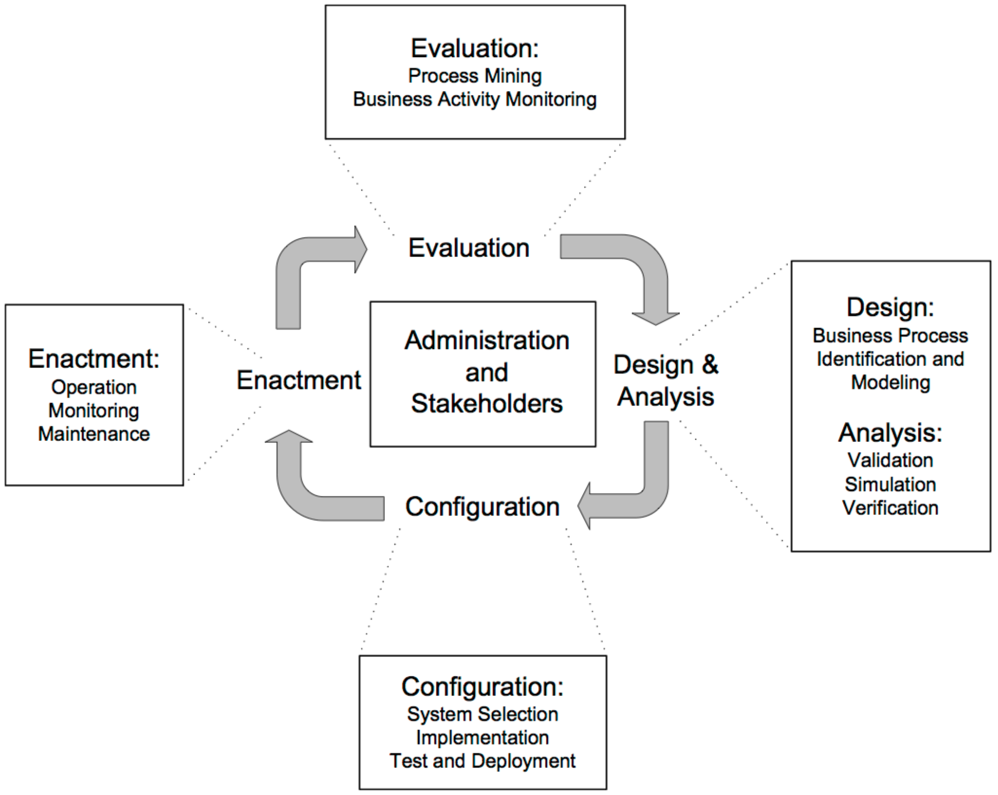
UNICAM
Università di Camerino
1336

BP Analysis: Relevant Properties

Barbara Re

BPM Life Cycle



M. Weske: Business Process Management,
© Springer-Verlag Berlin Heidelberg 2012, 2007

Validation

- The initial design must be validated by checking that **all valid process instances are reflected** by the business process model

- Useful instrument is a **workshop** where the persons involved can discuss the business process model

Simulation

- Simulation techniques can support validation

- Simulate undesired execution sequences to show deficits in the process model

- Via Simulation stakeholders can walk through the process in a step-by-step manner and to check whether the process actually exposes the desiderd behaviour

Verification

- The business process model must be analyzed and improved to make sure
  - It actually includes all desired instances
  - It does not contain any undesired properties

- Error-prone activities, to be repeated several times, for which automatic tools are necessary

Verification: Safeness and Soundness

Safeness refers to the occurrence of no more than one token along the same sequence edge of each process in the collaboration

Soundness results from the composition of the following properties

- **Option to Complete:** a process instance, once started, can always complete
- **Proper Completion:** when a process instance completes there exists no related activity of this instance which is still running or enabled
- **No dead activities:** a process model does not contain any dead activity, i.e., for each activity there exists at least one completed trace producible on that model and containing this activity

However, this notion does not take into account enqueued messages that will never be consumed.

Completeness according to the BPMN standard

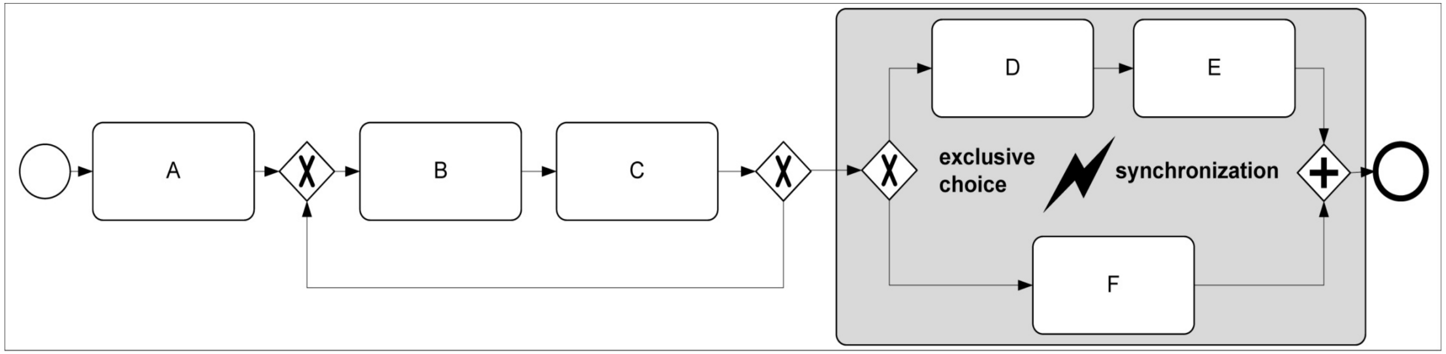
- A **process instance is completed** if and only if [...] there is no token remaining within the process instance; no activity of the process is still active

- For a **process instance** to become completed, all tokens in that instance must reach an end node and a sub-process instance completes when there are no more tokens in the Sub-Process and none of its Activities is still active

- A **collaboration *completes*** when all involved processes complete

Example: Option to Complete

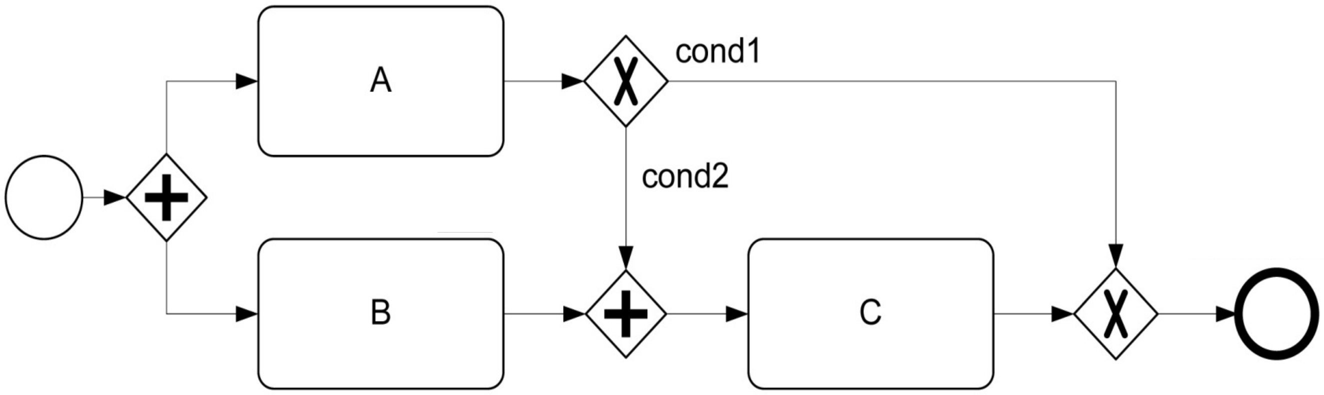
A process instance, once started, can always complete.



M. Reichert, B.Weber:
Enabling Flexibility in Process-Aware Information Systems,
@ Springer-Verlag Berlin Heidelberg 2012

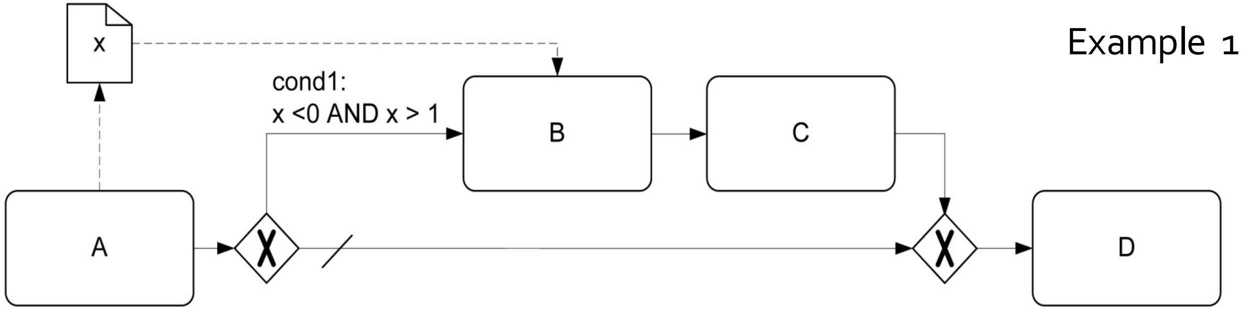
Example: Proper Completion

When a process instance completes there exists no related activity of this instance which is still running or enabled.



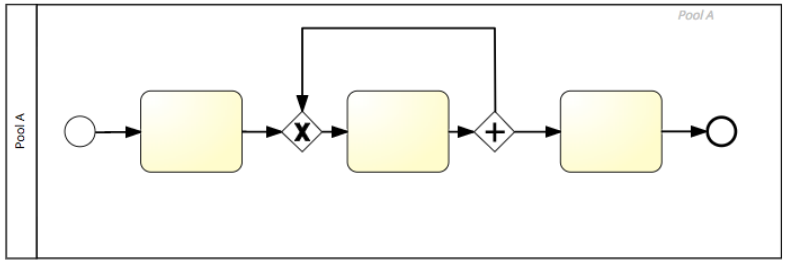
Example: Dead Activity

A process model does not contain any dead activity, i.e., for each activity there exists at least one completed trace producible on that model and containing this activity.

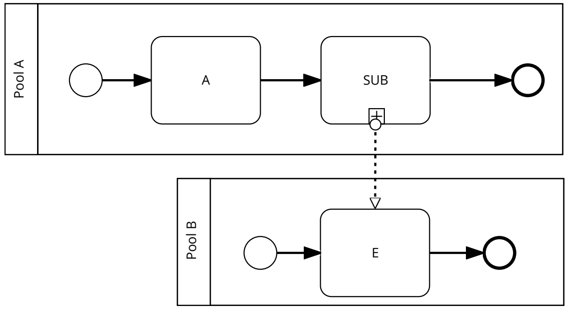


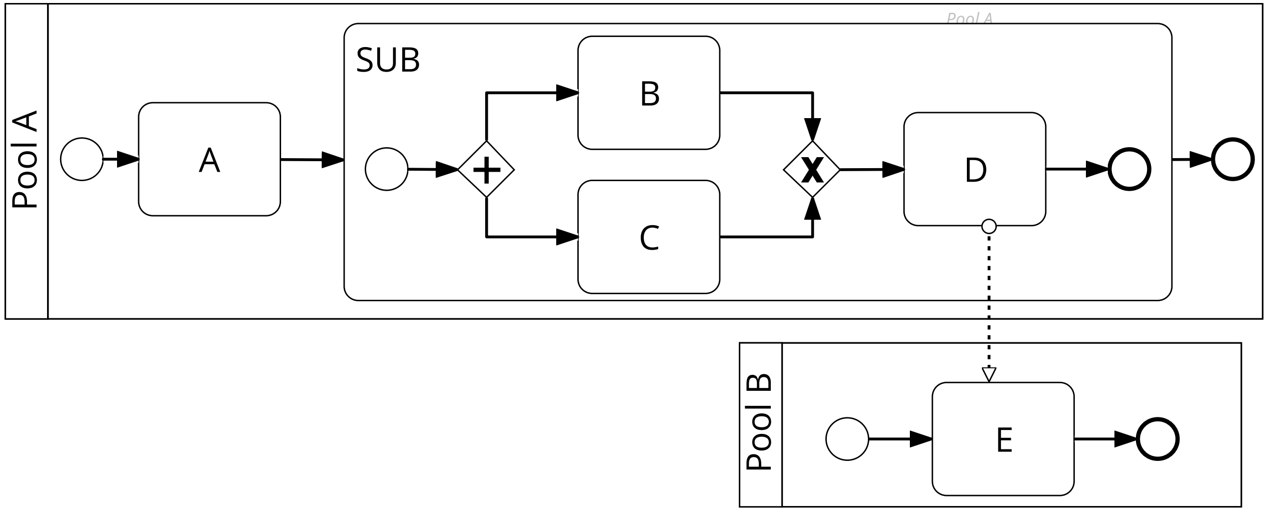
Example 1

Example 2



What about the sub-process and the collaborations?





Tool Support

- **S^3 S**oundness, message-disregarding **S**oundness, and **S**afeness verifier

http://pros.unicam.it:8080/S3/modeler/

Data Flow Correctness

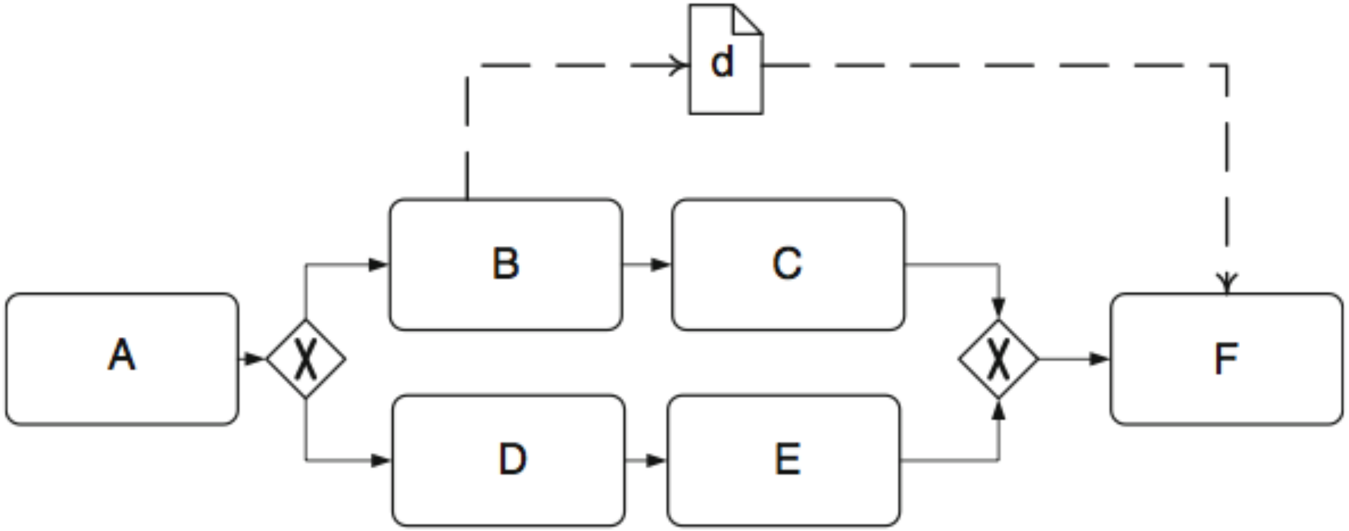
- **No missing data:** the data flow schema of a process model might cause missing data at run-time if a data object exists which can be read during run-time without having been written by any preceding activity or provided by the outside environment (i.e., by a start message)

- **Unnecessary data**: A data object written by an activity of process model is called unnecessary if it is not read by any subsequent activity or transition condition or passed to the outside environment via an end message

- **No lost updates:** The data flow schema of a process model might cause lost data at run-time if a data object, which is written by an activity, is updated by a subsequent activity, but without reading the data object in between

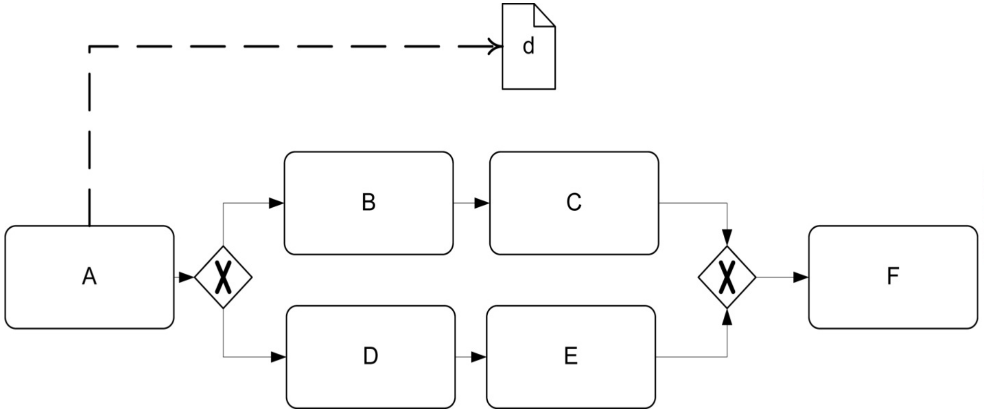
Example: Missing Data

The data flow schema of a process model might cause missing data at run-time if a data object exists which can be read during run-time without having been written by any preceding activity or provided by the outside environment (i.e., by a start message).



Data Flow Errors: Unnecessary Data

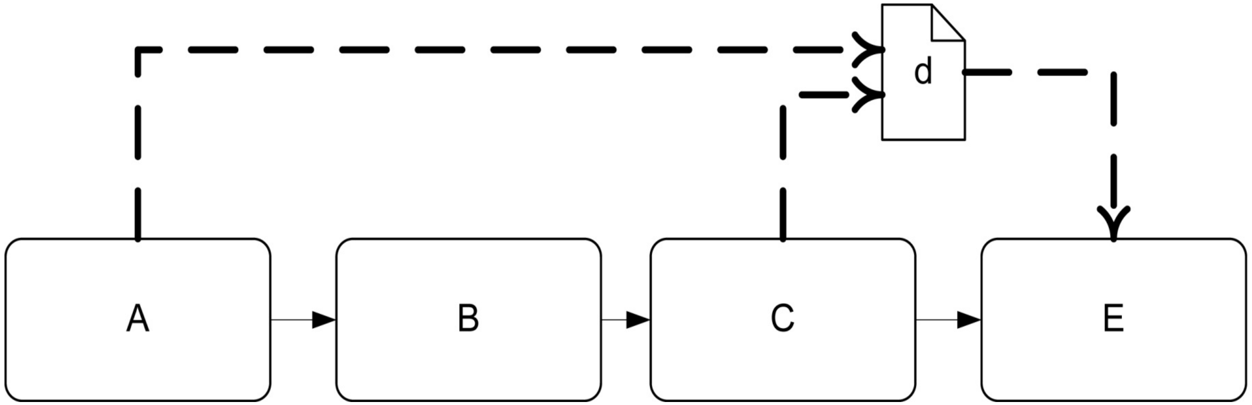
A data object written by an activity of process model is called unnecessary if it is not read by any subsequent activity or transition condition or passed to the outside environment via an end message.



M. Reichert, B.Weber:
Enabling Flexibility in Process-Aware Information Systems,
@ Springer-Verlag Berlin Heidelberg 2012

Data Flow Errors: Lost Updates

The data flow schema of a process model might cause lost data at run-time if a data object, which is written by an activity, is updated by a subsequent activity, but without reading the data object in between.



M. Reichert, B.Weber: Enabling Flexibility in Process-Aware Information Systems, @ Springer-Verlag Berlin Heidelberg 2012