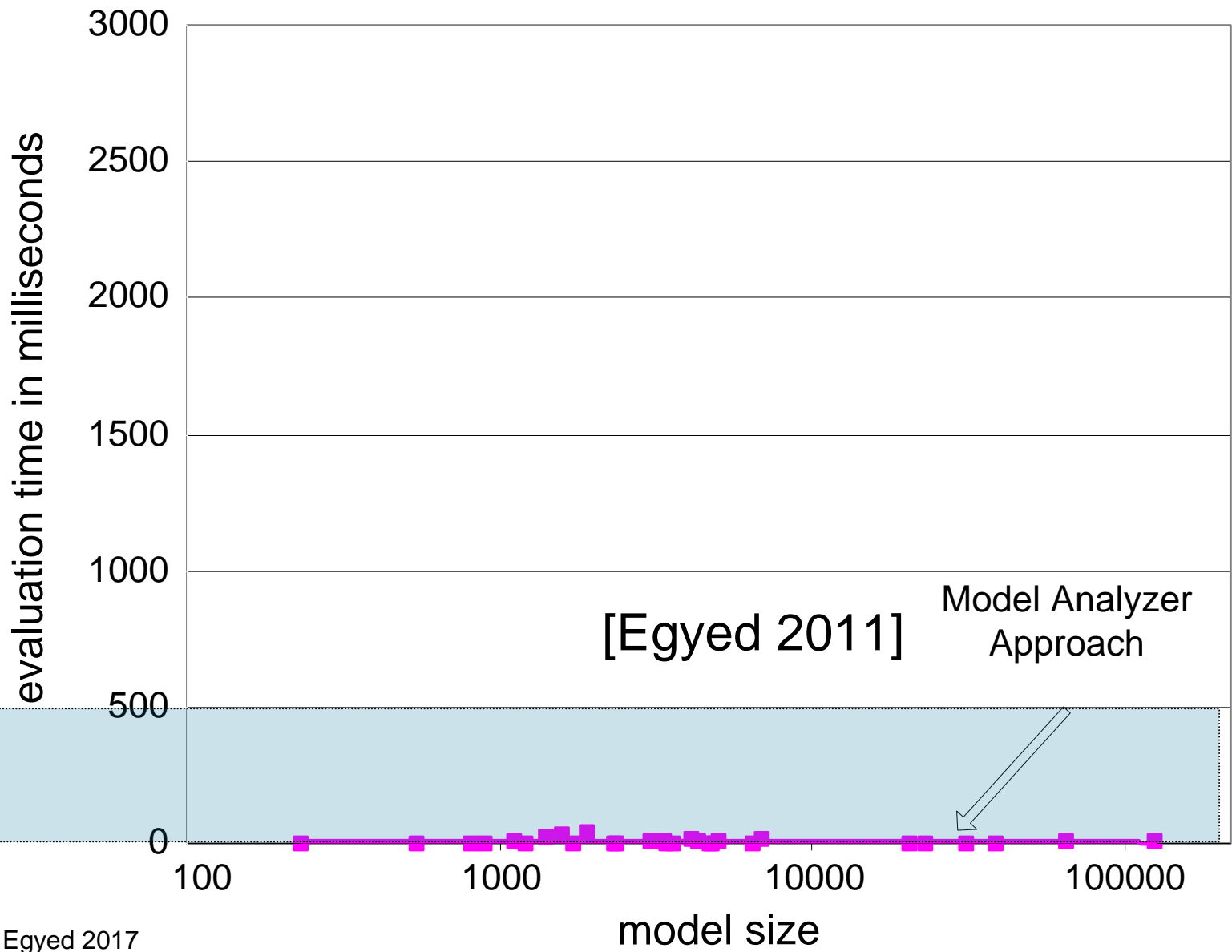
# CHANGE

```
Rule: self.receiveEvent.covered->  
forAll(represents.type.ownedOperation->  
exists(name=self.name))
```

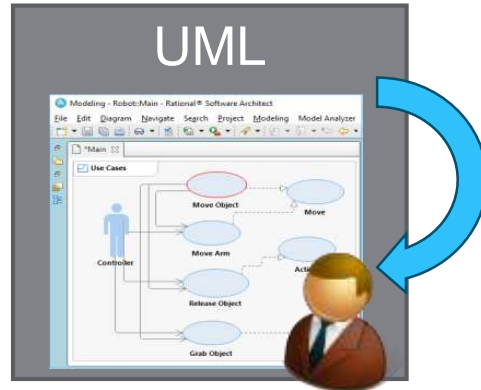


# Batch consistency checking

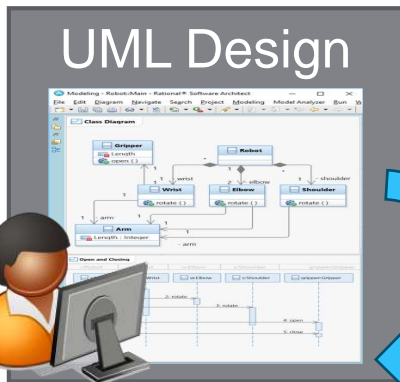




# CONSISTENCY CHECKING IN TOOLS



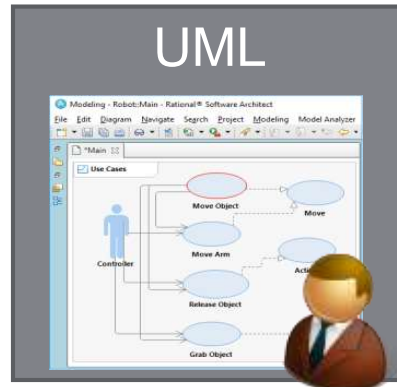
Local Consistency Checking



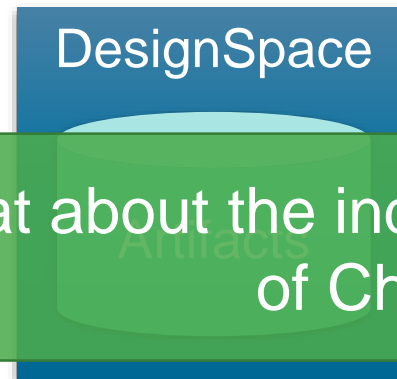
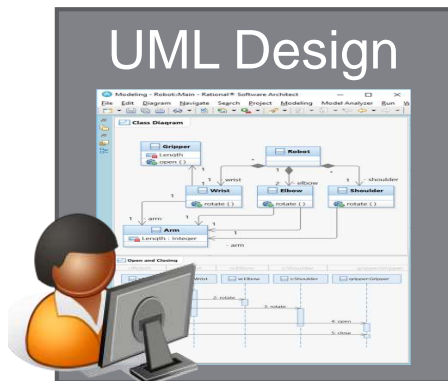
Computation

name	sign	formula	value	unit
1 alphaMin	amin		0.2	rad
2 alphaMax	amax		1.4	rad
3 betaMin	bmin		0.4	rad
4 betaMax	bmax		2.2	rad
5 distance			1.0725226	
6 angleLength			1.0725226	

Comprehensive Consistency Checking not straightforward - n-fold integration problem



**Comprehensive, Uniform  
Consistency Checking**



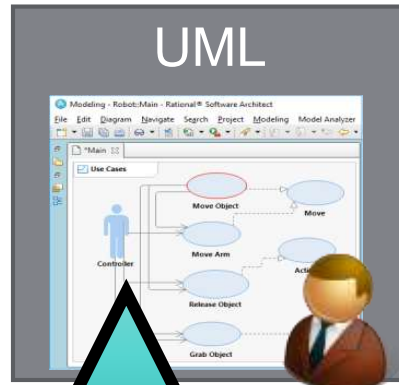
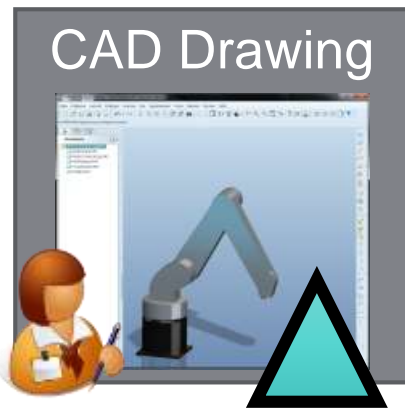
# Computation

	A	B	C	D	E
1	name	sign	formula	value	unit
2	alphaMin	amin		0.2	rad
3	alphaMax	amax		1.4	rad
4	betaMin	bmin		0.4	rad
5	betaMax	bmax		2.2	rad
6	desiredGraspPos	des		2	mm
7	desiredGraspDir	desDir		0.785398	rad
8	maxArmLength	maxLen		1.875	mm
9	minArmLength	minLen		1	mm
10	minAngle	minAng		0	rad

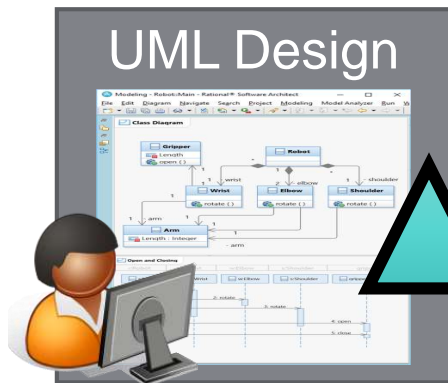
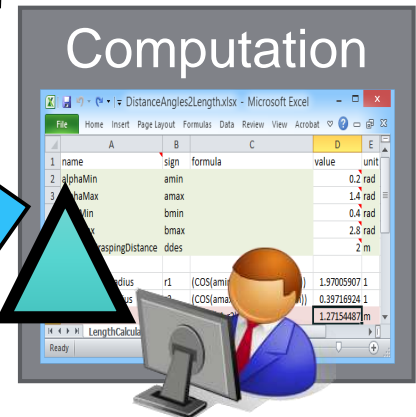
But what about the individual views? Impact of Changes?



[Demuth et al. 2016]

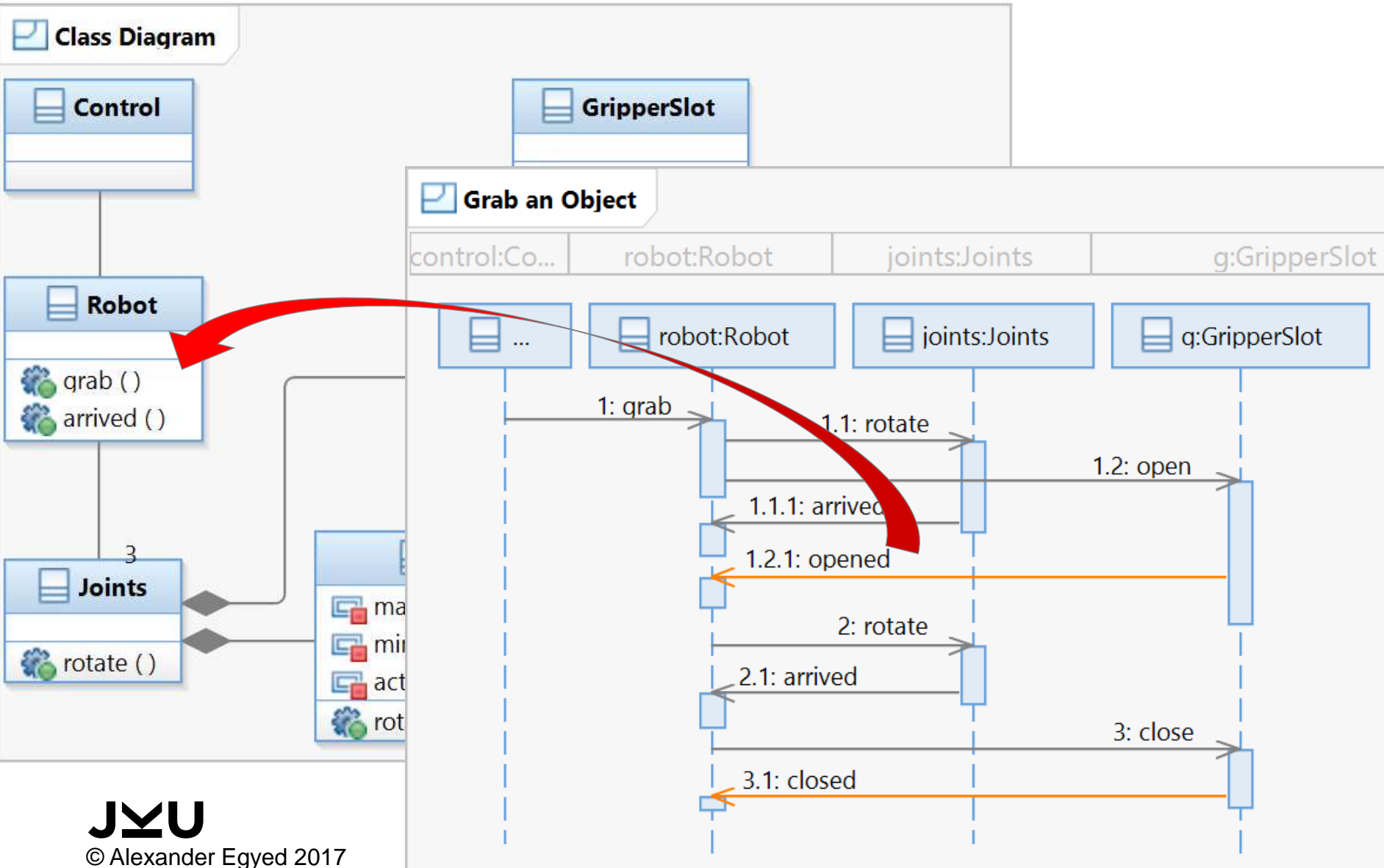


Comprehensive, Uniform  
Consistency Checking



# *Collaborative Repair and Change Propagation*

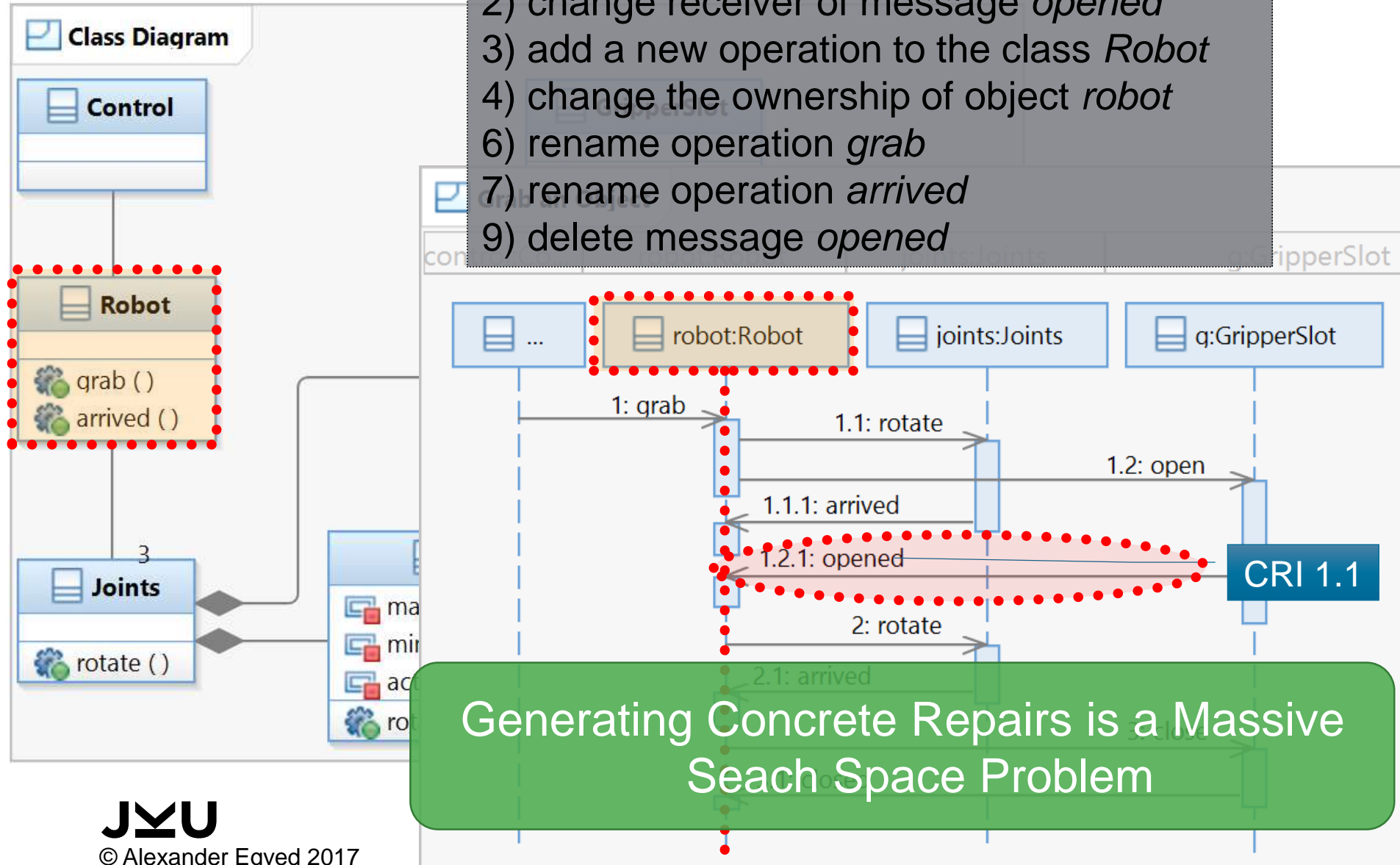
# REPAIR



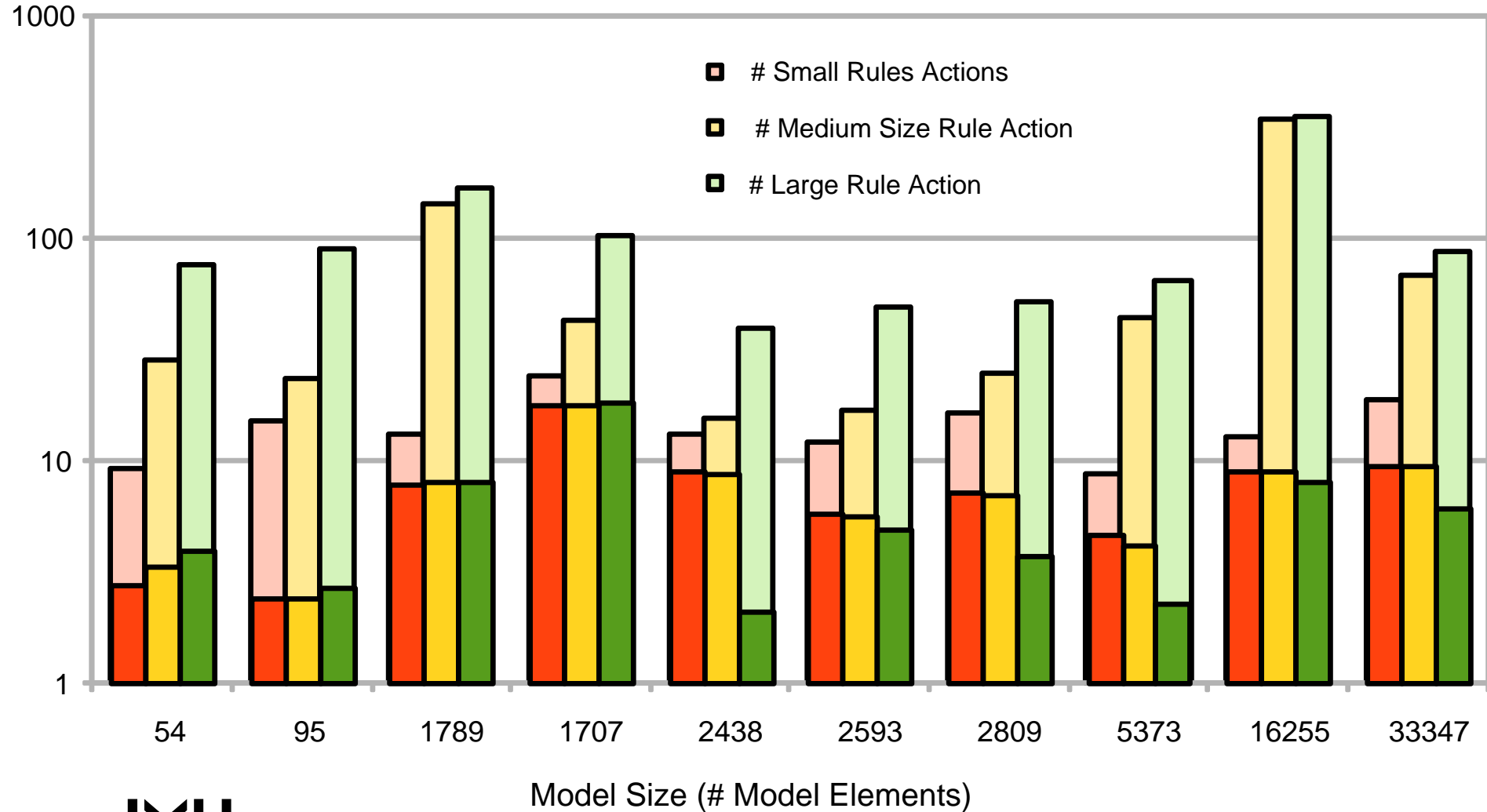
# REPAIR

**Locations** for Repair [Egyed 2007]:

- 1) rename message *opened*
- 2) change receiver of message *opened*
- 3) add a new operation to the class *Robot*
- 4) change the ownership of object *robot*
- 6) rename operation *grab*
- 7) rename operation *arrived*
- 9) delete message *opened*



# BENEFITS



# COLLABORATIVE REPAIR

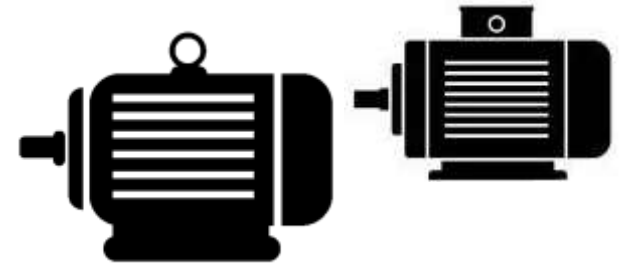
## ■ Individualized Repairs

- Considering private changes
- Filtering changes relevant to ones own work
- Complementary Changes
  - If Bob resolve the inconsistency this way then Carol's choices are reduced

# Reuse

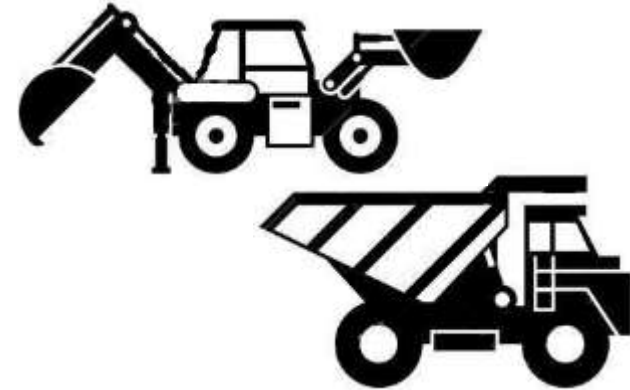


# VARIABILITY



■ *Variable System*: a system whose features are variable

- operating system and hardware platform
- different variants (demo, premium, ...)
- requirements differ for customers



Nobody plans this. This evolves  
over time!

Source: <https://www.ubuntu.com/>

# BUILDING THE FIRST ROBOT

## Robot 1

- CAD
- Computations (Matlab)
- EPlan
- Modelica
- SysML / UML
- Test/Use Case Scenarios
- Configuration Databases
- Commissioning Parameters
- Source Code
- ...

The collage shows various software tools used in the development of Robot 1:

- Microsoft Excel:** A spreadsheet titled 'DistanceAngles2Length.xlsx' containing a table of robot parameters and formulas.
- Eclipse IDE:** A Java source code file 'Robot.java' showing the implementation of a robot class with methods for distance calculation and rotation.
- Rational Software Architect:** SysML diagrams including a Use Case diagram with use cases like 'Move Object', 'Move Arm', 'Release Object', and 'Grab Object', and a Class Diagram showing the structure of the robot's components (Gripper, Wrist, Elbow, Shoulder, Arm).
- 3D CAD Model:** A 3D rendering of a robot arm.

name	sign	formula	value	unit
alphaMin	amin		0.2	rad
alphaMax	amax		1.4	rad
betaMin	bmin		0.4	rad
betaMax	bmax		2.8	rad
desired GraspingDistance	d-des		2	m
maxNormedRadius	r1	$(\cos(\text{amin}) - \cos(\text{amin} + \text{bmax}))$	1.97005907	1
minNormedRadius	r2	$(\cos(\text{amax}) - \cos(\text{amax} + \text{bmin}))$	0.39716924	1
minArmLength	x	$\text{d-des} / (\text{r1} - \text{r2})$	1.27154487	m

```
public class Robot {
    static double maxDistance = 2.0;

    public void grab(double distance) throws DistanceExceededException, PhiExceedException {
        if (distance > Robot.maxDistance) throw new DistanceExceededException();

        double d2 = Math.pow(distance, 2);
        double h2 = Math.pow(Robot.segmentHeight, 2);
        double c2 = Math.pow(Math.sqrt(d2+h2), 2);
        double l2 = Math.pow(Arm.Length, 2);

        double phi1 = Math.acos((c2/(2*c2+l2))*Math.acos((d2+c2-h2)/(2*d2*c2)));
        double phi2 = Math.acos((2*d2*c2)/(2*d2*c2+l2));
        double phi3 = Math.acos((c2/(2*c2+l2))*Math.acos((d2+h2-c2)/(2*d2*h2)));

        rotate(phi1, phi2, phi3);
    }

    public void rotate(double phi1, double phi2, double phi3) throws PhiExceedException {
        if (phi1 < 11.5 || phi1 > 80.2) throw new PhiExceededException();
        segment1.rotate(phi1);
        registerTimeout(phi1.segment1.currentPhi, 0.86666666);
    }
}

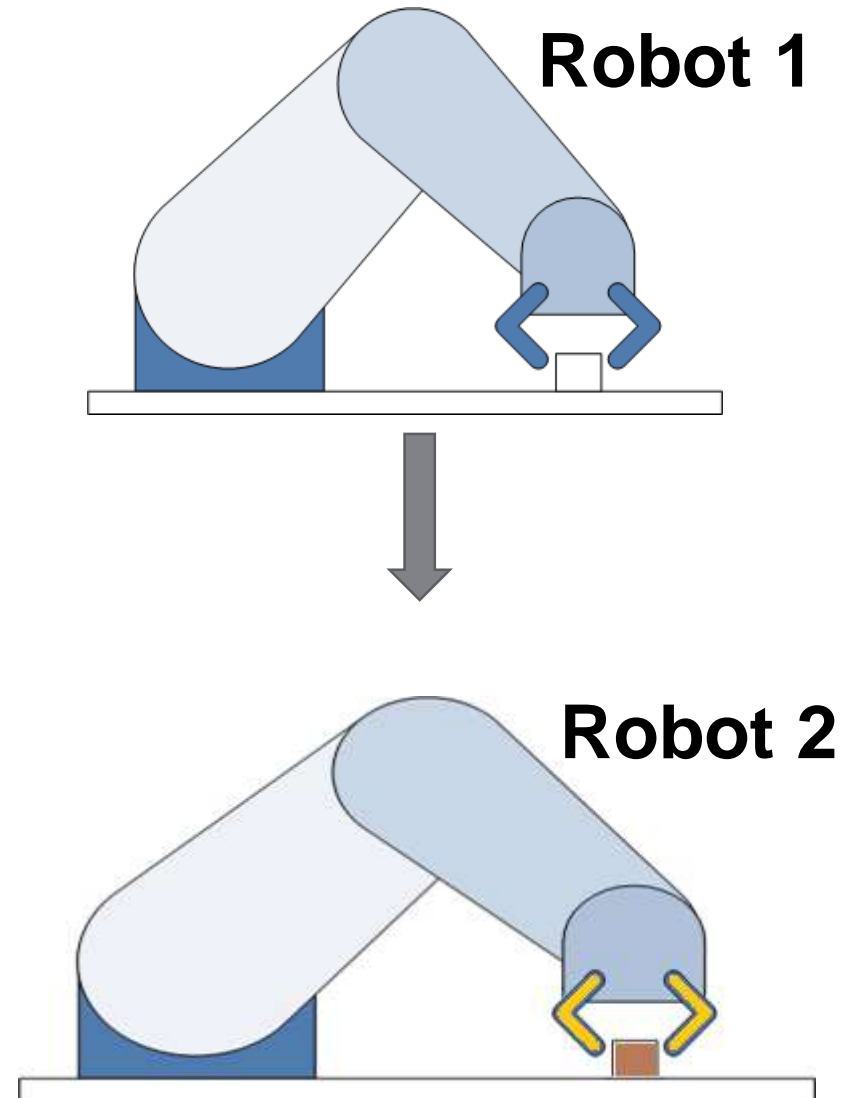
class Arm {
    double length = 1.5;
}

@Writable SmartInsert 2:31
```



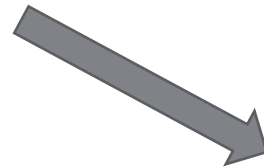
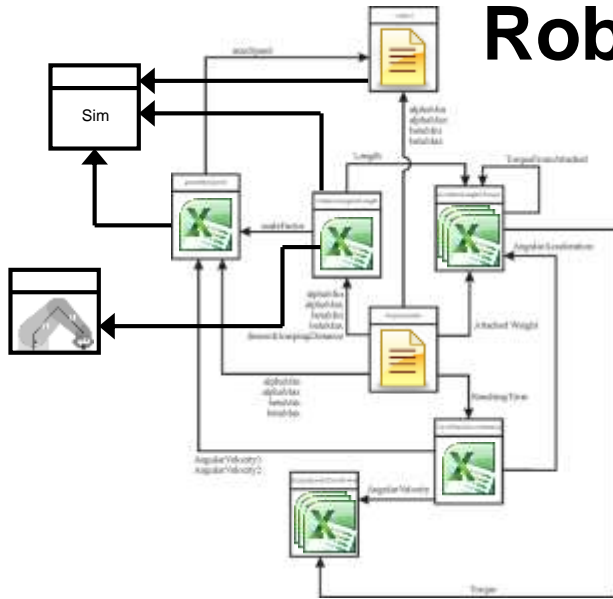
# BUILDING THE SECOND ROBOT

- The second robot is “kind of like” the first robot
- A max distance the robot needs to be able to reach is 2.8 meters
- Also it needs a different gripper



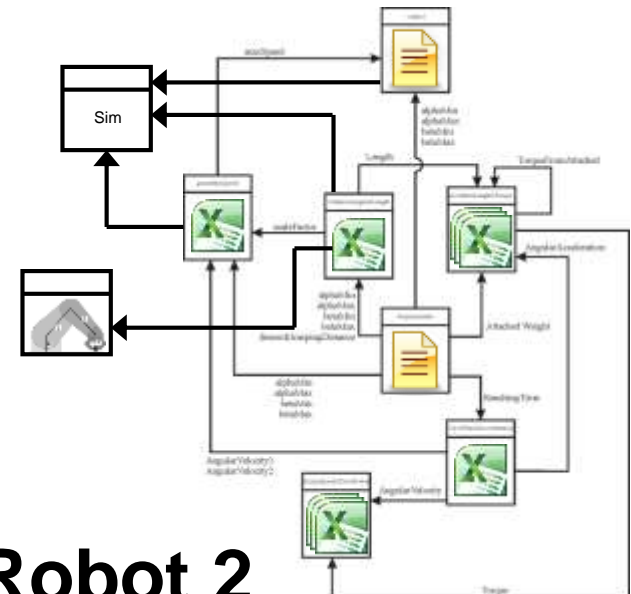
# CLONE AND OWN THE ENGINEERING ARTIFACTS OF THE FIRST ROBOT

## Robot 1



Clone and Own

## Robot 2



# MAKE CHANGES AS NECESSARY

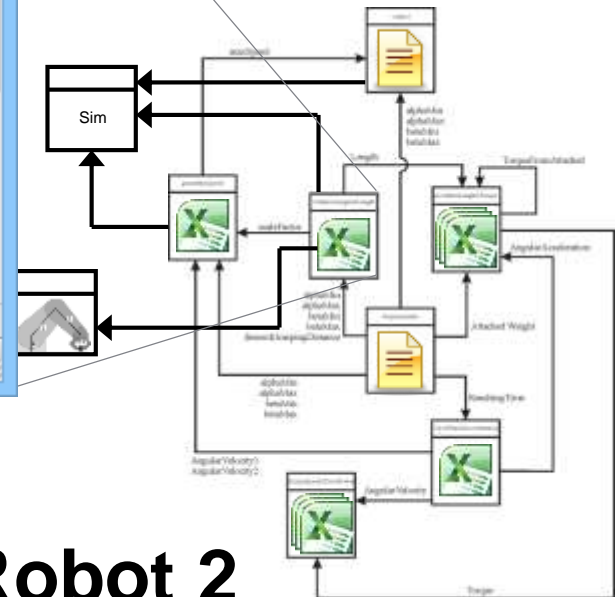
## Robot 1



DistanceAngles2Length.xlsx - Microsoft Excel

	A	B	C	D	E
1	name	sign	formula	value	unit
2	alphaMin	amin		0.2	rad
3	alphaMax	amax		1.4	rad
4	betaMin	bmin		0.4	rad
5	betaMax	bmax		2.8	rad
6	desired GraspingDistance	d-des		2.8	m
7					
8	maxNormedRadius	r1	$(\cos(\text{amin}) - \cos(\text{amin} + \text{bmax}))$	1.97005907	1
9	minNormedRadius	r2	$(\cos(\text{amax}) - \cos(\text{amax} + \text{bmin}))$	0.39716924	1
10	minArmLength	x	$\text{d-des} / (\text{r1} - \text{r2})$	1.78016282	m

Ready LengthCalculation 100%

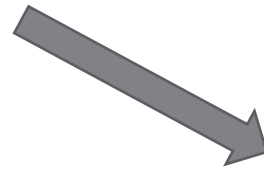
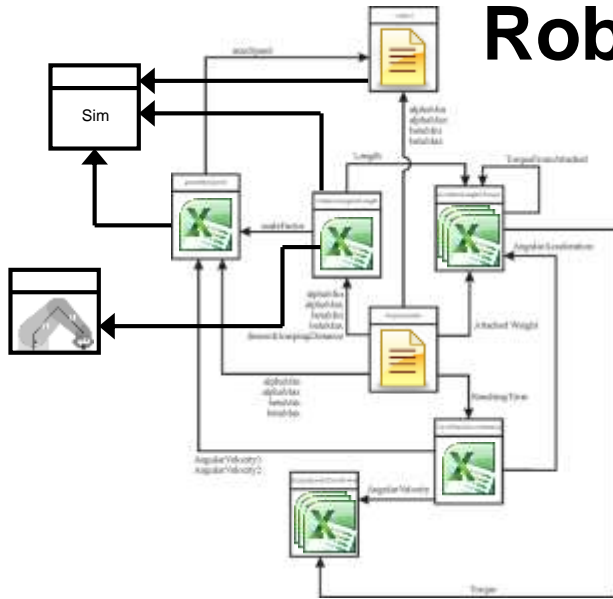


## Robot 2



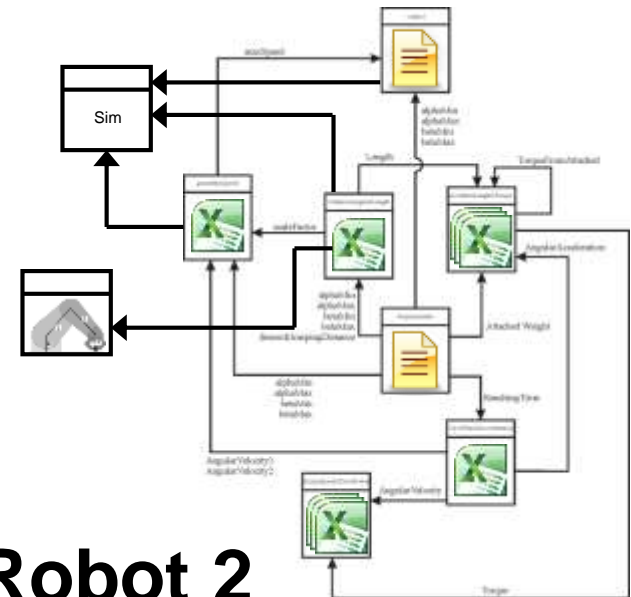
# MAKE CHANGES AS NECESSARY

## Robot 1



Clone and Own

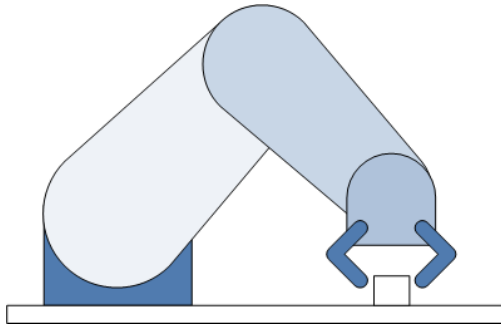
## Robot 2



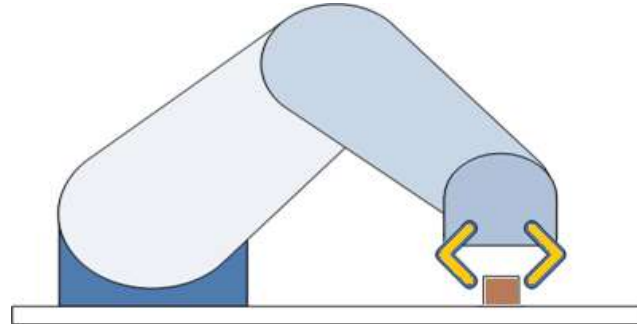


# BUILDING THE NEXT ROBOTS

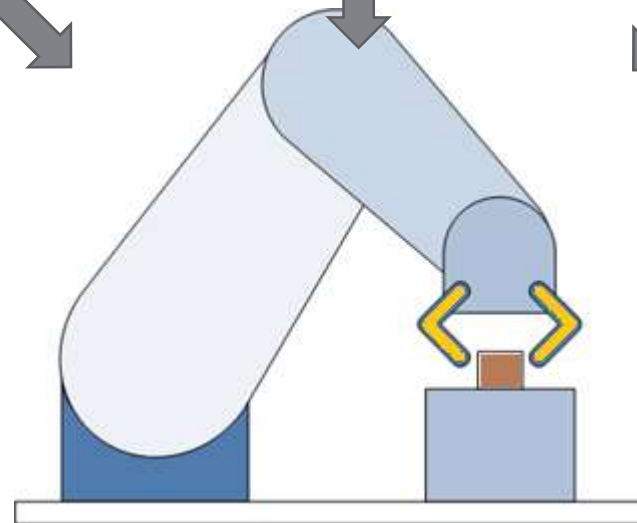
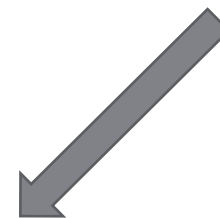
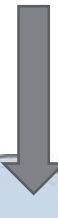
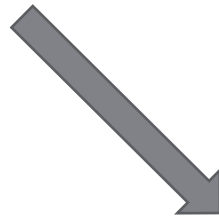
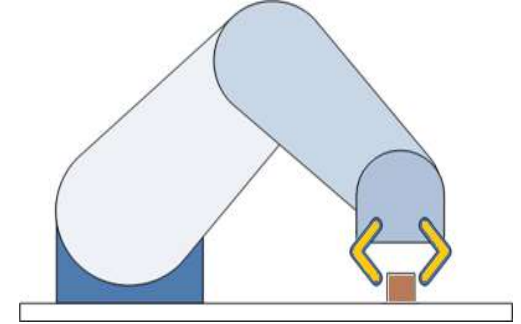
Robot 1



Robot 2



Robot 3

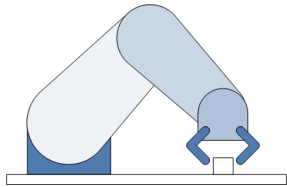


Robot n

# Solution

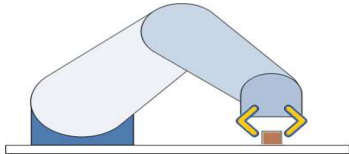
## Existing Products

### Robot 1



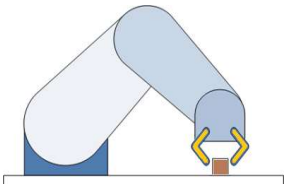
Short Upper Arm  
Short Lower Arm  
Blue Gripper

### Robot 2



Long Upper Arm  
Long Lower Arm  
Yellow Gripper

### Robot 3



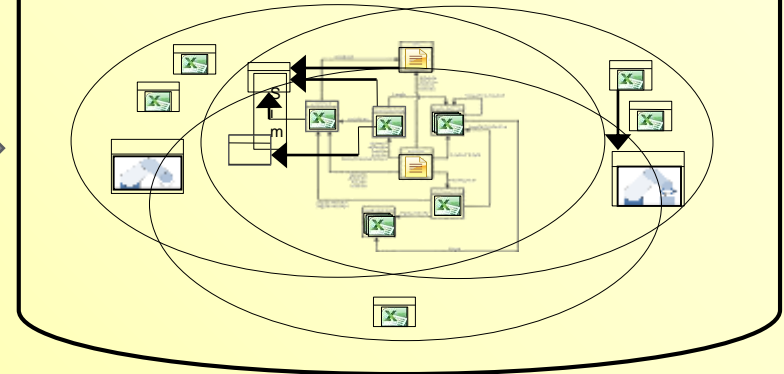
Short Upper Arm  
Short Lower Arm  
Yellow Gripper

# ECCO Tool

## Extraction

Compares products to identify commonalities and differences

## Fragment Repository

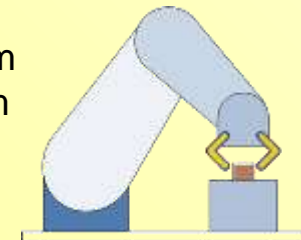


Commonalities and Differences are stored as Product Fragments

## Composition

## New Robot Composed from Existing Fragments

Short Upper Arm  
Long Lower Arm  
Yellow Gripper  
Pedestal **NEW!**



## Completion

```

public class Canvas extends JComponent implements MouseLi
protected List<Line> lines = new LinkedList<Line>();
Point start, end;
protected Line newLine = null;

public enum FigureTypes {
    NONE,
    LINE
};

public FigureTypes figureSelected = FigureTypes.NONE;

public void paintComponent(Graphics g) {
    super.paintComponent(g);
    g.setColor(Color.WHITE);
    g.fillRect(0, 0, getWidth(), getHeight());
    for (Line l : lines) {
        l.paint(g);
    }
}

public void wipe() {
    this.lines.clear();
    this.repaint();
}

```

```

public class Canvas extends JComponent implements MouseLi
protected List<Line> lines = new LinkedList<Line>();
Point start, end;
protected Line newLine = null;

public enum FigureTypes {
    NONE,
    LINE
};

public FigureTypes figureSelected = FigureTypes.NONE;
protected Color color = Color.BLACK;

public void paintComponent(Graphics g) {
    super.paintComponent(g);
    g.setColor(Color.WHITE);
    g.fillRect(0, 0, getWidth(), getHeight());
    for (Line l : lines) {
        l.paint(g);
    }
}

public void setColor(String colorString) {
    if (colorString.equals("Red"))
        color = Color.red;
    else if (colorString.equals("Green"))
        color = Color.green;
    else if (colorString.equals("Blue"))
        color = Color.blue;
    else
        color = Color.black;
}

```

```

public class Canvas extends JComponent implements MouseLi
protected List<Line> lines = new LinkedList<Line>();
protected List<BasicRectangle> rects = new LinkedList
Point start, end;
protected Line newLine = null;
protected BasicRectangle newRect = null;

public enum FigureTypes {
    NONE,
    LINE,
    RECT
};

public FigureTypes figureSelected = FigureTypes.NONE;
protected Color color = Color.BLACK;

public void paintComponent(Graphics g) {
    super.paintComponent(g);
    g.setColor(Color.WHITE);
    g.fillRect(0, 0, getWidth(), getHeight());
    for (Line l : lines) {
        l.paint(g);
    }
    for (BasicRectangle r : rects) {
        r.paint(g);
    }
}

public void setColor(String colorString) {
    if (colorString.equals("Red"))
        color = Color.red;
    else if (colorString.equals("Green"))
        color = Color.green;
    else if (colorString.equals("Blue"))
        color = Color.blue;
    else
        color = Color.black;
}

```

```

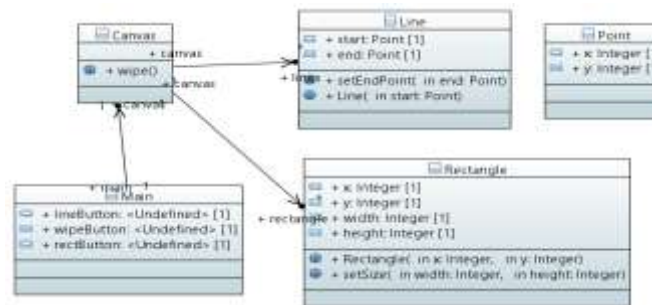
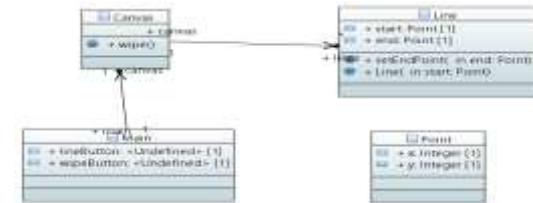
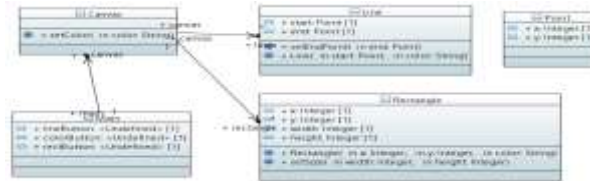
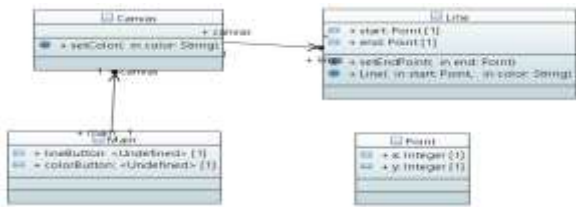
class Canvas {
    List<Line> lines;
    List<Rect> rects;
    void wipe() {
        this.lines.clear();
    } ...
}

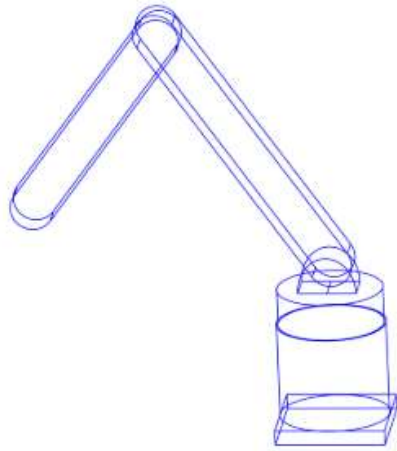
class Line {
    Line(Point start) {
        ...
    } ...
}

class Rect {
    Rect(Color c, int x, int y) {
        ...
    } ...
}

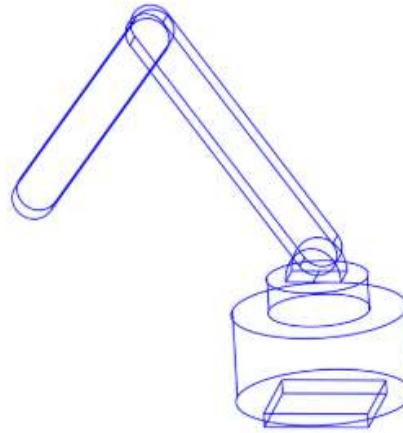
class Main extends JFrame{
    initContentPane() {
        toolPanel.add(lineButton);
        toolPanel.add(rectButton);
        toolPanel.add(wipeButton);
    } ...
}

```

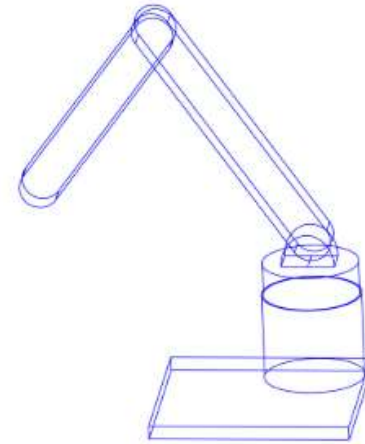




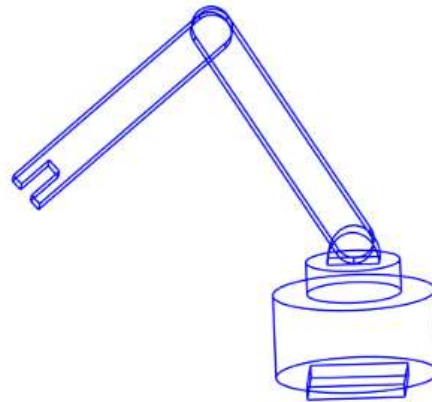
(a) Base Variant



(b) Bigger Base



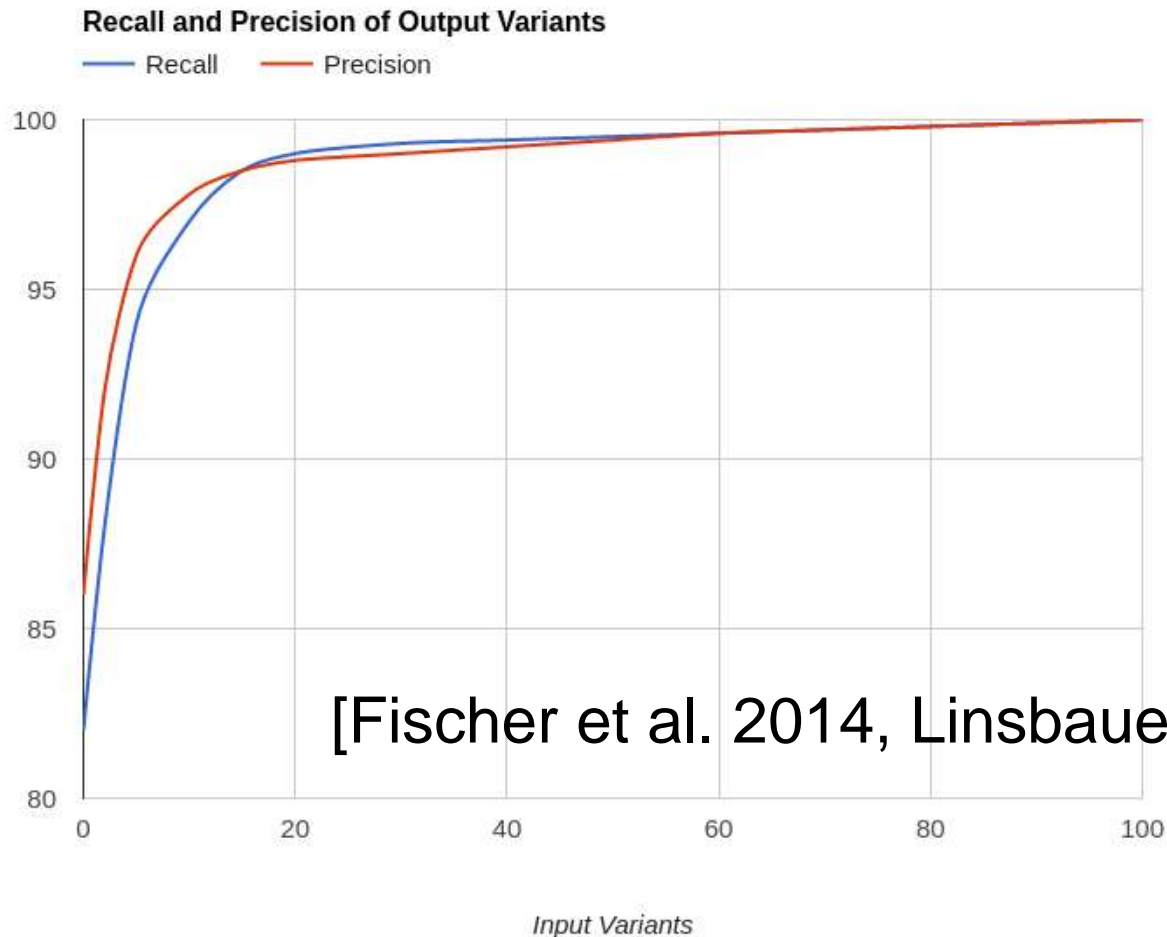
(c) Bigger Ground



(a) Claw Arm and Bigger Base

# COMPOSITION

- After only few input variants **precision** and **recall** are close to 100%
- 6 Systems: Source Code <344KLOC, <256 variants, <15 features



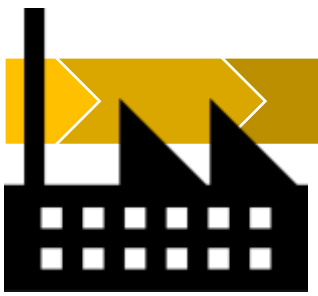
**Variant 1**



**Variant n**



**Product Life Cycle**



**Production Life Cycle**



**Operation Life Cycle**

Pro<sup>2</sup>Future



**JKU**  
JOHANNES KEPLER  
UNIVERSITÄT LINZ

**FWF** Der Wissenschaftsfonds.



**LEM** LINZ  
CENTER OF  
MECHATRONICS  
GMBH

