Visual Formalisms Revisited

Radu Grosu
Technische Universität München

joint work with
Gheorge Stefanescu and Manfred Broy

Motivation

Interactive Applications

- An important domain of concern of SEng.
- Difficult to develop:
  - data
  - behavior
  - interconnection
  - architecture
  - distribution

Developer/Customer Interaction

- Successful communication between
  customer  software expert
  successful software development.

- Many modern SE Methods, like UML, ROOM and SDL recommend the use of visual formalisms.

Use of Visual Formalisms

• data	- E/R diagrams
• behavior	- statecharts
• intercommunication	- message sequence charts
• architecture	- data-flow diagrams
• distribution	- deployment diagrams

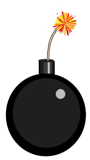
What are Visual Formalisms

- Directed graphs interpreted in a particular context.

- Intended to be compositional:
  - each node can be itself a graph
  - each node has a separate meaning

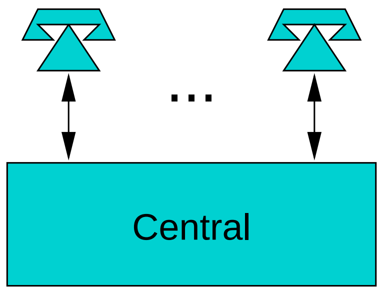
Problems

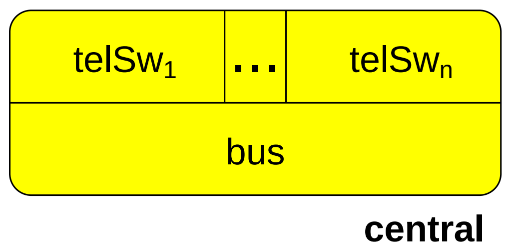
- No adequate hierarchic graphs model
- No clear denotational model

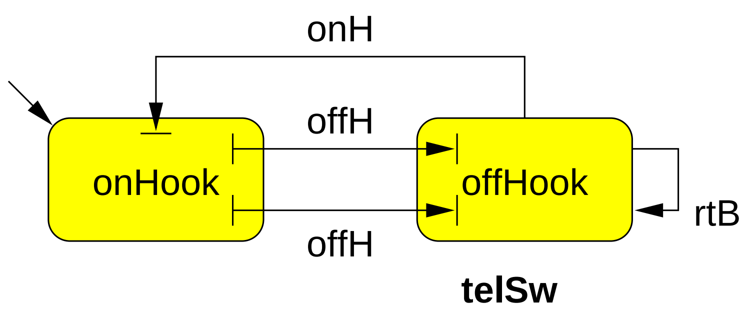


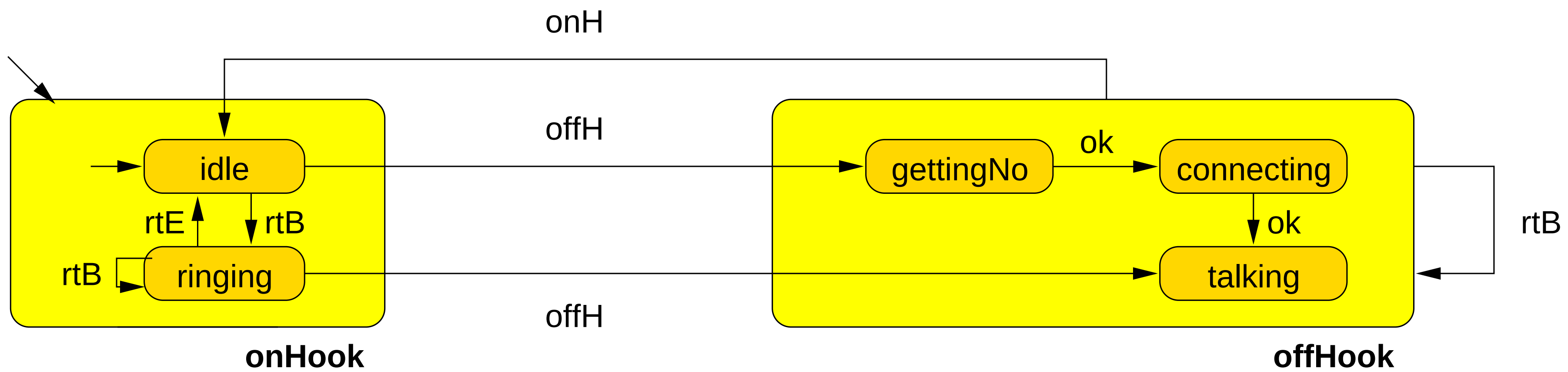
Severe consequences

A Telephone Central - with Statecharts

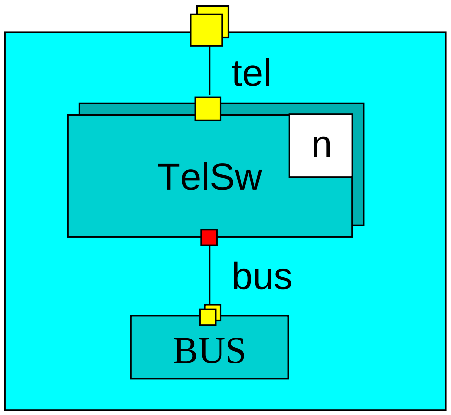




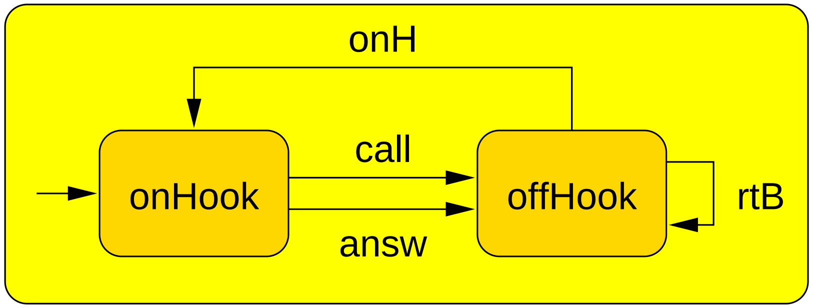




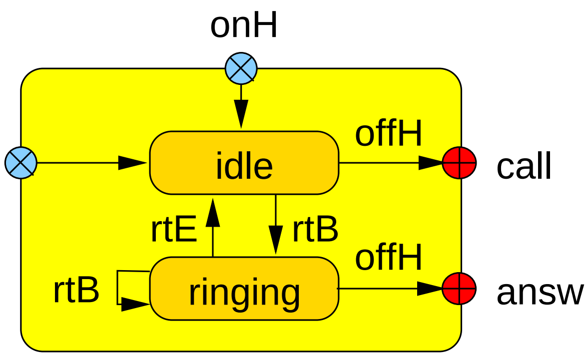
Telephone Central - Our Approach



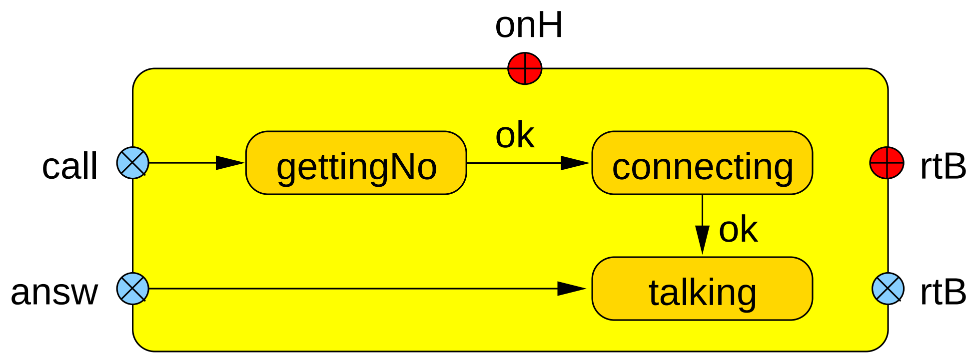
Central



telSw



onHook



offHook

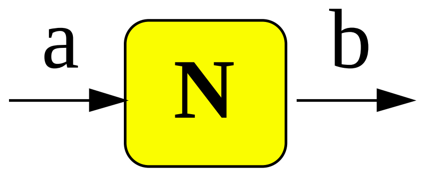
The Graphical Notation

Nodes and Arcs

Graph = a set of nodes connected by a set of arcs

Node interface = set of incoming/outgoing arcs

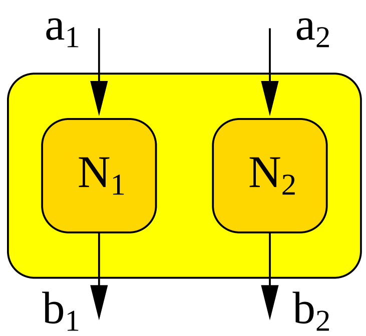
- arcs a denote types D_a



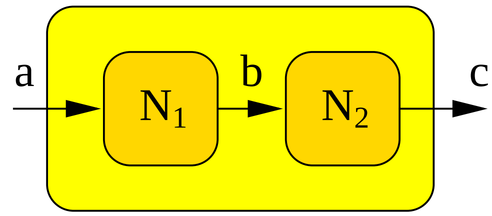
- nodes N denote relations $N \subseteq D_a \times D_b$

Graph Construction Primitives

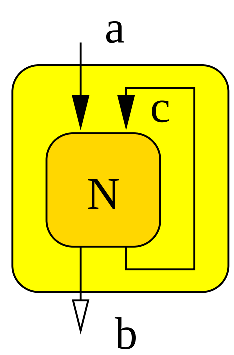
Operators on nodes



Visual attachment

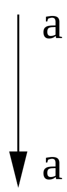


Sequential composition

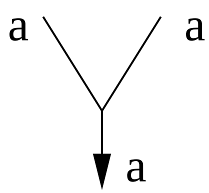


Feedback

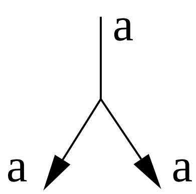
Connectors



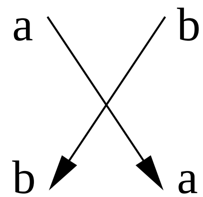
Identity



Identification

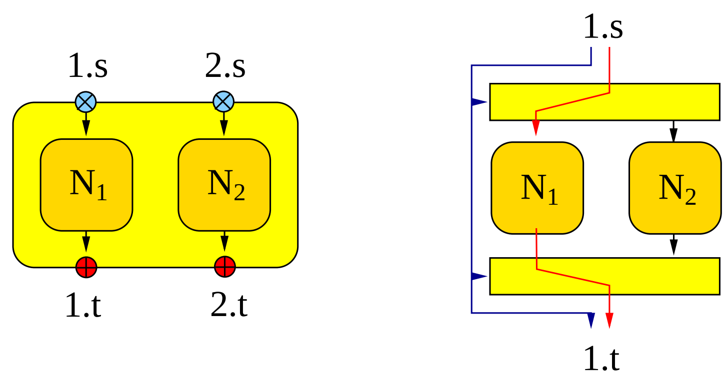


Ramification

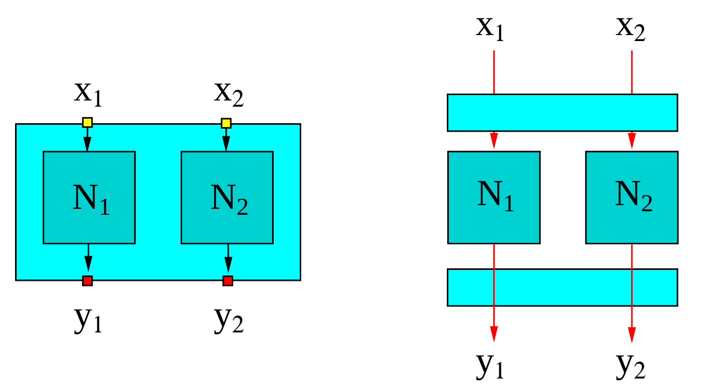


Transposition

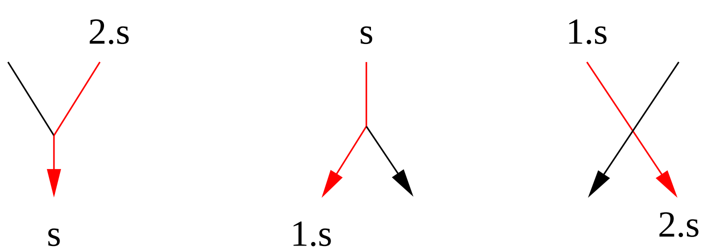
Additive and Multiplicative Interpretations



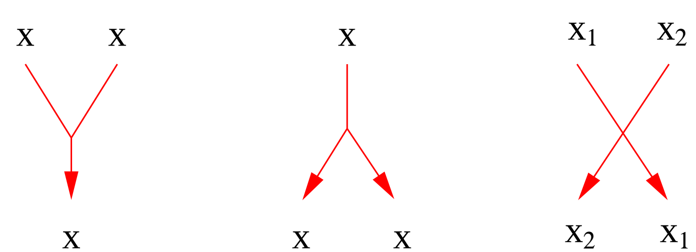
Additive (+) Interpretation of visual attachment



Multiplicative (x) interpretation of visual attachment



Additive (+) Interpretation of connectors

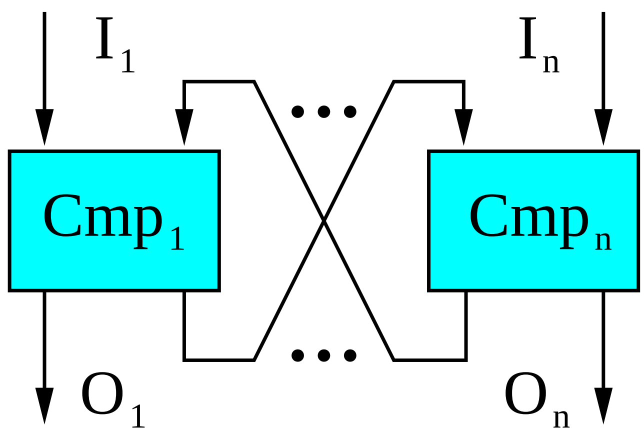


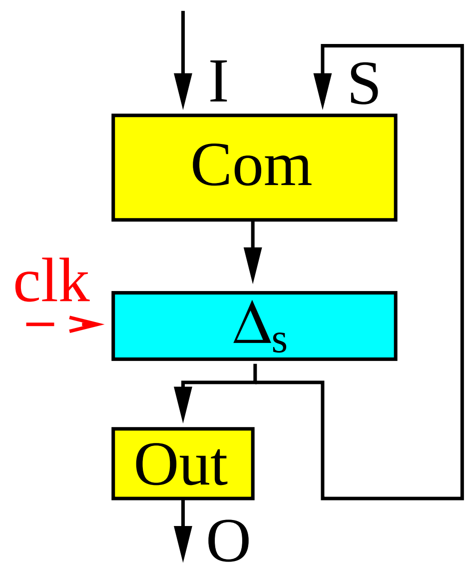
Multiplicative (x) interpretation of connectors

Computation Model

Interactive system = network of autonomous agents.

Agent = sequential machine.





$$Cmp(s) = (\ Com^* \ ; \Delta_S \ ; \ < \ ; (Out^* \times \mathrm{I}) \ ) \uparrow$$

Architecture Specification

Port Specification

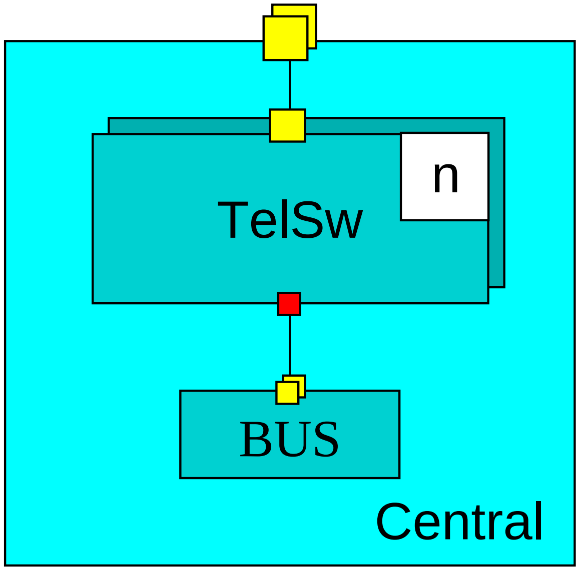
TelI = tk | onH | offH | dig(*I*)

TelO= tk | dtB | dtE | rtB | rtE | rbB | rbE | bsB | bsE

BusI = tk(*I*)| onH(*I*)| rtB(*I*)| rtE(*I*)| rbB(*I*)| rbE(*I*)| bsy(*I*)

BusO= BusI

Interconnection Specification



$$TelSw \in (TelI \times BusI)^N \to \wp(TelO \times BusO)^N$$

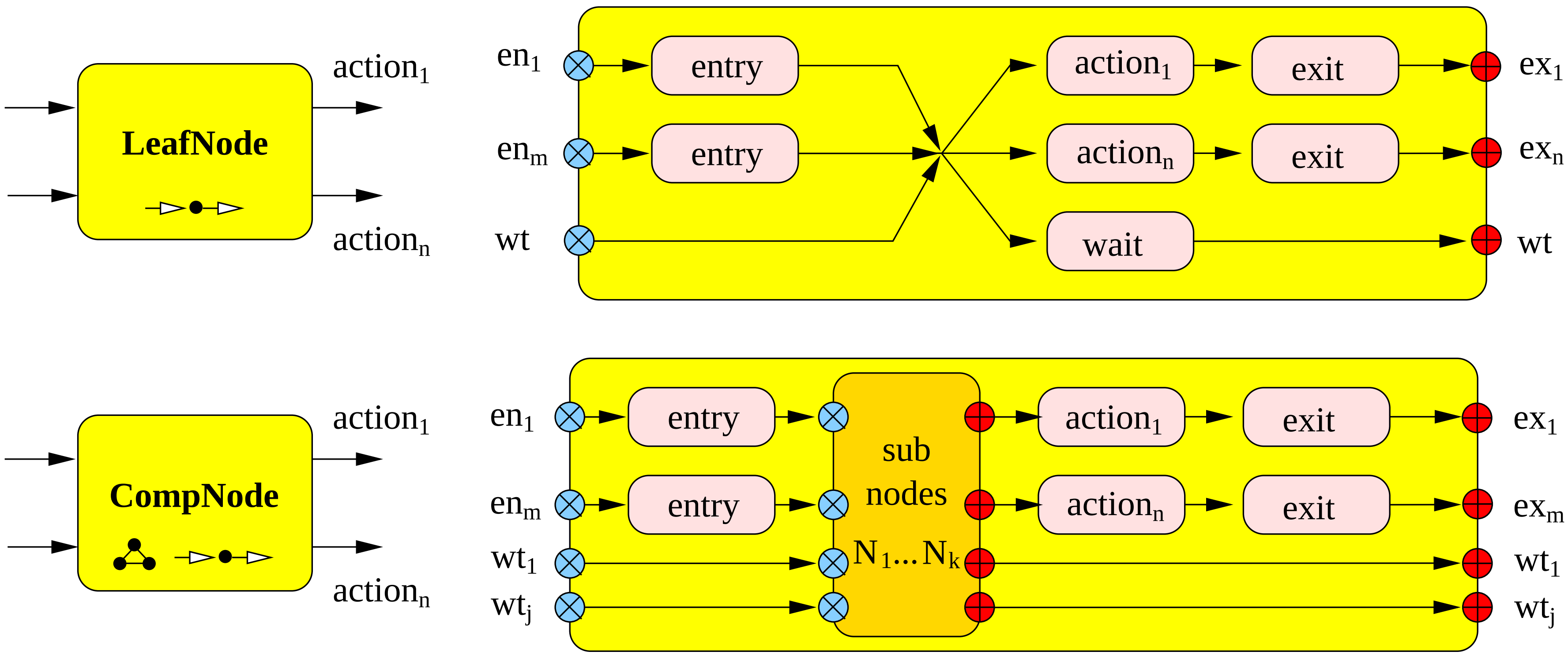
$$BUS \in (BusO^n)^N \to \wp((BusI^n)^N)$$

$$Central \in (TelI^n)^N \to \wp((TelO^n)^N)$$

$$Central = (\times_{i=1}^{n} TelSw) \otimes BUS$$

Component Specification

Leaf and Composed Nodes



$$Node \triangleq \Big(\mathop{+}_{i=1}^{m} (entry) + \mathrm{I} \Big);_{m+1} >;<_{n+1}; \Big(\mathop{+}_{i=1}^{n} (action_i ; exit) + wait \Big)$$

Actions

Action = relation between the current state and input and the next state: $a \subseteq (I \times S) \times S$

Specified by its characteristic predicate:

+ backprimed variables	- current input
+ plain variables	- current state
+ primed variables	- next state

Predefined Actions

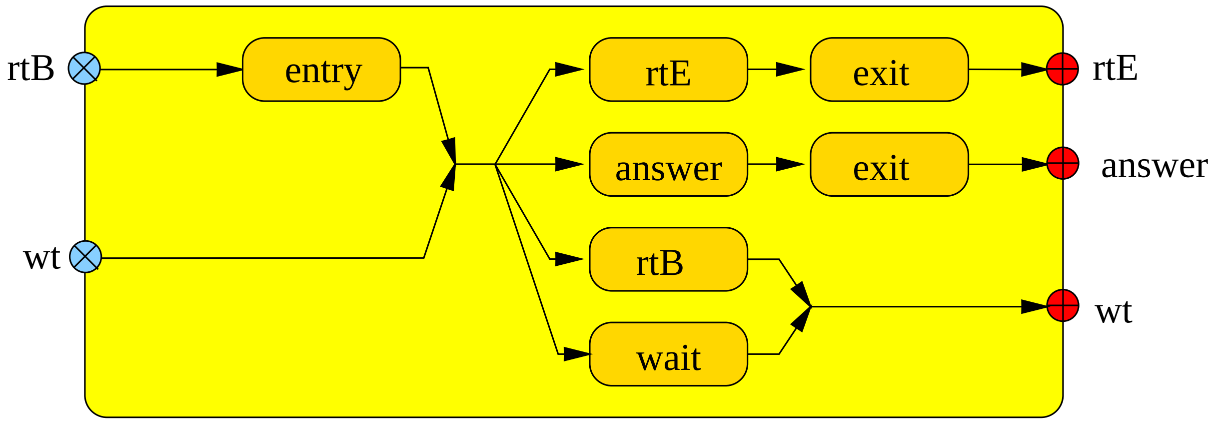
Events - modeled by togling boolean variables:

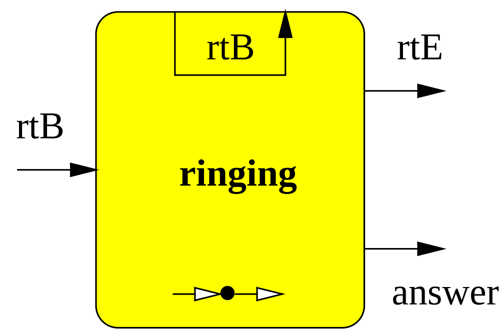
$e? \triangleq \text{‘}e \neq e \wedge e\text{’} = \text{‘}e$ $\qquad e! \triangleq e\text{’} = \neg e$

Message passing - modeled with pairs (e,m):

$e?a \triangleq e? \wedge \text{‘}m = a$ $\qquad e!a \triangleq e! \wedge m\text{’} = a$

The Leaf-Node ringing





$$entry \triangleq to!\mathrm{rtB}$$

$$rtE \triangleq bi?\mathrm{rtE}(nr)$$

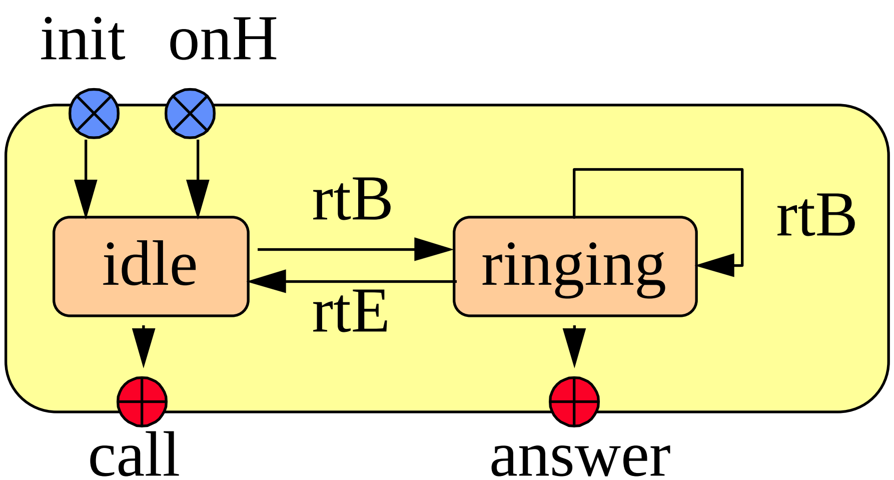
$$rtB \triangleq bi?\mathrm{rtB}(n) \wedge bo!\mathrm{bsy}(n)$$

$$exit \triangleq to!\mathrm{rtE}$$

$$answer \triangleq ti?\mathrm{offH}$$

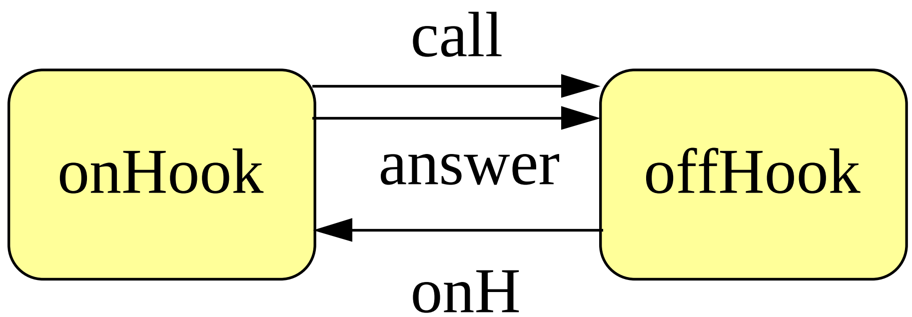
$$wait \triangleq \neg(ti? \vee bi?)$$

Hierarchical States



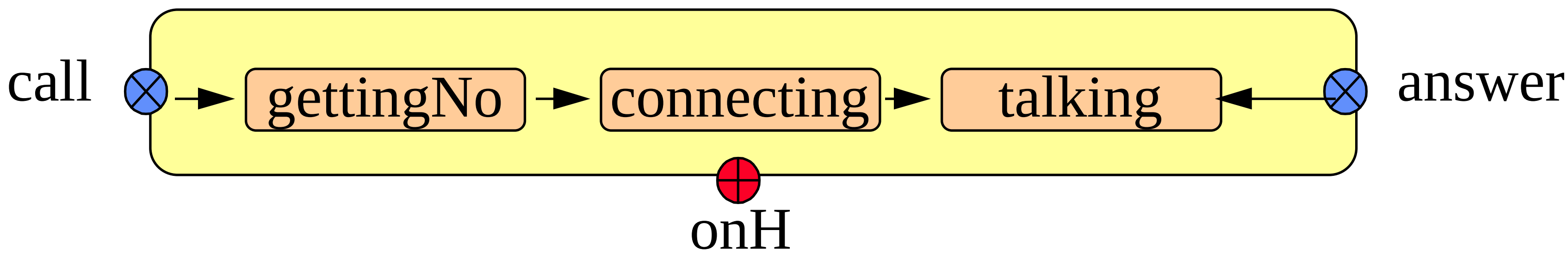
$$onHook \equiv idle \oplus ringing$$

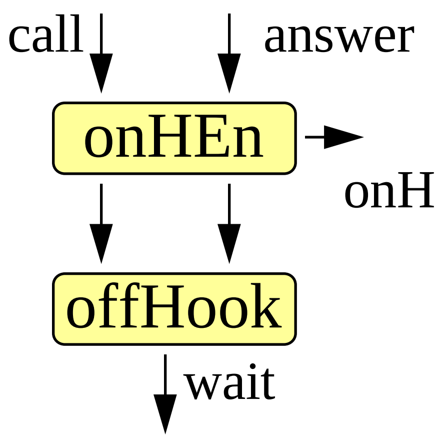
Transitions to Compound States



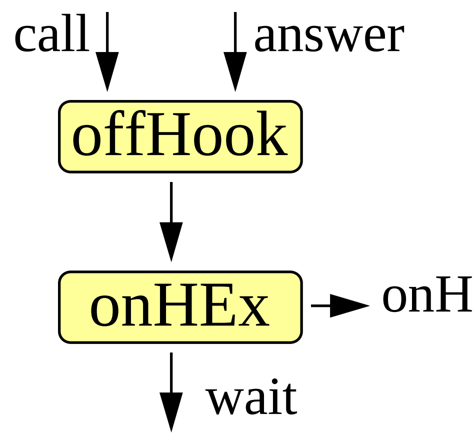
$$telSw \equiv onHook \oplus offHook$$

Strong and Weak Preemption



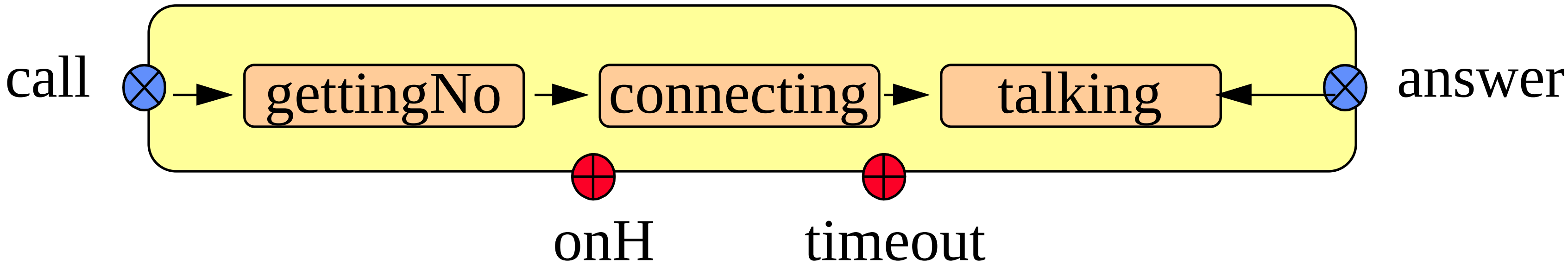


Strong preemption



Weak preemption

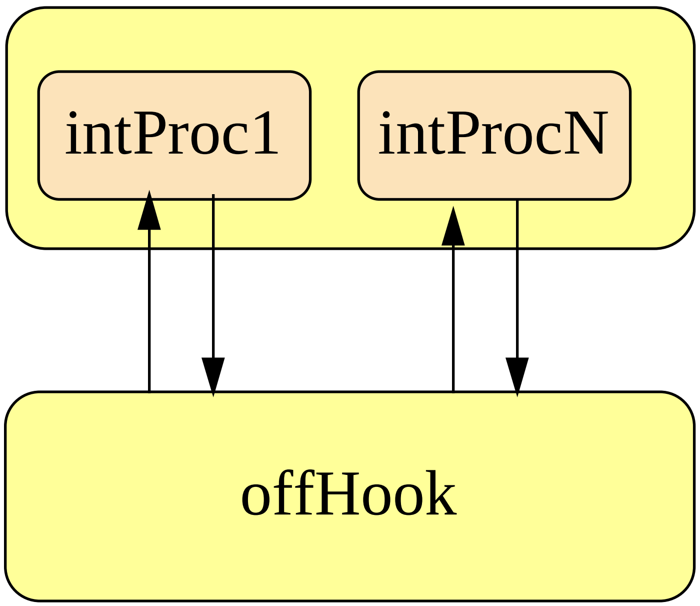
Entry/Exit Actions for Comp States



$$eOffHook \equiv 2\,\text{entry};\, pOffHook; 2\,\text{exit}$$

$$\text{entry} \equiv tmo!\mathrm{set}(60),\ \ \text{entry} \equiv tmo!reset$$

History Variables



$$admin \triangleq \sum_{i=1}^{N} intProc_i$$

Conclusions

We showed how to combine modular specifications of control and data-flow.

Practical relevance:

- ⇒ clear foundation for execution-tools,
- ⇒ basis for prototyping and visual transformation,
- ⇒ basis for verification and optimization.

Theoretical relevance:

- ⇒ semantics of interaction as mixed graph algebras,
- ⇒ model for linear and linear temporal logic.