Universiteit Antwerpen

DesignSpace

Model Driven Engineering

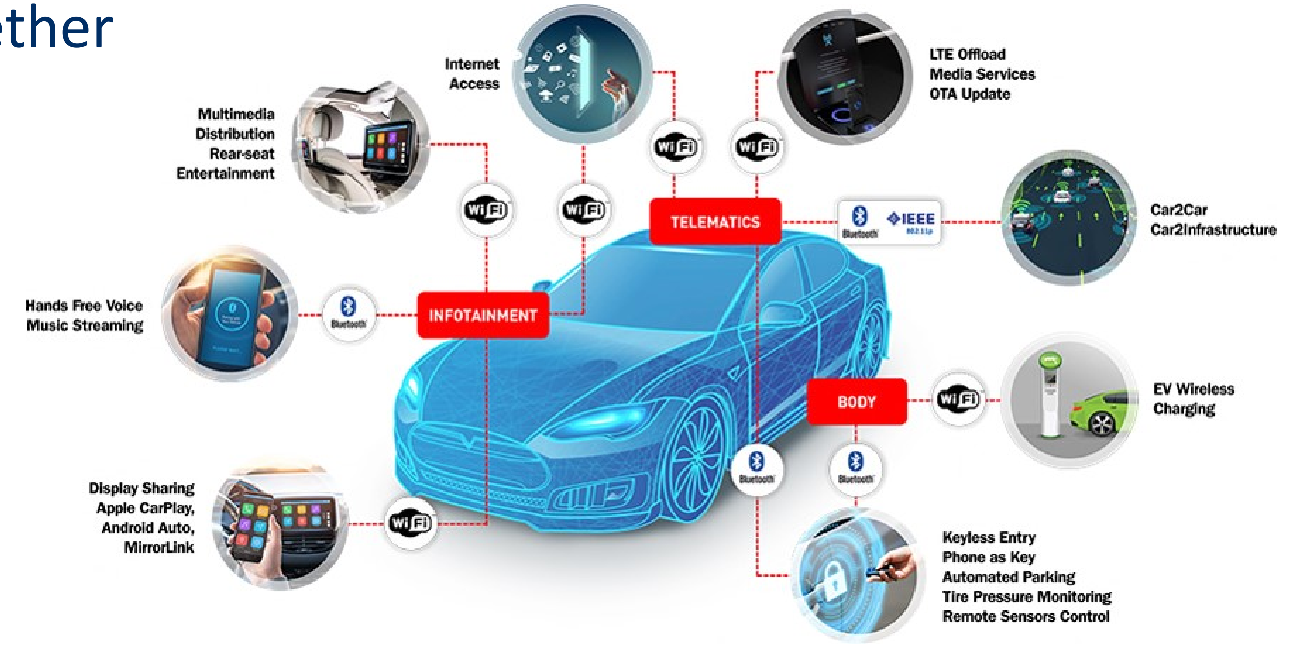
Thomas Van Onsem

Outline

Intro

- **Large Engineering Systems**
  - Often Software & Hardware in sync
  - Lots of different tools: CAD, Maple, MATLAB, IntelliJ…
  - Lots of different engineers: Software, Network, Mechanical, Electrical…

-> All need to work together



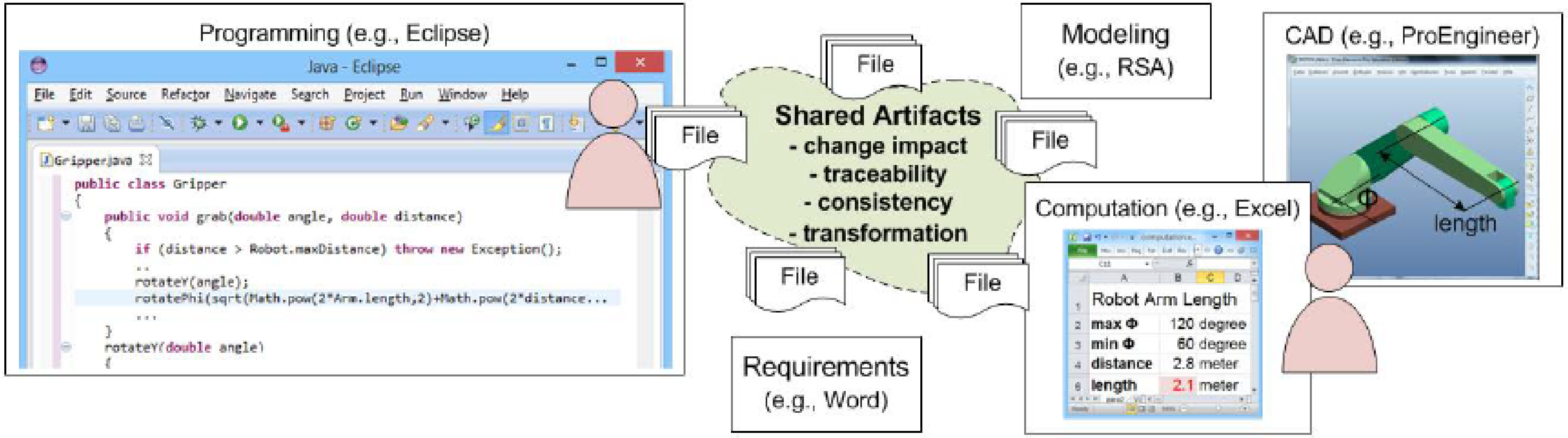
Intro

Conway’s Law:

“Any organization that designs a system will produce a design whose structure is a copy of the organization's communication structure.”

-> For a software module to function, multiple authors must communicate frequently with each other.

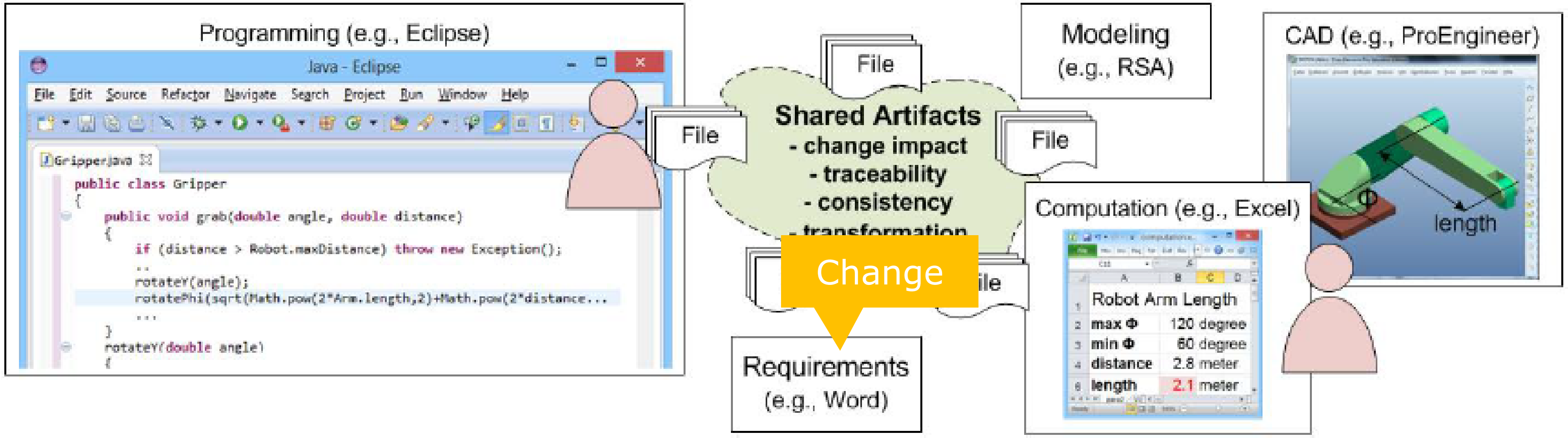
Problem



```
public class Gripper
{
    public void grab(double angle, double distance)
    {
        if (distance > Robot.maxDistance) throw new Exception();
        ..
        rotateY(angle);
        rotatePhi(sqrt(Math.pow(2*Arm.length,2)+Math.pow(2*distance...
        ...
    }
    rotateY(double angle)
    {
```

	A	B	C
1	Robot Arm Length		
2	max Φ	120	degree
3	min Φ	60	degree
4	distance	2.8	meter
6	length	2.1	meter

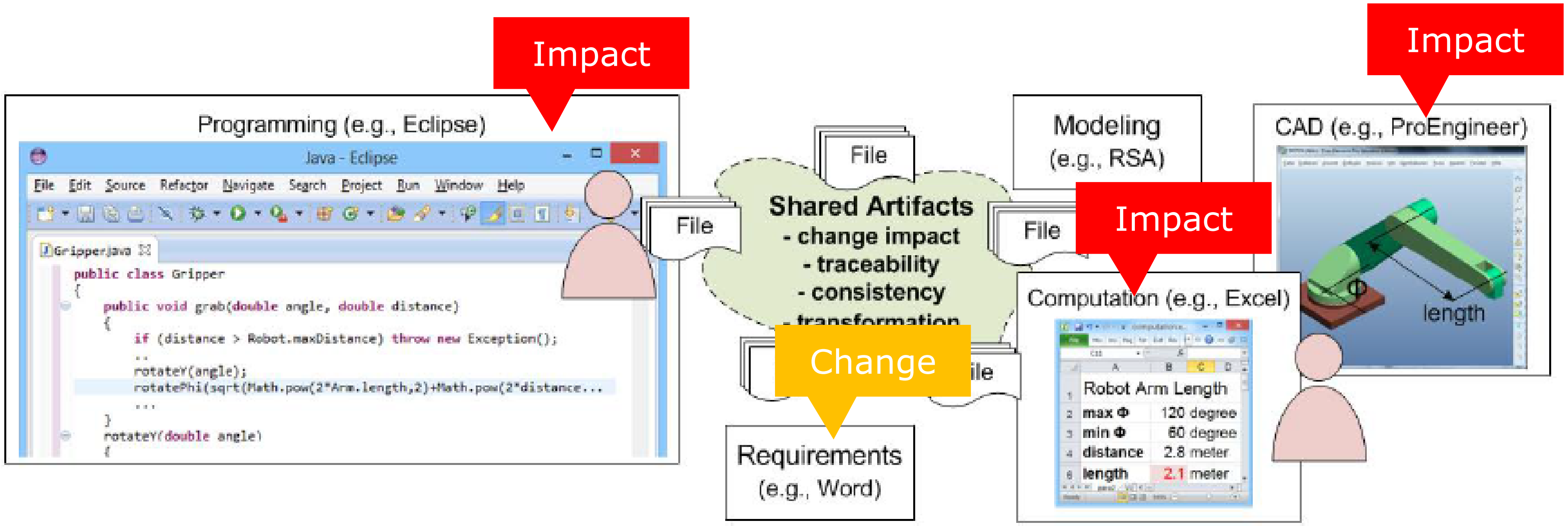
Problem



```
public class Gripper
{
    public void grab(double angle, double distance)
    {
        if (distance > Robot.maxDistance) throw new Exception();
        ..
        rotateY(angle);
        rotatePhi(sqrt(Math.pow(2*Arm.length,2)+Math.pow(2*distance...
        ...
    }
    rotateY(double angle)
    {
```

	A	B	C	D
1	Robot Arm Length			
2	max Φ	120	degree	
3	min Φ	60	degree	
4	distance	2.8	meter	
5	length	2.1	meter	

Problem



```
public class Gripper
{
    public void grab(double angle, double distance)
    {
        if (distance > Robot.maxDistance) throw new Exception();
        ..
        rotateY(angle);
        rotatePhi(sqrt(Math.pow(2*Arm.length,2)+Math.pow(2*distance...
        ...
    }
    rotateY(double angle)
    {
```

	A	B
1	Robot Arm Length	
2	max Φ	120 degree
3	min Φ	60 degree
4	distance	2.8 meter
5	length	2.1 meter

Problem

- **Programs used in development: CAD, Eclipse, Matlab, Excel…**
  - All single user platforms
  - No collaboration
  - But… often inter-dependencies between them
- **When changes are applied:**
  - Manually ensure consistency across artifacts
  - Error prone
  - Different interpretation and knowledge levels

-> *Huge engineering budgets and fails*

-> *One of biggest engineering challenges*

Solution: DesignSpace

- **Purpose:**
  - Lets engineers share knowledge and record inter-dependencies
  - Across different tools
  - Enables traceability between artifacts
  - Automatically checks consistency
- **Cloud based & Git-like workflow**
- **3 Services**
  - Data
  - Engineering
  - Collaboration

DesignSpace: Data Services

- **How to get artifacts from engineering tools?**
  - Tools propagate artifacts to DesignSpace (DesignSpace doesn't search for them)

  -> Tool specific adapter needed that gets artifacts from tools and posts to DesignSpace RESTful interface

- **What if tool does not support live sync with adapter?**

  -> Periodic parsing of local files

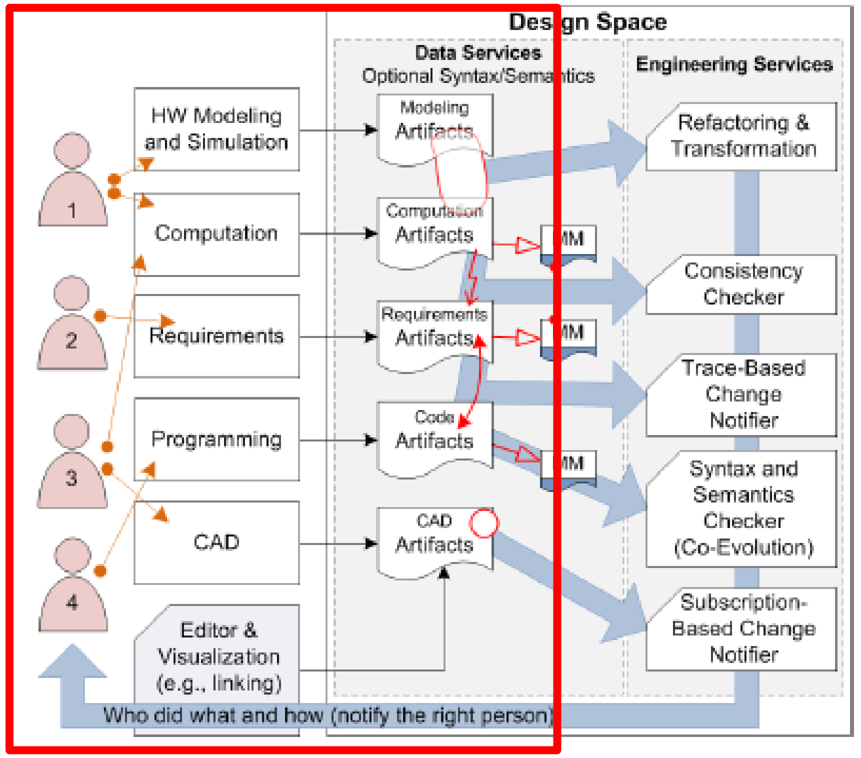
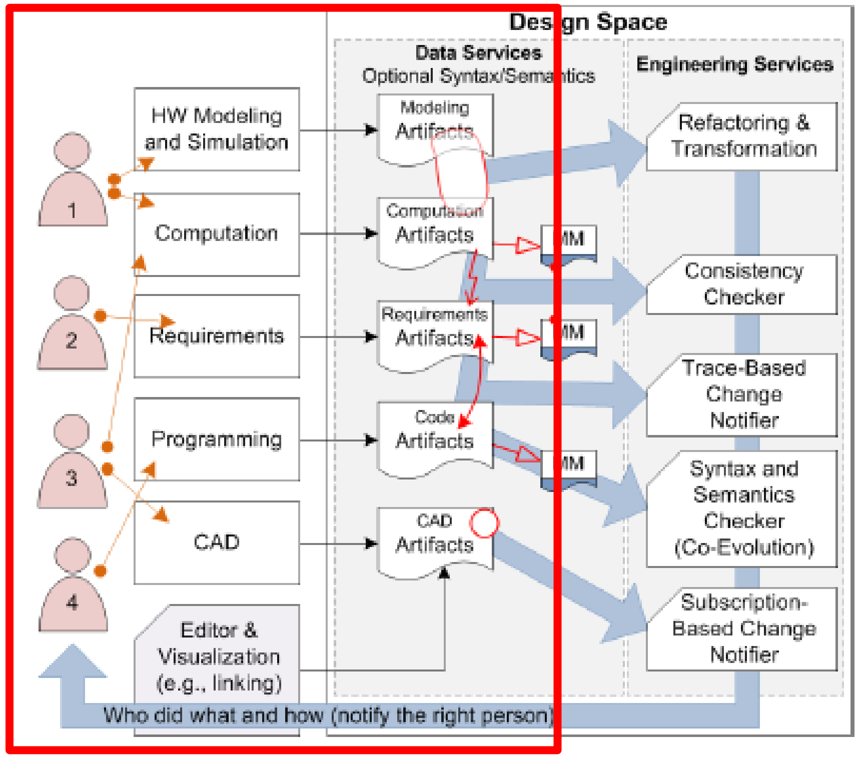

Design Space
Data Services
Optional Syntax/Semantics
Engineering Services
HW Modeling and Simulation
Computation
Requirements
Programming
CAD
Editor & Visualization (e.g., linking)
Modeling Artifacts
Computation Artifacts
Requirements Artifacts
Code Artifacts
CAD Artifacts
MM
MM
MM
1
2
3
4
Refactoring & Transformation
Consistency Checker
Trace-Based Change Notifier
Syntax and Semantics Checker (Co-Evolution)
Subscription-Based Change Notifier
Who did what and how (notify the right person)

Figure 2: DesignSpace Data and Engineering Services.

DesignSpace: Data Services

- **Artifact sharing:**
  - Complete: all artifacts are shared
  - Limited: engineers mark artifacts to share
- **Inside DesignSpace:**
  - Artifacts = Nodes with refs to other nodes
  - Connect nodes with visual editor to indicate dependency
- **Artifact metamodel weakly typed**

  -> Enables language/domain/metamodel evolution



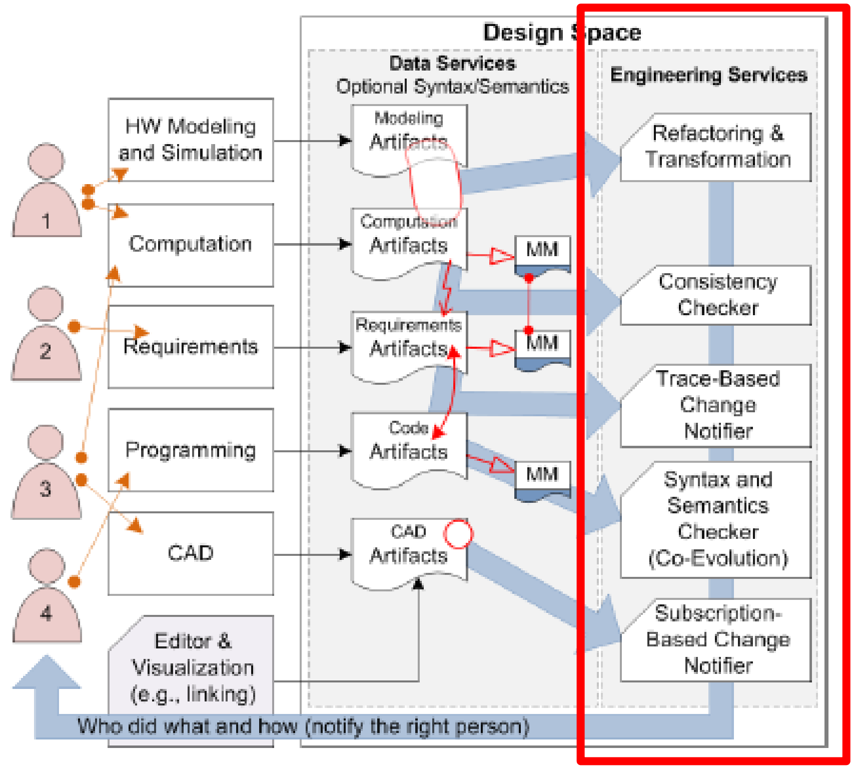
Design Space
Data Services
Optional Syntax/Semantics
Engineering Services
HW Modeling and Simulation
Computation
Requirements
Programming
CAD
Editor & Visualization (e.g., linking)
Modeling Artifacts
Computation Artifacts
Requirements Artifacts
Code Artifacts
CAD Artifacts
MM
MM
MM
1
2
3
4
Refactoring & Transformation
Consistency Checker
Trace-Based Change Notifier
Syntax and Semantics Checker (Co-Evolution)
Subscription-Based Change Notifier
Who did what and how (notify the right person)

Figure 2: DesignSpace Data and Engineering Services.

DesignSpace: Engineering Services

- **Consistency Checker**
  - Check predefined constraints
  - E.g.: Make sure dimensions of element are the same across artifacts
- **Change Notifier**
  - Subscription-Based: Informs subscribed engineers of changes to elements
  - Trace-Based: Informs engineers of changes to elements linked via trace link
    - E.g.: Trace link between specification and it's implementation

      -> Engineer that made implementation gets notified of specification changes



Design Space
Data Services
Optional Syntax/Semantics
Engineering Services
HW Modeling and Simulation
Computation
Requirements
Programming
CAD
Editor & Visualization (e.g., linking)
Modeling Artifacts
Computation Artifacts
Requirements Artifacts
Code Artifacts
CAD Artifacts
MM
MM
MM
Refactoring & Transformation
Consistency Checker
Trace-Based Change Notifier
Syntax and Semantics Checker (Co-Evolution)
Subscription-Based Change Notifier
1
2
3
4
Who did what and how (notify the right person)

Figure 2: DesignSpace Data and Engineering Services.

DesignSpace: Engineering Services

- **Refactoring & Transformations**
  - Refactoring across tools
  - Know where to apply refactoring based on traces
  - Bi-directional synchronization to apply in tools
- **Syntax & Semantics Checker**
  - Enforce conformance of artifacts to their language
  - Only useful if engineers diverse from tool default language

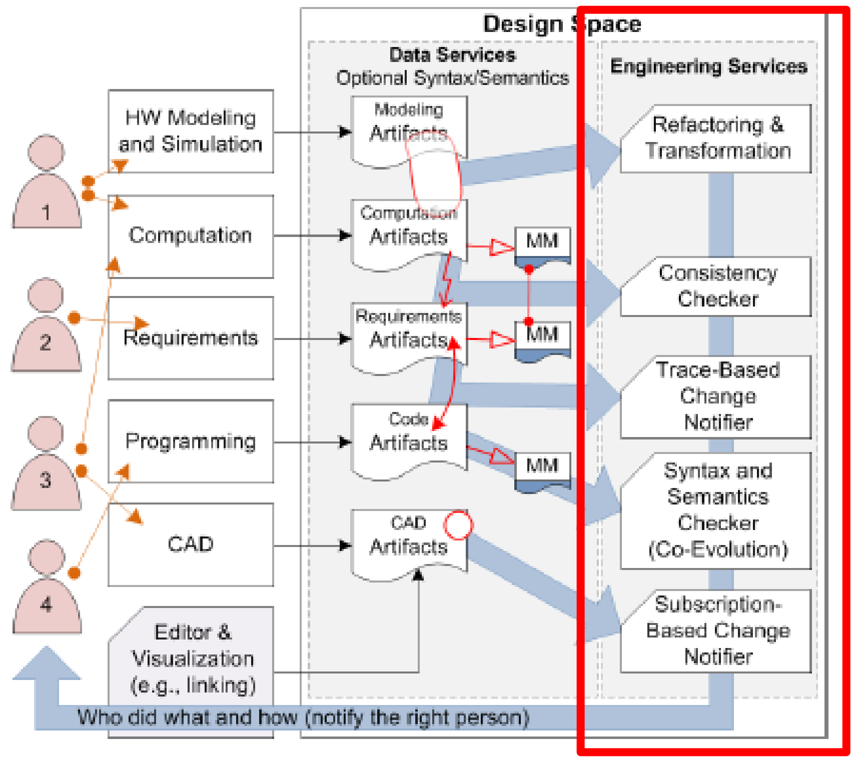
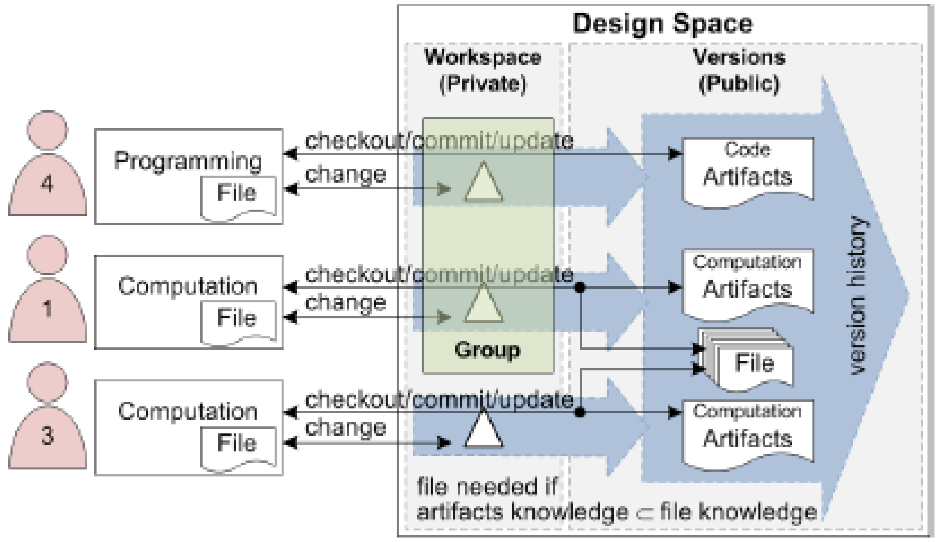

Design Space
Data Services
Optional Syntax/Semantics
Engineering Services
HW Modeling and Simulation
Computation
Requirements
Programming
CAD
Editor & Visualization (e.g., linking)
Modeling Artifacts
Computation Artifacts
Requirements Artifacts
Code Artifacts
CAD Artifacts
MM
MM
MM
1
2
3
4
Refactoring & Transformation
Consistency Checker
Trace-Based Change Notifier
Syntax and Semantics Checker (Co-Evolution)
Subscription-Based Change Notifier
Who did what and how (notify the right person)

Figure 2: DesignSpace Data and Engineering Services.

DesignSpace: Collaboration Services

- **Different workspaces:**
  - 1 public workspace
  - Many private workspaces
  - Everyone can choose where to work
  - Workspace groups to share without using public
- **Versioning system:**
  - Enables change notification
  - Enables change propagation
- **Inform engineers of cross-artifact inconsistencies caused by commit**



DesignSpace: Summary

- **Current Project State:**
  - Core services implemented
  - Adaptors for Excel, ProEngineer, IBM Software Architect
- **Why?**
  - It’s an all-in-one system: Collaboration, (Meta)Modelling, Consistency, Traceability and Evolution
  - Can be used to improve project development:
    - Avoid change ripples
    - Better program organization
    - Modelling software: Make change to model -> propagate change to code