A TMBR-based, Semiformal Method for Early Requirements Definition of Training Simulators

Władysław Sowul
Unmanned Aerial Vehicles Department
Military Aviation Works No. 2, part of the Polish Armaments Group
TU Delft Aerospace Engineering BSc with HPB in Robotics 2024
INCOSE Poland VP at Large, ASEP

Far-reaching goal

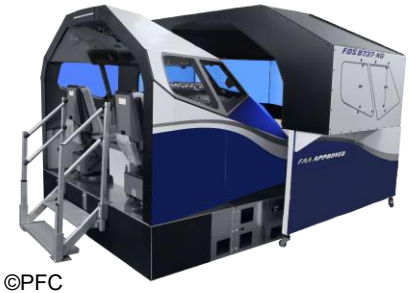
To exemplify benefits from establishing cross-system, semi-formal model transformations in SE workflow.

Agenda

- Training simulators context
- TMBR brief
- Partial method description
- Partial example
- Closing

Training simulator examples

+



Training simulators contractual context

The assumed stakeholder's principal need:

The stakeholders need the training simulator, in the simulation mode,

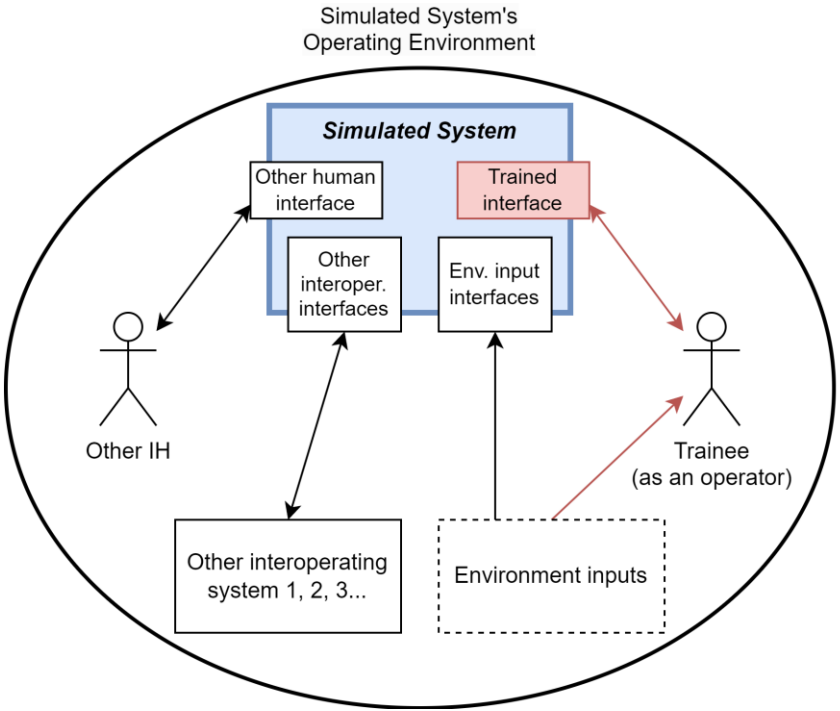
to provide the trainee with an interface of behavior similar,
to the furthest extent possible,
to the interfaces provided by the simulated system in the XXX mode
and its operating environment,

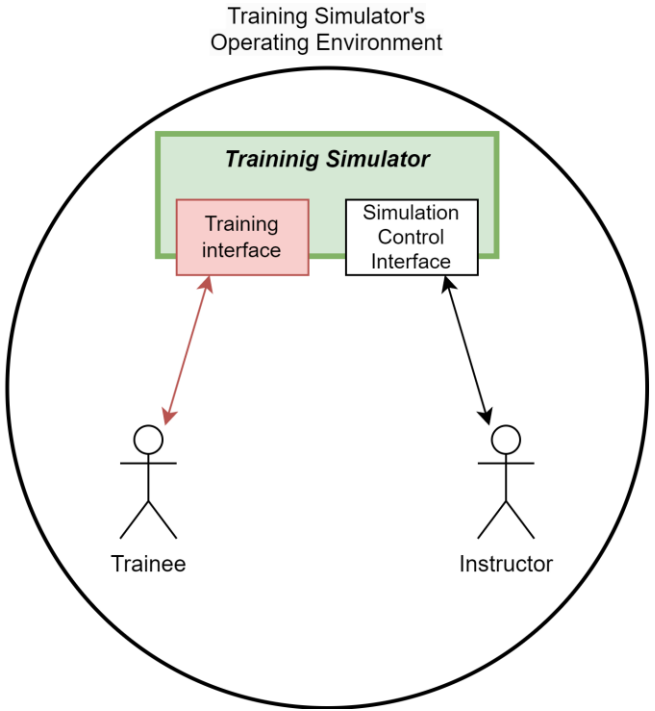
under the freely controlled by the instructor, yet possible,
trajectory of states of the simulated system's operating environment.

Often simulator contracted together
with the system to-be-simulated,
when both systems are in the concept stage.

↓

Need for double system-agnostic requirements


Simulated System's
Operating Environment
Simulated System
Other human interface
Trained interface
Other interoper. interfaces
Env. input interfaces
Other IH
Trainee
(as an operator)
Other interoperating system 1, 2, 3...
Environment inputs


Training Simulator's
Operating Environment
Traininig Simulator
Training interface
Simulation Control Interface
Trainee
Instructor

How to formulate complete and verifiable training simulator requirements without assumptions on simulated system architecture?

Define the requirements as a reference to the simulated system.

How to formally reference the simulated system?

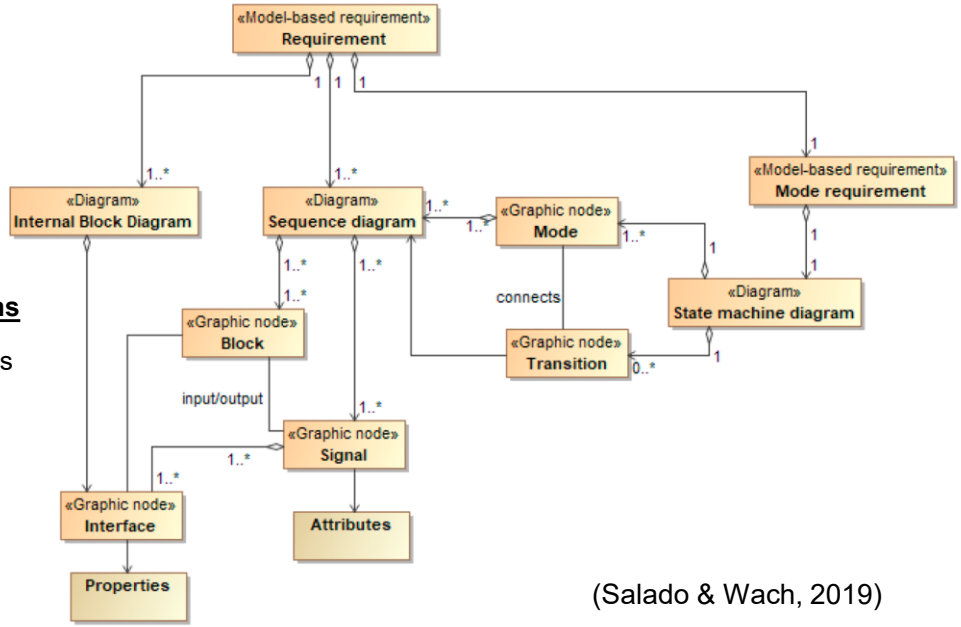
Reference simulated system model.

How to enable early elaboration of the simulator requirements wrt. system requirements?

Model and reference requirements using TMBR.

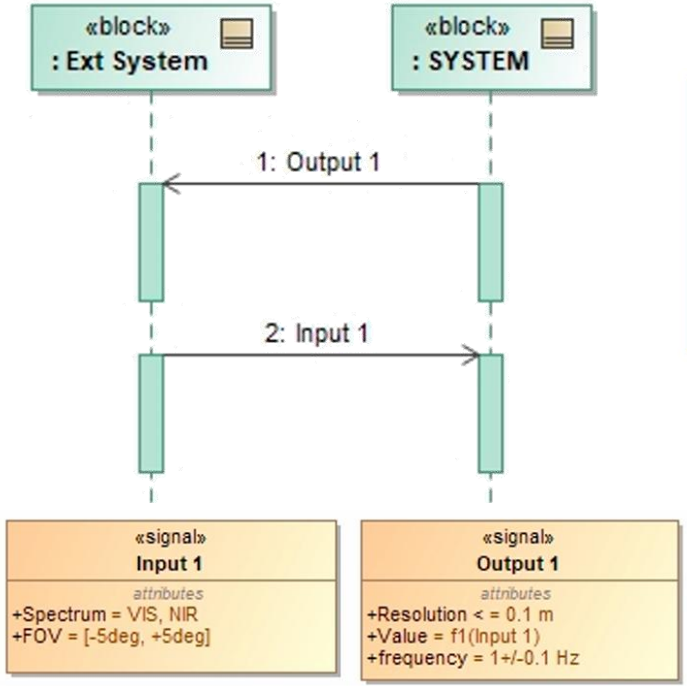
True Model-Based Requirements (TMBR) summary

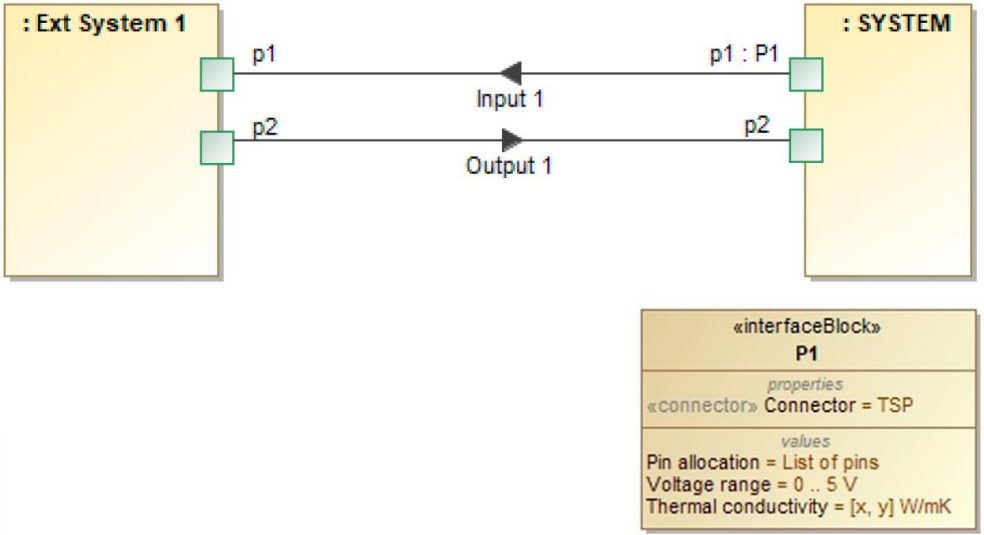
- Published (Salado & Wach, 2019)
- “The central proposition […] is that every requirement can be modeled as an input/output transformation.”
- Requirement taxonomy (prior work by Salado & Nilchiani)
  - “**Functional – What the system must do**
  - **Performance – How well the system must perform its functions**
  - Resource – What the system may consume to perform its functions at the required performance
  - Environment – Settings or contexts in which the system must perform its functions”
- Expressed in a SysML derivative



(Salado & Wach, 2019)

True Model-Based Requirements (TMBR) summary





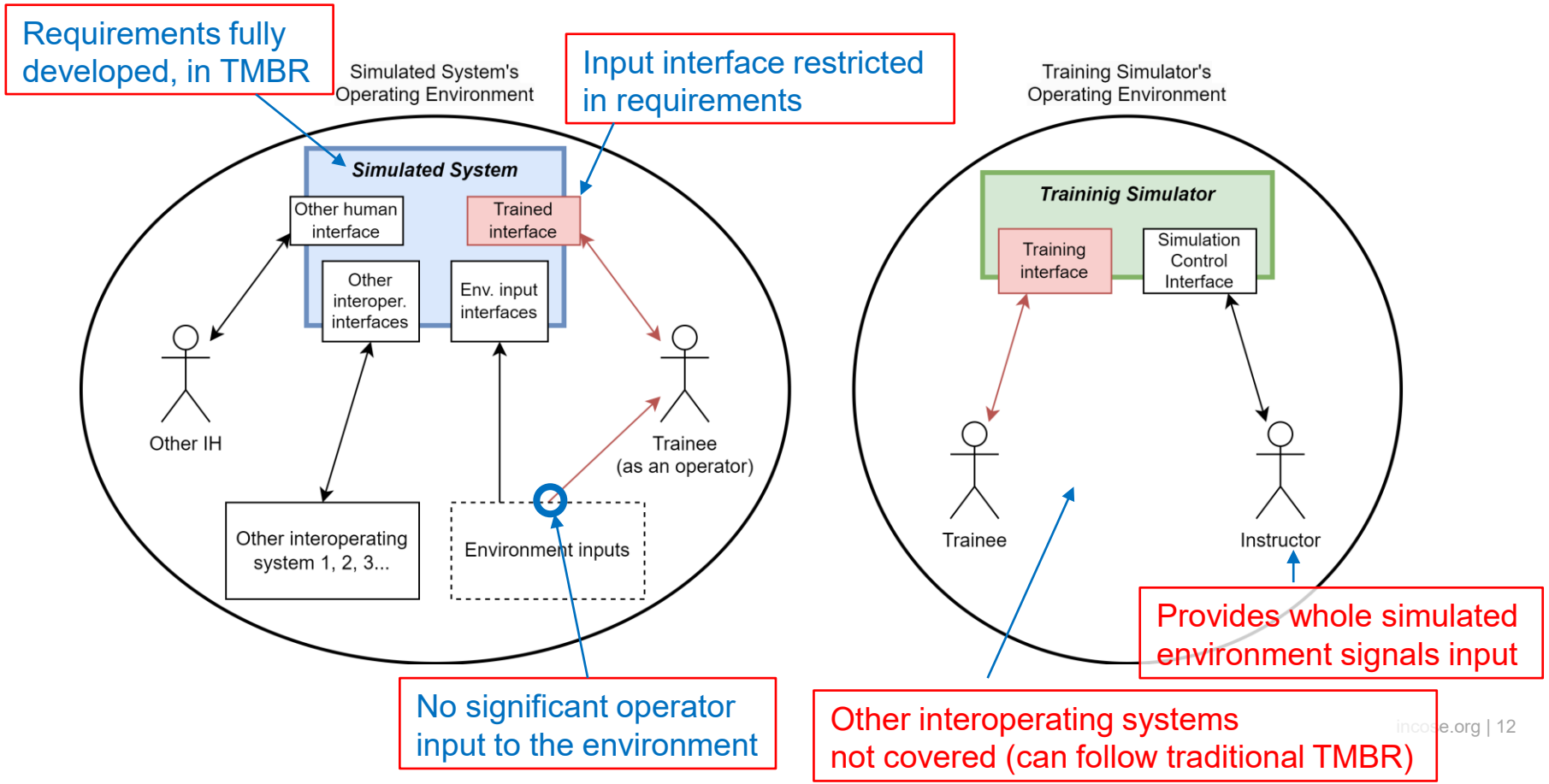
(Salado & Wach, 2019)

Approach

- **Find a transformation: System TMBR (and possibly other models) → Simulator TMBR**

- The transformation should yield a need for additional input, such as extra performance requirements, which is to be derived from the training simulator stakeholder needs.

- Ideally, the transformation itself should be clear enough to be a suitable substitution for directly defined requirements for a training simulator acquired together with the simulated system, for contractual purposes (where applicable), so that the resulting requirements evolve together with the simulated system's model.


Requirements fully developed, in TMBR
Simulated System's Operating Environment
Input interface restricted in requirements
Training Simulator's Operating Environment
Simulated System
Other human interface
Trained interface
Other interoper. interfaces
Env. input interfaces
Other IH
Trainee (as an operator)
Other interoperating system 1, 2, 3...
Environment inputs
Trainig Simulator
Training interface
Simulation Control Interface
Trainee
Instructor
Provides whole simulated environment signals input
No significant operator input to the environment
Other interoperating systems not covered (can follow traditional TMBR)

Central proposition

Define interfaces and to-be signals’ attributes early,

copy signals later

Proposed transformation summary

- Transformation covers:
  - Transformed once in system life (ideally):
    - Modes, State Machine Diagram, and Transition Sequence Diagram
    - **Interfaces with Properties**
  - Transformed iteratively:
    - **Signals, Sequence (Requirements) Diagrams**, and Mode Requirements
    - Signal to Interface allocation
    - **Signal Attributes**

- Neither formally verified nor proved to be the only feasible one (most likely not).

Interfaces

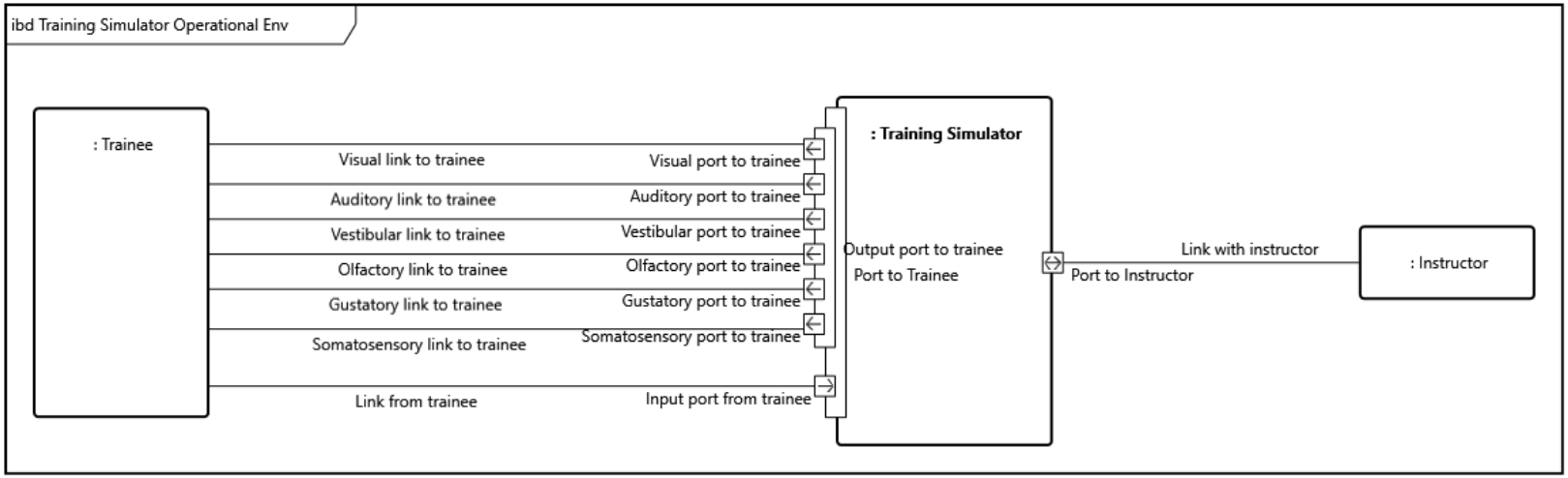
- ***Training interface*** **divided into output and input portion**
  - *And elaborated according to the simulated system trained interface* knowledge level
    - Output to trainee – not known or not fully known → Elaborated in simulator using human sensory breakdown to enable defining interface replication fidelity
    - Input from trainee – known and fixed → Mapped 1 to 1

- ***Simulation control interface*** **introduced without further elaboration**

Example (Wolfe et al., 2020)

1. Visual system – eyes
2. Auditory system – ears
3. Vestibular system – inner ear (responsible for acceleration sensing)
4. Olfactory system – nose
5. Gustatory system – mouth
6. Somatosensory system – skin

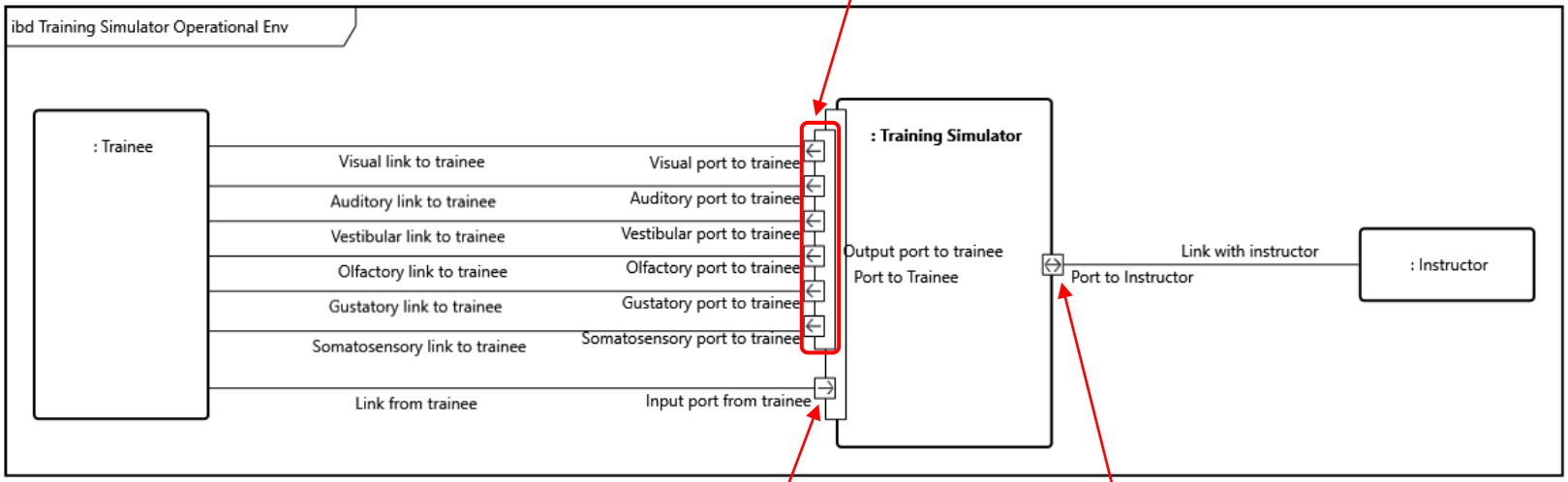
Interfaces


ibd Training Simulator Operational Env
: Trainee
Visual link to trainee
Auditory link to trainee
Vestibular link to trainee
Olfactory link to trainee
Gustatory link to trainee
Somatosensory link to trainee
Link from trainee
Visual port to trainee
Auditory port to trainee
Vestibular port to trainee
Olfactory port to trainee
Gustatory port to trainee
Somatosensory port to trainee
Input port from trainee
: Training Simulator
Output port to trainee
Port to Trainee
Link with instructor
Port to Instructor
: Instructor

Interfaces' properties

Example
- Visual
  - Allowed wavelength spectral deviation max. 10nm
- Somatosensory
  - Allowed full body acceleration deviation max. 1m/s²

Replication fidelity requirements

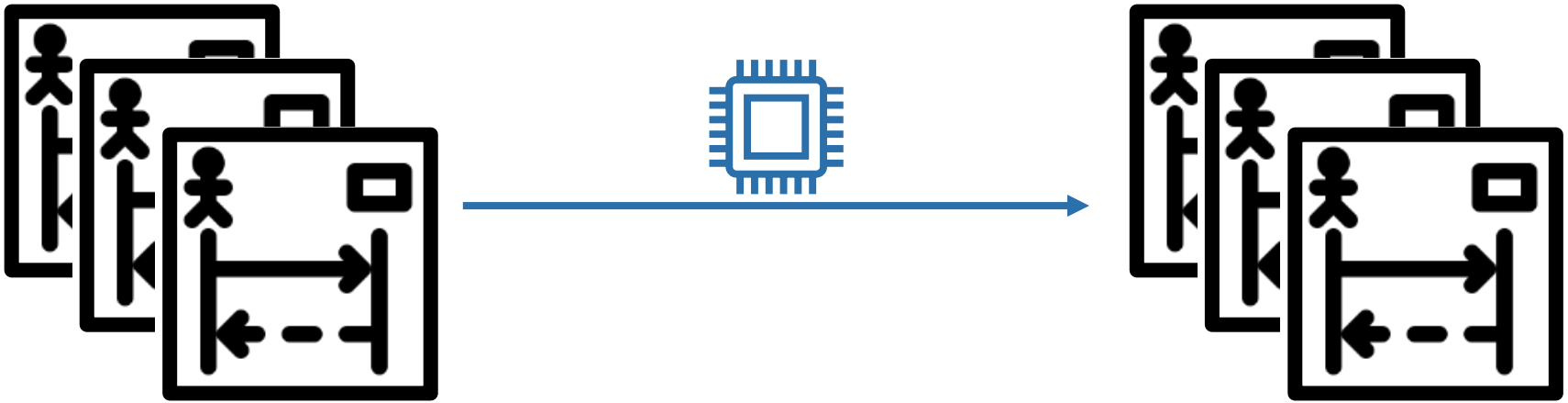


Full replication required

Virtually no "functional" requirements (as long as accommodates signals)

Signals and Sequence Diagrams

Where the magic happens…



Simulated system sequence diagrams

Simulator sequence diagrams

Signals and Sequence Diagrams

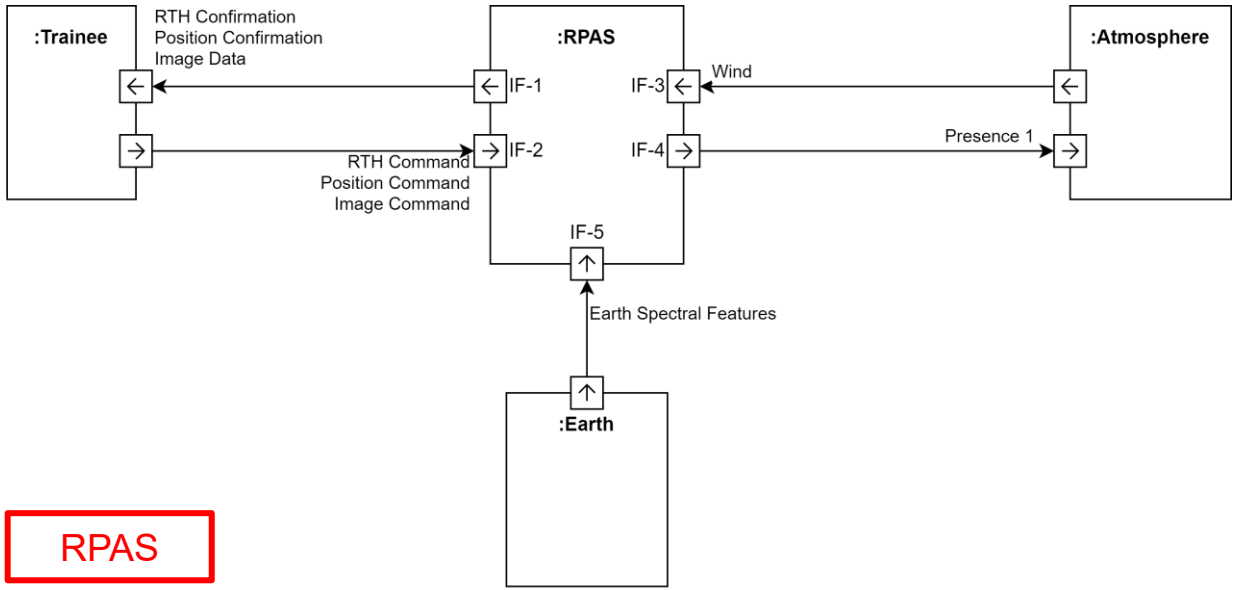
Where the magic happens…

1. The *simulated system* block shall be changed to the *training simulator*.
2. The origin of any signal providing an input to the *trainee* shall be changed to the *training simulator* lifeline.
   a) For any signal whose origin was actually changed, an additional signal, encapsulating the former signal into human input and preceding the former signal shall be added, originating at the i*nstructor's* lifeline and flowing to the *simulation system* lifeline.
3. All signals originating at the *trainee*'s lifeline and flowing to the lifeline other than the *training simulator* shall be removed.
4. Any block other than the *trainee* and *simulated system* shall be substituted with the *instructor*. If any signal appears to be transmitted within the same lifeline, it should be removed. All signals shall be encapsulated as human input/output.
   a) If any conditional operator refers to the state of any block other than the *trainee* or *training system*, substitute the conditional operator with the *instructor's* decision.
5. The duration constraints shall be adjusted (possibly with a change of their formulation) so that they reflect the allowed deviation of sequence timing in the simulation from the real-world scenario.

RPAS Example

+

Example -- Interfaces



RPAS

<<interfaceBlock>>
IF-1

values
Type = "HMI"
Subtype = "Visual"
Atention area = < = 0.3m^2
Readability distance = <0.4; 0.7>m

<<interfaceBlock>>
IF-2

values
Type = "HMI"
Subtype = "Manual, right-hand-operated"
Reachability = "According to sitting ergonomy as per ISOXXXX"

<<interfaceBlock>>
IF-3

values
Type = "Mechanical"

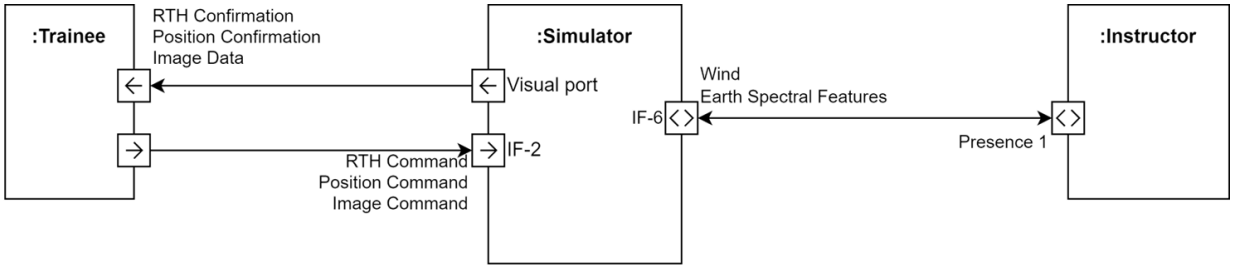
<<interfaceBlock>>
IF-4

values
Type = "Logical"
Subtype = "Location"

<<interfaceBlock>>
IF-5

values
Type = "Logical"
Subtype = "Spectral exposure"

Example -- Interfaces



<<interfaceBlock>>
Visual

values
Type = "HMI output"
Subtype = "Visual"
Allowed wavelength spectral deviation = +/- 10nm
Allowed angular deviation wrt. trainee eye at any point = +/- 1deg

<<interfaceBlock>>
IF-2

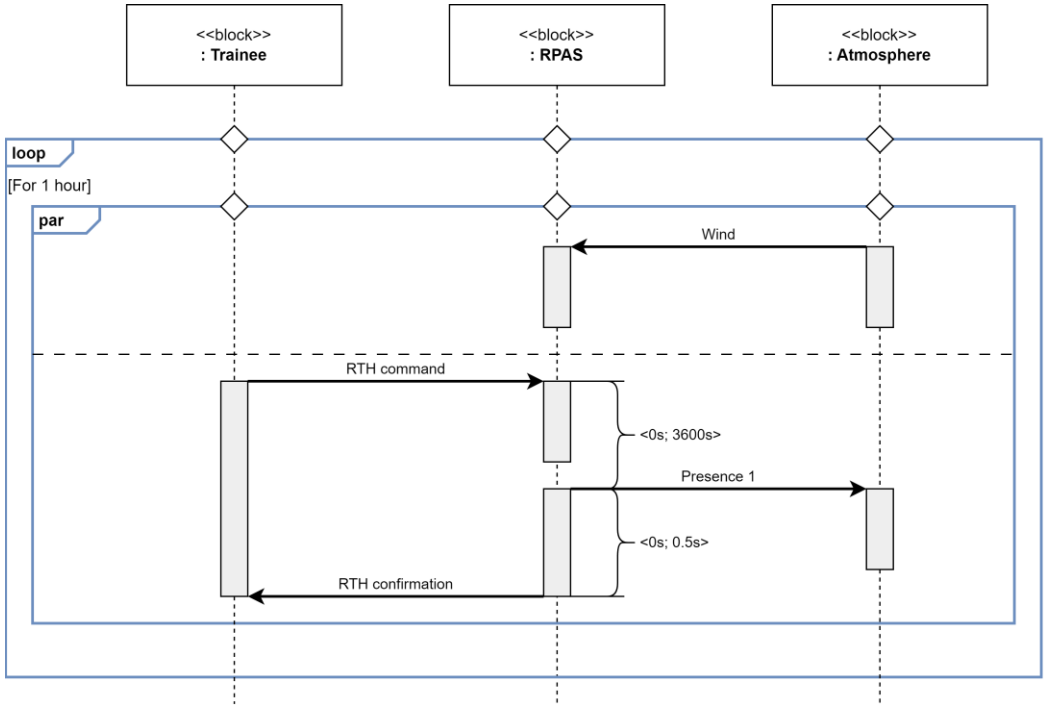
values
Type = "HMI"
Subtype = "Manual, right-hand-operated"
Reachability = "According to sitting ergonomy as per ISOXXXX"
More properties to come with RPAS model elaboration...

<<interfaceBlock>>
IF-6

values
Type = "HMI"
Subtype = "Simulation Control Interface"

Simulator

Example – Sequence Diagrams



<<signal>>
Wind

attributes
+Flow type = Continuous
+SpeedRange = <0; 40m/s>
+DirectionRange = <0; 360>deg

<<signal>>
RTH command

attributes
+Flow type = Trigger

<<signal>>
Presence 1

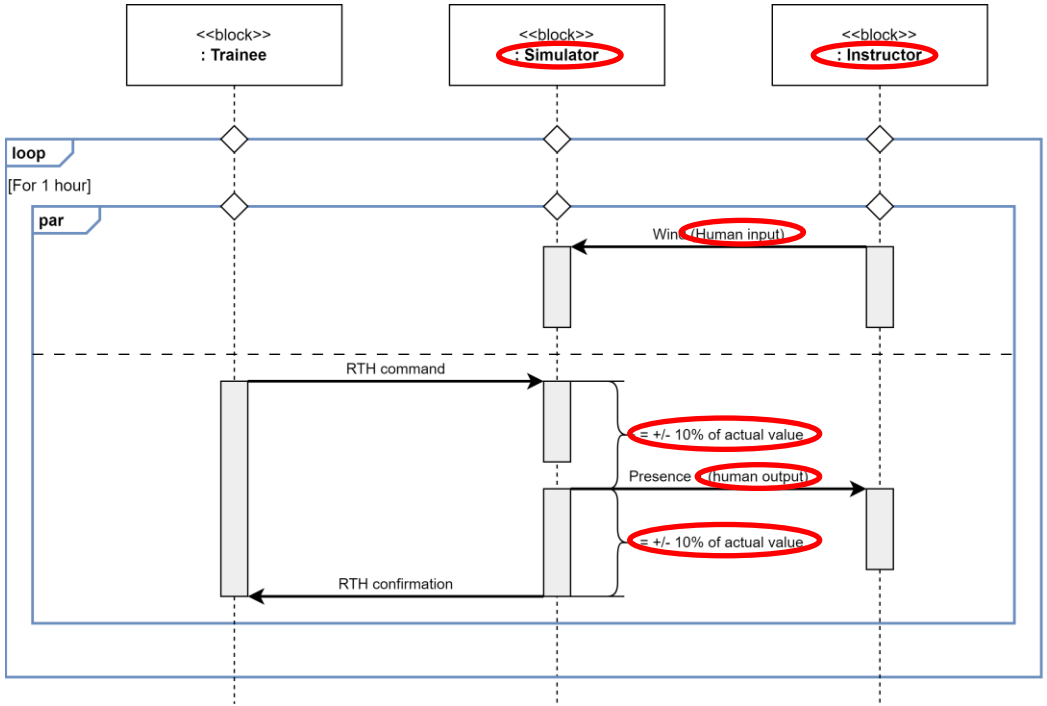
attributes
+Flow type = Trigger, logic
+Meaning = "The whole RPAS is within 10km sphere from the takeoff point"

<<signal>>
RTH confirmation

attributes
+Flow type = Trigger

RPAS

Example – Sequence Diagrams



<<signal>>
Wind (human input)
attributes
+Flow type = Continuous

<<signal>>
RTH command
attributes
+Flow type = Trigger

<<signal>>
Presence (human output)
attributes
+Flow type = Trigger, logic
+Meaning = "The whole RPAS is within 10km sphere from the takeoff point"

<<signal>>
RTH confirmation
attributes
+Flow type = Trigger

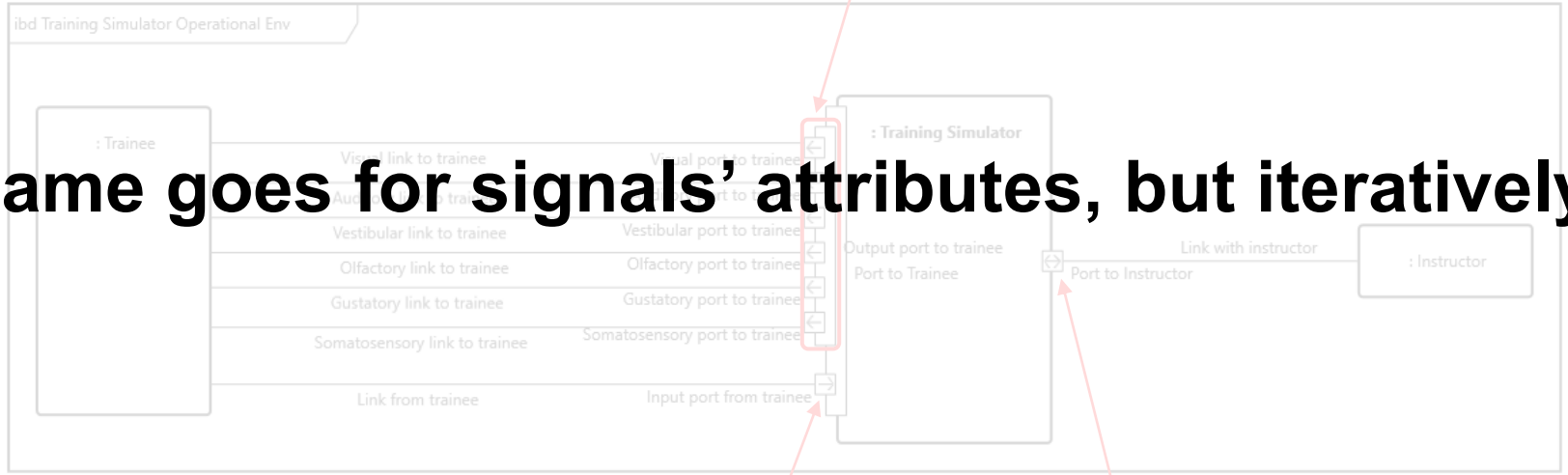
Simulator

Interfaces' properties

- Click to add text

Example
- Visual
  - Allowed wavelength spectral deviation max. 10nm
- Somatosensory
  - Allowed full body acceleration deviation max. 1m/s²

Replication fidelity requirements



Same goes for signals' attributes, but iteratively

Full replication required

Virtually no "functional" requirements (as long as accommodates signals)

Conclusions and Recommendations

Conclusions

- Method proposed can potentially help in training simulators requirements definition
- Model transformation-based requirements definition was exemplified
- Nowhere close to contractual usage

Recommendations

- Formally verify and live-project validate the transformation
- Extend the method to cover high-data-rate environment input
- Transformation can be implemented in a MBSE tool

References

Bartneck, C., Belpaeme, T., Eyssel, F., Kanda, T., Keijsers, M., & Šabanović, S. (2020). Human-Robot Interaction: An Introduction. Cambridge University Press. https://doi.org/10.1017/9781108676649

Friedenthal, S., Stoewer, H., Davey, C., Sky Matthews, David Nichols, Paul Nielsen, Christopher Oster, Taylor Riethle, Garry Roedler, Paul Schreinemakers, & Emma Sparks. (2021). Systems Engineering Vision 2035. INCOSE.

Gómez, V., & Figueroa, P. (2024). ProtoColVR: Requirements Gathering and Collaborative Rapid Prototyping of VR Training Simulators for Multidisciplinary Teams. IEEE Transactions on Visualization and Computer Graphics, 30(5), 2549–2558. https://doi.org/10.1109/TVCG.2024.3372057

Karray, F., Alemzadeh, M., Saleh, J. A., & Arab, M. N. (2017). Human-Computer Interaction: Overview on State of the Art. International Journal on Smart Sensing and Intelligent Systems, 1(1), 137–159. https://doi.org/10.21307/ijssis-2017-283

Komiienko, A., Kelemen, M., & Jevčák, J. (2023). The Ergonomic Design of the Tower Simulator Concept for the Training Needs of Air Traffic Control Students. 2023 New Trends in Aviation Development (NTAD), 144–149. https://doi.org/10.1109/NTAD61230.2023.10380152

Louis S. Wheatcraft, Michael J. Ryan, & Tami Edner Katz. (2024). INCOSE Needs and Requirements Manual: Needs, Requirements, Verification, Validation Across the Lifecycle.

McCracken, J., & Aldrich, T. B. (1984). Analyses of Selected LHX Mission Functions: Implications for Operator Workload and System Automation Goals. https://api.semanticscholar.org/CorpusID:109670107

MIL-STD-1472H Human Engineering. (2020). https://quicksearch.dla.mil/qsDocDetails.aspx?ident_number=36903

Munim, Z. H., Schramm, H. J., Krabbel, H., Nyairo, F., Haavardtun, P., Kim, T.-E., & Bustgaard, M. (2023). User Requirements for Learning Analytics Dashboard in Maritime Simulator Training. 2023 IEEE International Conference on Industrial Engineering and Engineering Management (IEEM), 0406–0410. https://doi.org/10.1109/IEEM58616.2023.10406321

Polcak, M., & Frantis, P. (2024). Resolution Requirements for Virtual and Mixed Reality Pilot Training. 2024 AIAA DATC/IEEE 43rd Digital Avionics Systems Conference (DASC), 1–5. https://doi.org/10.1109/DASC62030.2024.10749612

Ryan, M., & Wheatcraft, L. (2023). Guide to Writing Requirements. Requirements Working Group, International Council on Systems Engineering.

Salado, A. (2023). Model-Based Requirements. In A. M. Madni, N. Augustine, & M. Sievers (Eds.), Handbook of Model-Based Systems Engineering (pp. 349–377). Springer International Pub-lishing. https://doi.org/10.1007/978-3-030-93582-5_19

Salado, A., & Wach, P. (2019). Constructing True Model-Based Requirements in SysML. Systems, 7(2), Article 2. https://doi.org/10.3390/systems7020019

Wolfe, J. M., Kluender, K. R., & Levi, D. M. (2020). Sensation & perception (Sixth Edition). Sinauer Associates: Oxford University Press.

Wymore, A. W. (1993). Model-Based Systems Engineering. CRC Press.

Zimmermann, M., & de Weck, O. (2022). Formulating Engineering Systems Requirements. In A. Maier, J. Oehmen, & P. E. Vermaas (Eds.), Handbook of Engineering Systems Design (pp. 441–491). Springer International Publishing. https://doi.org/10.1007/978-3-030-81159-4_33

Software tools: Vitech Genesys, draw.io

Let's connect

Władysław Sowul

e-mail: wladyslaw@sowul.net

Systems Engineer

Unmanned Aerial Vehicles Department

Military Aviation Works No. 2, part of the Polish Armaments Group

TU Delft Aerospace Engineering BSc with HPB in Robotics 2024

INCOSE Poland VP at Large, ASEP