2018

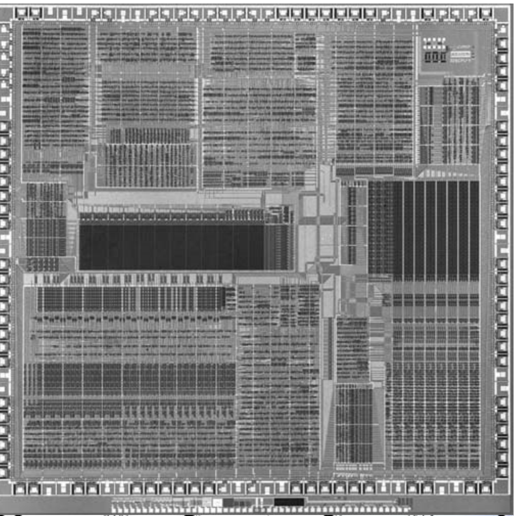
Annual INCOSE

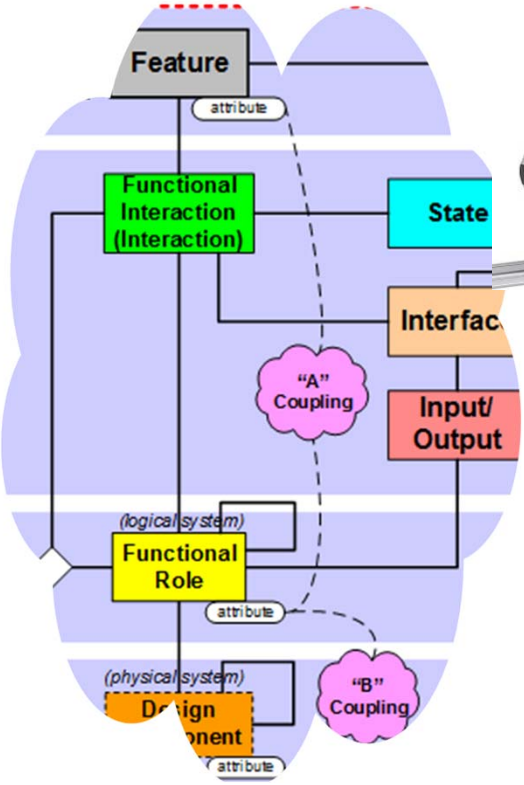
international workshop

Jacksonville, FL, USA

January 20 - 23, 2018

MBSE Patterns Working Group





V1.4.2

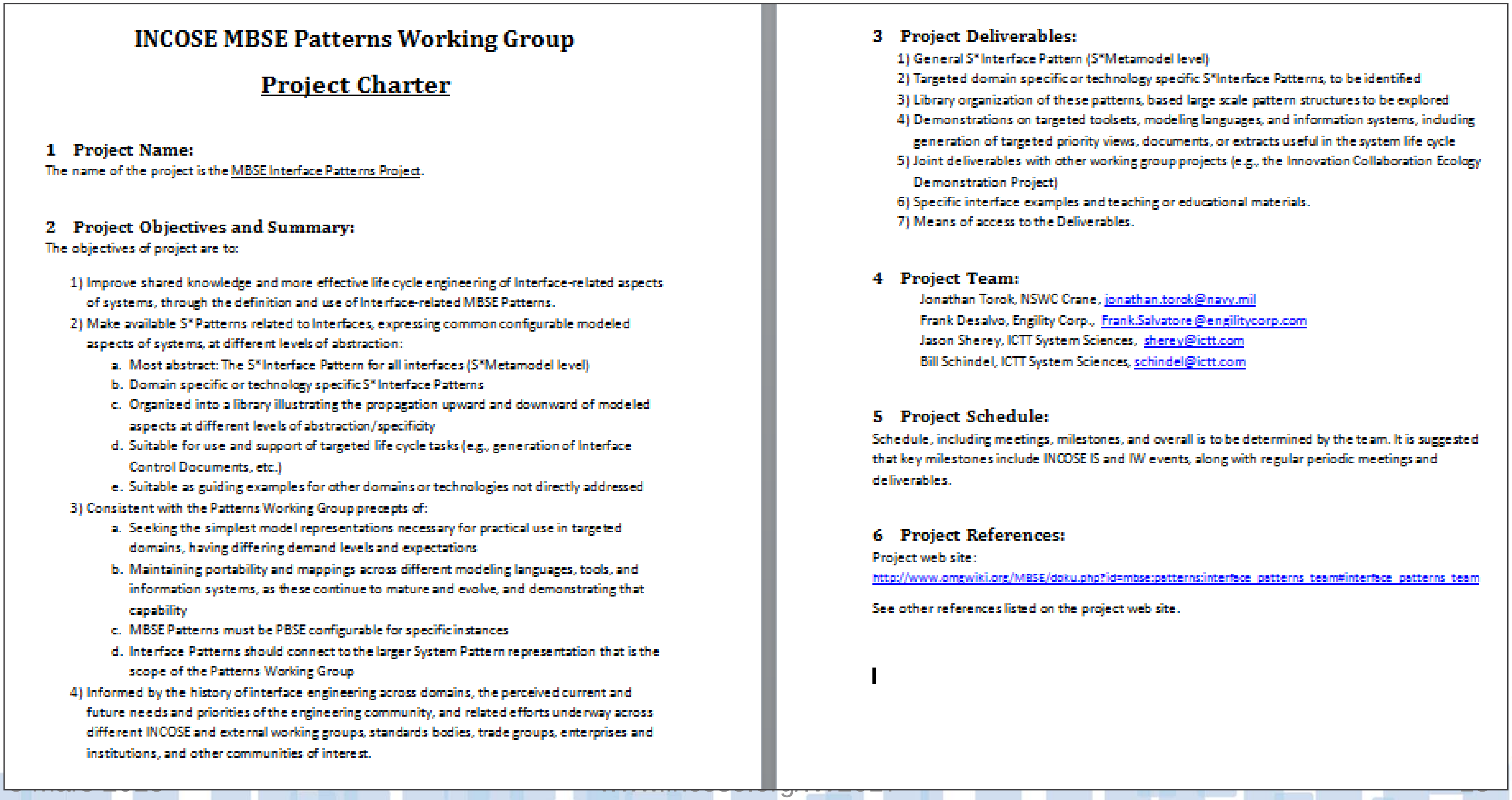
WG Project Team:
Jon Torok, Frank Salvatore,
Jason Sherey, Stephen Lewis

INCOSE
MBSE Patterns Working Group

Interface Patterns Project: Jan 2018 Status

- Have identified relevant subset of S*Metamodel providing a basis for S*Interface Patterns
- Have identified and high level model framed a few interface types for initial attention, as configurable S*Interface Patterns (including initial use in V4 Institute and ST4SE Ontology work)
- Now creating high level model framed general Interface Control Document (ICD) query for any S*Interface
- Have reviewed related draft SysML 2.0 and JPL publications, and provided formal written feedback to SysML 2.0—in particular, on Interfaces, where some SysML updates are noted in direction of S*Interface metamodel.
- Meeting at IW2018 on 1/22 for next steps.

Current project example: Interface Patterns Project



INCOSE MBSE Patterns Working Group

Project Charter

1 Project Name:

The name of the project is the MBSE Interface Patterns Project.

2 Project Objectives and Summary:

The objectives of project are to:

1) Improve shared knowledge and more effective life cycle engineering of Interface-related aspects of systems, through the definition and use of Interface-related MBSE Patterns.
2) Make available S*Patterns related to Interfaces, expressing common configurable modeled aspects of systems, at different levels of abstraction:
    - a. Most abstract: The S*Interface Pattern for all interfaces (S*Metamodel level)
    - b. Domain specific or technology specific S*Interface Patterns
    - c. Organized into a library illustrating the propagation upward and downward of modeled aspects at different levels of abstraction/specificity
    - d. Suitable for use and support of targeted life cycle tasks (e.g., generation of Interface Control Documents, etc.)
    - e. Suitable as guiding examples for other domains or technologies not directly addressed
3) Consistent with the Patterns Working Group precepts of:
    - a. Seeking the simplest model representations necessary for practical use in targeted domains, having differing demand levels and expectations
    - b. Maintaining portability and mappings across different modeling languages, tools, and information systems, as these continue to mature and evolve, and demonstrating that capability
    - c. MBSE Patterns must be PBSE configurable for specific instances
    - d. Interface Patterns should connect to the larger System Pattern representation that is the scope of the Patterns Working Group
4) Informed by the history of interface engineering across domains, the perceived current and future needs and priorities of the engineering community, and related efforts underway across different INCOSE and external working groups, standards bodies, trade groups, enterprises and institutions, and other communities of interest.

3 Project Deliverables:

1) General S*Interface Pattern (S*Metamodel level)
2) Targeted domain specific or technology specific S*Interface Patterns, to be identified
3) Library organization of these patterns, based large scale pattern structures to be explored
4) Demonstrations on targeted toolsets, modeling languages, and information systems, including generation of targeted priority views, documents, or extracts useful in the system life cycle
5) Joint deliverables with other working group projects (e.g., the Innovation Collaboration Ecology Demonstration Project)
6) Specific interface examples and teaching or educational materials.
7) Means of access to the Deliverables.

4 Project Team:

Jonathan Torok, NSWC Crane, jonathan.torok@navy.mil
Frank Desalvo, Engility Corp., Frank.Salvatore@engilitycorp.com
Jason Sherey, ICTT System Sciences, sherey@ictt.com
Bill Schindel, ICTT System Sciences, schindel@ictt.com

5 Project Schedule:

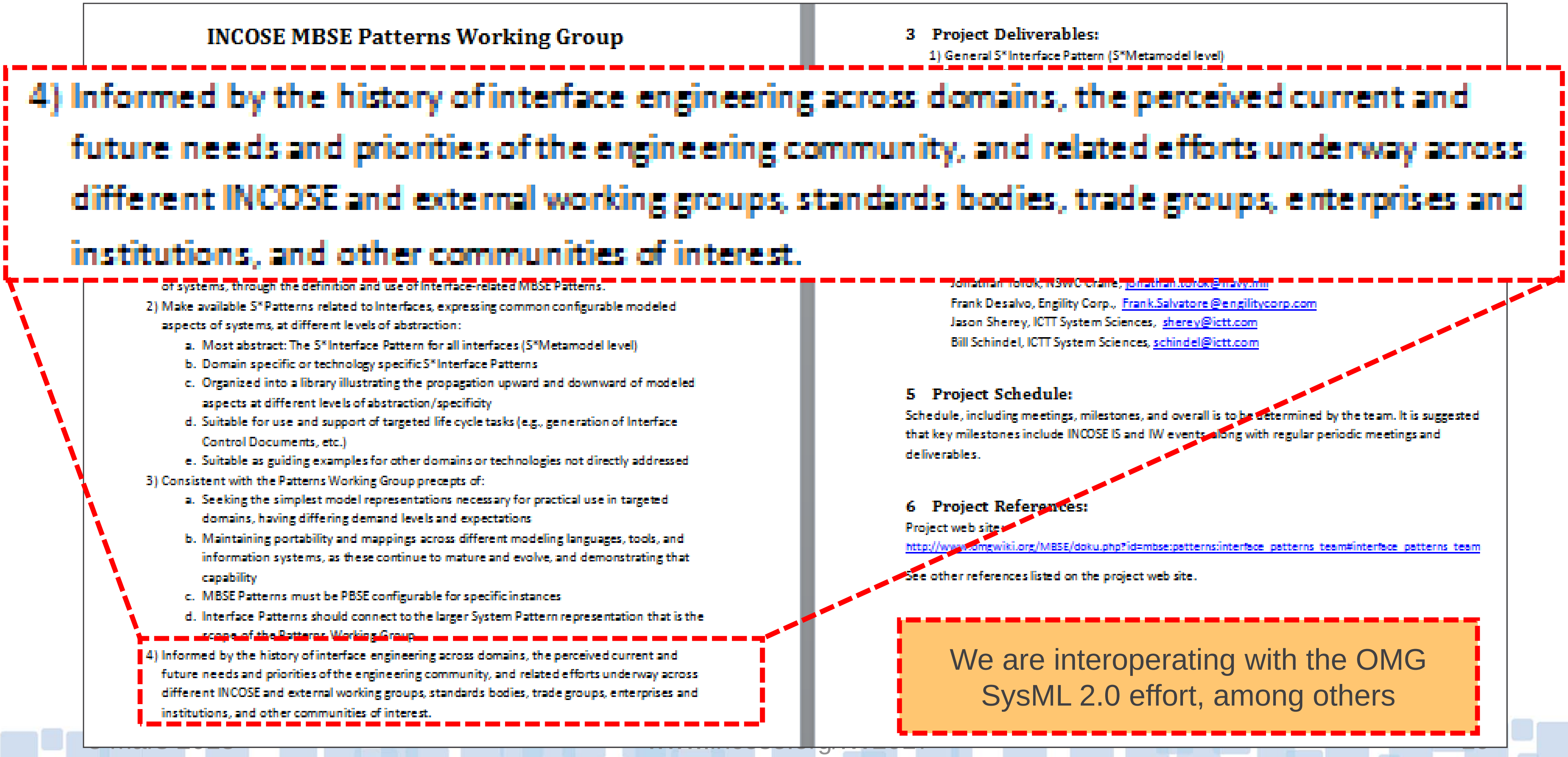
Schedule, including meetings, milestones, and overall is to be determined by the team. It is suggested that key milestones include INCOSE IS and IW events, along with regular periodic meetings and deliverables.

6 Project References:

Project web site:
http://www.omgwiki.org/MBSE/doku.php?id=mbse:patterns:interface_patterns_team#interface_patterns_team

See other references listed on the project web site.

Current project example: Interface Patterns Project



4) Informed by the history of interface engineering across domains, the perceived current and future needs and priorities of the engineering community, and related efforts underway across different INCOSE and external working groups, standards bodies, trade groups, enterprises and institutions, and other communities of interest.

INCOSE MBSE Patterns Working Group

of systems, through the definition and use of Interface-related MBSE Patterns.

2) Make available S*Patterns related to Interfaces, expressing common configurable modeled aspects of systems, at different levels of abstraction:
   a. Most abstract: The S*Interface Pattern for all interfaces (S*Metamodel level)
   b. Domain specific or technology specific S*Interface Patterns
   c. Organized into a library illustrating the propagation upward and downward of modeled aspects at different levels of abstraction/specificity
   d. Suitable for use and support of targeted life cycle tasks (e.g., generation of Interface Control Documents, etc.)
   e. Suitable as guiding examples for other domains or technologies not directly addressed

3) Consistent with the Patterns Working Group precepts of:
   a. Seeking the simplest model representations necessary for practical use in targeted domains, having differing demand levels and expectations
   b. Maintaining portability and mappings across different modeling languages, tools, and information systems, as these continue to mature and evolve, and demonstrating that capability
   c. MBSE Patterns must be PBSE configurable for specific instances
   d. Interface Patterns should connect to the larger System Pattern representation that is the scope of the Patterns Working Group

4) Informed by the history of interface engineering across domains, the perceived current and future needs and priorities of the engineering community, and related efforts underway across different INCOSE and external working groups, standards bodies, trade groups, enterprises and institutions, and other communities of interest.

3 Project Deliverables:

1) General S*Interface Pattern (S*Metamodel level)

Jonathan Torok, NSWC Crane, jonathan.torok@navy.mil
Frank Desalvo, Engility Corp., Frank.Salvatore@engilitycorp.com
Jason Sherey, ICTT System Sciences, sherey@ictt.com
Bill Schindel, ICTT System Sciences, schindel@ictt.com

5 Project Schedule:

Schedule, including meetings, milestones, and overall is to be determined by the team. It is suggested that key milestones include INCOSE IS and IW events, along with regular periodic meetings and deliverables.

6 Project References:

Project web site:
http://www.omgwiki.org/MBSE/doku.php?id=mbse:patterns:interface_patterns_team#interface_patterns_team

See other references listed on the project web site.

We are interoperating with the OMG SysML 2.0 effort, among others

Current project example: Interface Patterns Project

Project Workstreams:

1. Identify interface aspects of the S*Metamodel (the most abstract interface pattern)
2. Create library of interface patterns of different types (specializations of 1) showing techniques in mechanical, communication, visual, etc.
3. Identify queries and views that are interface-based (e.g., ICD, etc.), what metadata should appear in each of these.
4. Identify interface-oriented tasks, activities in the engineering life cycle (the reasons we are doing this project)
5. Down the road, issues of governance of the resulting patterns, their life cycles
6. Tactical level tool specific items, not necessarily all interface-oriented, along with mappings to SysML or specific tools

Discussion of S*Interface System of Access (SOA) Semantics

Interface Patterns Project Meeting

06.30.2017

Purpose of Following Material

1. The purpose of this material is to define a question, and propose an answer to it, concerning the underlying nature and meaning of one aspect of Interfaces.
2. This subject is about the underlying nature of interfaces, and not about any specific modeling language or notation.
3. This discussion therefore uses some basic concepts from the S*Metamodel description of Interfaces, not specific to any modeling language, notation, etc.
4. If we agree on the question and answer proposed here, then a follow-up action would be to agree on how to map it into SysML representation.
5. Trying to answer (4) before (1) – (3) seems to lead to confusion of what are the underlying issues versus language-specific representation issues.

General Setting

- Consider two interacting systems, exchanging at least one Input-Output (e.g., a Force, Energy Flow, Mass Flow, or Information), during Interaction D:

Interaction D

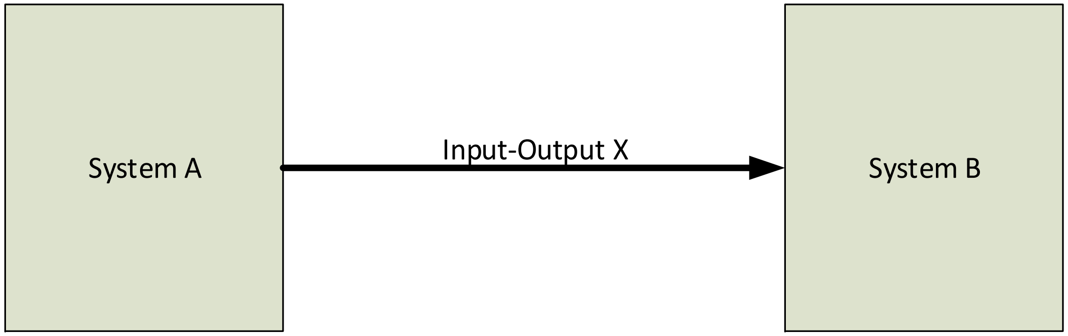


Figure 1: (Exact notation used not important to this discussion)

- In certain (important to identify) circumstances, we need to represent Interfaces involved in Interaction D.
- No matter what (graphical or other) modeling language or notation is used, the S*Metamodel tells us that an Interface is an association of:
  - A System, which "has" the Interface;
  - A (set of) Input-Output(s), which "pass through" the Interface;
  - A (set of) Interaction(s), which describe "behavior at the Interface;
  - A System of Access (SOA), providing the interaction "medium":

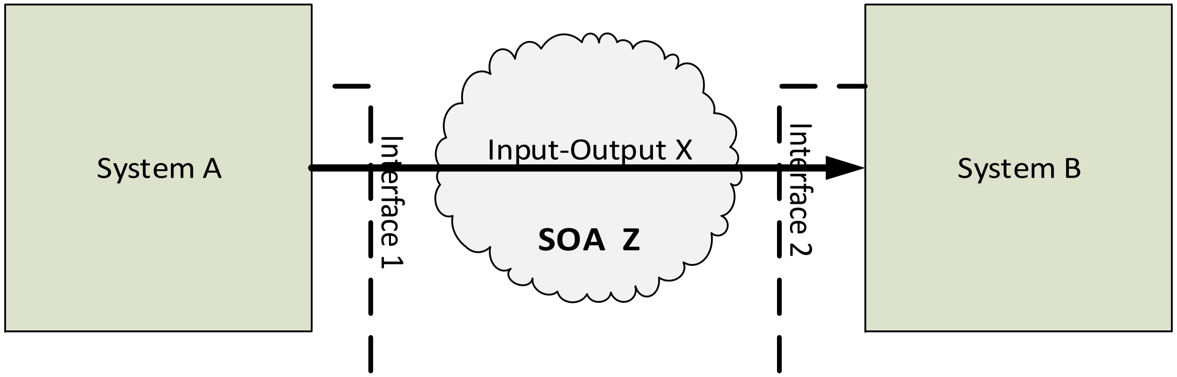


Figure 2: (Exact notation used not important to this discussion)

- However, there is a subtle inconsistency in the transition between Figure 1 and Figure 2 above:
  - – Figure 1 and Figure 2 imply that the scope of “System A” must have changed between the two diagrams, . . .
  - – Because, System A in Figure 2 can interact with an external-looking SOA Z, but . . . .
  - – System A in Figure 1 implies that the scope of System A is such that it can interact directly with System B.

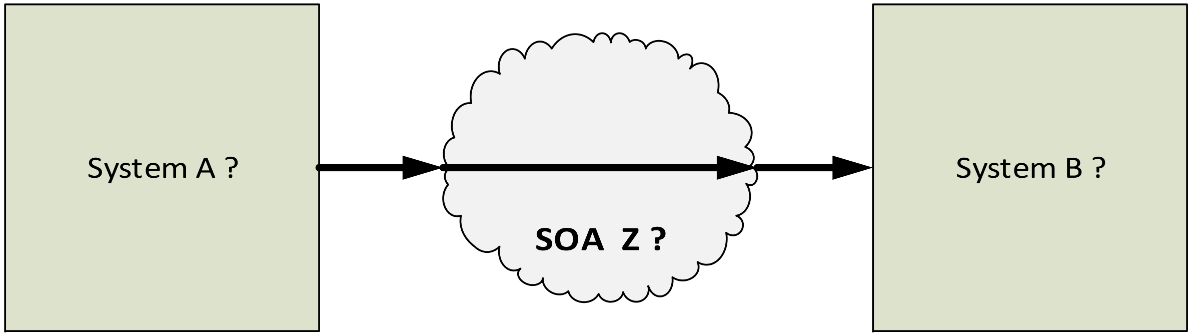


Figure 3: (Exact notation used not important to this discussion)

- The problem here is that even intended “neutral” notations can be specific enough to mislead us, or create ambiguities.
- The real problem is that, independent of notation, the System of Access by definition has larger scope than Figure 2 implied:

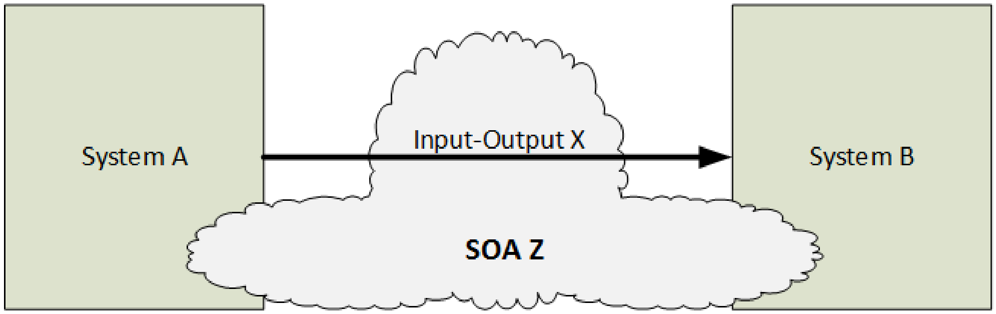


Figure 4: (Exact notation used not important to this discussion)

- Part of the scope of the System of Access for two interacting systems must necessarily be within the two interacting systems . . .

- So, to avoid conflicting or ambiguous definitions of the scope of System A, we have to recognize a slightly larger system, shown in Figure 5 as System A’
- The additional scope adds the SOA role shown here as SASOA:

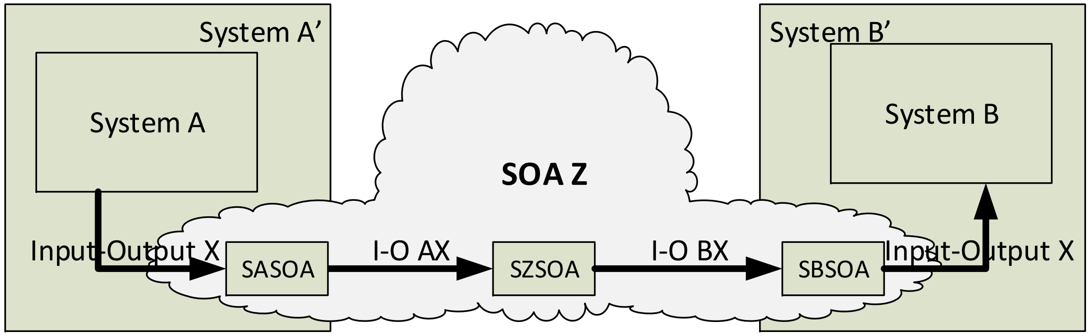


Figure 5: (Exact notation used not important to this discussion)

- The foregoing discussion simply reminds us that any system which we claim “has” an interface must include (inside it) the behavioral (SOA) role(s) necessary to support it (SASOA in Figure 5).
- And, if we model a system that “does not have” any interface (or does not have it “yet”), then we should not (later, or otherwise) see the same system boundary name and claim that it does have an interface—because the behavior boundary is different (System A versus System A’ in Figure 5.)

Implications for any Specific Language

- The above implies that, when we get ready to map to SysML or any specific modeling language/notation:
  - No matter what notation convention is used to show an Interface on a system boundary, applying it must mean that the named system includes the roles to support the interface; and . . .
  - When we show interacting systems that are not shown as having Interfaces, then those named system boundaries should not (even later in a design process) carry the <u>same name </u>as a system boundary that <u>does</u> have an interface.
- That is, System A is not System A':
  - System A' can show an Interface on its boundary (by whatever notational means is selected)
  - System A should not show any Interface on its boundary, but simply be shown as exchanging I/O with System B.

Valid Combinations

Not Valid Combinations

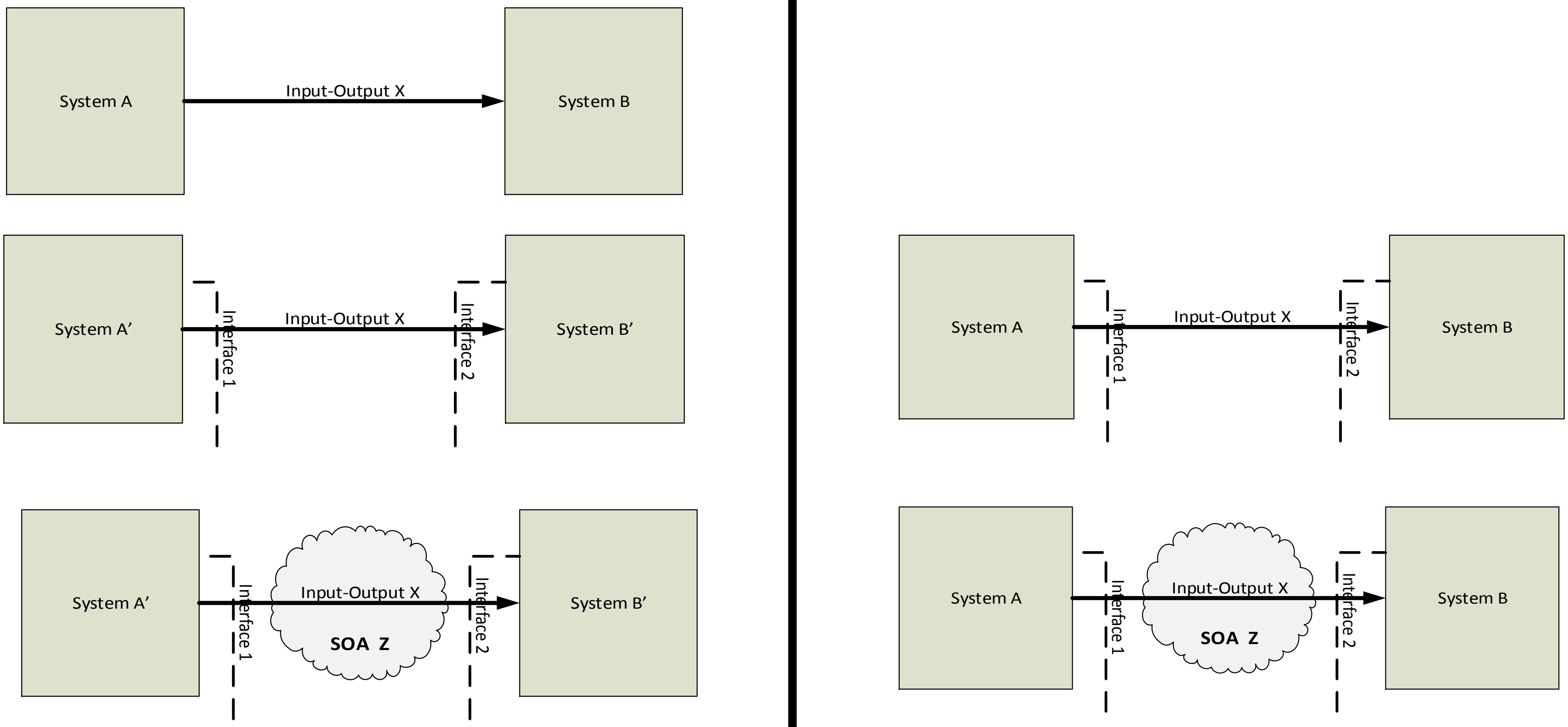


Figure 6: (Exact notation used not important to this discussion)

Do we agree on this?

- More discussion needed?
- If we agree, then let’s move on to discussion of what the SysML notation and mapping would be.