A simple classification of discrete system interactions and some consequences for the solution of the interoperability puzzle

Johannes Reich [1] Tizian Schröder [2]

[1]johannes.reich@sap.com, SAP SE, Walldorf, Germany

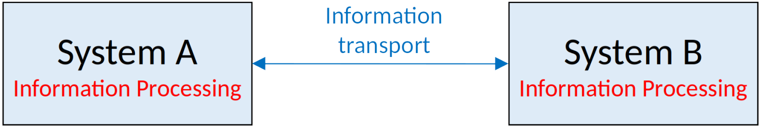
[2]tizian.schroeder@ovgu.de, Otto-von-Guericke-Universität, Magdeburg, Germany

2020-06-05

Introduction

Interoperability

Systems that exchange information are interoperable if their information processing is accomplished in a way that, by interaction, a specified purpose is fulfilled.



Interface

- Represents a system in its interaction.
- Provides necessary information about the system's capability to interoperate.

→ Is well defined only as it refers to transformational [i.e. mapping] behavior.

→ Different classes of interactions have to represented by different classes of interfaces.

Interaction classification

Information flow (=transport)

unidirectional vs. bidirectional

Information processing (=semantics)

Choice based on relevance for interface form

- A receiving system behaves deterministically if its (interaction-related) transition relation represents a transition function. Otherwise it behaves nondeterministically
- A receiving system behaves statefully if its set of reachable internal state values Q has more than one element. Otherwise it behaves stateless.
- A sending system behaves synchronously if the completion of the receiver's transition is a necessary and sufficient requirement for the sending system's next transition. Otherwise it behaves asynchronously.

Unidirectional information flow

Note: No backward information flow → No sync./async. distinction

Pipes

An overall computational function is computed in a number of successive steps on a "data flow", where the input of each pipe component is the output of the predecessor component (except for the first one). Thereby pipes provide the means for sequential and parallel system coupling.

Observation

The sender system "makes no assumptions" on the determinism and statefulness of the receiver system. E.g. realizable be publish-subscribe technology.

Bidirectional information flow

Distinction horizontal/vertical with respect to information processing (=semantics)

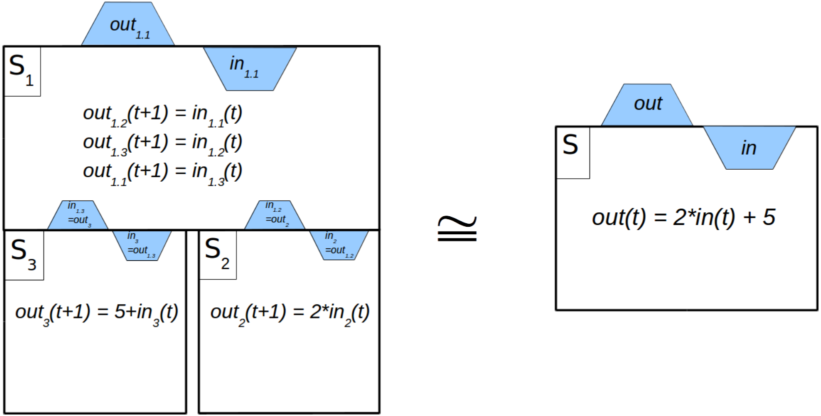
Symmetric case: Horizontal interaction

Consists of stateful, nondeterministic, asynchronous interactions in the form of semantic protocols (or choreographies). It is described by protocols, a bi- or multilateral interface, where the knowledge of the respective role of all involved systems is necessary to assure important properties of the interactions, like completeness, consistency, no deadlocks, livelocks and starvation.

Asymmetric case: Vertical interaction

Consist of top-down function calls (with exceptions) and bottom-up eventing. It is described by unilateral interfaces, i.e. only one interaction partner is described by the interface (the provider of the operation and events and not the consumer of the functionality and events).

An example for a simple system composition: $f(x) = 2x + 5$



$out_{1.2}(t+1) = in_{1.1}(t)$
$out_{1.3}(t+1) = in_{1.2}(t)$
$out_{1.1}(t+1) = in_{1.3}(t)$

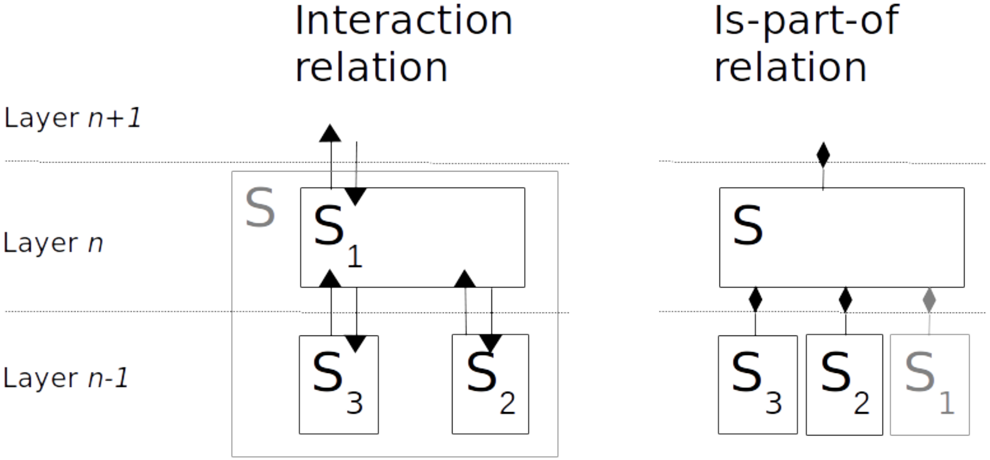
$out_3(t+1) = 5+in_3(t)$

$out_2(t+1) = 2*in_2(t)$

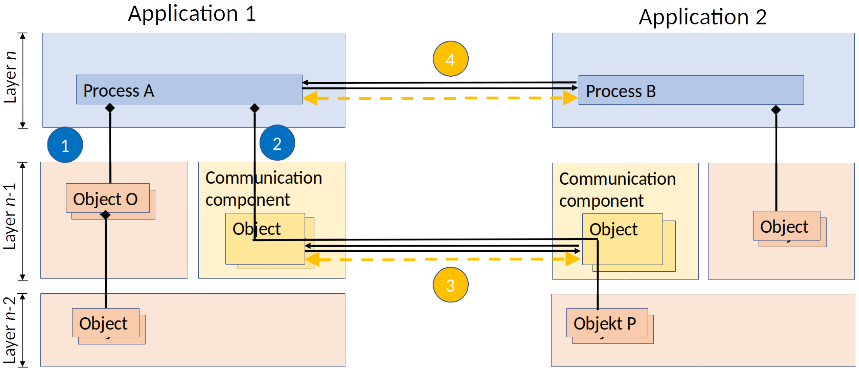
$out(t) = 2*in(t) + 5$

$\mathcal{S}_1 : f_1(x) =$ multiple 1:1 I/O-mapping
$\mathcal{S}_2 : f_2(x) = 2x$
$\mathcal{S}_1 : f_3(x) = x + 5$
$\mathcal{S} \ : f \ (x) = 2x + 5$

The same systems - two different hierarchies



Layered application architecture



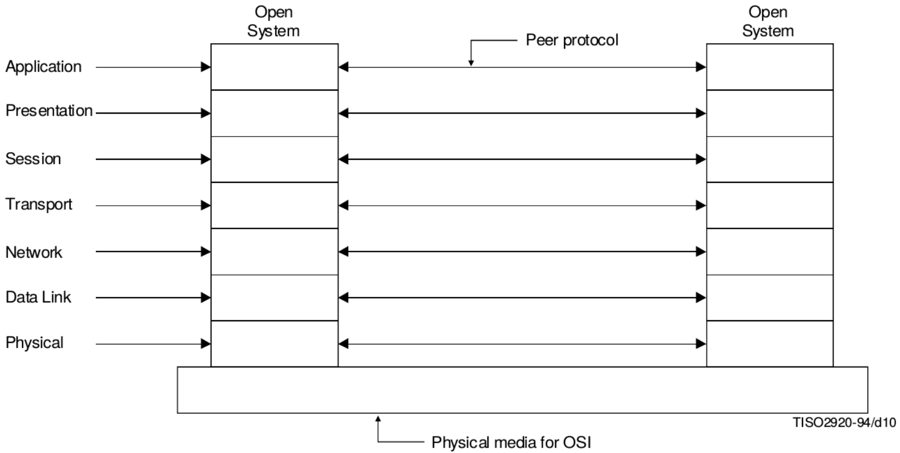
→ Flow of information

⧫ Vertical „is-part-of" relation

← - → Horizontal interaction (protocol)

1. Local „is-part-of" relation between proc. A and object O
2. Remote „is-part-of" relation between proc. A and object P
3. „Technical" protocol to realize transport functionality
4. „Semantic" protocol to realize a business interaction

Ref. Model of Open Systems Interconnection (OSI-Model)



The inconsistencies of the OSI-model

- The OSI-model "*is concerned only with the exchange of information between open systems*" - *and not with the internal "coordination [of] its interactions" / " internal functioning of each individual real open system*"

  ↯ Without relating to the structure of information processing, no statement about the structure of the application is possible!
- No clear hierarchy criterion: see interaction vs. "is-part-of"-relation.

  ↯ False assumption that we cannot remotely access deeper layers by protocol-mediated interactions.
- False assumption of the role of state.

  ↯ Only in the case of vertical interactions, the interaction related state can be encapsulated within an intermediate "session layer". In the horizontal interaction, it always relates to the layer, the interaction takes place.

Q: Is the concept of a SOA-Service well or ill-defined?

WSDL 1.1: defines the ”operation” of a service by 4 ”transmission primitives” a so called ”endpoint” has to support. No reference to any transformational semantics

- ”One-way”: The ”endpoint” receives a message.
- ”Request-response”: The ”endpoint” receives a message and sends back a correlated message.
- ”Solicit-response”: The ”endpoint” sends a message and receives a correlated message.
- ”Notification”: The ”endpoint” sends a message..

WSDL 2.0: defines (Section 2.2.1) an ”Interface component” as a ”*sequences of messages that a service sends and/or receives*”. According to WSDL 2.0, an ”operation” is an ”*interaction with the service consisting of a set of (ordinary and fault) messages exchanged between the service and the other parties involved in the interaction*”. No reference to any transformational semantics

BPMN 2.0.2: Is based on WSDL 2.0 as non-normative reference. Defines (section 8.5.3) ”*An operation defines messages that are consumed and, optionally, produced when the operation is called.*” No reference to any transformational semantics

A: No.

Q: Is the concept of a REST-service well or ill-defined?

A REST-service is supposed to follow 2-3 "principles":

- Addressability: each resource has to have a unique URI.
- Statelessness: each REST-message is supposed to contain all the information that is necessary for the processing which it initiates.
- Idem potency (sometimes): the called service is supposed to have an identical effect, no matter when it is called.

Strange consequences if we transfer "principles" that simplify information transportation to the domain of information processing.

A: No.

Summary: the delivered puzzle pieces

Some recommended questions: If someone talks about . . .

. . . interfaces. — Ask: What is the transformational and compositional behavior it represents?

. . . layered architectures. — Ask: What is your order criterion?

. . . components. — Ask: What is their compositional behavior?

. . . IoT use cases. — Ask: What kind of interaction class do you want to exemplify?

. . . horizontal interaction. — Ask: Is there bidirectional information flow and is it document based, nondeterministic, asynchronous and stateful? And: What are the competency domains of the participants?

. . . changing remote state by operations. — Ask: How reliable is the communication channel compared to the rest of the local application?

. . . technologies supporting horizontal interactions. — Ask: How do they support the implementation of protocols?

Thank You!

Questions?