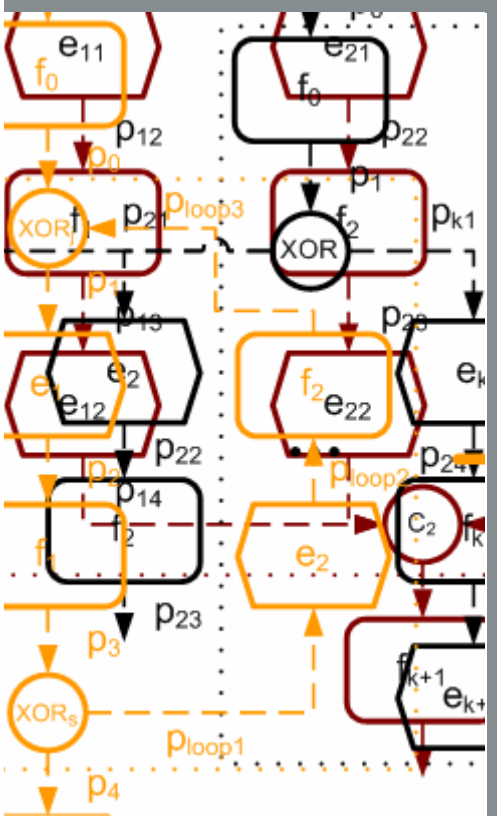


Reducing Complexity of Large EPCs

Artem Polyvyanyy
Sergey Smirnov
Mathias Weske

EPK 2008 27 November 2008

Motivation

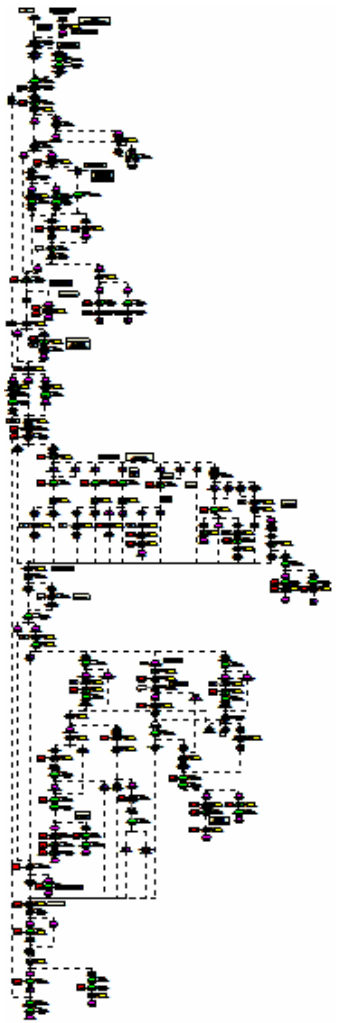
Research project with AOK Brandenburg

Goal: detailed process models ➞ abstract process models

- ≈ 4 000 EPCs
- graph-structured process models
- model elements have annotations:
  - connections → probabilities
  - functions → average execution time

average execution time of a process *must be preserved*

Motivation



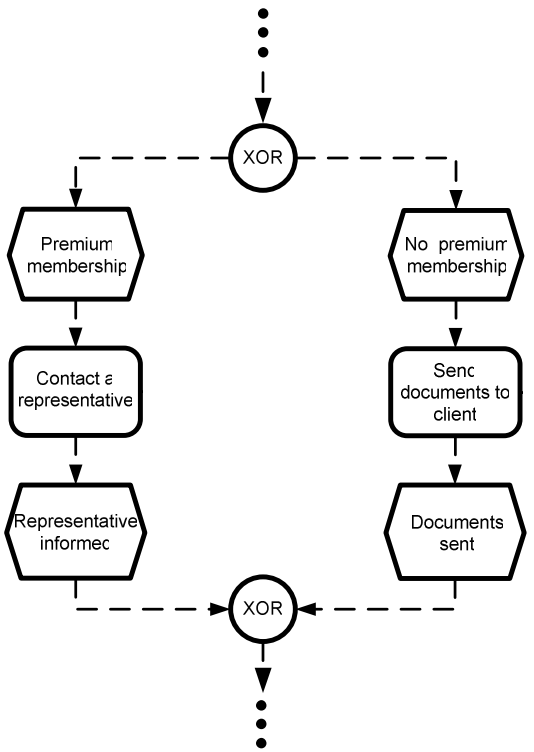
Example model:

- > 300 nodes
- >150 functions
- graph-structured model

Extended EPC

$(E, F, C, A, t, e_r, p_r)$ is an *extended EPC*

- E is a set of events, $E \neq \emptyset$
- F is a set of functions, $F \neq \emptyset$
- C is a set of connectors
- $N = E \cup F \cup C$ is a set of nodes
- $A \subseteq N \times N$ is a set of connections
- $t : C \rightarrow \{and, or, xor\}$
- $e_r : F \rightarrow \mathbb{R}^+$
- $p_r : A \rightarrow [0, 1]$



Mean occurrence number of a node

the mean number of node occurrence in a process instance

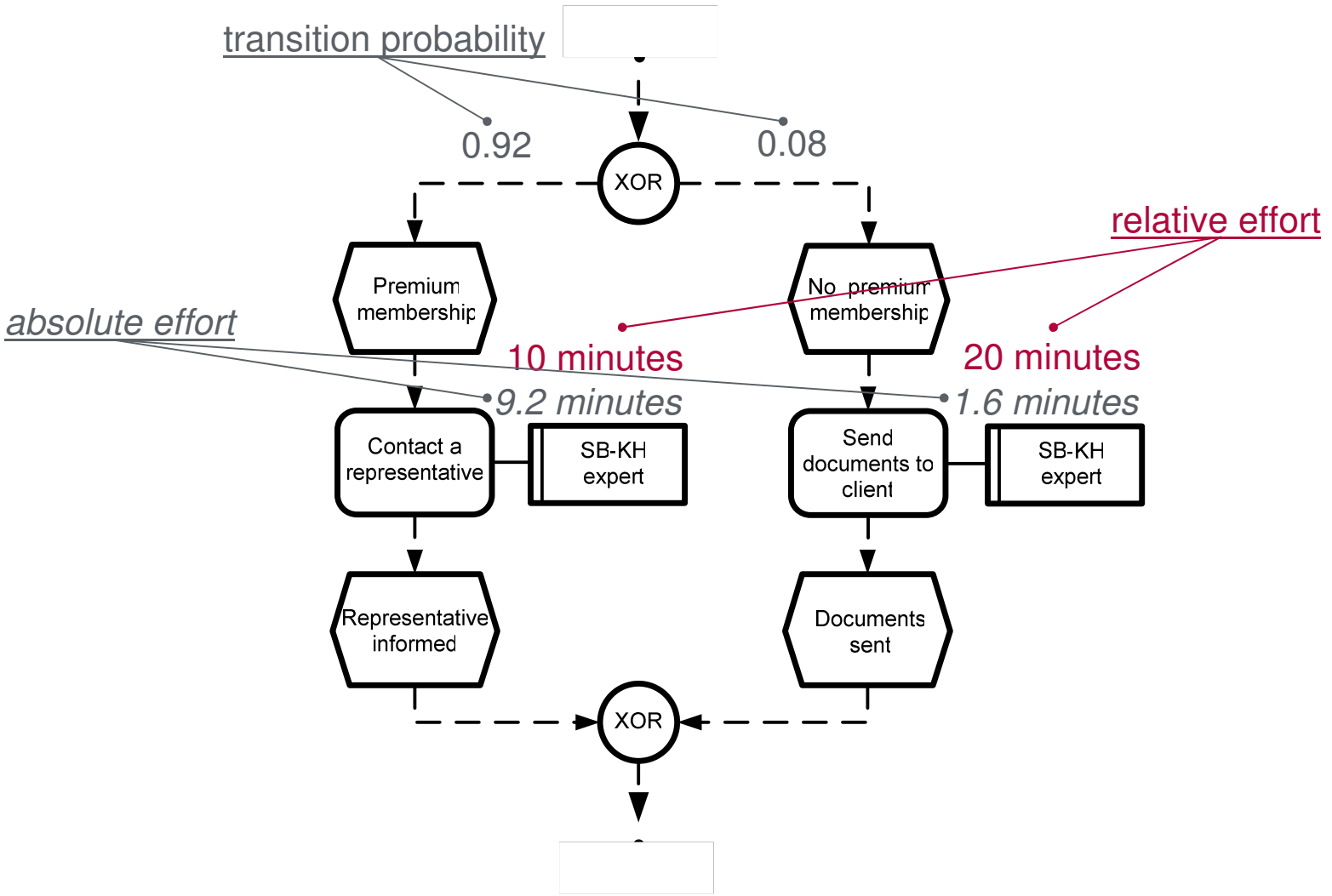
Absolute effort of a function e_a

the mean effort contributed to the execution of the function in a process instance

Process absolute effort e^p_a

the mean effort required to execute a process instance

Example


transition probability
0.92
0.08
XOR
relative effort
absolute effort
Premium membership
No premium membership
10 minutes
20 minutes
9.2 minutes
1.6 minutes
Contact a representative
SB-KH expert
Send documents to client
SB-KH expert
Representative informed
Documents sent
XOR

What

model elements are insignificant?

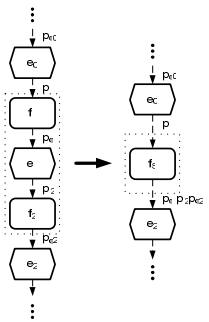
- *absolute effort of a function* e_a

How

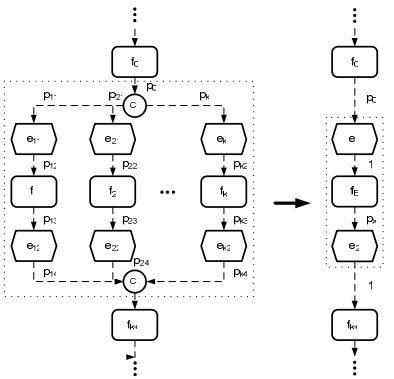
to abstract insignificant elements?

- elementary abstractions
  transformation of a fragment
- abstraction strategy
  organization of elementary abstractions

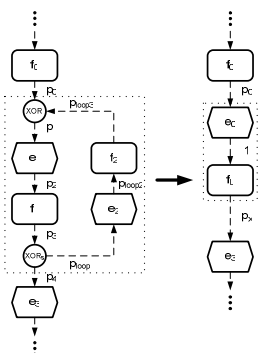
Elementary Abstractions



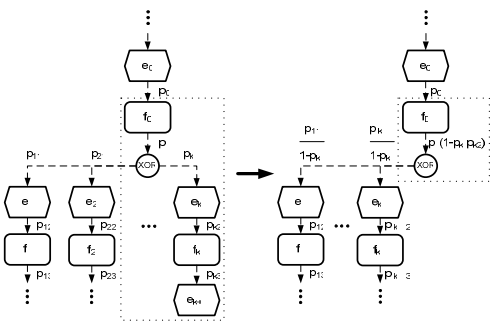
Sequence



Block

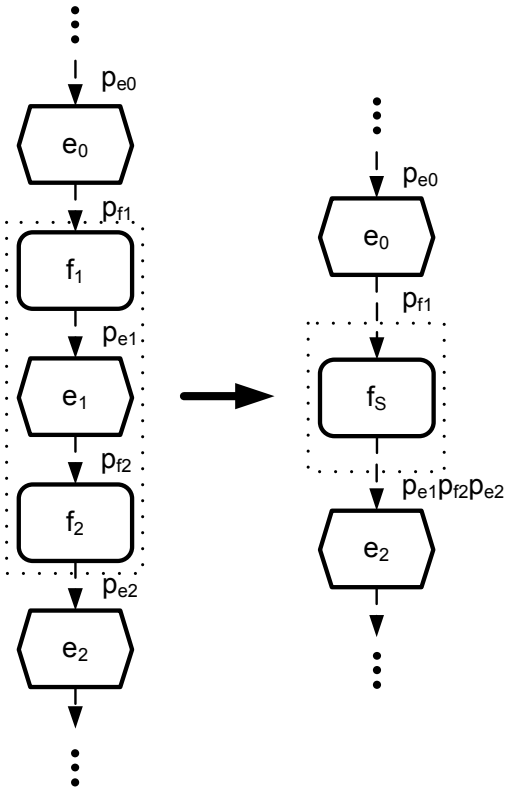


Loop



Dead end

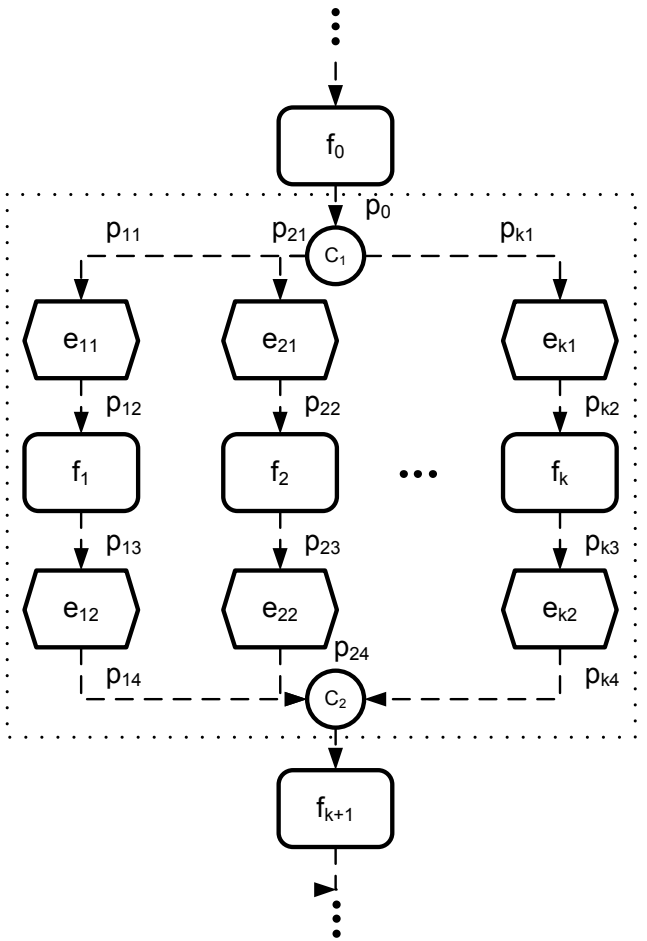
Sequence Elementary Abstraction

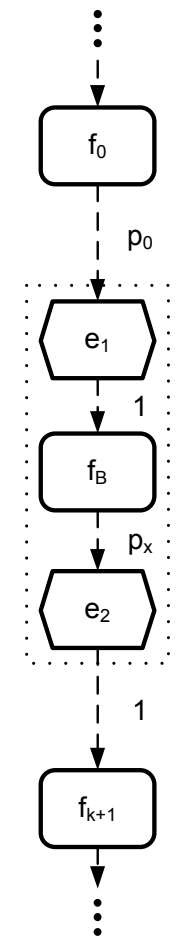


$$e_r(f_S) = e_r(f_1) + e_r(f_2) \cdot p_r((f_1, e_1)) \cdot p_r((e_1, f_2))$$

$$p_r((f_S, e_2)) = p_r(f_1, e_1) \cdot p_r(e_1, f_2) \cdot p_r(f_2, e_2)$$

Block Elementary Abstraction





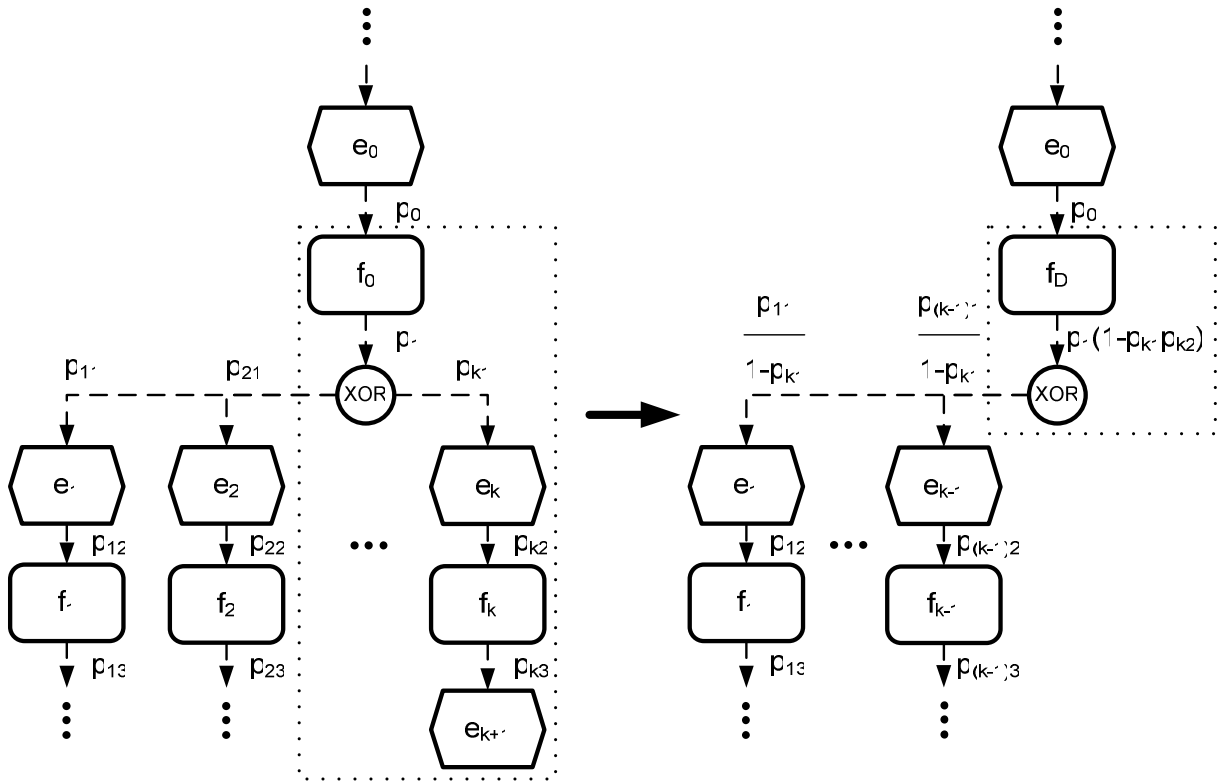
(∧) $p_r((f_B, e_2)) = \prod_{i=1}^{k} p_i$

(XOR) $p_r((f_B, e_2)) = \sum_{i=1}^{k} p_i$

(∨) !

$$e_r(f_B) = \sum_{i=1}^{k} e_r(f_i) \cdot p_r((c_1, e_{i1})) \cdot p_r((e_{i1}, f_i))$$

Dead End Elementary Abstraction

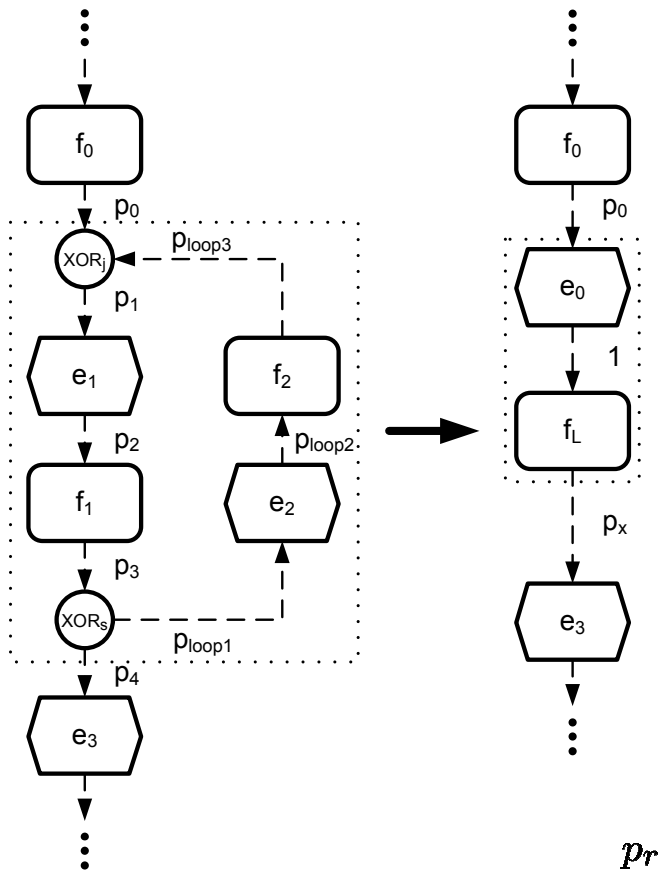


$$e_r(f_D) = e_r(f_0) + e_r(f_k) \cdot p_r((f_0, xor)) \cdot p_r((xor, e_k)) \cdot p_r((e_k, f_k))$$

$$p_r((e_0, f_D)) = p_r((e_0, f_0)) \qquad p_r((f_D, xor)) = p_r((f_0, xor)) \cdot (1 - p_r((xor, e_k)) \cdot p_r((e_k, f_k)))$$

$$p'_r((xor, e_i)) = \frac{p_r((xor, e_i))}{1 - p_r((xor, e_k))}$$

Loop Elementary Abstraction



$$e_r(f_L) = p_r((xor_j, e_1)) \cdot p_r((e_1, f_1)) \cdot \frac{1}{1-p} \cdot (e_r(f_1) + + e_r(f_2) \cdot p_r((e_1, xor_s)) \cdot p_l \cdot p_r((e_2, f_2))),$$

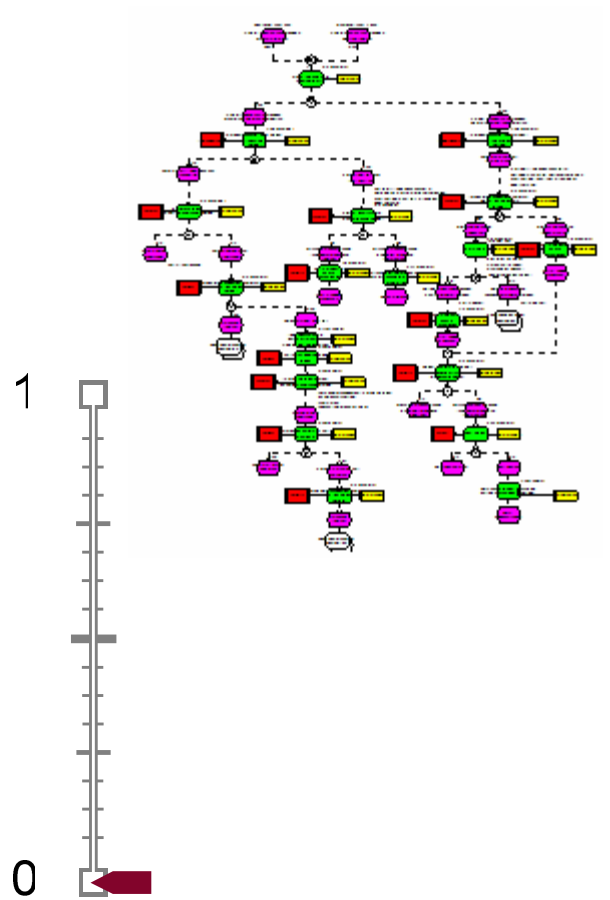
where $p = p_r((xor_j, e_1)) \cdot p_r((e_1, f_1)) \cdot p_r((e_1, xor_s)) \cdot p_r((e_2, f_2)) \cdot p_r((f_2, xor_j)) \cdot p_l$ and $p_l = p_r((xor_s, e_2))$

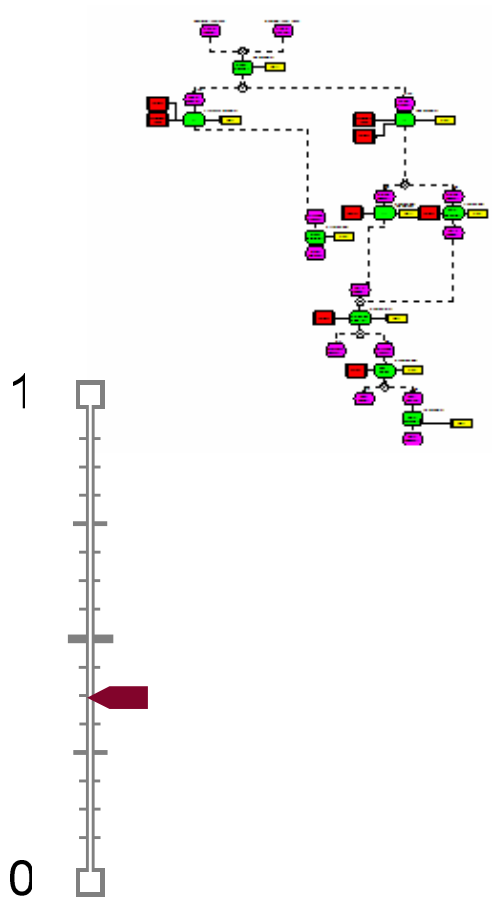
$$p_r((f_L, e_3)) = 1 - \frac{1}{1-p} \cdot (p_{stop}^{path} + (1 - p_{stop}^{path}) \cdot p_l \cdot p_{stop}^{loop}).$$

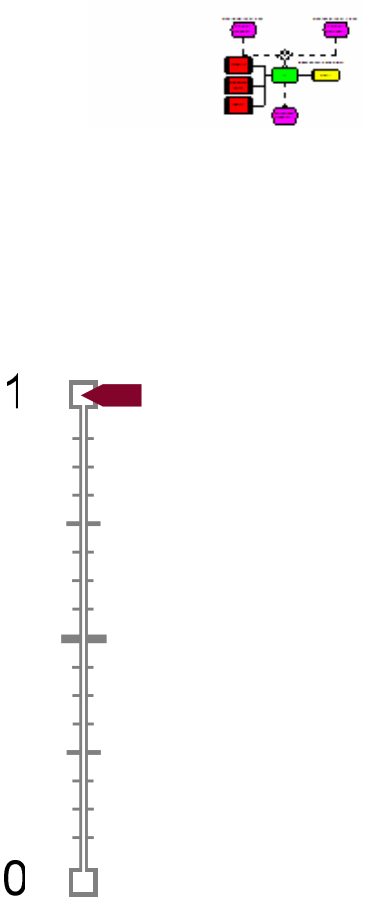
$$p_{stop}^{path} = 1 - p_r((xor_j, e_1)) \cdot p_r((e_1, f_1)) \cdot p_r((f_1, xor_s))$$

$$p_{stop}^{loop} = 1 - p_r((e_2, f_2)) \cdot p_r((f_2, xor_j))$$

Abstraction Example







Conclusions

Conceptualized the problem of

business process model abstraction

Proposed the abstraction approach based on rules describing

- structural transformations
- dependencies between elements non-functional properties

Developed the implementation of the approach