BP Patterns

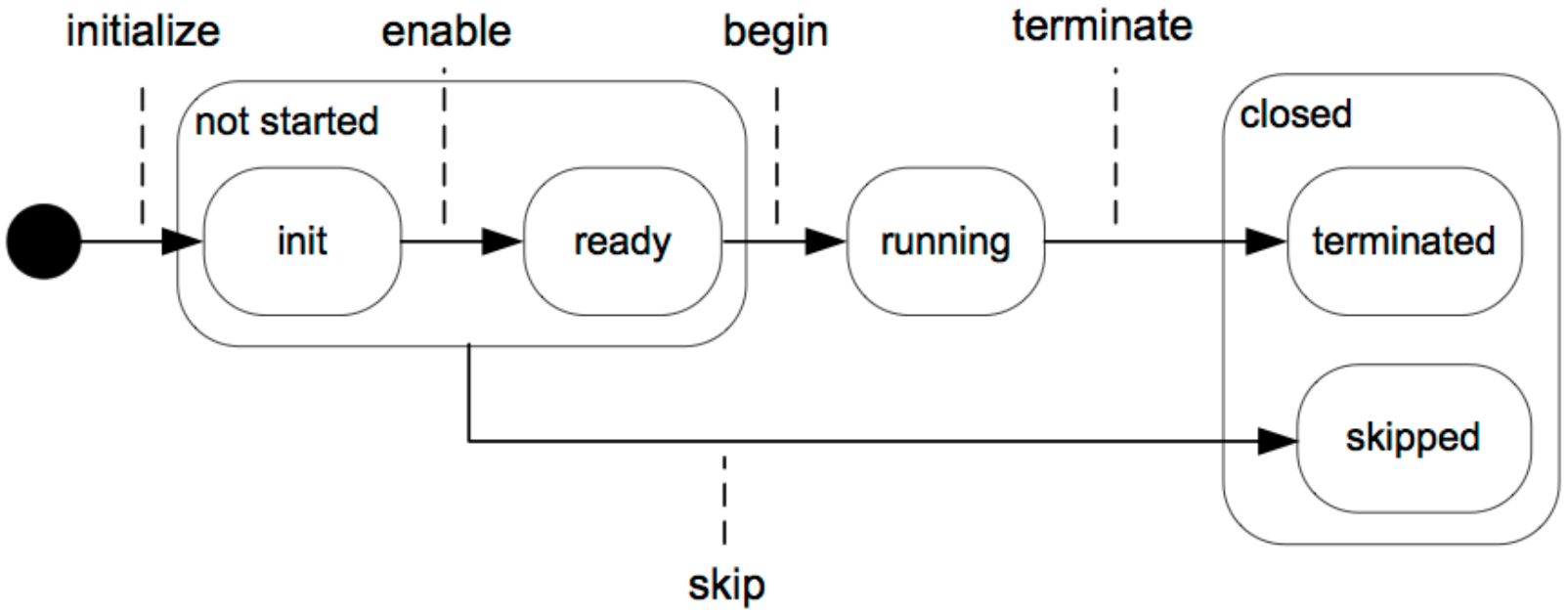
Business Process Management and Flexibility

Barbara Re, Phd

Control flow patterns

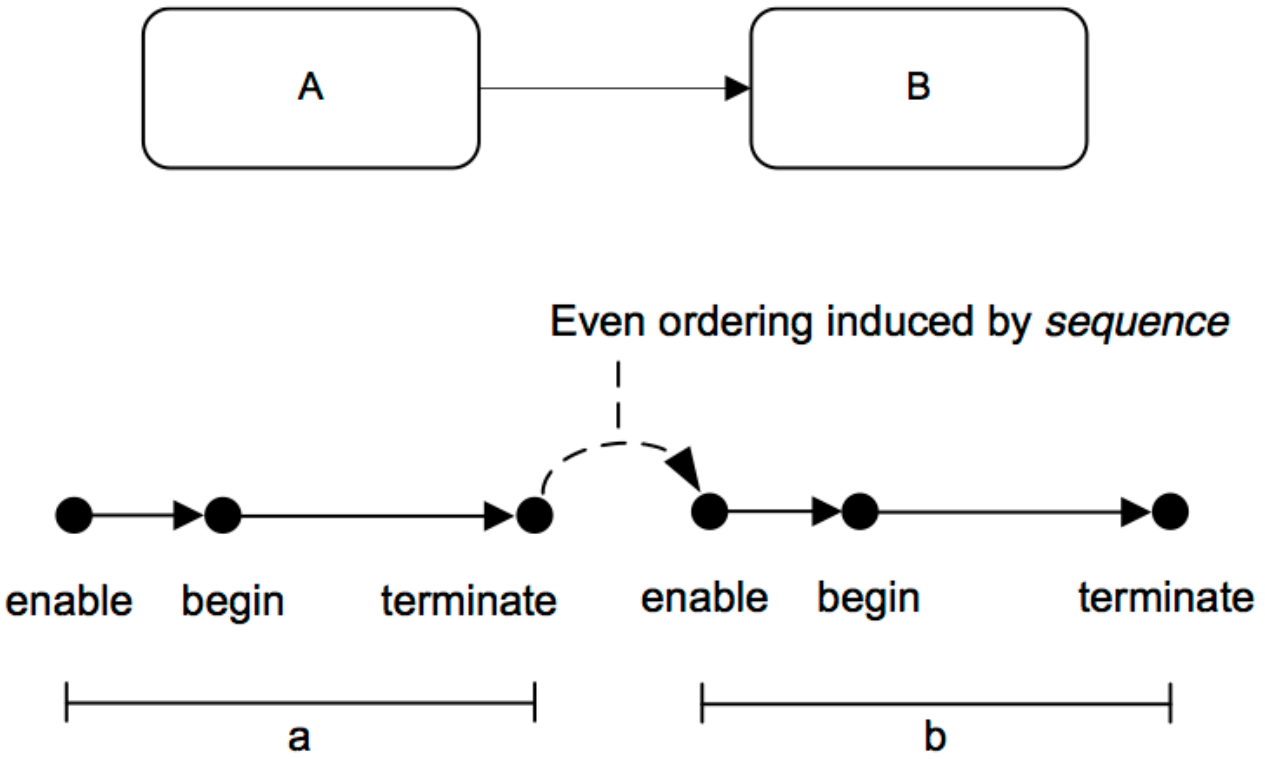
- Control flow patterns provide a way for expressing process orchestrations
- Control flow patterns are **independent of concrete process languages**, so that each pattern can be expressed in **different process languages**
- Control flow patterns can also be **used to compare the expressiveness of process languages**
- **Basic control flow patterns** include *sequence*, *and split*, and *and join*, as well as *exclusive or split* and *exclusive or join*
- These control flow patterns are supported by virtually any process meta-model
- Control flow patterns are **defined at the process model level** and their **execution semantics is applies at process instances**

State transition diagram for activity instance



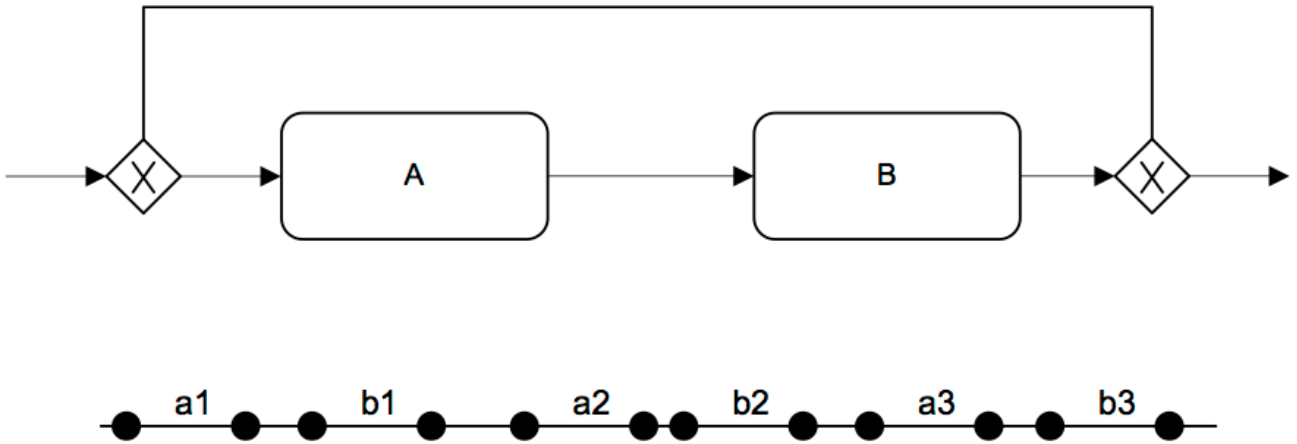
M. Weske: Business Process Management,
© Springer-Verlag Berlin Heidelberg 2012, 2007

Sequence pattern, with event diagram process instance



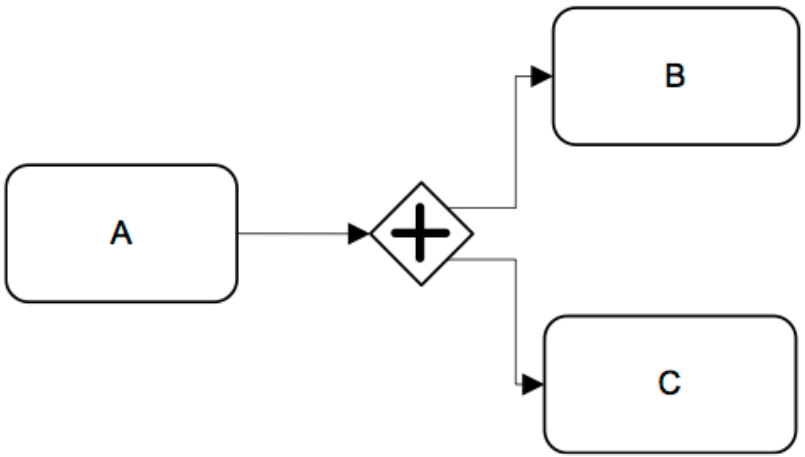
M. Weske: Business Process Management,
© Springer-Verlag Berlin Heidelberg 2012, 2007

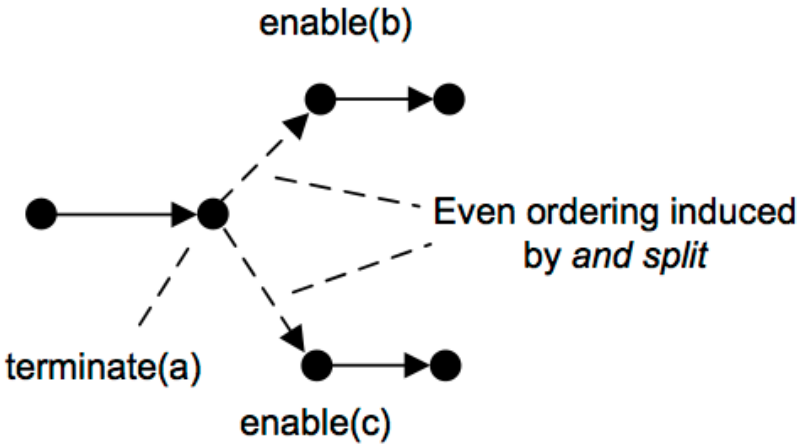
Sequence pattern as part of loop



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

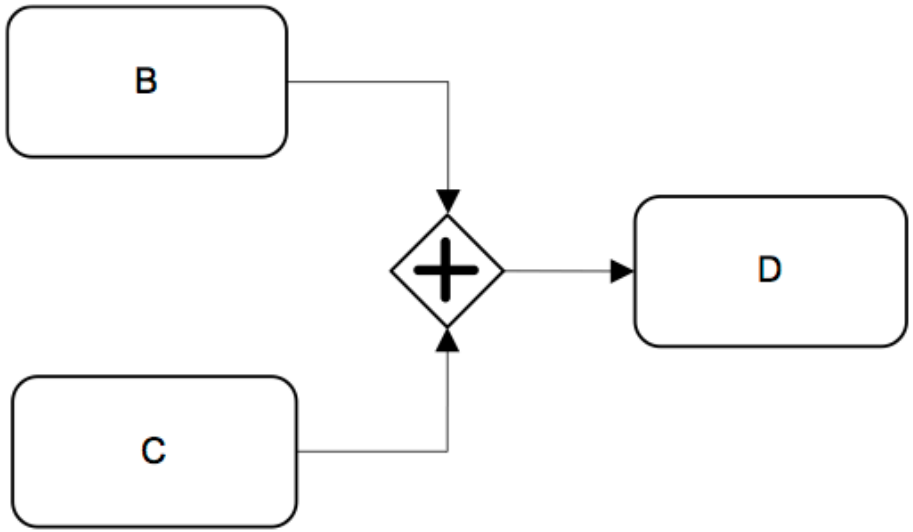
And-split pattern

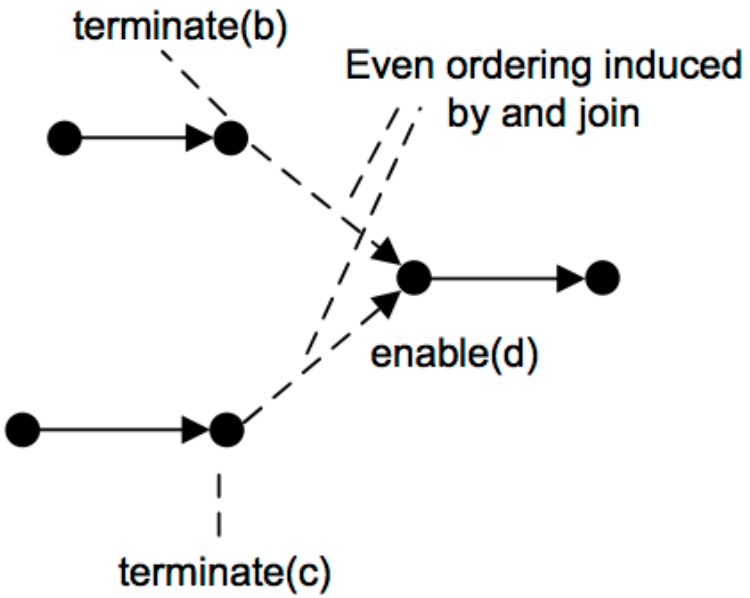




M. Weske: Business Process Management,
© Springer-Verlag Berlin Heidelberg 2012, 2007

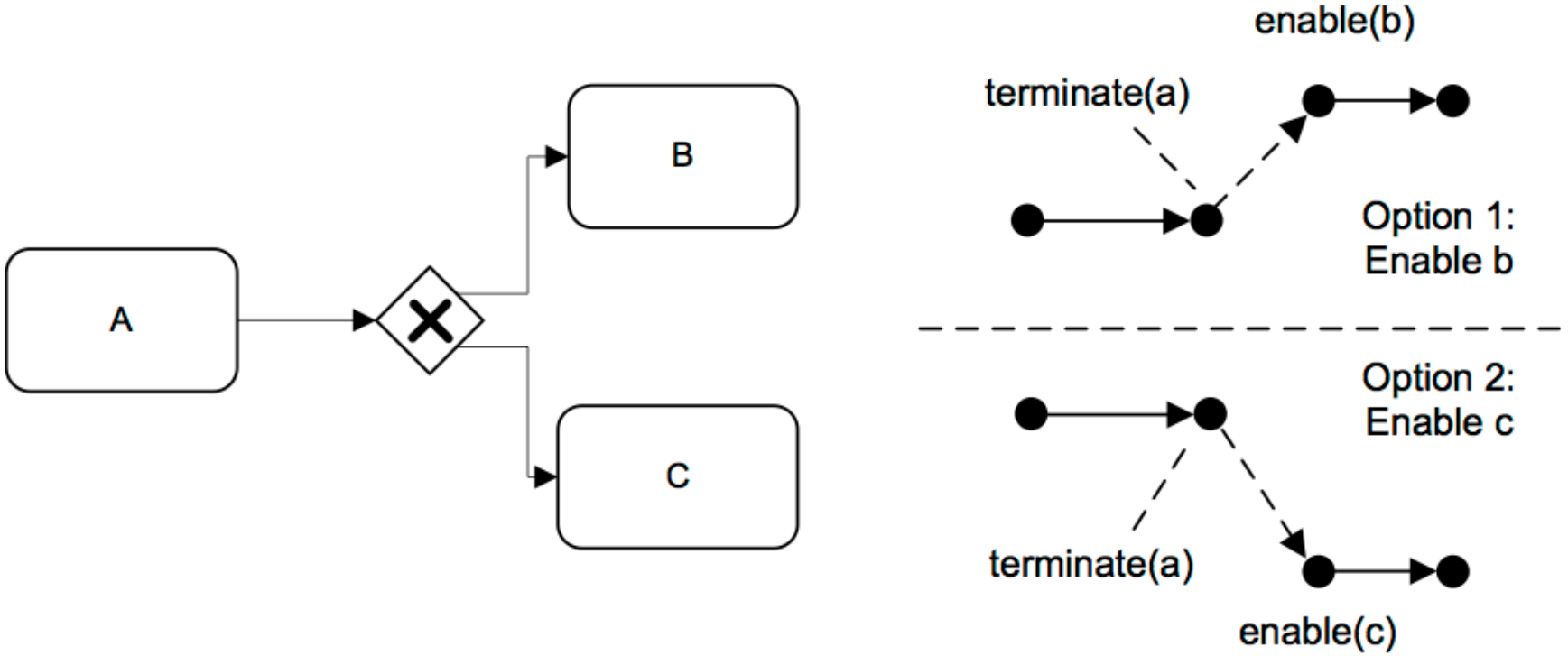
And join pattern





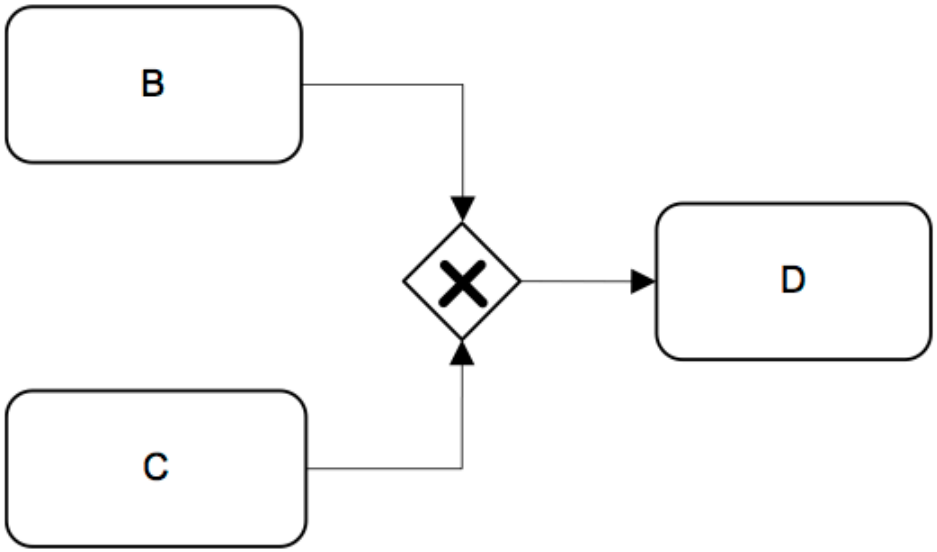
M. Weske: Business Process Management,
© Springer-Verlag Berlin Heidelberg 2012, 2007

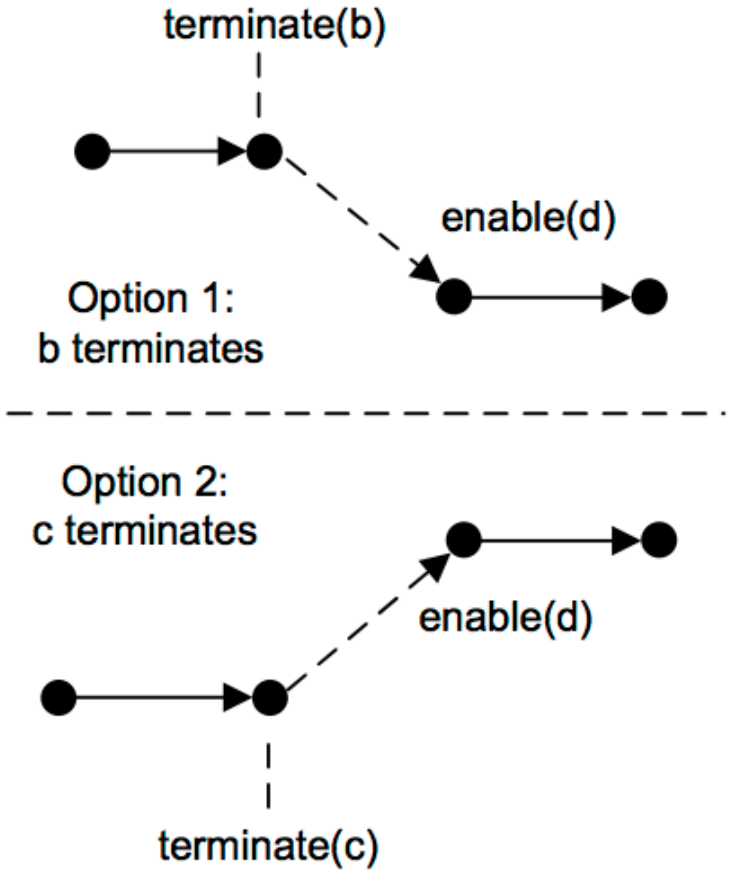
Xor split pattern



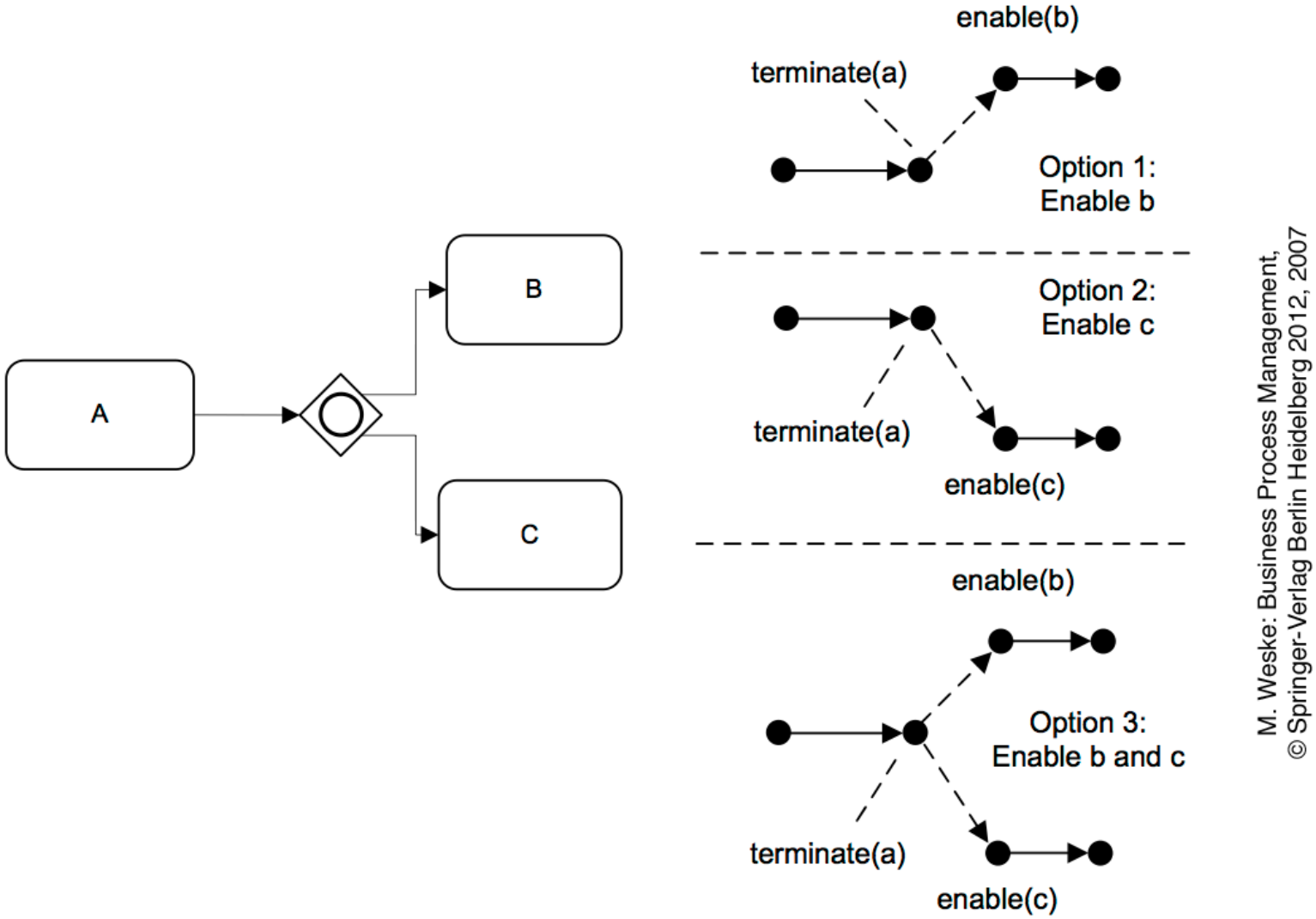
M. Weske: Business Process Management,
© Springer-Verlag Berlin Heidelberg 2012, 2007

Xor join pattern

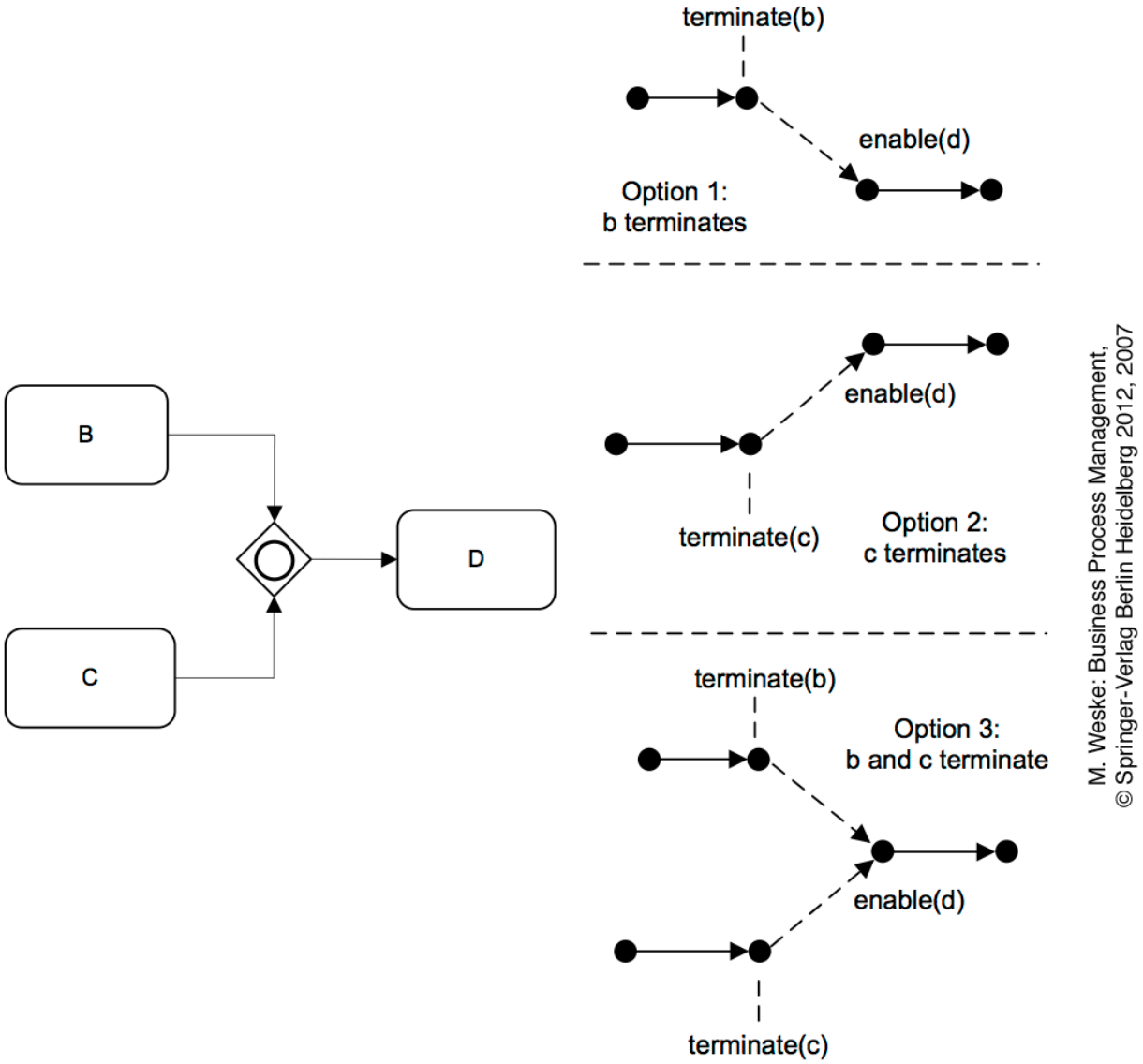




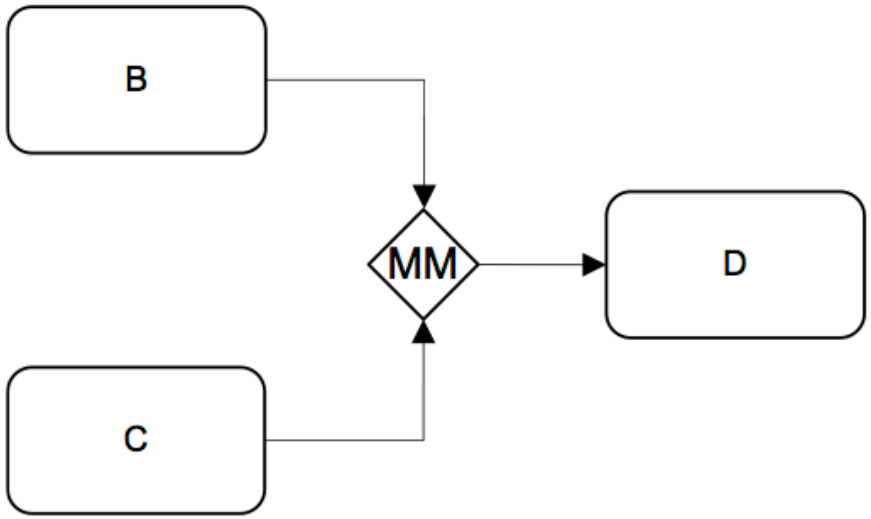
Or split pattern

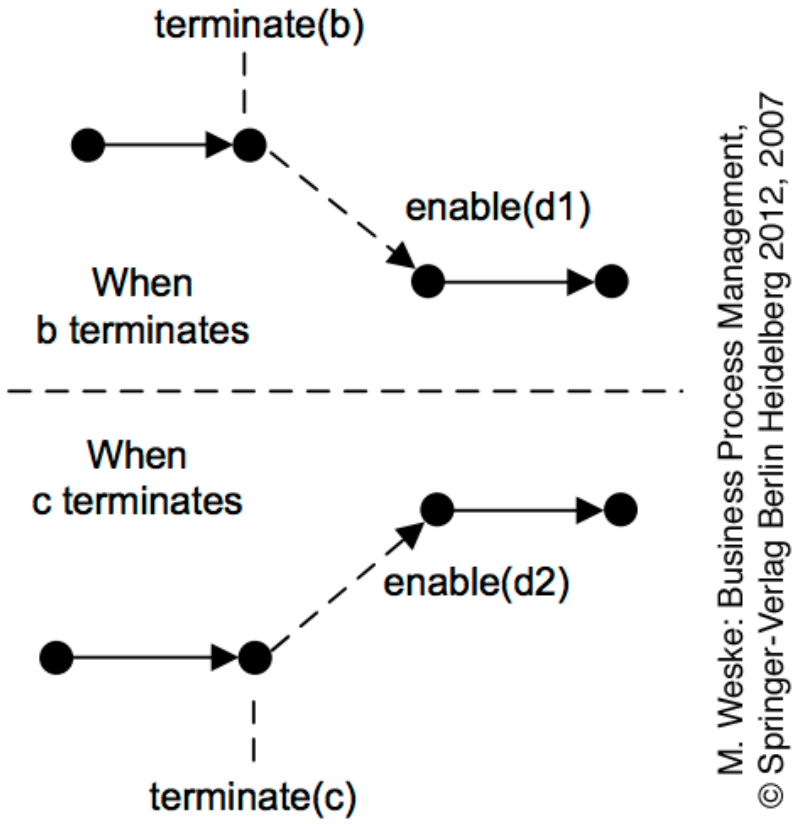


Or join pattern



Multi-merge pattern

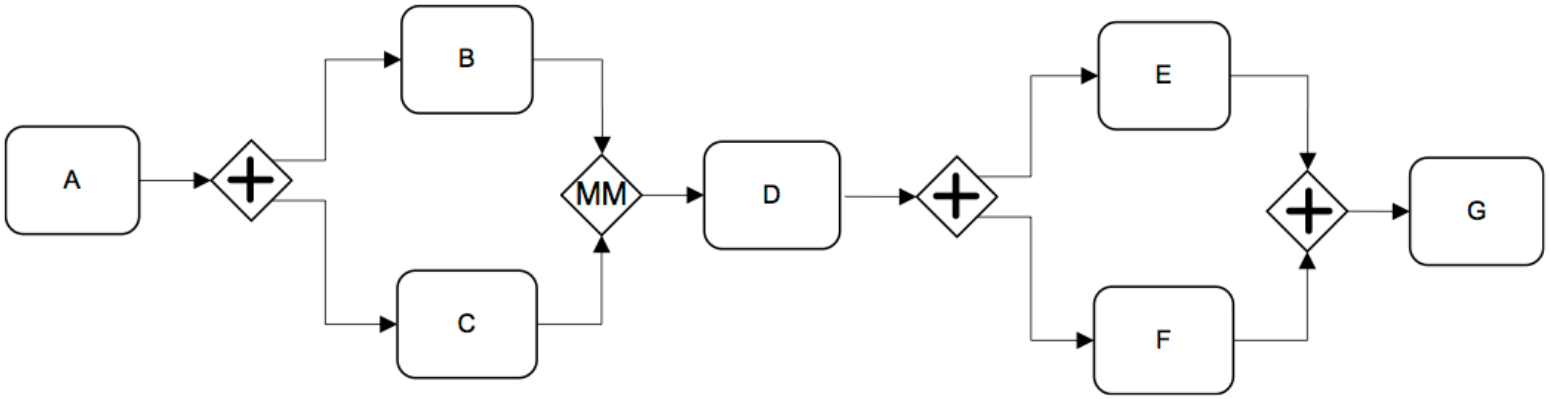




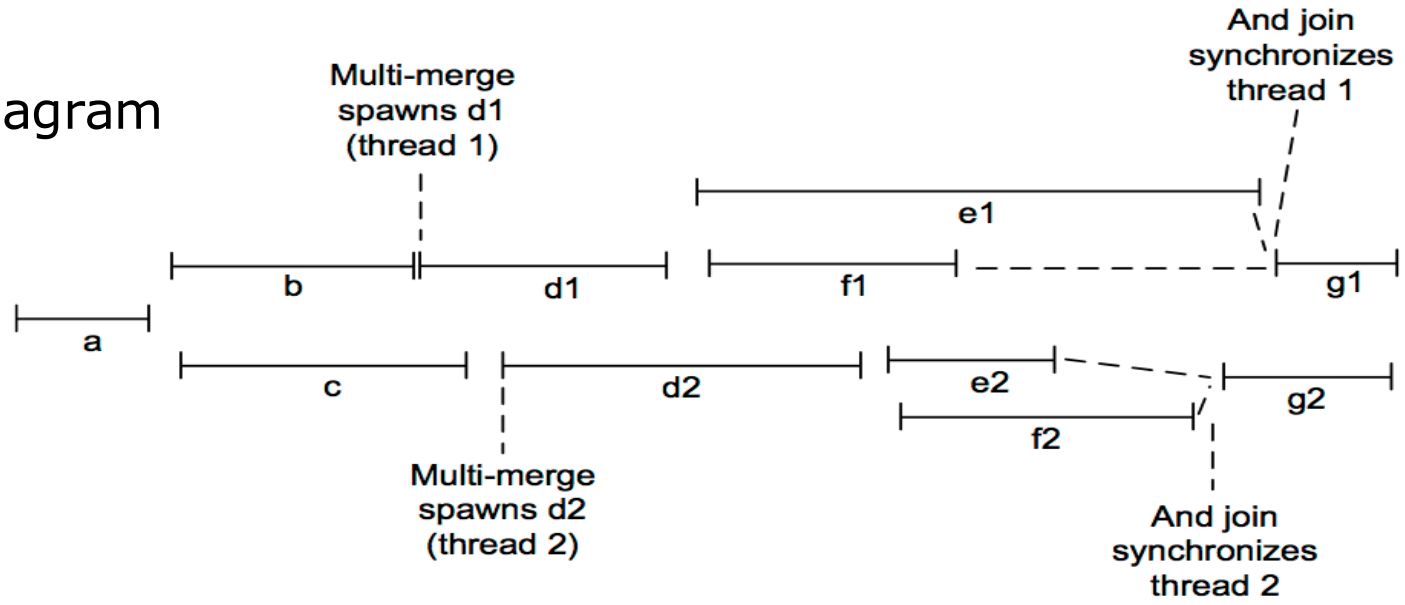
The activity following the merge is started for every activation of every incoming branch

Multi-merge example

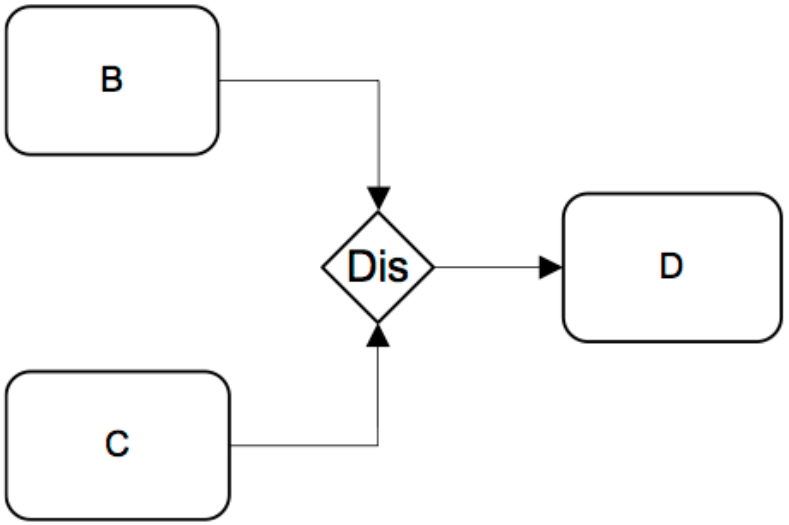
Process Model

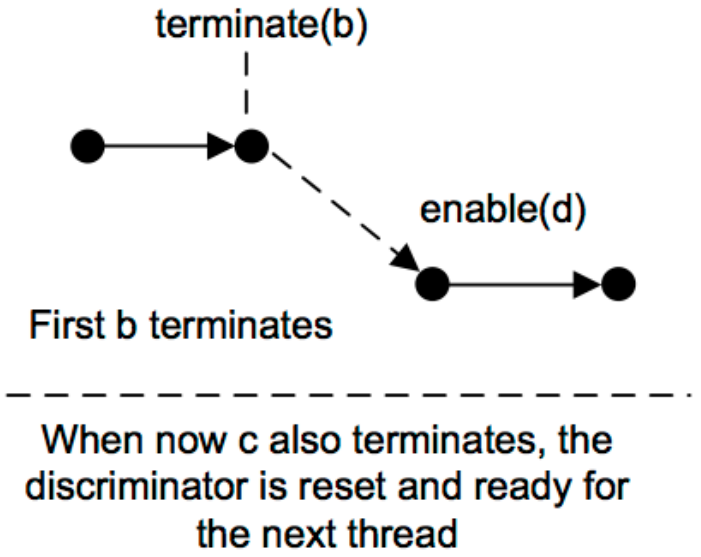


Event Diagram



Discriminator pattern

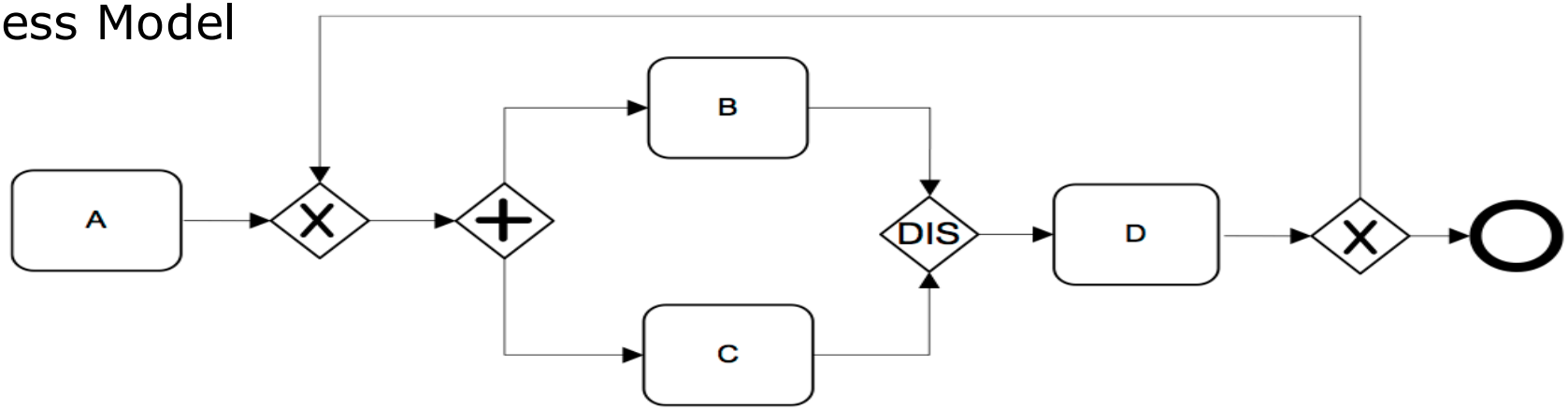




M. Weske: Business Process Management,
© Springer-Verlag Berlin Heidelberg 2012, 2007

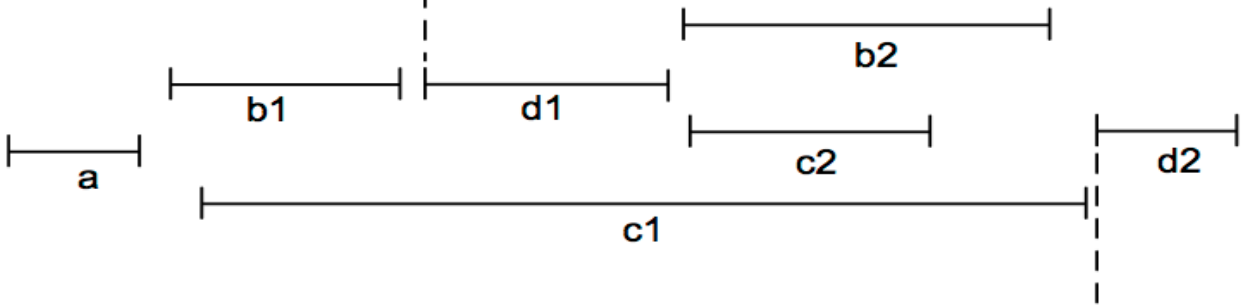
Discriminator Example

Process Model



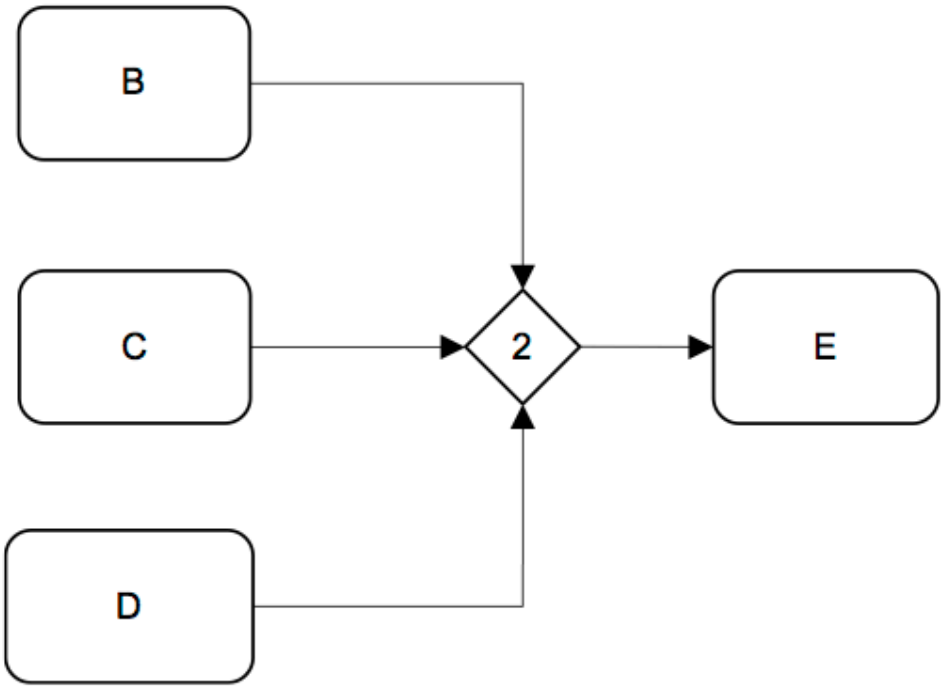
Discriminator spawns off d1 on termination of b1, while c1 is still running

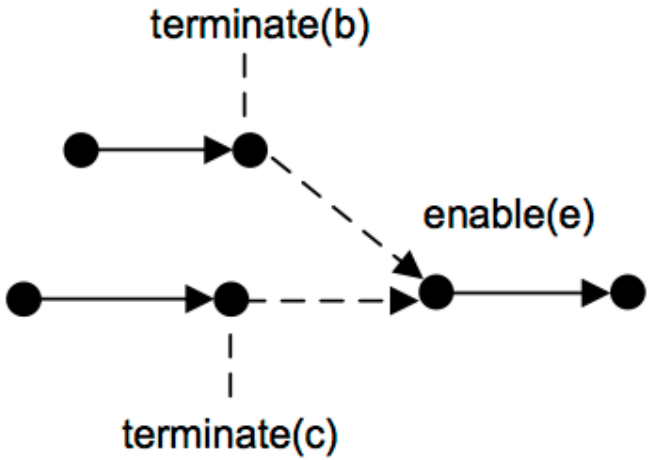
Event Diagram



Discriminator makes sure that d2 only enabled after c1 of first iteration completes

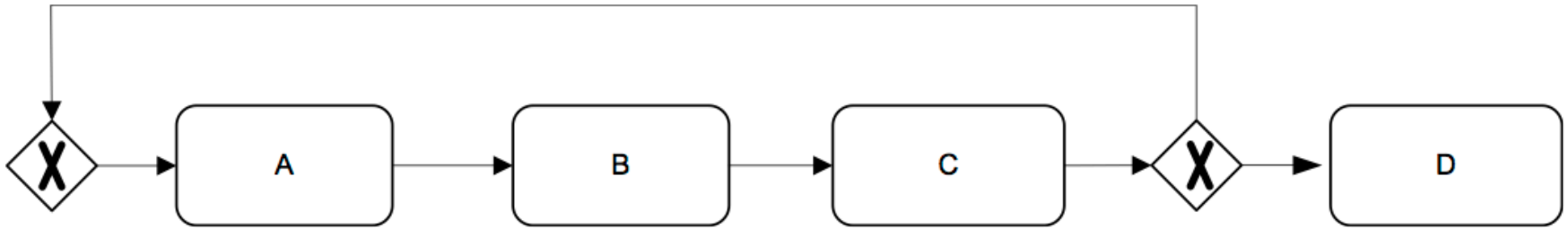
N-out-of-M join pattern





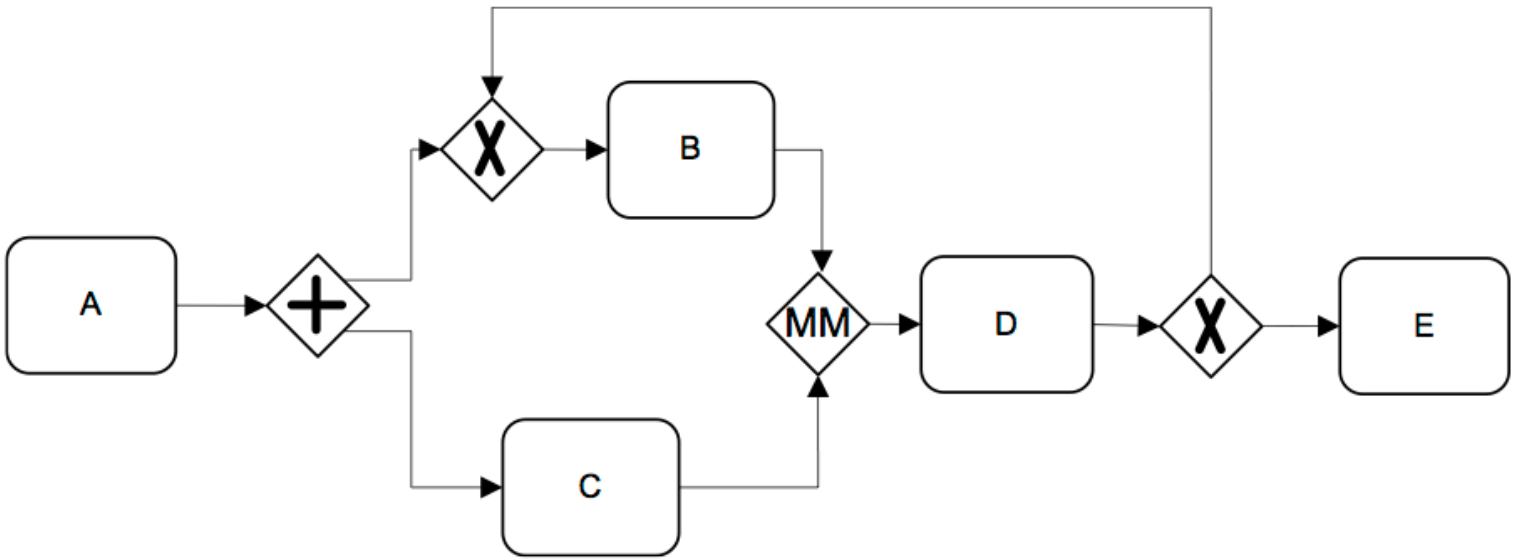
When any 2 activity instances in {b,c,d} have terminates, e can be enabled
(in the example b and c terminated)

Arbitrary cycles pattern – graphical representation



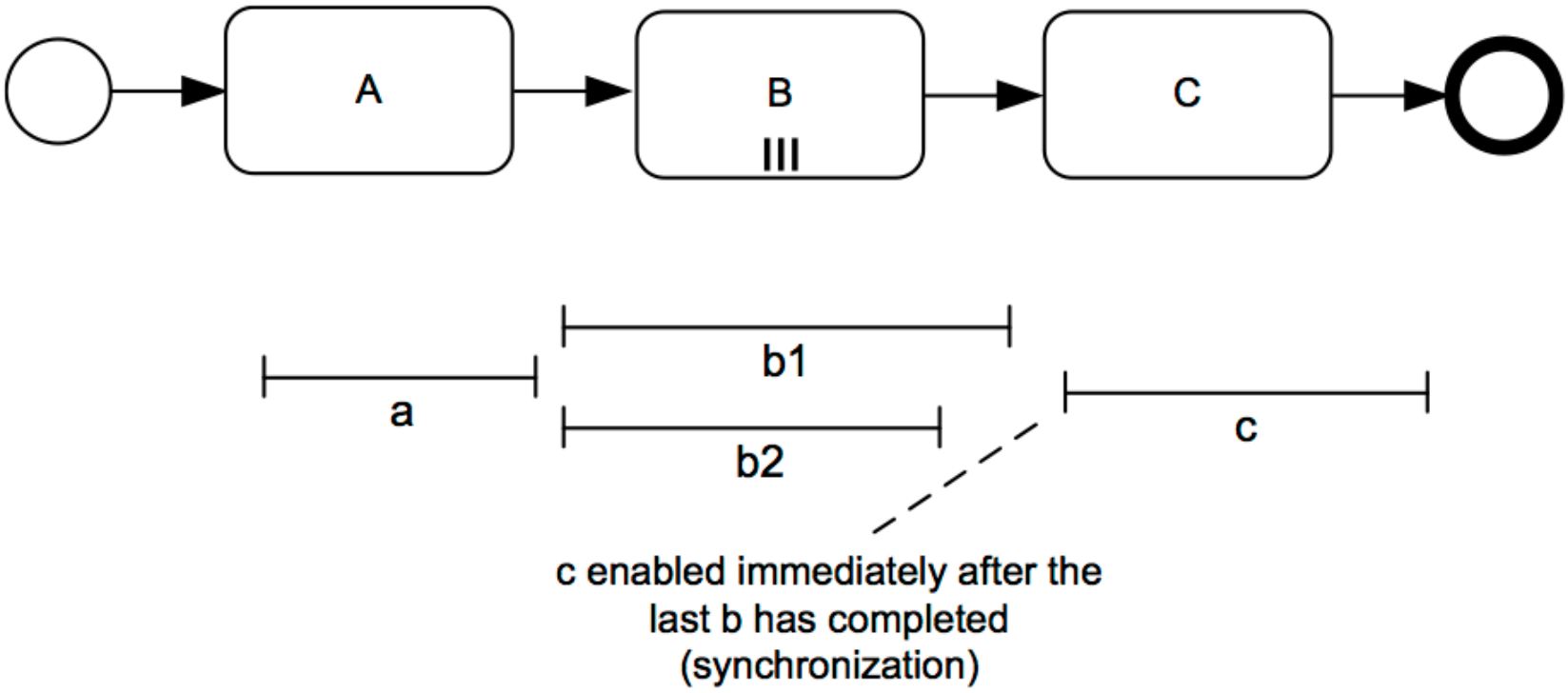
M. Weske: Business Process Management,
© Springer-Verlag Berlin Heidelberg 2012, 2007

Arbitrary cycles example – using multiple merge pattern



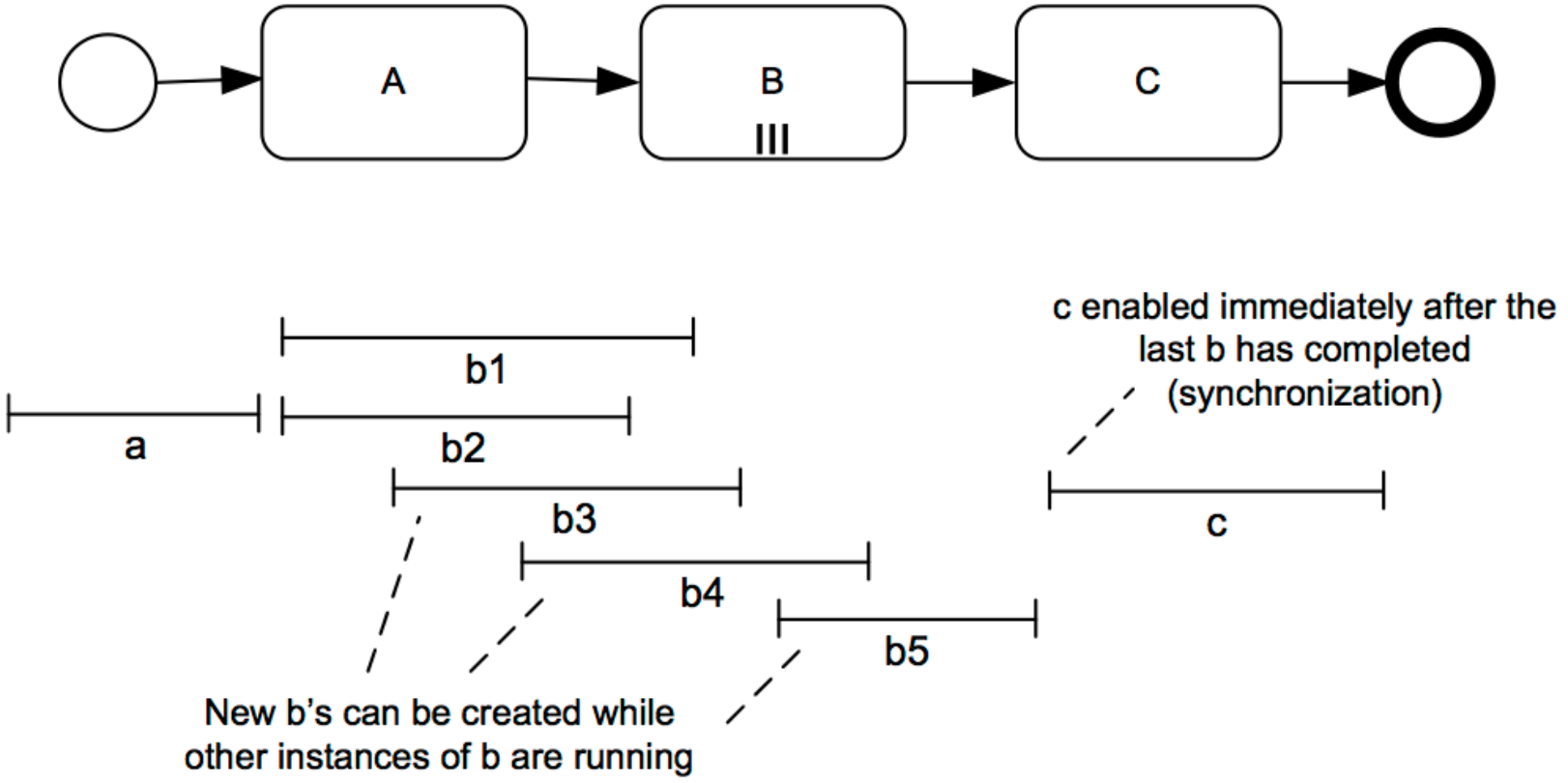
M. Weske: Business Process Management,
© Springer-Verlag Berlin Heidelberg 2012, 2007

Example for multiple instances with a priori design time knowledge



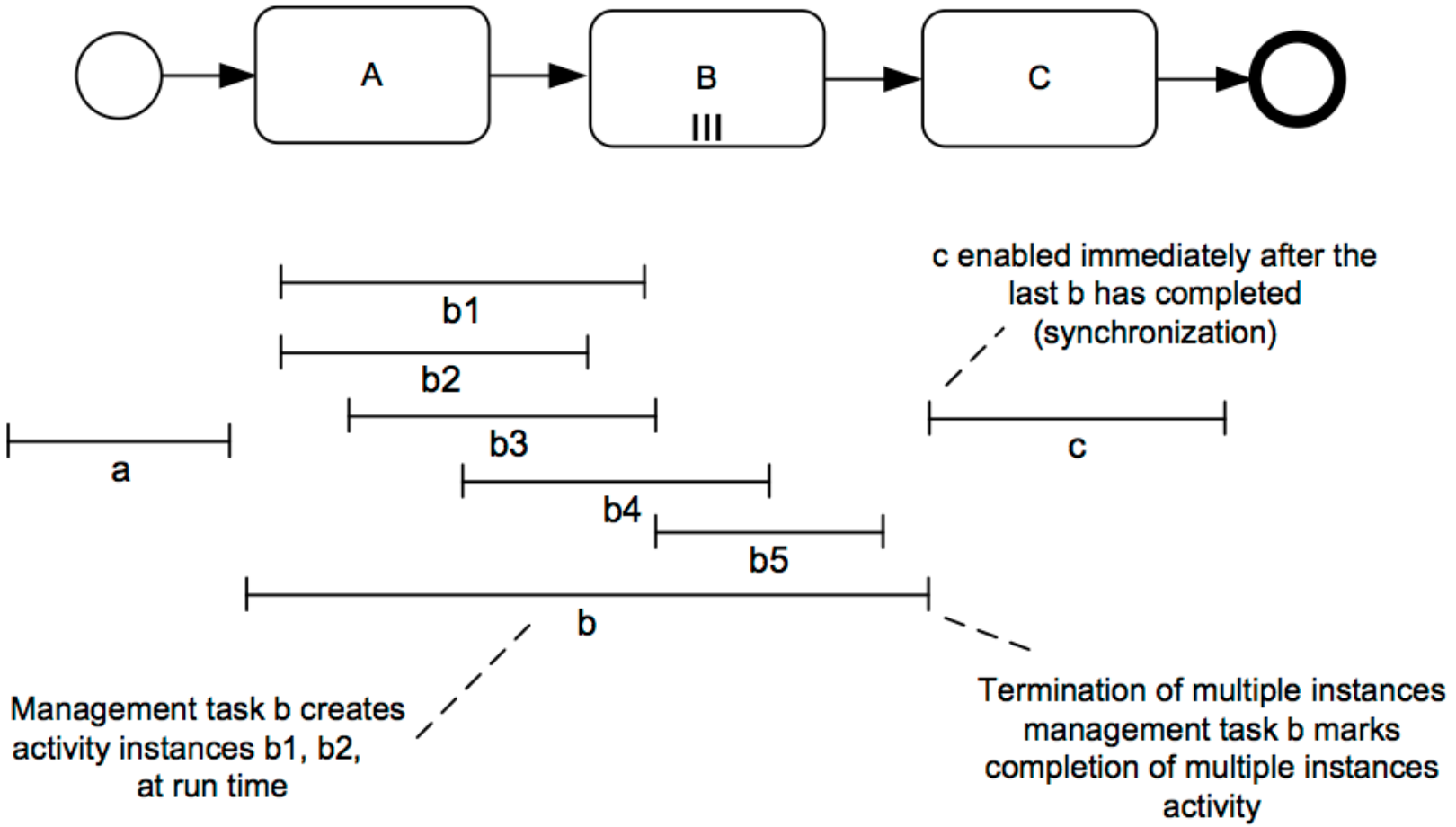
M. Weske: Business Process Management,
© Springer-Verlag Berlin Heidelberg 2012, 2007

Example for multiple instances without a priori run time knowledge pattern

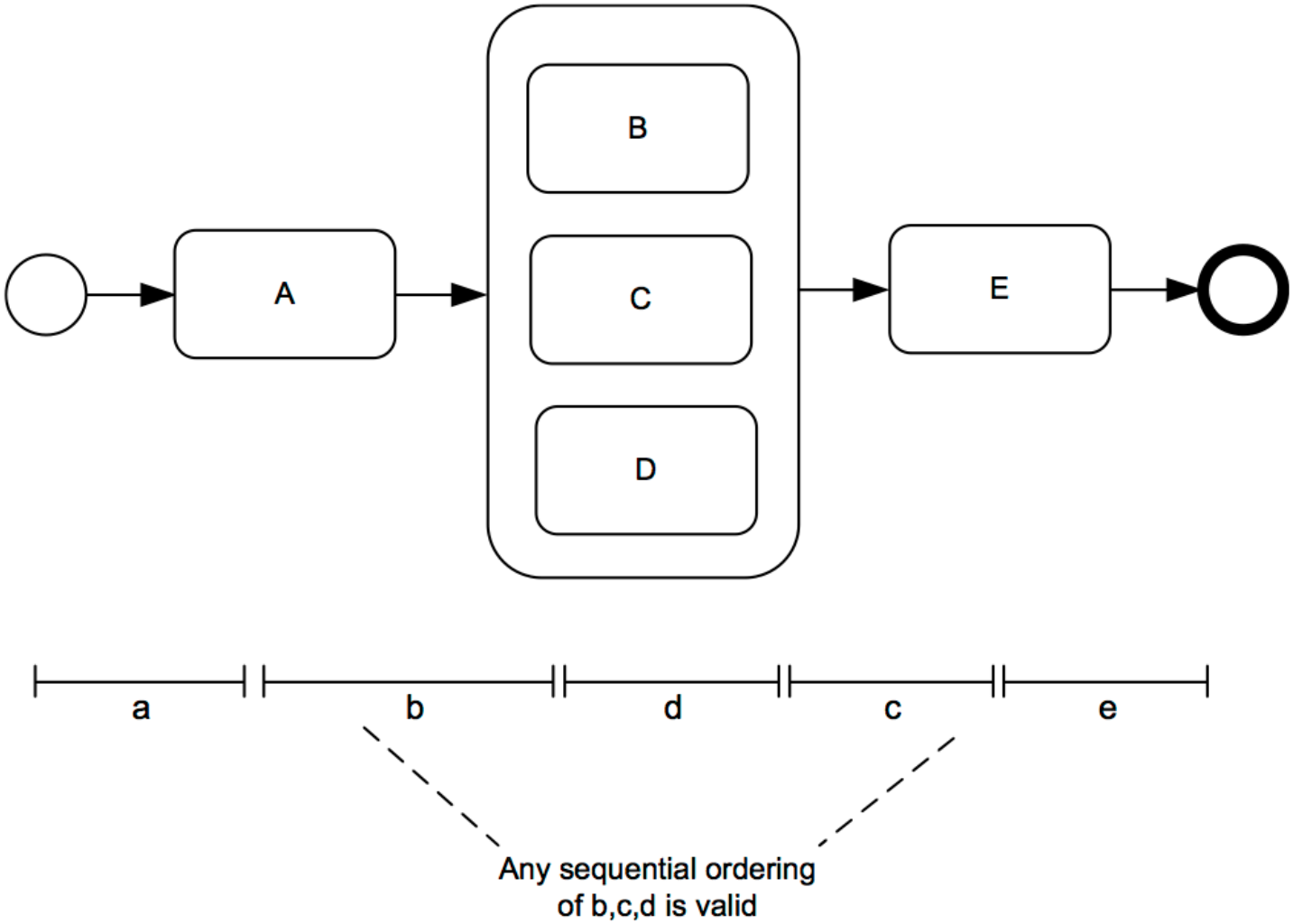


M. Weske: Business Process Management,
© Springer-Verlag Berlin Heidelberg 2012, 2007

Multiple instance without a priori run time knowledge pattern, including management task



Sequential execution without a priori design time knowlede

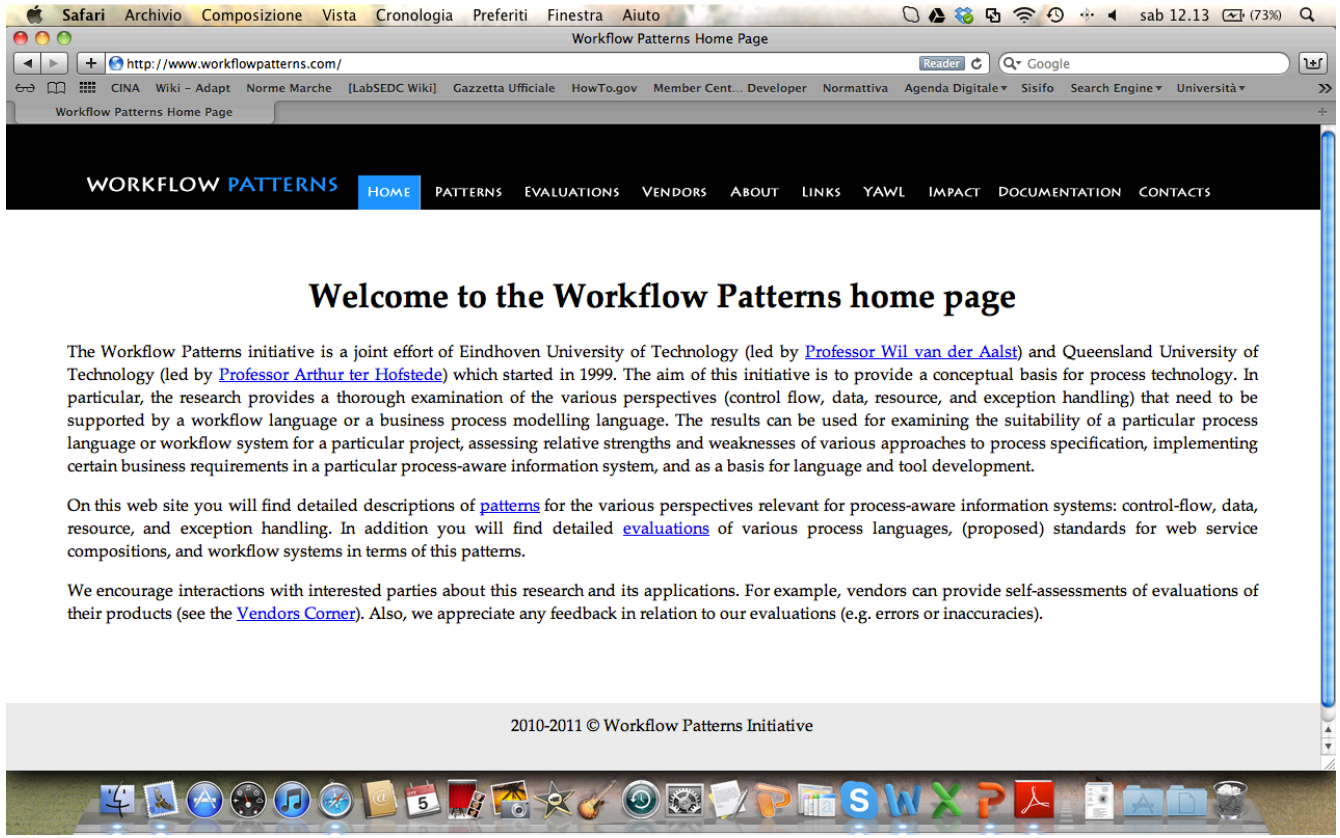


M. Weske: Business Process Management,
© Springer-Verlag Berlin Heidelberg 2012, 2007

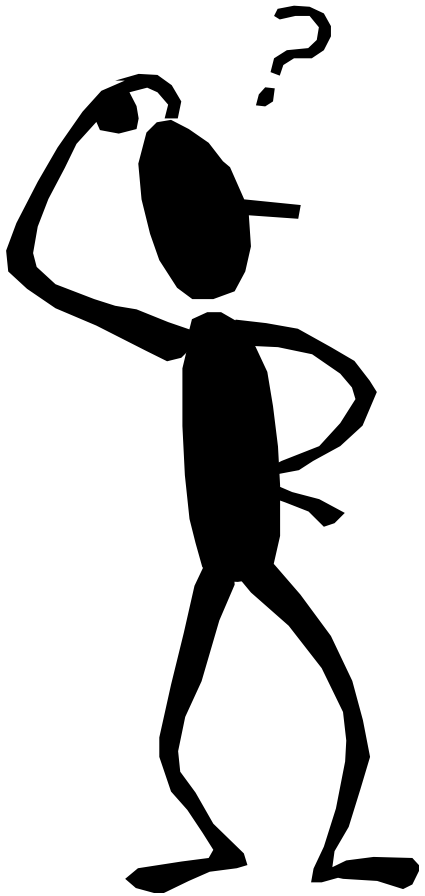
Other types of patterns

- Resource
- Data
- Exception Handling
- Presentation
- …

More on pattern



http://www.workflowpatterns.com/



Questions?