TU/e technische universiteit eindhoven

Process Mining: The next step in Business Process Management

Prof.dr.ir. Wil van der Aalst

Eindhoven University of Technology
Department of Information and Technology
P.O. Box 513, 5600 MB Eindhoven
The Netherlands
w.m.p.v.d.aalst@tm.tue.nl
&
Centre for Information Technology Innovation (CITI)
Queensland University of Technology (QUT)
Brisbane, Australia

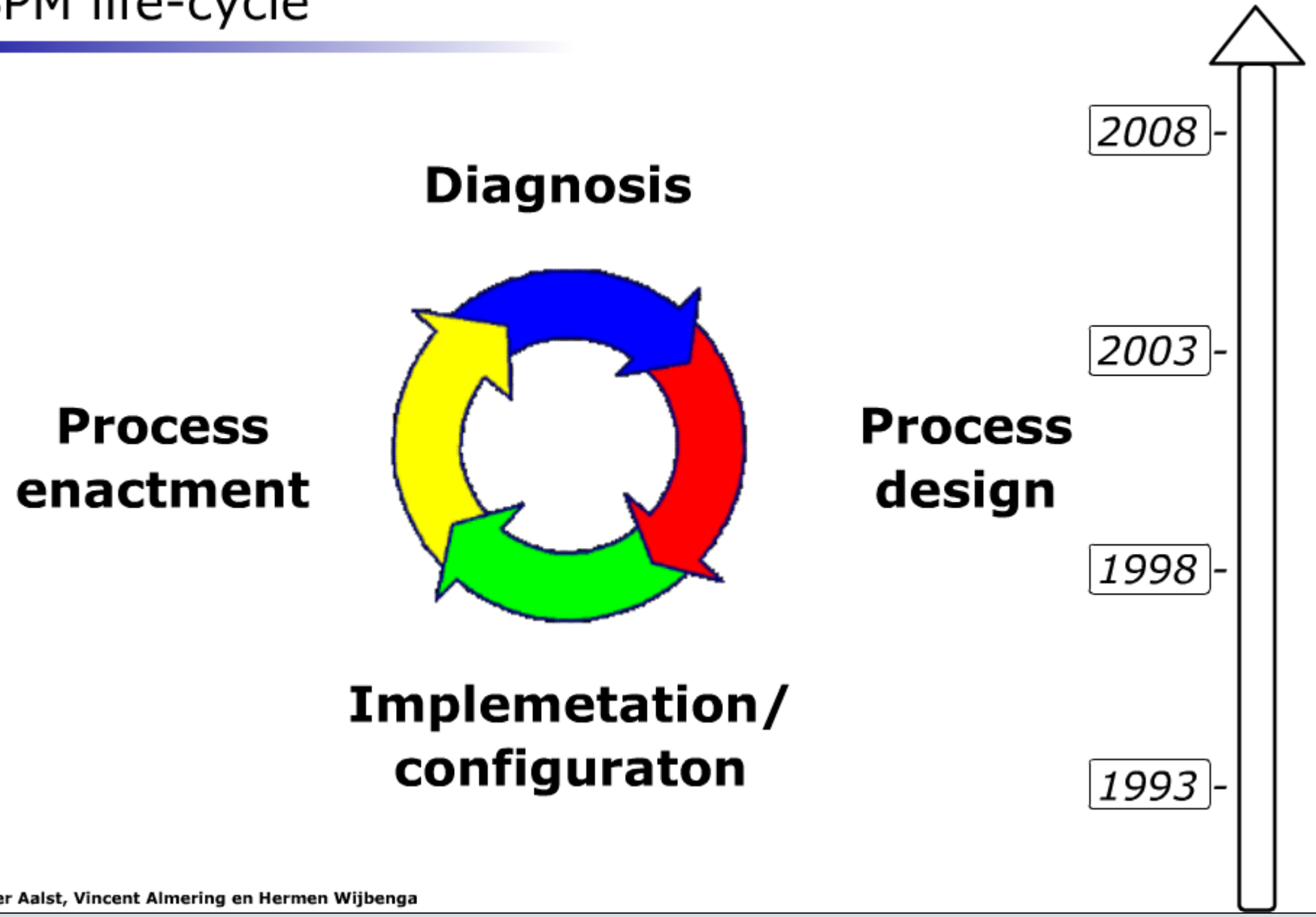
Outline

- Motivation
- Overview of process mining
  - Basic performance metrics
  - Process models
  - Organizational models
  - Social networks
  - Performance characteristics
- Process Mining: Some of our tools
  - EMiT
  - Thumb
  - MinSocN
- Conclusion

Motivation

Press arrow to start

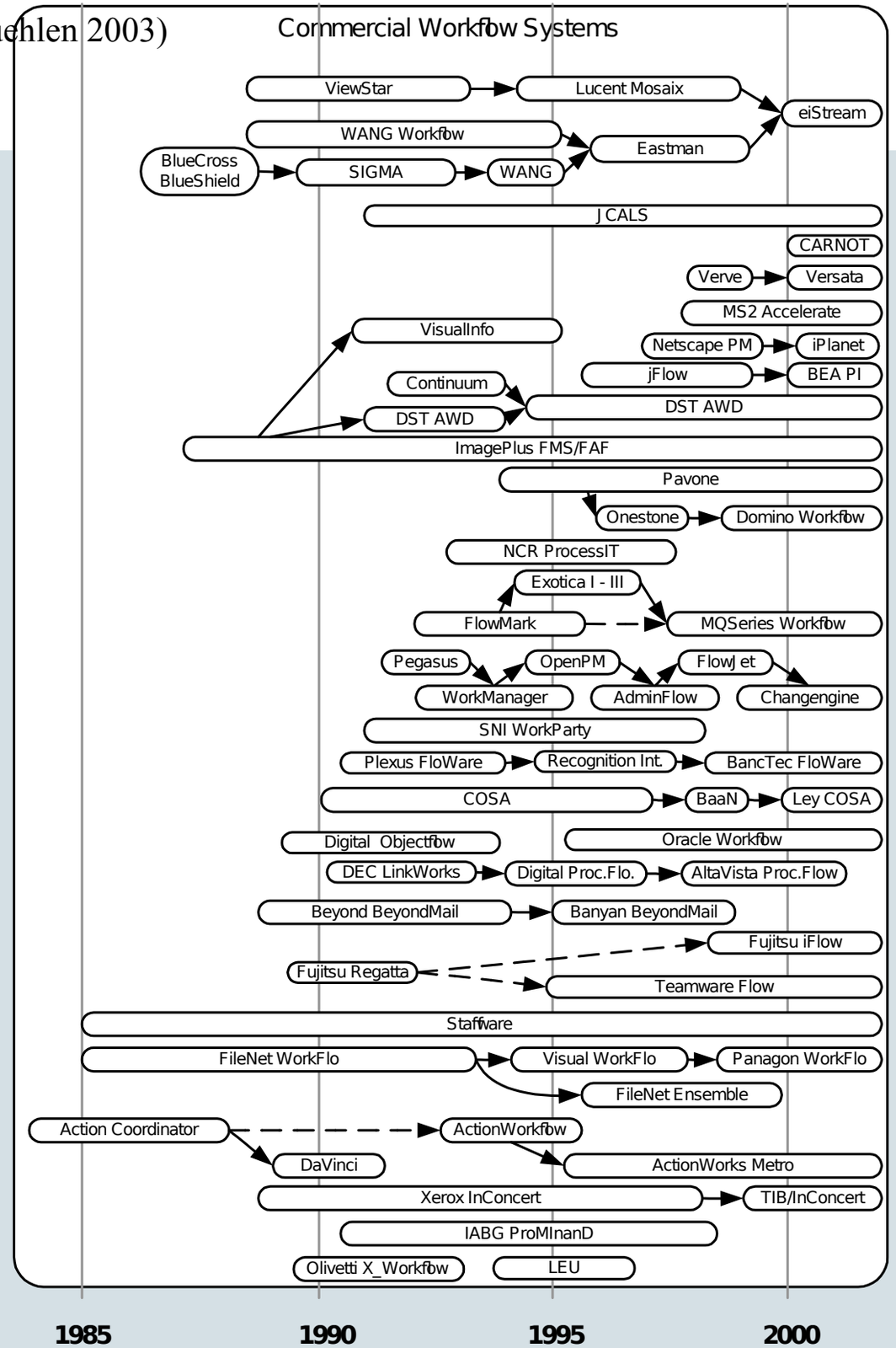
The BPM life-cycle



25 years of workflow

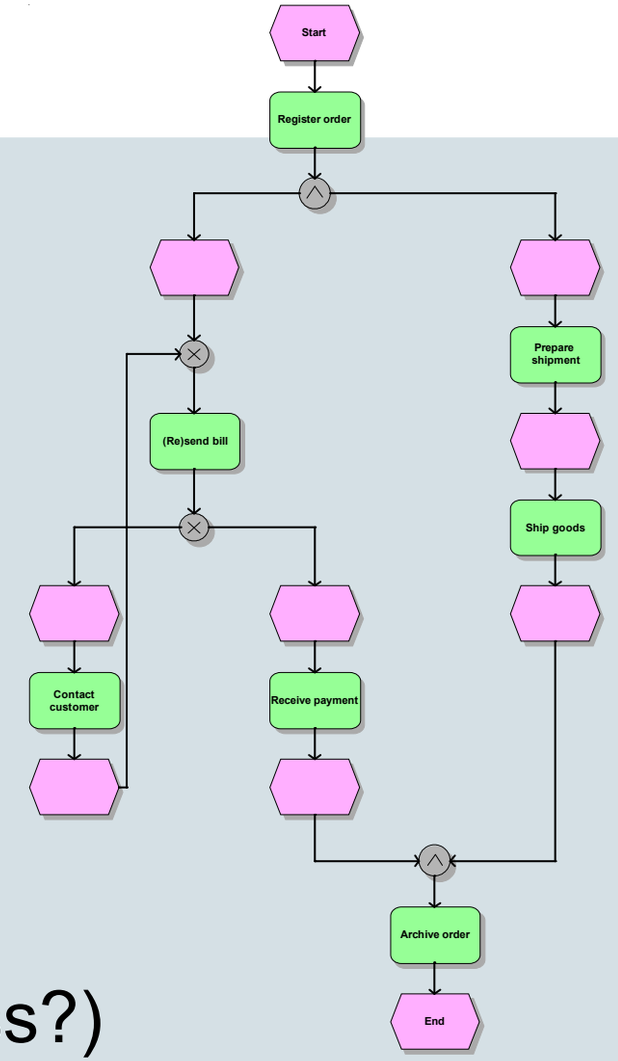
- Pioneers like Skip Ellis and Michael Zisman already worked on “office automation” in the 70-ties
- The WFM hype is over …, but there are more and more applications, it has become a mature technology, and WFM is adopted by many other technologies (ERP, Web Services, etc.).

(Zur Muehlen 2003)



Let us reverse the process!

process mining

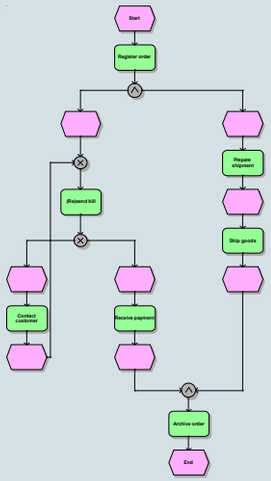


- Process mining can be used for:
  - Process discovery (What is the process?)
  - Delta analysis (Are we doing what was specified?)
  - Performance analysis (How can we improve?)
- Particularly interesting in pre- and post-workflow settings!

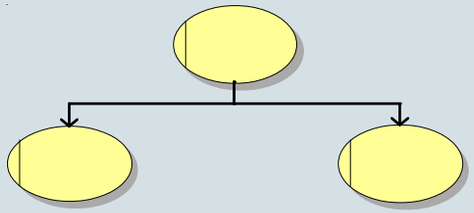
TU/e technische universiteit eindhoven

Process mining: Overview

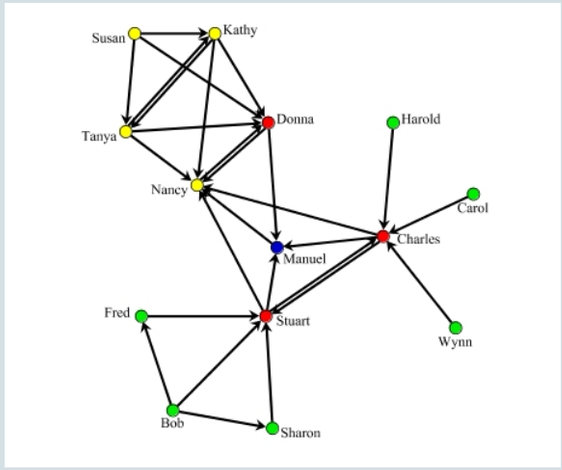
2) process model



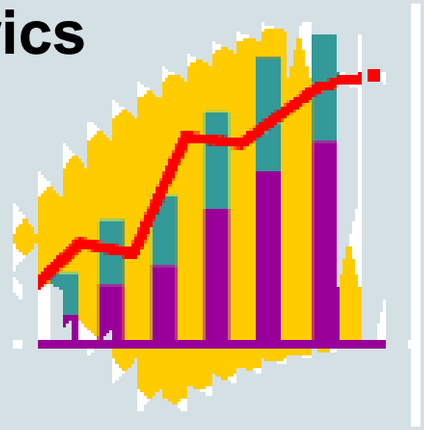
3) organizational model



4) social network



1) basic performance metrics



5) performance characteristics

If …then …

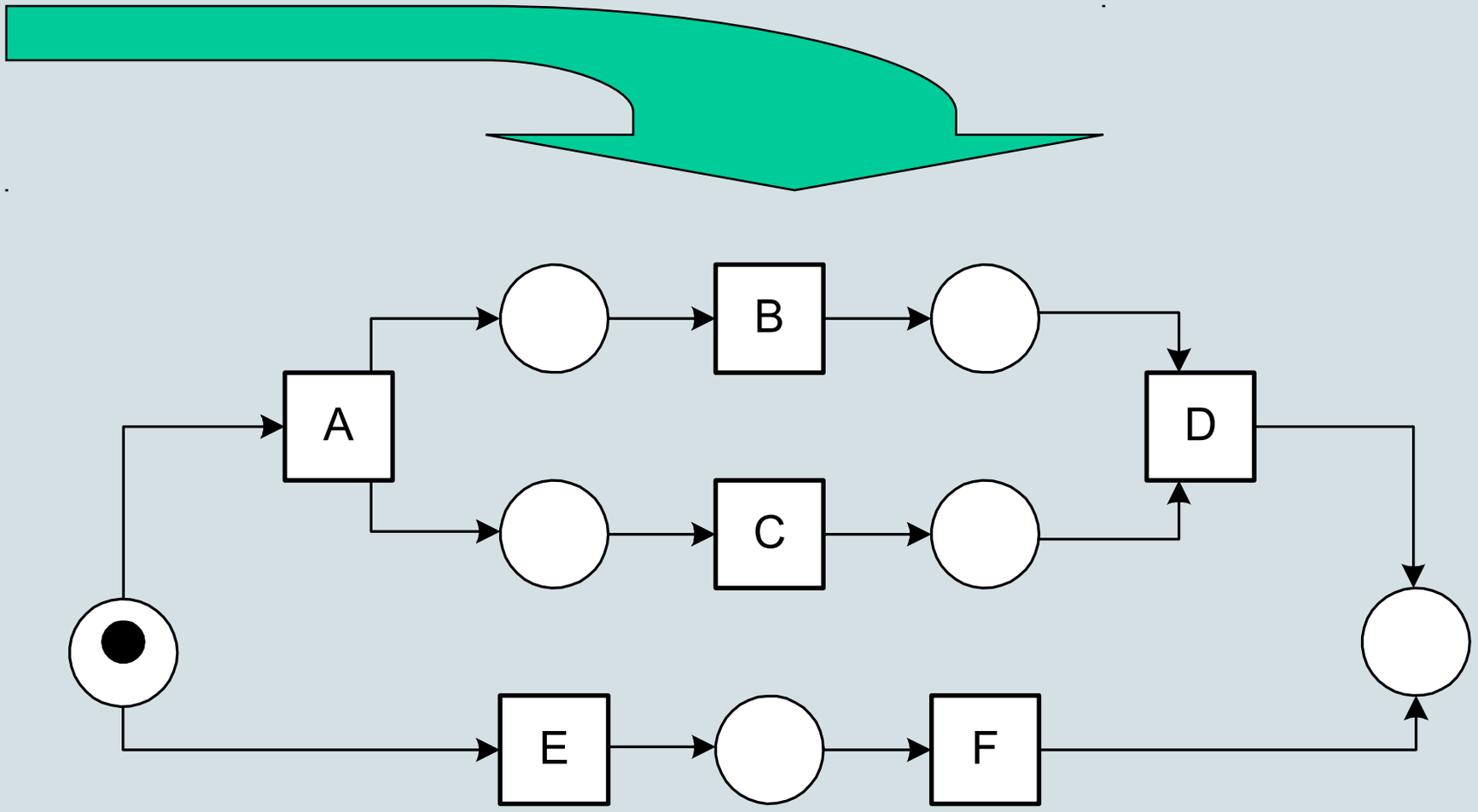
(1) Determine basic performance metrics

- Process/control-flow perspective: *flow time*, *waiting time*, *processing time* and *synchronization time*.
  Questions:
  - *What is the average flow time of orders?*
  - *What is the maximum waiting time for activity approve?*
  - *What percentage of requests is handled within 10 days?*
  - *What is the minimum processing time of activity reject?*
  - *What is the average time between scheduling an activity and actually starting it?*
- Resource perspective: *frequencies*, *time*, *utilization*, and *variability*.
  Questions:
  - *How many times did Sue complete activity reject claim?*
  - *How many times did John withdraw activity go shopping?*
  - *How many times did Clare suspend some running activity?*
  - *How much time did Peter work on instances of activity reject claim?*
  - *How much time did people with role Manager work on this process?*
  - *What is the utilization of John?*
  - *What is the average utilization of people with role Manager?*
  - *How many times did John work for more than 2 hours without interruption?*

(2) Determine process model

- Discover a *process model* (e.g., in terms of a PN or EPC) without prior knowledge about the structure of the process.

case 1 : task A
case 2 : task A
case 3 : task A
case 3 : task B
case 1 : task B
case 1 : task C
case 2 : task C
case 4 : task A
case 2 : task B
case 2 : task D
case 5 : task E
case 4 : task C
case 1 : task D
case 3 : task C
case 3 : task D
case 4 : task B
case 5 : task F
case 4 : task D



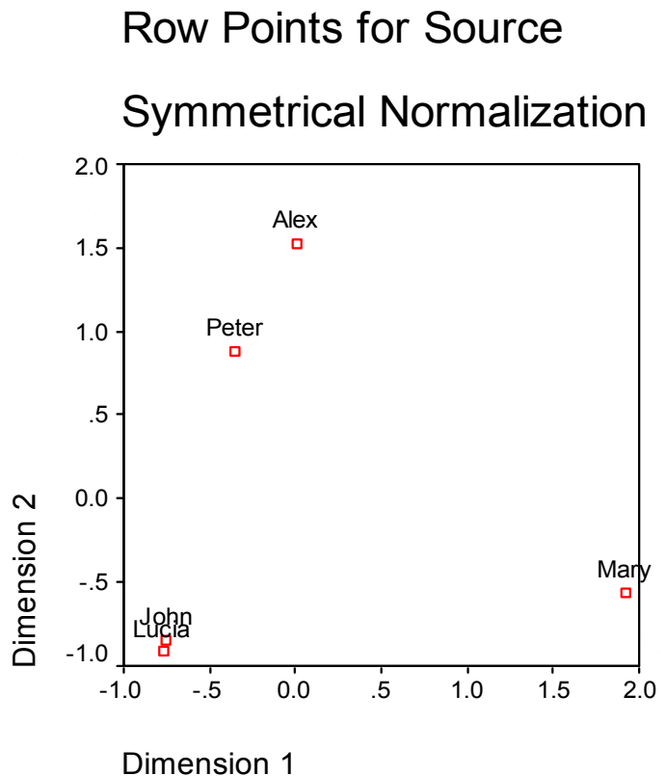
$\alpha(W)$

(3) Determine organizational model

- Discover the *organizational model* (i.e., roles, departments,etc.) without prior knowledge about the structure of the organization.

	A	B	C	D	E	F
John	88	0	8	0	38	50
Alex	0	189	0	2	0	0
Lucia	112	0	0	0	62	40
Peter	0	11	192	0	0	0
Mary	0	0	0	198	0	0

e.g., correspondence analysis (typically applied in ecology)



(4) Analyze social network

- Social Network Analysis (SNA)
- Based on:
  - *Handover of work*
  - *Subcontracting*
  - *Working together*
  - *Reassignments*
  - *Doing similar tasks*

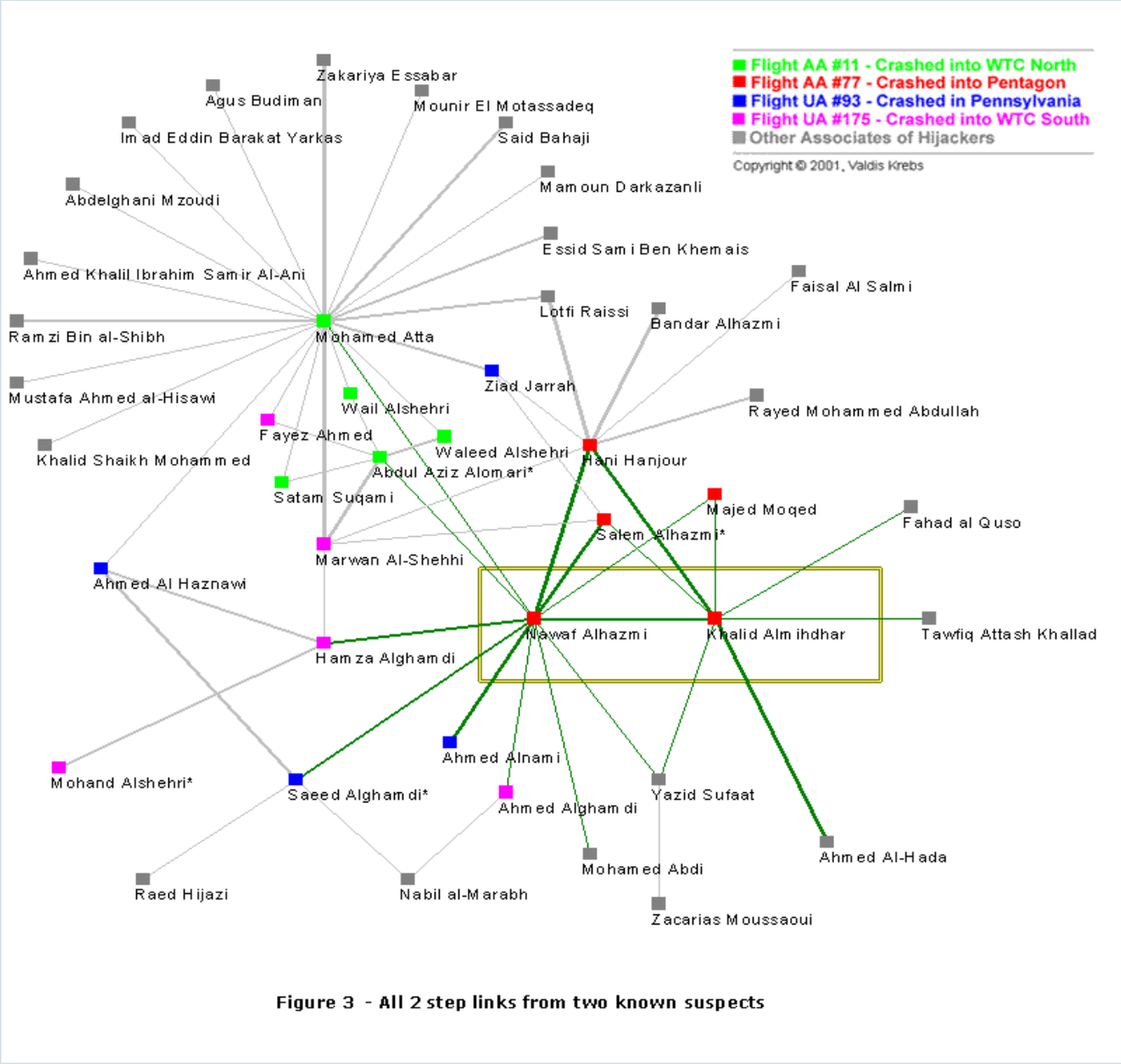


Figure 3 - All 2 step links from two known suspects

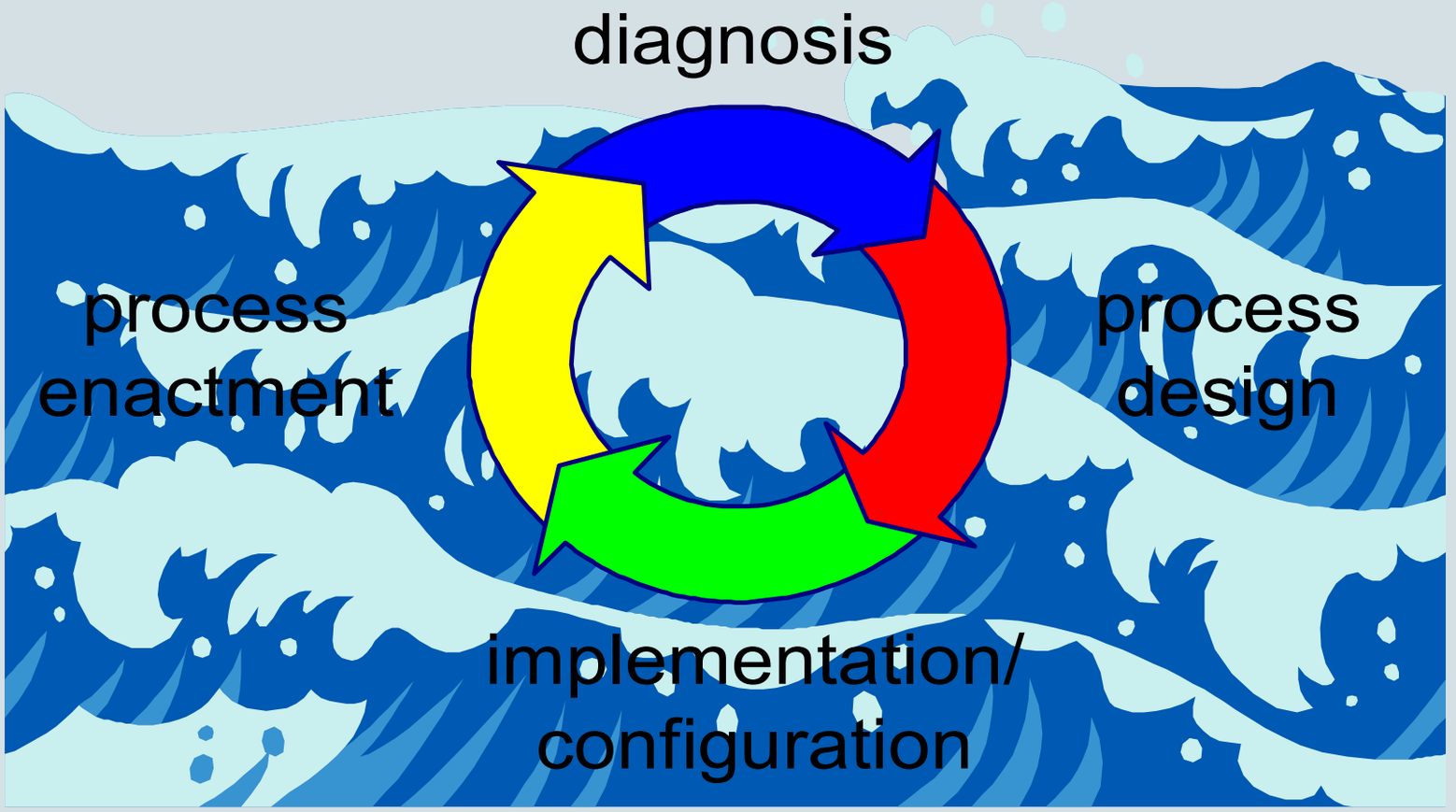
(5) Analyze performance characteristics

- Each case (process/workflow instance) has a number of properties:
  - Resource that worked on a specific activity
  - Value of a characteristic data element (e.g., size of order, age of customer, etc.)
  - Performance metrics of case (e.g., flow time)
- Using machine-learning techniques it is possible to find relevant relations between these properties.

Conclusion

Conclusion (1)

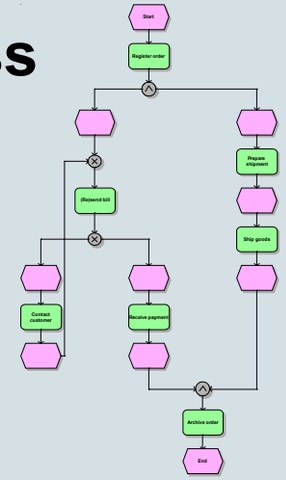
- Process mining is practically relevant and the logical next step in Business Process Management.



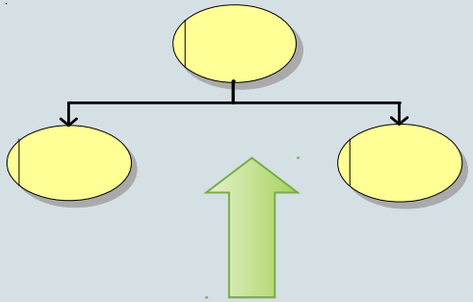
Conclusion (2)

- Process mining provides many interesting challenges for scientists, customers, users, managers, consultants, and tool developers.

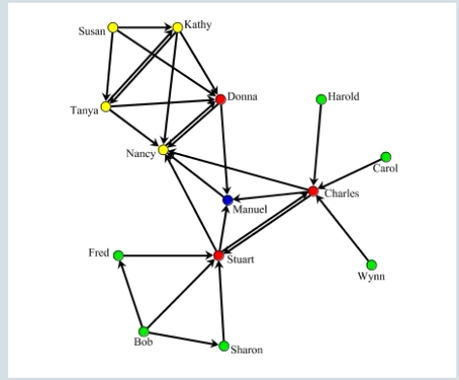
2) process model



3) organizational model



4) social network



1) basic performance metrics

5) performance characteristics

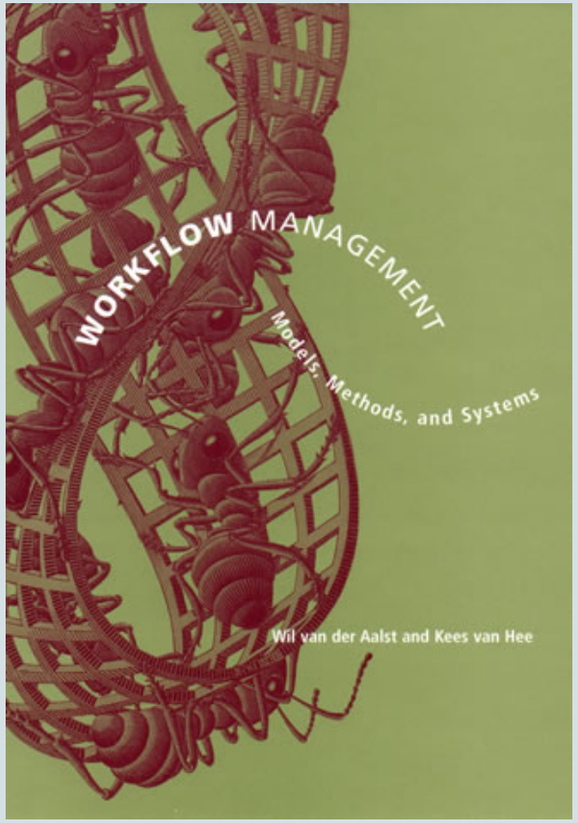
If …then …

More information

http://www.tm.tue.nl/it/research/workflow_mining.htm

http://www.tm.tue.nl/it/research/patterns

http://www.tm.tue.nl/it/staff/wvdaalst

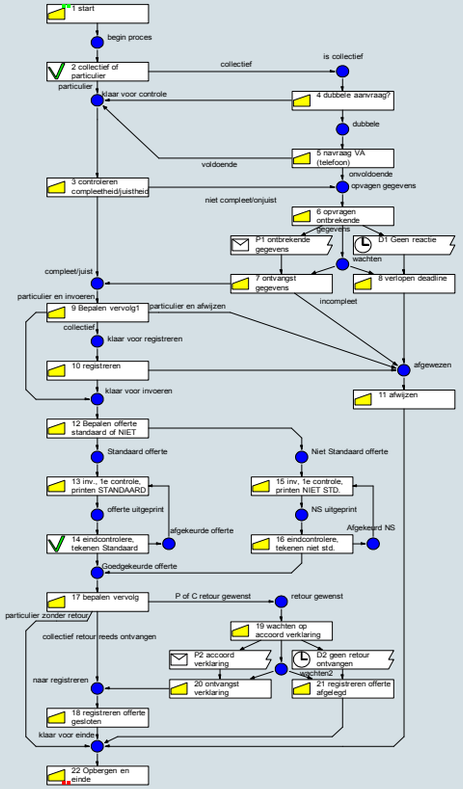


W.M.P. van der Aalst and K.M. van Hee.
Workflow Management: Models, Methods, and Systems.
MIT press, Cambridge, MA, 2002.

Appendix: A concrete algorithm

Process Mining: The alpha algorithm

alpha algorithm



Process log

- Minimal information in log: case id's and task id's.
- Additional information: event type, time, resources, and data.
- In this log there are three possible sequences:
  - ABCD
  - ACBD
  - EF

```
case 1 : task A
case 2 : task A
case 3 : task A
case 3 : task B
case 1 : task B
case 1 : task C
case 2 : task C
case 4 : task A
case 2 : task B
case 2 : task D
case 5 : task E
case 4 : task C
case 1 : task D
case 3 : task C
case 3 : task D
case 4 : task B
case 5 : task F
case 4 : task D
```

>,→,||,# relations

- **Direct succession**: x>y iff for some case x is directly followed by y.
- **Causality**: x→y iff x>y and not y>x.
- **Parallel**: x||y iff x>y and y>x
- **Choice**: x#y iff not x>y and not y>x.

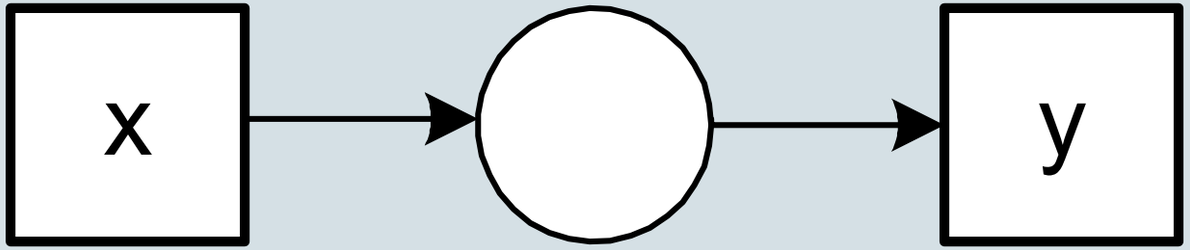
case 1	:	task A
case 2	:	task A
case 3	:	task A
case 3	:	task B
case 1	:	task B
case 1	:	task C
case 2	:	task C
case 4	:	task A
case 2	:	task B
case 2	:	task D
case 5	:	task E
case 4	:	task C
case 1	:	task D
case 3	:	task C
case 3	:	task D
case 4	:	task B
case 5	:	task F
case 4	:	task D

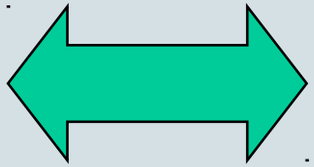
A>B
A>C
B>C
B>D
C>B
C>D
E>F

B||C
C||B

A→B
A→C
B→D
C→D
E→F

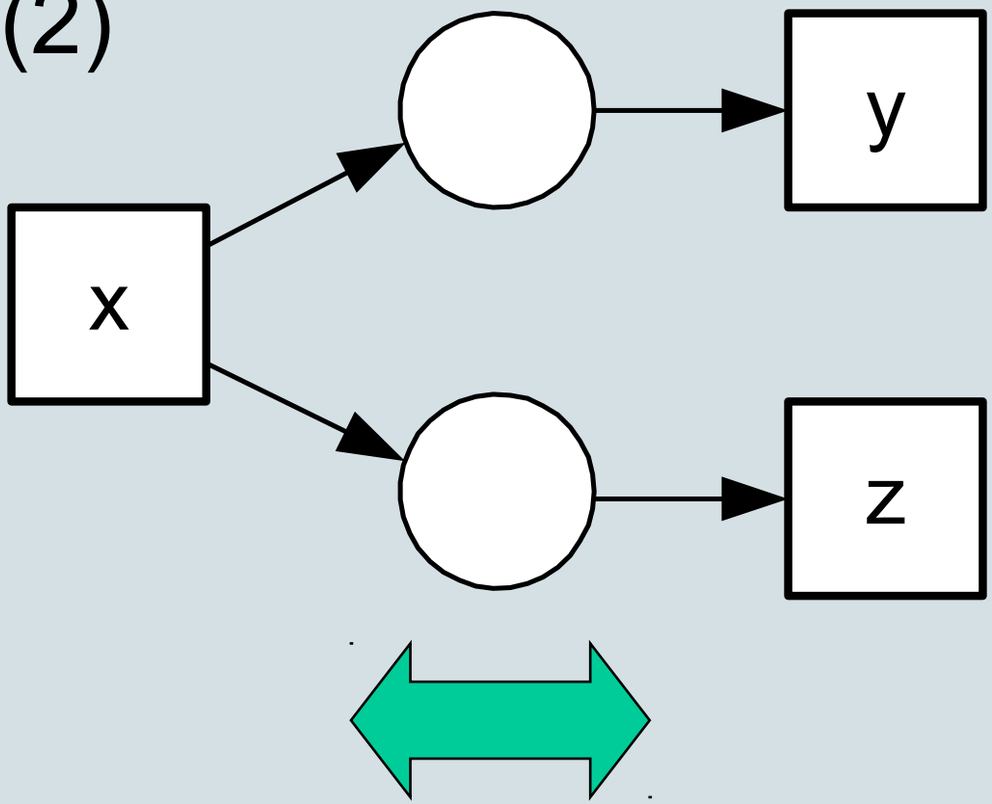
Basic idea (1)





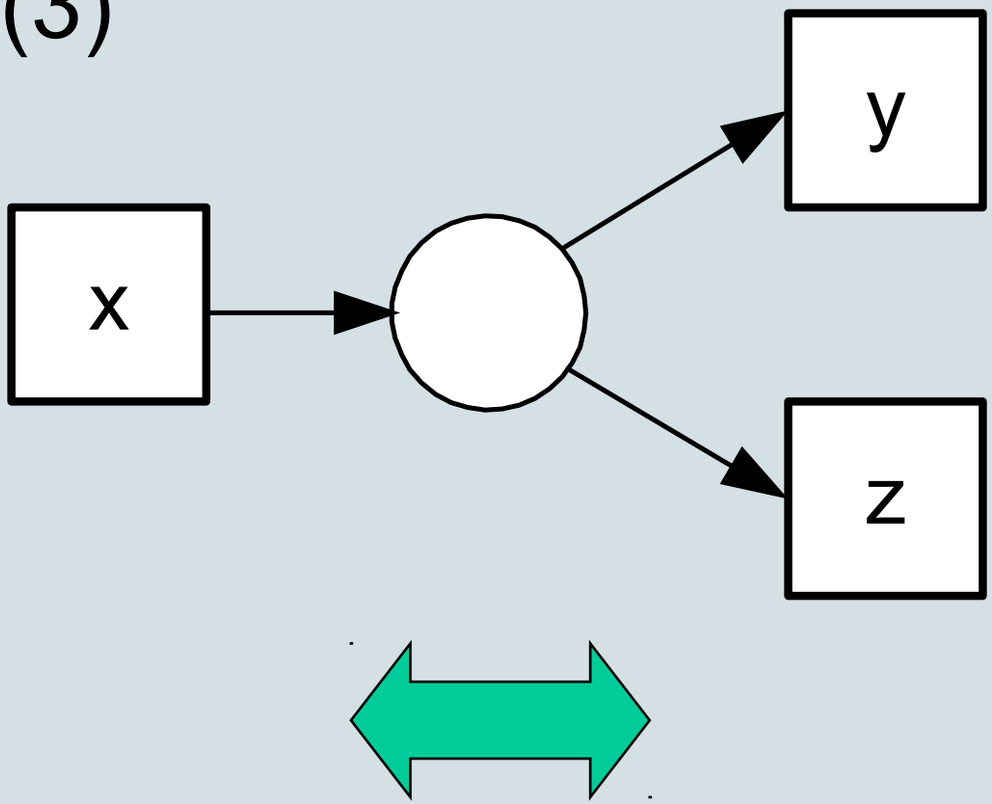
$$x \rightarrow y$$

Basic idea (2)



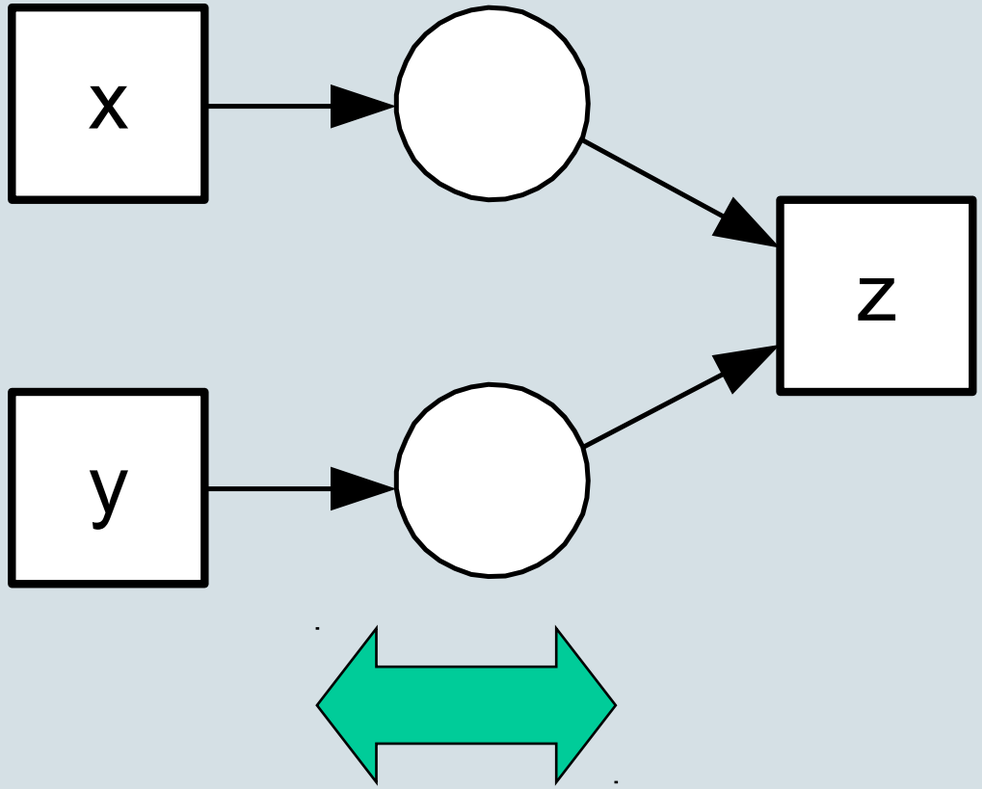
$x \rightarrow y$, $x \rightarrow z$, and $y \| z$

Basic idea (3)



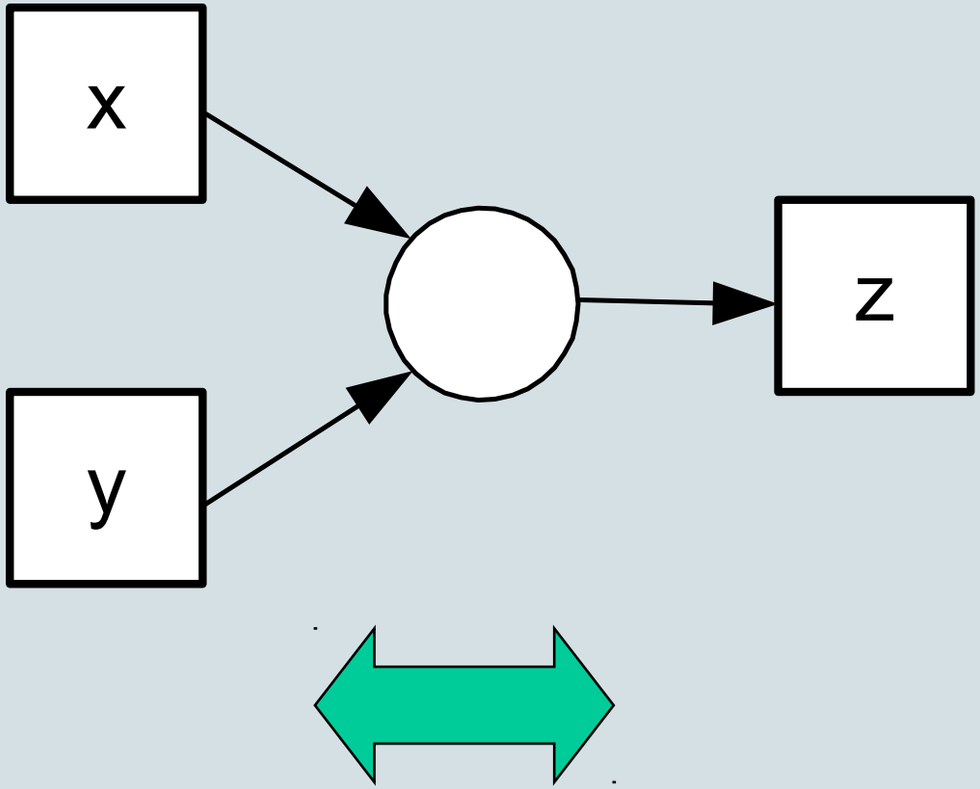
$x \rightarrow y$, $x \rightarrow z$, and $y \# z$

Basic idea (4)



$x \rightarrow z$, $y \rightarrow z$, and $x \| y$

Basic idea (5)



$x \rightarrow z$, $y \rightarrow z$, and $x \# y$

It is not that simple: Basic alpha algorithm

Let W be a workflow log over T. $\alpha(W)$ is defined as follows.

1. $T_W = \{ t \in T \mid \exists_{\sigma \in W}\ t \in \sigma \}$,
2. $T_I = \{ t \in T \mid \exists_{\sigma \in W}\ t = \mathit{first}(\sigma) \}$,
3. $T_O = \{ t \in T \mid \exists_{\sigma \in W}\ t = \mathit{last}(\sigma) \}$,
4. $X_W = \{ (A,B) \mid A \subseteq T_W \wedge B \subseteq T_W \wedge \forall_{a \in A} \forall_{b \in B}\ a \rightarrow_W b \wedge \forall_{a1,a2 \in A}\ a_1 \#_W a_2 \wedge \forall_{b1,b2 \in B}\ b_1 \#_W b_2 \}$,
5. $Y_W = \{ (A,B) \in X \mid \forall_{(A',B') \in X} A \subseteq A' \wedge B \subseteq B' \Rightarrow (A,B) = (A',B') \}$,
6. $P_W = \{ p_{(A,B)} \mid (A,B) \in Y_W \} \cup \{i_W, o_W\}$,
7. $F_W = \{ (a, p_{(A,B)}) \mid (A,B) \in Y_W \wedge a \in A \} \cup \{ (p_{(A,B)}, b) \mid (A,B) \in Y_W \wedge b \in B \} \cup \{ (i_W, t) \mid t \in T_I \} \cup \{ (t, o_W) \mid t \in T_O \}$, and
8. $\alpha(W) = (P_W, T_W, F_W)$.

Basic alpha algorithm

T_w – Tasks (mapped to transitions)

T_I – Start tasks

T_O – End tasks

1. $T_W = \{ t \in T \mid \exists_{\sigma \in W}\, t \in \sigma \}$,
2. $T_I = \{ t \in T \mid \exists_{\sigma \in W}\, t = \mathit{first}(\sigma) \}$,
3. $T_O = \{ t \in T \mid \exists_{\sigma \in W}\, t = \mathit{last}(\sigma) \}$,
4. $X_W = \{ (A,B) \mid A \subseteq T_W \wedge B \subseteq T_W \wedge \forall_{a \in A} \forall_{b \in B}\, a \rightarrow_W b \wedge \forall_{a1,a2 \in A}\, a_1 \#_W a_2 \wedge \forall_{b1,b2 \in B}\, b_1 \#_W b_2 \}$,

X_W={(A,B)} - A tasks cause B tasks, and choice among A tasks and among B tasks

5. $Y_W = \{ (A,B) \in X \mid \forall_{(A',B') \in X} A \subseteq A' \wedge B \subseteq B' \Rightarrow (A,B) = (A',B') \}$,

Y_W -pairs from X_W which are maximum sets

6. $P_W = \{ p_{(A,B)} \mid (A,B) \in Y_W \} \cup \{i_W, o_W\}$,

P_W - places between transitions

7. $F_W = \{ (a,p_{(A,B)}) \mid (A,B) \in Y_W \wedge a \in A \} \cup \{ (p_{(A,B)},b) \mid (A,B) \in Y_W \wedge b \in B \} \cup \{ (i_W,t) \mid t \in T_I \} \cup \{ (t,o_W) \mid t \in T_O \}$, and

F_W – flow relation

Basic alpha algorithm: Example

A>B
A>C
B>C
B>D
C>B
C>D
E>F

B||C
C||B

A→B
A→C
B→D
C→D
E→F

T_w ={A,B,C,D,E,F}

T_I = {A,E}

T_O={D,F}

1. $T_W = \{ t \in T \mid \exists_{\sigma \in W}\ t \in \sigma \}$,
2. $T_I = \{ t \in T \mid \exists_{\sigma \in W}\ t = first(\sigma) \}$,
3. $T_O = \{ t \in T \mid \exists_{\sigma \in W}\ t = last(\sigma) \}$,
4. $X_W = \{ (A,B) \mid A \subseteq T_W \wedge B \subseteq T_W \wedge \forall_{a \in A} \forall_{b \in B}\ a \rightarrow_W b \wedge \forall_{a1,a2 \in A}\ a_1 \#_W a_2 \wedge \forall_{b1,b2 \in B}\ b_1 \#_W b_2 \}$,

X_W={(A,B),(A,C),(B,D),(C,D),(E,F)}

5. $Y_W = \{ (A,B) \in X \mid \forall_{(A',B') \in X} A \subseteq A' \wedge B \subseteq B' \Rightarrow (A,B) = (A',B') \}$,

Y_W={(A,B),(A,C),(B,D),(C,D),(E,F)}

6. $P_W = \{ p_{(A,B)} \mid (A,B) \in Y_W \} \cup \{i_W, o_W\}$,

P_W={p_{AB},p_{AC},p_{BD},p_{CD},p_{EF}, i, o}

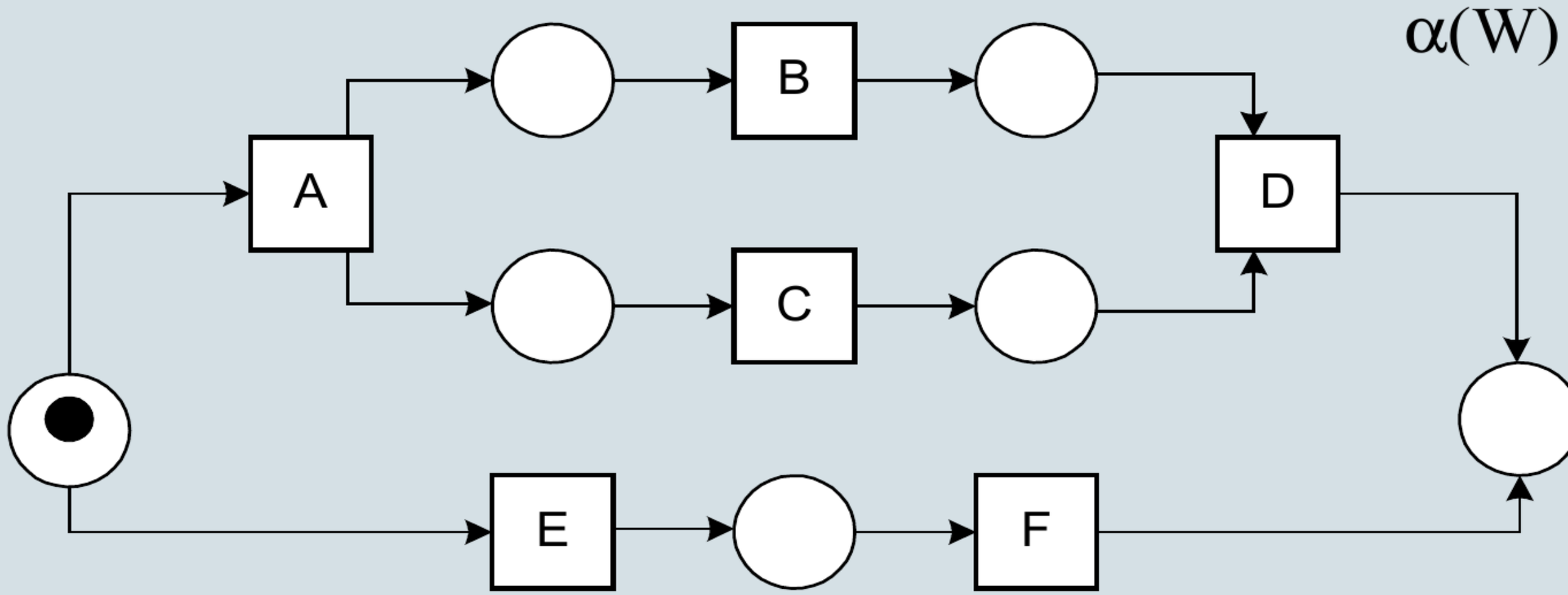
7. $F_W = \{ (a, p_{(A,B)}) \mid (A,B) \in Y_W \wedge a \in A \} \cup \{ (p_{(A,B)}, b) \mid (A,B) \in Y_W \wedge b \in B \} \cup \{ (i_W, t) \mid t \in T_I \} \cup \{ (t, o_W) \mid t \in T_O \}$, and

F_W={(A,p_{AB}), (A,p_{AC}), (B,p_{BD}), (C,p_{CD}), (E,p_{EF}),(i,A),(i,E)

($p_{AB,}$,B), (p_{AC},C), (p_{BD},D), (p_{CD},D), (p_{EF},F),(D,o),(F,o)}

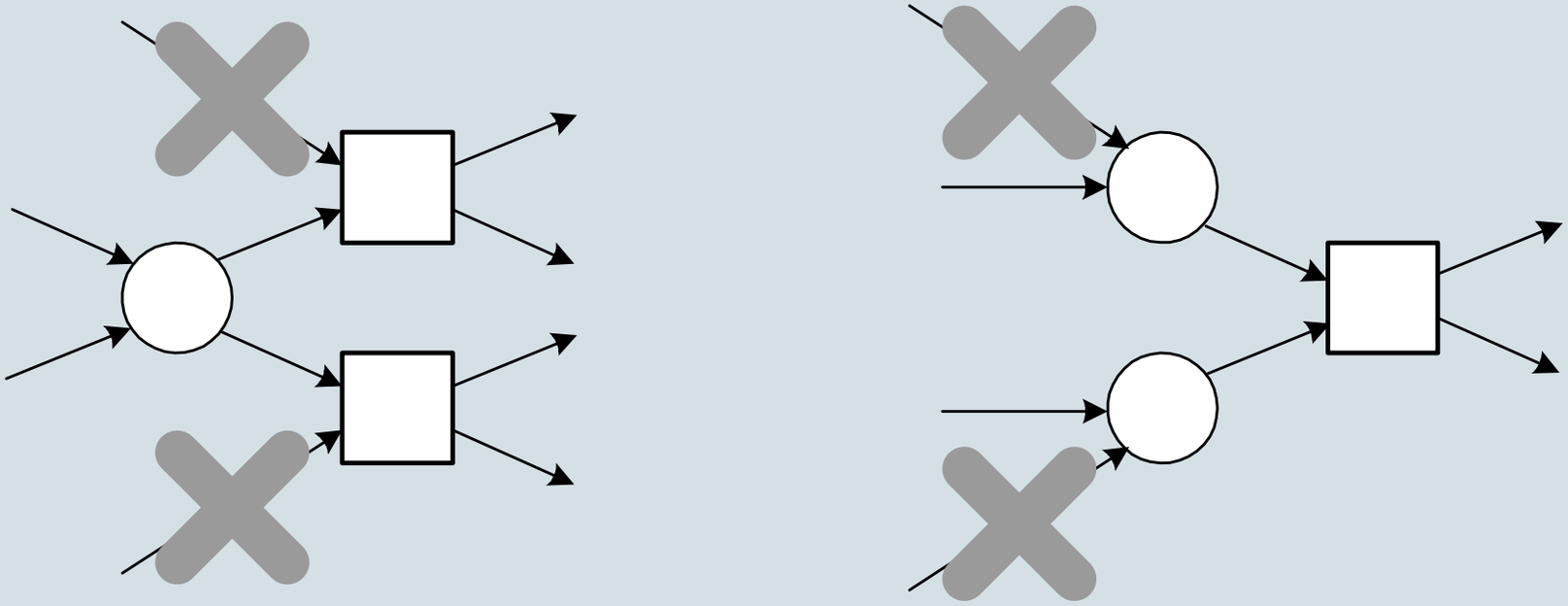
Basic alpha algorithm: Example (2)

F_W={(A,p_{AB}), (A,p_{AC}), (B,p_{BD}), (C,p_{CD}), (E,p_{EF}),(I,A),(I,E)
($p_{AB,}$,B), (p_{AC},C), (p_{BD},D), (p_{CD},D), (p_{EF},F),(D,O),(F,O)}



Results

- If log is complete with respect to relation >, it can be used to mine any SWF-net!
- *Structured Workflow Nets* (SWF-nets) have no implicit places and the following two constructs cannot be used:



(Short loops require some refinement but not a problem.)