On the Formal Generation of Process Redesigns

Mariska Netjes

Hajo A. Reijers
Wil M.P. van der Aalst

Eindhoven University of Technology
m.netjes@tue.nl

Outline

- Introduction
- The evolutionary approach towards process redesign:
  - Process definition
  - Parallelization:
    - Selection
    - Transformation
    - Replacement
  - Other transformations
- Conclusion and future work

Business Process Redesign (BPR)

- Radical restructuring of a business process
- with wide-scale application of information technology (Hammer, 1993)

- “The great majority of users want improved processes.” (BPM Market Survey 2007, BPTrends)

The research challenge

- "How to get from the as-is to the to-be [in a BPR project] isn't explained, so we conclude that during the break, the famous ATAMO procedure is invoked – And Then, A Miracle occurs."
  (Sharp & McDermott, 2002)

- BPR knows many methodologies, case studies, papers, books, etc. but… it is still *difficult* to find a good design.

There are many existing BPR applications and tools but…

- Are limited in domain application,
- Have not succeeded to gain widespread adoption in industry;
  - ProcessWise methodology, MIT Process Handbook, MIT's process recombinator tool, CBR solutions, KOPeR tool…

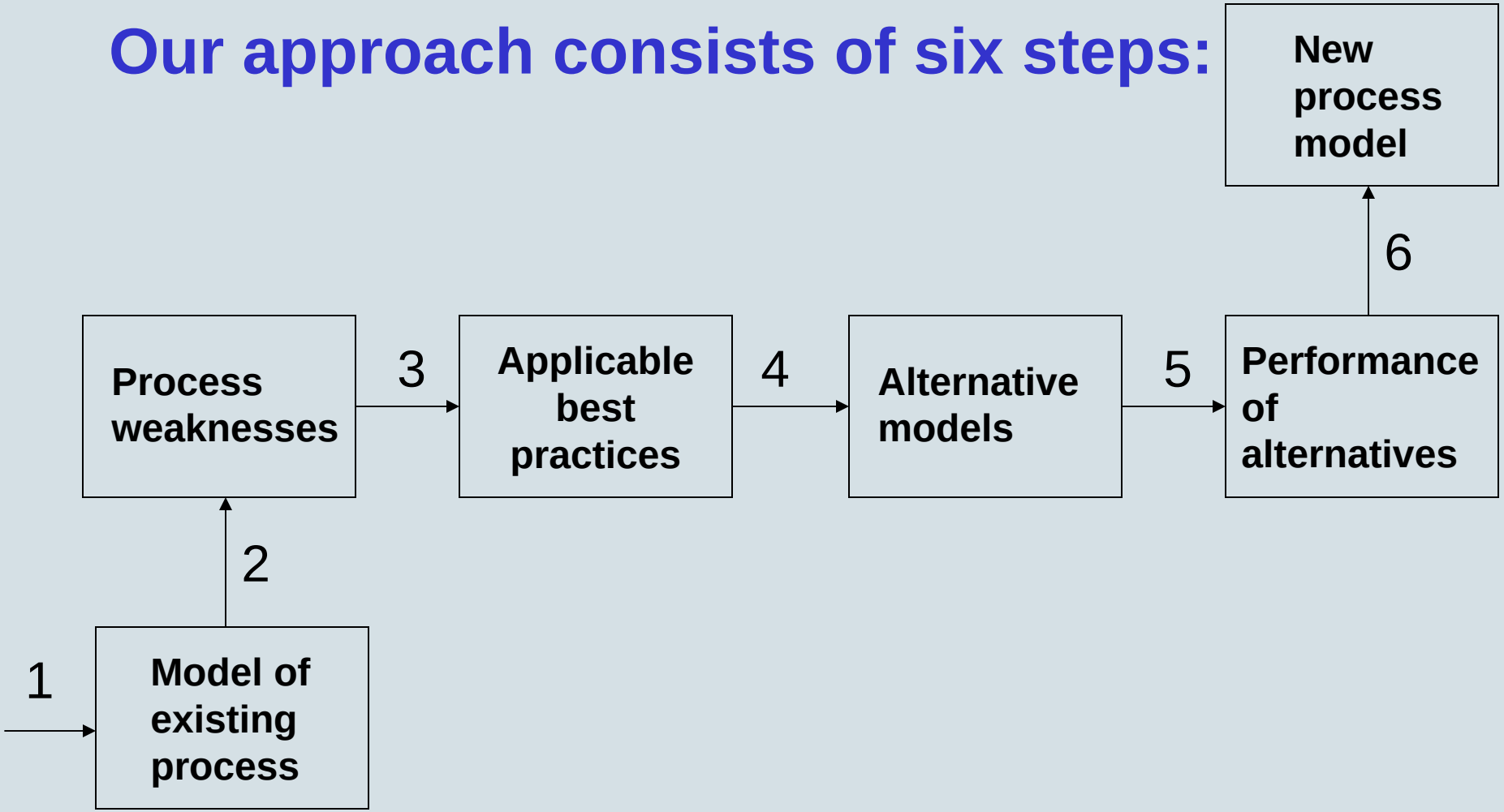
We propose an evolutionary approach towards business process redesign:

- provides and supports concrete redesign steps
- improves the existing process gradually
- uses redesign best practices => helps redesign novice
- creates and evaluates redesign alternatives

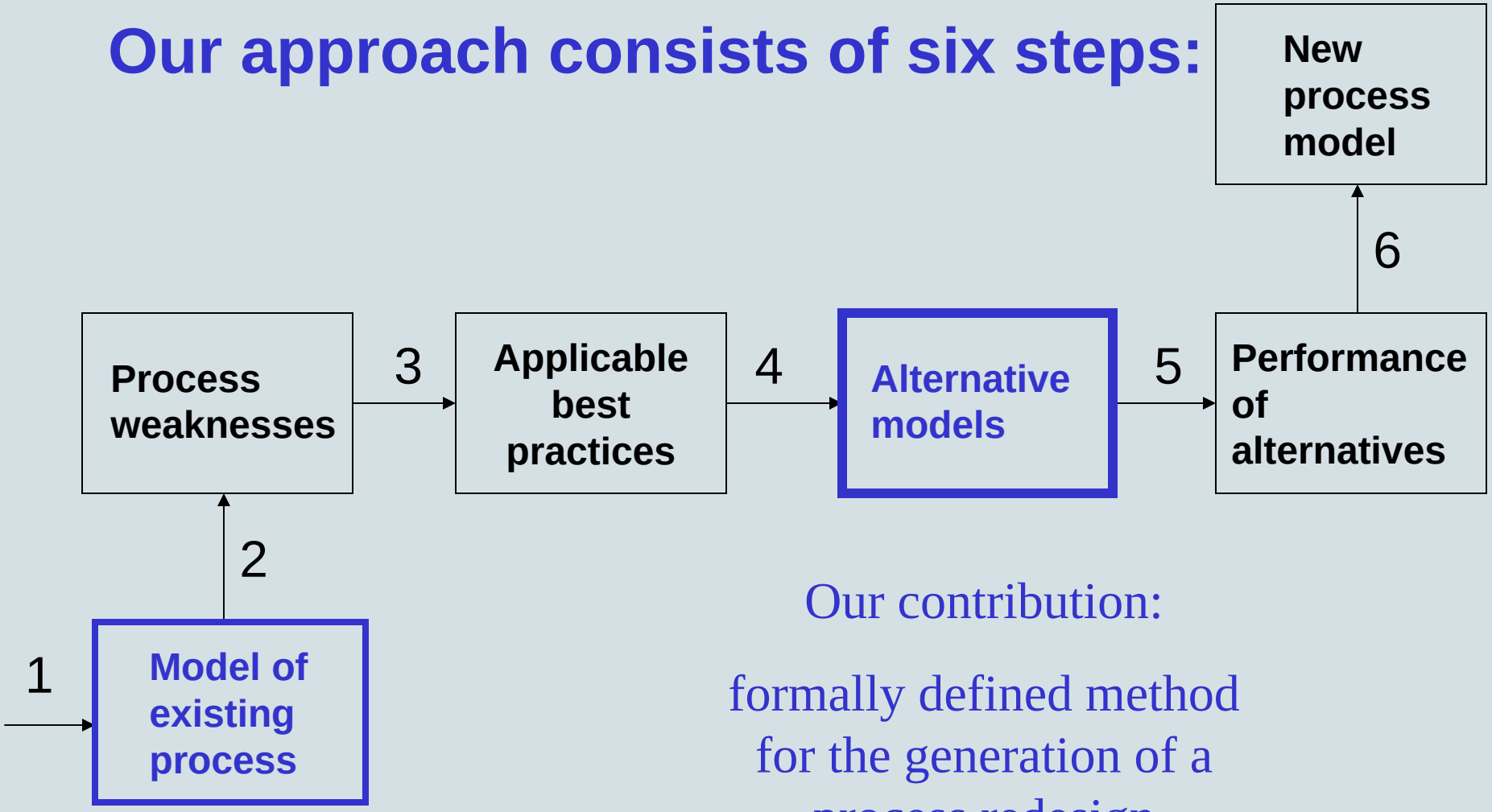
Towards a redesign tool:

- automation
- (intelligent) interaction with redesigner

Our approach consists of six steps:



Our approach consists of six steps:

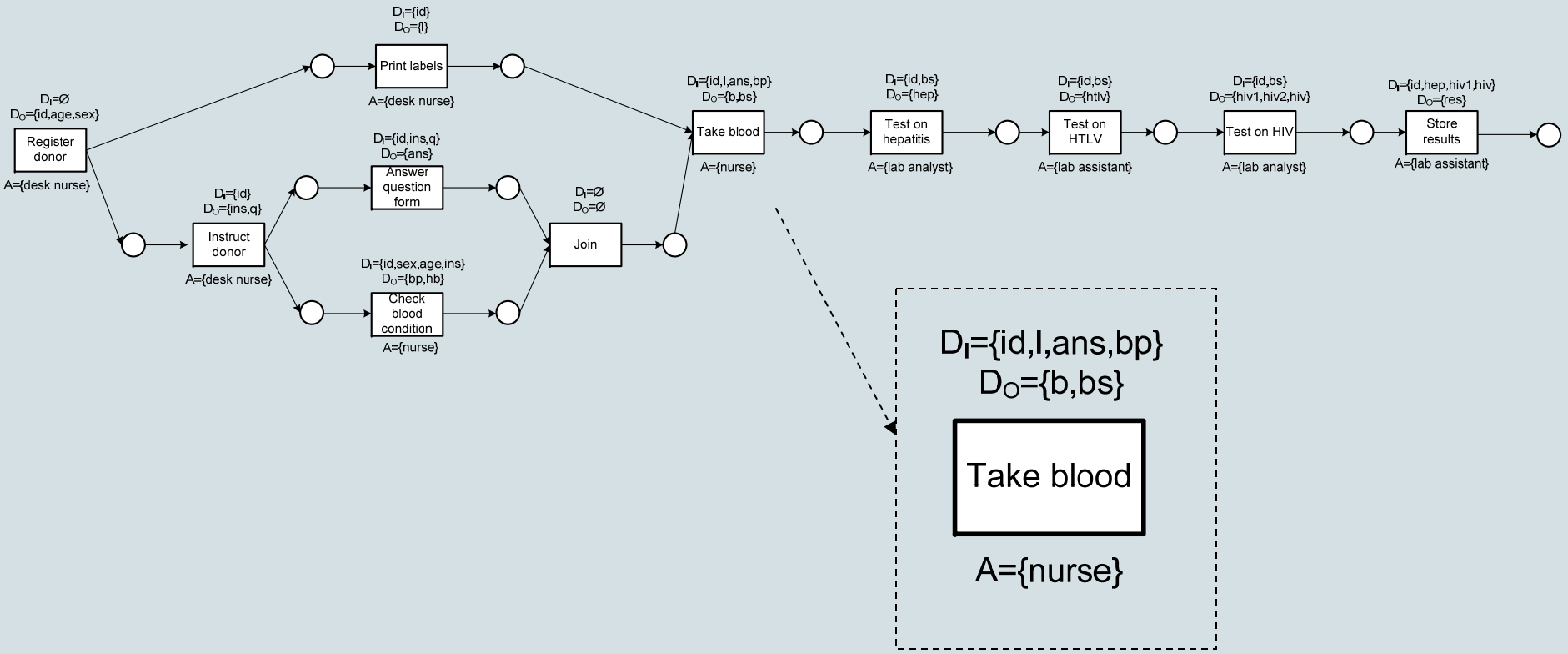


Our contribution:

formally defined method for the generation of a process redesign

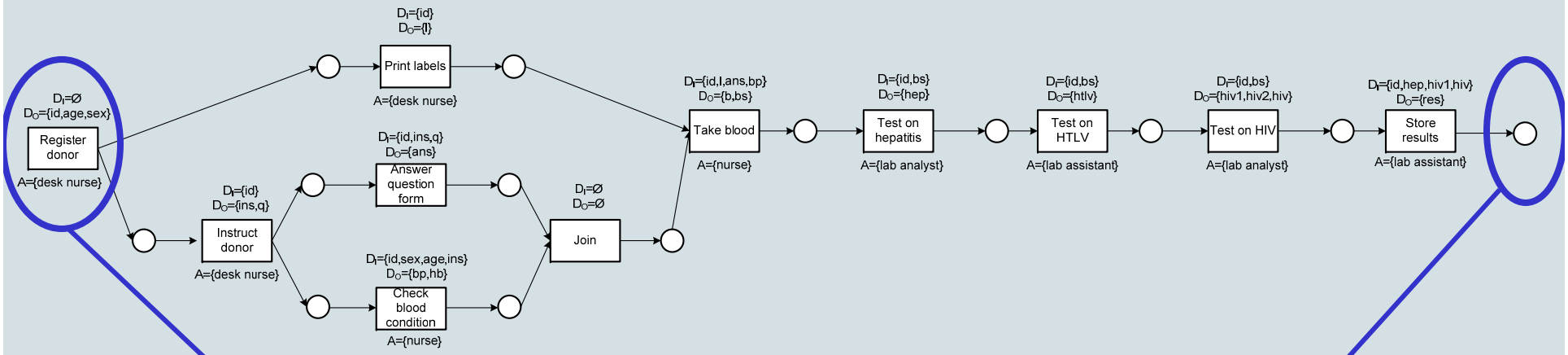
Step 1: Existing process model

Process definition



Step 1: Existing process model

Process properties

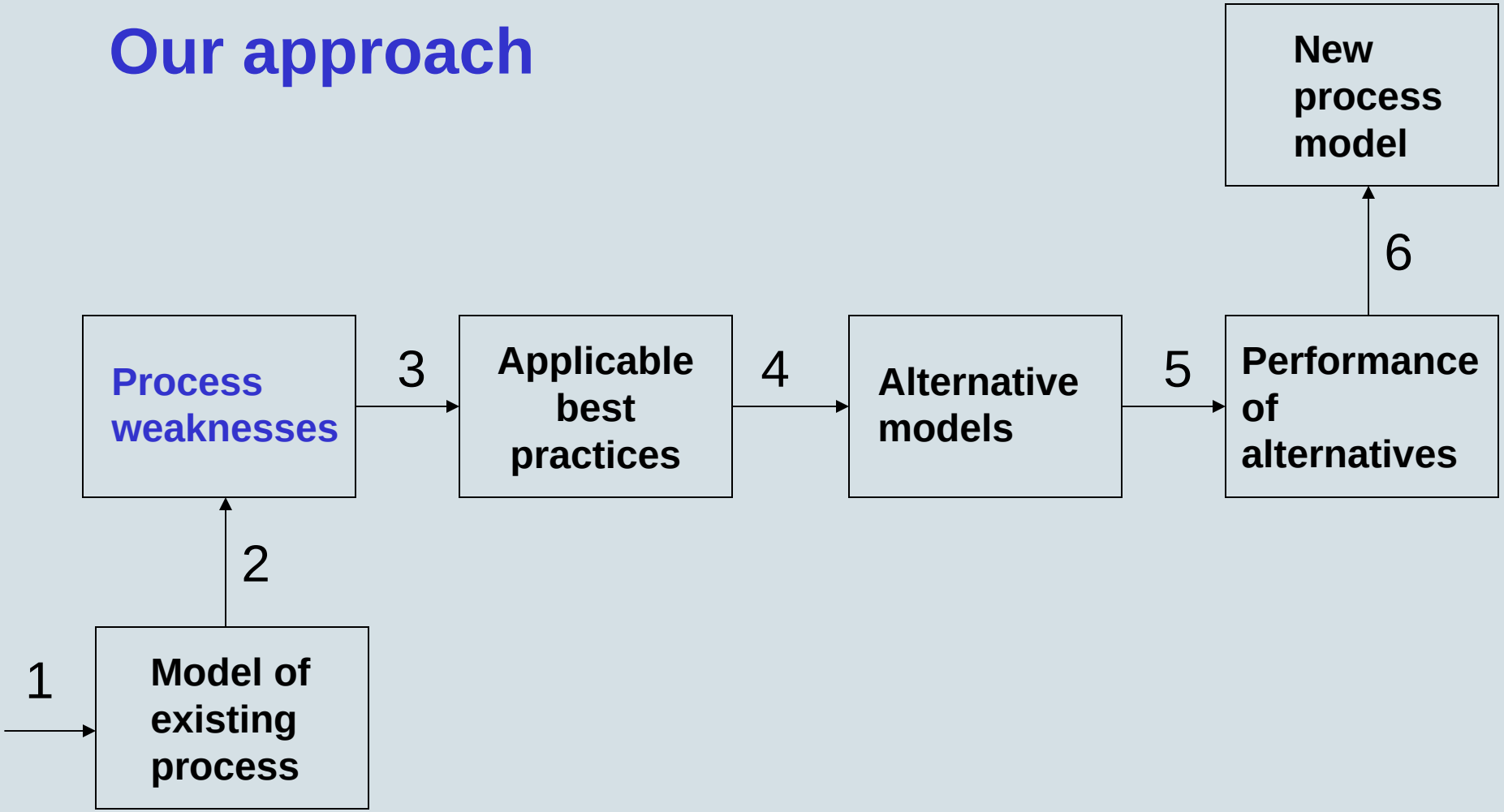


input

- safeness
- soundness

output

Our approach

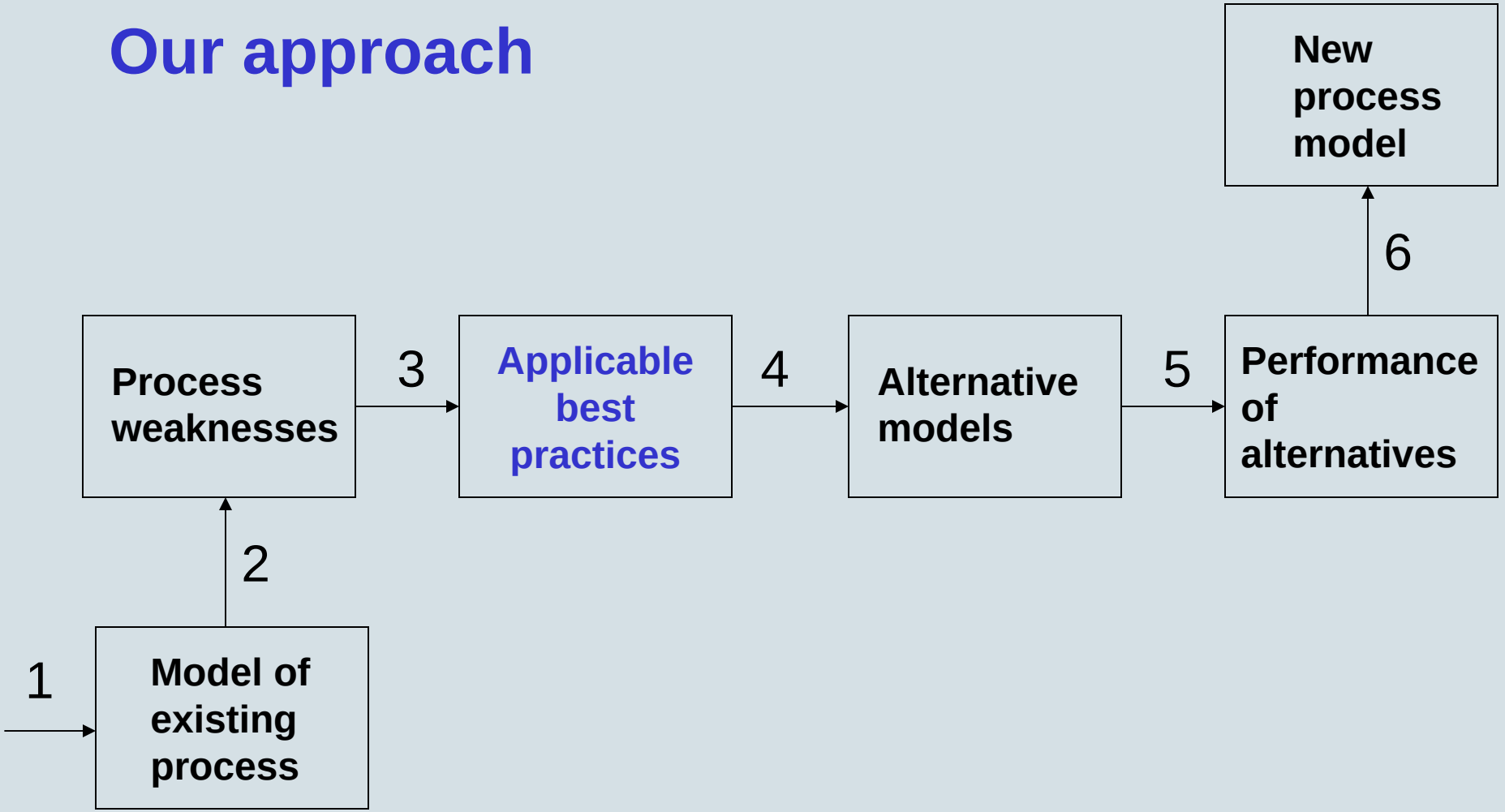


Step 2: Process weaknesses

- Purpose: Find inefficiencies in the process
- Method: global view with process measures
- Examples:
  - process size = number of tasks
  - IT automation = percentage of automated tasks
  - parallellism = percentage of parallel tasks
  - process hand overs = percentage of work that is handed over
  - role usage = percentage of actively involved roles

Netjes, M., Limam Manser, S., Reijers, H.A., Aalst, W.M.P. van der Aalst: An Evolutionary Approach for Business Process Redesign: Towards an Intelligent System. In: Proceedings of ICEIS 2007.

Our approach



Step 3: Applicable best practices

Redesign best practices

- Collection of 29 best practices from literature and hands-on experience (Reijers and Limam Mansar, 2005)
- Examples:
  - Parallelism: consider whether tasks may be executed in parallel
  - Task composition: combine small tasks with the same role into composite tasks

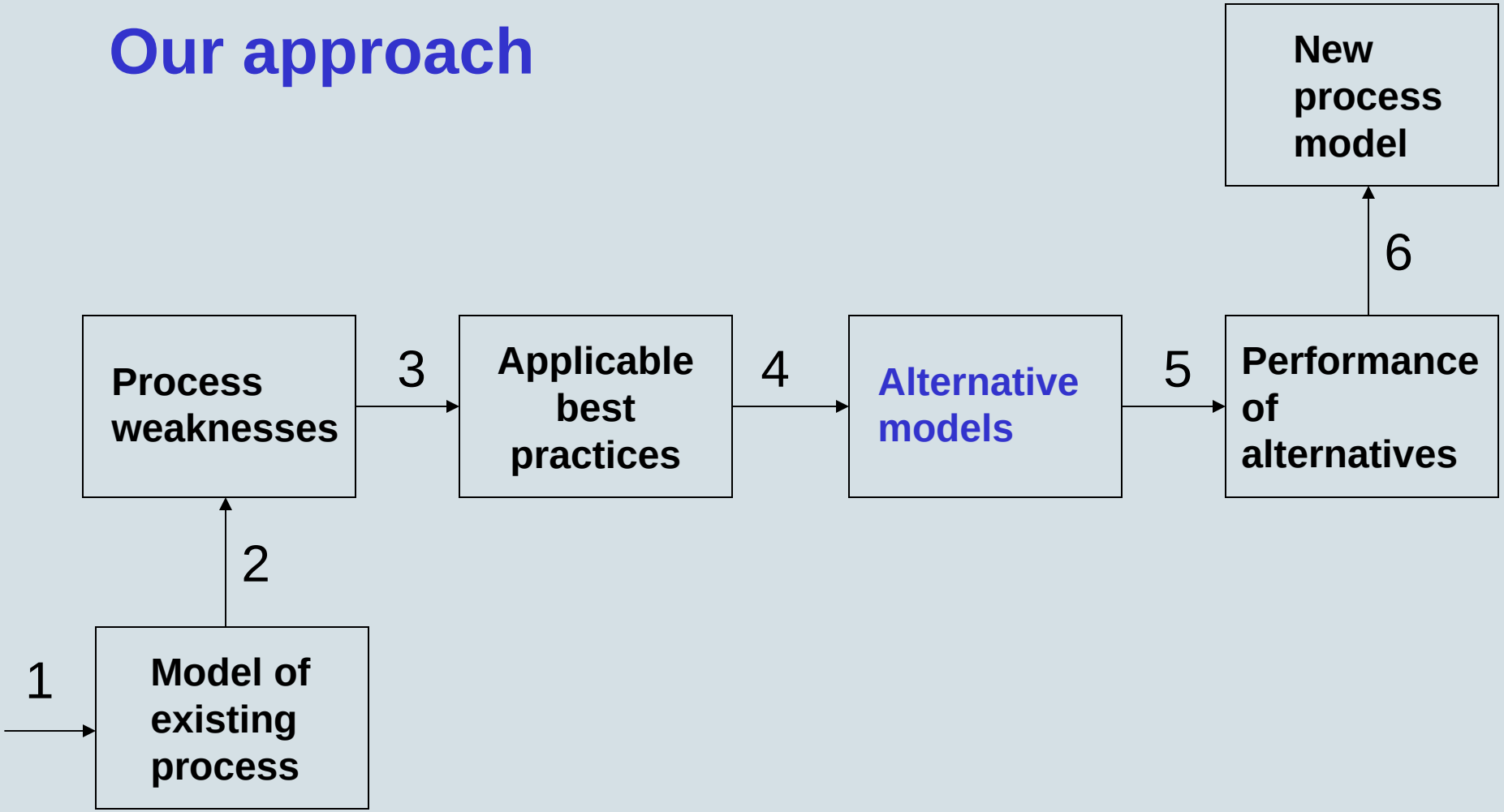
Step 3: Appicable best practices

Condition statements

- Purpose: evaluate applicability of each best practice
- Method: condition statement = a combination of process measures
- Examples:
  - Parallelism: apply if parallelism < 0.1
  - Composition: apply if parallelism < 0.1 AND Process hands off < 0.3

Netjes, M., Limam Manser, S., Reijers, H.A., Aalst, W.M.P. van der Aalst: An Evolutionary Approach for Business Process Redesign: Towards an Intelligent System. In: Proceedings of ICEIS 2007.

Our approach



Step 4: Alternative models

Generation of a process redesign

- Changing part of an existing process model
- Change performed in three steps:
  - Selection
  - Transformation
  - Replacement

Step 4: Alternative models

Selection (1)

- Component *: selected process part that should be changed

Definition 5 (Component) *Let S be an annotated SISO-net. Then C is a component in S if and only if:*

- $C \subseteq \pi_P(S) \cup \pi_T(S)$,
- *there are source and sink nodes* $i_C, o_C \in C$ *such that:*
  - $i_C \neq o_C$
  - $\bullet(C \setminus \{i_C\}) \subseteq C \setminus \{o_C\}$,
  - $(C \setminus \{o_C\})\bullet \subseteq C \setminus \{i_C\}$, *and*
  - $(o_C, i_C) \notin \pi_F(S)$.

* Aalst, W.M.P. van der Aalst, Bisgaard Lassen, K.: Translating Unstructured Workflos Processes to Readable BPEL: Theory and Implementation. Information and Software Technology, 2008.

Step 4: Alternative models

Selection (2)

- Projection of the net on the component

Definition 23 (Projection) *Let S be an annotated SISO-net and let C be a component in S. The projection of S on C $S||_C$, is then defined as $S||_C = (P, T, F, D, D_I, D_O, L, A)$ with:*

- $P = \pi_P(S) \cap C$ *is the set of places,*
- $T = \pi_T(S) \cap C$ *is the set of transitions,*
- $F = \pi_F(S) \cap (C \times C)$ *is the flow relation,*
- $D = \bigcup_{t \in T} \pi_{D_I}(S)(t) \cup \pi_{D_O}(S)(t)$ *is the set of dependencies,*
- $D_I \in T \rightarrow \mathcal{P}(D)$ *is the set of input dependencies such that* $\forall_{t \in T} : D_I(t) = \pi_{D_I}(S)(t)$,
- $D_O \in T \rightarrow \mathcal{P}(D)$ *is the set of output dependencies such that* $\forall_{t \in T} : D_O(t) = \pi_{D_O}(S)(t)$,
- $L = \bigcup_{t \in T} \pi_A(S)(t)$ *is the set of labels, and*
- $A \in T \not\rightarrow L$ *is the label assignment such that* $dom(A) = dom(\pi_A(S)) \cap C$ *and* $\forall_{t \in dom(A)} : A(t) = \pi_A(S)(t)$.

Step 4: Alternative models

Selection (3)

- The compositional nature of safe and sound SISO-nets.

Theorem 1. *Let S be a safe and sound annotated SISO-net and let C be a component of S. The projection* $S||_C$ *is a safe and sound annotated SISO-net.*

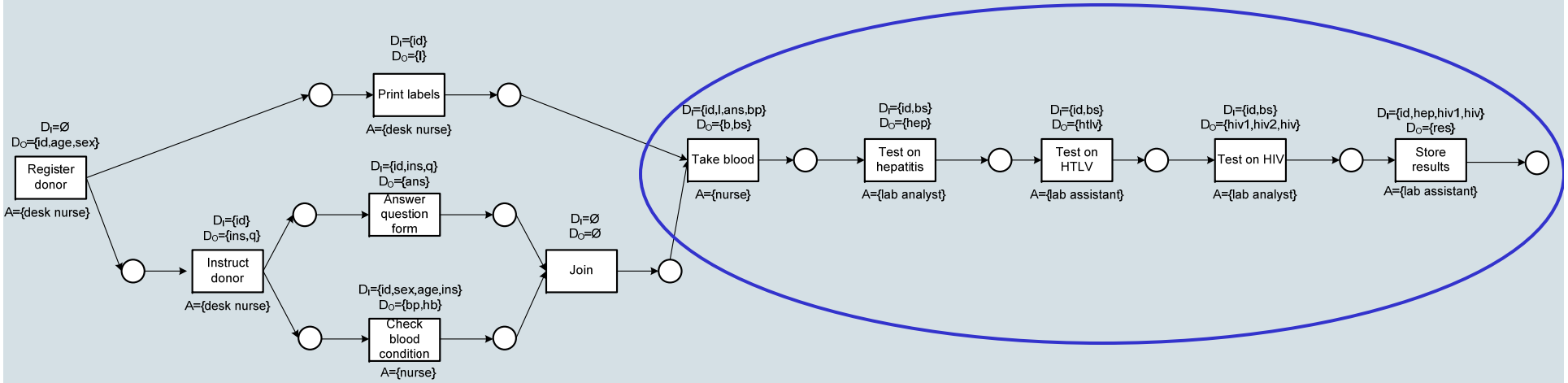
- Soundness and safeness are propagated to any component in the net.

Proof in: Netjes, M., Reijers, H.A., Aalst, W.M.P. van der Aalst: The creation of Process Redesign by Selecting, Transforming and Replacing Process Parts. BETA Working Paper Series, 2008.

Step 4: Alternative models

Selection - component

- Selected process part that should be changed



Step 4: Alternative models

Transformation

- Basic soundness preserving transformation rules * as starting point
- Generation of an alternative process part
- Type of change depends on chosen transformation:
  - Parallel transformation
  - Sequence transformation
  - Unfold transformation
  - Merge transformation

* Aalst, W.M.P. van der Aalst: Verification of Workflow Nets. In: LNCS 1248, Application and Theory of Petri Nets, 1997

Step 4: Alternative models

Parallel transformation

- No dependencies between tasks, but still ordered:
  - unnecessary delays
- Perform tasks without dependencies in parallel
  - benefit: reduction in throughput time
  - disadvantage: loss of quality / more complexity
- All tasks with a disjoint set of dependencies are placed in parallel => maximum parallelization
- Method: put relations between tasks that share a dependency

Step 4: Alternative models

Requirements for annotation

- Input dependencies of a task are fulfilled before the tasks becomes enabled
- Dependency is output before input
  - Task with output is placed before tasks with input
- Implications
  - dependency has to be output
  - dependency does not have to be input
  - task does not have same input and output
  - component = acyclic and marked graph

Step 4: Alternative models

Parallel transformation - formal

Definition 7 (Parallel) *Let S be an annotated SISO-net and let C be a choice*[3] *construct free component in S with its projection $S||_C$ being acyclic. Operation* `parallel` *changes C into the annotated graph $G = (N, E, D, D_I, D_O, L, A)$, i.e.,* `parallel`$(S, C) = (N, E, D, D_I, D_O, L, A)$ *with:*

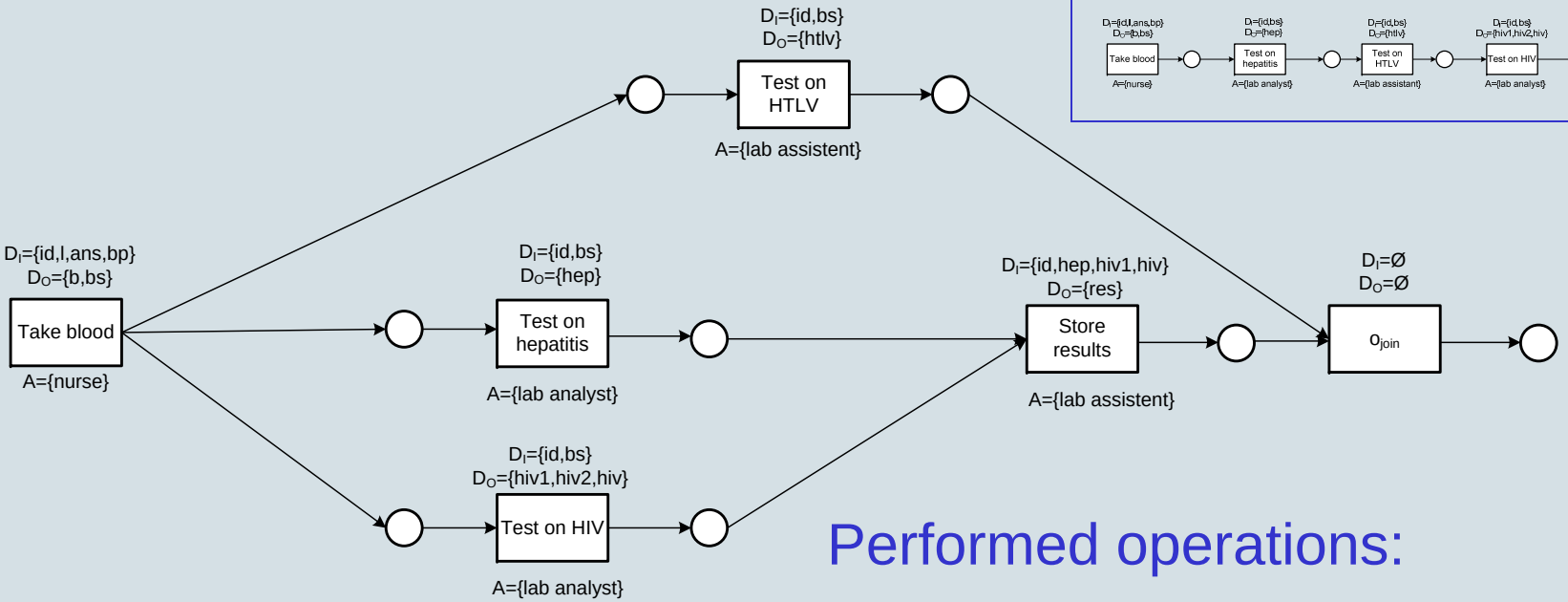
- $N = \{n \in \pi_T(S||_C) \mid \pi_{D_I}(S)(n) \neq \emptyset \vee \pi_{D_O}(S)(n) \neq \emptyset\}$[4] *is the set of nodes,*
- $E = \{(n_1, n_2) \in N \times N \mid (\pi_{D_O}(S)(n_1) \cap \pi_{D_I}(S)(n_2)) \neq \emptyset\}$ *is the set of edges,*
- $D = \pi_D(S||_C)$ *is the set of dependencies,*
- $D_I \in N \rightarrow \mathcal{P}(D)$ *is the set of input dependencies such that* $\forall_{n \in N} : D_I(n) = \pi_{D_I}(S)(n)$,
- $D_O \in N \rightarrow \mathcal{P}(D)$ *is the set of output dependencies such that* $\forall_{n \in N} : D_O(n) = \pi_{D_O}(S)(n)$,
- $L = \pi_L(S||_C)$ *is the set of labels, and*
- $A \in T \not\rightarrow L$ *is the label assignment such that* $dom(A) = dom(\pi_A(S||_C))$, *and* $\forall_{t \in dom(\pi_A(S||_C))} : A(t) = \pi_A(S)(t)$.

[3] Putting choice tasks in parallel would interfere with the choice behavior.

[4] Routing transitions, i.e., transitions for which holds $\pi_{D_I}(S)(t) = \pi_{D_O}(S)(t) = \emptyset$, are not considered and henceforth removed.

Step 4: Alternative models

Parallel transformation - example



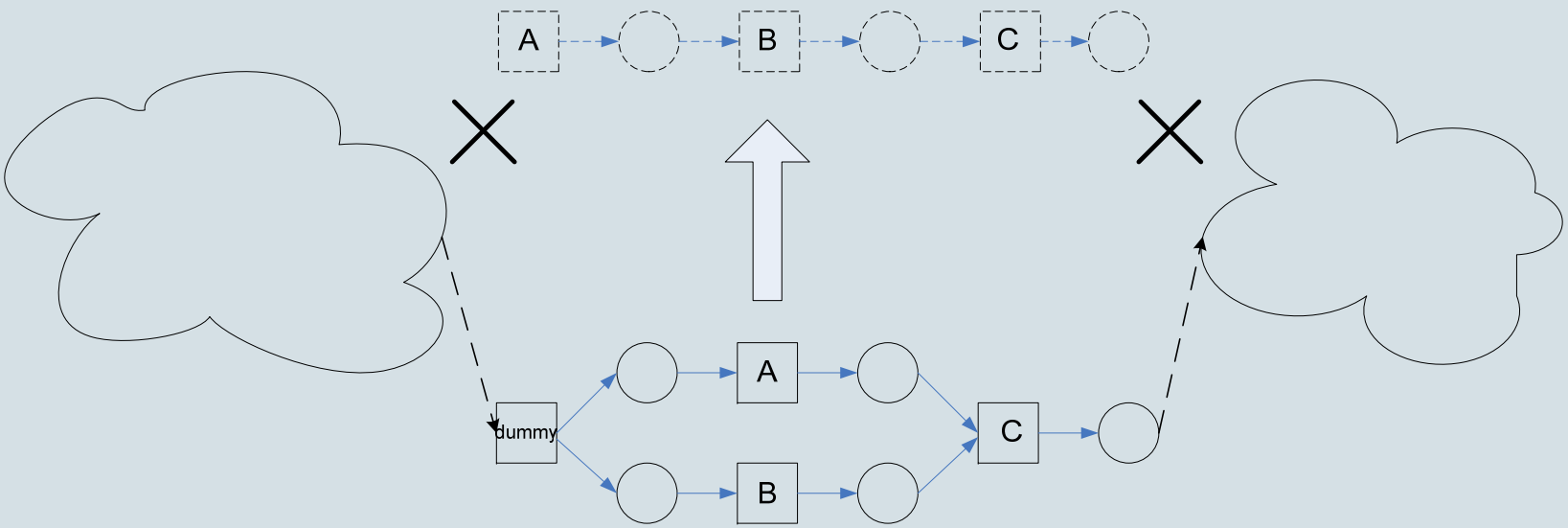
Component

Performed operations:

- addition of single output
- removal of superfluous relations
- translation to annotated SISO-net

Step 4: Alternative models

Replacement (1)



Step 4: Alternative models

Replacement (2a)

Definition 8 (Replace) *Let S_1 and S_2 be two annotated SISO-nets. Let C, with source node i_C and sink node o_C, be a non-trivial component in S_1. Let $\texttt{in}(S_2)$ be a place if and only if i_C is a place and let $\texttt{out}(S_2)$ be a place if and only if o_C is a place. Operation $\texttt{replace}$ substitutes $S_1||_C$ in S_1 with S_2 resulting in $\texttt{replace}(S_1, C, S_2) = (P_3, T_3, F_3, D_3, D_{I_3}, D_{O_3}, L_3, A_3)$ with:*

- $P_3 = (\pi_P(S_1) \setminus C) \cup \pi_P(S_2)$ *is the set of places* [5],
- $T_3 = (\pi_T(S_1) \setminus C) \cup \pi_T(S_2)$ *is the set of transitions,*
- $F_3 = (\pi_F(S_1) \cap ((P_3 \times T_3) \cup (T_3 \times P_3))) \cup$
  $\pi_F(S_2) \cup$
  $\{(n, \texttt{in}(S_2)) \mid (n, i_C) \in \pi_F(S_1)\} \cup$
  $\{(\texttt{out}(S_2), n) \mid (o_C, n) \in \pi_F(S_1)\}$ *is the flow relation,*

[5] We assume there are no "name clashes".

Step 4: Alternative models

Replacement (2b)

- $D_3 = \left(\bigcup_{t \in \pi_T(S_1) \setminus C} \pi_{D_I}(S_1)(t) \cup \pi_{D_O}(S_1)(t) \right) \cup \pi_D(S_2)$ *is the set of dependencies,*
- $D_{I_3} \in T_3 \rightarrow \mathcal{P}(D_3)$ *is the set of input dependencies such that:*
  - $\forall_{t \in \pi_T(S_1) \setminus C} : D_{I_3}(t) = \pi_{D_I}(S_1)(t)$, *and*
  - $\forall_{t \in \pi_T(S_2)} : D_{I_3}(t) = \pi_{D_I}(S_2)(t)$,
- $D_{O_3} \in T_3 \rightarrow \mathcal{P}(D_3)$ *is the set of output dependencies such that:*
  - $\forall_{t \in \pi_T(S_1) \setminus C} : D_{O_3}(t) = \pi_{D_O}(S_1)(t)$, *and*
  - $\forall_{t \in \pi_T(S_2)} : D_{O_3}(t) = \pi_{D_O}(S_2)(t)$,
- $L_3 = \left(\bigcup_{t \in \pi_T(S_1) \setminus C} \pi_A(S_1)(t) \right) \cup \pi_L(S_2)$ *is the set of labels, and*
- $A_3 \in T_3 \not\rightarrow L_3$ *is the label assignment such that:*
  - $dom(A_3) = (dom(\pi_A(S_1)) \setminus C) \cup dom(\pi_A(S_2))$,
  - $\forall_{t \in dom(\pi_A(S_1)) \setminus C} : A_3(t) = \pi_A(S_1)(t)$, *and*
  - $\forall_{t \in dom(\pi_A(S_2))} : A_3(t) = \pi_A(S_2)(t)$

Step 4: Alternative models

Replacement (3)

- The result of the replacement is again a safe and sound annotated SISO-net.

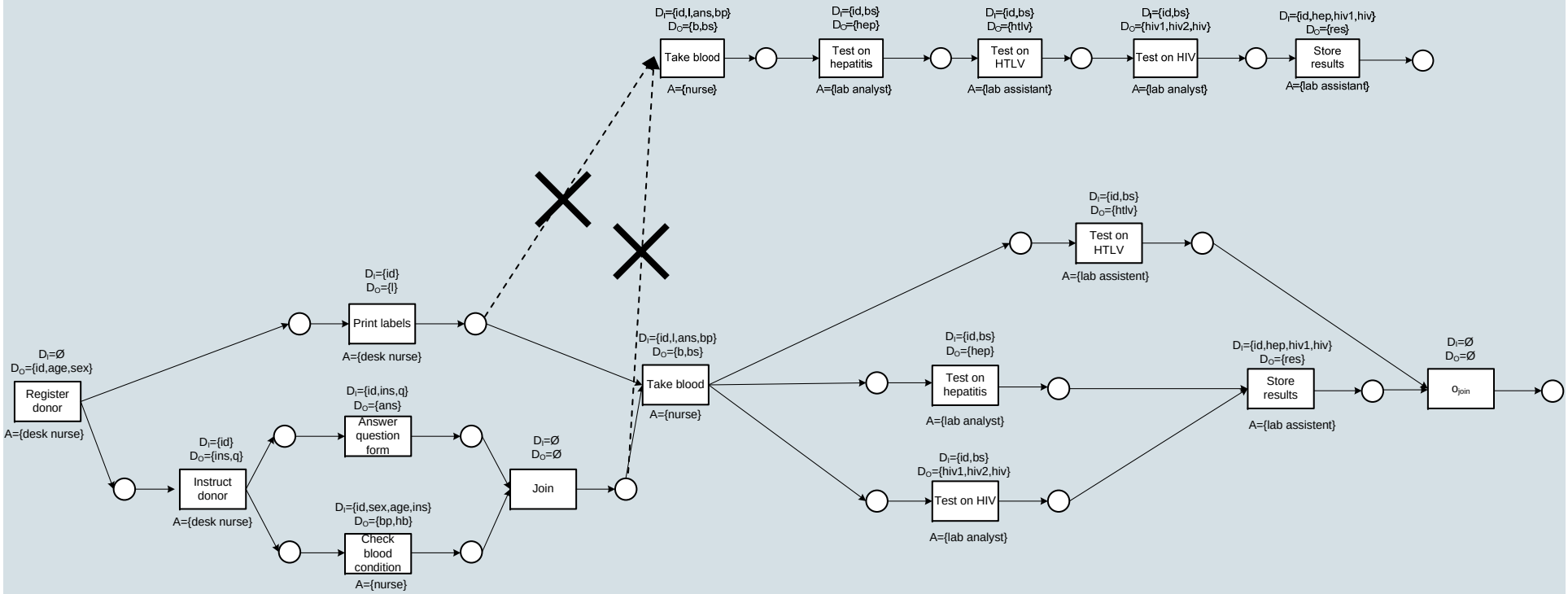
Theorem 2. *Let S_1 and S_2 be two annotated SISO-nets. Let C be a non-trivial component in S_1. Let $S_3 = \texttt{replace}(S_1, C, S_2)$.*
Then:

- *S_3 is an annotated SISO-net.*
- *If S_1 is safe and sound and S_2 is safe and sound, then S_3 is safe and sound.*

Proof in: Netjes, M., Reijers, H.A., Aalst, W.M.P. van der Aalst: The creation of Process Redesign by Selecting, Transforming and Replacing Process Parts. BETA Working Paper Series, 2008.

Step 4: Alternative models

Replacement (2)



Step 4: Alternative models

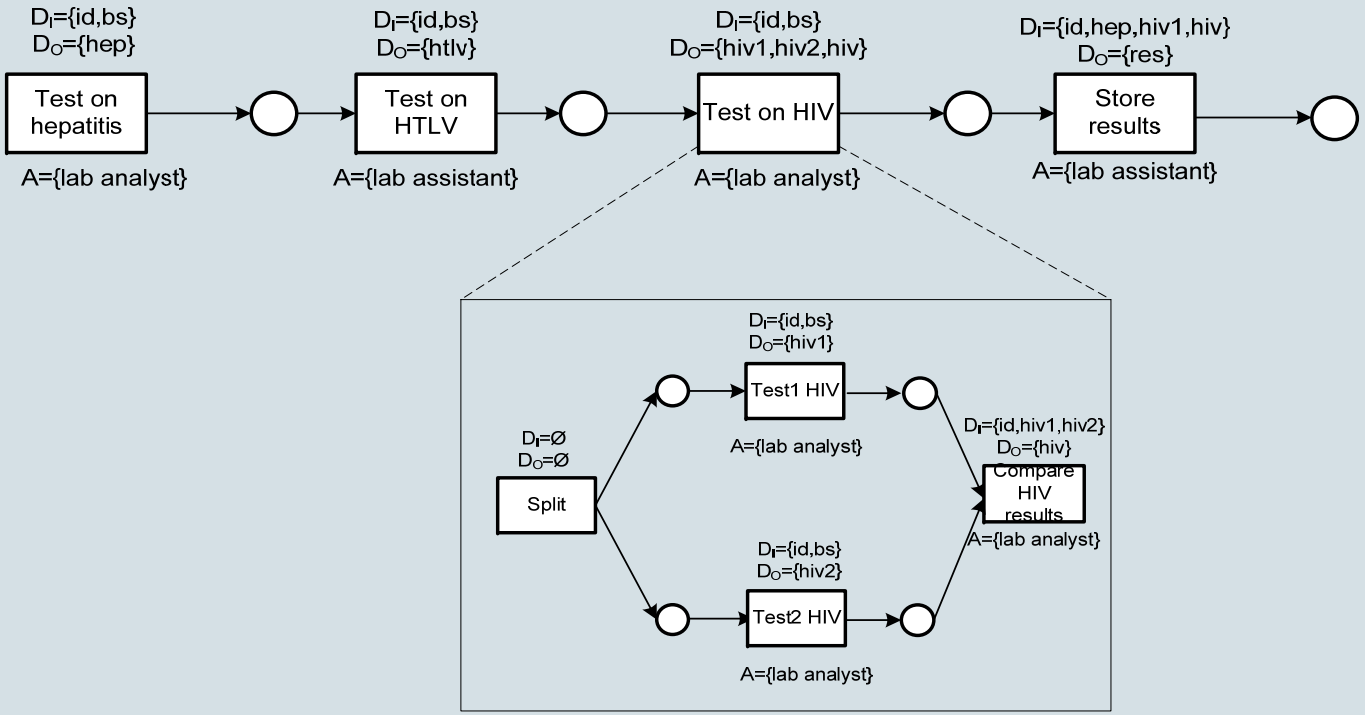
Other transformations - sequence

- Counterpart of parallel transformation
- Transitions are placed in a fixed order; a sequence
- Benefits: simpler, higher quality, no synchronisation
- Drawback: longer throughput times
- Transformation:
  - Create a sequence while preserving dependencies

Step 4: Alternative models

Process definition - revisited

- Layered annotated SISO-net with aggregated tasks
- Detailed specification of complex tasks



Step 4: Alternative models

Other transformations - unfold

- Aggregated tasks are splitted up
- Benefits: higher run-time flexibility, higher quality
- Drawback: longer setup times
- Transformation:
  - Replace each aggregated task by its lower layer net
  - Lower layer SISO-net starts and ends with a transition

Step 4: Alternative models

Other transformations - merge

- Counterpart of unfold transformation
- Tasks are combined into aggregated task
- Benefits: reduction of setup times, higher quality
- Transformation:
  - Combine similar tasks, i.e., tasks with the same label, while preserving dependencies

Our approach

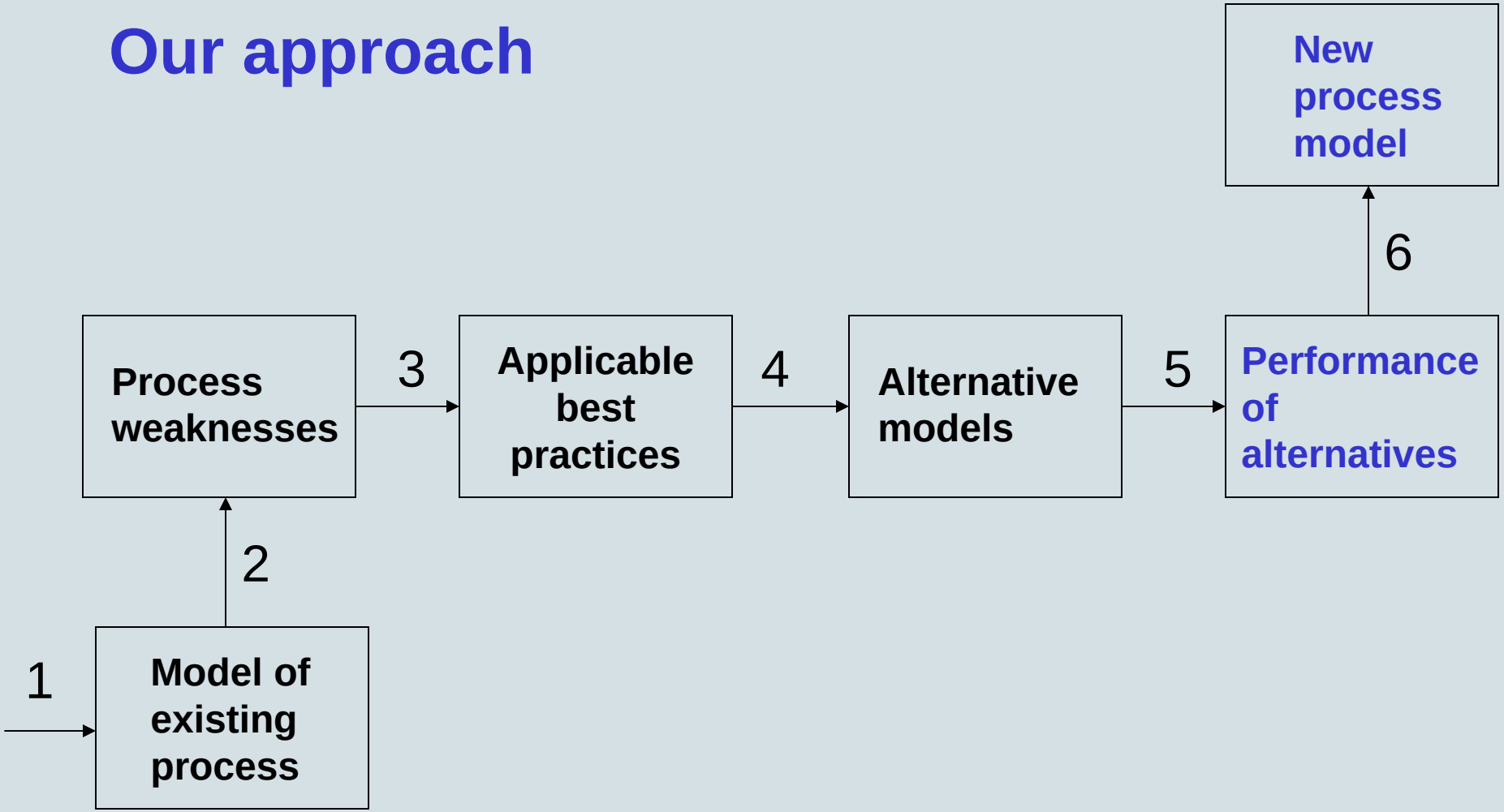


Illustration of the remaining steps

- Step 5: Performance of alternatives
  - Evaluation by simulation or analytical techniques
  - Performance data is collected from event logs
- Step 6: New process model
  - Implement the best alternative

Conclusion

- Concrete method for the generation of alternative process models
- Extreme changes
- Generic set of process attributes
- Current set of process transformations is starting point for process redesign

Outlook

- Development of a supporting tool
  - Automation
  - "Intelligent" support
- Support for:
  - Finding alternatives (redesign novices)
  - Evaluating alternatives (experienced redesigners)

Thank you for your attention

Mariska Netjes
Hajo A. Reijers
Wil M.P. van der Aalst

m.netjes@tue.nl