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 and software expert leads to 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
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.

\[ Cmp(s) = (\text{Com}^*; \Delta_s; <; (\text{Out}^* \times I)) \uparrow \]
Architecture Specification
Port Specification

\[
\text{TelI} = tk \mid \text{onH} \mid \text{offH} \mid \text{dig}(I)
\]

\[
\text{TelO} = tk \mid \text{dtB} \mid \text{dtE} \mid \text{rtB} \mid \text{rtE} \mid \text{rbB} \mid \text{rbE} \mid \text{bsB} \mid \text{bsE}
\]

\[
\text{BusI} = tk(I) \mid \text{onH}(I) \mid \text{rtB}(I) \mid \text{rtE}(I) \mid \text{rbB}(I) \mid \text{rbE}(I) \mid \text{bsy}(I)
\]

\[
\text{BusO} = \text{BusI}
\]
Interconnection Specification

\[ \text{TelSw} \in (\text{TelI} \times \text{BusI})^N \rightarrow \varnothing(\text{TelO} \times \text{BusO})^N \]

\[ \text{BUS} \in (\text{BusO}^n)^N \rightarrow \varnothing((\text{BusI}^n)^N) \]

\[ \text{Central} \in (\text{TelI}^n)^N \rightarrow \varnothing((\text{TelO}^n)^N) \]

\[ \text{Central} = (\bigotimes_{i=1}^n \text{TelSw}) \otimes \text{BUS} \]
Component Specification
Leaf and Composed Nodes

\[
\text{Node} := \bigoplus_{i=1}^{m} ((entry) + \text{I}) \oplus_{i=1}^{m+1} \ldots \oplus_{i=1}^{n+1} ((action_i; exit) + \text{wait})
\]
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:
\[ e? \triangleq \ 'e \neq e \land e' = 'e \quad e! \triangleq e' = \neg e \]

Message passing - modeled with pairs \((e,m)\):
\[ e?a \triangleq e? \land 'm = a \quad e!a \triangleq e! \land m' = a \]
The Leaf-Node **ringing**

\[
\begin{align*}
\text{entry} & \triangleq to!rtB \\
rtE & \triangleq bi ?rtE(nr) \\
rtB & \triangleq bi ?rtB(n) \land bo!bsy(n) \\
\text{exit} & \triangleq to!rtE \\
\text{answer} & \triangleq ti ?offH \\
\text{wait} & \triangleq \neg(ti ?\lor bi ?)
\end{align*}
\]
Hierarchical States

\[ \text{onHook} \equiv \text{idle} \oplus \text{ringing} \]
Transitions to Compound States

telSw ≡ onHook ⊕ offHook
Strong and Weak Preemption

![Diagram showing the states of a call with strong and weak preemption]

**Strong preemption**
- Call
- Getting No
- Connecting
- Talking
- OnH

**Weak preemption**
- Call
- OnHEN
- OffHook
- Wait
- OnH

- Call
- OffHook
- OnHEX
- Wait
- OnH
Entry/Exit Actions for Comp States

\[
e^{\cdot} \text{OffHook} \equiv 2 \text{entry}; p^{\cdot} \text{OffHook}; 2 \text{exit}
\]

\[
\text{entry} \equiv tmo!\text{set}(60), \ \text{entry} \equiv tmo!\text{reset}
\]
History Variables

\[
\text{admin} \overset{\Delta}{=} \sum_{i=1}^{N} \text{intProc}_i
\]
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.