A System Perspective on Processes and Their Interactions

Johannes Reich

2012-05-23

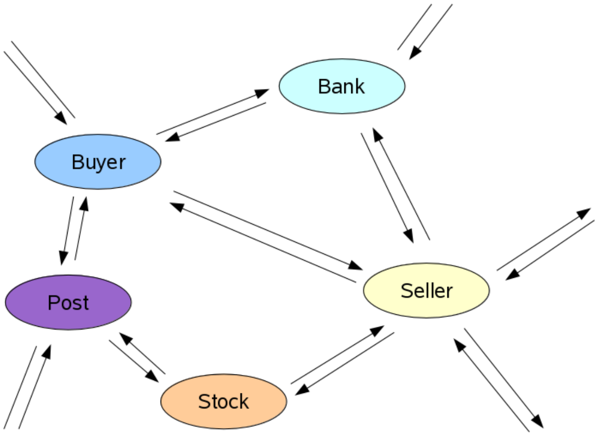
Table of Contents

1. Introduction
2. Systems
3. Interactions
4. Games
5. Processes
6. Components
7. Summary

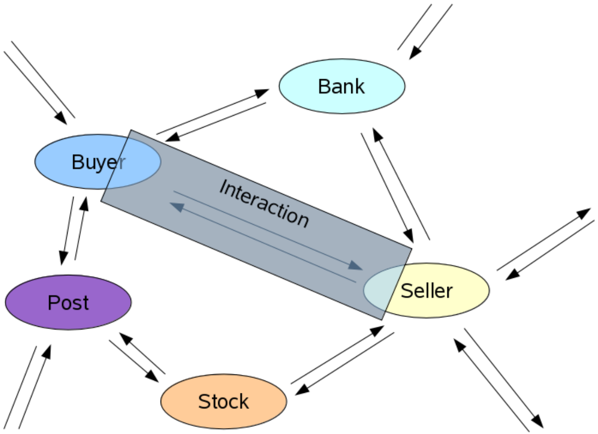
Claude E. Shannon (1948): A Mathematical Theory of Communication

"The fundamental problem of communication is that of reproducing at one point either exactly or approximately a message selected at another point. Frequently the messages have meaning; that is they refer to or are correlated according to some system with certain physical or conceptual entities. These semantic aspects of communication are irrelevant to the engineering problem."

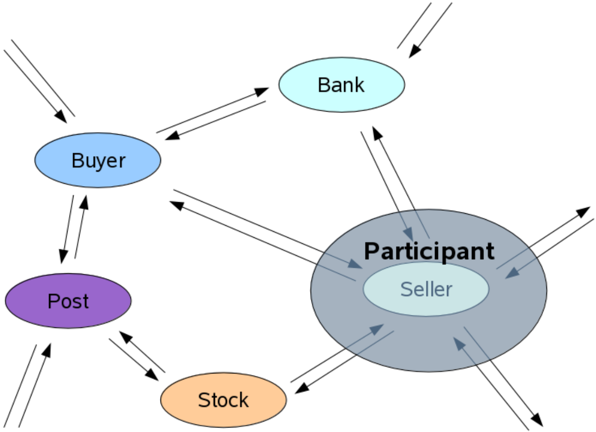
We are living in an open world of nondeterministic relations between systems



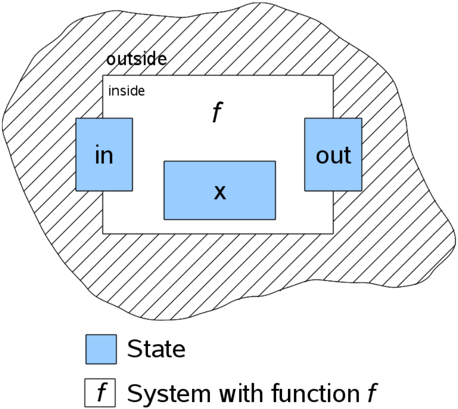
We are living in an open world of nondeterministic relations between systems: Focus on interactions

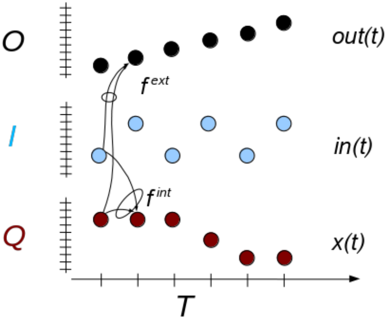


We are living in an open world of nondeterministic relations between systems: Focus on participants



Systems - Informally





(Finite) Systems - Formal Definition

A **finite system** is defined by a tuple $\mathcal{S} = (T, succ, Q, I, O, x, in, out, f)$.

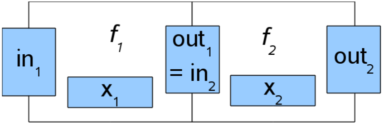
- T is the enumerable set of time values starting with 0 such that $succ : T \to T$ is the invertible time successor function.
- Q, I and O are the finite sets of state values for the internal, input and output states $(x, in, out) : T \to (Q, I^\epsilon, O^\epsilon)$.
- $f = (f^{ext}, f^{int}) : I^\epsilon \times Q \to O^\epsilon \times Q$ is a function describing the time evolution or system operation triggered by an update of its input parameters and updating the internal and output state in one time step for each $t \in T$:

$$\begin{pmatrix} out(t+1) \\ x(t+1) \end{pmatrix} = \begin{pmatrix} f^{ext}(in(t), x(t)) \\ f^{int}(in(t), x(t)) \end{pmatrix}.$$

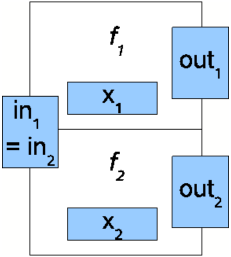
ϵ symbolizes the empty character and $I^\epsilon = I \cup \epsilon$ and $O^\epsilon = O \cup \epsilon$.
The n-fold application of $succ$ is written as $t +_{\mathcal{S}} n := succ^n_{\mathcal{S}}(t)$

System Composition/Super System Formation

Sequential Composition ($\mathcal{S}_2 \circ \mathcal{S}_1$)

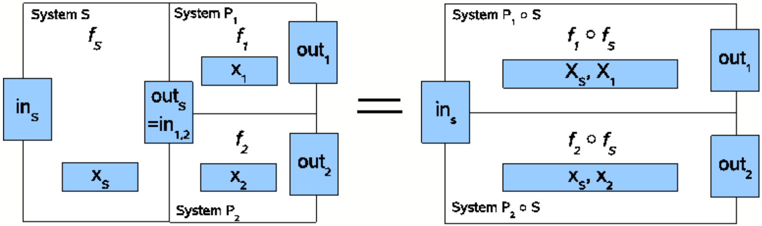


Parallel composition ($\mathcal{S}_2 || \mathcal{S}_1$)

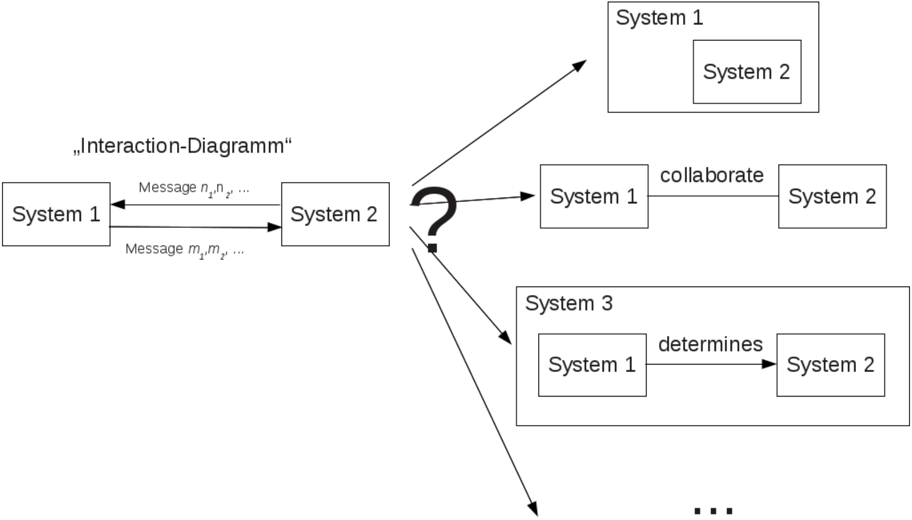


Combined System Composition

The right distribution law holds: $(\mathcal{P}_1 || \mathcal{P}_2) \circ \mathcal{S} = (\mathcal{P}_1 \circ \mathcal{S}) || (\mathcal{P}_2 \circ \mathcal{S})$



Richer Interaction Semantics



Projection of a System

Def.: A tupel $\mathcal{T} = (T, succ, Q_{\mathcal{T}}, I_{\mathcal{T}}, O_{\mathcal{T}}, x_{\mathcal{T}}, in_{\mathcal{T}}, out_{\mathcal{T}}, \Delta_{\mathcal{T}})$ is a **projection** of a system $\mathcal{S}$ if $Q_{\mathcal{T}} \subseteq Q_{\mathcal{S}}$, $I_{\mathcal{T}} \subseteq I_{\mathcal{S}}$, $O_{\mathcal{T}} \subseteq O_{\mathcal{S}}$ and $\Delta_{\mathcal{T}} \subseteq Q_{\mathcal{T}} \times Q_{\mathcal{T}} \times I_{\mathcal{T}} \times O_{\mathcal{T}}$ and a projection function $\pi = (\pi_Q, \pi_I, \pi_O) : Q_{\mathcal{S}} \times I_{\mathcal{S}}^{\epsilon} \times O_{\mathcal{S}}^{\epsilon} \to Q_{\mathcal{S}} \times I_{\mathcal{S}}^{\epsilon} \times O_{\mathcal{S}}^{\epsilon}$ with $\pi \circ \pi = \pi$ exists such that $\delta \in \Delta_{\mathcal{T}}$ iff there is a point in time $t \geq 0$ in a sequence such that $\delta = (\pi_Q(x(t)), \pi_Q(x(t+1)), \pi_I(in(t)), \pi_O(out(t+1)))$.

Nondeterministic Finite I/O-Automata (Transducer)

Def.: A **nondeterministic finite I/O automaton (NFIOA)** is defined by a tuple $\mathcal{A} = (Q, I, O, q_0, Acc, \Delta)$.

- Q is the non-empty finite set of state values,
- I and O are the (possible empty) finite input and output alphabets,
- q_0 is the initial state value,
- Acc is the acceptance component (e.g. a finite set of states for finite input sequences) and
- $\Delta \subseteq Q \times Q \times I^\epsilon \times O^\epsilon$ is the transition relation.

Def.: An NFIOA $\mathcal{A} = (Q_\mathcal{A}, I_\mathcal{A}, O_\mathcal{A}, q_0, Acc, \Delta)$ **specifies** the projection $\mathcal{T}$ of a system $\mathcal{S}$, if $Q_\mathcal{A} = Q_\mathcal{T}$, $I_\mathcal{A} = I_\mathcal{T}$, $O_\mathcal{A} = O_\mathcal{T}$, $q_0 = x_\mathcal{T}(0)$ and $\Delta_\mathcal{A} = \Delta_\mathcal{T}$.

Product Automata

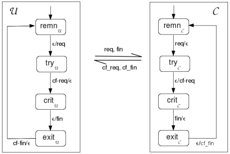
Any coupling of NFIOAs requires at least to view them as a single product automaton:

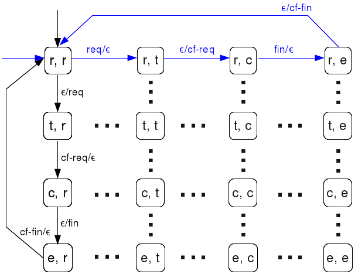
Def.: The **weakly synchronized product** of a set of n NFIOAs $\mathcal{A}_k$ is defined by NFIOA $\mathcal{B} = (Q, I, O, \vec{q}_0, Acc, \Delta)$, with $Q_\mathcal{B} = \times Q_k$, $I_\mathcal{B} = \times I_k$, $O_\mathcal{B} = \times O_k$, $\overrightarrow{q}_{0\mathcal{B}} = ({q_0}_1, \ldots, {q_0}_n)$, the common acceptance component represents the logical conjunction of the individual components, symbolized as $Acc_\mathcal{B} = \bigwedge Acc_k$, $\Delta_\mathcal{B} := \{(\vec{p}, \vec{q}, \vec{i}, \vec{o})|$ the components of $\vec{p}$ are reachable states of the $\mathcal{A}_i$ and $\mathcal{A}_k$ provides a transition (p_k, q_k, i_k, o_k) with $\vec{q} = \vec{p}\left[\frac{q_k}{p_k}, k\right]$ and $\vec{i} = \epsilon[i_k, k]$ and $\vec{o} = \epsilon[o_k, k]$ $\}$. I also write $\mathcal{B} = \bigotimes_{i=1}^{n} \mathcal{A}_i$.

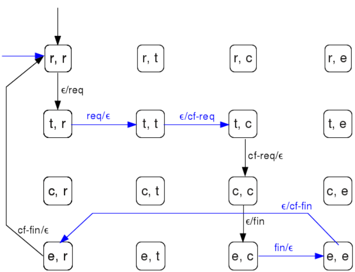
"Stronger" synchronization then means to eliminate some transitions from the transition relation $\Delta_\mathcal{P}$ and thereby to eliminate the symmetry of the unsynchronized product

The causal relation between the output and the input of different systems - Channel Based Restriction

Example of the Protocol of Mutual Exclusion




Weakly synchronized product


Channel based restricted product

Protocols

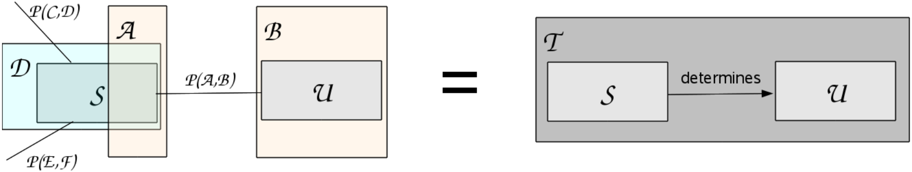
Two necessary conditions obviously are that first, there has to be a receiving transition for each sending transition for each channel and second, that the acceptance condition can still be met.

Def.: A cbr-automaton is called **well formed** if for every channel mediated transition which sends a character (different from ϵ) there exists an induced transition to process it.

Def.: A well formed cbr-automaton is called **consistent** if for each reachable state value either the acceptance condition is met or there is at least one continuation such that the acceptance condition can be met.

Def.: A **protocol** is a cbr-product automaton with no open input or output components. The individual factor automata are called **roles**.

Deterministically Interacting Systems and Supersystem Formation



Proposition: Let $\mathcal{S}$ and $\mathcal{U}$ be two systems described by DFIOAs $\mathcal{D}$ and $\mathcal{B}$ respectively. $\mathcal{U}$ interacts with $\mathcal{S}$ only by the consistent protocol $\mathcal{P}(\mathcal{A},\mathcal{B})$ with a set of final states as acceptance component, where $\mathcal{A}$ is an NFIOA describing only a projection of $\mathcal{S}$. $\mathcal{S}$ additionally interacts with other systems, denoted by $\sim\mathcal{U}$, by other consistent protocols. Then $\mathcal{S}$ and $\mathcal{U}$ are subsystems of a larger system $\mathcal{T}$.

J.Reich (2010), Finite system composition and interaction, in Klaus-Peter Fähnrich, Bogdan Franczyk (Eds), GI Lecture Notes in Informatics, Proceedings of the 40. Annual Conference of the dt. Gesellschaft für Informatik 2010 in Leipzig, Vol. 2, pp. 624-637.

Games versus Protocols

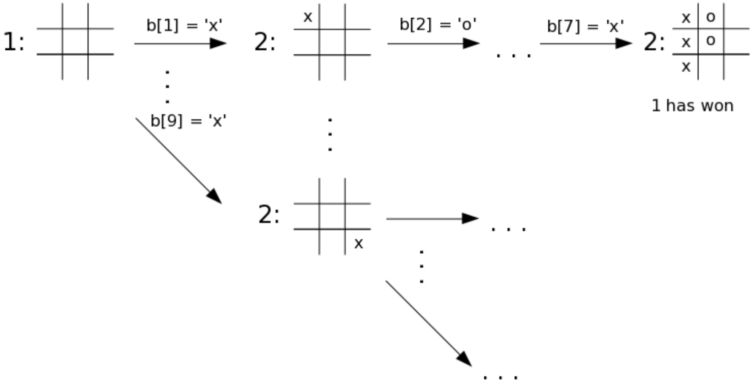
Game theory and protocol/process theory use the same interaction model!

Protocols + Decisions = Games - Payoff

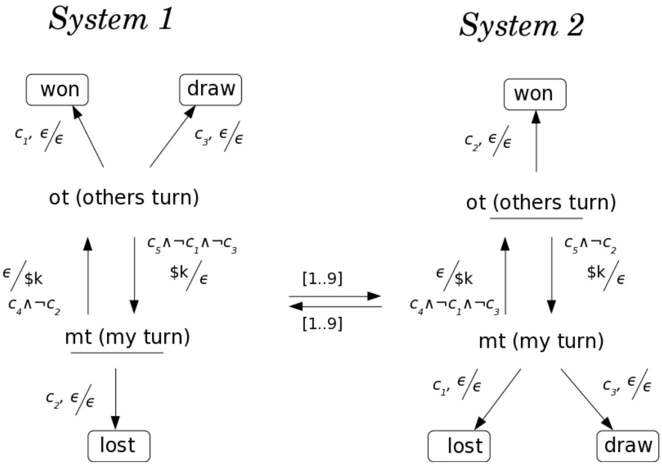
1. Create a product automaton between a protocol and an "decision automaton" and synchronize it such that the original protocol behavior is retained
2. Partition the set of states into "decision closures", i.e. all states that are automatically (by protocol interactions) reached from a given decision

J. Reich (2009). The relation between protocols and games, in S. Fischer, E. Maehle und R. Reischuk (Hrsg.), GI Lecture Notes in Informatics, Proceedings of the 39. Annual Conference of the Deutsche Gesellschaft für Informatik 2009 in Lübeck, pp. 3453-3464.

Example: tic tac toe as an extensive form game



Example: tic tac toe as a protocol

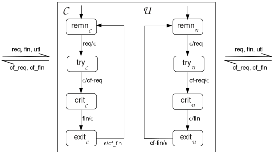


c_1 if player 1 wins.
c_2 if player 2 wins.
c_3 if it is a draw.
c_4 if the k-th position is empty.
c_5 if the '\$k'-th position is empty.

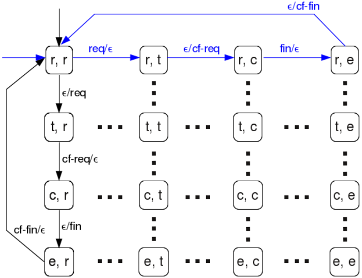
Two systems playing tic tac toe. System 1 makes the initial move.

Causal Relation Between the Input and Output of the Same System - Condition Based Restriction

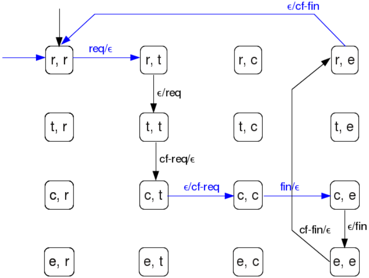
Example "Man in the Middle" in the Protocol of Mutual Exclusion



Unsynchronized product



Inner synchronized product



Condition Based Restriction

The interesting property of the condition based restriction is, that it may not affect the projection:
Def.: A projection π of an NFIOA $\mathcal{T}$ is called **unaffected** by a condition based restriction $cond_E$ if $\pi(\mathcal{T}) = \pi(cond_E(\mathcal{T}))$.

Def.: A condition based restricted automaton is called **consistent** if for each reachable state value either the acceptance condition is met or there is at least one continuation such that the acceptance condition can be met.

Coordination Semantics

The goal is to restrict the transition set of the weakly synchronized product automaton such that at least a quasi-determinism of the formerly nondeterministic transition set is achieved in a sense that from each reachable state there is at most one transition for each input character, including the empty character, while the factor NFIOAs (the roles) still can be regained by projection.

Transitions can be eliminated, if

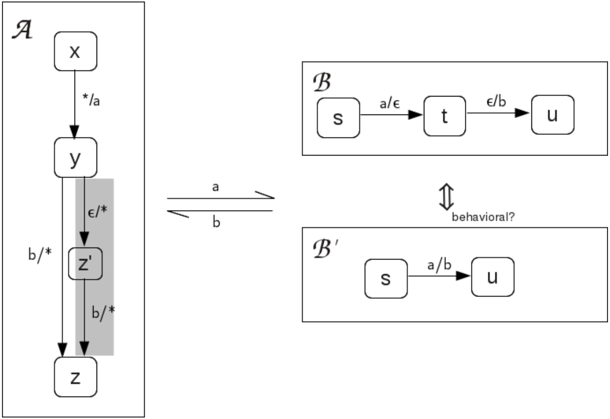
1. The projection of the restricted product automaton still results in the original NFIOAs.
2. The common acceptance condition still holds,
3. The elimination provides the coordination semantics of all related interactions.

Determination (elimination of ϵ-steps)

Some issues

- The set of state values might change, necessitating a restatement of the acceptance component.
- The projection relation between the restricted product automaton and its constituting roles gets lost.
- Several output characters might occur in one step.

More Issues with Determination



Synchronous processing: the caller (system $\mathcal{A}$) waits
Asynchronous processing: the caller meanwhile may do some intermediate steps.

Components

Frege Principle of Semantics

Two components are semantically equivalant, if they can be exchanged.

Components

Frege Principle of Semantics

Two components are semantically equivalant, if they can be exchanged.

What is with processes? The signature of processes are protocols.

Components

Frege Principle of Semantics

Two components are semantically equivalant, if they can be exchanged.

What is with processes? The signature of processes are protocols.
Thus, components representing processes need protocols in their signature.

Components

Frege Principle of Semantics

Two components are semantically equivalant, if they can be exchanged.

What is with processes? The signature of processes are protocols.
Thus, components representing processes need protocols in their signature.

In contrast to Claude Shannon: "The fundamental problem of communication is that the exchanged messages have meaning; that is they refer to or are correlated according to some system with certain physical or conceptual entities. These semantic aspects of communication are most relevant to the engineering problem."

Summary

- Process theory is more complex than I have thought.

Summary

- Process theory is more complex than I have thought.
- Network-like interactions are not deterministic, but are nondeterministic. The actions are not determined by the interactions!

Summary

- Process theory is more complex than I have thought.
- Network-like interactions are not deterministic, but are nondeterministic. The actions are not determined by the interactions!
- The openness of the interaction networks is represented by the incompleteness of the interaction descriptions in the sense of protocols with respect to the interaction coordinating processes.

Summary

- Process theory is more complex than I have thought.
- Network-like interactions are not deterministic, but are nondeterministic. The actions are not determined by the interactions!
- The openness of the interaction networks is represented by the incompleteness of the interaction descriptions in the sense of protocols with respect to the interaction coordinating processes.
- The causal relation between the input and output of different systems is represented by channel based restrictions.

Summary

- Process theory is more complex than I have thought.
- Network-like interactions are not deterministic, but are nondeterministic. The actions are not determined by the interactions!
- The openness of the interaction networks is represented by the incompleteness of the interaction descriptions in the sense of protocols with respect to the interaction coordinating processes.
- The causal relation between the input and output of different systems is represented by channel based restrictions.
- The causal relation between the input and output of the same system is represented by condition based restrictions.

Summary

- Process theory is more complex than I have thought.
- Network-like interactions are not deterministic, but are nondeterministic. The actions are not determined by the interactions!
- The openness of the interaction networks is represented by the incompleteness of the interaction descriptions in the sense of protocols with respect to the interaction coordinating processes.
- The causal relation between the input and output of different systems is represented by channel based restrictions.
- The causal relation between the input and output of the same system is represented by condition based restrictions.
- It are the spontaneous transitions which provide the freedom to combine different interactions to coordinating processes. So, protocol design strives for providing role descriptions which can be most easily coordinated with other roles.

Summary

- Process theory is more complex than I have thought.
- Network-like interactions are not deterministic, but are nondeterministic. The actions are not determined by the interactions!
- The openness of the interaction networks is represented by the incompleteness of the interaction descriptions in the sense of protocols with respect to the interaction coordinating processes.
- The causal relation between the input and output of different systems is represented by channel based restrictions.
- The causal relation between the input and output of the same system is represented by condition based restrictions.
- It are the spontaneous transitions which provide the freedom to combine different interactions to coordinating processes. So, protocol design strives for providing role descriptions which can be most easily coordinated with other roles.
- Determining quasi-deterministic processes raises new problems (asynchronous versus synchronous processing).

Summary

- Process theory is more complex than I have thought.
- Network-like interactions are not deterministic, but are nondeterministic. The actions are not determined by the interactions!
- The openness of the interaction networks is represented by the incompleteness of the interaction descriptions in the sense of protocols with respect to the interaction coordinating processes.
- The causal relation between the input and output of different systems is represented by channel based restrictions.
- The causal relation between the input and output of the same system is represented by condition based restrictions.
- It are the spontaneous transitions which provide the freedom to combine different interactions to coordinating processes. So, protocol design strives for providing role descriptions which can be most easily coordinated with other roles.
- Determining quasi-deterministic processes raises new problems (asynchronous versus synchronous processing).
- Components containing processes need protocols in their signature.

Thank You!

Any questions?

Johannes.Reich@sophoscape.de