Reactive and Event Based Systems
Lecture 2: Process Modeling and DCR Graphs

Tijs Slaats

Monday 29th of November 2021

Overview

- **Process Modelling**
- **Imperative vs Declarative Process Models**
- **Dynamic Condition Response (DCR) Graphs**
- Hierarchy in DCR GRaphs
- Semantics of DCR Graphs
- Assignment 1

Processes

Process: “A series of actions or steps taken in order to achieve a particular end.”[1]

Examples:

- Production of car
- Handling of an insurance claim
- Treatment for lung cancer
- Software development
- Algorithms

[1] Oxford’s free English dictionary

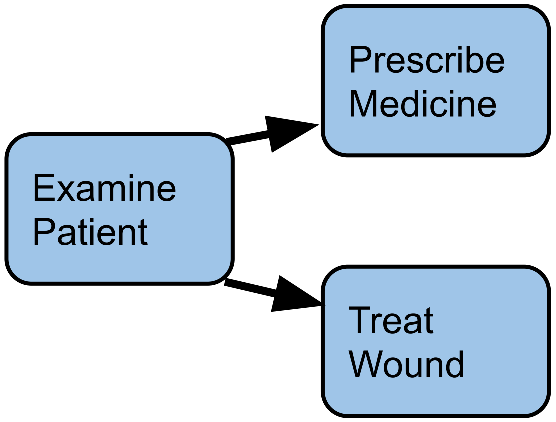
Process Modelling

How do we model a process?

- Plain text:

"Attach wheels and engine to frame."

- Drawings:



Process Modelling

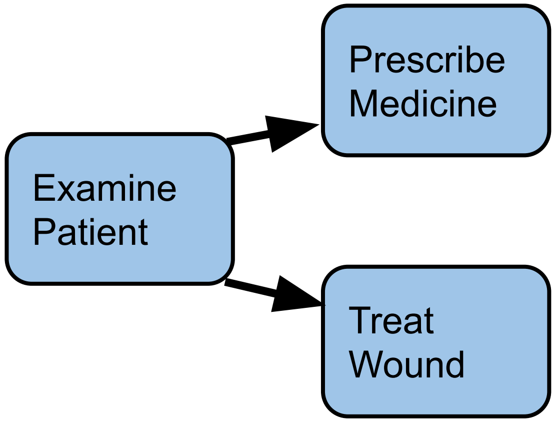
How do we model a process?

- Plain text:

"Attach wheels and engine to frame."

Allowed to happen at the same time?

- Drawings:

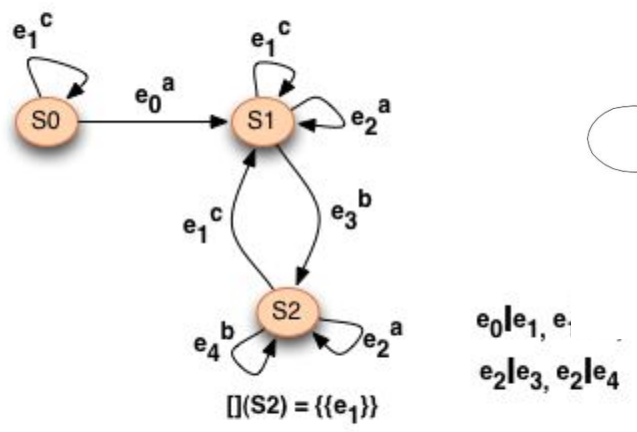


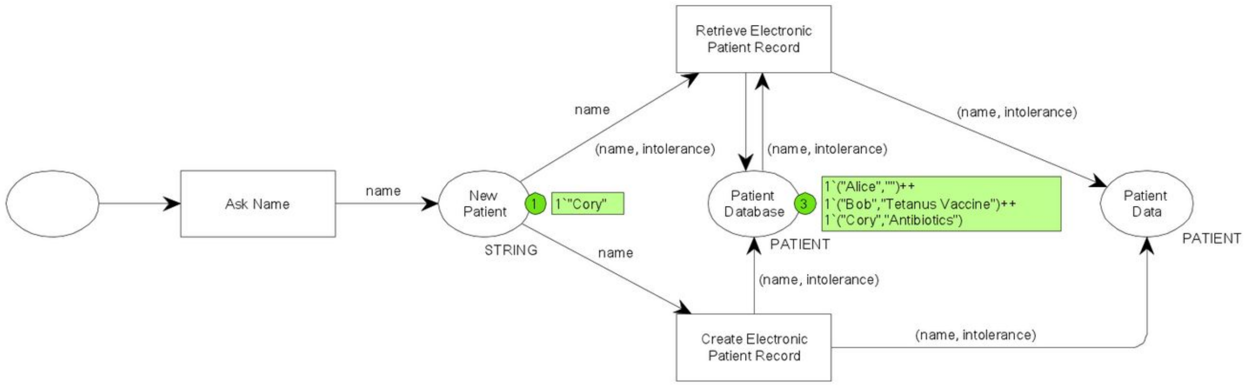
Do we do both or choose one?

Ambiguity!

Process Modelling and Computer Science

We need a well-defined language for our models: **Formal Methods**





$\square$(Recieve Claim $\Rightarrow$ $\Diamond$Evaluate Claim)
$\square$(Approve Claim $\Rightarrow$ $\Diamond$Payout Claim)
$\square$($\neg$Payout Claim W Approve Claim)

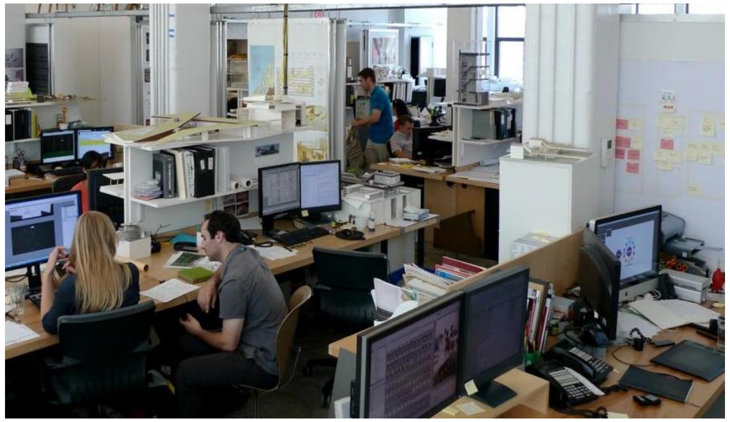
Process Modelling and Computer Science

Formal models offer:

- Unambiguous semantics
- Verification
- Model checking
- Conformance checking
- Simulation
- Execution
  - Automated (fx assembly lines)
  - User guidance (fx call centers)

Imperative vs Declarative Process Modelling

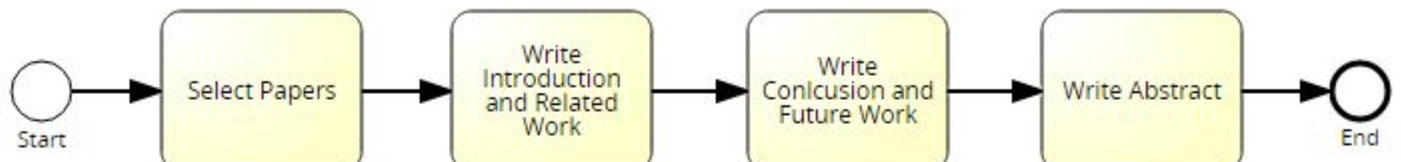
Imperative notations:	**Declarative notations:**
Used for *structured* processes	Used for *flexible* processes
Describes *flow*	Describes *constraints*
Nothing allowed by default	*Everything* allowed by default
Describe *desired* behaviour	Describe *forbidden* behaviour



Imperative Process Modelling

Business Process Model Notation (BPMN):

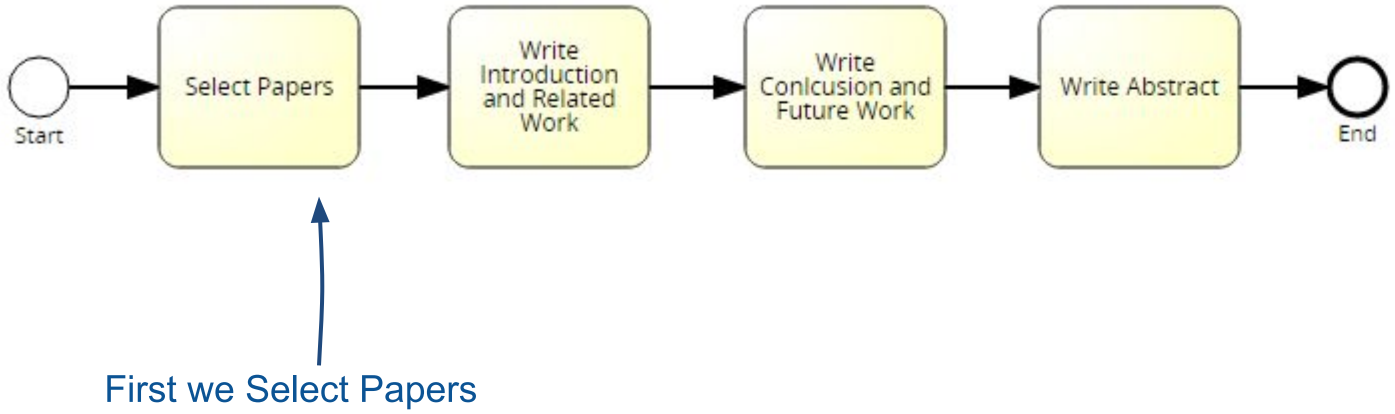
- Standard notation used for business processes
- Is flow-based:


Start
Select Papers
Write Introduction and Related Work
Write Conlcusion and Future Work
Write Abstract
End

Imperative Process Modelling

Business Process Model Notation (BPMN):

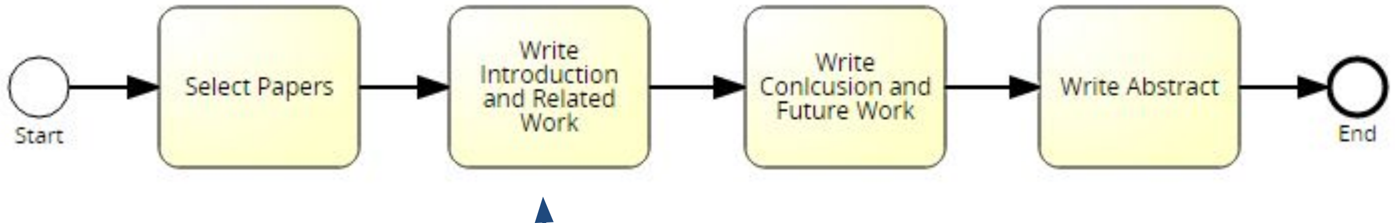
- Standard notation used for business processes
- Is flow-based:


Start
Select Papers
Write Introduction and Related Work
Write Conlcusion and Future Work
Write Abstract
End
First we Select Papers

Imperative Process Modelling

Business Process Model Notation (BPMN):

- Standard notation used for business processes
- Is flow-based:

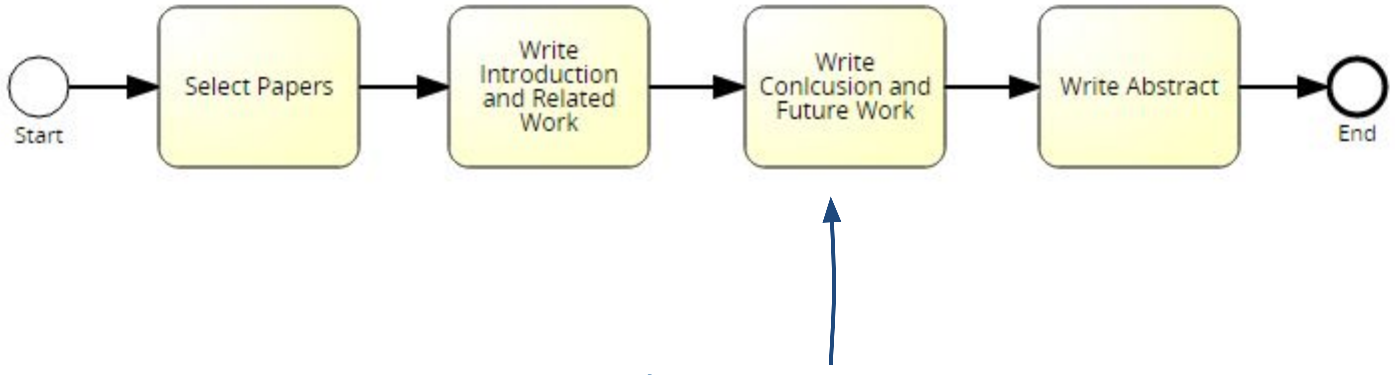

Start
Select Papers
Write Introduction and Related Work
Write Conlcusion and Future Work
Write Abstract
End

Afterwards we write an introduction and related work section

Imperative Process Modelling

Business Process Model Notation (BPMN):

- Standard notation used for business processes
- Is flow-based:

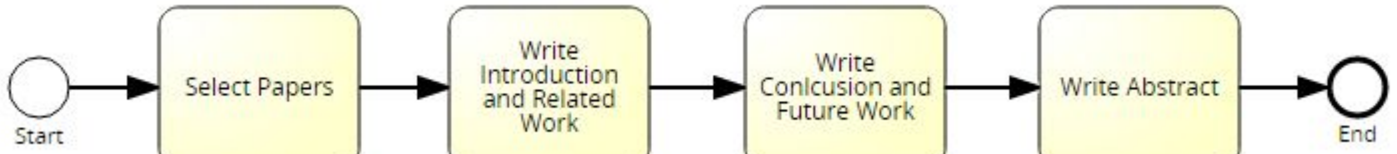

Start
Select Papers
Write Introduction and Related Work
Write Conlcusion and Future Work
Write Abstract
End

Afterwards we write a conclusion and future work section

Imperative Process Modelling

Business Process Model Notation (BPMN):

- Standard notation used for business processes
- Is flow-based:

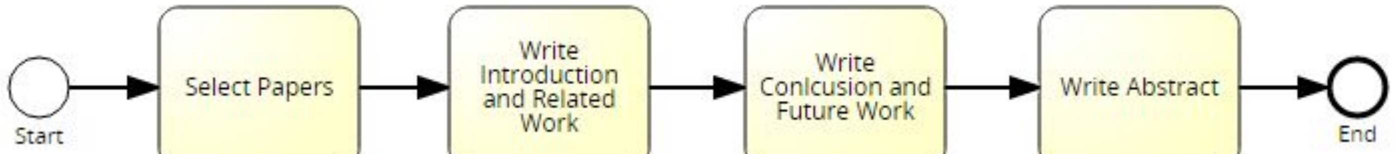

Start
Select Papers
Write Introduction and Related Work
Write Conlcusion and Future Work
Write Abstract
End

Finally we write the abstract

Imperative Process Modelling

Business Process Model Notation (BPMN):

- Standard notation used for business processes
- Is flow-based:

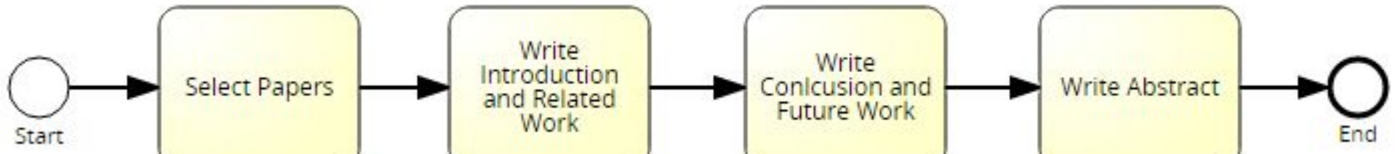

Start
Select Papers
Write Introduction and Related Work
Write Conlcusion and Future Work
Write Abstract
End

Allowed traces of the model:
<Select Papers, Write introduction, Write Conclusion, Write Abstract>

Imperative Process Modelling

Business Process Model Notation (BPMN):

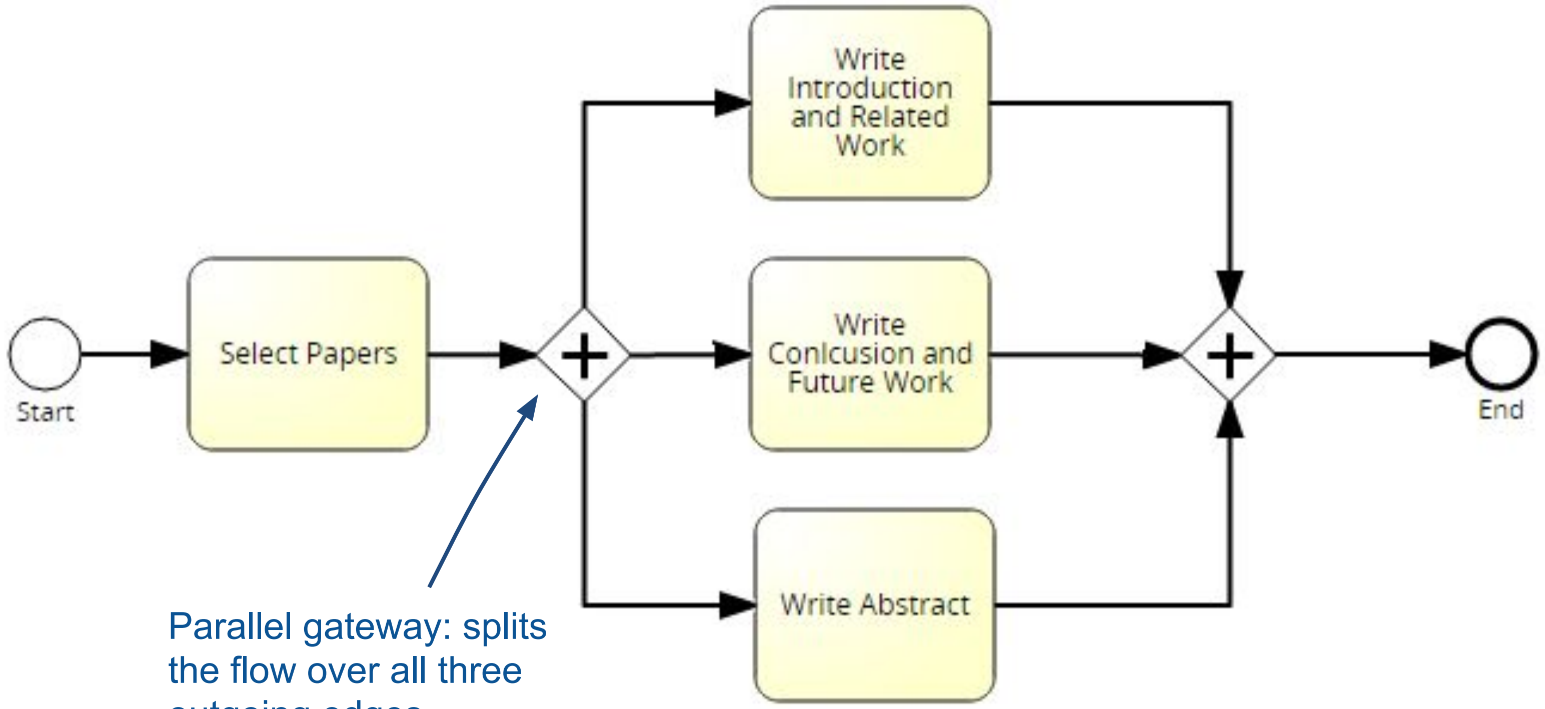
- Standard notation used for business processes
- Is flow-based:


Start
Select Papers
Write Introduction and Related Work
Write Conlcusion and Future Work
Write Abstract
End

Nice for straightforward, strict processes, but what if we want something more flexible?

Imperative Process Modelling

What if we want to be able of choosing the order of activities?

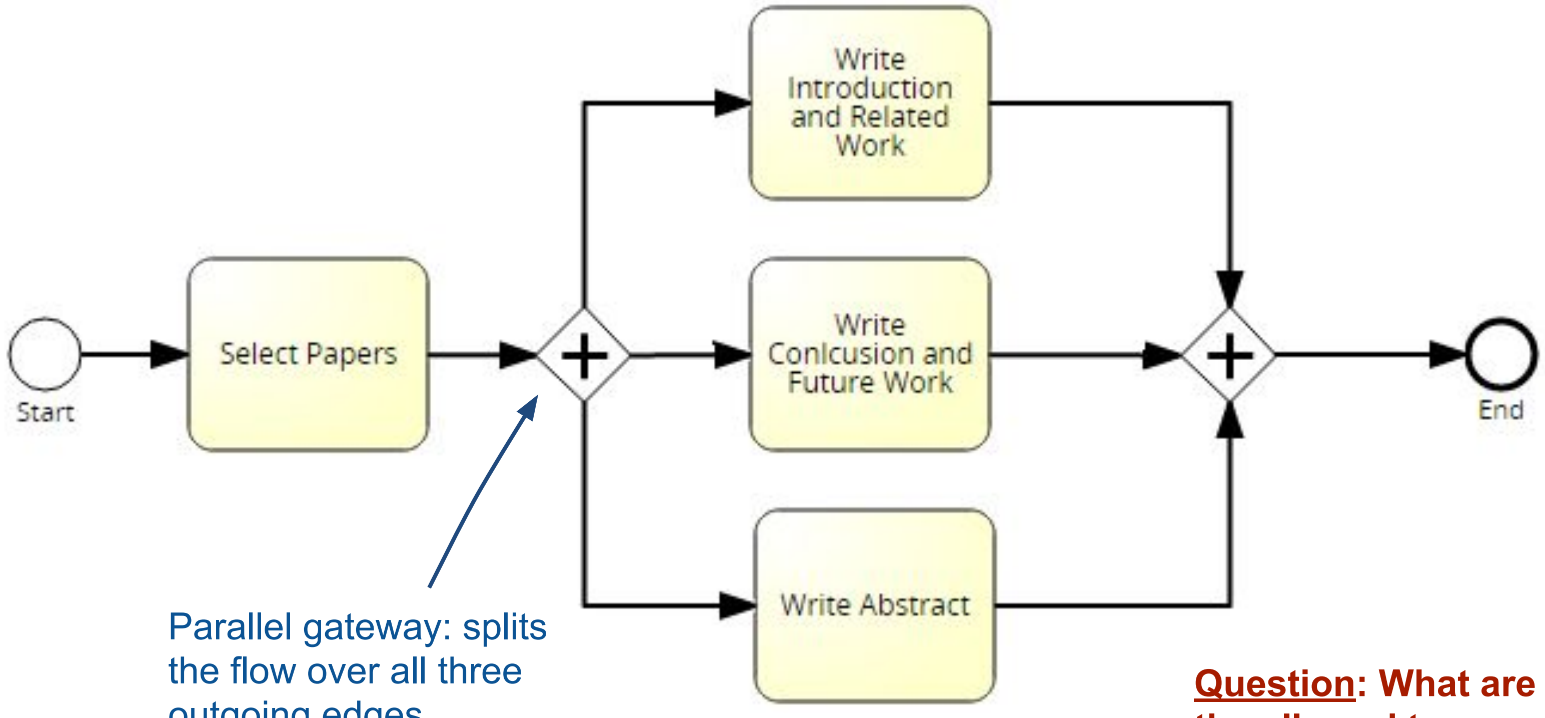


Start
Select Papers
Write Introduction and Related Work
Write Conlcusion and Future Work
Write Abstract
End

Parallel gateway: splits the flow over all three outgoing edges

Imperative Process Modelling

What if we want to be able of choosing the order of activities?

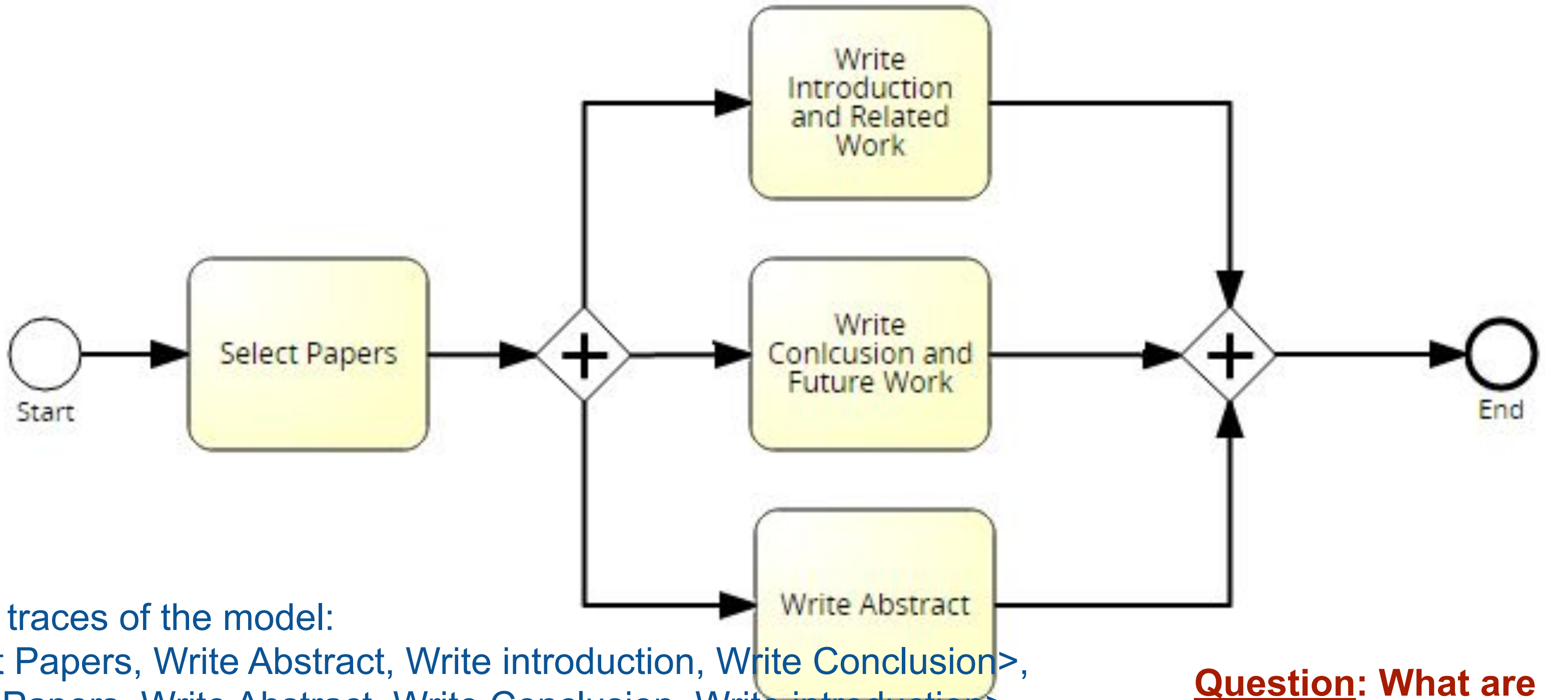

Start
Select Papers
Write Introduction and Related Work
Write Conlcusion and Future Work
Write Abstract
End

Parallel gateway: splits the flow over all three outgoing edges

<u>Question</u>: What are the allowed traces of this model?

Imperative Process Modelling

What if we want to be able of choosing the order of activities?

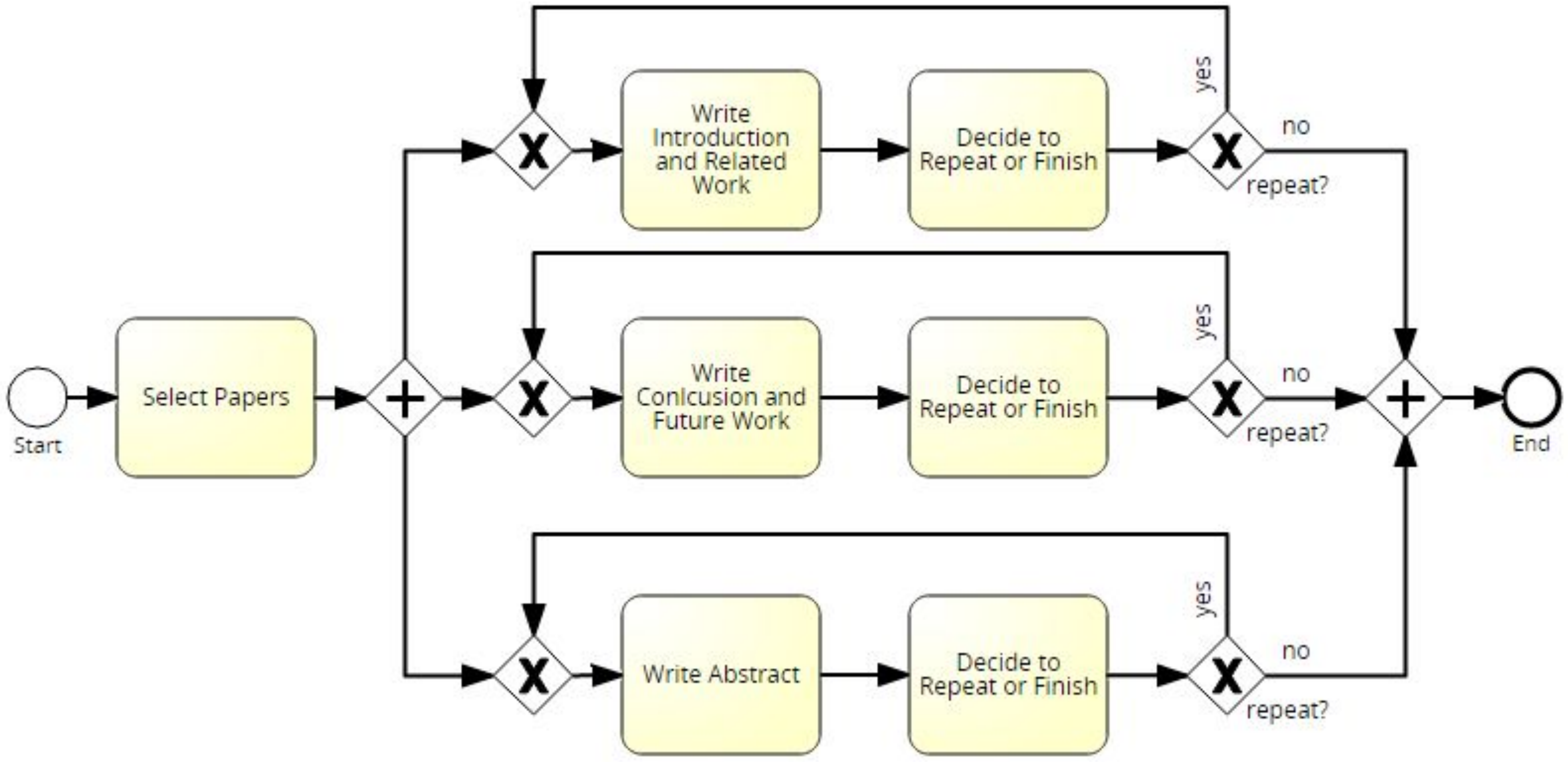

Start
Select Papers
Write Introduction and Related Work
Write Conlcusion and Future Work
Write Abstract
End

Allowed traces of the model:
{<Select Papers, Write Abstract, Write introduction, Write Conclusion>,
<Select Papers, Write Abstract, Write Conclusion, Write introduction>,
<Select Papers, Write introduction, Write Conclusion, Write Abstract>,
<Select Papers, Write introduction, Write Abstract, Write Conclusion>,
<Select Papers, Write Conclusion, Write introduction, Write Abstract>,
<Select Papers, Write Conclusion, Write Abstract, Write introduction>}

Question: What are the allowed traces of this model?

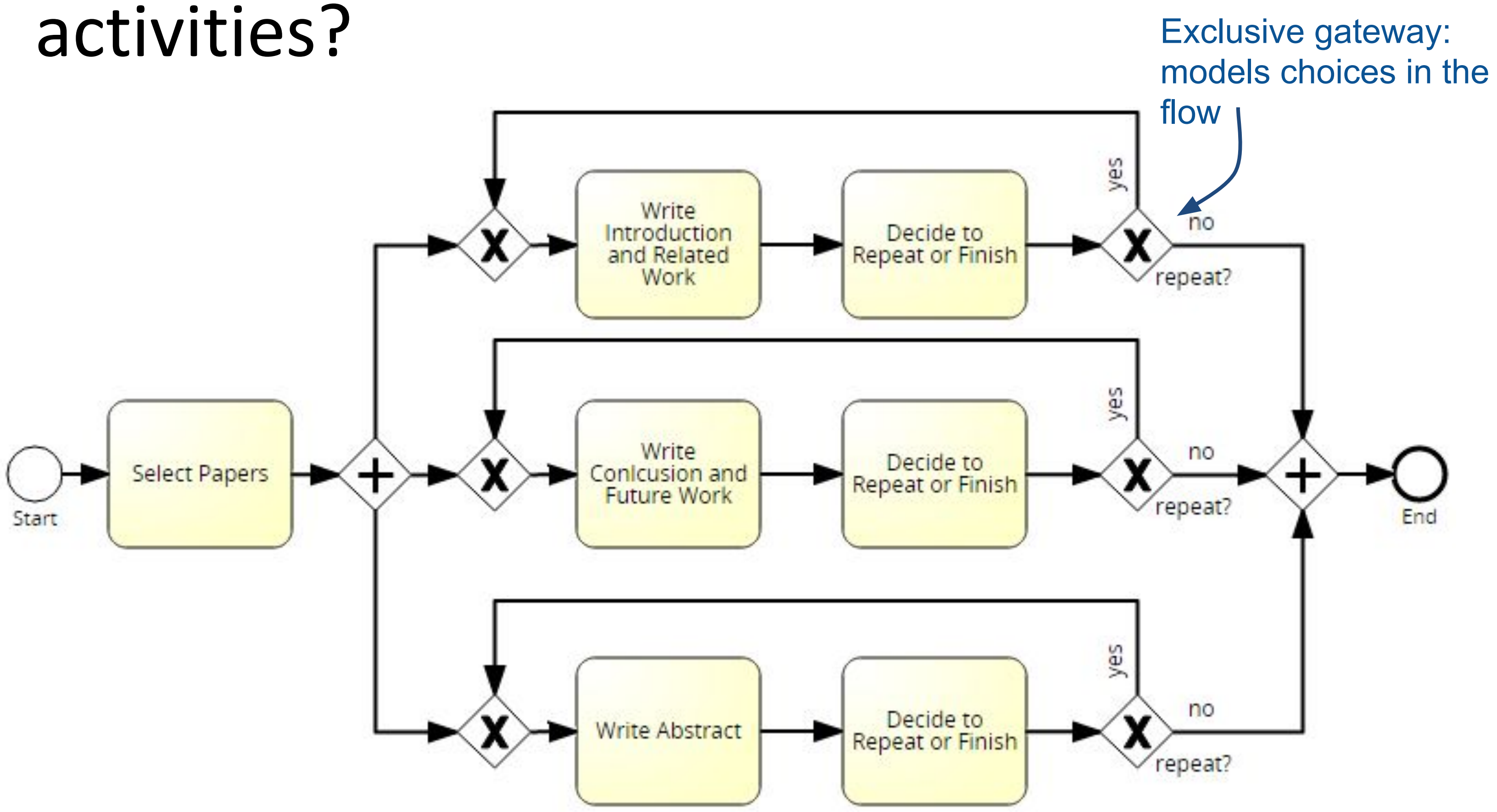
Imperative Process Modelling

What if we want to be able of repeating activities?


Start
Select Papers
Write Introduction and Related Work
Decide to Repeat or Finish
yes
no
repeat?
Write Conlcusion and Future Work
Decide to Repeat or Finish
yes
no
repeat?
Write Abstract
Decide to Repeat or Finish
yes
no
repeat?
End

Imperative Process Modelling

What if we want to be able of repeating activities?



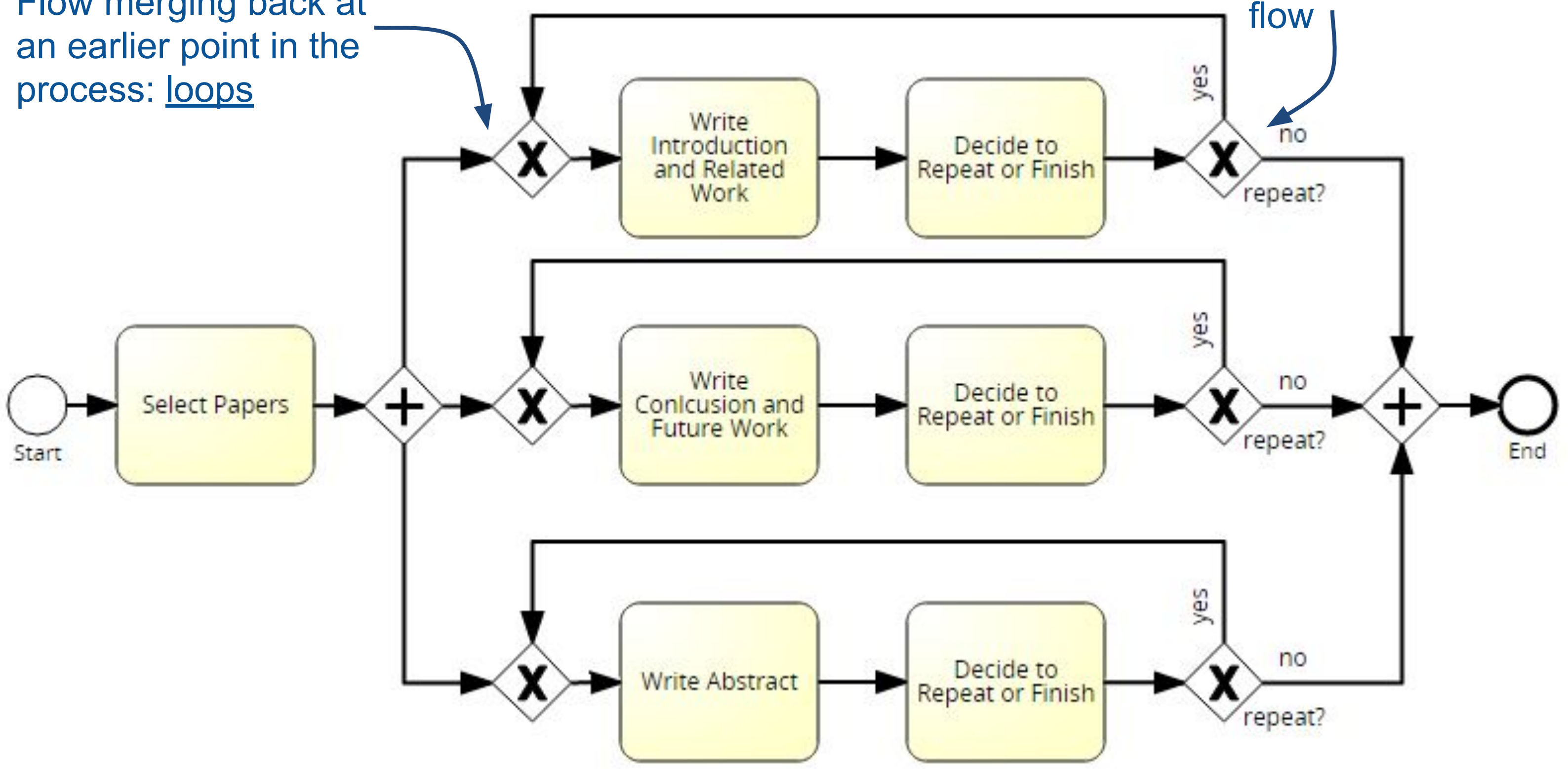
Exclusive gateway: models choices in the flow
Start
Select Papers
Write Introduction and Related Work
Decide to Repeat or Finish
yes
no
repeat?
Write Conlcusion and Future Work
Decide to Repeat or Finish
yes
no
repeat?
Write Abstract
Decide to Repeat or Finish
yes
no
repeat?
End

Imperative Process Modelling

What if we want to be able of repeating activities?

Flow merging back at an earlier point in the process: loops

Exclusive gateway: models choices in the flow

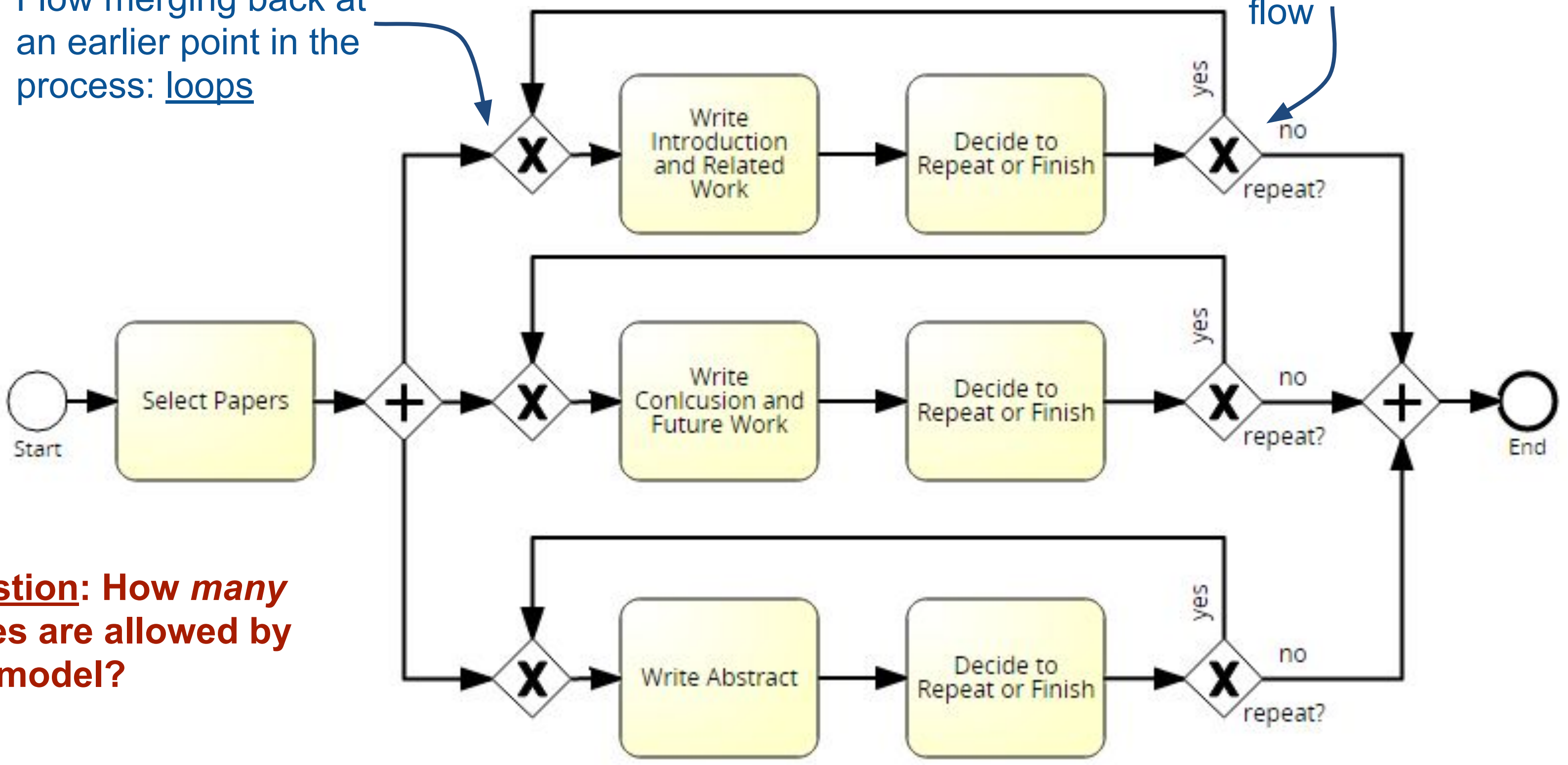

Start
Select Papers
Write Introduction and Related Work
Decide to Repeat or Finish
yes
no
repeat?
Write Conlcusion and Future Work
Decide to Repeat or Finish
yes
no
repeat?
Write Abstract
Decide to Repeat or Finish
yes
no
repeat?
End

Imperative Process Modelling

What if we want to be able of repeating activities?

Flow merging back at an earlier point in the process: loops

Exclusive gateway: models choices in the flow


Start
Select Papers
Write Introduction and Related Work
Decide to Repeat or Finish
yes
no
repeat?
Write Conlcusion and Future Work
Decide to Repeat or Finish
yes
no
repeat?
Write Abstract
Decide to Repeat or Finish
yes
no
repeat?
End

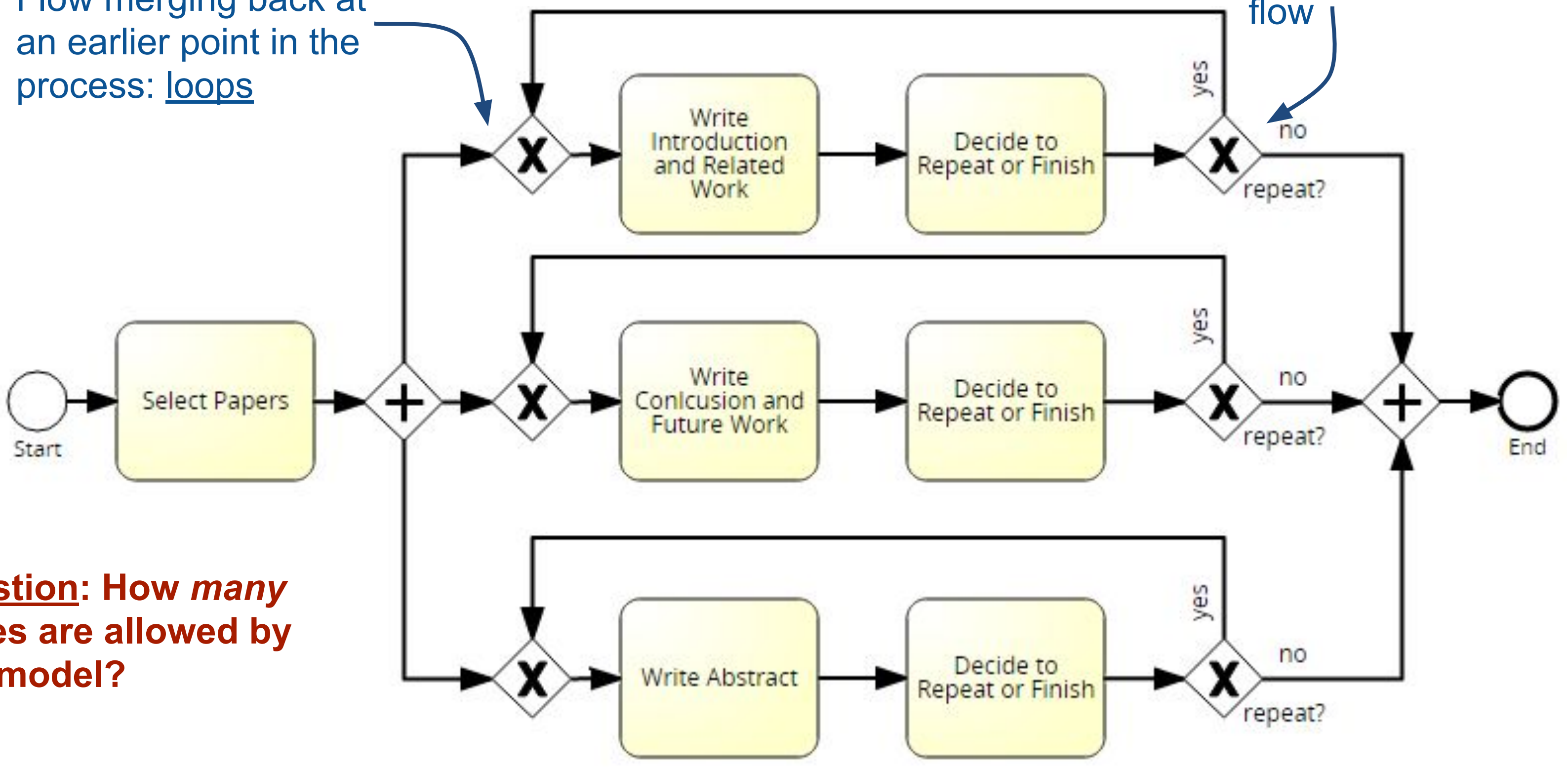
Question: How *many* traces are allowed by this model?

Imperative Process Modelling

What if we want to be able of repeating activities?

Flow merging back at an earlier point in the process: loops

Exclusive gateway: models choices in the flow

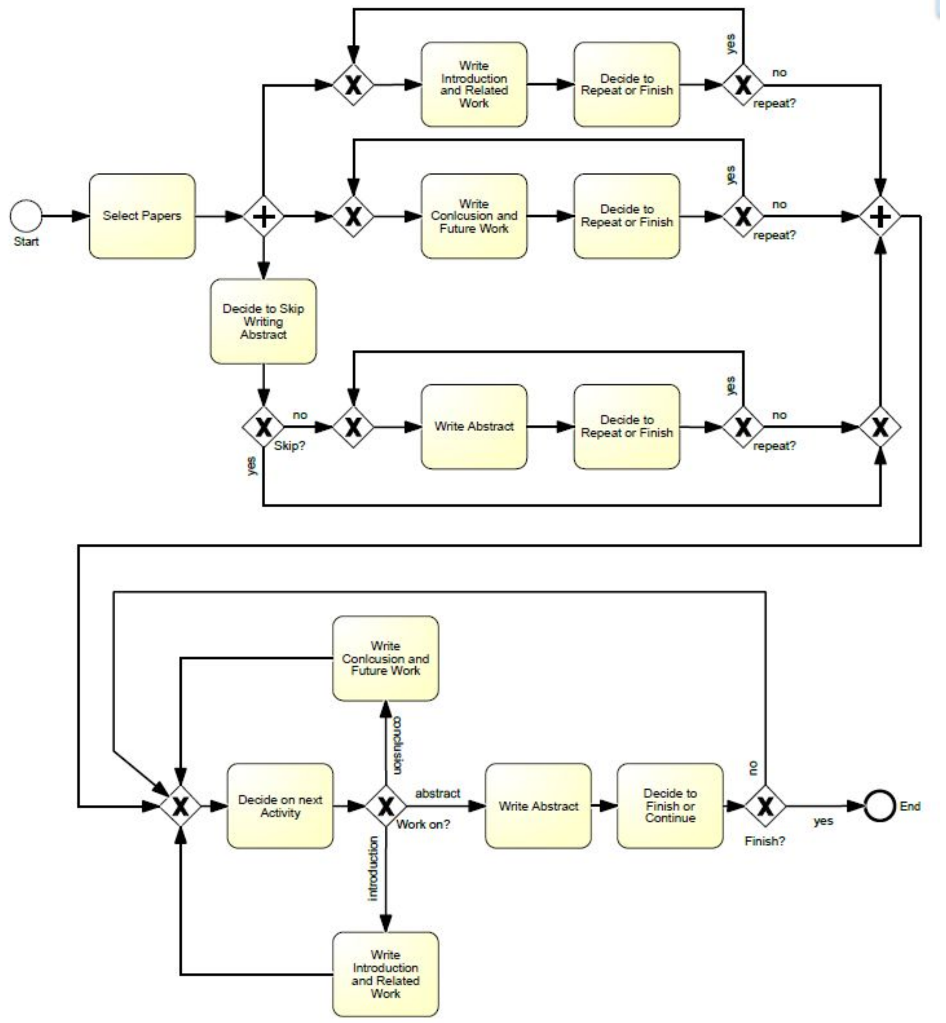

Start
Select Papers
Write Introduction and Related Work
Decide to Repeat or Finish
yes
no
repeat?
Write Conlcusion and Future Work
Decide to Repeat or Finish
yes
no
repeat?
Write Abstract
Decide to Repeat or Finish
yes
no
repeat?
End

Question: How *many* traces are allowed by this model?

Allowed traces of the model: infinitely many...

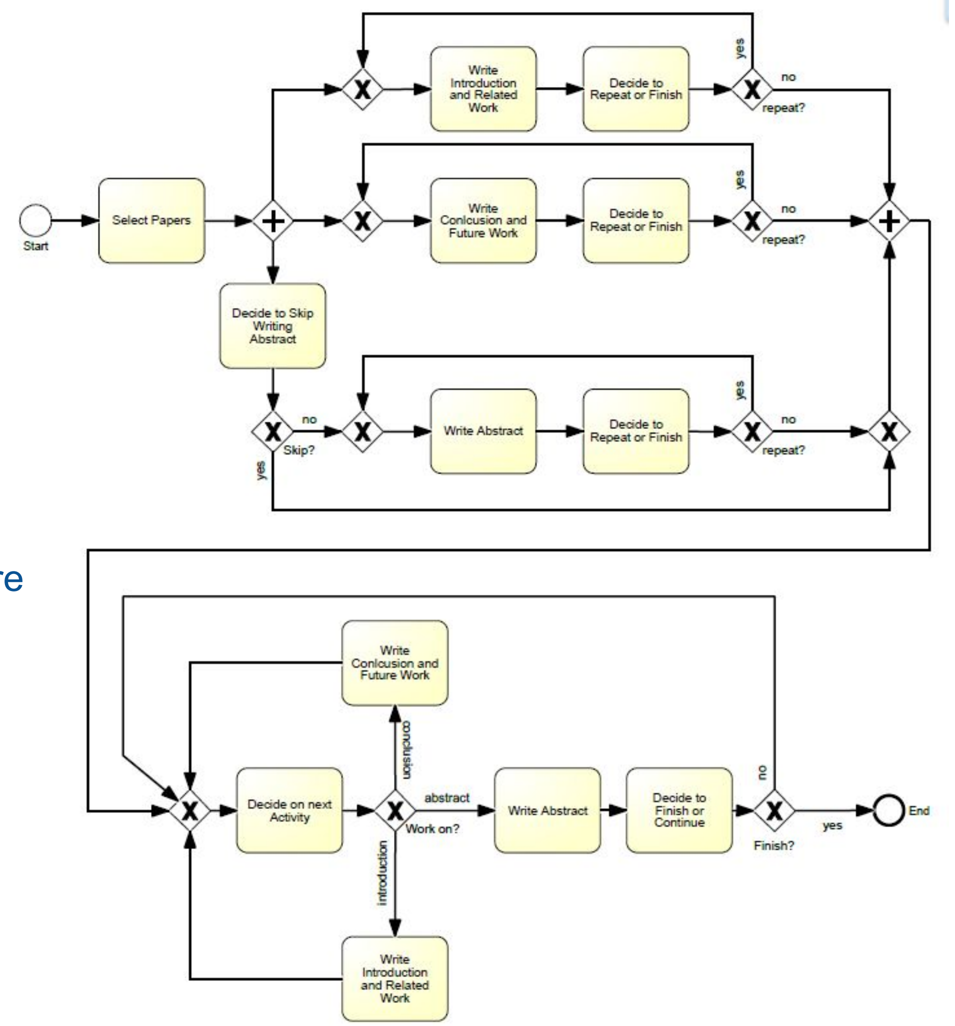
Imperative Process Modelling

What if we add a rule that we always need to update the abstract last?


Start
Select Papers
Write Introduction and Related Work
Decide to Repeat or Finish
yes
no
repeat?
Write Conlcusion and Future Work
Decide to Repeat or Finish
yes
no
repeat?
Decide to Skip Writing Abstract
no
Skip?
yes
Write Abstract
Decide to Repeat or Finish
yes
no
repeat?
Write Conlcusion and Future Work
conclusion
Decide on next Activity
abstract
Work on?
Write Abstract
Decide to Finish or Continue
no
yes
Finish?
End
introduction
Write Introduction and Related Work

Imperative Process Modelling

What if we add a rule that we always need to update the abstract last?

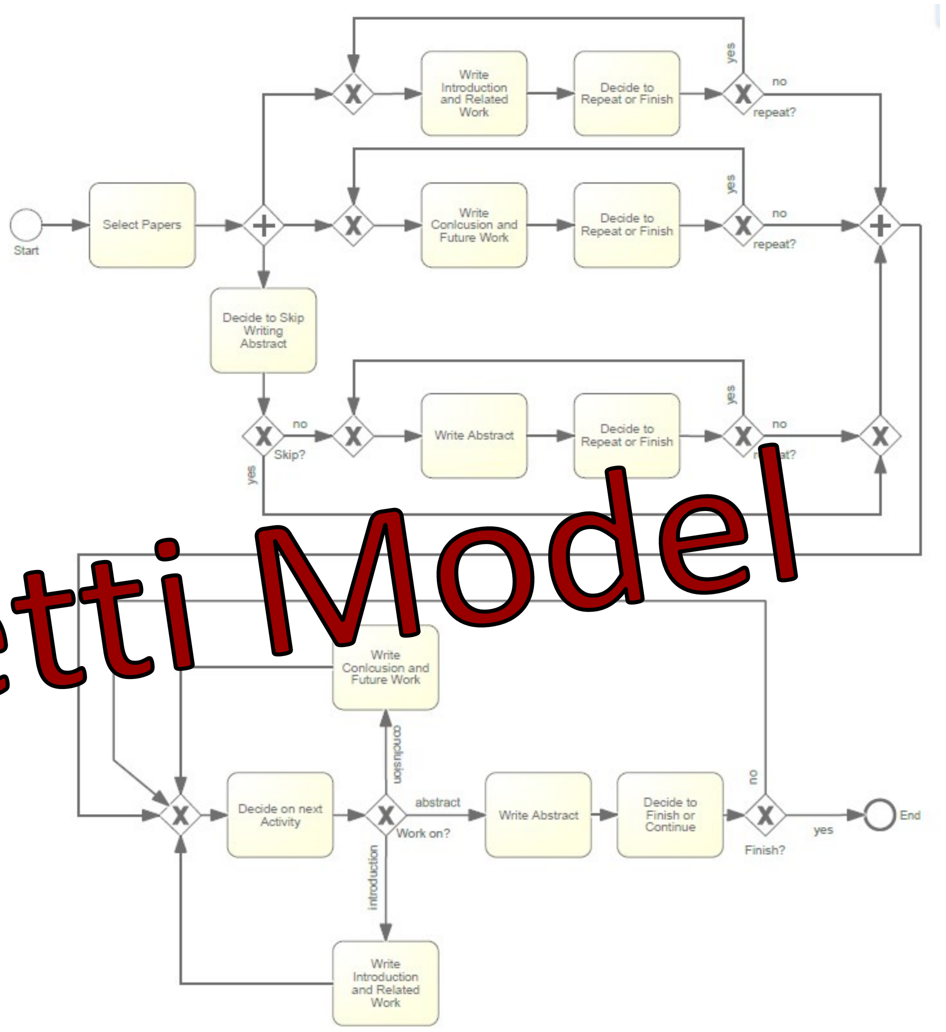


Start
Select Papers
Decide to Skip Writing Abstract
Skip?
no
yes
Write Introduction and Related Work
Write Conlcusion and Future Work
Write Abstract
Decide to Repeat or Finish
repeat?
yes
no
Decide on next Activity
Work on?
abstract
conclusion
introduction
Decide to Finish or Continue
Finish?
yes
no
End

Lots of repeated activities because there are different states to the process

Business Process Modelling

What if we add a rule that we always need to update the abstract last?


Start
Select Papers
Write Introduction and Related Work
Decide to Repeat or Finish
yes
no
repeat?
Write Conlcusion and Future Work
Decide to Repeat or Finish
yes
no
repeat?
Decide to Skip Writing Abstract
no
Skip?
yes
Write Abstract
Decide to Repeat or Finish
yes
no
Write Conlcusion and Future Work
conclusion
Decide on next Activity
abstract
Work on?
Write Abstract
Decide to Finish or Continue
no
yes
Finish?
End
introduction
Write Introduction and Related Work

Spaghetti Model

Knowledge Workers

- Knowledge Workers:
  - Solve diverse problems
  - Are experts at what they do
  - Require freedom to make their own decisions
- However, rules do exist:
  - Laws
  - Business practices

Declarative Process Notations

- Declarative Process Notations:
  - Better suited to modelling flexible processes
  - Focus on describing **constraints** instead of the **flow** of work
  - Tools can offer users all possible choices that follow the rules, while still advising on best-practice
  - Are more easily adapted to change (new laws, changing business practices)

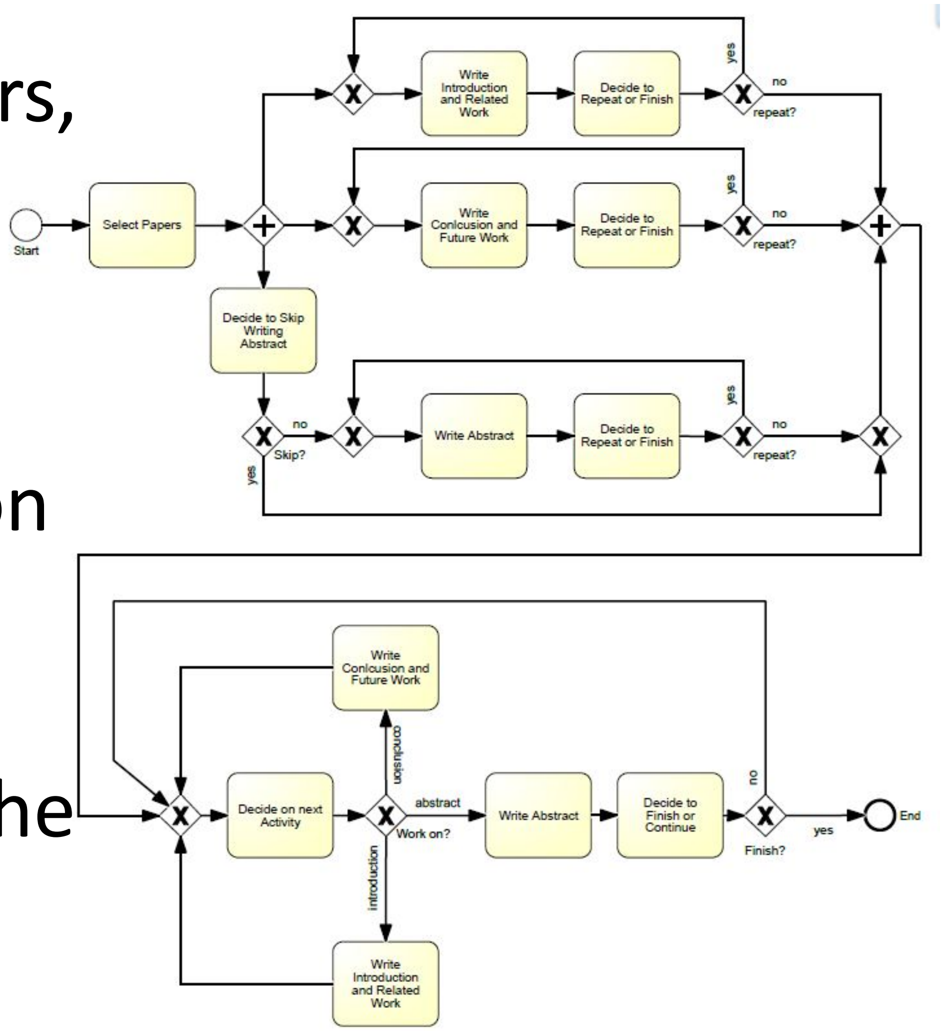
DCR Graphs

A declarative workflow notation, consisting of:

- *Events* (activities)
  - Unconstrained events can happen at any time and any number of times
- *Constraints* (rules) between events
- State represented as a *marking* consisting of *executed*, *pending* and *included* events

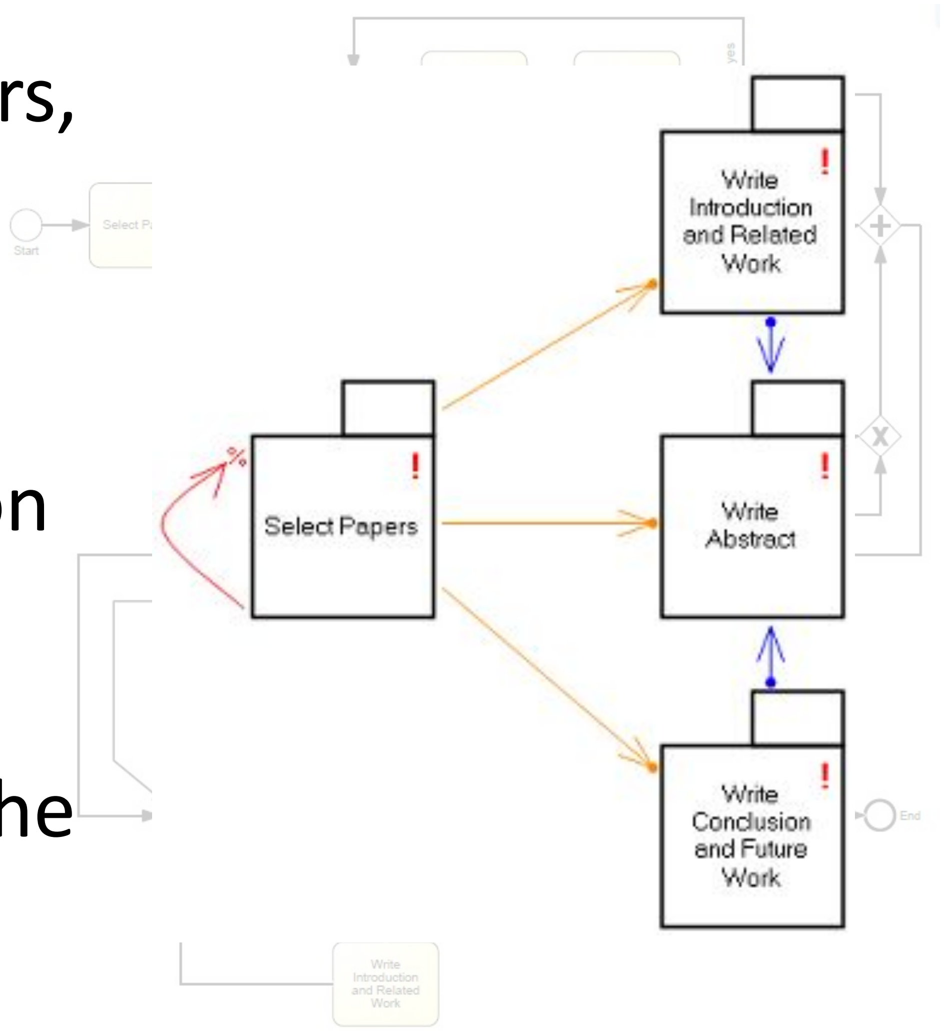
Example

- We first select papers, then:
- In any order, but at least once:
  - Write Introduction
  - Write Conclusion
  - Write Abstract
- We always update the abstract last


Start
Select Papers
Write Introduction and Related Work
Decide to Repeat or Finish
yes
no
repeat?
Write Conlcusion and Future Work
Decide to Skip Writing Abstract
Skip?
Write Abstract
Decide on next Activity
Work on?
conclusion
abstract
introduction
Decide to Finish or Continue
Finish?
End

Example

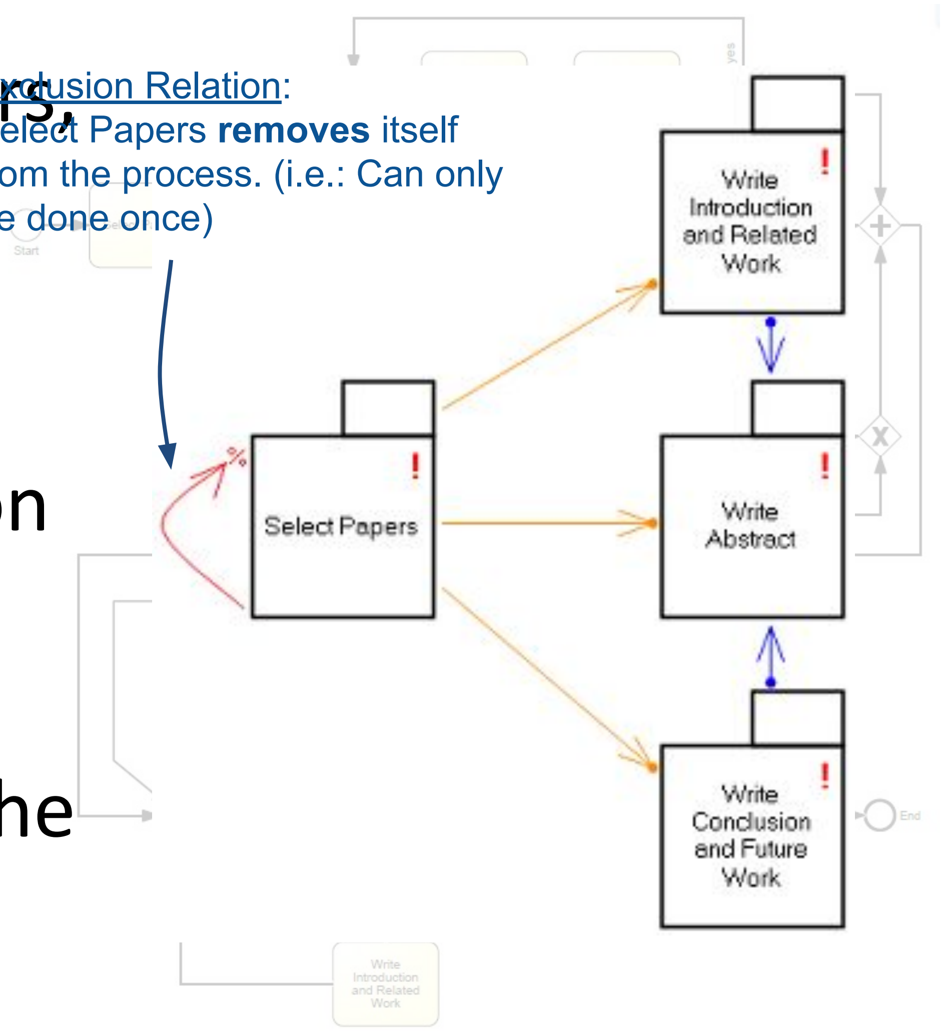
- We first select papers, then:
- In any order, but at least once:
  - Write Introduction
  - Write Conclusion
  - Write Abstract
- We always update the abstract last


Select Papers
Write Introduction and Related Work
Write Abstract
Write Conclusion and Future Work
Start
End
%
!

Example

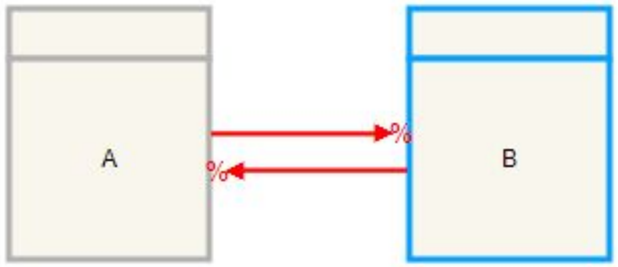
- We first select papers, then:
- In any order, but at least once:
  - Write Introduction
  - Write Conclusion
  - Write Abstract
- We always update the abstract last

Exclusion Relation:
Select Papers **removes** itself from the process. (i.e.: Can only be done once)



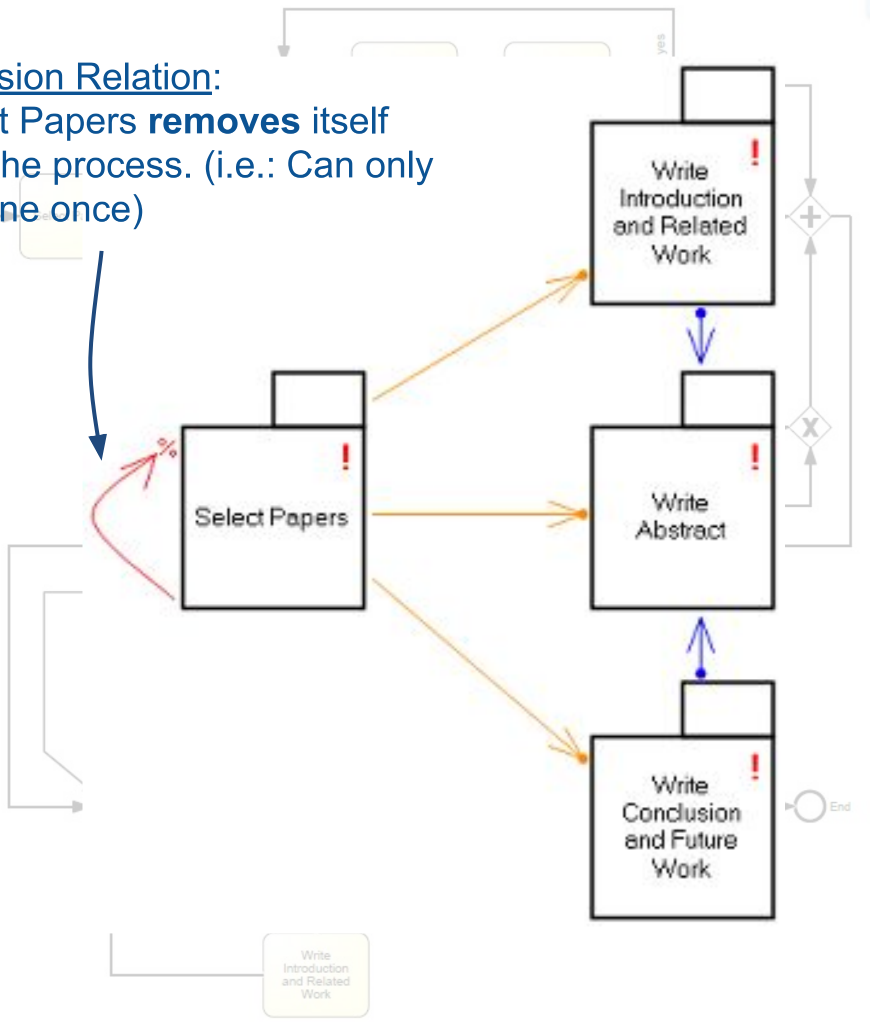
Select Papers
%
!
Write Introduction and Related Work
Write Abstract
Write Conclusion and Future Work

Example

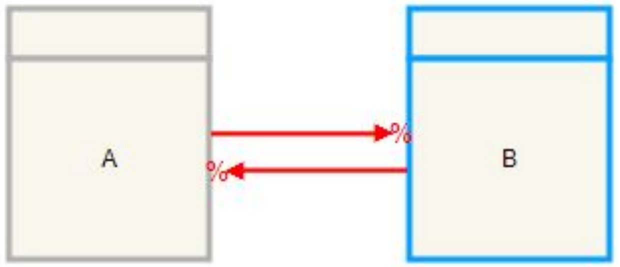


Question: What language does this model capture?

Exclusion Relation:
Select Papers **removes** itself from the process. (i.e.: Can only be done once)


Select Papers
Write Introduction and Related Work
Write Abstract
Write Conclusion and Future Work
%
!
!
!
!
End
Start

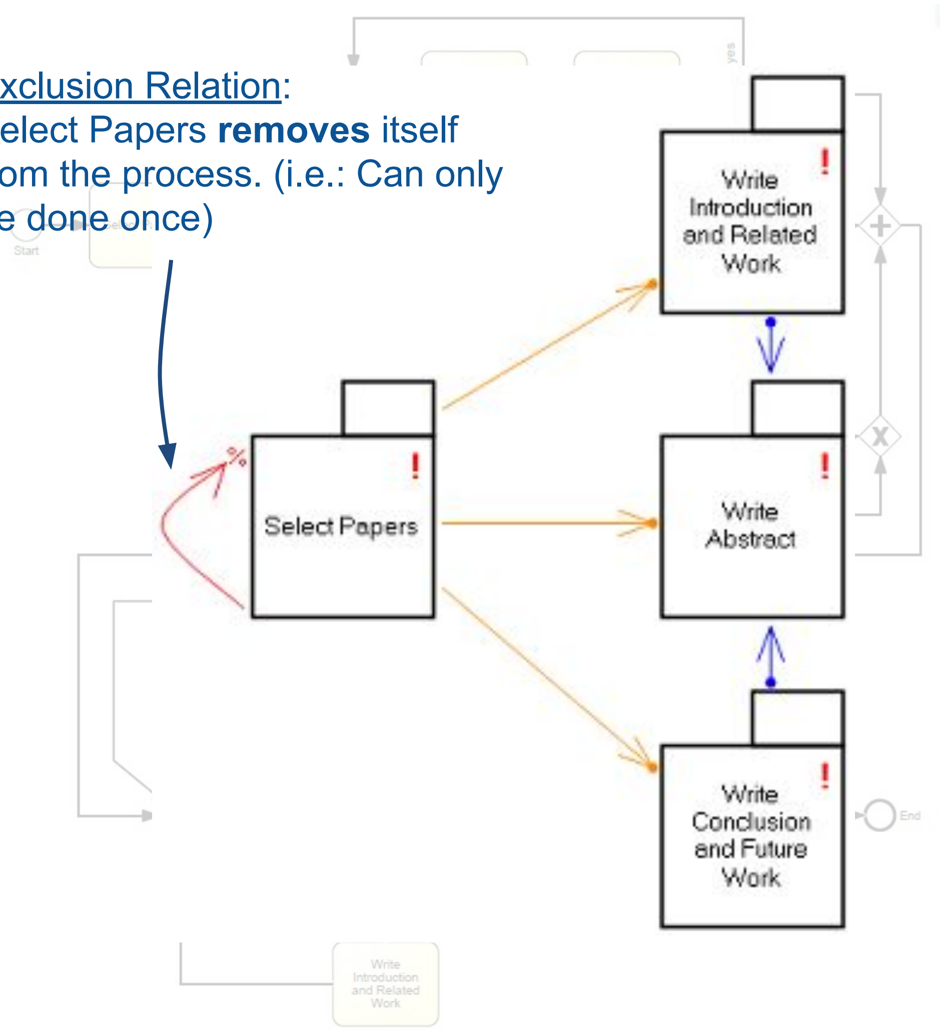
Example



<u>Question</u>: What language does this model capture?

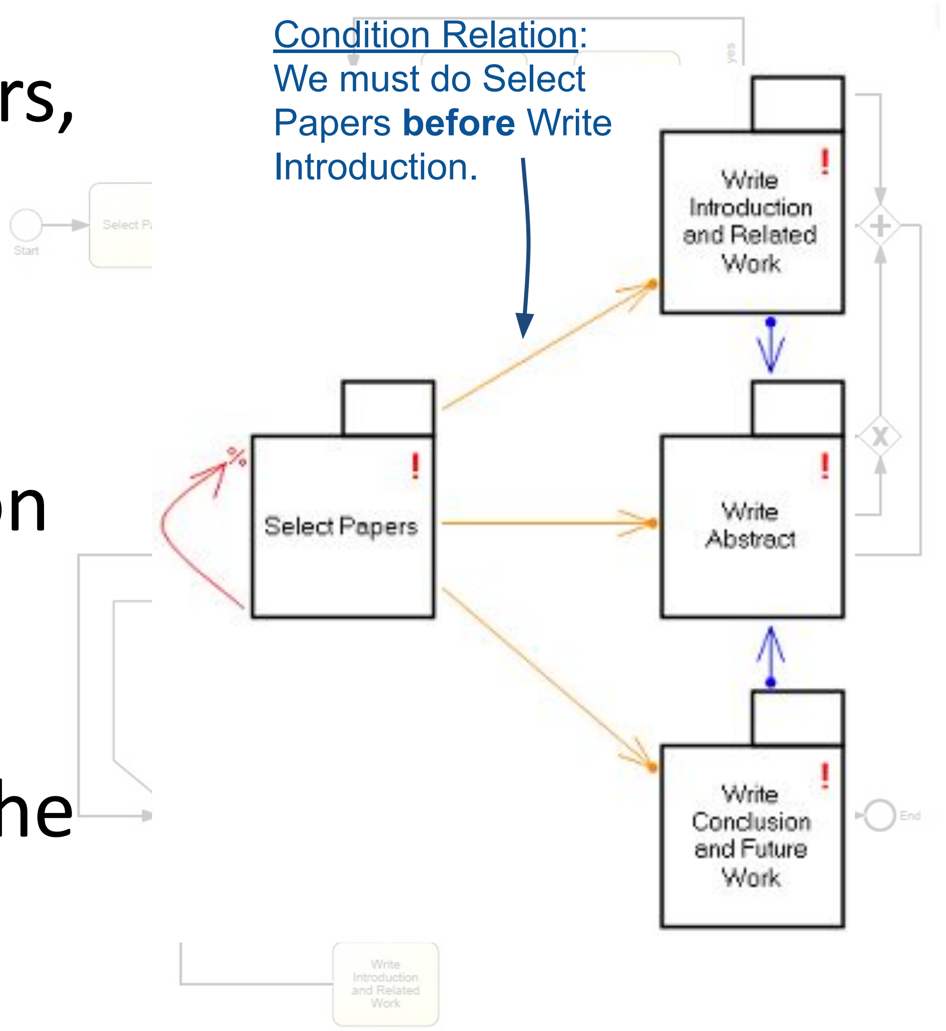
Can only do A or B

<u>Exclusion Relation</u>:
Select Papers **removes** itself from the process. (i.e.: Can only be done once)

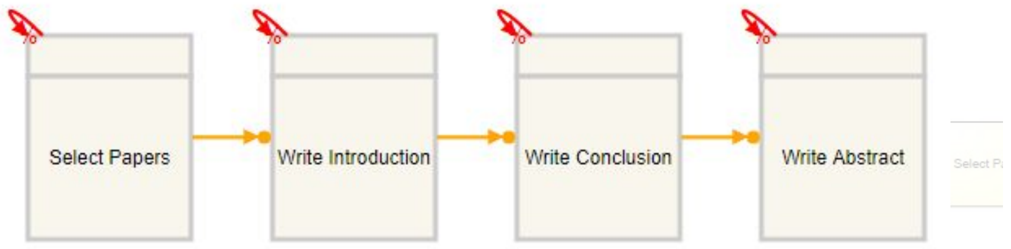

Select Papers
Write Introduction and Related Work
Write Abstract
Write Conclusion and Future Work
%
!
Start
End

Example

- We first select papers, then:
- In any order, but at least once:
  - Write Introduction
  - Write Conclusion
  - Write Abstract
- We always update the abstract last

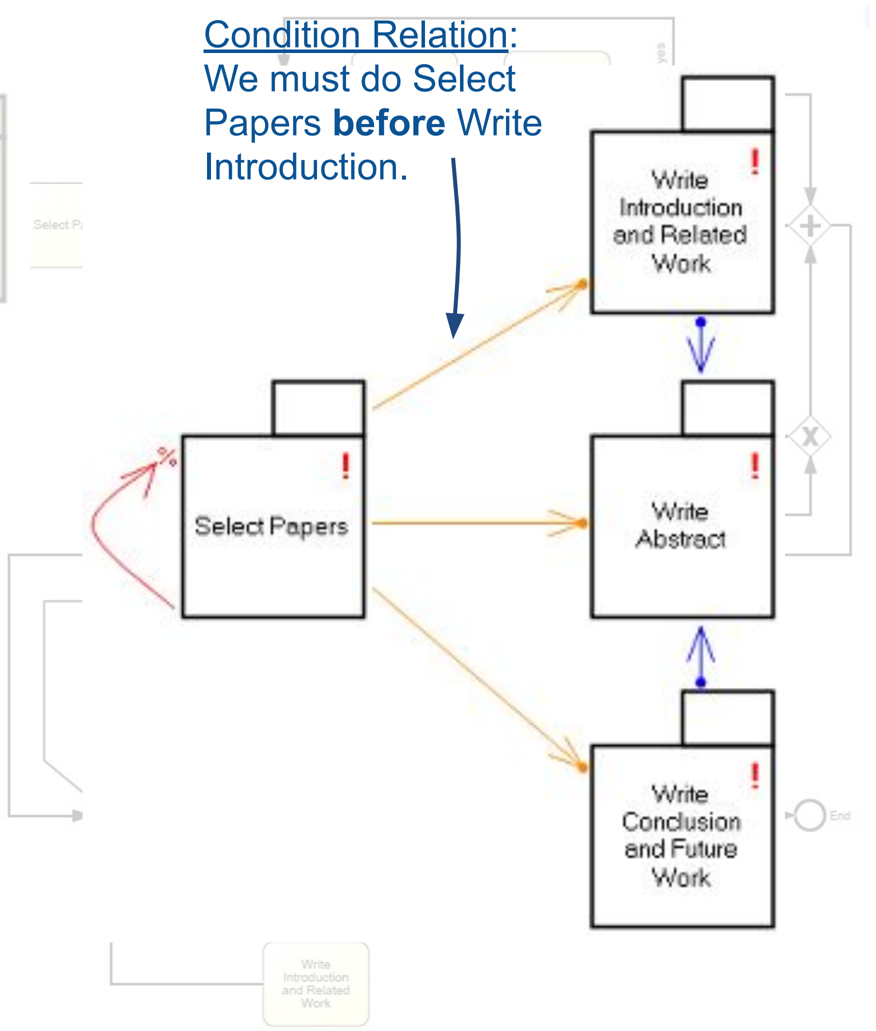

Condition Relation:
We must do Select Papers **before** Write Introduction.
Select Papers
Write Introduction and Related Work
Write Abstract
Write Conclusion and Future Work
%
!
Start
End

Example



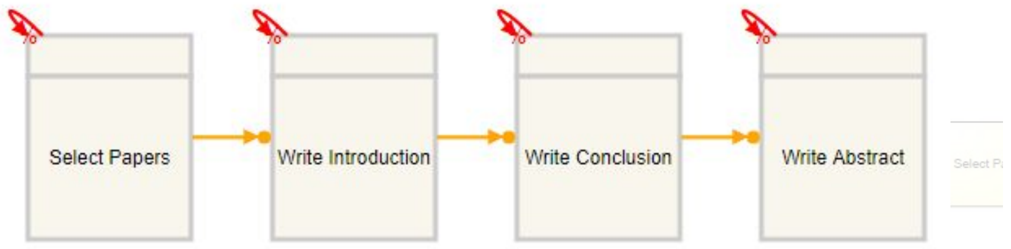
Question: What language does this model capture?

Condition Relation:
We must do Select Papers **before** Write Introduction.



Select Papers
Write Introduction and Related Work
Write Abstract
Write Conclusion and Future Work
End

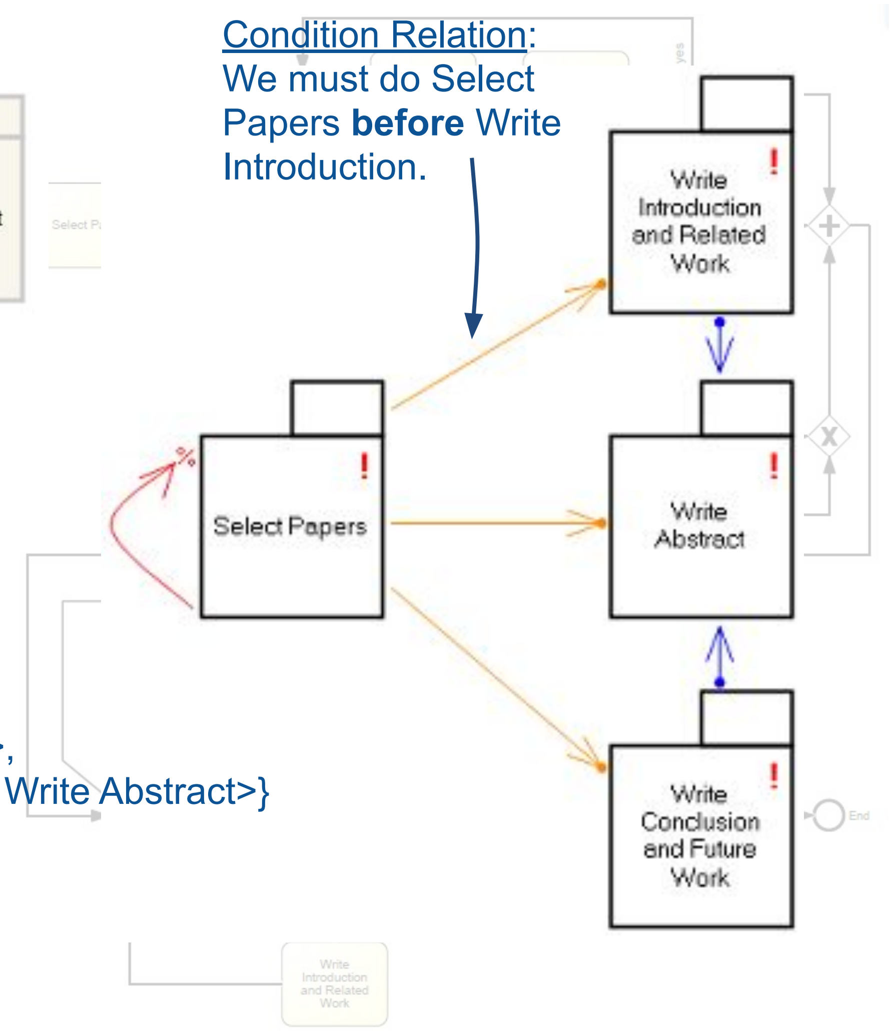
Example



Question: What language does this model capture?

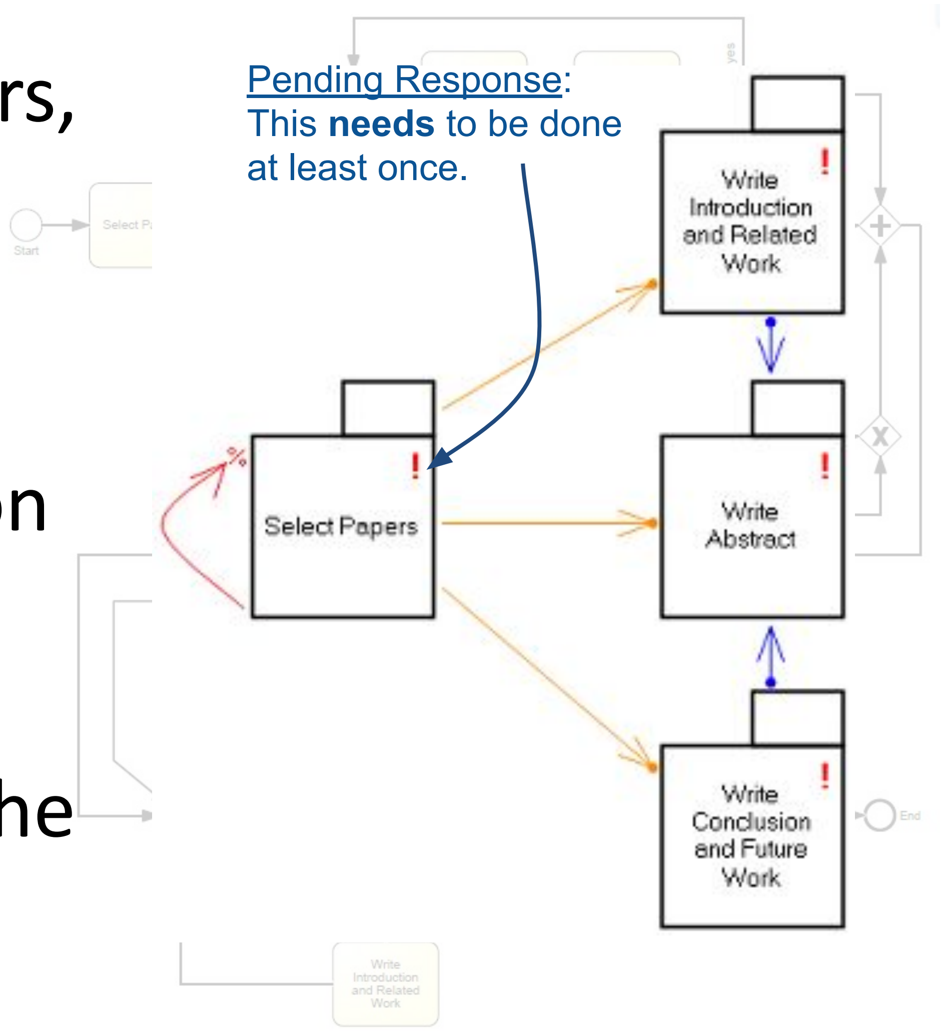
{<>,
<Select Papers>,
<Select Papers, Write Introduction>,
<Select Papers, Write Introduction, Write Conclusion>,
<Select Papers, Write Introduction, Write Conclusion, Write Abstract>}

Condition Relation:
We must do Select Papers **before** Write Introduction.


Write Introduction and Related Work
Select Papers
Write Abstract
Write Conclusion and Future Work
End

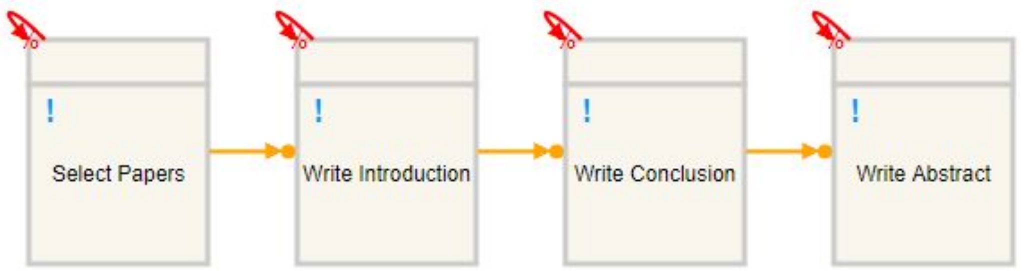
Example

- We first select papers, then:
- In any order, but at least once:
  - Write Introduction
  - Write Conclusion
  - Write Abstract
- We always update the abstract last

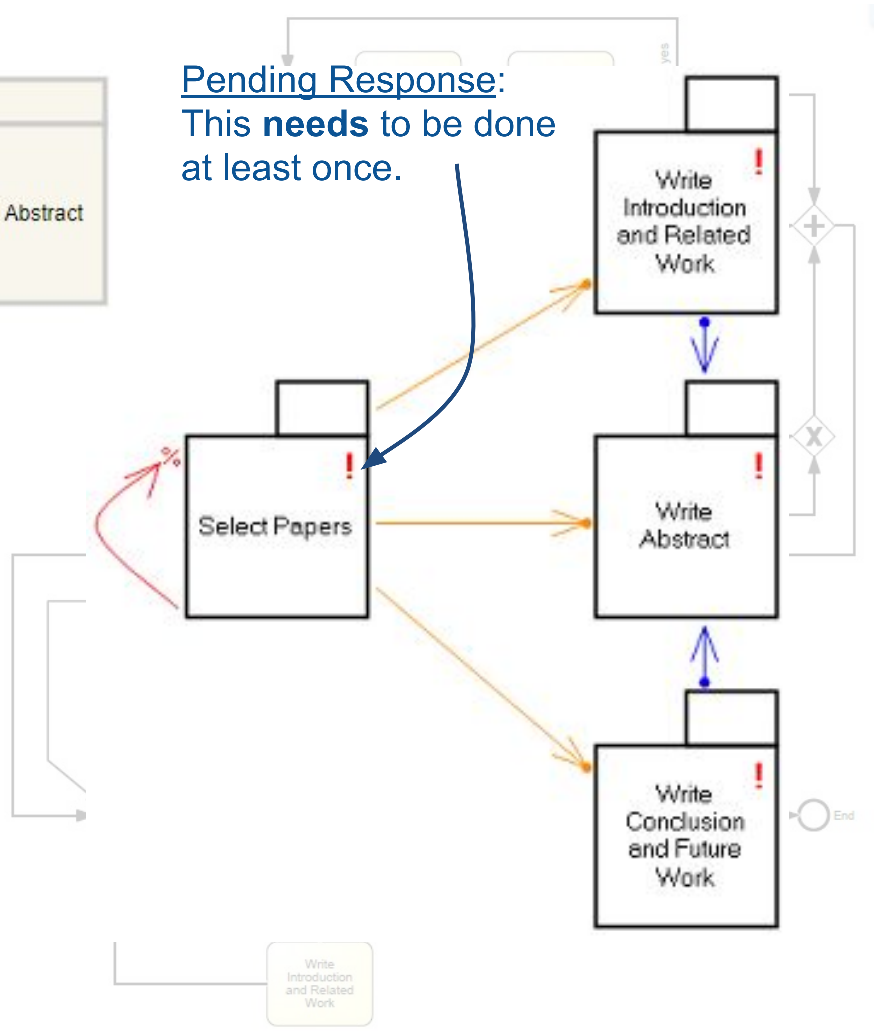


Pending Response: This **needs** to be done at least once.
Start
Select Papers
Write Introduction and Related Work
Write Abstract
Write Conclusion and Future Work
End
yes
%
!

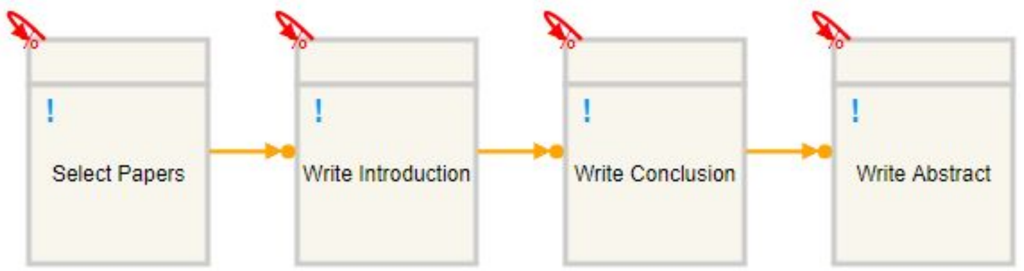
Example



Question: What language does this model capture?


Pending Response:
This **needs** to be done at least once.
Select Papers
Write Introduction and Related Work
Write Abstract
Write Conclusion and Future Work
End

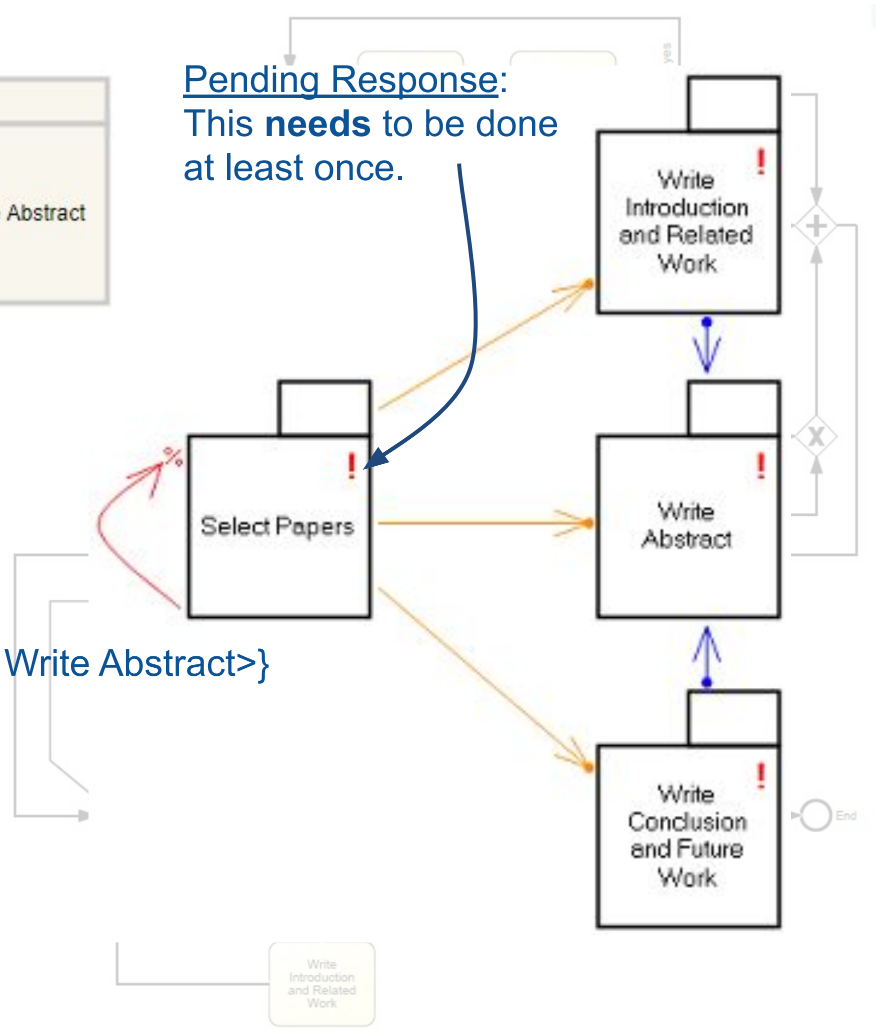
Example



Question: What language does this model capture?

{<Select Papers, Write Introduction, Write Conclusion, Write Abstract>}

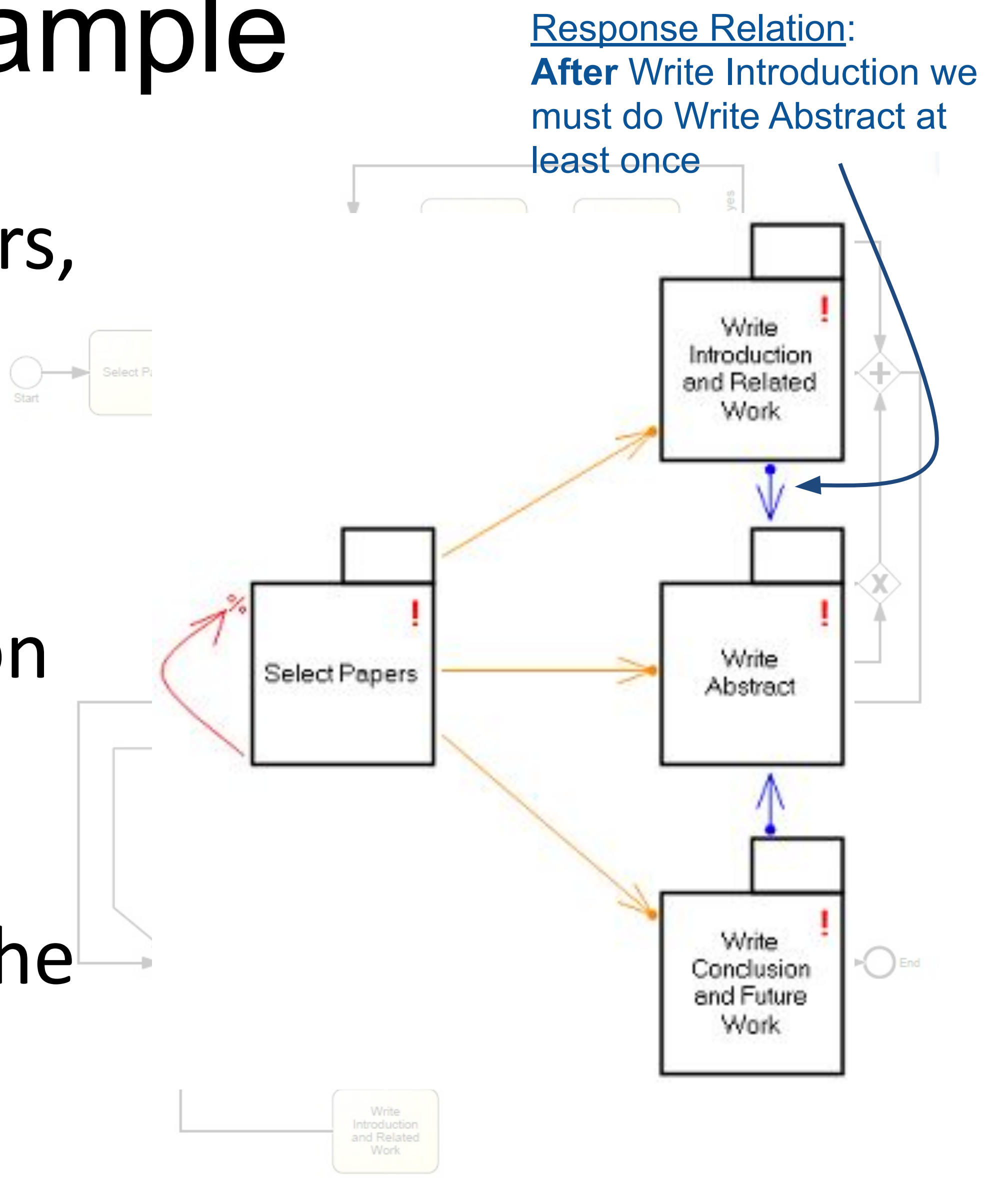
Pending Response: This **needs** to be done at least once.



Write Introduction and Related Work
Select Papers
Write Abstract
Write Conclusion and Future Work
yes
End
Write Introduction and Related Work

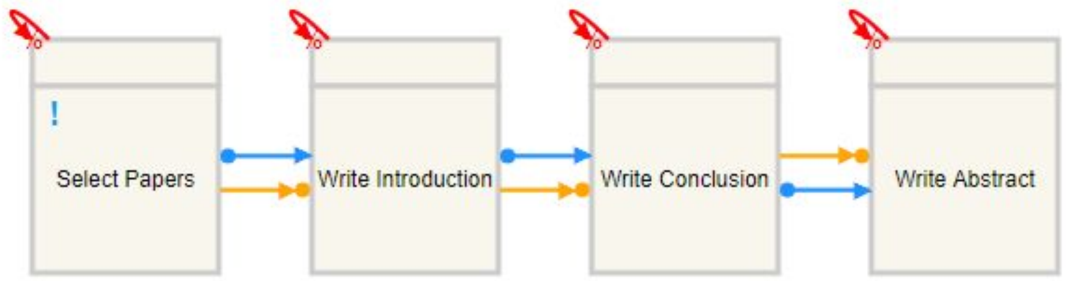
Example

- We first select papers, then:
- In any order, but at least once:
  - Write Introduction
  - Write Conclusion
  - Write Abstract
- We always update the abstract last



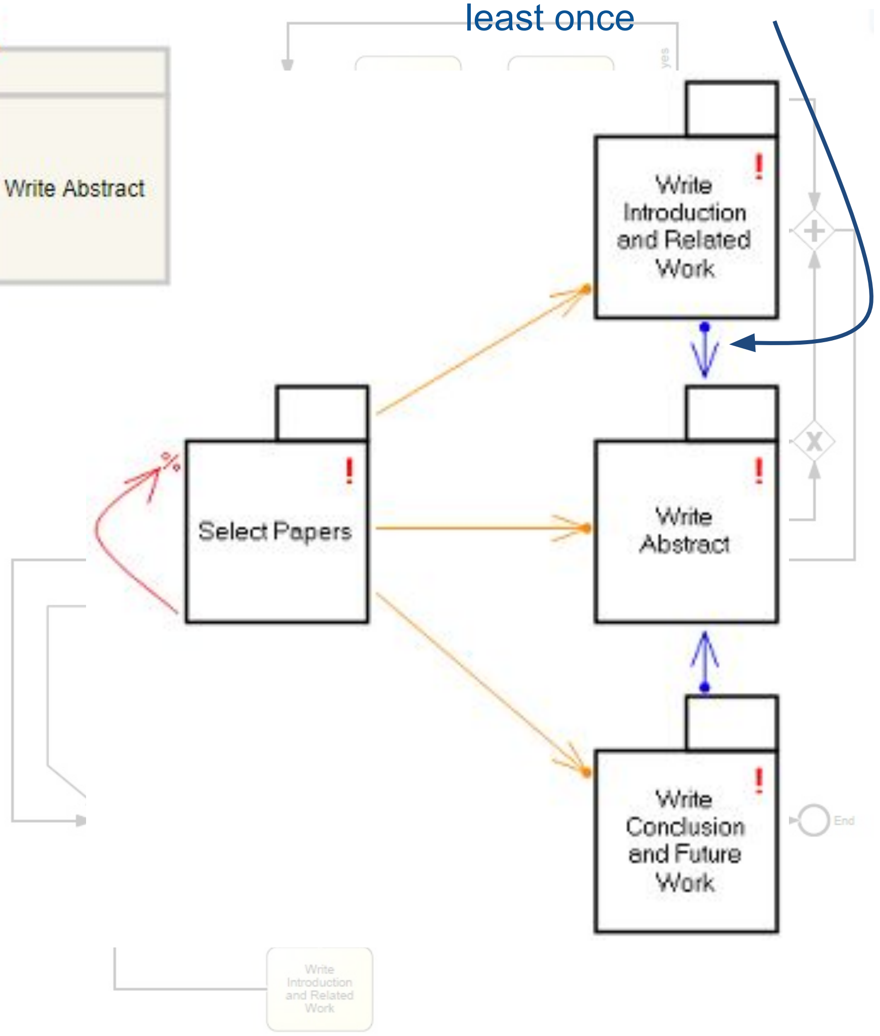
Response Relation:
After Write Introduction we must do Write Abstract at least once
Select Papers
Write Introduction and Related Work
Write Abstract
Write Conclusion and Future Work
Start
End

Example



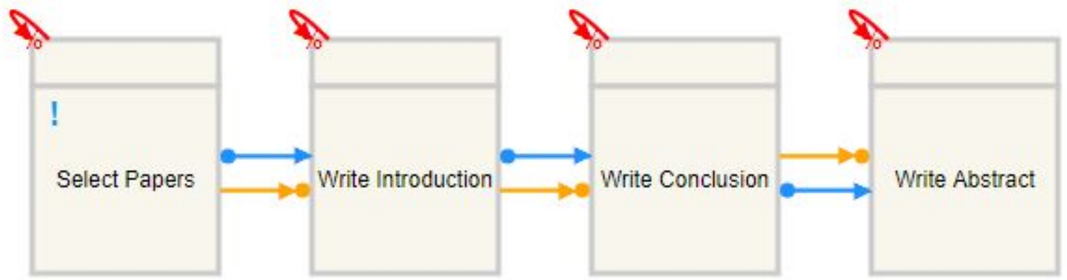
<u>Question</u>: What language does this model capture?

<u>Response Relation</u>:
After Write Introduction we must do Write Abstract at least once

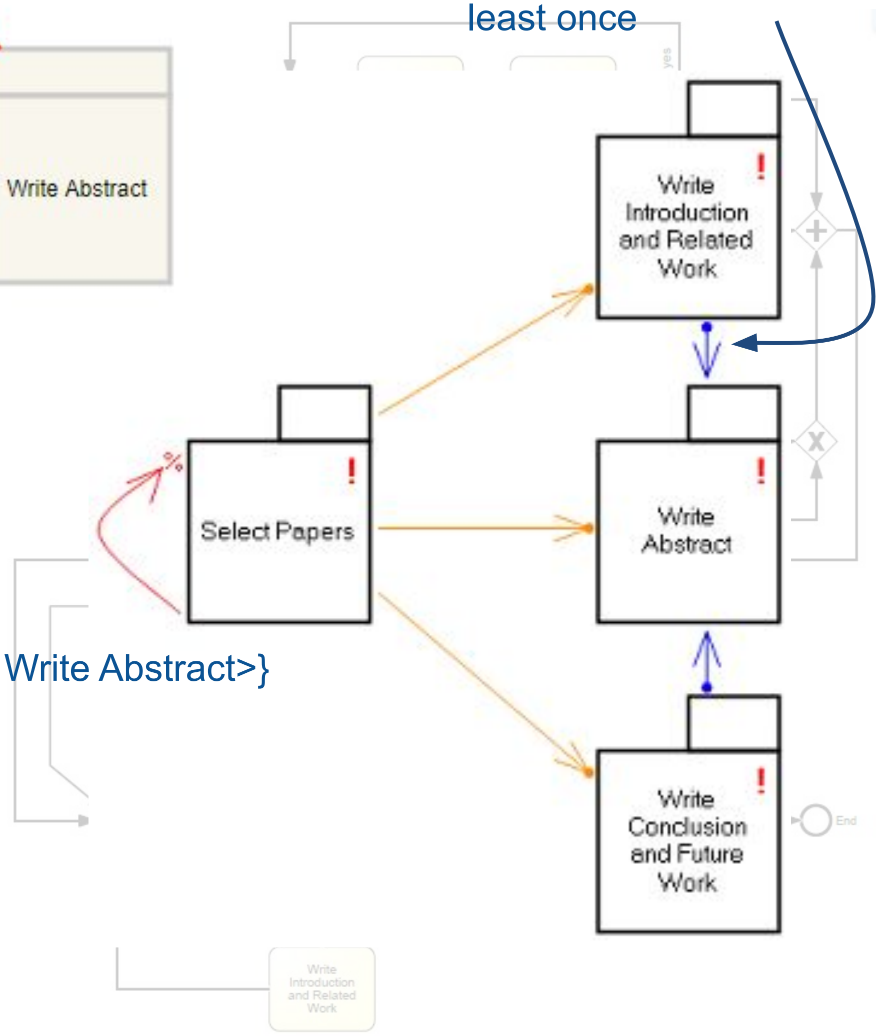


Write Introduction and Related Work
Select Papers
Write Abstract
Write Conclusion and Future Work
End

Example



Response Relation:
After Write Introduction we must do Write Abstract at least once



Write Introduction and Related Work
Select Papers
Write Abstract
Write Conclusion and Future Work
End

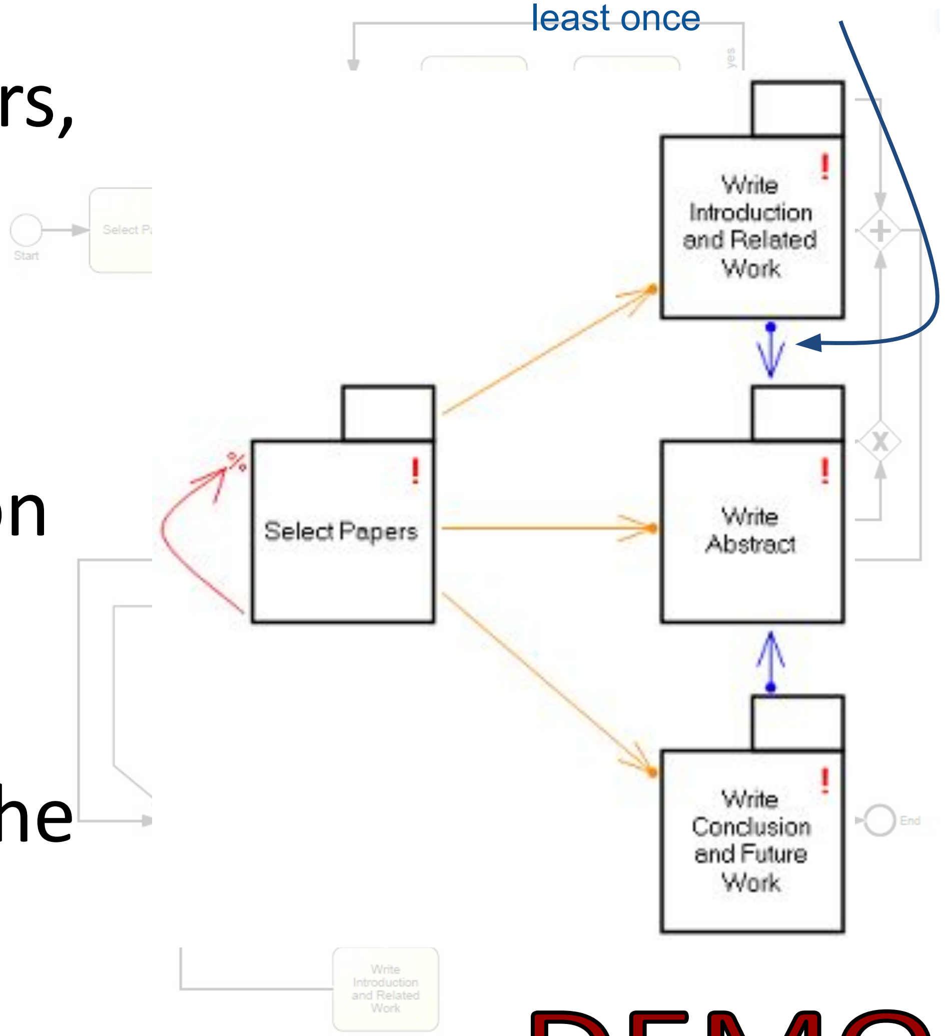
Question: What language does this model capture?

{<Select Papers, Write Introduction, Write Conclusion, Write Abstract>}

Example

- We first select papers, then:
- In any order, but at least once:
  - Write Introduction
  - Write Conclusion
  - Write Abstract
- We always update the abstract last

Response Relation:
After Write Introduction we must do Write Abstract at least once

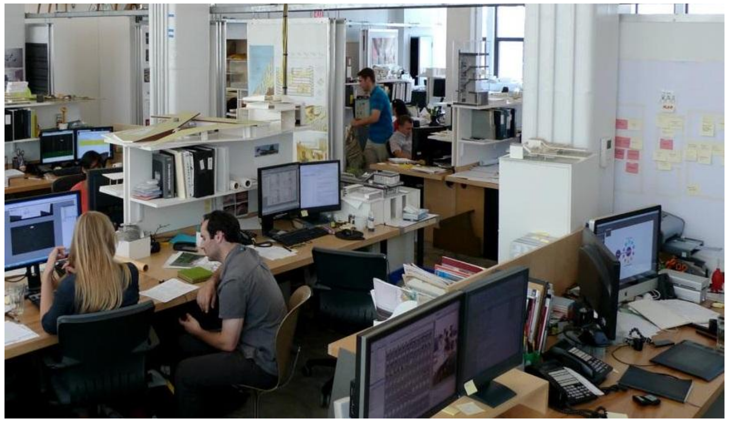


Start
Select Papers
Write Introduction and Related Work
Write Abstract
Write Conclusion and Future Work
End
%
!

DEMO

RECAP

Imperative notations:	**Declarative notations:**
Used for *structured* processes	Used for *flexible* processes
Describes *flow*	Describes *constraints*
Nothing allowed by default	*Everything* allowed by default
Describe *desired* behaviour	Describe *forbidden* behaviour



BREAK

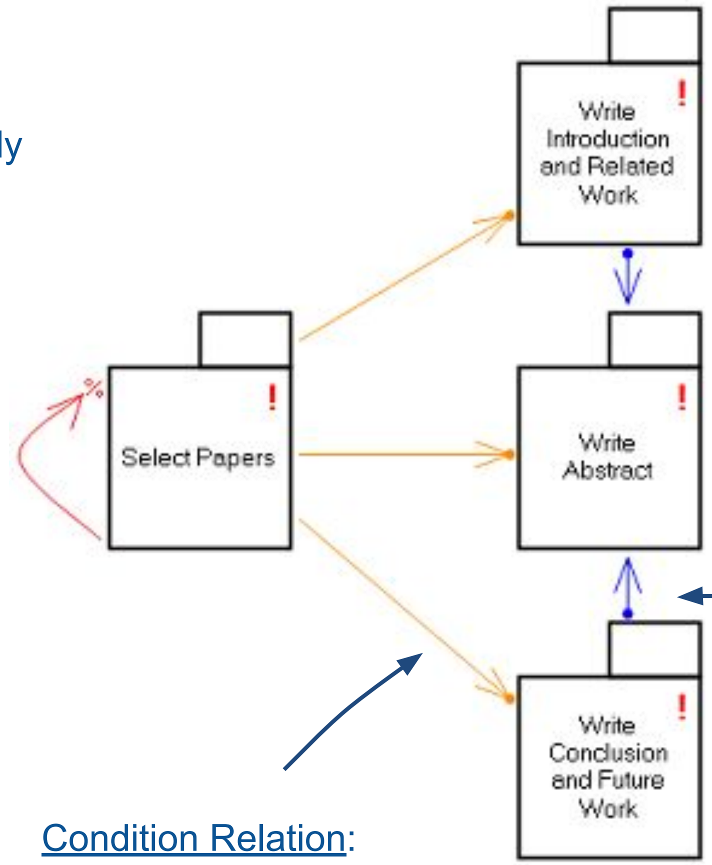
Overview

- Process Modelling
- Imperative vs Declarative Process Models
- **Dynamic Condition Response (DCR) Graphs**
- **Hierarchy in DCR GRaphs**
- Semantics of DCR Graphs
- Assignment 1

RECAP

<u>Pending Response</u>:
This **needs** to be done at least once.

<u>Exclusion Relation</u>:
Select Papers **removes** itself from the process. (i.e.: Can only be done once)



Write Introduction and Related Work
!
%
Select Papers
!
Write Abstract
!
Write Conclusion and Future Work
!

<u>Response Relation</u>:
After Write Introduction we must do Write Abstract at least once

<u>Condition Relation</u>:
We must do Select Papers **before** Write Introduction.

Another example

Electronic Case Management System

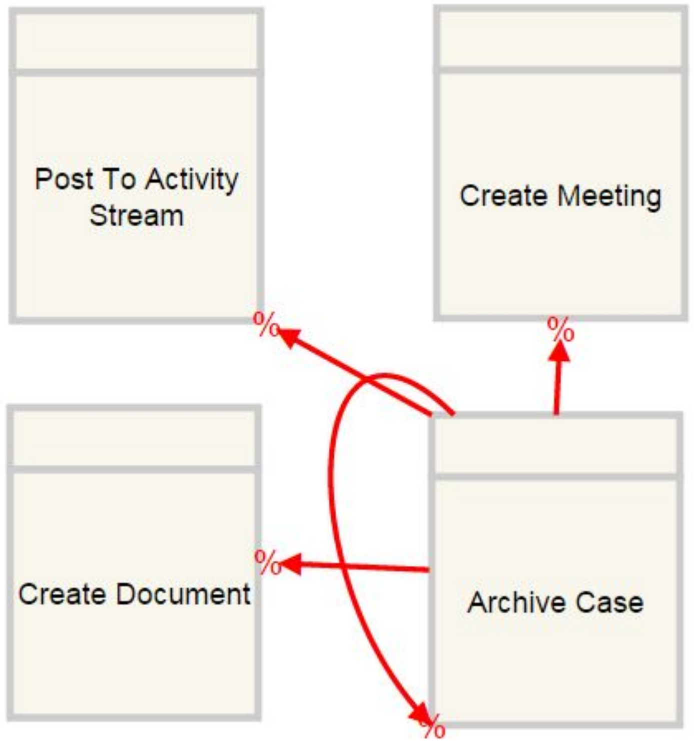
- Centered around concept of a case:
  - Legal cases
  - Insurance claims
  - Patient care
  - etc…
- Focuses on facilitating the communication, document management and workflow of caseworkers

ECM Example

Three main activities:

- Post to Activity Stream
- Create Meeting
- Create Document

Archive Case closes the case by removing all activities

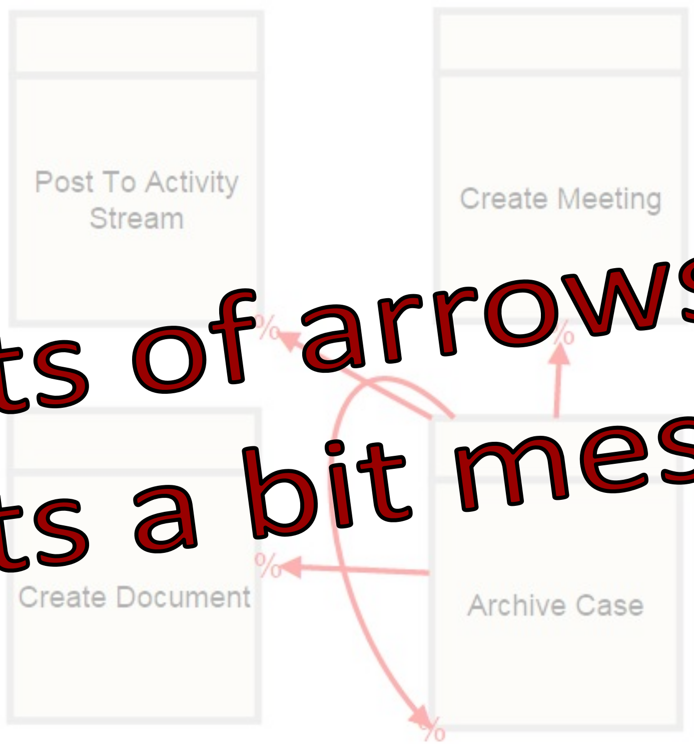


ECM Example

Three main activities:

- Post to Activity Stream
- Create Meeting
- Create Document

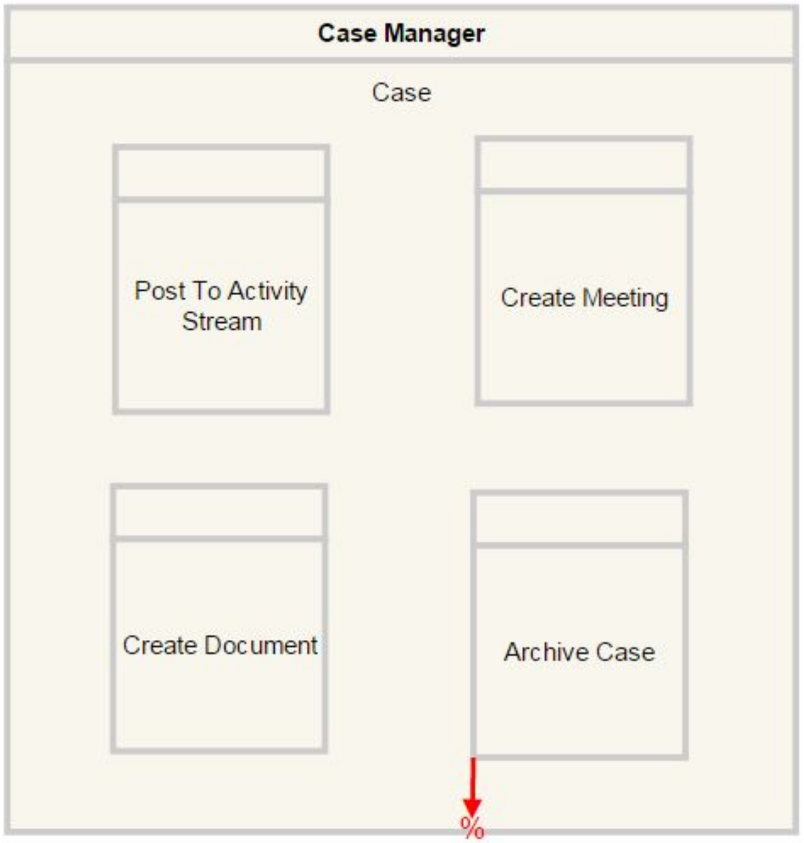
Archive Case closes the case by removing all activities



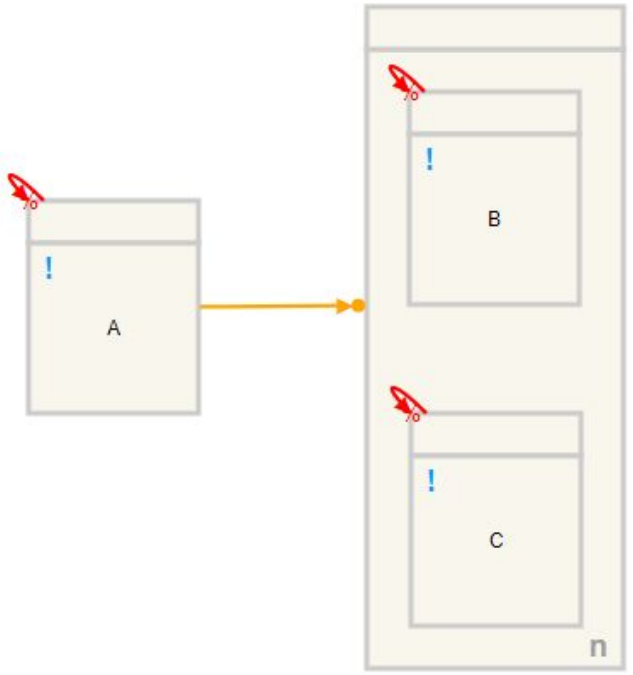
Lots of arrows... gets a bit messy!

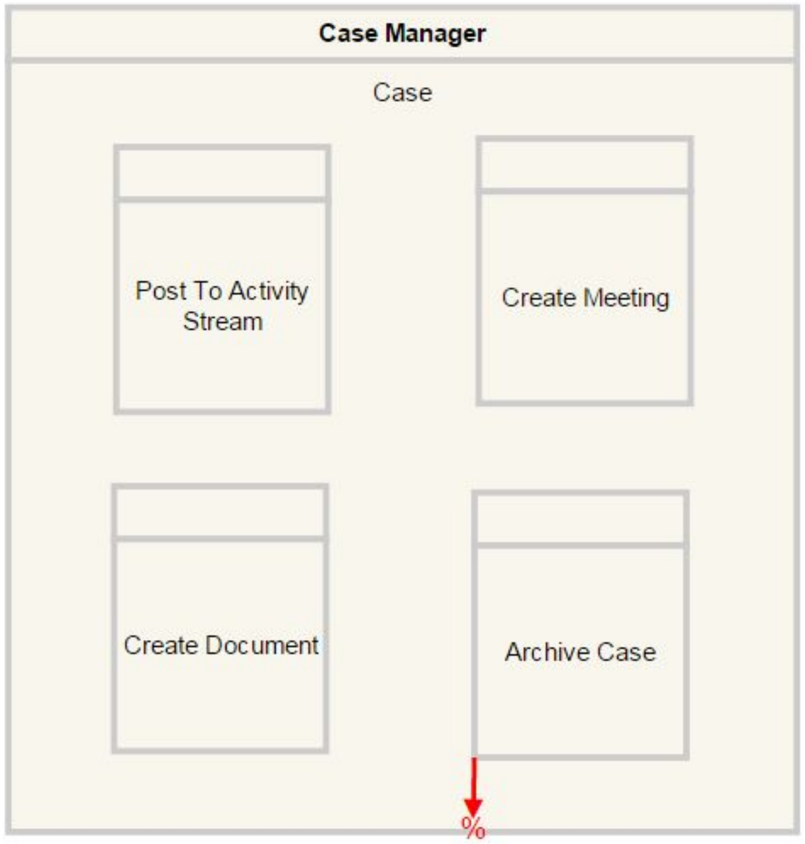
ECM Example - Nesting

- Group activities together
- Only atomic activities are executable
- Nesting serves as a shorthand for applying relations to more than one activity



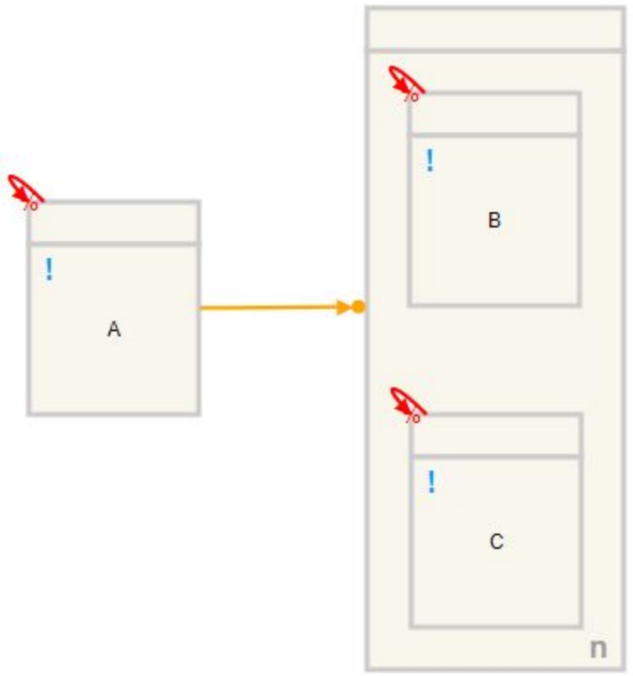
ECM Example - Nesting

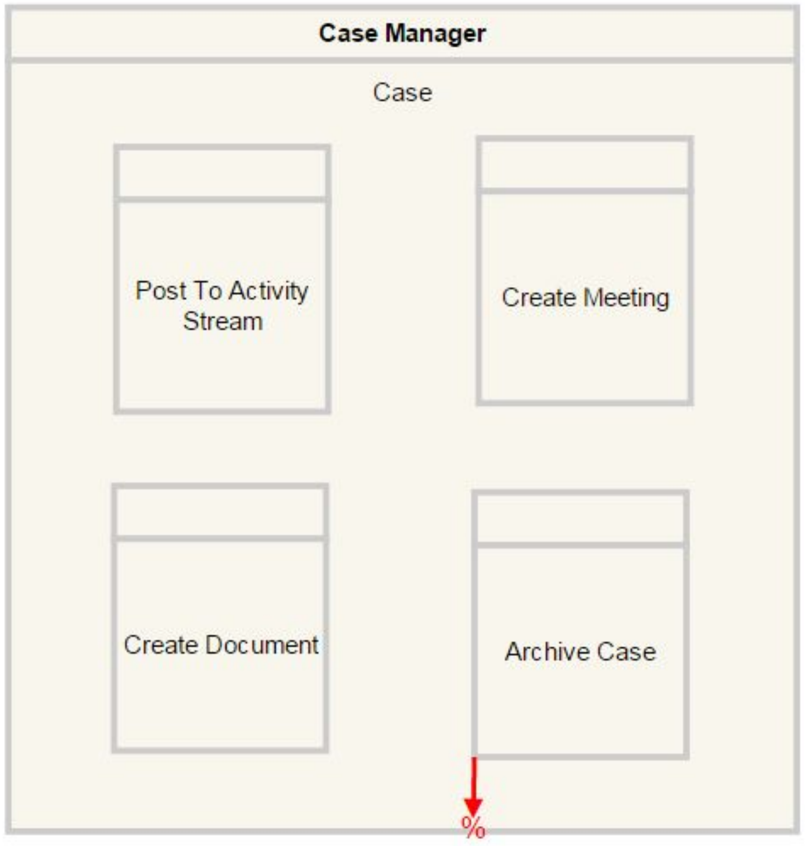




Question: What language does this model capture?

ECM Example - Nesting

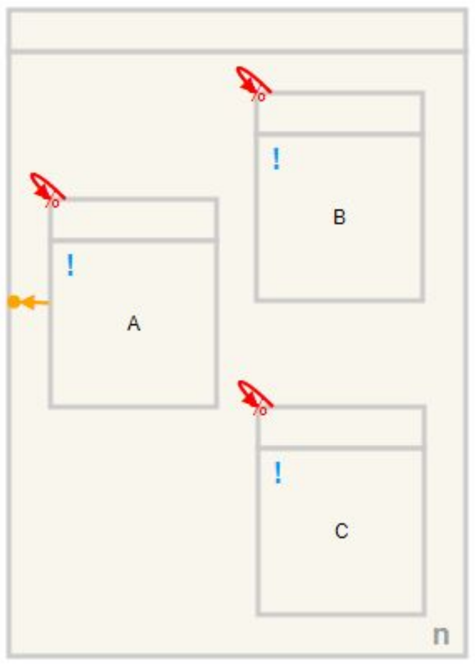


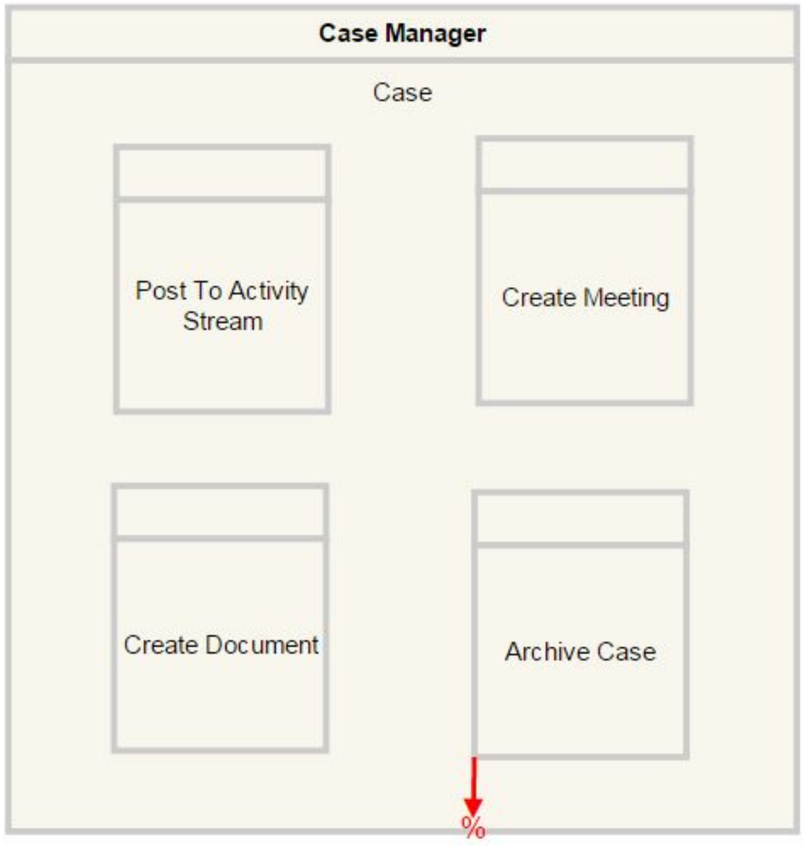


Question: What language does this model capture?

{<A, B, C>,
<A, C, B>}

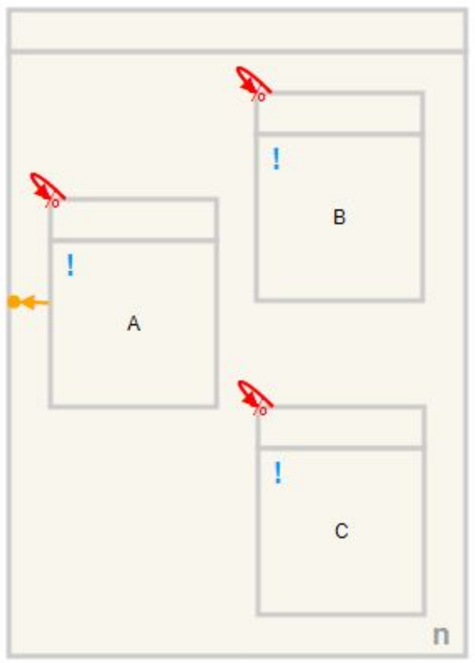
ECM Example - Nesting

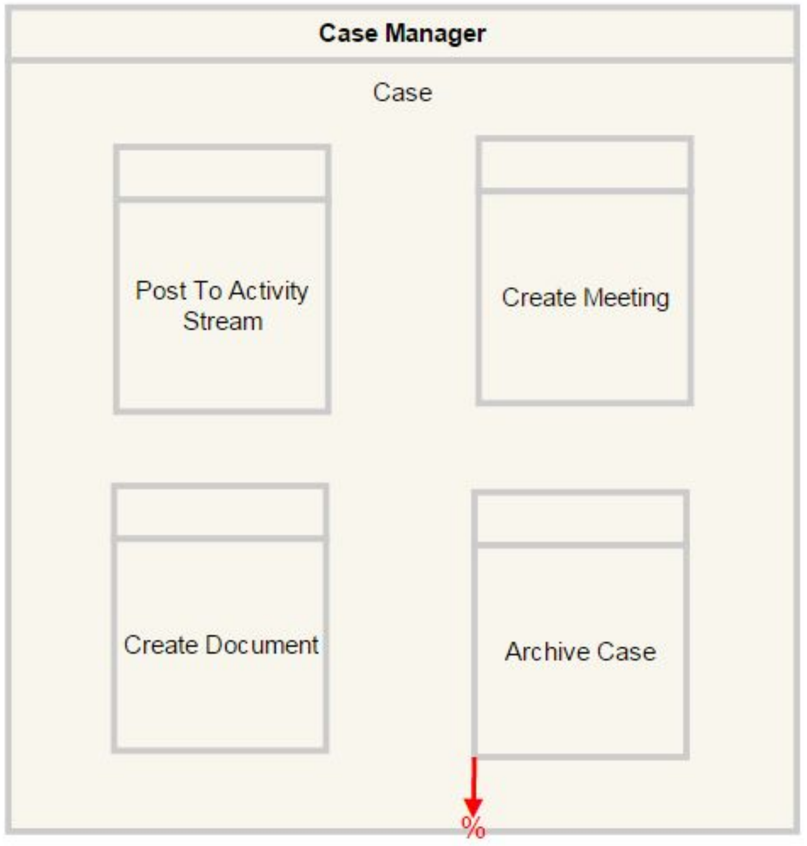




Question: What language does this model capture?

ECM Example - Nesting





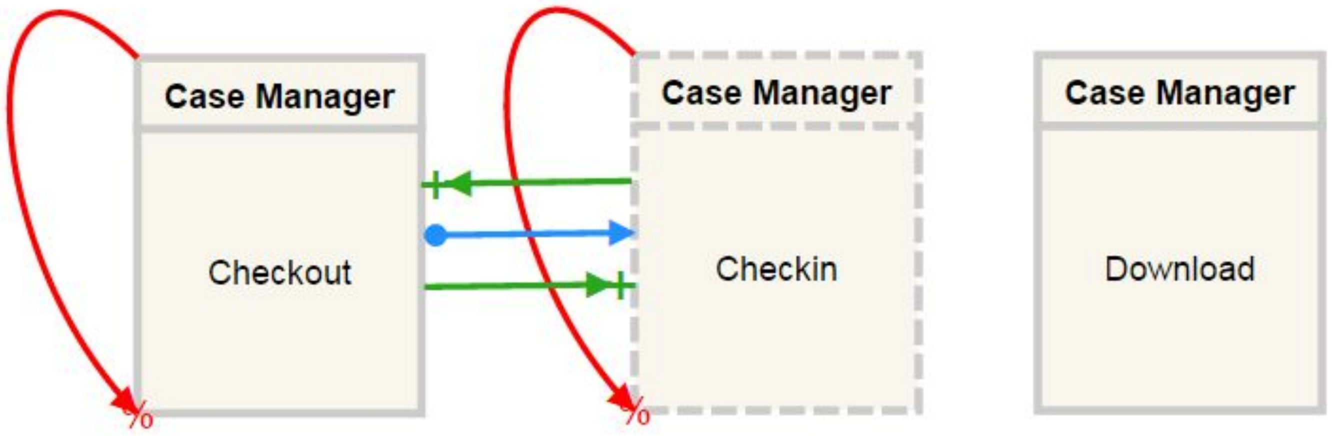
Question: What language does this model capture?

{}

ECM Example

Document handling process

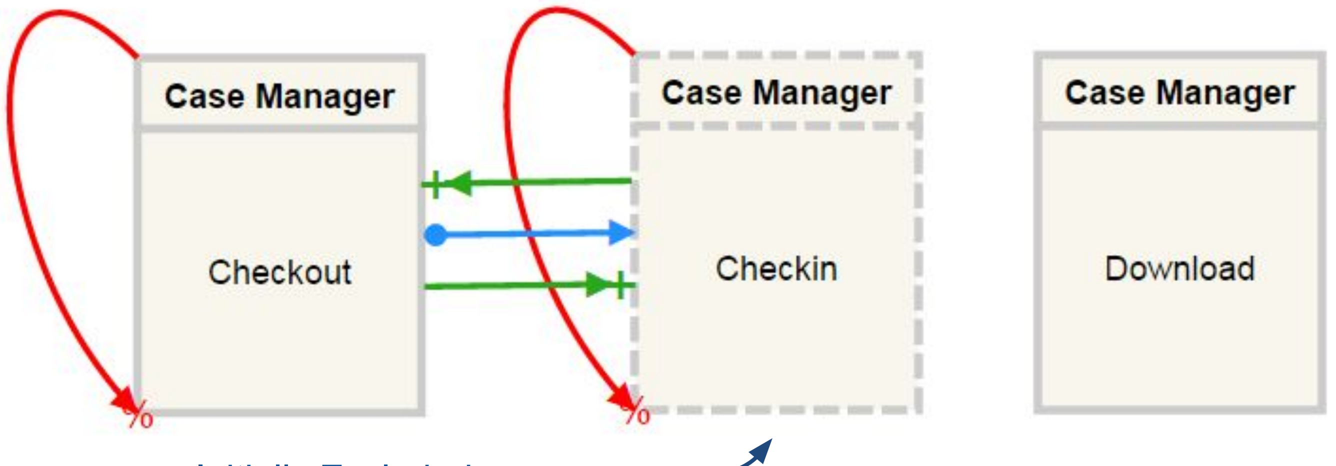
- A file is **checked in** or **checked out**
- Eventually the file should always be checked in
- A file can always be downloaded for viewing



ECM Example

Document handling process

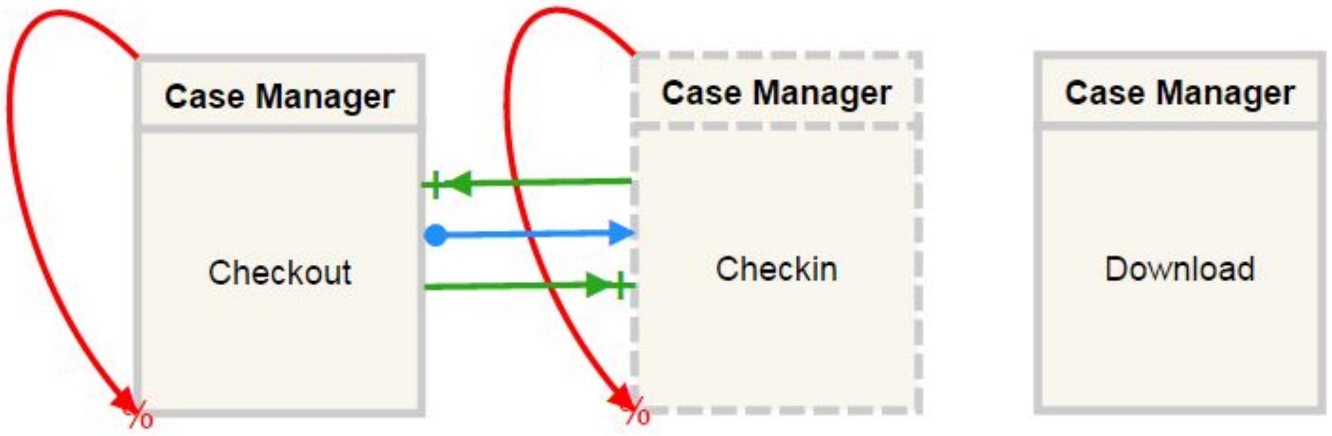
- A file is **checked in** or **checked out**
- Eventually the file should always be checked in
- A file can always be downloaded for viewing



Initially Excluded:
When a file is created it is already checked in, so this activity is not yet enabled.

ECM Example

Question: What language does this model capture?

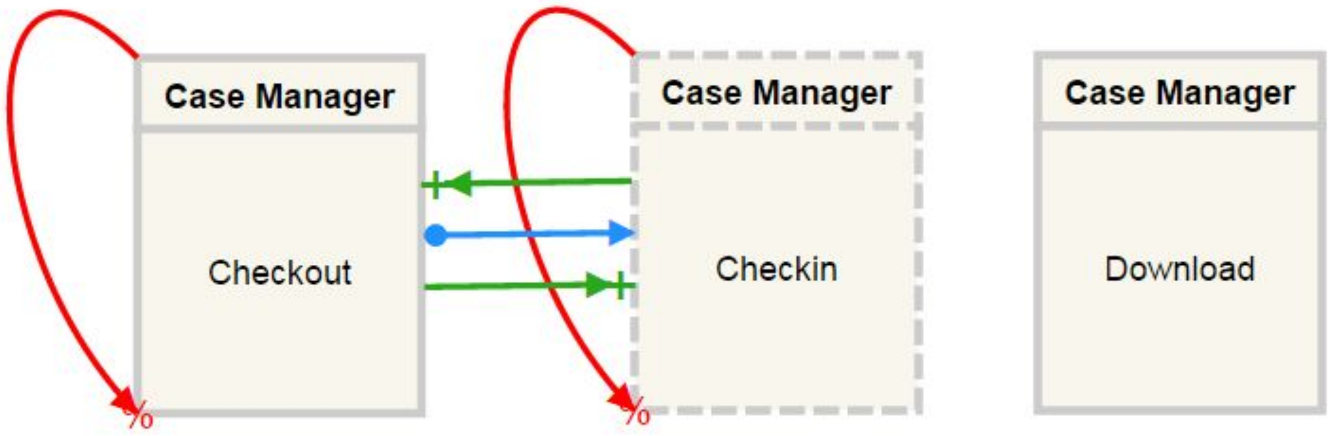


Initially Excluded:
When a file is created it is already checked in, so this activity is not yet enabled.

ECM Example

Question: What language does this model capture?

{<>}

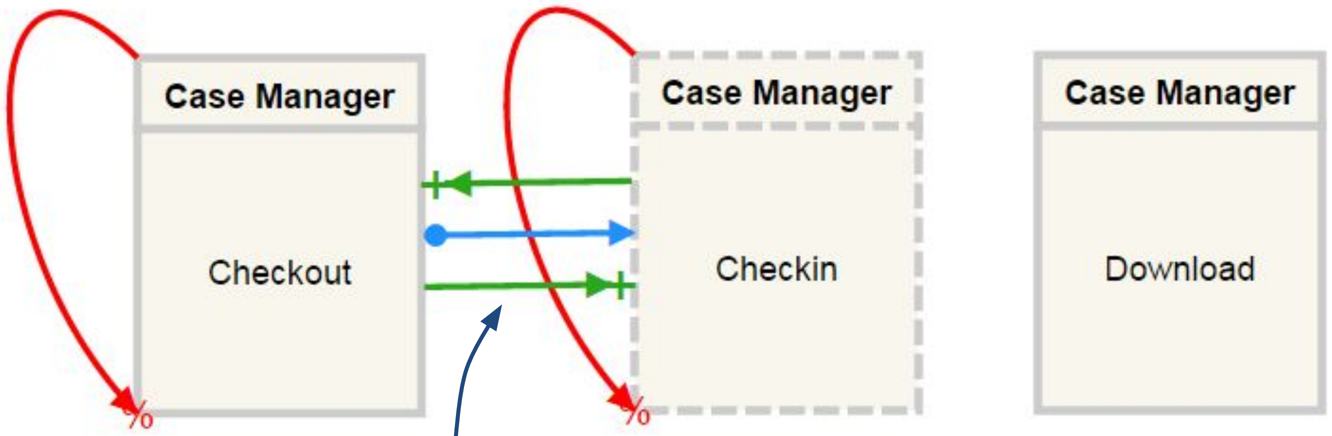


Initially Excluded:
When a file is created it is already checked in, so this activity is not yet enabled.

ECM Example

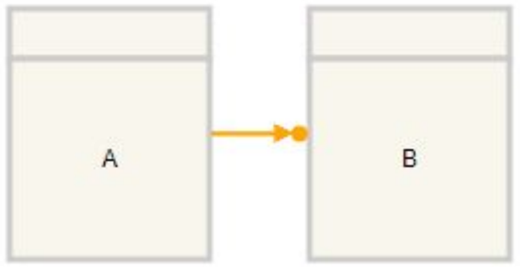
Document handling process

- A file is **checked in** or **checked out**
- Eventually the file should always be checked in
- A file can always be downloaded for viewing

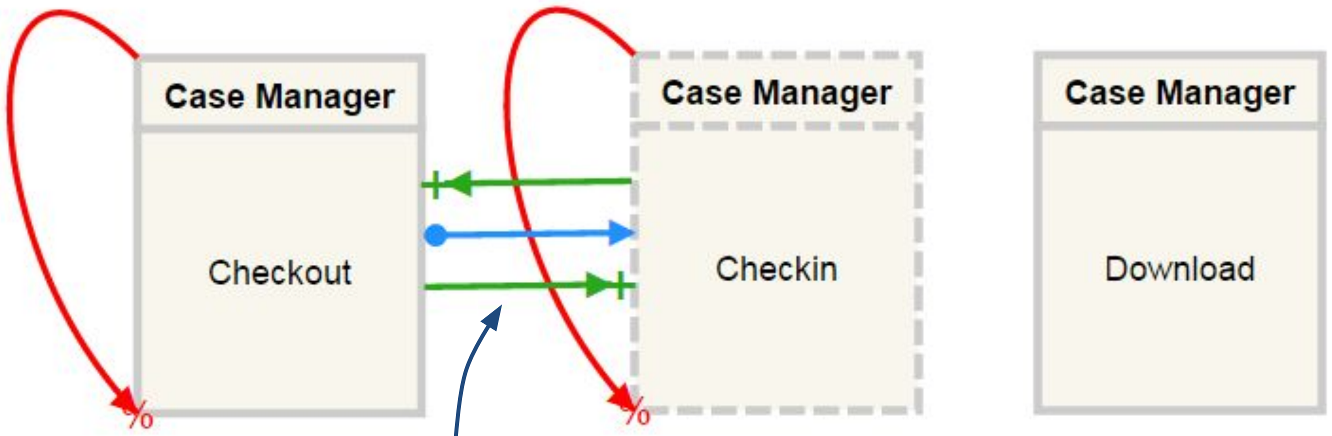


Inclusion Relation:
Checkout **Adds** Checkin (back) into the process

ECM Example

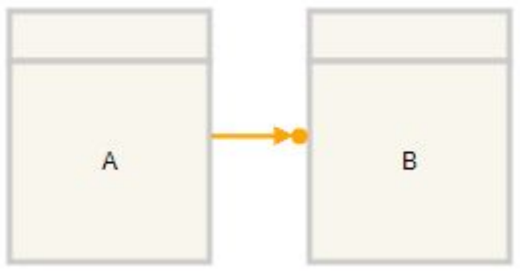


Question: Can I model the same process without using conditions?

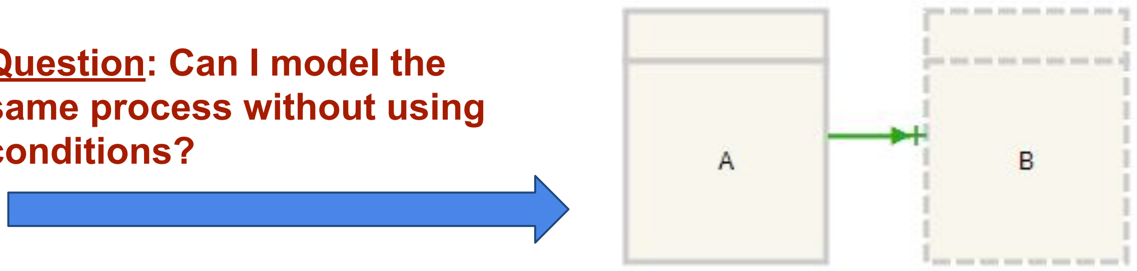


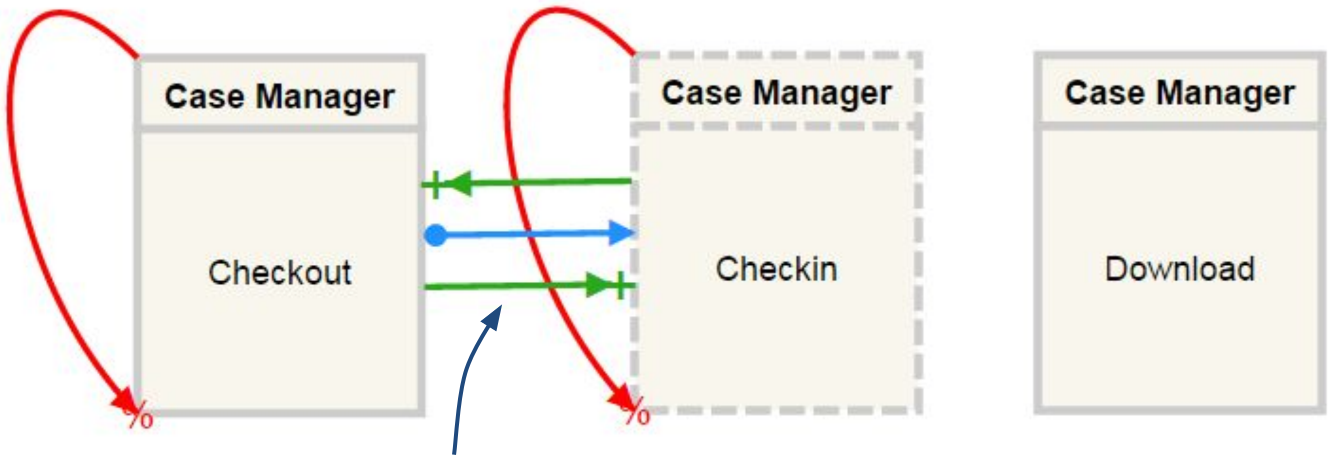
Inclusion Relation:
Checkout **Adds** Checkin (back) into the process

ECM Example



Question: Can I model the same process without using conditions?



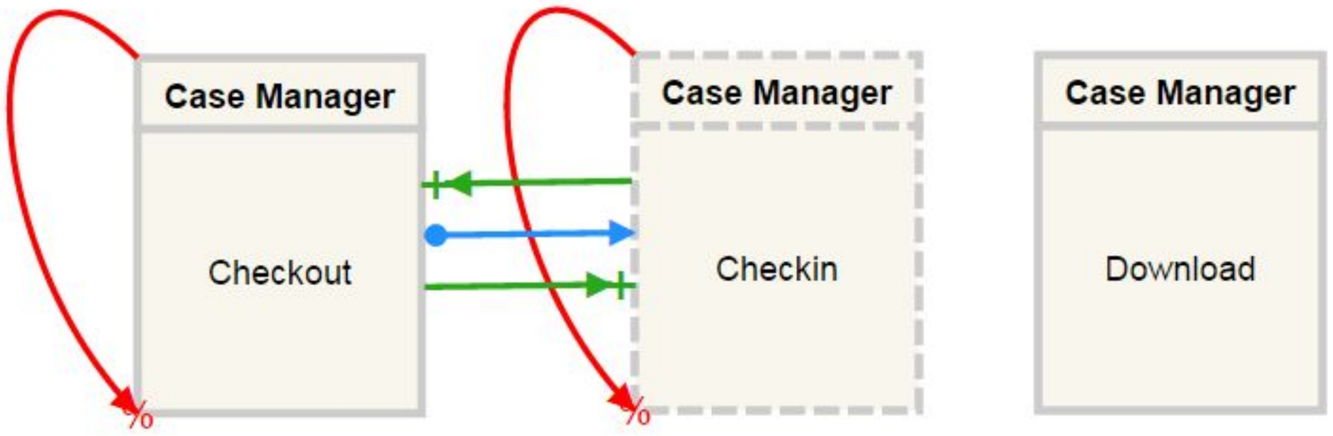


Inclusion Relation:
Checkout **Adds** Checkin (back) into the process

ECM Example

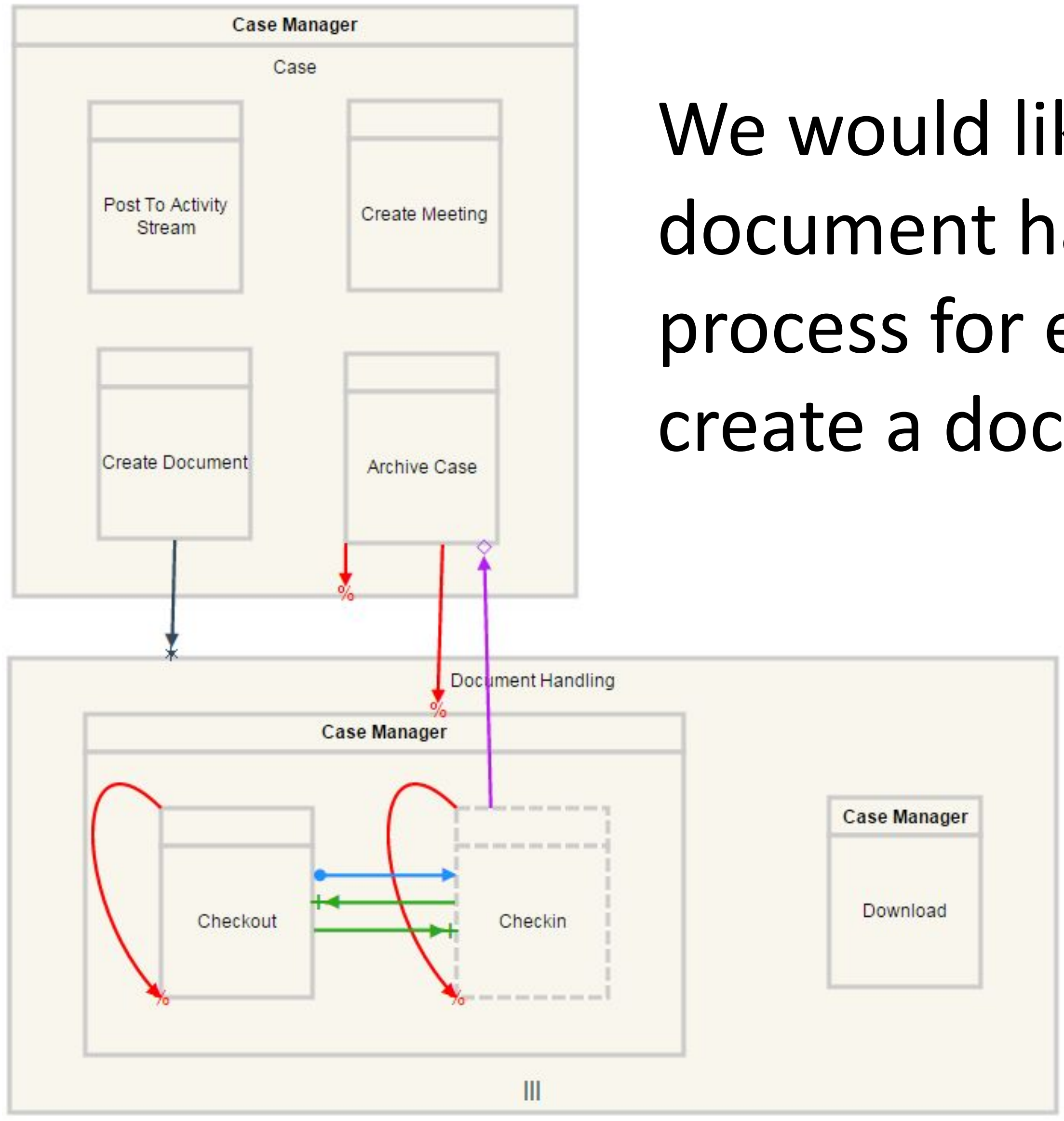
Document handling process

- A file is **checked in** or **checked out**
- Eventually the file should always be checked in
- A file can always be downloaded for viewing



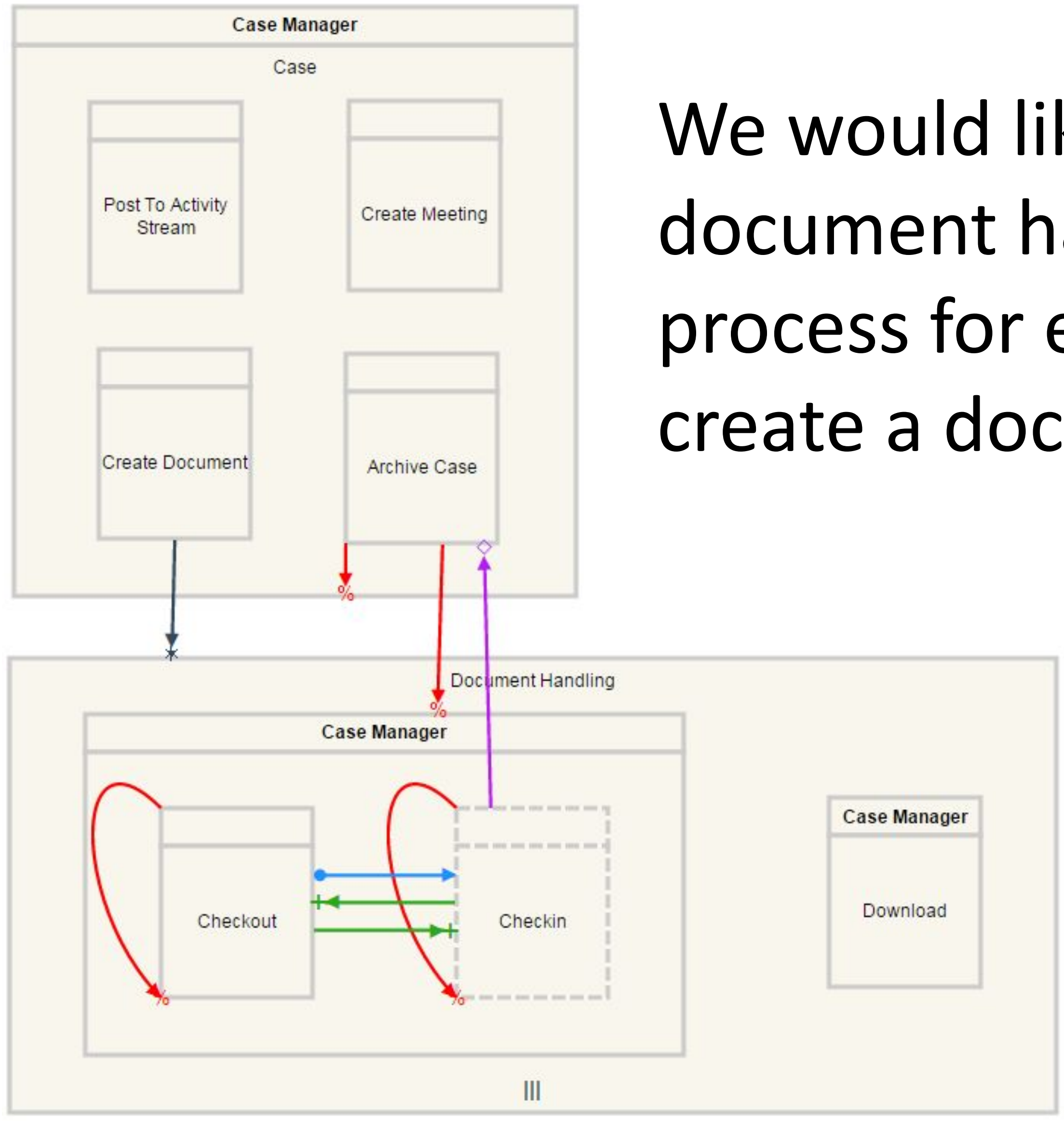
Note: The expressiveness of DCR Graphs with these 4 basic relations is equal to the union of regular and ω-regular languages

ECM Example - Subprocesses



We would like to start a document handling sub-process for each time we create a document.

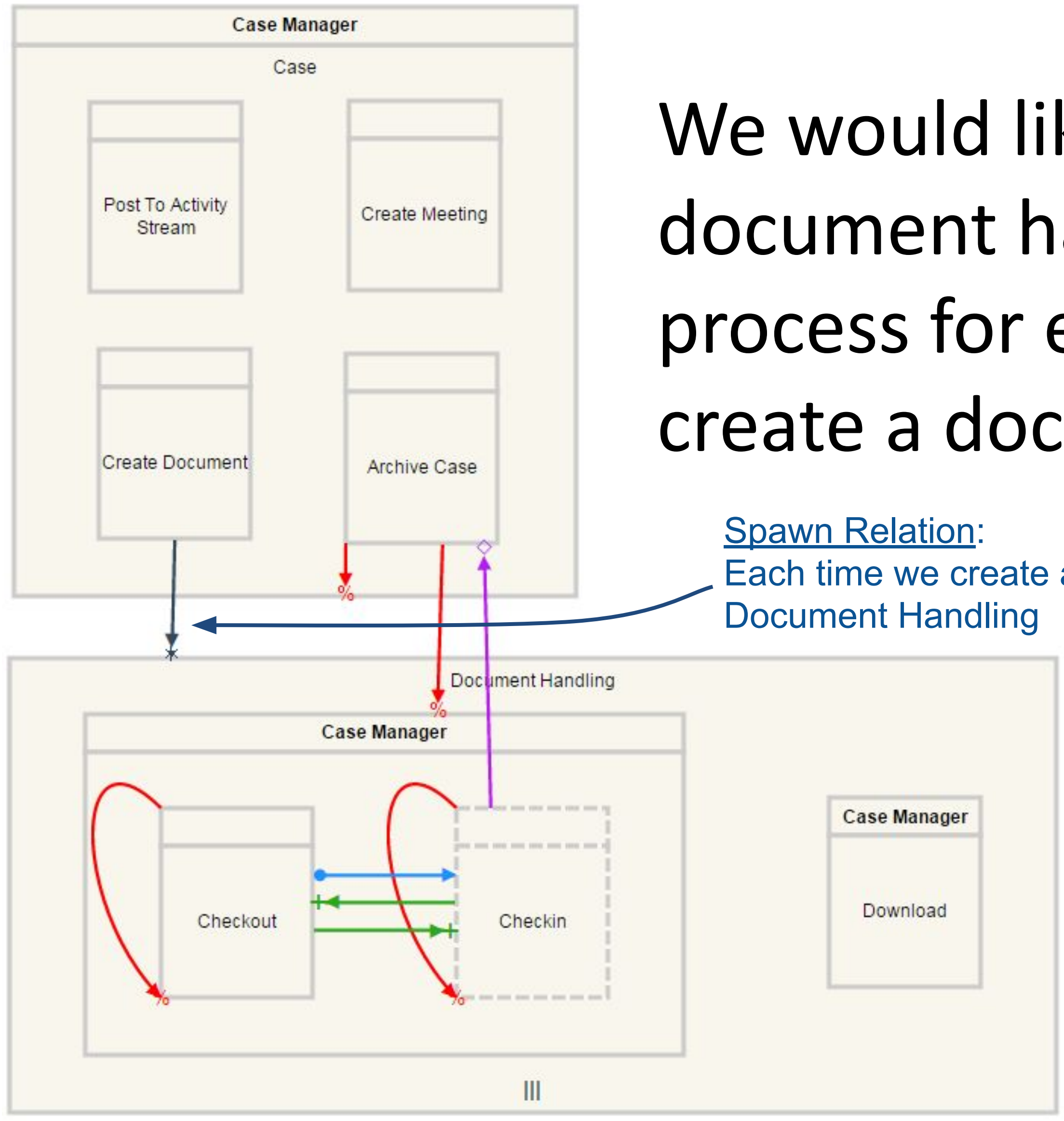
ECM Example - Subprocesses



We would like to start a document handling sub-process for each time we create a document.

Multi-instance Sub-process:
A **template** of another process, does not exist on its own but needs to be **instantiated**

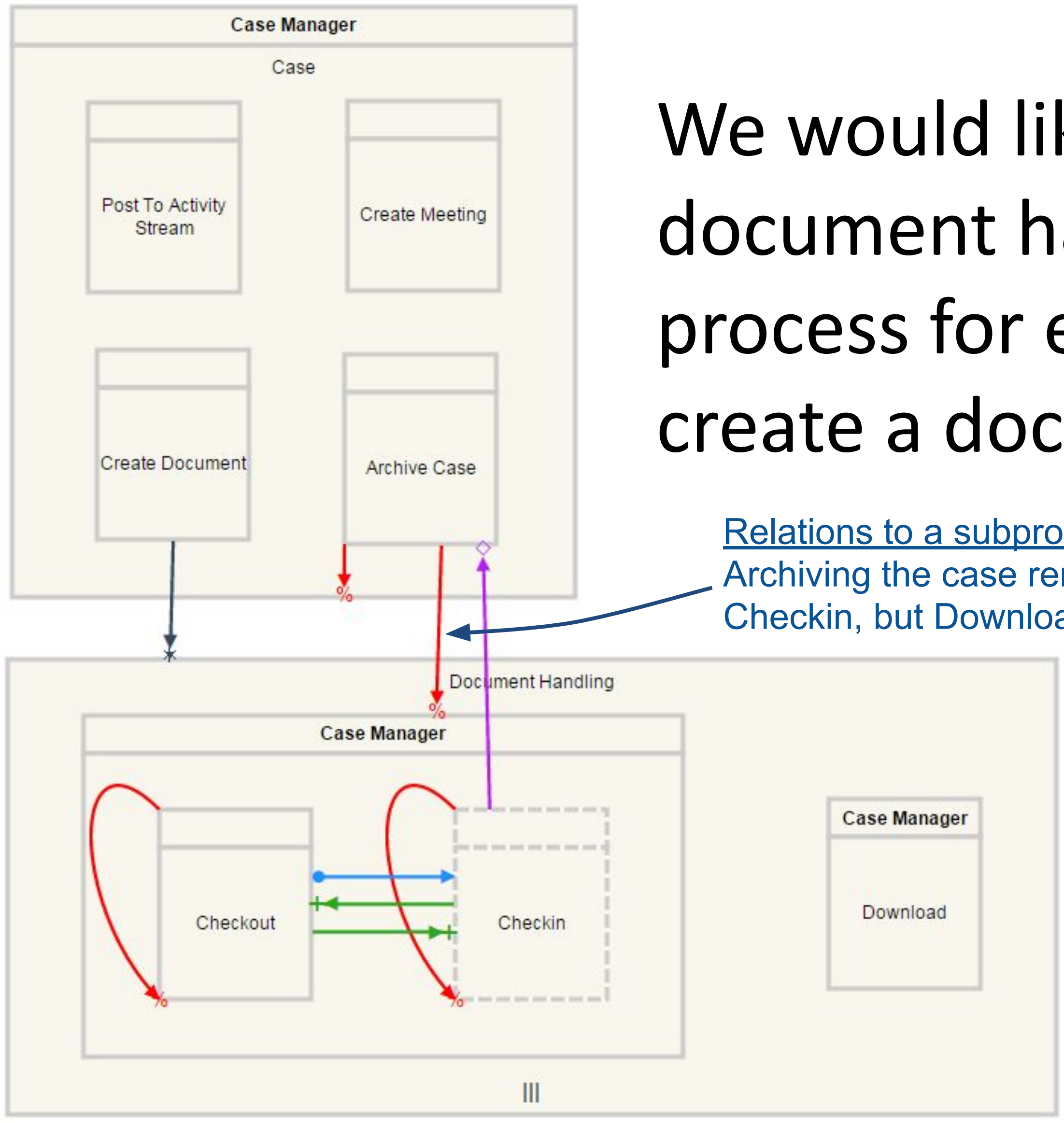
ECM Example - Subprocesses



We would like to start a document handling sub-process for each time we create a document.

Spawn Relation:
Each time we create a document, we **spawn** a new copy of Document Handling

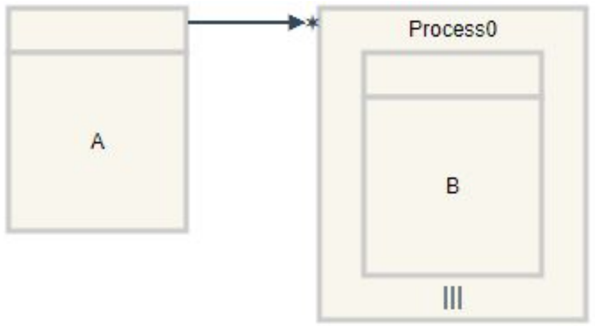
ECM Example - Subprocesses



We would like to start a document handling sub-process for each time we create a document.

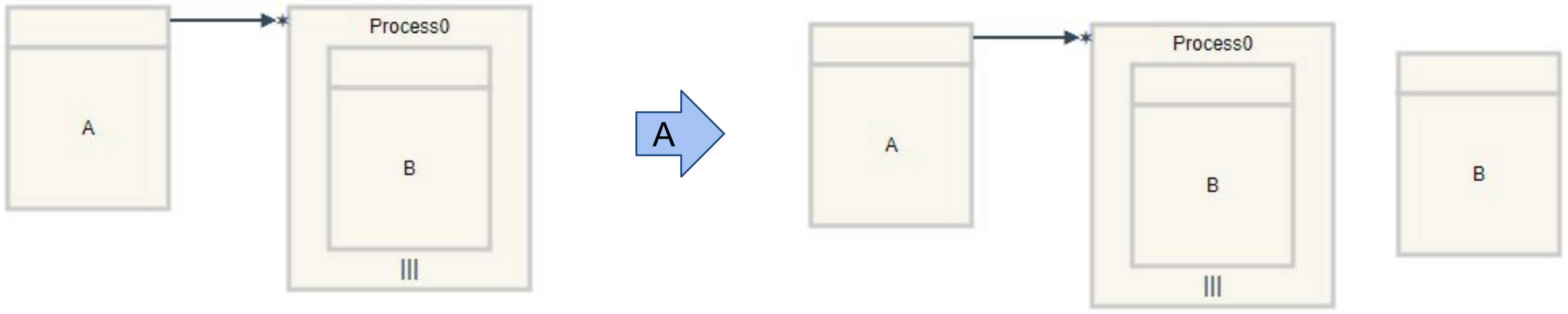
Relations to a subprocess:
Archiving the case removes **all** instances of Checkout and Checkin, but Download remains available

Subprocesses



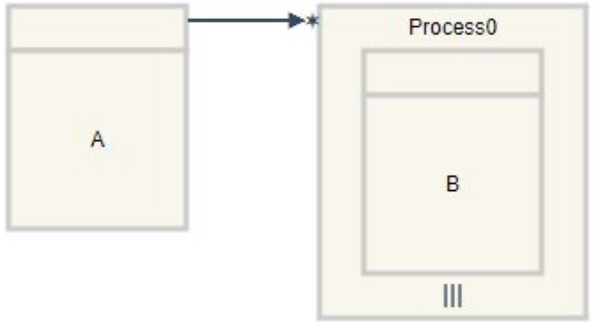
Enabled: A

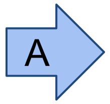
Subprocesses

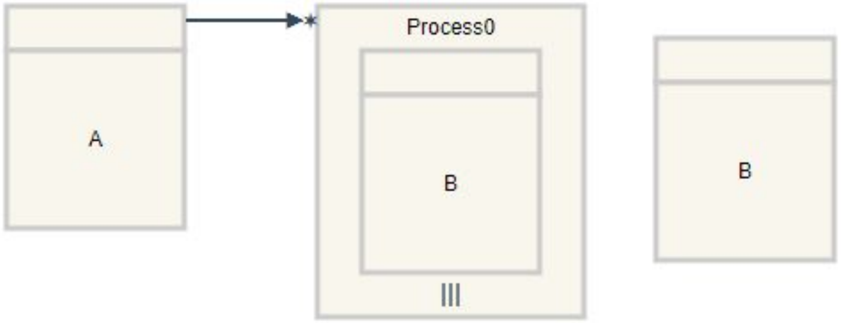


Enabled: A

Subprocesses

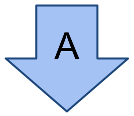


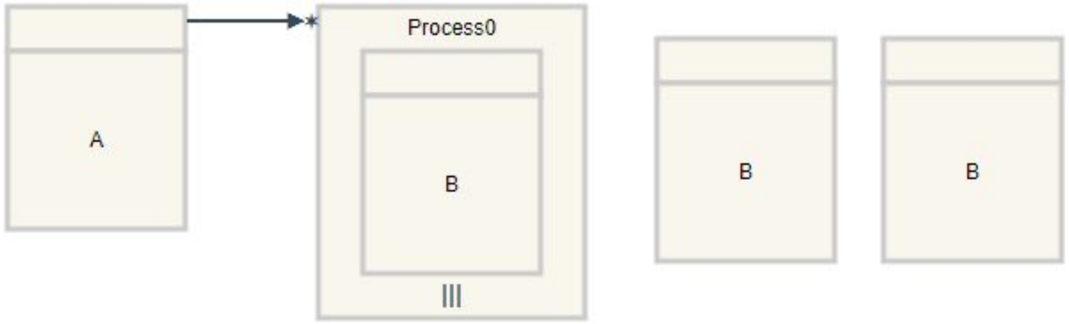




Enabled: A

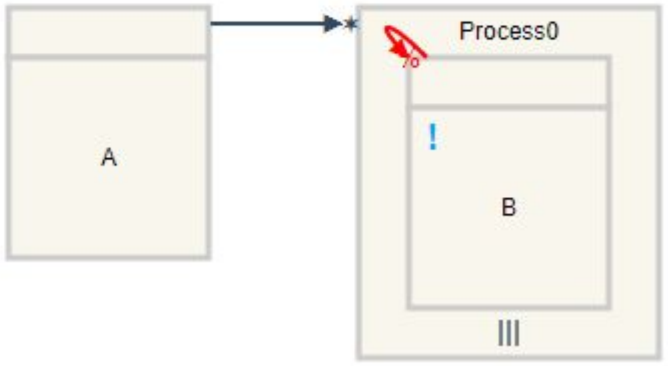
Enabled: A, B





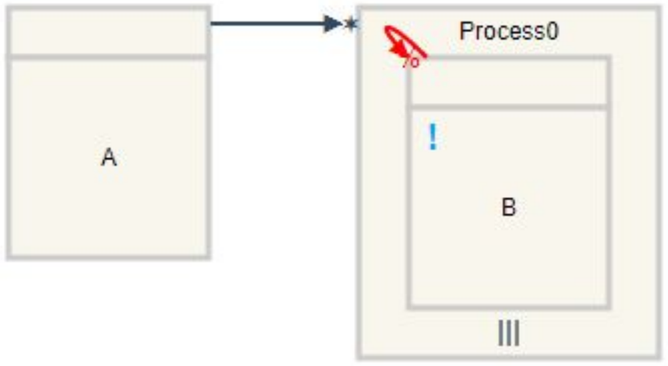
Enabled: A, B, B

Subprocesses



<u>Question</u>: What language does this DCR Graph capture?

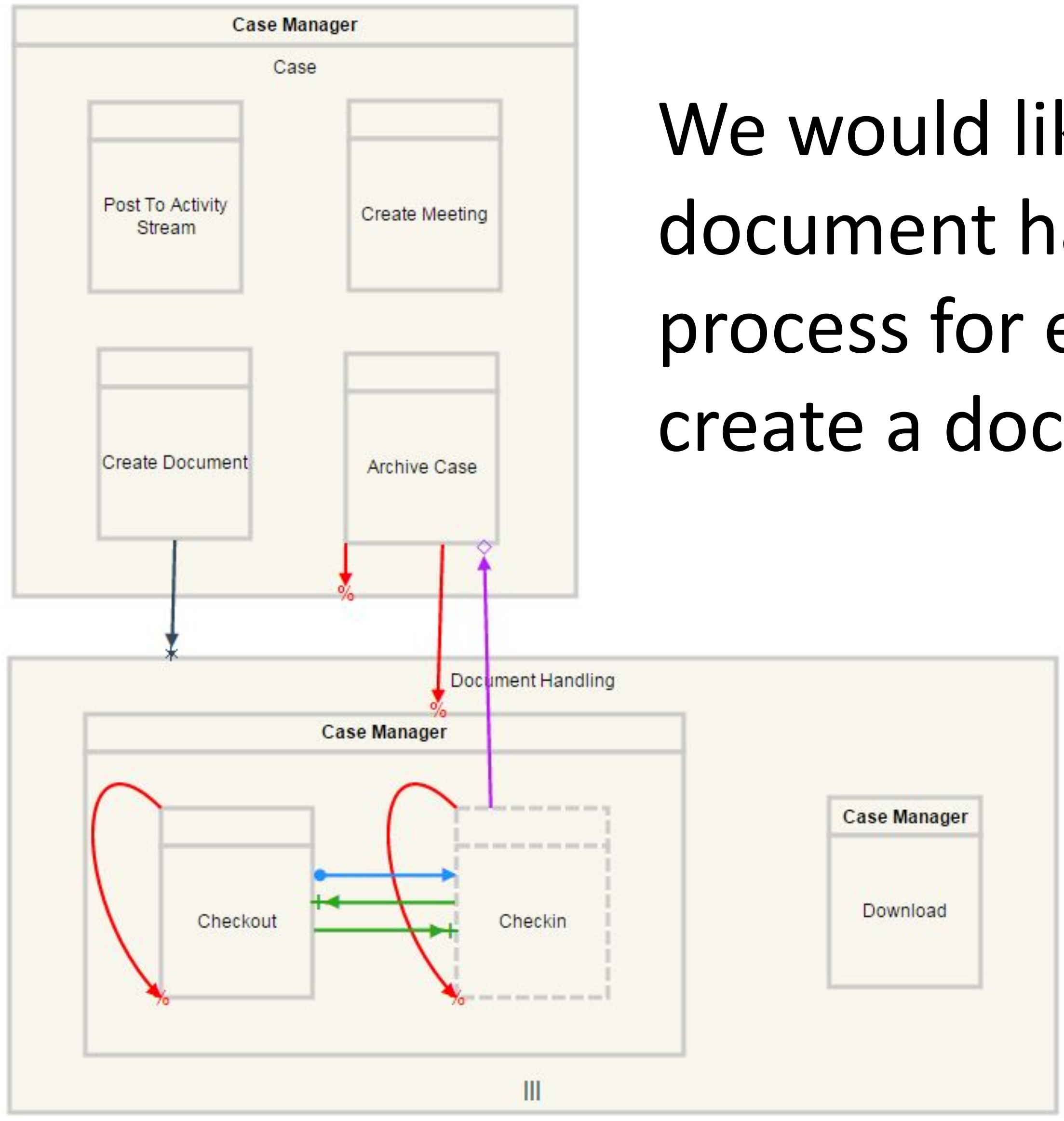
Subprocesses



Question: What language does this DCR Graph capture?

For each A there must be exactly one B.

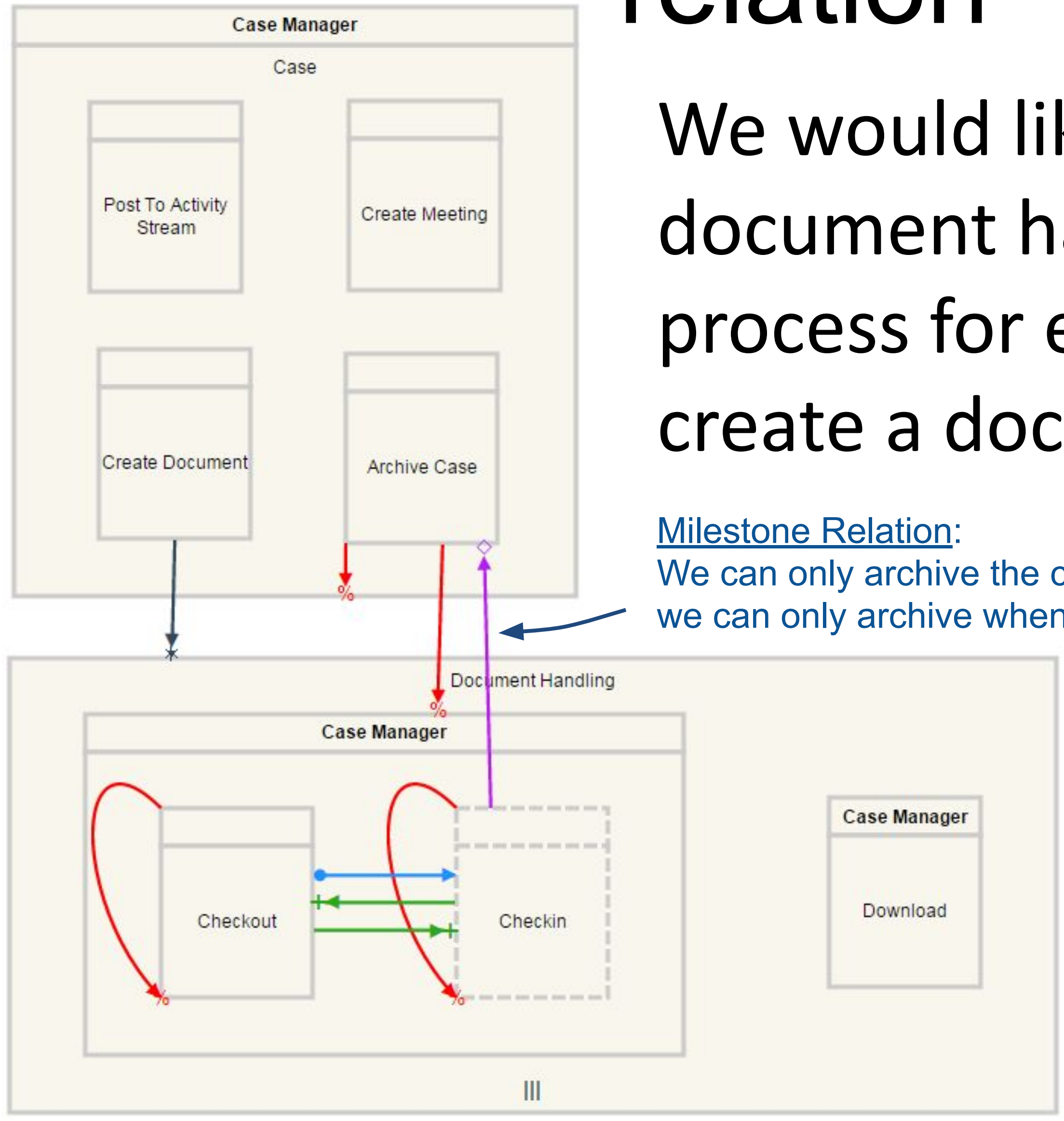
ECM Example - Subprocesses



We would like to start a document handling sub-process for each time we create a document.

Note: Adding the spawn relation makes DCR Graphs Turing complete!

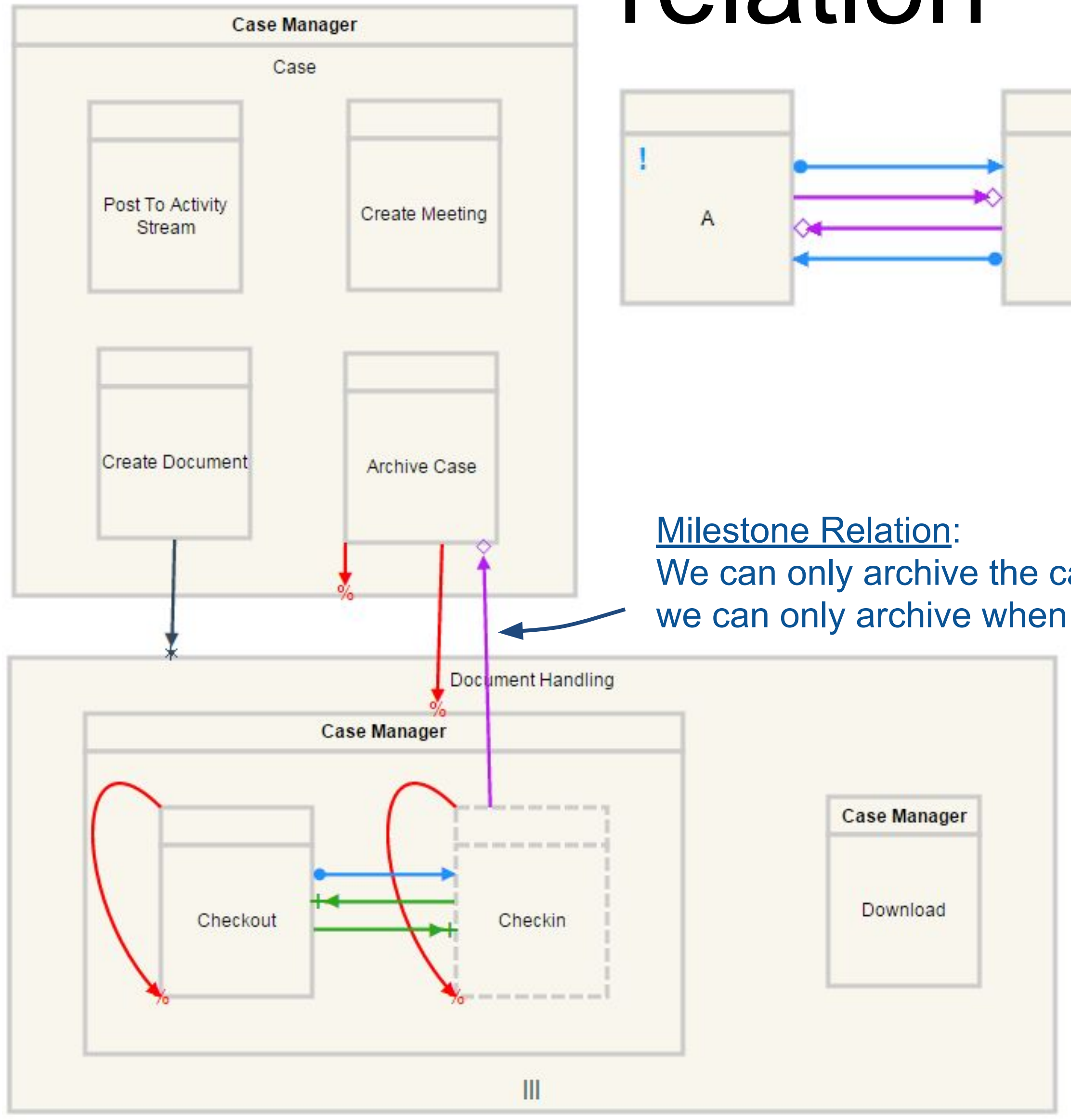
ECM Example - Milestone relation

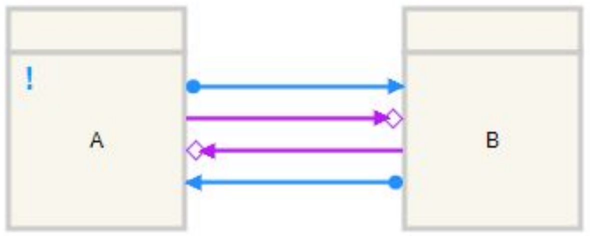


We would like to start a document handling sub-process for each time we create a document.

Milestone Relation:
We can only archive the case while Checkin is **not pending**, i.e.: we can only archive when all document have been checked in

ECM Example - Milestone relation

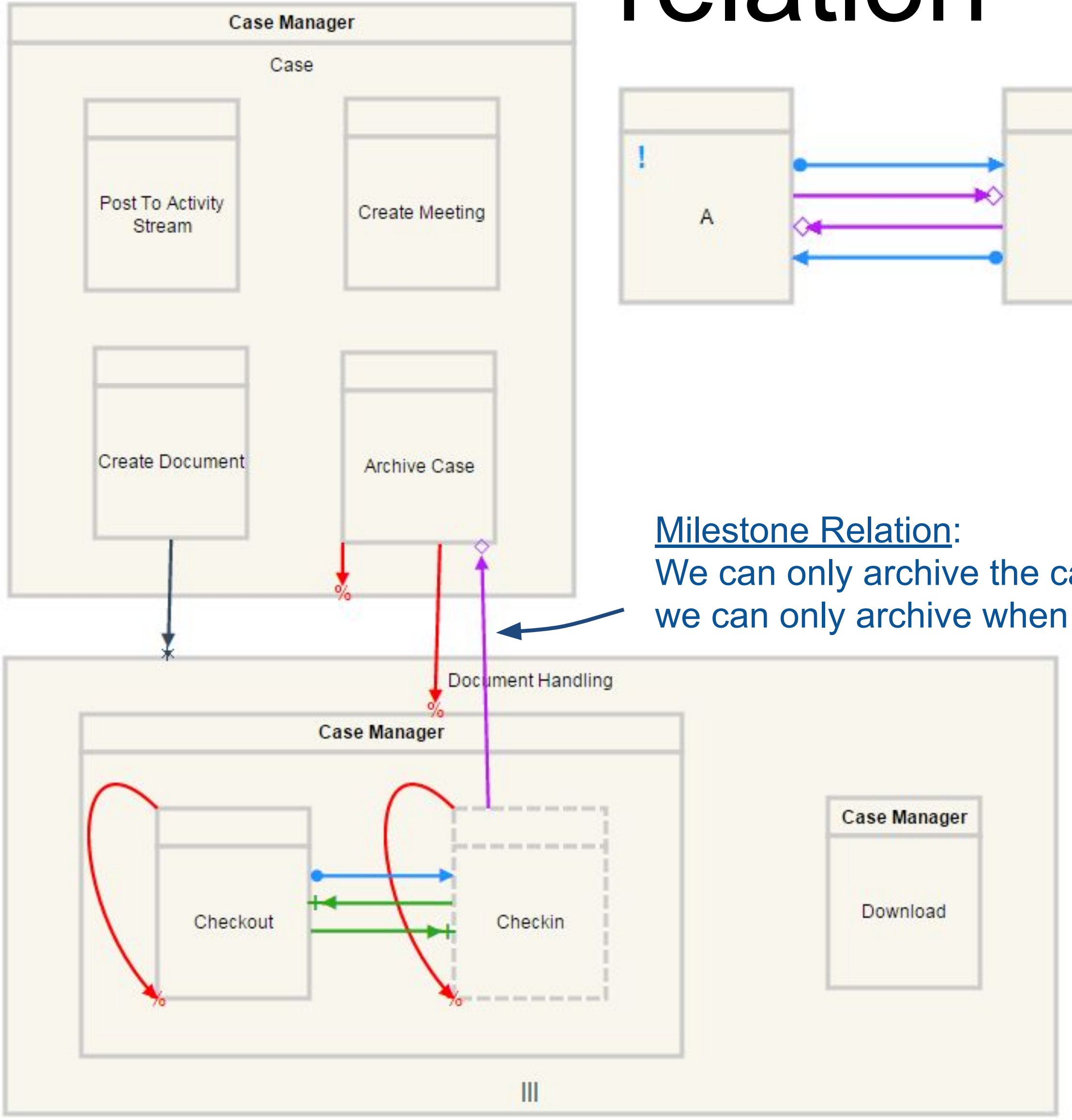


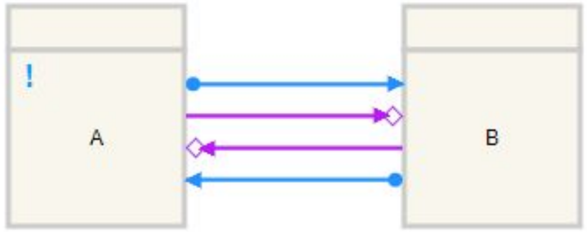


Question: What language does this model capture?

Milestone Relation:
We can only archive the case while Checkin is **not pending**, i.e.: we can only archive when all document have been checked in

ECM Example - Milestone relation



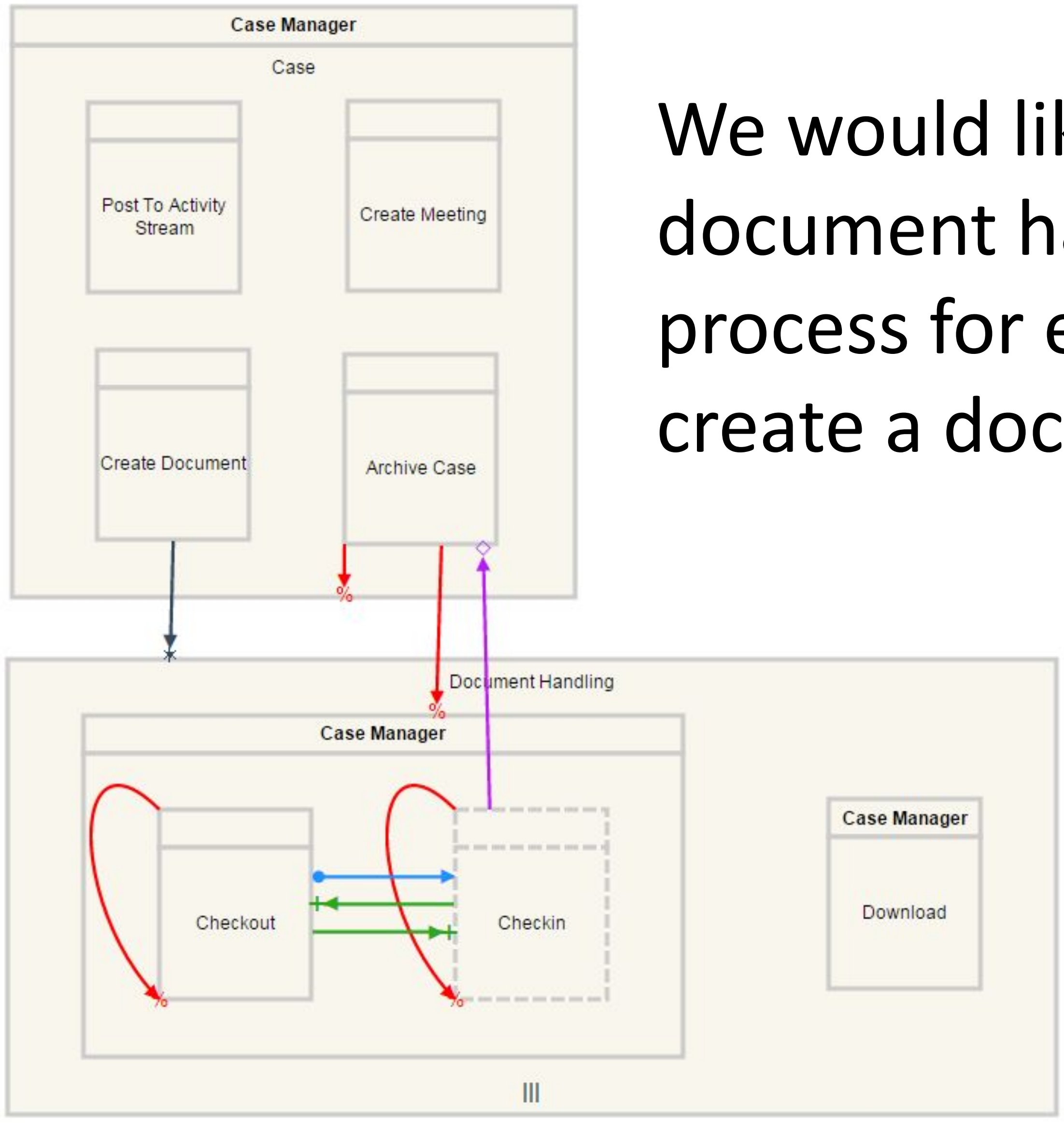


Question: What language does this model capture?

{<A,B,A,B,....>}

Milestone Relation:
We can only archive the case while Checkin is **not pending**, i.e.: we can only archive when all document have been checked in

ECM Example - Subprocesses



We would like to start a document handling sub-process for each time we create a document.

DEMO

DCR Graphs in Industry

- **2011**: Adopted by Exformatics to enable process support in their case management solution.
- **2014**: Development of stand-alone modelling and simulation tools, leading to the founding of a new company (2018)
- **2017**: Adoption by several new Danish partners. In particular: integration of the DCR engine in KMDs case management system, ran by 70% of Danish central government institutions.

How can I use DCR Graphs?

- Academic tool: http://dcr.itu.dk/
  - Pros: Most features for advanced users
  - Cons: Not user-friendly for regular users

- Commercial tool: http://www.dcrgraphs.net/
  - Pros: Made for regular users, graphical editor, extensive support for collaboration with other users, actively supported with regular updates.
  - Cons: Less advanced features

Both are **free** for academic use.

Overview

- Process Modelling
- Imperative vs Declarative Process Models
- Dynamic Condition Response (DCR) Graphs
- Hierarchy in DCR GRaphs
- Semantics of DCR Graphs
- **Assignment 1.1**

Assignment 1

Part 1: Modelling Event Patterns as DCR Graphs

Model the following patterns as DCR Graphs, based on the Dreyers log introduced and examined in the paper The Analysis of a Real Life Declarative Process:

1) Fill out application should always be the first event of the case.

2) Reject should always eventually be followed by Applicant informed and Change phase to Abort.

3) First payment should only occur once in every case.

4) Lawyer Review and Architect Review should never occur in the same case.

Assignment 1

Hints for Part 1:

1) You do not need to model the patterns together in one graph, i.e. you will have 4 graphs, 1 for each pattern.

2) To find the names of all activities in the process, look at the *Title* column in the log.

Questions?