EPC Modelling based on Implicit Arc Types

Jan Mendling, Markus Nüttgens, Universität Trier, IS 2
Contents

1. Event-driven Process Chains (EPCs)
2. Explicit and Implicit Types
3. Type Consistency with Implicit Types
4. Validation Scenarios
5. Future Work
1. EPCs (I)

1. EPCs

Design Process

Requirements Engineering

Design Process

EPC Symbols

Event

Function

Process Interface

Connectors

Control Flow Arc

Start

List requirements

Cannot be fulfilled

Can be fulfilled

To Design Process

From Requirements Engineering

Can be fulfilled

Design

Design finished

Jan Mendling, Markus Nüttgens, Universität Trier
1. EPCs (II)

- Conceptual Modelling of Business Processes

- Advantages:
  - Intuitive Understanding
  - Used by SAP for Reference Modelling
  - Supported by major BPM Tools

- Disadvantages:
  - Syntax Check for Connector Chains complicated
1. EPCs (III)

Informal Syntax Rules (Type Consistency)

- Events are followed by Functions or ProcessInterfaces
- Functions are followed by Events
- ProcessInterfaces are followed by Events
- There may be an arbitrary number of connectors in between

→ Chains of consecutive connectors demand transitive type check
### 2. Explicit and Implicit Types

#### (I)

<table>
<thead>
<tr>
<th>Explicit Element Type</th>
<th>Implicit Element Types</th>
</tr>
</thead>
</table>
| Event $E$             | $E_S$  
|                       | $E_{int}$  
|                       | $E_E$  |
| Function $F$          | $F$  |
| Process Interface $P$ | $P_S$  
|                       | $P_E$  |
| Connector AND         | $E_{FS}$  
|                       | $E_{EFJ}$  
|                       | $E_{FES}$  
|                       | $E_{FEJ}$  |
| Connector OR          | $E_{EFJ}$  
|                       | $E_{FES}$  
|                       | $E_{FEJ}$  |
| Connector XOR         | $E_{EFJ}$  
|                       | $E_{FES}$  
|                       | $E_{FEJ}$  |
2. Explicit and Implicit Types (II)

## Two kinds of Arcs:
- FunctionEventArcs
- EventFunctionArcs

<table>
<thead>
<tr>
<th>FROM</th>
<th>TO (above)</th>
<th>E₅</th>
<th>E₁₅</th>
<th>E₁</th>
<th>F</th>
<th>Pₛ</th>
<th>Pₑ</th>
<th>ANDₑₛ</th>
<th>ANDₑₑ</th>
<th>ANDₑₑ</th>
<th>ORₑₑ</th>
<th>ORₑₑ</th>
<th>ORₑₑ</th>
<th>XORₑₑ</th>
<th>XORₑₑ</th>
</tr>
</thead>
<tbody>
<tr>
<td>E₅</td>
<td>→</td>
<td>→</td>
<td>→</td>
<td>→</td>
<td>→</td>
<td>→</td>
<td>→</td>
<td>→</td>
<td>→</td>
<td>→</td>
<td>→</td>
<td>→</td>
<td>→</td>
<td>→</td>
<td>→</td>
</tr>
<tr>
<td>E₁₅</td>
<td>→</td>
<td>→</td>
<td>→</td>
<td>→</td>
<td>→</td>
<td>→</td>
<td>→</td>
<td>→</td>
<td>→</td>
<td>→</td>
<td>→</td>
<td>→</td>
<td>→</td>
<td>→</td>
<td>→</td>
</tr>
<tr>
<td>E₁</td>
<td>→</td>
<td>→</td>
<td>→</td>
<td>→</td>
<td>→</td>
<td>→</td>
<td>→</td>
<td>→</td>
<td>→</td>
<td>→</td>
<td>→</td>
<td>→</td>
<td>→</td>
<td>→</td>
<td>→</td>
</tr>
<tr>
<td>F</td>
<td>→</td>
<td>→</td>
<td>→</td>
<td>→</td>
<td>→</td>
<td>→</td>
<td>→</td>
<td>→</td>
<td>→</td>
<td>→</td>
<td>→</td>
<td>→</td>
<td>→</td>
<td>→</td>
<td>→</td>
</tr>
<tr>
<td>Pₛ</td>
<td>→</td>
<td>→</td>
<td>→</td>
<td>→</td>
<td>→</td>
<td>→</td>
<td>→</td>
<td>→</td>
<td>→</td>
<td>→</td>
<td>→</td>
<td>→</td>
<td>→</td>
<td>→</td>
<td>→</td>
</tr>
<tr>
<td>Pₑ</td>
<td>→</td>
<td>→</td>
<td>→</td>
<td>→</td>
<td>→</td>
<td>→</td>
<td>→</td>
<td>→</td>
<td>→</td>
<td>→</td>
<td>→</td>
<td>→</td>
<td>→</td>
<td>→</td>
<td>→</td>
</tr>
<tr>
<td>ANDₑₑ</td>
<td>→</td>
<td>→</td>
<td>→</td>
<td>→</td>
<td>→</td>
<td>→</td>
<td>→</td>
<td>→</td>
<td>→</td>
<td>→</td>
<td>→</td>
<td>→</td>
<td>→</td>
<td>→</td>
<td>→</td>
</tr>
<tr>
<td>ANDₑₑ</td>
<td>→</td>
<td>→</td>
<td>→</td>
<td>→</td>
<td>→</td>
<td>→</td>
<td>→</td>
<td>→</td>
<td>→</td>
<td>→</td>
<td>→</td>
<td>→</td>
<td>→</td>
<td>→</td>
<td>→</td>
</tr>
<tr>
<td>ANDₑₑ</td>
<td>→</td>
<td>→</td>
<td>→</td>
<td>→</td>
<td>→</td>
<td>→</td>
<td>→</td>
<td>→</td>
<td>→</td>
<td>→</td>
<td>→</td>
<td>→</td>
<td>→</td>
<td>→</td>
<td>→</td>
</tr>
<tr>
<td>ORₑₑ</td>
<td>→</td>
<td>→</td>
<td>→</td>
<td>→</td>
<td>→</td>
<td>→</td>
<td>→</td>
<td>→</td>
<td>→</td>
<td>→</td>
<td>→</td>
<td>→</td>
<td>→</td>
<td>→</td>
<td>→</td>
</tr>
<tr>
<td>ORₑₑ</td>
<td>→</td>
<td>→</td>
<td>→</td>
<td>→</td>
<td>→</td>
<td>→</td>
<td>→</td>
<td>→</td>
<td>→</td>
<td>→</td>
<td>→</td>
<td>→</td>
<td>→</td>
<td>→</td>
<td>→</td>
</tr>
<tr>
<td>ORₑₑ</td>
<td>→</td>
<td>→</td>
<td>→</td>
<td>→</td>
<td>→</td>
<td>→</td>
<td>→</td>
<td>→</td>
<td>→</td>
<td>→</td>
<td>→</td>
<td>→</td>
<td>→</td>
<td>→</td>
<td>→</td>
</tr>
<tr>
<td>XORₑₑ</td>
<td>→</td>
<td>→</td>
<td>→</td>
<td>→</td>
<td>→</td>
<td>→</td>
<td>→</td>
<td>→</td>
<td>→</td>
<td>→</td>
<td>→</td>
<td>→</td>
<td>→</td>
<td>→</td>
<td>→</td>
</tr>
<tr>
<td>XORₑₑ</td>
<td>→</td>
<td>→</td>
<td>→</td>
<td>→</td>
<td>→</td>
<td>→</td>
<td>→</td>
<td>→</td>
<td>→</td>
<td>→</td>
<td>→</td>
<td>→</td>
<td>→</td>
<td>→</td>
<td>→</td>
</tr>
<tr>
<td>XORₑₑ</td>
<td>→</td>
<td>→</td>
<td>→</td>
<td>→</td>
<td>→</td>
<td>→</td>
<td>→</td>
<td>→</td>
<td>→</td>
<td>→</td>
<td>→</td>
<td>→</td>
<td>→</td>
<td>→</td>
<td>→</td>
</tr>
<tr>
<td>XORₑₑ</td>
<td>→</td>
<td>→</td>
<td>→</td>
<td>→</td>
<td>→</td>
<td>→</td>
<td>→</td>
<td>→</td>
<td>→</td>
<td>→</td>
<td>→</td>
<td>→</td>
<td>→</td>
<td>→</td>
<td>→</td>
</tr>
</tbody>
</table>

Jan Mendling, Markus Nüttgens,  
Universität Trier
3. Type Consistency with Implicit Types

3. Type Consistency with Implicit Types (I)

Event Types (from) $ET_{\text{from}} = E_S \cup E_{\text{int}} \cup \text{AND}_{\text{EFS}} \cup \text{AND}_{\text{EFJ}} \cup \text{OR}_{\text{EFJ}} \cup \text{XOR}_{\text{EFJ}}$

Function Types (from) $FT_{\text{from}} = F \cup P_S \cup \text{AND}_{\text{EFS}} \cup \text{AND}_{\text{EFJ}} \cup \text{OR}_{\text{EFS}} \cup \text{OR}_{\text{EFJ}} \cup \text{XOR}_{\text{EFS}} \cup \text{XOR}_{\text{EFJ}}$

Event Types (to) $ET_{\text{to}} = E_{\text{int}} \cup E_E \cup \text{AND}_{\text{EFS}} \cup \text{AND}_{\text{EFJ}} \cup \text{OR}_{\text{EFJ}} \cup \text{XOR}_{\text{EFJ}}$

Function Types (to) $FT_{\text{to}} = F \cup P_E \cup \text{AND}_{\text{EFS}} \cup \text{AND}_{\text{EFJ}} \cup \text{OR}_{\text{EFJ}} \cup \text{XOR}_{\text{EFJ}}$

FunctionEventArcs $FEA \subseteq (FT_{\text{from}} \times ET_{\text{to}})$

EventFunctionArcs $EFA \subseteq (ET_{\text{from}} \times FT_{\text{to}})$

The precondition of a vertex is made up by the set of ancestor arcs written as $\rightarrow v := \{(x,v) \in A\} \text{ with } v,x \in V$.

The postcondition of a vertex is defined as the set of descending arcs: $v\rightarrow := \{(v,x) \in A\} \text{ with } v,x \in V$. 

Jan Mendling, Markus Nüttgens, Universität Trier
3. Type Consistency with Implicit Types

Concerning vertex types the following type consistency constraints apply:

1. Start Events: \( \forall v \in E_S: v \rightarrow \subseteq EFA. \)
2. Inner Events: \( \forall v \in E_{\text{Int}}: \rightarrow v \subseteq FEA \) and \( v \rightarrow \subseteq EFA. \)
3. End Events: \( \forall v \in E_E: \rightarrow v \subseteq FEA. \)
4. Start ProcessInterface: \( \forall v \in P_S: v \rightarrow \subseteq FEA. \)
5. End ProcessInterface: \( \forall v \in P_E: \rightarrow v \subseteq EFA. \)
6. Function: \( \forall v \in F: \rightarrow v \subseteq EFA \) and \( v \rightarrow \subseteq FEA. \)
7. Event-Function-Connects: \( \forall v \in AND_{EFS} \cup AND_{EFJ} \cup OR_{EFJ} \cup XOR_{EFJ}: \rightarrow v \in EFA \) and \( v \rightarrow \in EFA. \)
8. Function-Event-Connects: \( \forall v \in AND_{FES} \cup AND_{FEJ} \cup OR_{FES} \cup OR_{FEJ} \cup XOR_{FES} \cup XOR_{FEJ}: \rightarrow v \in FEA \) and \( v \rightarrow \in FEA. \)
4. Validation Scenarios

- Guided Modelling
  - keep book on possible implicit types of an explicit symbol while modelling in a set called $\Pi$ for each symbol instance.
  - When there is a new arc drawn, update $\Pi$ of the explicit symbol and arc instances involved (intersect $\Pi$ with the set of consistent types to that arc)
  - If there is a syntax error, you get an empty set in the elements or arcs concerned.

- Model Checking
  - establish a mapping from a model with only explicit types to a model with implicit types
  - use type consistency rules
5. Future Work

- Usage of implicit element and arc types in EPC Markup Language (EPML)
- Type consistency can be expressed with Schematron