

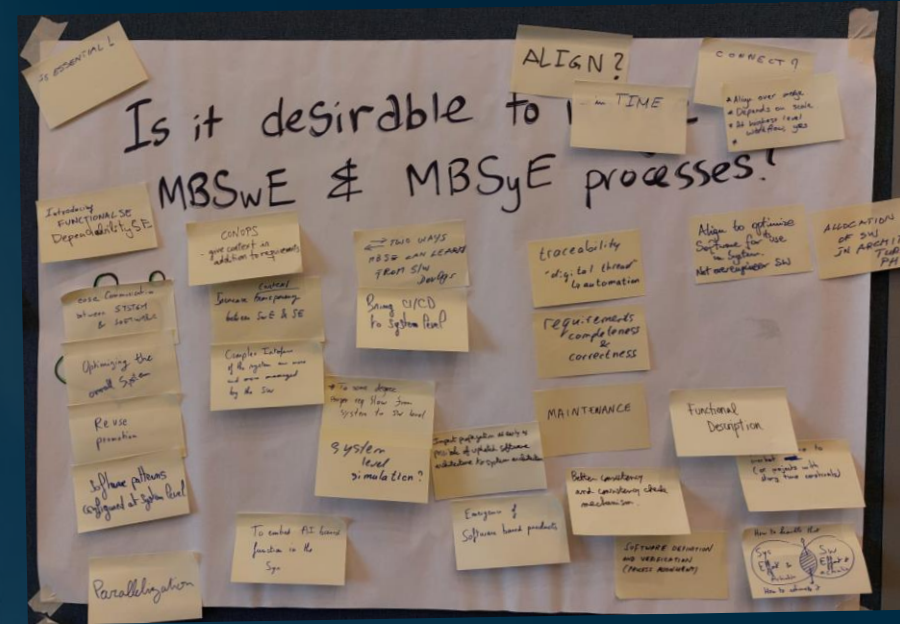
Software integration in MBSE

Group Presentation

MBSE 2023 Workshop
16/11/2023

align Is it desirable to ~~merge~~ MBSwE and MBSyE processes ?

- Question: “merge” processes → “align” processes!
- Bi-directional communication:
 - MBSwE can support MBSyE, e.g. earlier feedback from SW architectural decisions
 - MBSyE can provide clearer context (via models) to SW development team, e.g. ConOps, (HW/functional) architecture
- System may learn from SW development processes (DevOps)
- Functional behaviour description (as a model) important input to SW, dedicated task of functional system engineer?
- Model based SW design to be used at System level for: Performance/timing/dependability analysis
- Traceability via models instead of pure requirements based: Important for change management/maintenance

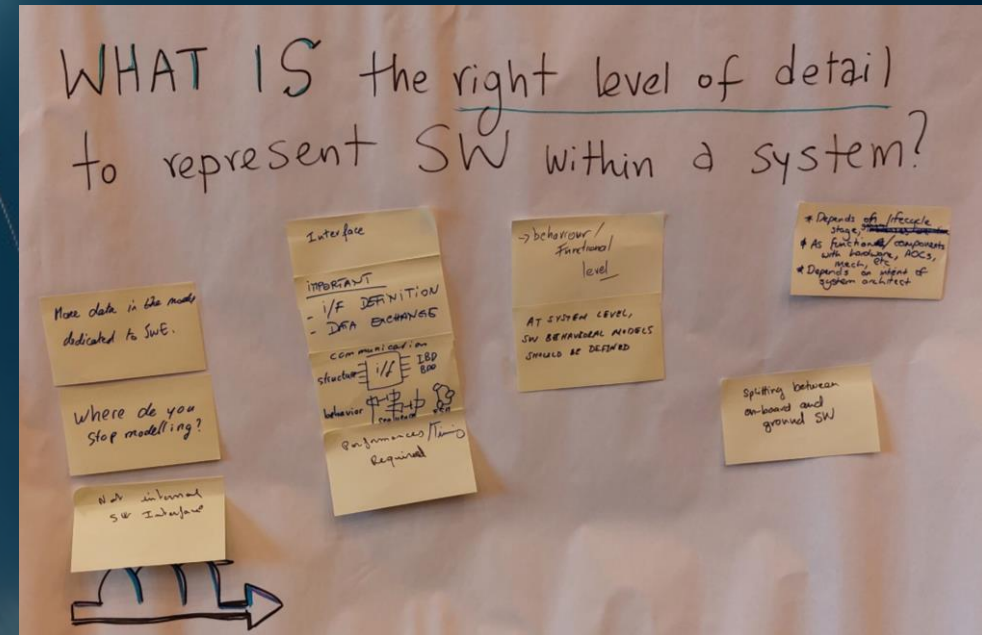


What is the right level of detail to represent software within a system ?

- Interfaces definition and data exchanges primary target
- Model more information (in particular **functional**) relevant for SwE
- Model performance/timing requirements for e.g. functional chains
- Think also about split between flight and ground software (is ground considered to be modelled and if so, into which detail?)

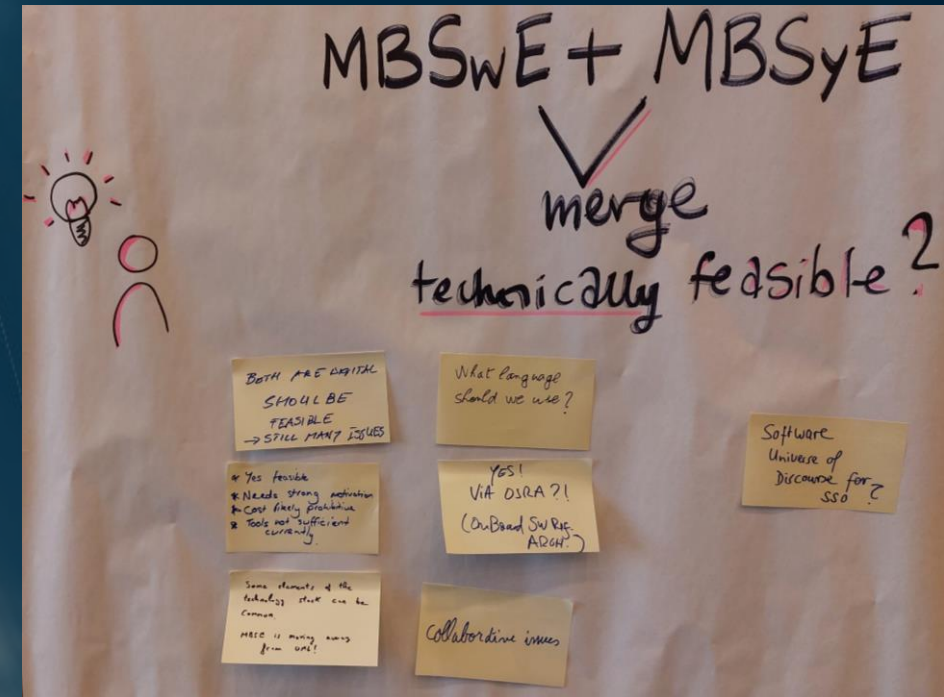
→ Level of detail depends on life cycle stage (and might depend on system architect intend)

→ Be careful, don't model too much! Where to stop?



Is it technically feasible ?

- Should be feasible, everything is digital 😊
 - ...but needs strong motivation, cost likely prohibitive
 - Current tools not sufficient
- Having a (common) language/ontology/mapping is not enough
 - Needs strong support from tools
- Issues with collaborative working on shared models



→ Still many issues to be solved

What would be the benefits and drawbacks ?

Benefits

- Early feedback and “model check” possible
- Model based software specification
- Ease impact propagation / change management
- SW from afterthought (“We are confident that the SW guys will manage to get all functions implemented”) towards parallel development process

Drawbacks

- System engineers are not SW engineers, two different disciplines and hence different modelling cultures
- Risk to go to much in SW at System level, might drive the solution to early
- “Unifying models” has risk of losing information

