Model Synchronization in a Joint-Cognitive Paradigm

Mathsig Presentation

7th of December 2022

Siyuan Ji

Loughborough University

OMG Document Number mathsig/2022-12-02

Context & Motivation

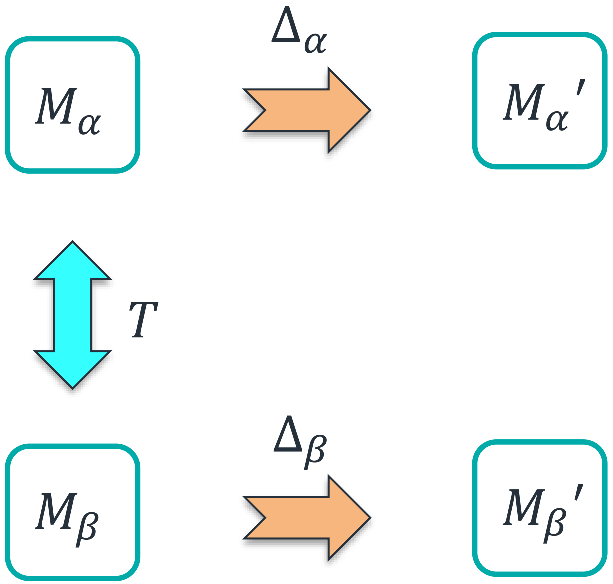
- Some challenges observed in practicing MBSE methodologies
  - How to maintain consistency between models?
  - How can we establish traceability without messing up the model complexity?
  - How do we handle change impacts in models?

- Our solution: **Model Synchronisation** with Structure Preserving Transformations [1]

What is Model Synchronisation?

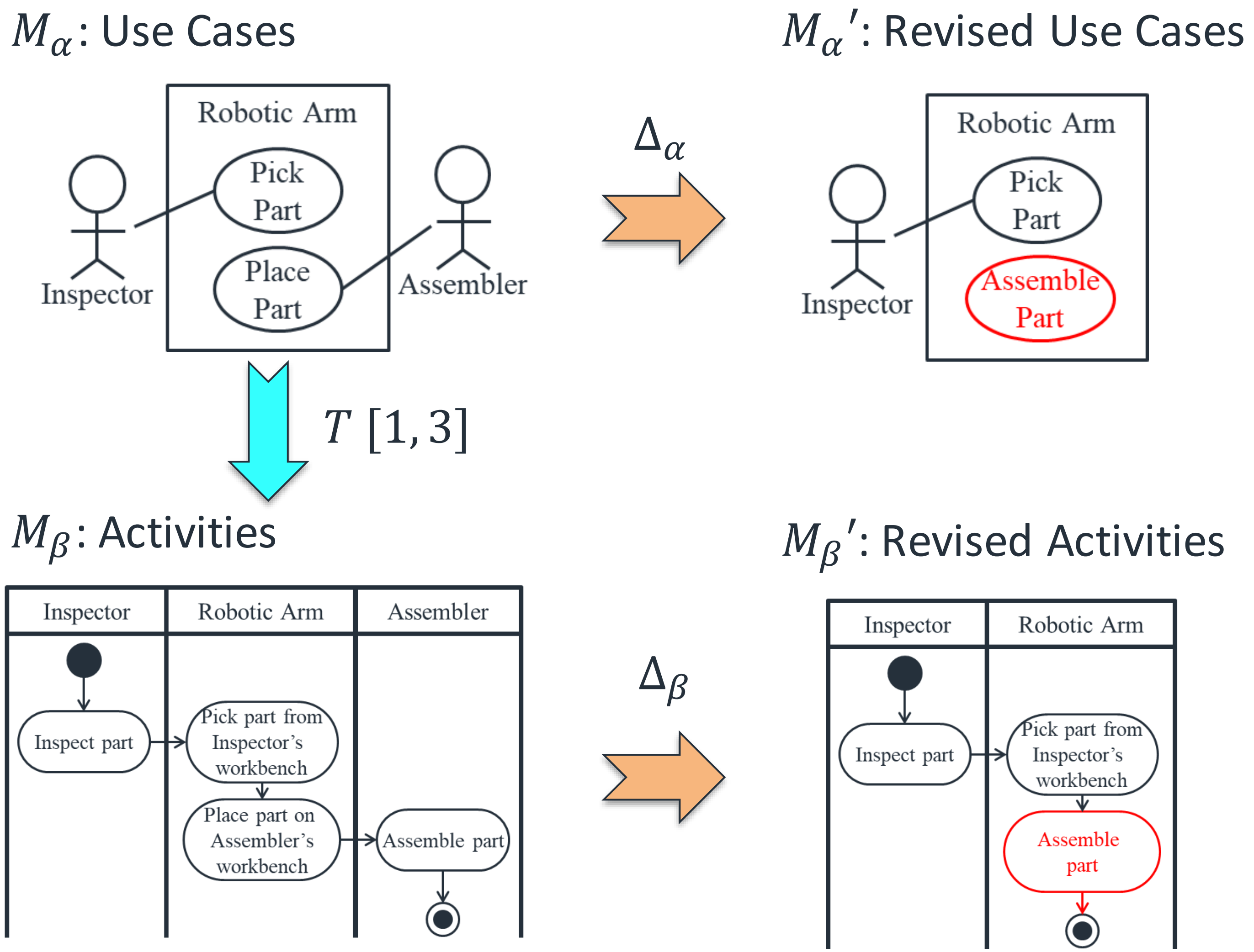
- Two **consistent** models, M_α and M_β
- A change made to a M_α, denoted as Δ_α
- Synchronisation means [2]

  a change Δ_β is mandatory to M_β in order to maintain the consistency between M_α' and M_β'
- Issues with manual synchronisation: labour intensive, error-prone, lacking generality & repeatability
- Proposal: Semi-automated, model transformations, T, following a **joint cognitive** approach



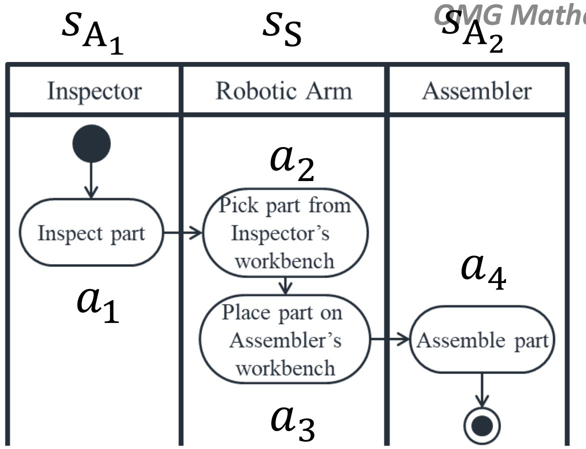
Robotic Arm Example

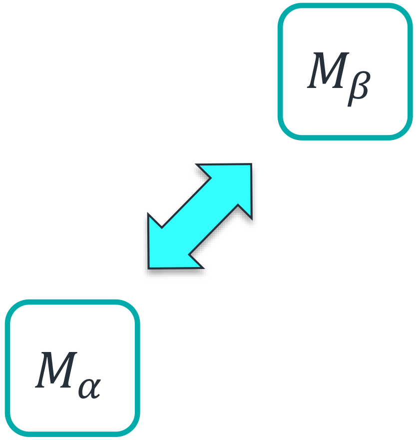
- Scenario:
  - Inspector checks the part, if OK
  - Robotic Arm then **picks** the part and **places** it on Assembler's workbench
  - Assembler finally assembles the part
- Change to functionality:
  - From Pick & Place, to Pick & Assemble
- Benefit:
  - Higher efficiency with reduced safety risks



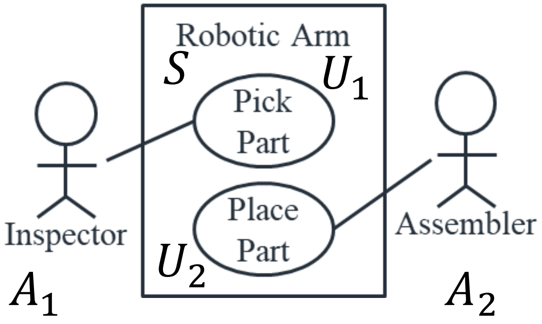
Matrix Representation

- Row/Column Headers: **model elements**
- Matrix element: **dependencies**
- $W_{i,j}$: **in-model dependencies,**
  e.g., A_1 is associated with U_1
- $Q_{i,j}$: **cross-model dependencies**
  e.g., A_1 is concordant with s_{A_1}
  e.g., U_1 is decomposed into a_2





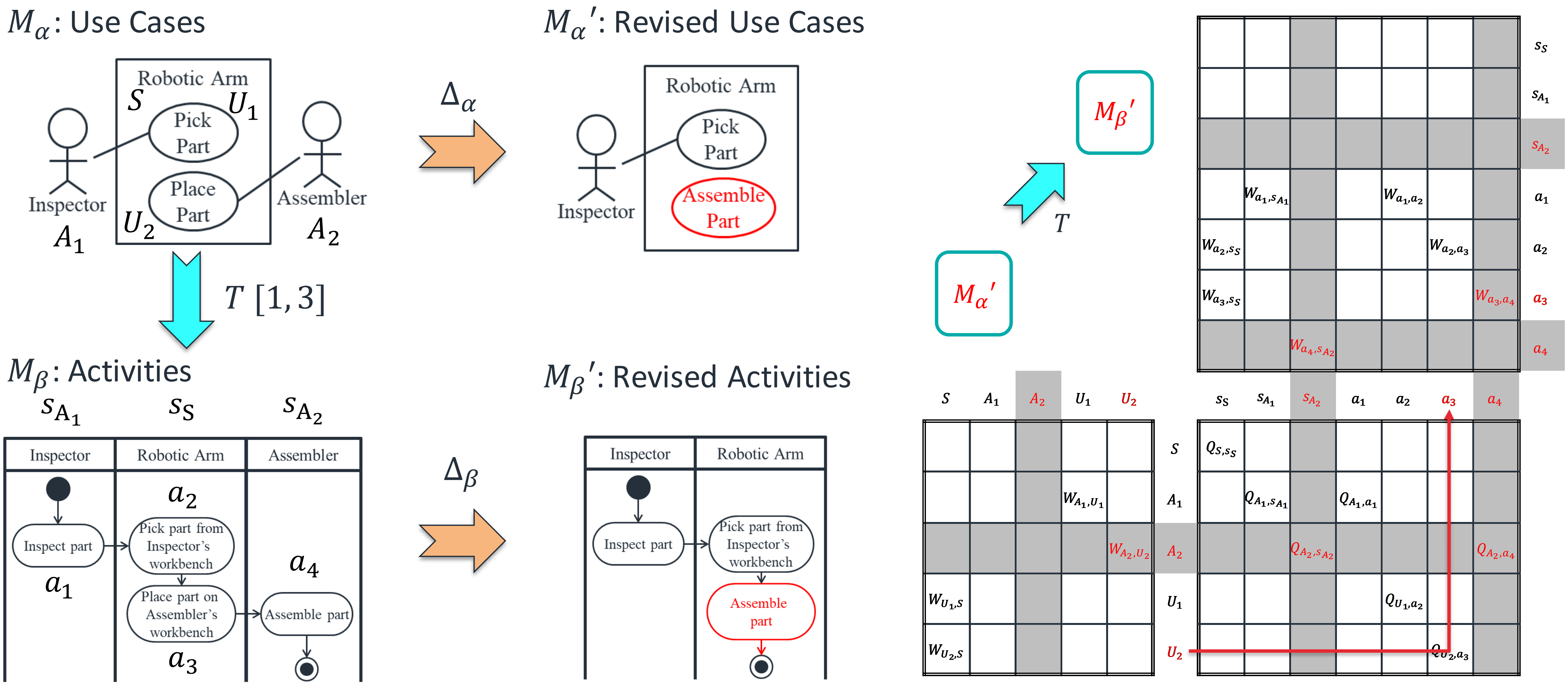
	s_S	s_{A_1}	s_{A_2}	a_1	a_2	a_3	a_4
s_S							
s_{A_1}							
s_{A_2}							
a_1		$W_{a_1,s_{A_1}}$			W_{a_1,a_2}		
a_2	W_{a_2,s_S}					W_{a_2,a_3}	
a_3	W_{a_3,s_S}						W_{a_3,a_4}
a_4			$W_{a_4,s_{A_2}}$				



	S	A_1	A_2	U_1	U_2
S					
A_1				W_{A_1,U_1}	
A_2					W_{A_2,U_2}
U_1	$W_{U_1,S}$				
U_2	$W_{U_2,S}$				

	s_S	s_{A_1}	s_{A_2}	a_1	a_2	a_3	a_4
S	Q_{S,s_S}						
A_1		$Q_{A_1,s_{A_1}}$		Q_{A_1,a_1}			
A_2			$Q_{A_2,s_{A_2}}$				Q_{A_2,a_4}
U_1					Q_{U_1,a_2}		
U_2						Q_{U_2,a_3}	

Synchronising Changes



Summary

- Users can follow any MBSE methodology to develop initially ***consistent*** diagrams/models
- The tool should ***maintain the consistency*** in the absence of explicitly modelled cross-model dependencies.
- This is achieved through **model synchronisation** enabled by **model transformation**
- A joint-cognitive approach. For example: the tool analyses elements being affected by a change while user decides how to make consistent changes to the affected elements.
- MapleMBSE – clearly has the capability for capturing in-model dependencies, but what about cross-model dependencies?

References

[1] Ji, S., Wilkinson, M., & Dickerson, C. E. (2022). Structure Preserving Transformations for Practical Model-based Systems Engineering. To appear in *8th IEEE International Symposium on Systems Engineering*, ISSE2022, *arXiv preprint* https://arxiv.org/abs/2209.07935

[2] Hettel, T., Lawley, M., & Raymond, K. (2008, July). Model synchronisation: Definitions for round-trip engineering. In *International Conference on Theory and Practice of Model Transformations* (pp. 31-45). Springer, Berlin, Heidelberg.

[3] Dickerson, C. E., & Ji, S. (2021). *Essential Architecture and Principles of Systems Engineering*. CRC Press.