Ontologies & Business Process modeling languages: two proposals for a fruitful pairing

Chiara Ghidini

Process & Data Intelligence, FBK-irst, Trento, Italy

Extensive credits to Marco Montali and Marco Rospocher

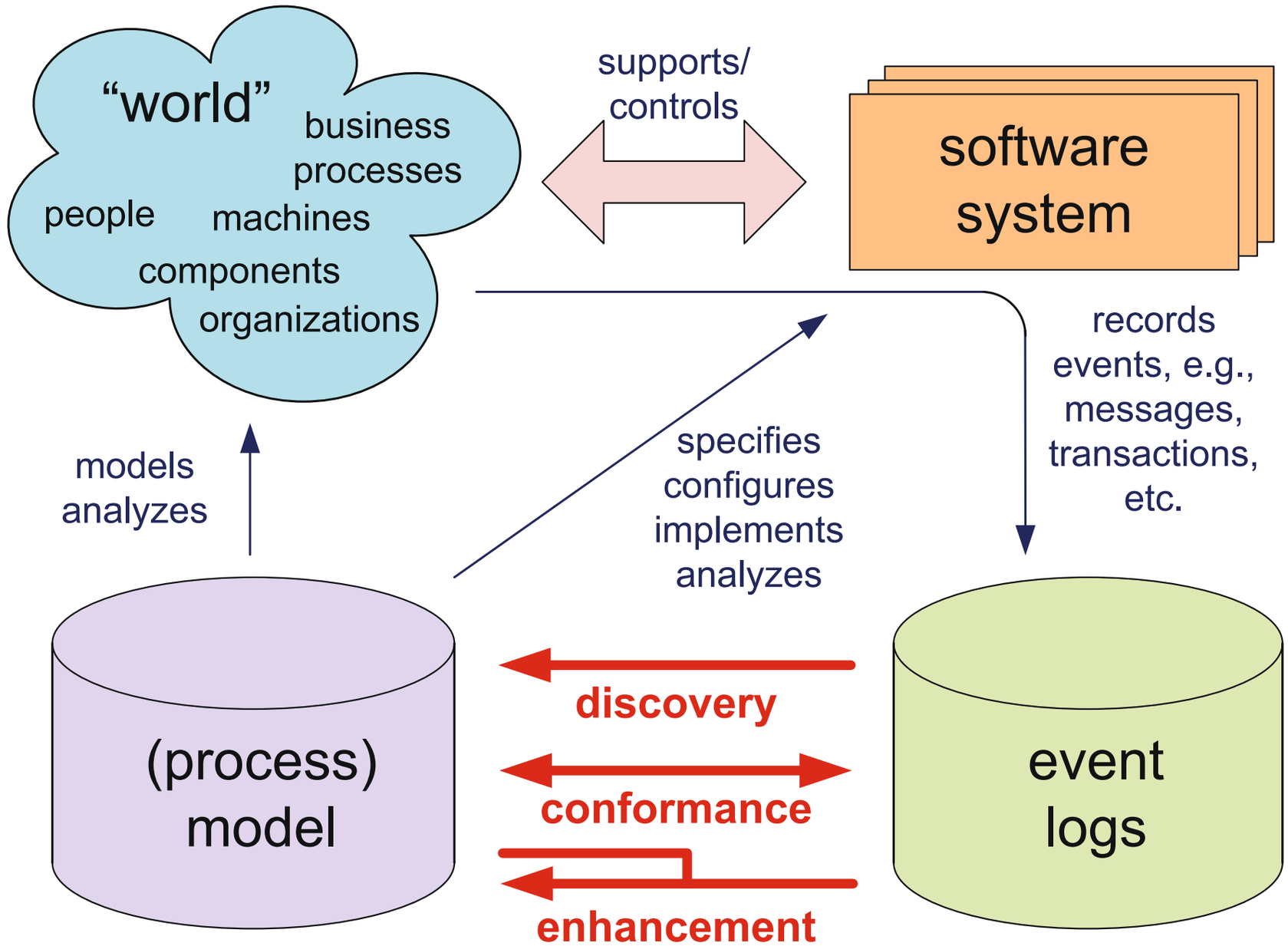
Outline

- Context and Motivations
- Two approaches (at least):
    1. ontologies to describe model diagrams
        - The **BPMN ontology**
        - and its applications
    2. ontological analysis to refine the semantics of model diagrams
        - (preliminary) ongoing work

Outline

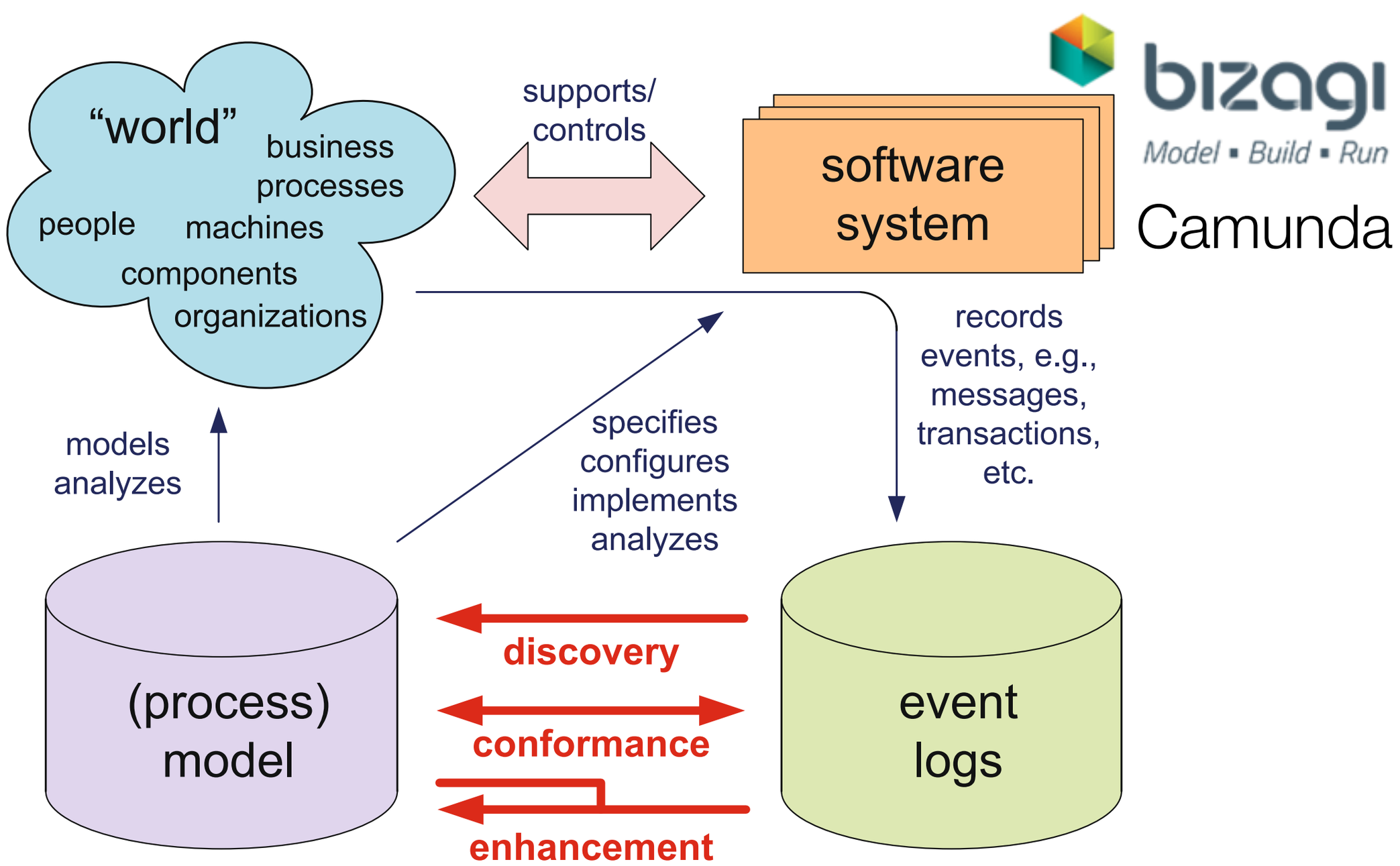
- Context and Motivations
- Two approaches (at least):
    1. ontologies to describe model diagrams
        - The **BPMN ontology**
        - and its applications
    2. ontological an
       model diagra
        - (prelimina

Disclaimer: I'm not an ontologist! I'm only a computer scientist :) nowadays working mainly on BPM



[Process Mining Manifesto]

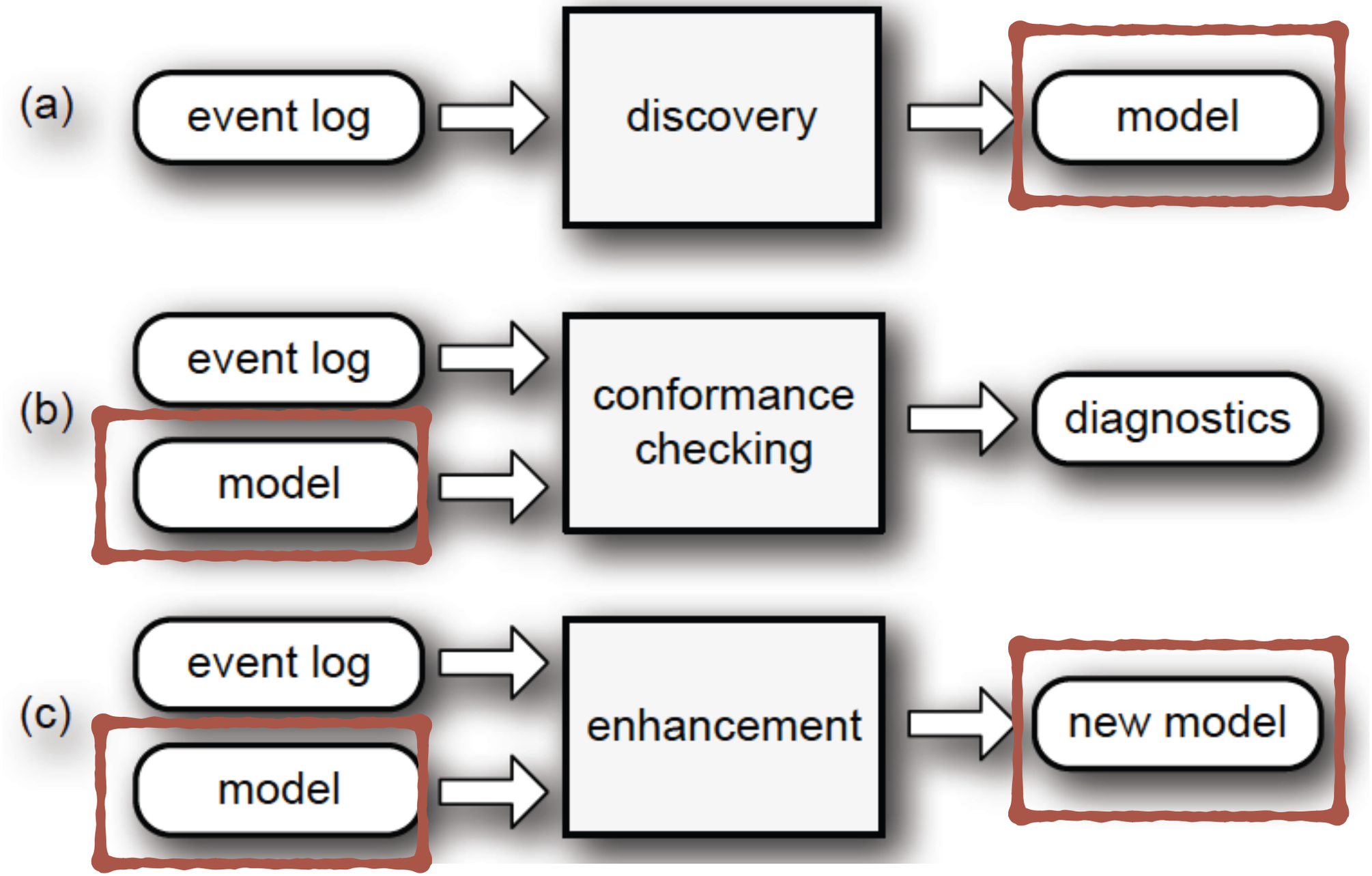
Processes everywhere | more than ever



bizagi
Model ▪ Build ▪ Run
Camunda

[Process Mining Manifesto]

Processes everywhere | more than ever

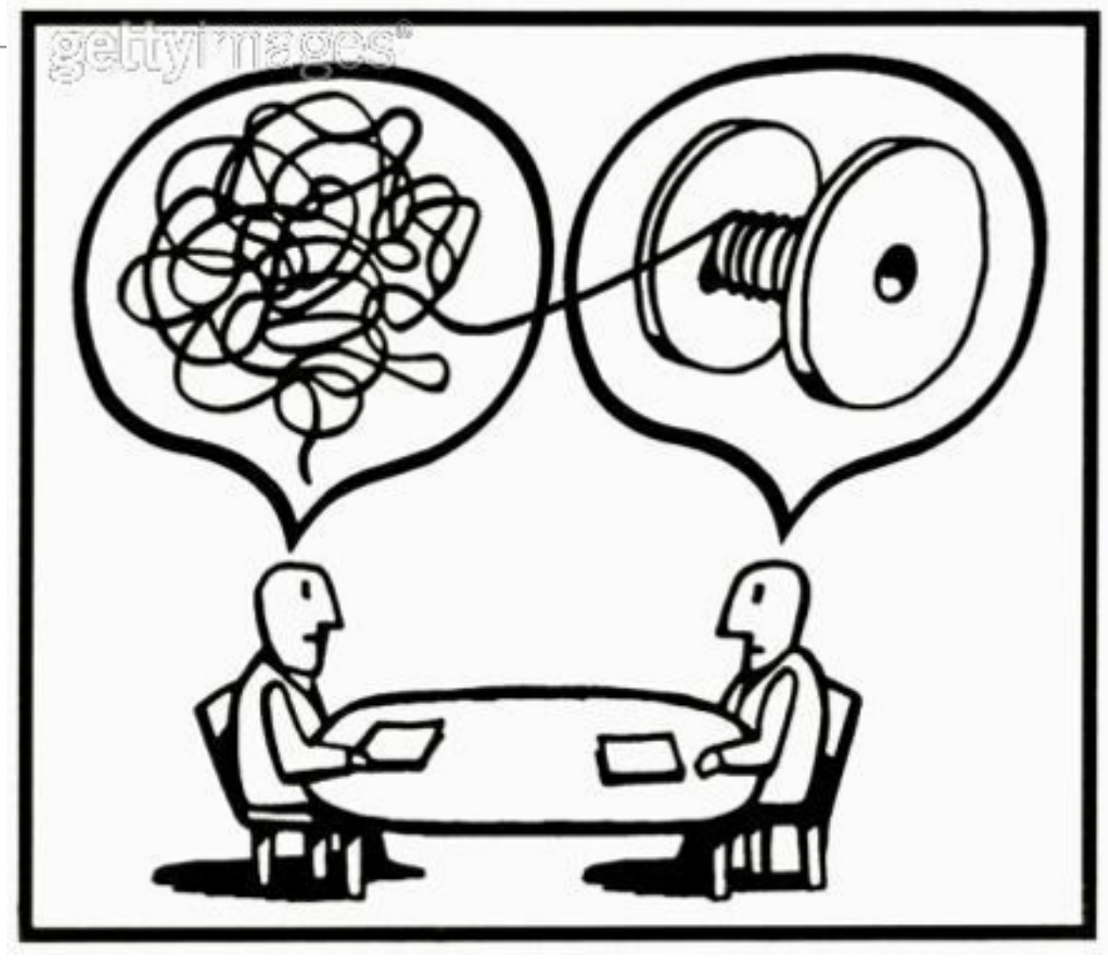


[Process Mining Manifesto]

Models everywhere

even in a mining manifesto!

Why (conceptual) Models?



The activity of **formally describing some** aspects of the **physical** and **social** world around us for the purposes of **understanding** and **communication**.

(John Mylopoulos, 1992)

Understanding and **communication**

Event Log

PurchasingExample.csv

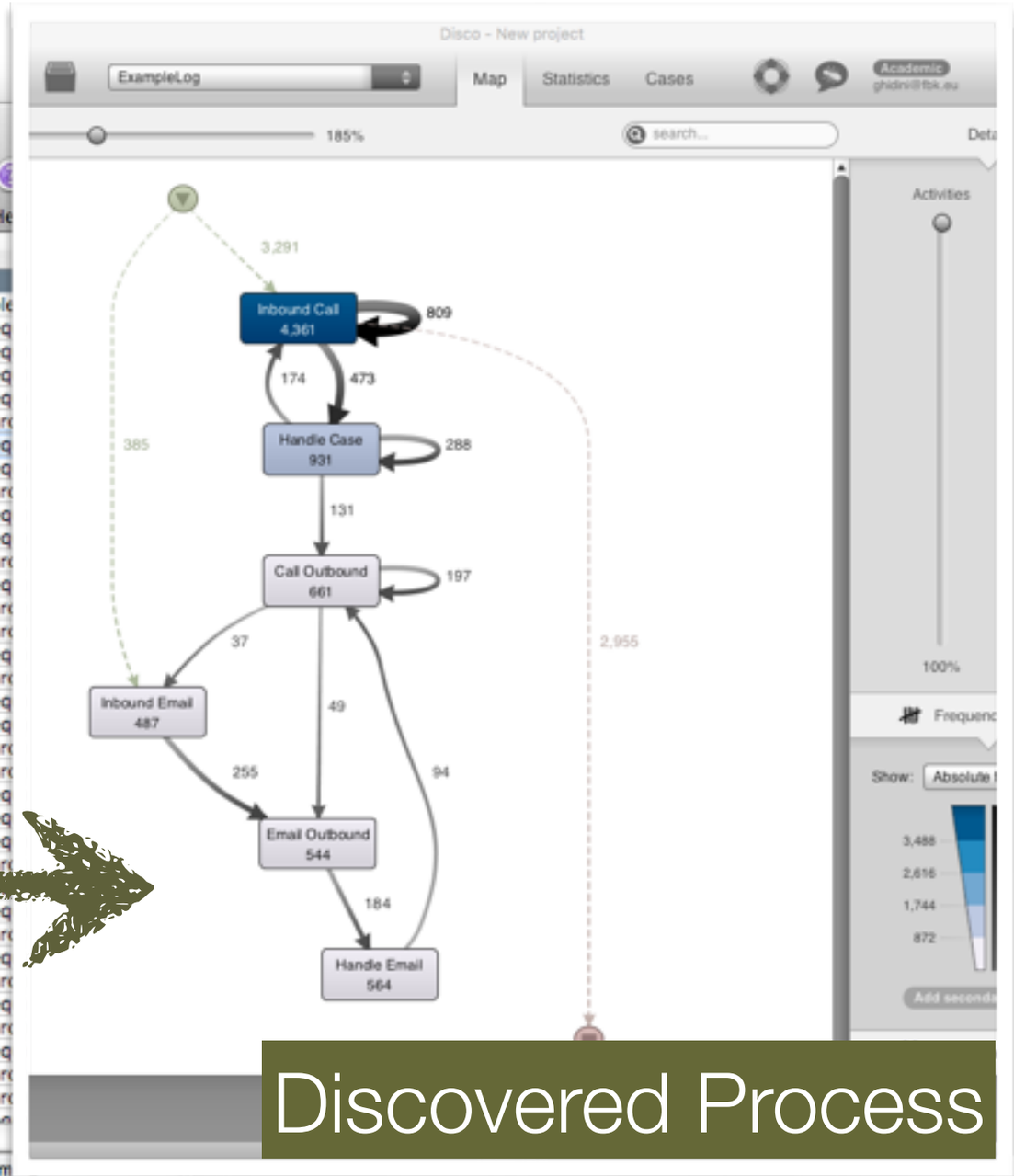
New Open Save Print Import Copy Paste Format Undo Redo AutoSum Sort A-Z Sort Z-A Gallery Toolbox Zoom

Sheets Charts SmartArt Graphics WordArt

	A	B	C	D	E
1	Case ID	Start Timestamp	Complete Timestamp	Activity	Resource
2	1	2011/01/01 00:00:00.000	2011/01/01 00:37:00.000	Create Purchase Requisition	Kim Passa
3	2	2011/01/01 00:16:00.000	2011/01/01 00:29:00.000	Create Purchase Requisition	Immanuel Karagianni
4	3	2011/01/01 02:23:00.000	2011/01/01 03:03:00.000	Create Purchase Requisition	Kim Passa
5	1	2011/01/01 05:37:00.000	2011/01/01 05:45:00.000	Create Request for Quotation	Kim Passa
6	1	2011/01/01 06:41:00.000	2011/01/01 06:55:00.000	Analyze Request for Quotation	Karel de Groot
7	2	2011/01/01 08:16:00.000	2011/01/01 08:26:00.000	Create Request for Quotation	Alberto Duport
8	4	2011/01/01 08:39:00.000	2011/01/01 09:00:00.000	Create Purchase Requisition	Fjodor Kowalski
9	2	2011/01/01 09:34:00.000	2011/01/01 09:38:00.000	Analyze Request for Quotation	Karel de Groot
10	5	2011/01/01 09:49:00.000	2011/01/01 10:35:00.000	Create Purchase Requisition	Esmana Liubiata
11	2	2011/01/01 10:16:00.000	2011/01/01 10:21:00.000	Amend Request for Quotation	Christian Francois
12	2	2011/01/01 11:15:00.000	2011/01/01 11:48:00.000	Analyze Request for Quotation	Magdalena Predutta
13	6	2011/01/01 11:20:00.000	2011/01/01 11:37:00.000	Create Purchase Requisition	Christian Francois
14	1	2011/01/01 11:43:00.000	2011/01/01 12:09:00.000	Send Request for Quotation to Supplier	Karel de Groot
15	1	2011/01/01 12:32:00.000	2011/01/01 16:03:00.000	Create Quotation comparison Map	Magdalena Predutta
16	2	2011/01/01 12:33:00.000	2011/01/01 12:39:00.000	Amend Request for Quotation	Esmana Liubiata
17	2	2011/01/01 13:28:00.000	2011/01/01 13:38:00.000	Analyze Request for Quotation	Karel de Groot
18	7	2011/01/01 14:05:00.000	2011/01/01 15:00:00.000	Create Purchase Requisition	Esmana Liubiata
19	8	2011/01/01 14:27:00.000	2011/01/01 15:17:00.000	Create Purchase Requisition	Fjodor Kowalski
20	2	2011/01/01 15:18:00.000	2011/01/01 15:40:00.000	Send Request for Quotation to Supplier	Francois de Perrier
21	2	2011/01/01 15:55:00.000	2011/01/01 16:43:00.000	Create Quotation comparison Map	Karel de Groot
22	9	2011/01/01 16:17:00.000	2011/01/01 16:34:00.000	Create Purchase Requisition	Tesca Lobes
23	6	2011/01/01 17:32:00.000	2011/01/01 17:45:00.000	Create Request for Quotation	Alberto Duport
24	8	2011/01/01 18:00:00.000	2011/01/01 18:07:00.000	Create Request for Quotation	Tesca Lobes
25	6	2011/01/01 18:39:00.000	2011/01/01 18:55:00.000	Analyze Request for Quotation	Magdalena Predutta
26	4	2011/01/01 18:45:00.000	2011/01/01 18:51:00.000	Analyze Purchase Requisition	Heinz Gutschmidt
27	4	2011/01/01 18:56:00.000	2011/01/01 18:58:00.000	Create Request for Quotation	Heinz Gutschmidt
28	8	2011/01/01 19:04:00.000	2011/01/01 19:27:00.000	Analyze Request for Quotation	Francois de Perrier
29	6	2011/01/01 19:47:00.000	2011/01/01 19:55:00.000	Amend Request for Quotation	Penn Osterwalder
30	4	2011/01/01 19:58:00.000	2011/01/01 20:19:00.000	Analyze Request for Quotation	Francois de Perrier
31	8	2011/01/01 20:21:00.000	2011/01/01 20:34:00.000	Amend Request for Quotation	Tesca Lobes
32	6	2011/01/01 20:55:00.000	2011/01/01 21:28:00.000	Analyze Request for Quotation	Francois de Perrier
33	4	2011/01/01 21:06:00.000	2011/01/01 21:14:00.000	Amend Request for Quotation	Nico Ojenbeer
34	8	2011/01/01 21:35:00.000	2011/01/01 22:01:00.000	Analyze Request for Quotation	Magdalena Predutta
35	4	2011/01/01 22:01:00.000	2011/01/01 22:24:00.000	Analyze Request for Quotation	Francois de Perrier

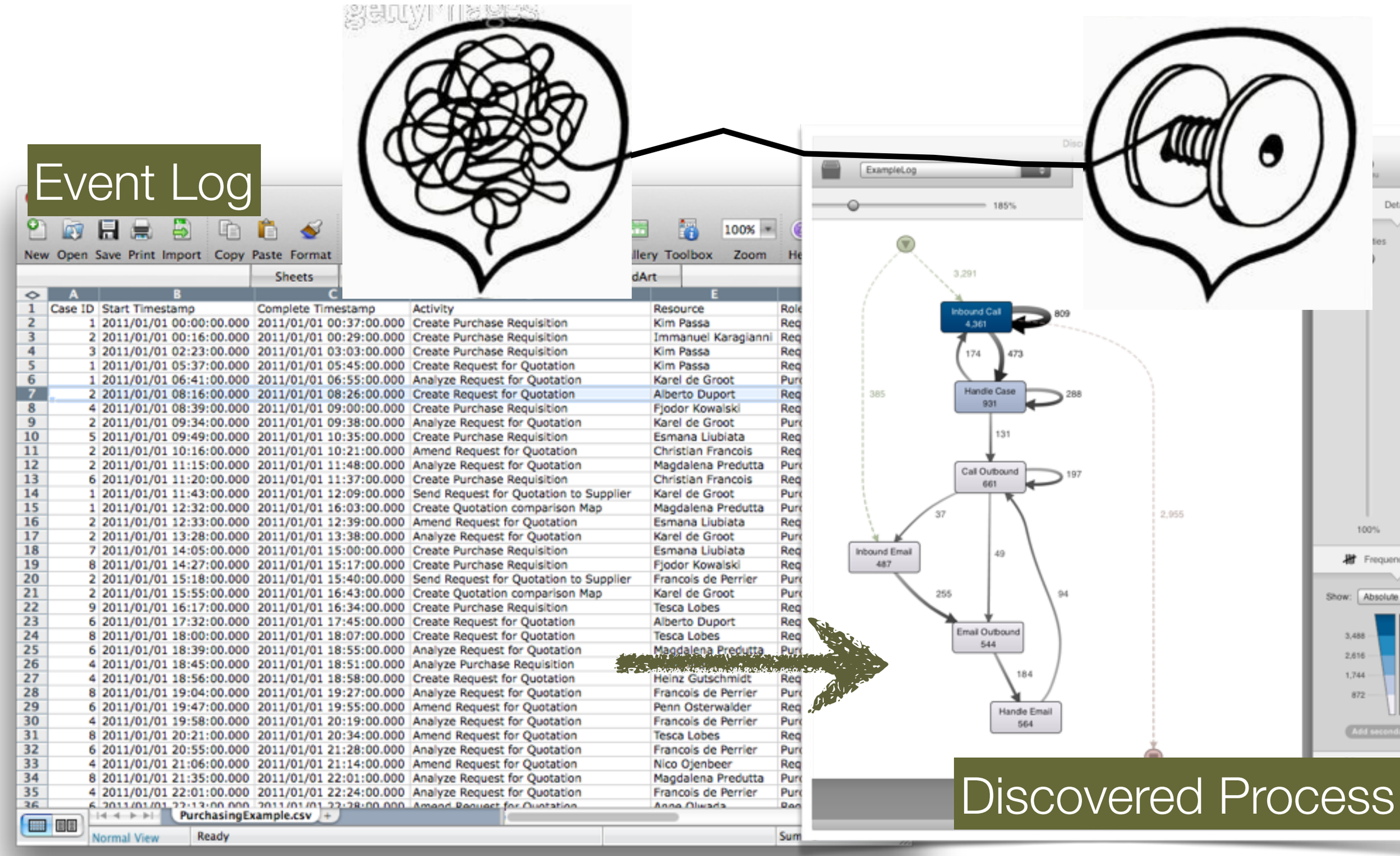
PurchasingExample.csv

Normal View Ready



Discovered Process

Understanding and **communication**

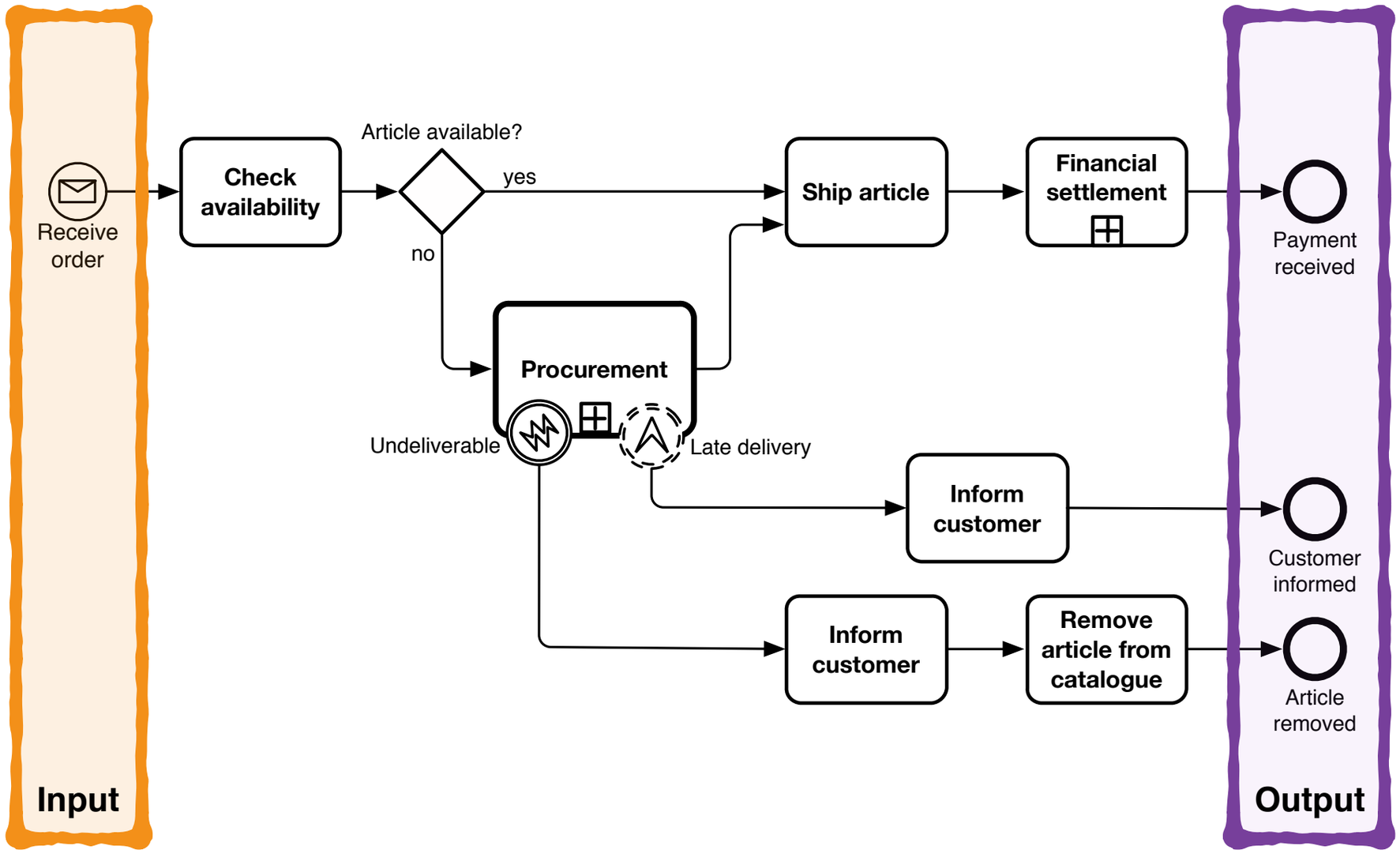



GOOD JOB!
Models everywhere!

Effect: Proliferation of BPM languages

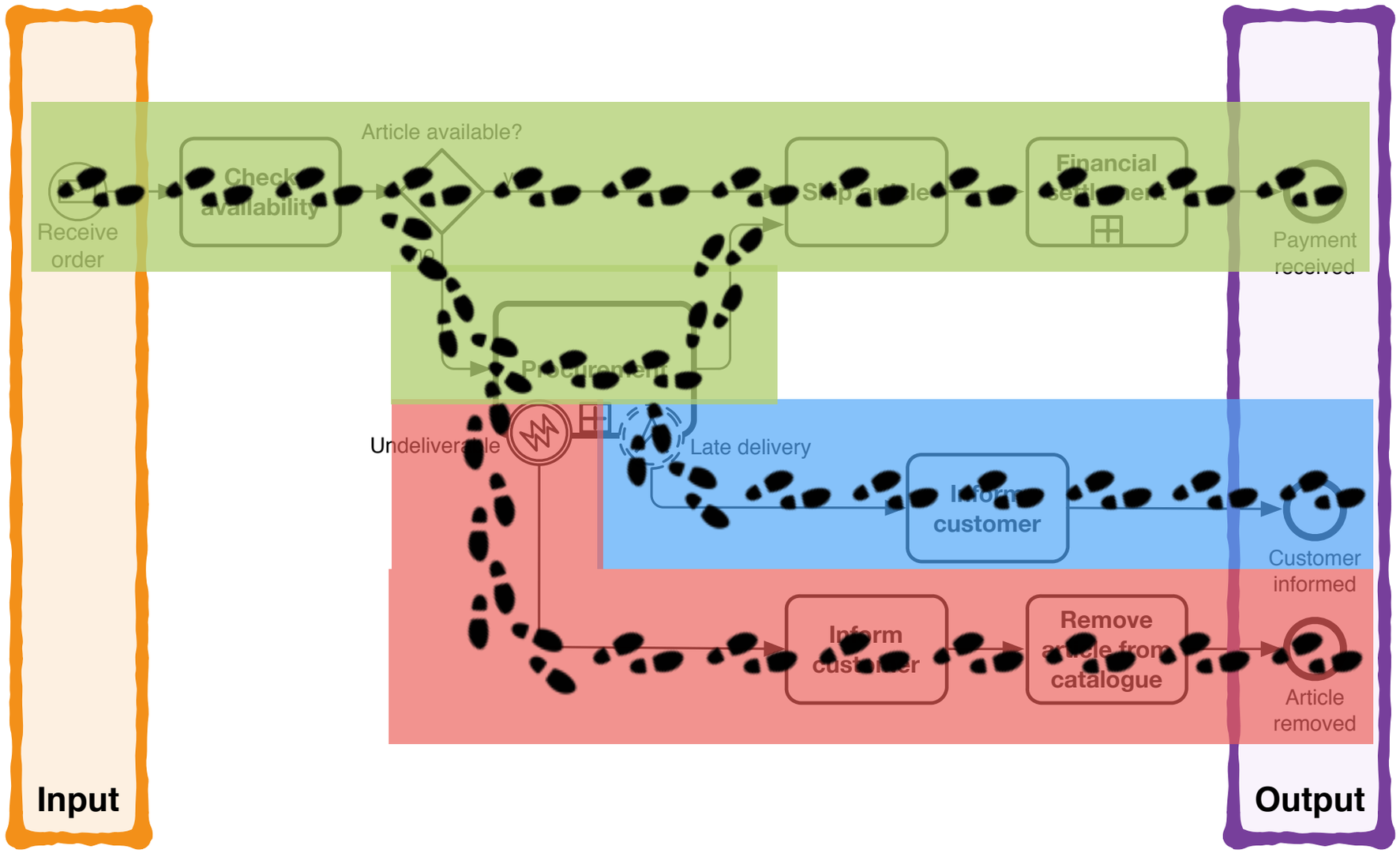
Effect: Proliferation of BPM languages

1. **Imperative** vs Declarative



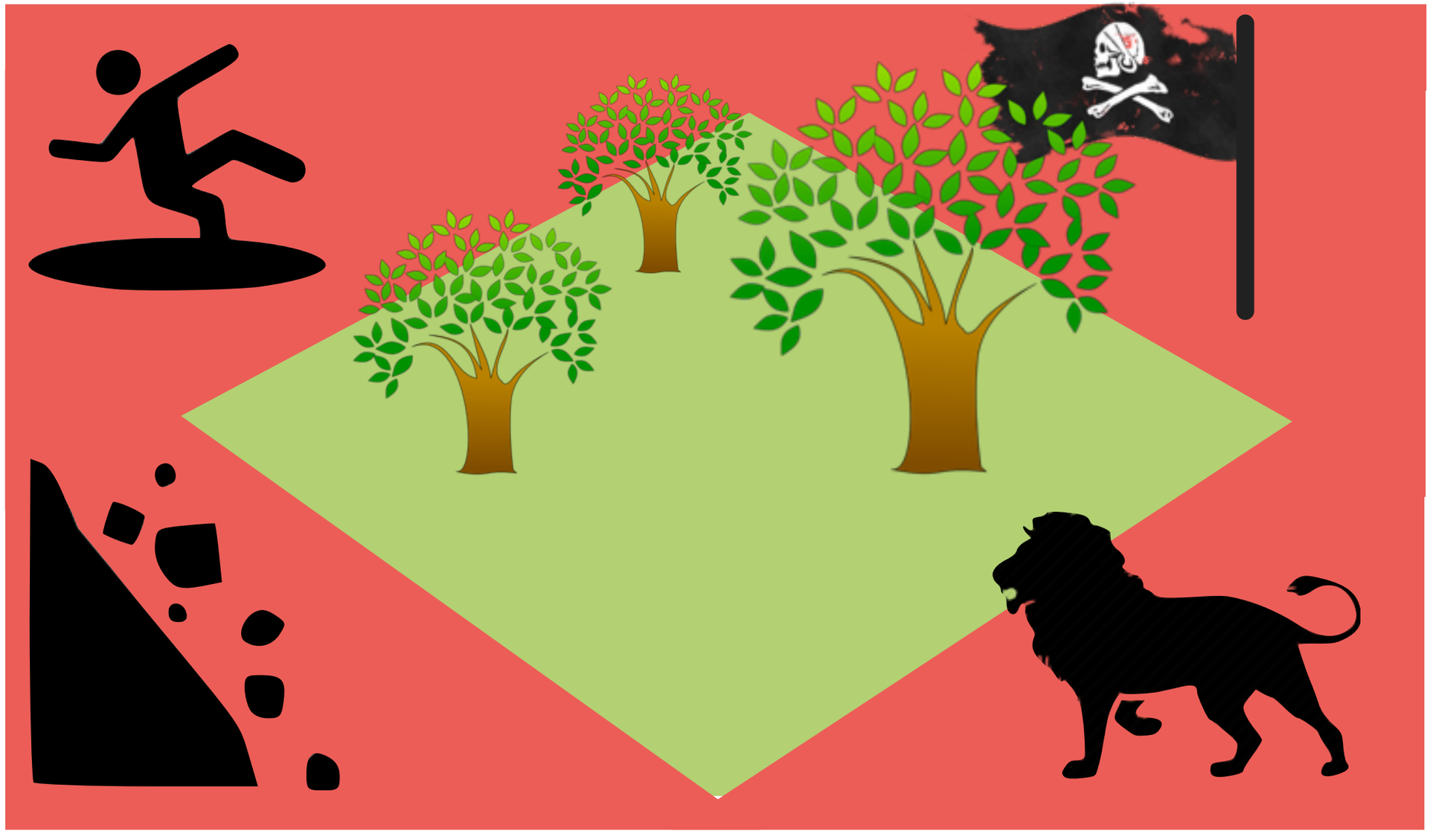
Effect: Proliferation of BPM languages

1. **Imperative** vs Declarative



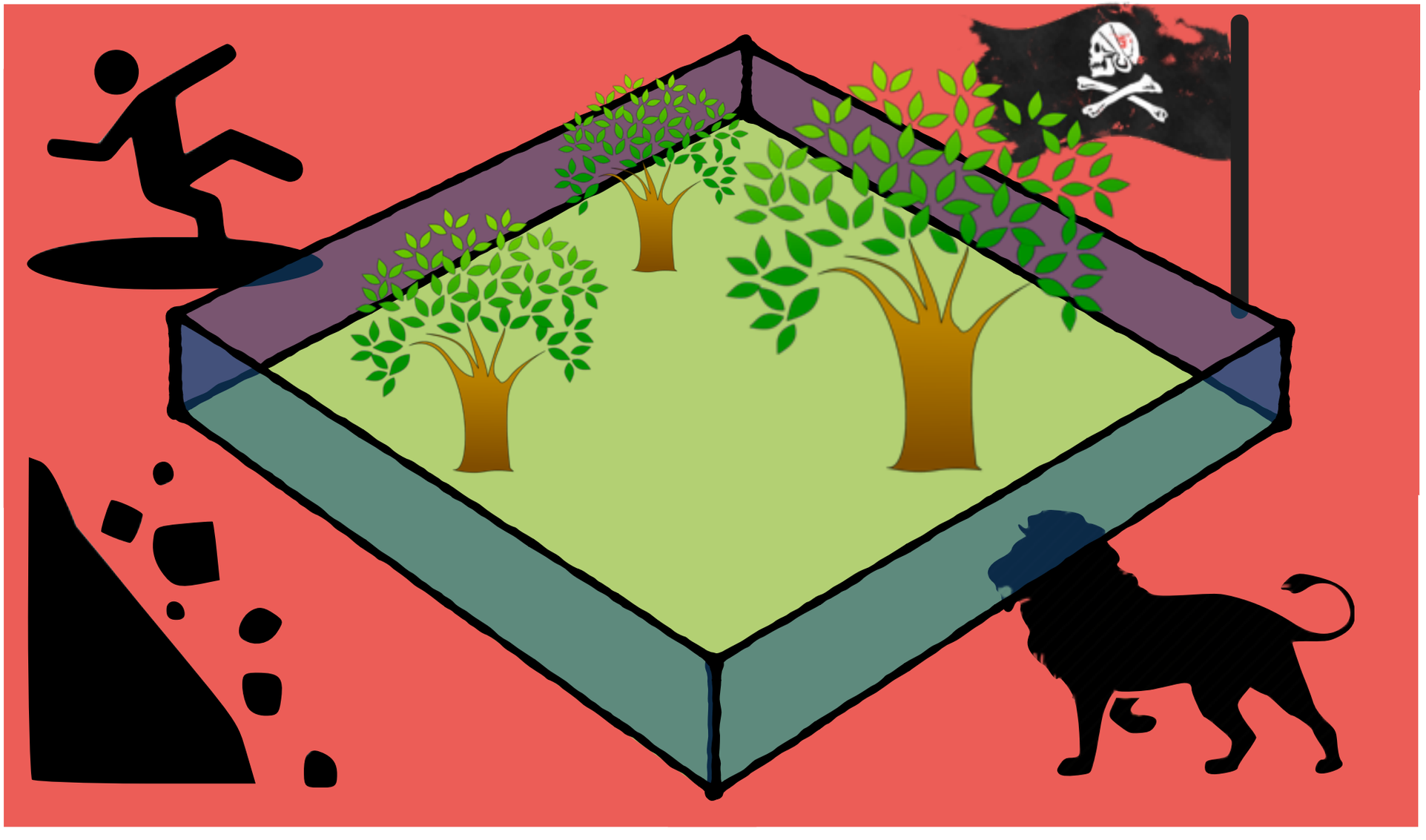
Effect: Proliferation of BPM languages

1. Imperative vs **Declarative**



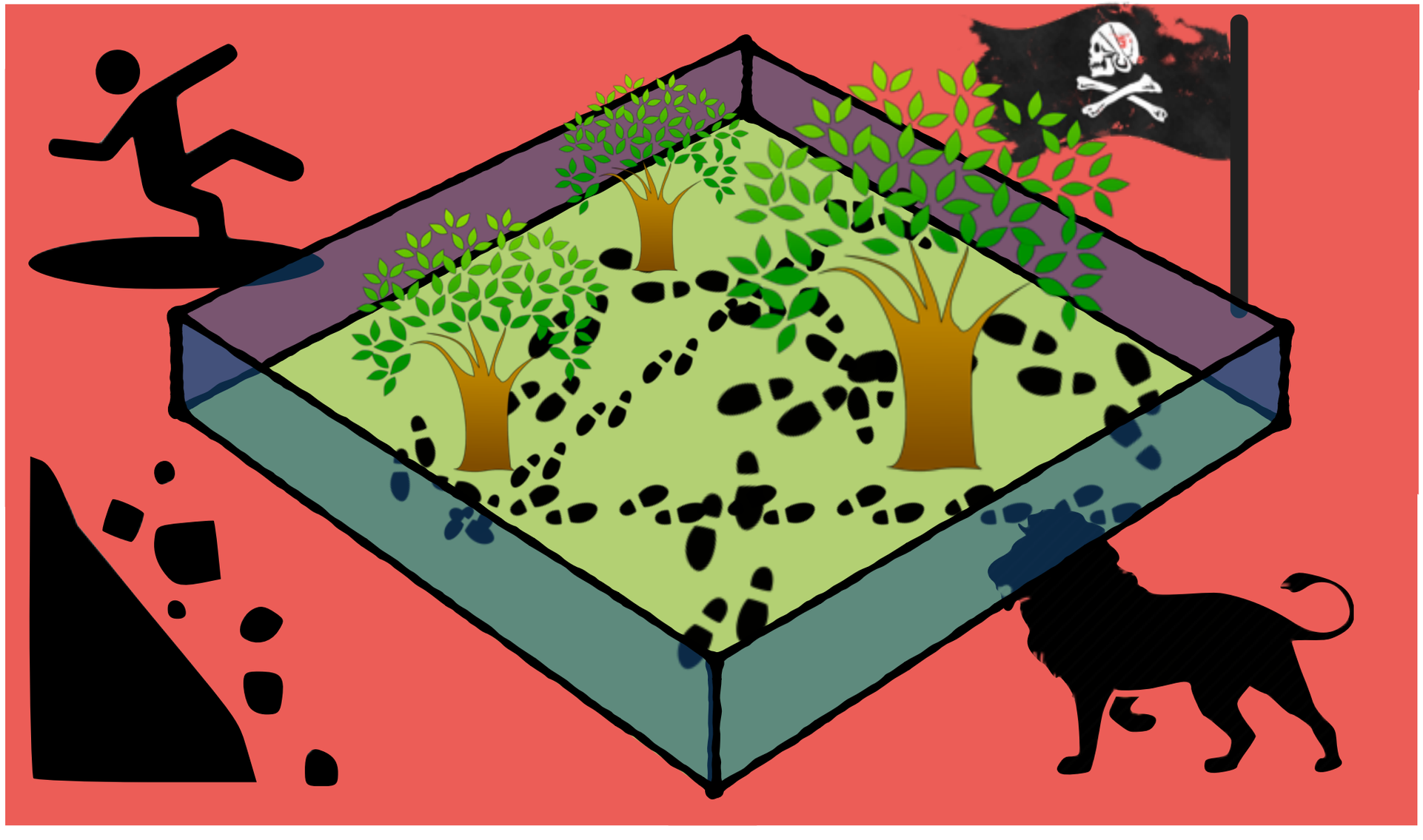
Effect: Proliferation of BPM languages

1. Imperative vs **Declarative**



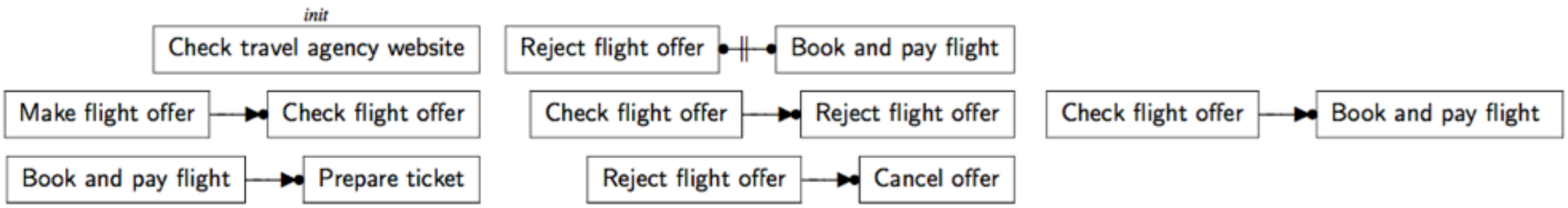
Effect: Proliferation of BPM languages

1. Imperative vs **Declarative**



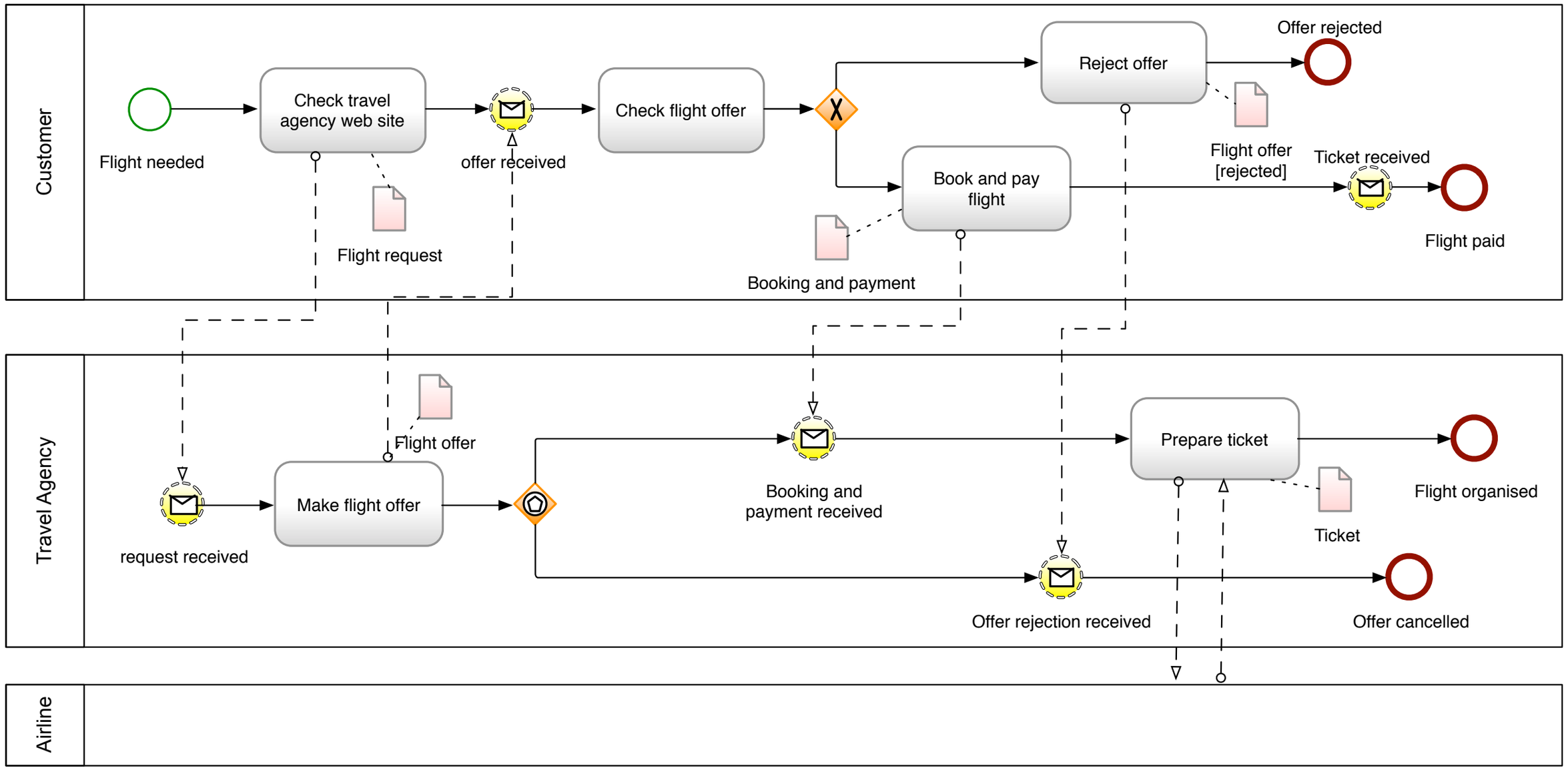
Effect: Proliferation of BPM languages

1. Imperative vs **Declarative**



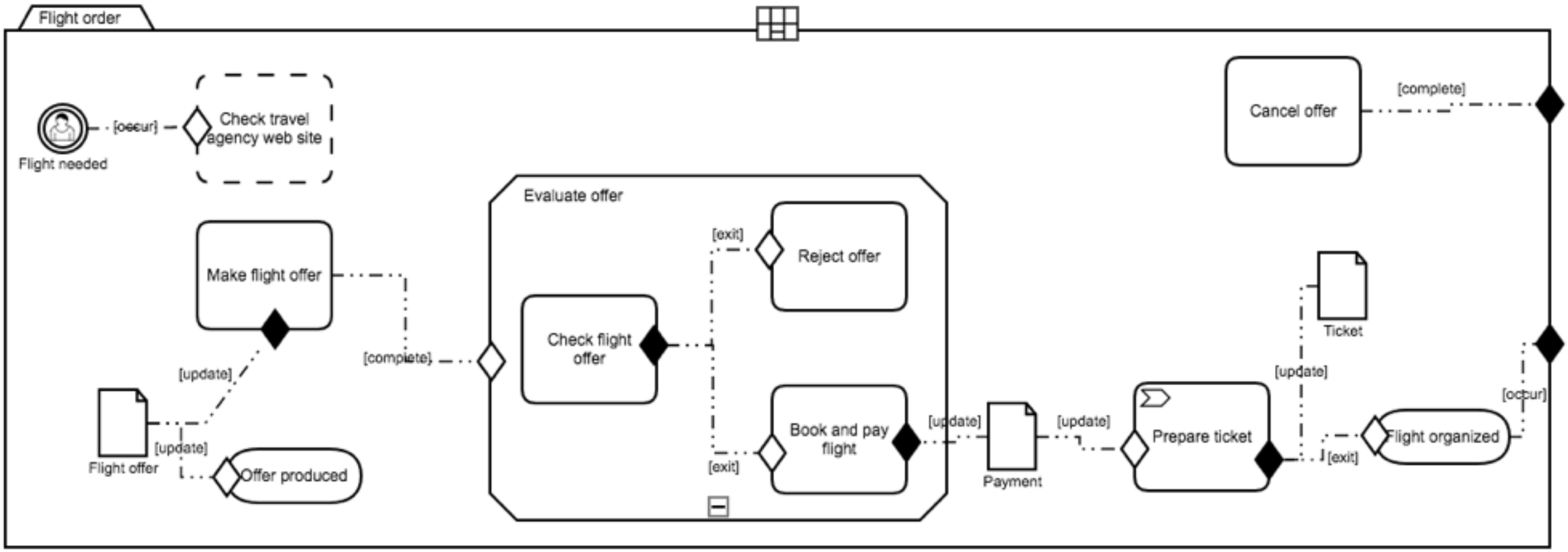
Effect: Proliferation of BPM languages

2. **Activity**- vs Data- centric



Effect: Proliferation of BPM languages

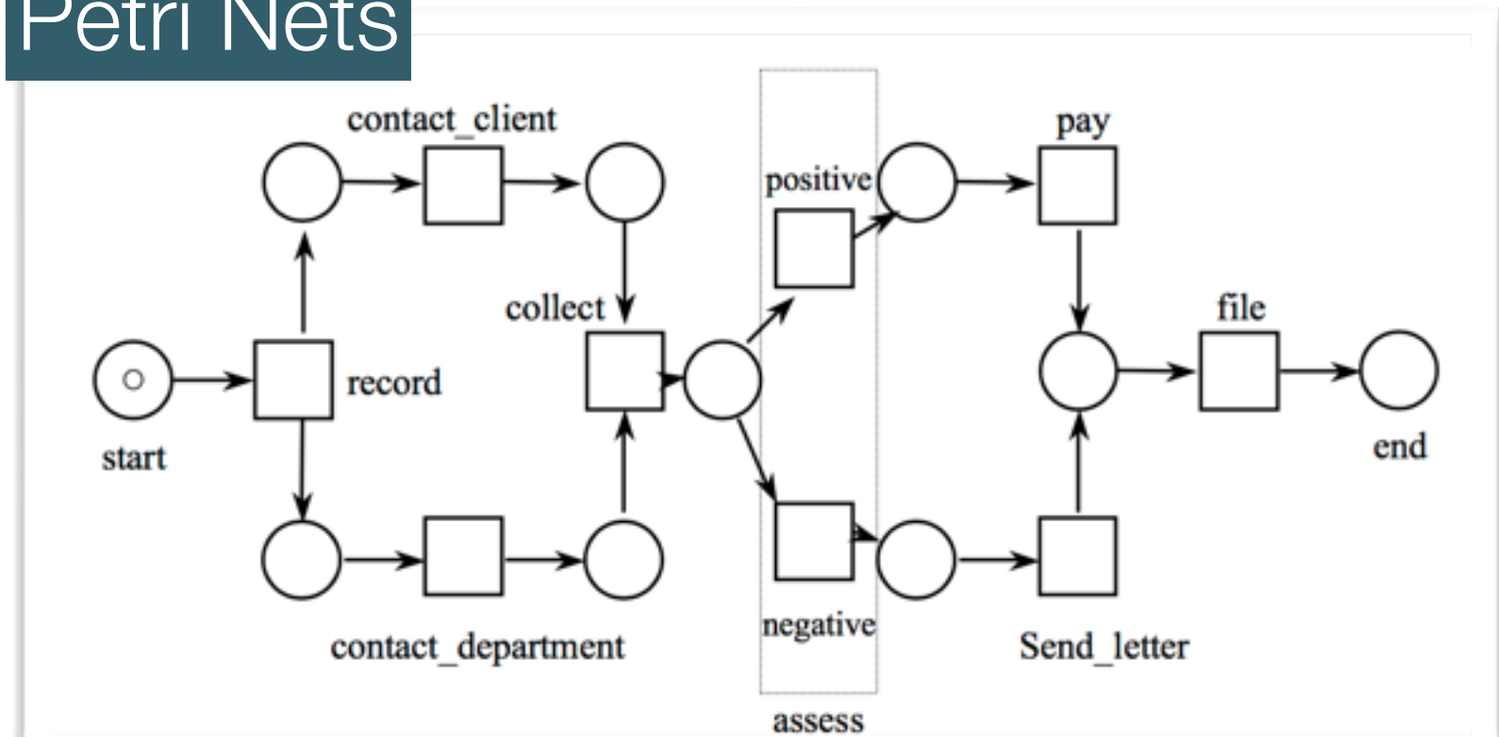
2. Activity- vs **Data**- centric



Effect: Proliferation of BPM languages

3. **Formal semantics** vs informal notations

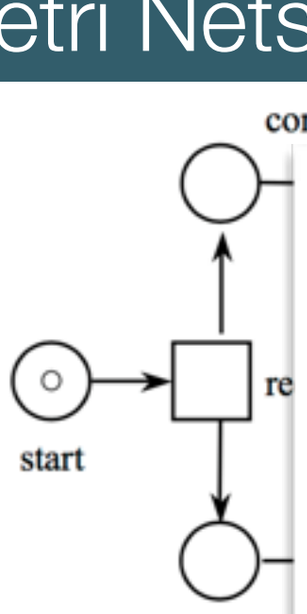
Petri Nets



Effect: Proliferation of BPM languages

3. **Formal semantics** vs informal notations

Petri Nets

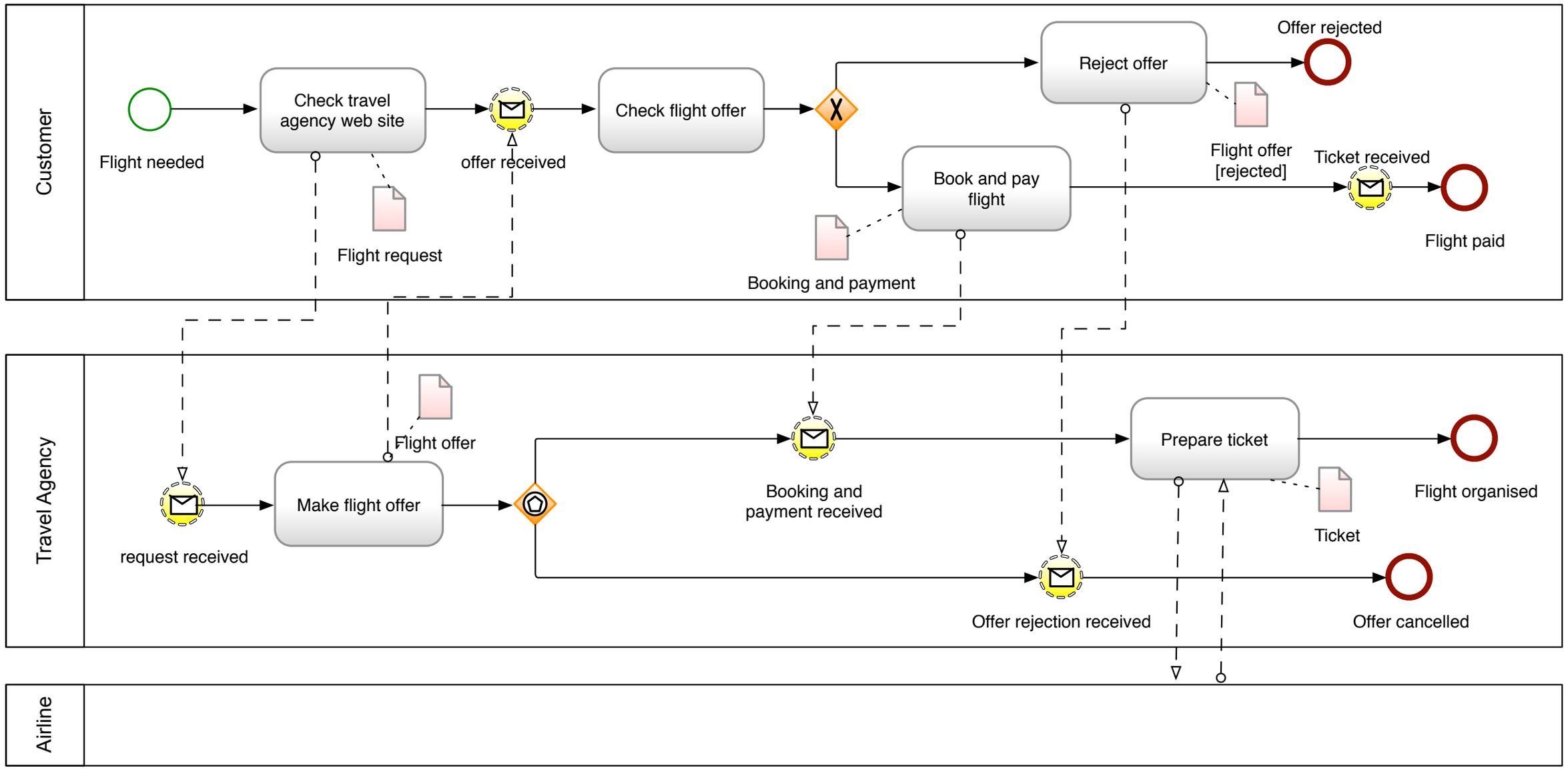


TEMPLATE	FORMALIZATION	NOTATION	DESCRIPTION
existence(A)	$\Diamond A$	1..* A	A has to occur at least once
absence(A)	$\neg\Diamond A$	0 A	A has to never occur
exactly1(A)	$\Diamond A \wedge \neg(A \wedge \bigcirc(\Diamond A))$	1 A	A has to occur exactly once
coexistence(A,B)	$(\Diamond A \rightarrow \Diamond B) \wedge \neg(\Diamond B \rightarrow \Diamond A)$	A ● — ● B	If A occurs, B must occur and viceversa
not coexistence(A,B)	$(\Diamond A \rightarrow \neg\Diamond B)$	A ● ‖ ● B	If A occurs, B must not occur and viceversa
response(A,B)	$\Box(A \rightarrow \Diamond B)$	A ● → ● B	If A occurs, B must eventually follow
chain response(A,B)	$\Box(A \rightarrow \bigcirc B)$	A ● ⇒ ● B	If A occurs, B must occur next

Declare templates

Effect: Proliferation of BPM languages

3. Formal semantics vs **informal notations**



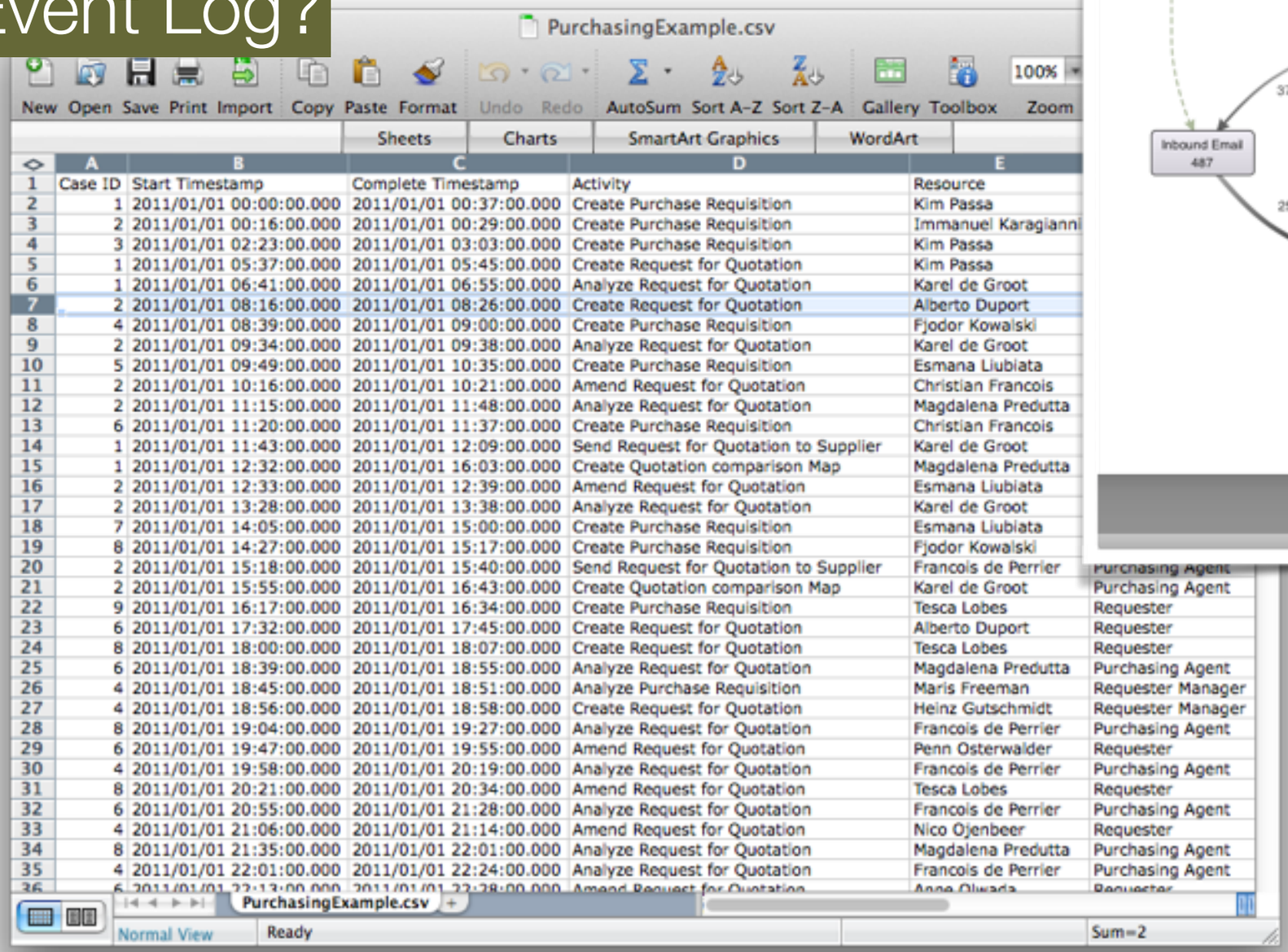
Good job!

- Being able to choose is good
- ..... but.....

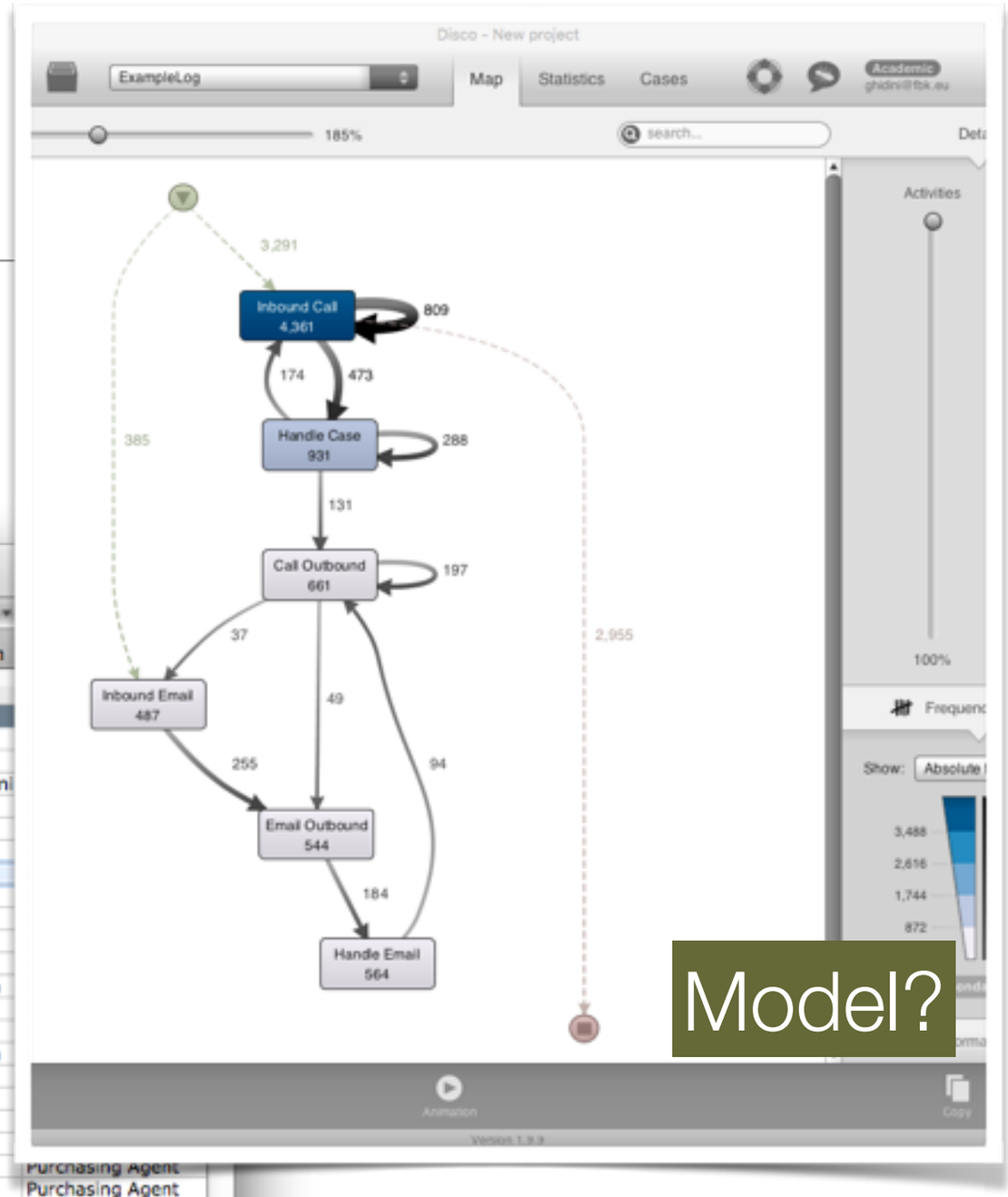
Problems!

- What is a process?

Event Log?



	A	B	C	D	E
1	Case ID	Start Timestamp	Complete Timestamp	Activity	Resource
2	1	2011/01/01 00:00:00.000	2011/01/01 00:37:00.000	Create Purchase Requisition	Kim Passa
3	2	2011/01/01 00:16:00.000	2011/01/01 00:29:00.000	Create Purchase Requisition	Immanuel Karagianni
4	3	2011/01/01 02:23:00.000	2011/01/01 03:03:00.000	Create Purchase Requisition	Kim Passa
5	1	2011/01/01 05:37:00.000	2011/01/01 05:45:00.000	Create Request for Quotation	Kim Passa
6	1	2011/01/01 06:41:00.000	2011/01/01 06:55:00.000	Analyze Request for Quotation	Karel de Groot
7	2	2011/01/01 08:16:00.000	2011/01/01 08:26:00.000	Create Request for Quotation	Alberto Duport
8	4	2011/01/01 08:39:00.000	2011/01/01 09:00:00.000	Create Purchase Requisition	Fjodor Kowalski
9	2	2011/01/01 09:34:00.000	2011/01/01 09:38:00.000	Analyze Request for Quotation	Karel de Groot
10	5	2011/01/01 09:49:00.000	2011/01/01 10:35:00.000	Create Purchase Requisition	Esmana Liubiata
11	2	2011/01/01 10:16:00.000	2011/01/01 10:21:00.000	Amend Request for Quotation	Christian Francois
12	2	2011/01/01 11:15:00.000	2011/01/01 11:48:00.000	Analyze Request for Quotation	Magdalena Predutta
13	6	2011/01/01 11:20:00.000	2011/01/01 11:37:00.000	Create Purchase Requisition	Christian Francois
14	1	2011/01/01 11:43:00.000	2011/01/01 12:09:00.000	Send Request for Quotation to Supplier	Karel de Groot
15	1	2011/01/01 12:32:00.000	2011/01/01 16:03:00.000	Create Quotation comparison Map	Magdalena Predutta
16	2	2011/01/01 12:33:00.000	2011/01/01 12:39:00.000	Amend Request for Quotation	Esmana Liubiata
17	2	2011/01/01 13:28:00.000	2011/01/01 13:38:00.000	Analyze Request for Quotation	Karel de Groot
18	7	2011/01/01 14:05:00.000	2011/01/01 15:00:00.000	Create Purchase Requisition	Esmana Liubiata
19	8	2011/01/01 14:27:00.000	2011/01/01 15:17:00.000	Create Purchase Requisition	Fjodor Kowalski
20	2	2011/01/01 15:18:00.000	2011/01/01 15:40:00.000	Send Request for Quotation to Supplier	Francois de Perrier
21	2	2011/01/01 15:55:00.000	2011/01/01 16:43:00.000	Create Quotation comparison Map	Karel de Groot
22	9	2011/01/01 16:17:00.000	2011/01/01 16:34:00.000	Create Purchase Requisition	Tesca Lobes
23	6	2011/01/01 17:32:00.000	2011/01/01 17:45:00.000	Create Request for Quotation	Alberto Duport
24	8	2011/01/01 18:00:00.000	2011/01/01 18:07:00.000	Create Request for Quotation	Tesca Lobes
25	6	2011/01/01 18:39:00.000	2011/01/01 18:55:00.000	Analyze Request for Quotation	Magdalena Predutta
26	4	2011/01/01 18:45:00.000	2011/01/01 18:51:00.000	Analyze Purchase Requisition	Maris Freeman
27	4	2011/01/01 18:56:00.000	2011/01/01 18:58:00.000	Create Request for Quotation	Heinz Gutschmidt
28	8	2011/01/01 19:04:00.000	2011/01/01 19:27:00.000	Analyze Request for Quotation	Francois de Perrier
29	6	2011/01/01 19:47:00.000	2011/01/01 19:55:00.000	Amend Request for Quotation	Penn Osterwalder
30	4	2011/01/01 19:58:00.000	2011/01/01 20:19:00.000	Analyze Request for Quotation	Francois de Perrier
31	8	2011/01/01 20:21:00.000	2011/01/01 20:34:00.000	Amend Request for Quotation	Tesca Lobes
32	6	2011/01/01 20:55:00.000	2011/01/01 21:28:00.000	Analyze Request for Quotation	Francois de Perrier
33	4	2011/01/01 21:06:00.000	2011/01/01 21:14:00.000	Amend Request for Quotation	Nico Ojenbeer
34	8	2011/01/01 21:35:00.000	2011/01/01 22:01:00.000	Analyze Request for Quotation	Magdalena Predutta
35	4	2011/01/01 22:01:00.000	2011/01/01 22:24:00.000	Analyze Request for Quotation	Francois de Perrier

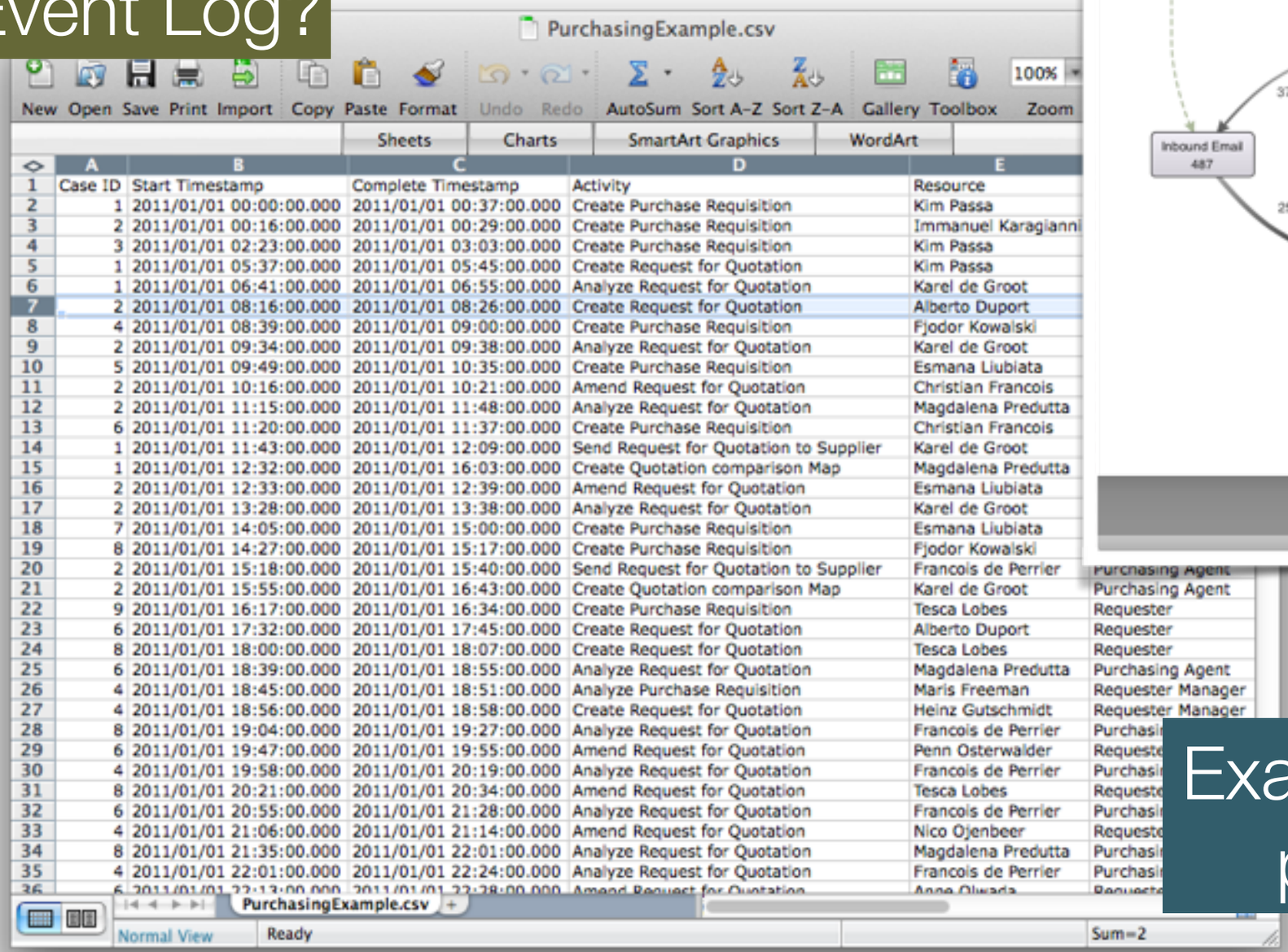


Model?

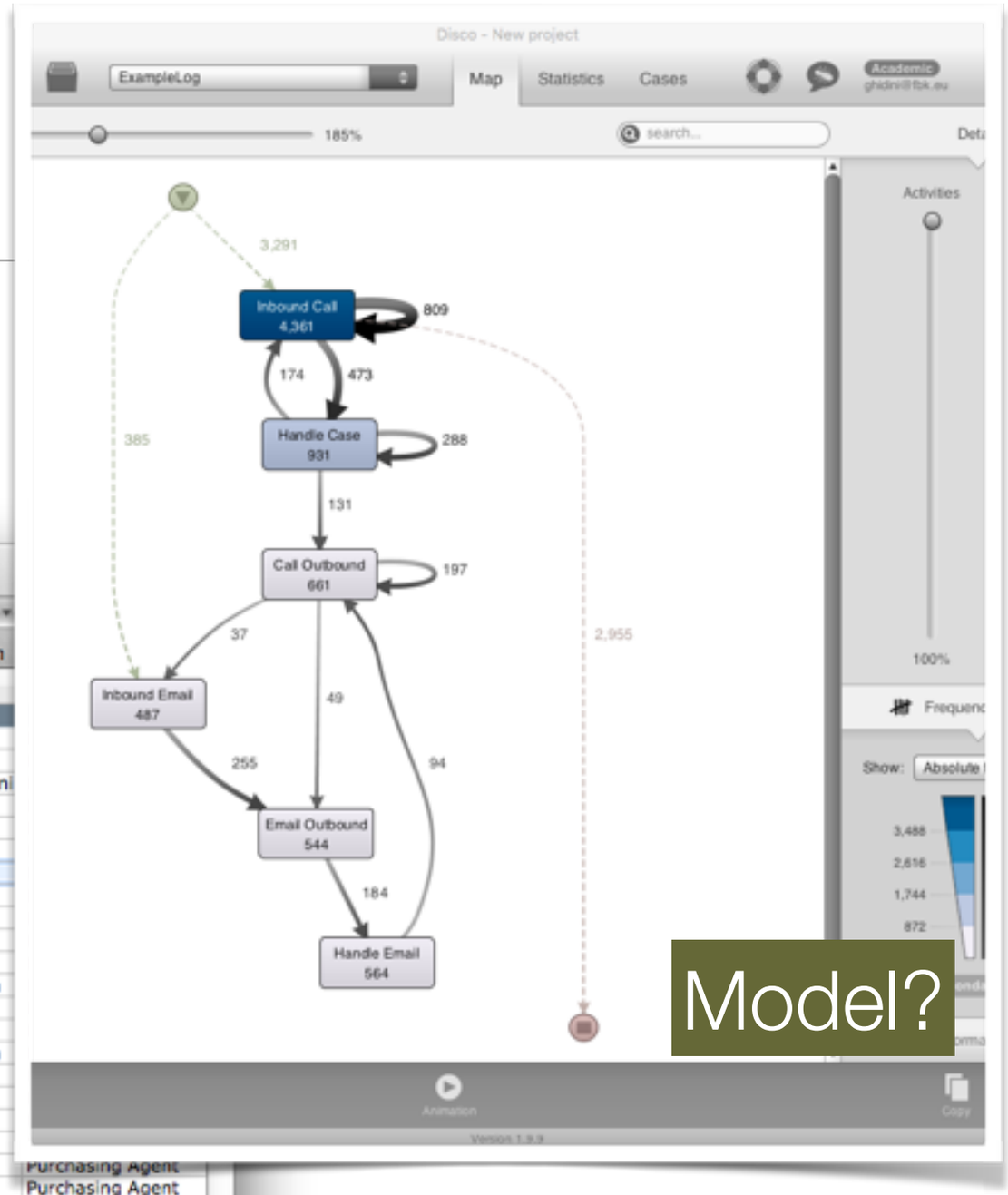
Problems!

- What is a process?

Event Log?



	A	B	C	D	E
1	Case ID	Start Timestamp	Complete Timestamp	Activity	Resource
2	1	2011/01/01 00:00:00.000	2011/01/01 00:37:00.000	Create Purchase Requisition	Kim Passa
3	2	2011/01/01 00:16:00.000	2011/01/01 00:29:00.000	Create Purchase Requisition	Immanuel Karagianni
4	3	2011/01/01 02:23:00.000	2011/01/01 03:03:00.000	Create Purchase Requisition	Kim Passa
5	1	2011/01/01 05:37:00.000	2011/01/01 05:45:00.000	Create Request for Quotation	Kim Passa
6	1	2011/01/01 06:41:00.000	2011/01/01 06:55:00.000	Analyze Request for Quotation	Karel de Groot
7	2	2011/01/01 08:16:00.000	2011/01/01 08:26:00.000	Create Request for Quotation	Alberto Duport
8	4	2011/01/01 08:39:00.000	2011/01/01 09:00:00.000	Create Purchase Requisition	Fjodor Kowalski
9	2	2011/01/01 09:34:00.000	2011/01/01 09:38:00.000	Analyze Request for Quotation	Karel de Groot
10	5	2011/01/01 09:49:00.000	2011/01/01 10:35:00.000	Create Purchase Requisition	Esmana Liubiata
11	2	2011/01/01 10:16:00.000	2011/01/01 10:21:00.000	Amend Request for Quotation	Christian Francois
12	2	2011/01/01 11:15:00.000	2011/01/01 11:48:00.000	Analyze Request for Quotation	Magdalena Predutta
13	6	2011/01/01 11:20:00.000	2011/01/01 11:37:00.000	Create Purchase Requisition	Christian Francois
14	1	2011/01/01 11:43:00.000	2011/01/01 12:09:00.000	Send Request for Quotation to Supplier	Karel de Groot
15	1	2011/01/01 12:32:00.000	2011/01/01 16:03:00.000	Create Quotation comparison Map	Magdalena Predutta
16	2	2011/01/01 12:33:00.000	2011/01/01 12:39:00.000	Amend Request for Quotation	Esmana Liubiata
17	2	2011/01/01 13:28:00.000	2011/01/01 13:38:00.000	Analyze Request for Quotation	Karel de Groot
18	7	2011/01/01 14:05:00.000	2011/01/01 15:00:00.000	Create Purchase Requisition	Esmana Liubiata
19	8	2011/01/01 14:27:00.000	2011/01/01 15:17:00.000	Create Purchase Requisition	Fjodor Kowalski
20	2	2011/01/01 15:18:00.000	2011/01/01 15:40:00.000	Send Request for Quotation to Supplier	Francois de Perrier
21	2	2011/01/01 15:55:00.000	2011/01/01 16:43:00.000	Create Quotation comparison Map	Karel de Groot
22	9	2011/01/01 16:17:00.000	2011/01/01 16:34:00.000	Create Purchase Requisition	Tesca Lobes
23	6	2011/01/01 17:32:00.000	2011/01/01 17:45:00.000	Create Request for Quotation	Alberto Duport
24	8	2011/01/01 18:00:00.000	2011/01/01 18:07:00.000	Create Request for Quotation	Tesca Lobes
25	6	2011/01/01 18:39:00.000	2011/01/01 18:55:00.000	Analyze Request for Quotation	Magdalena Predutta
26	4	2011/01/01 18:45:00.000	2011/01/01 18:51:00.000	Analyze Purchase Requisition	Maris Freeman
27	4	2011/01/01 18:56:00.000	2011/01/01 18:58:00.000	Create Request for Quotation	Heinz Gutschmidt
28	8	2011/01/01 19:04:00.000	2011/01/01 19:27:00.000	Analyze Request for Quotation	Francois de Perrier
29	6	2011/01/01 19:47:00.000	2011/01/01 19:55:00.000	Amend Request for Quotation	Penn Osterwalder
30	4	2011/01/01 19:58:00.000	2011/01/01 20:19:00.000	Analyze Request for Quotation	Francois de Perrier
31	8	2011/01/01 20:21:00.000	2011/01/01 20:34:00.000	Amend Request for Quotation	Tesca Lobes
32	6	2011/01/01 20:55:00.000	2011/01/01 21:28:00.000	Analyze Request for Quotation	Francois de Perrier
33	4	2011/01/01 21:06:00.000	2011/01/01 21:14:00.000	Amend Request for Quotation	Nico Ojenbeer
34	8	2011/01/01 21:35:00.000	2011/01/01 22:01:00.000	Analyze Request for Quotation	Magdalena Predutta
35	4	2011/01/01 22:01:00.000	2011/01/01 22:24:00.000	Analyze Request for Quotation	Francois de Perrier

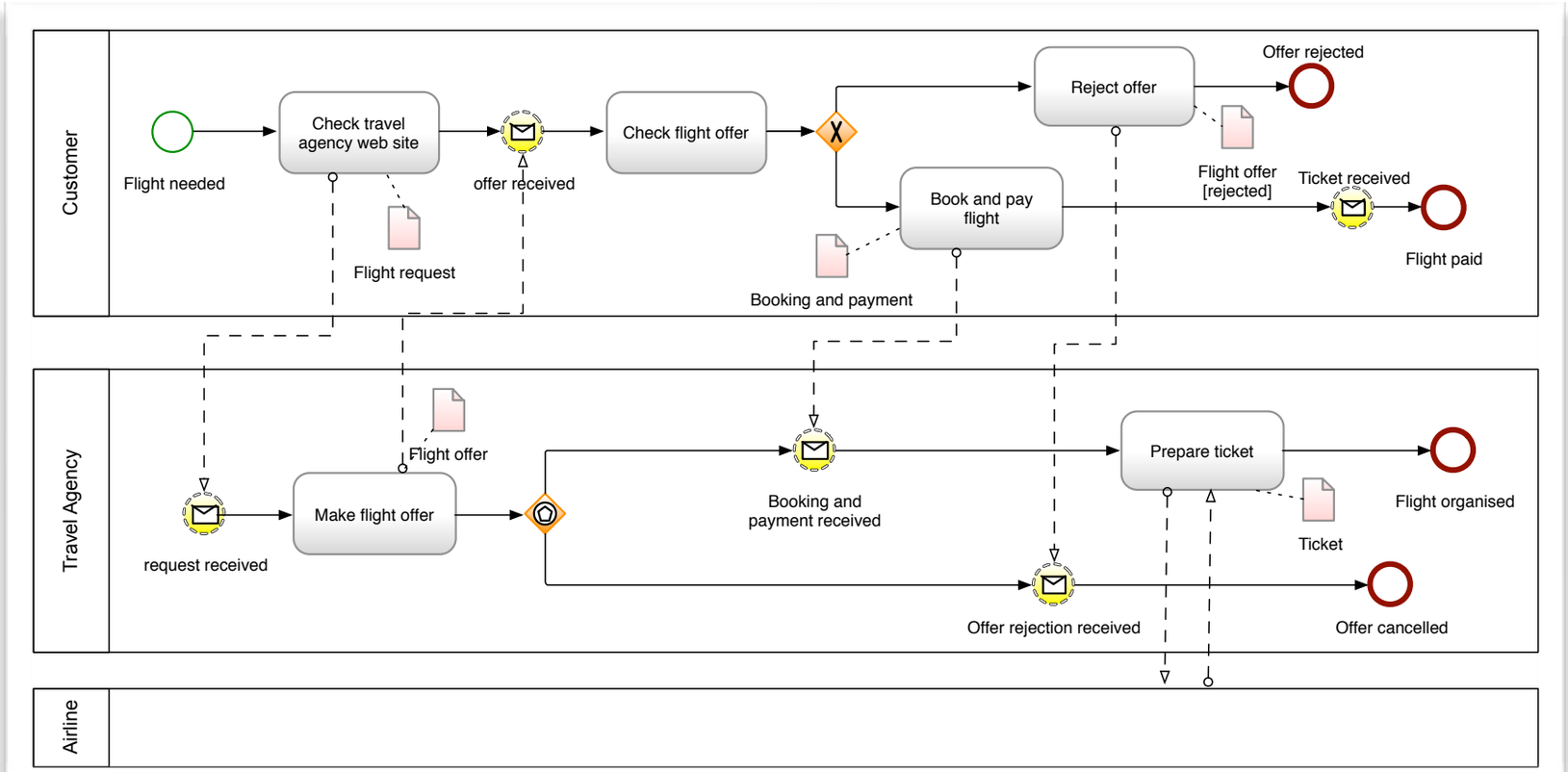


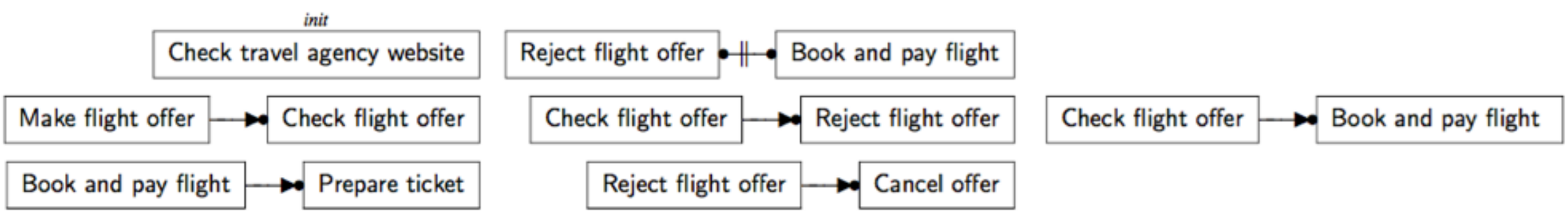
Model?

Example: non compliant process execution

Problems!

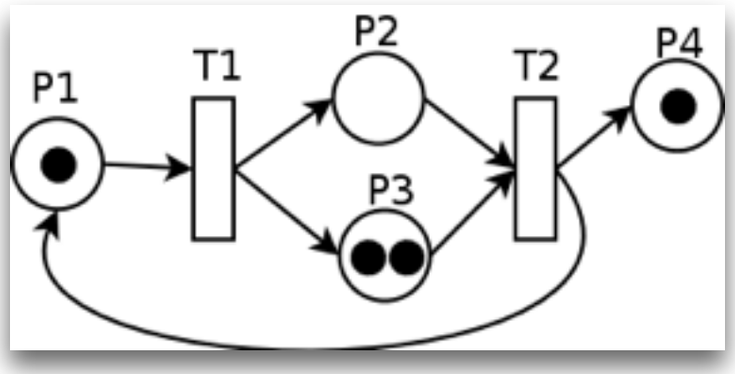
- Are these representing the `same' process?



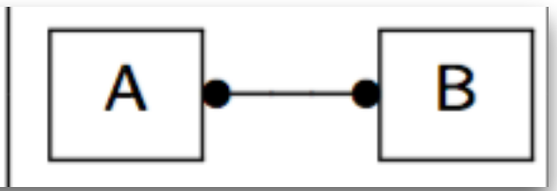


Problems!

Formal semantics of what?



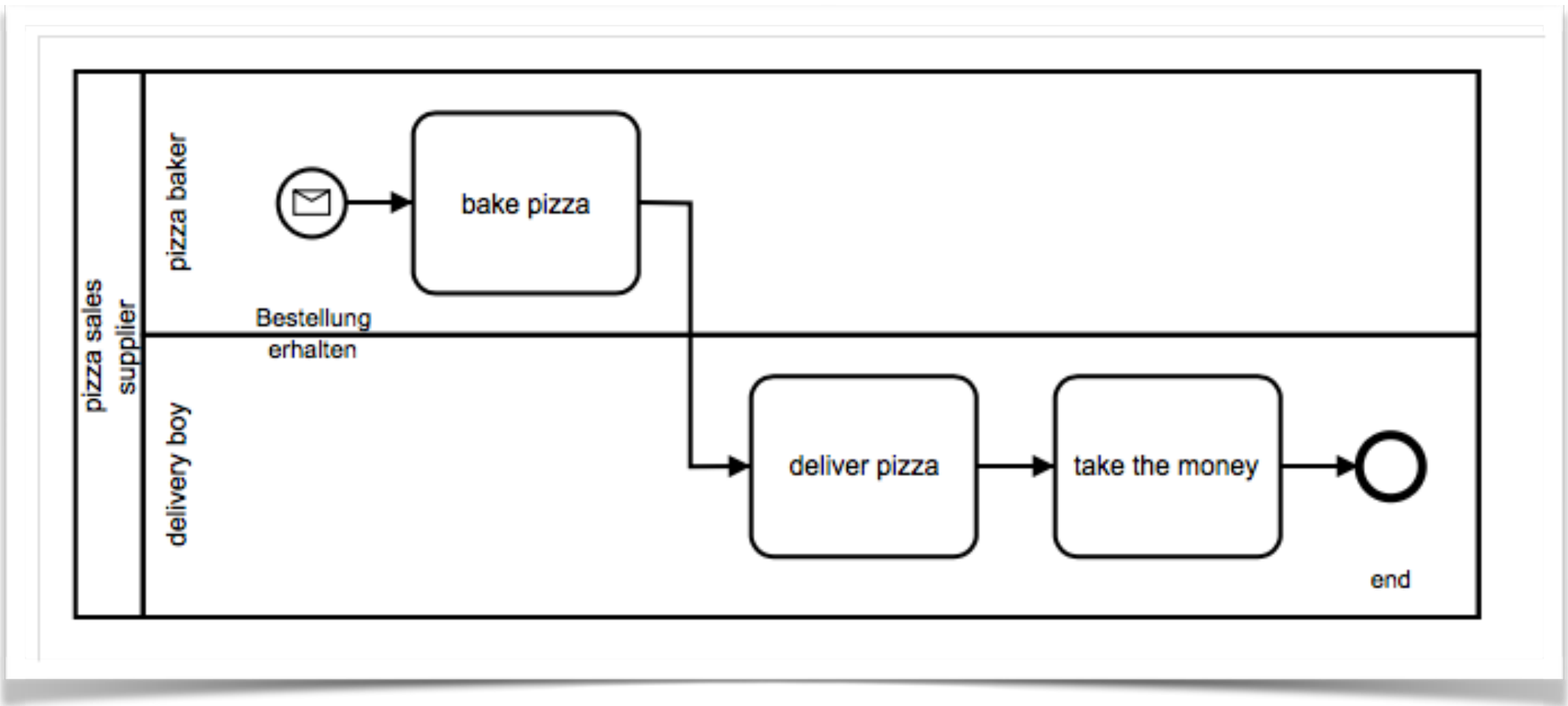
$$(\Diamond A \rightarrow \Diamond B) \wedge \neg(\Diamond B \rightarrow \Diamond A)$$



The execution of the control flow

Problems!

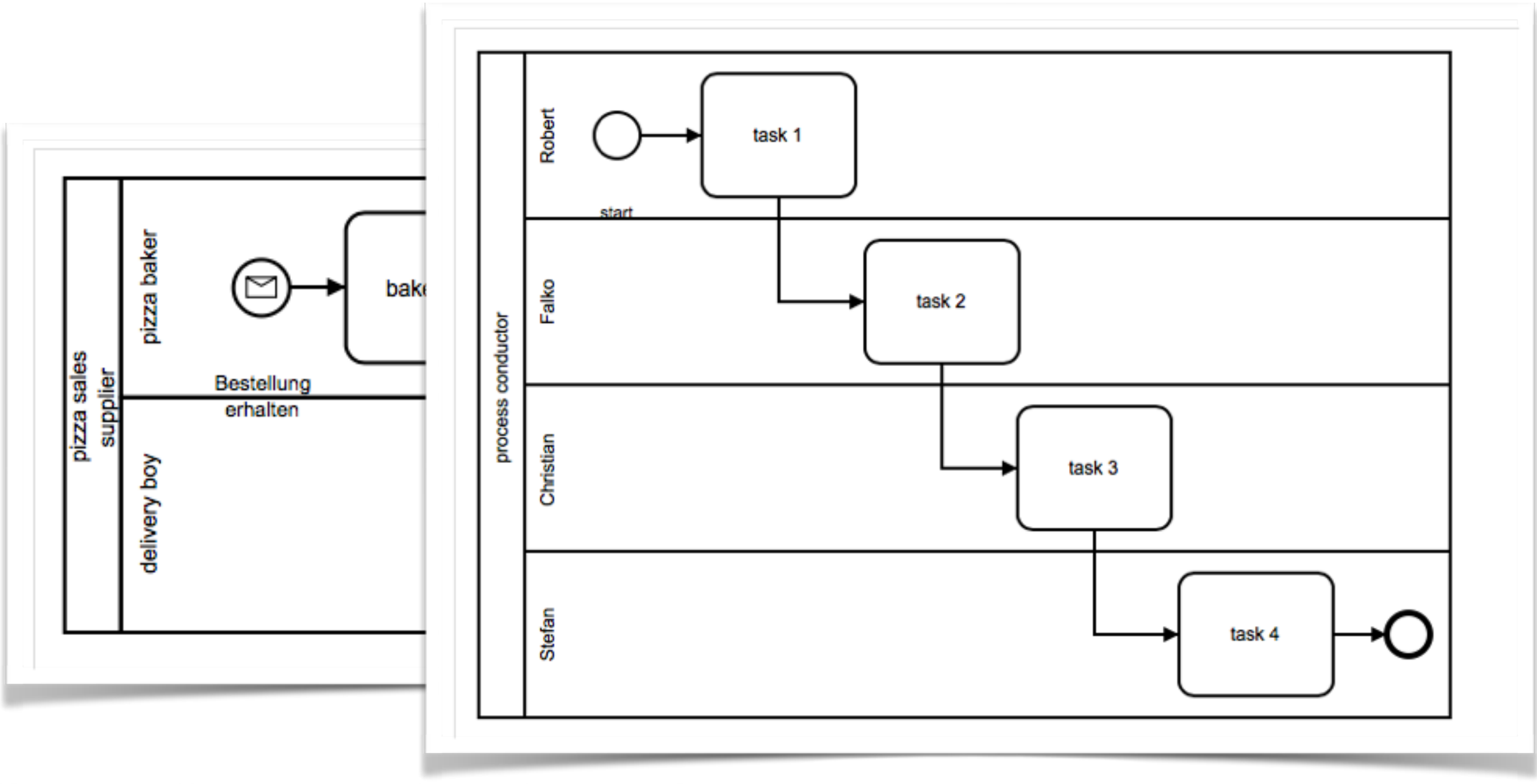
- What is the meaning of the different constructs?



[See https://camunda.org/bpmn/reference]

Problems!

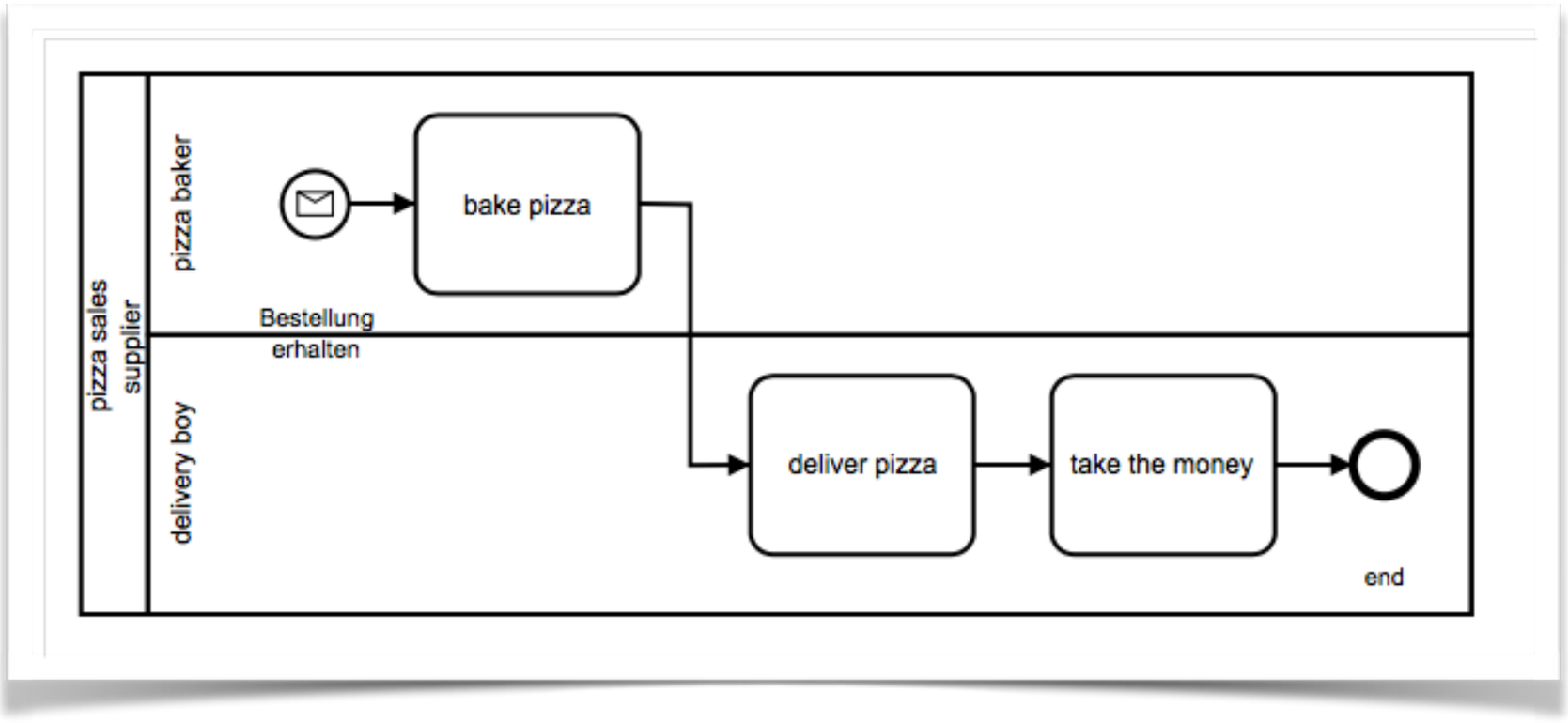
- What is the meaning of the different constructs?



[See https://camunda.org/bpmn/reference]

Problems!

- What is the meaning of the different constructs?



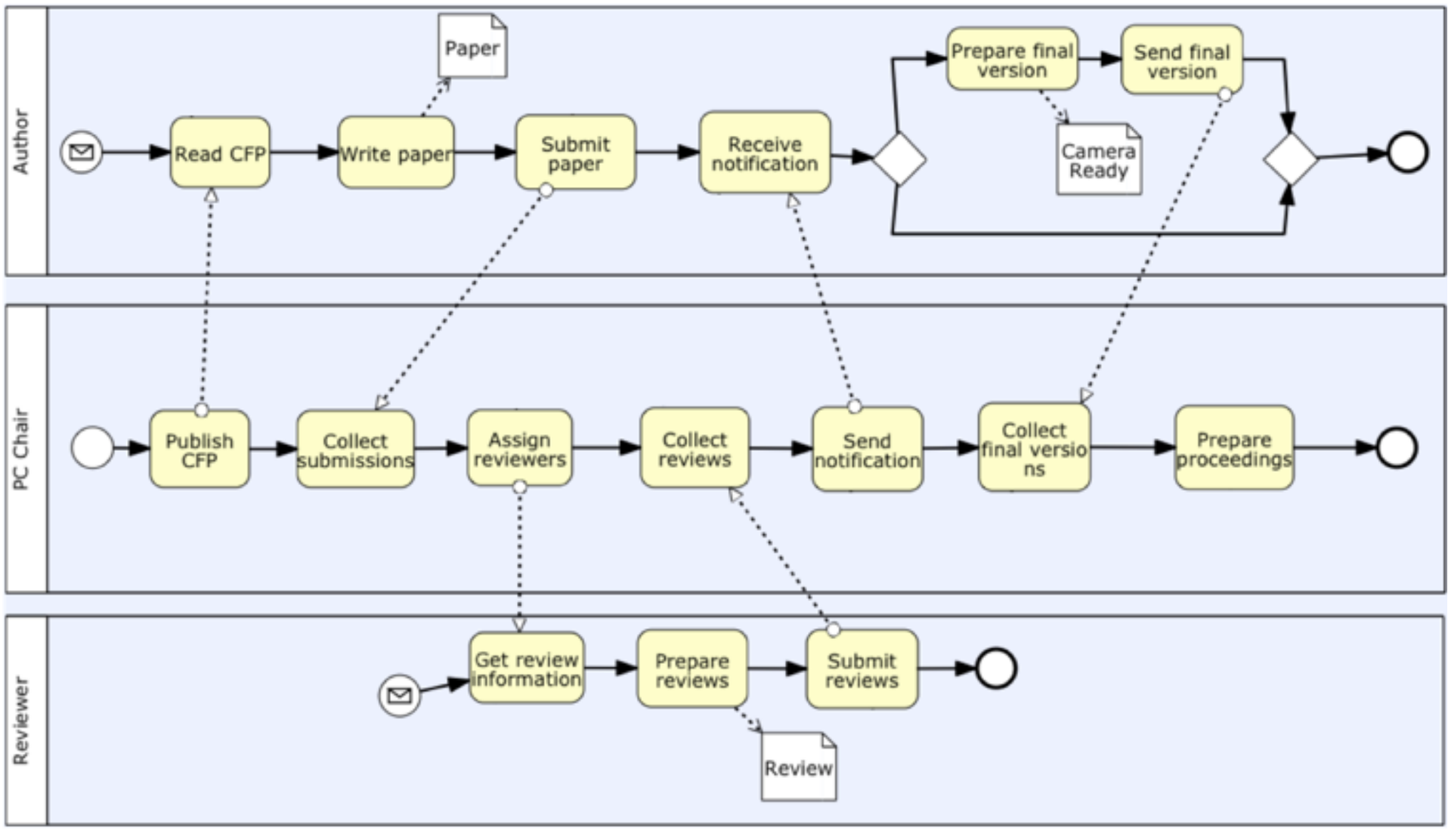
Can you deliver before baking?

Can you get paid before delivering?

[See https://camunda.org/bpmn/reference]

Problems!

- How to provide a semantics behind the control flow execution?

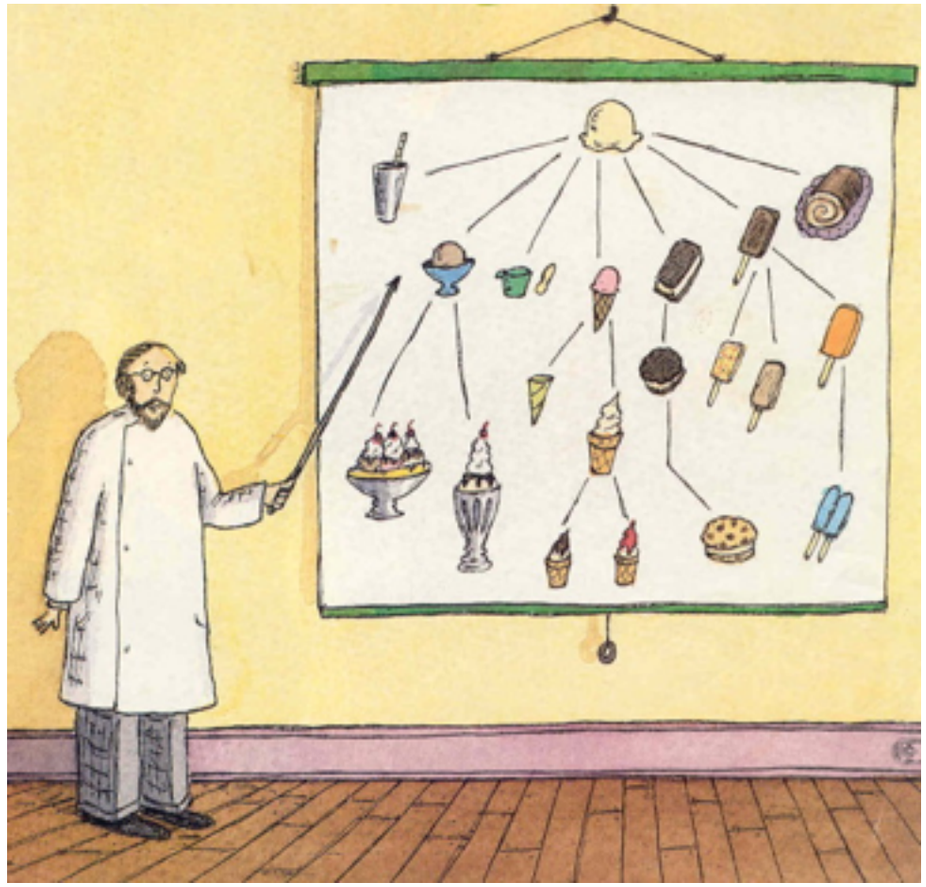


Exploiting ontologies

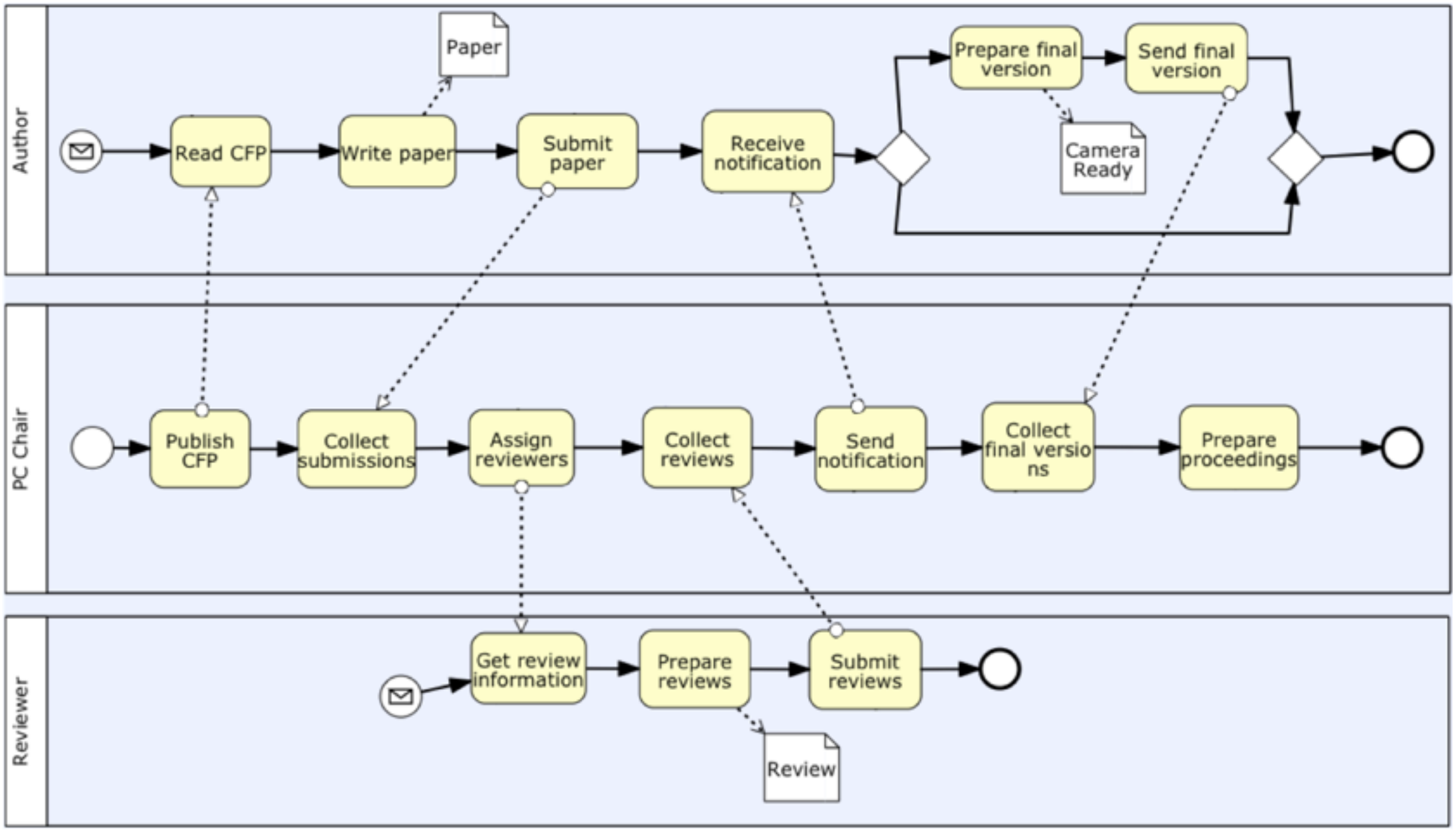
- How to provide a semantics behind the control flow!
- Idea 1: build an ontology providing a semantics to business process diagrams.

An ontology for the Business Process Modelling Notation

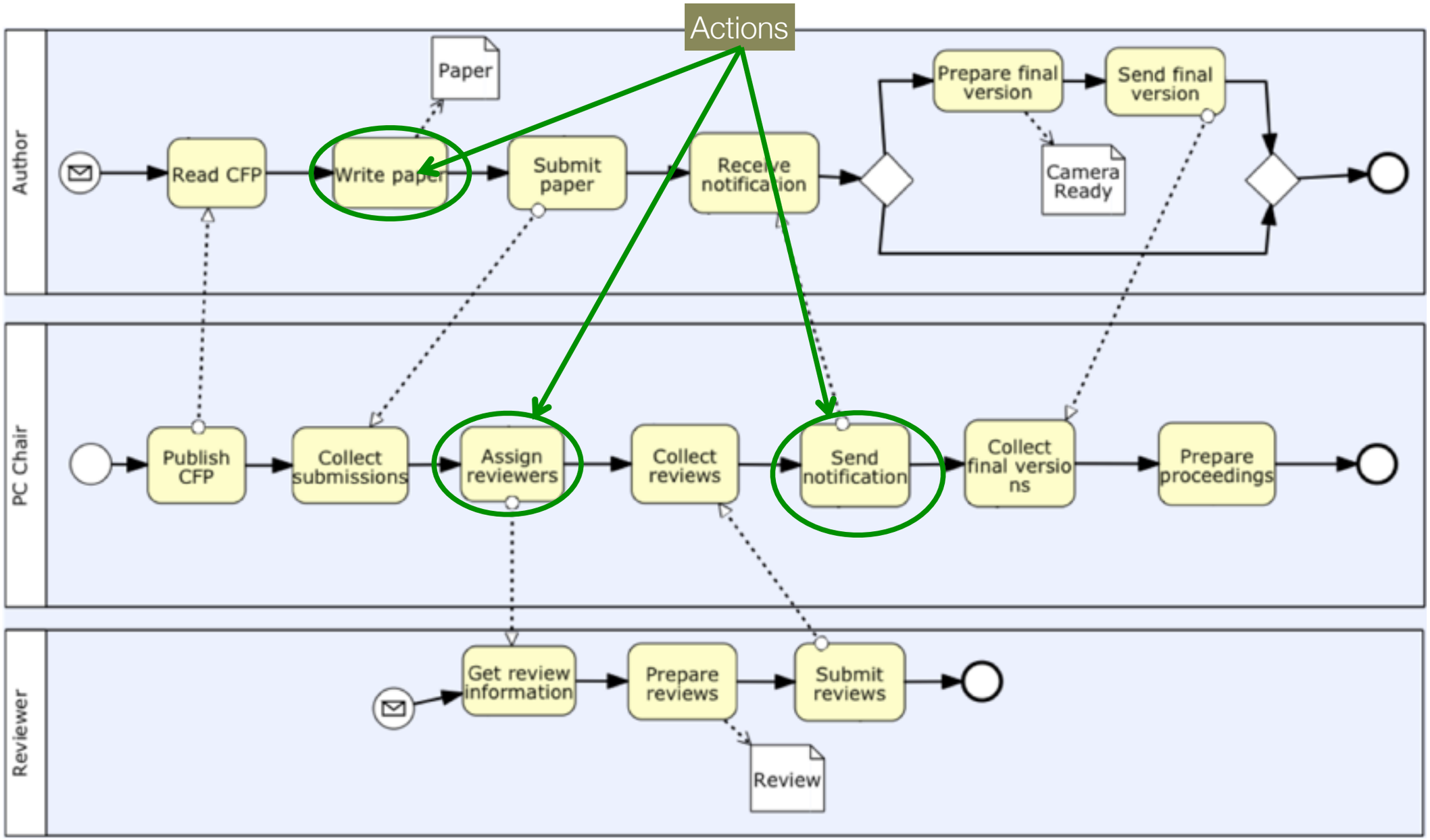
Joint work with Marco Rospocher, Luciano Serafini
Chiara Di Francescomarino, Mauro Dragoni



Business Process Modelling Notation (BPMN)

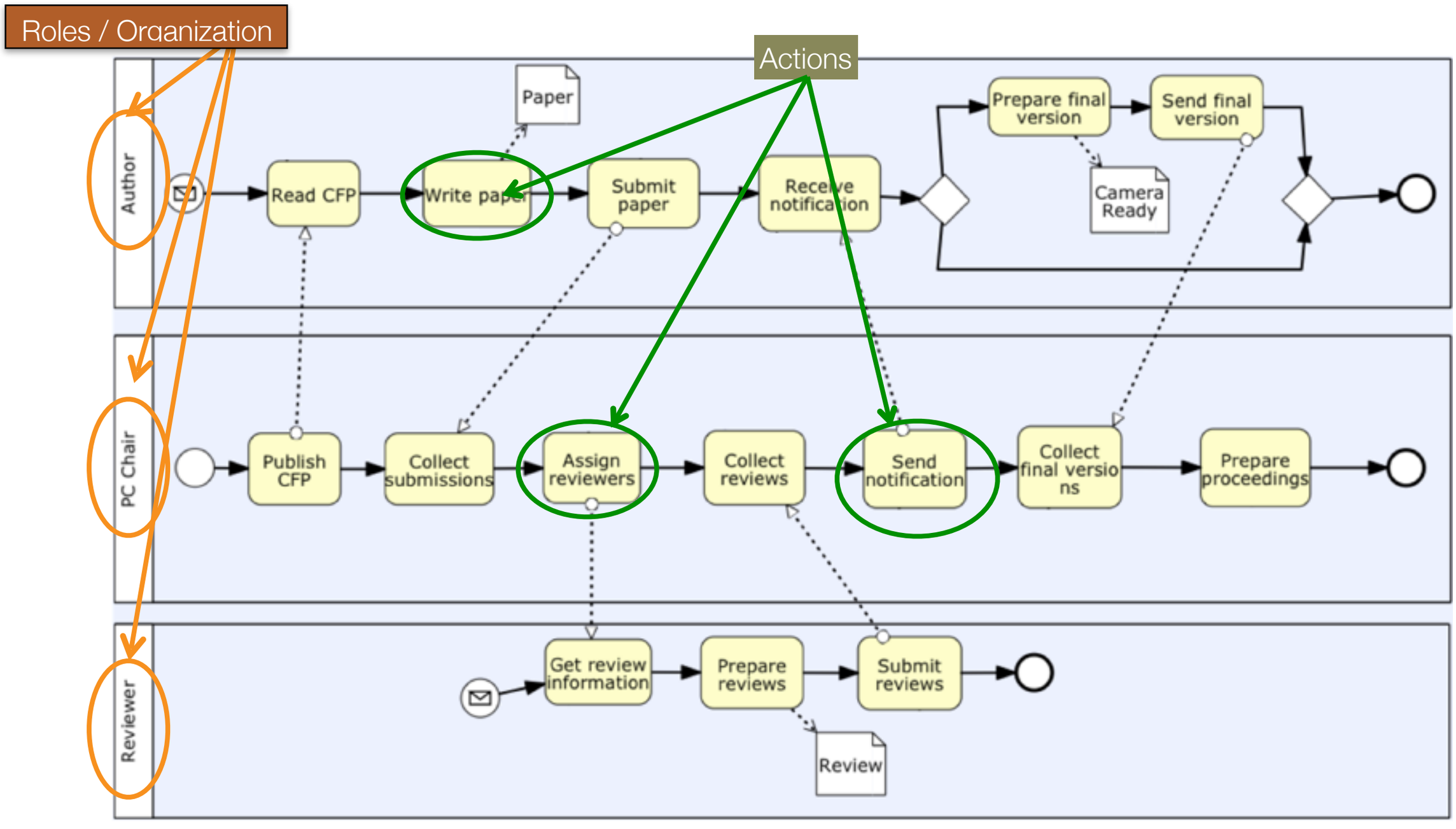


Business Process Modelling Notation (BPMN)



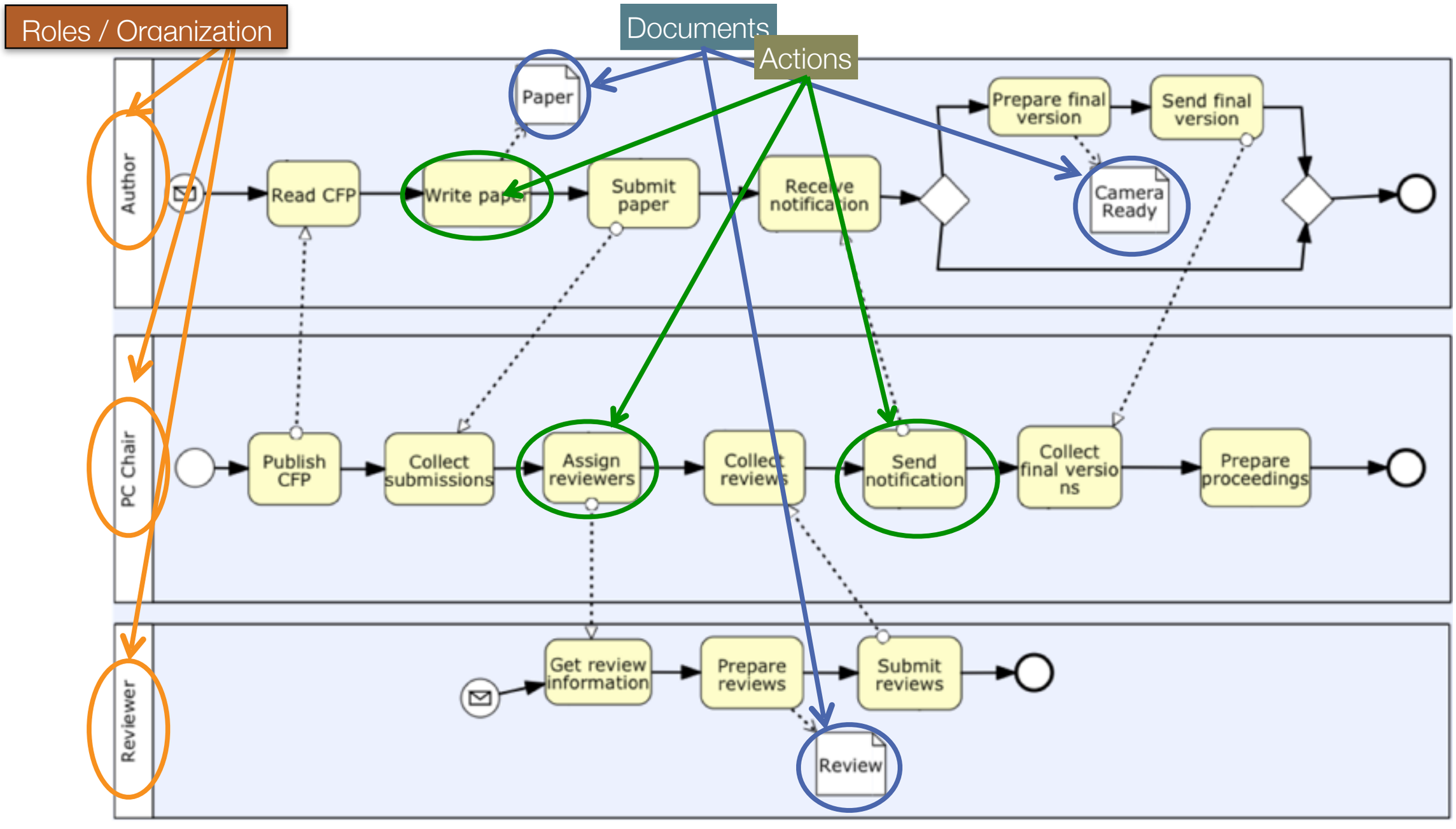
Business Process Modelling Notation (BPMN)

- State of the art graphical language for the specification of business processes



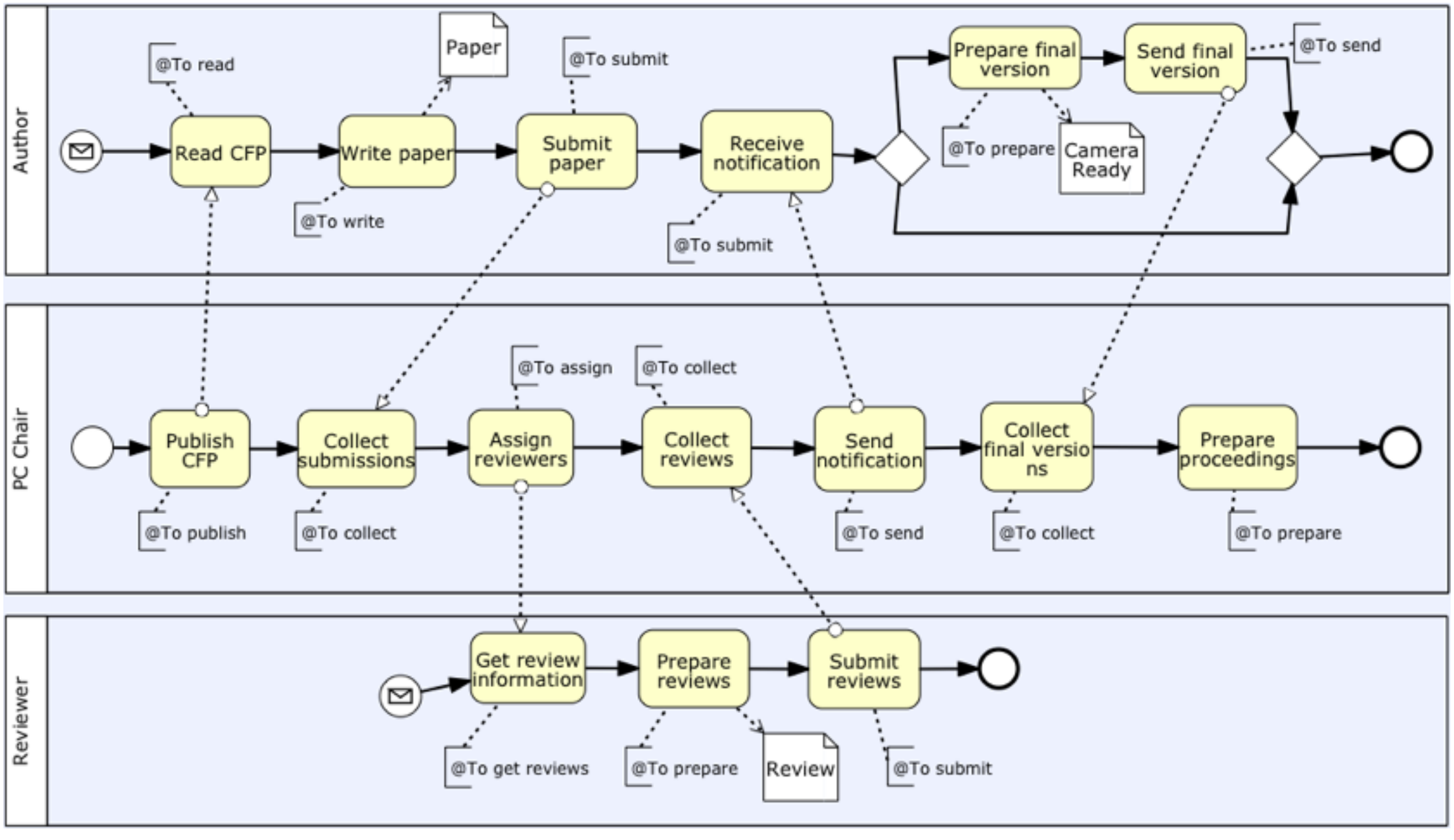
Business Process Modelling Notation (BPMN)

- State of the art graphical language for the specification of business processes



Business Process Modelling Notation (BPMN)

- Annotations!



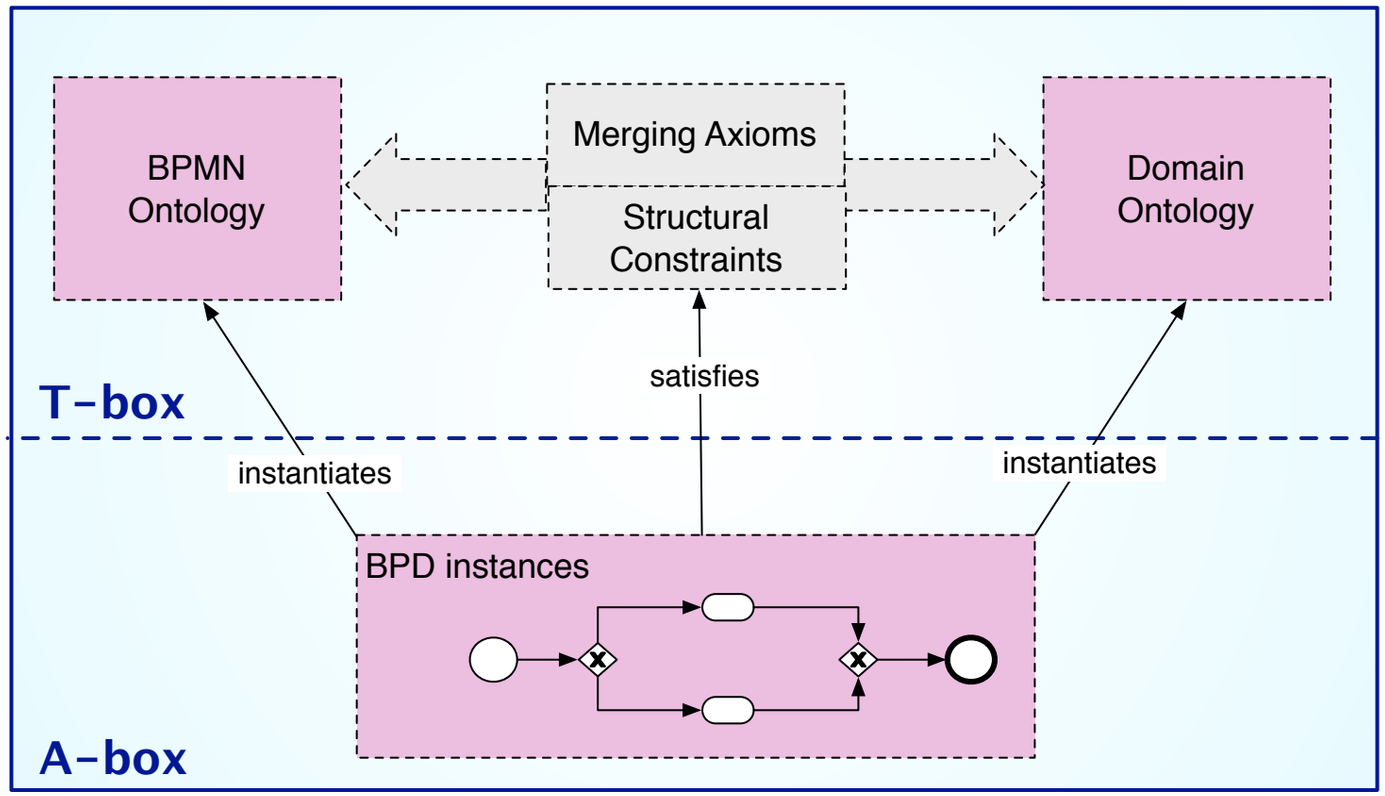
Why going beyond the control flow?

- Example of queries that encompass the mere process execution:
  - *What are the **activities** performed by a certain **role** (e.g. PC Chair)?*
  - ***Where** are **documents** (e.g. reviews, notifications) produced?*
  - *What are the **activities** where **something is published**? What are the **activities** where **something is sent out**?*
  - *What are the **activities** an **author** perform **right before** submitting something?*

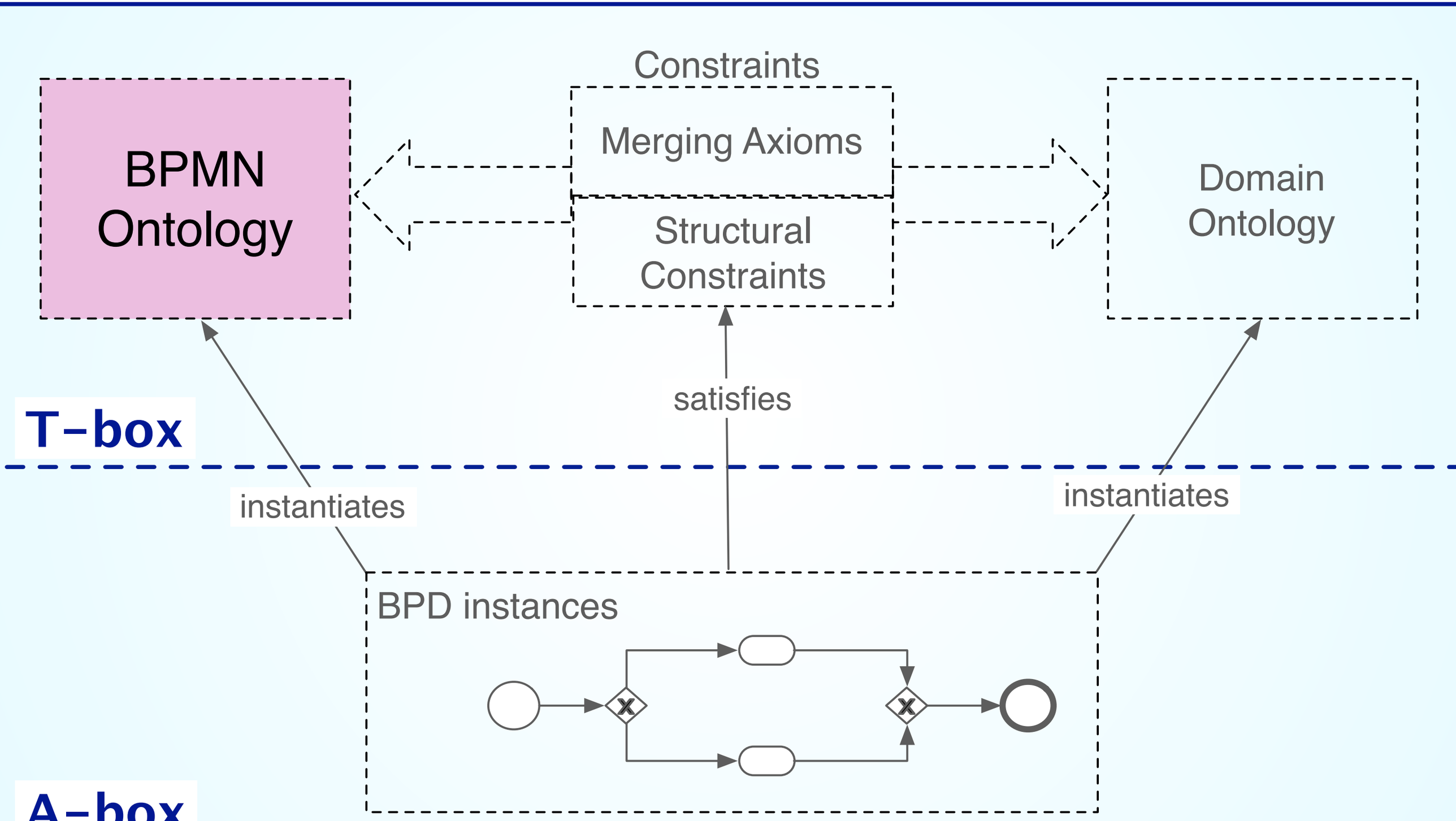
- Examples of application that requires querying for both ontological and process knowledge: **cross-cutting concerns, critical patterns**
  - *Where does the user make **selections**?*
  - *Before confirming an order the user must choose a shipment method*

Semantically Annotated Business Processes

- Semantically annotated business processes are encoded into a logical knowledge base implemented in OWL



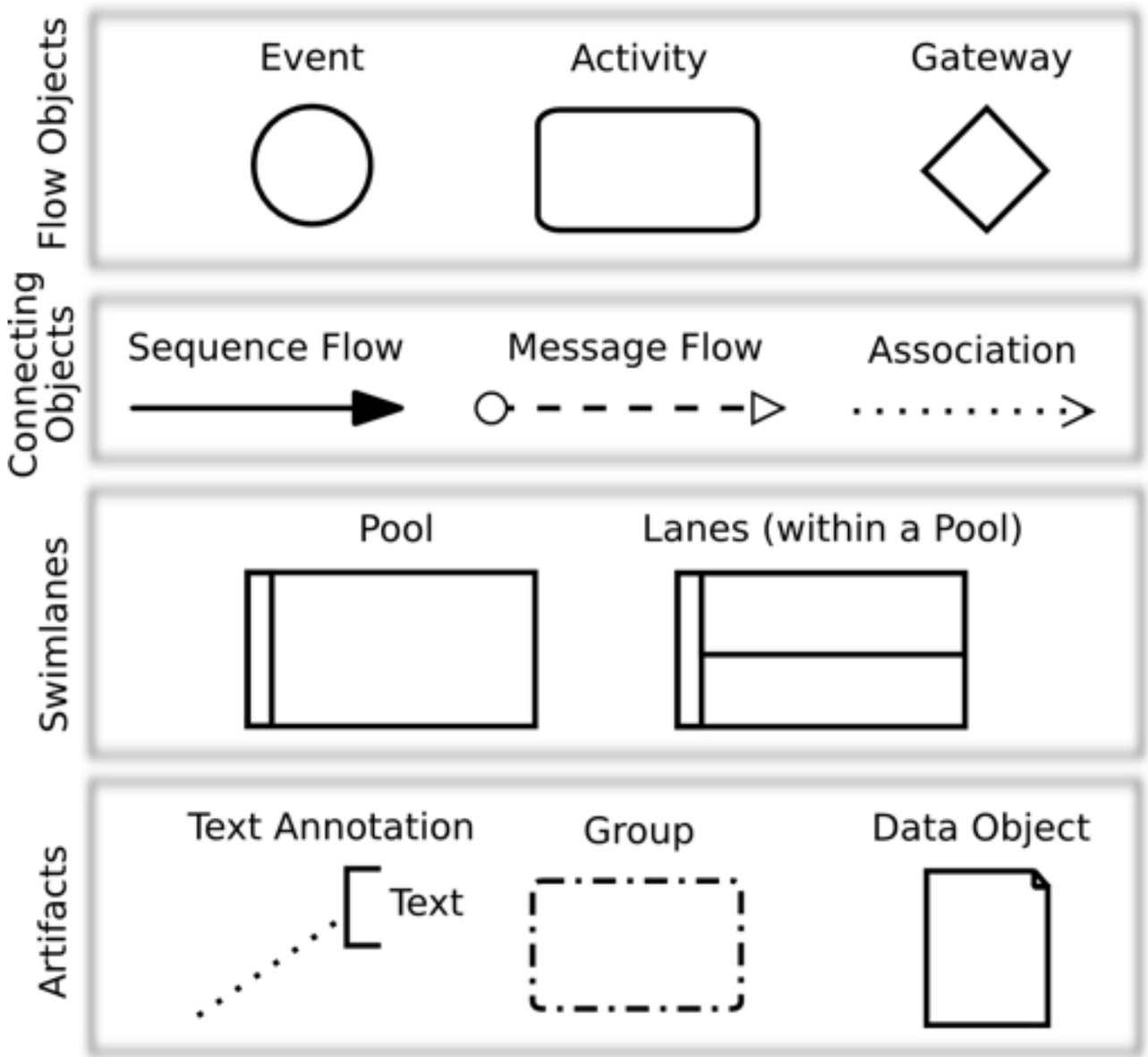
- Note: Business Process Diagrams (BPDs) are specified using the Business Process Modelling Notation (BPMN).



The BPMN ontology

Business Process Modelling Notation (BPMN)

Core Element Set



for representing something that happens (event), work to be performed (activity), and control flow elements (gateway);

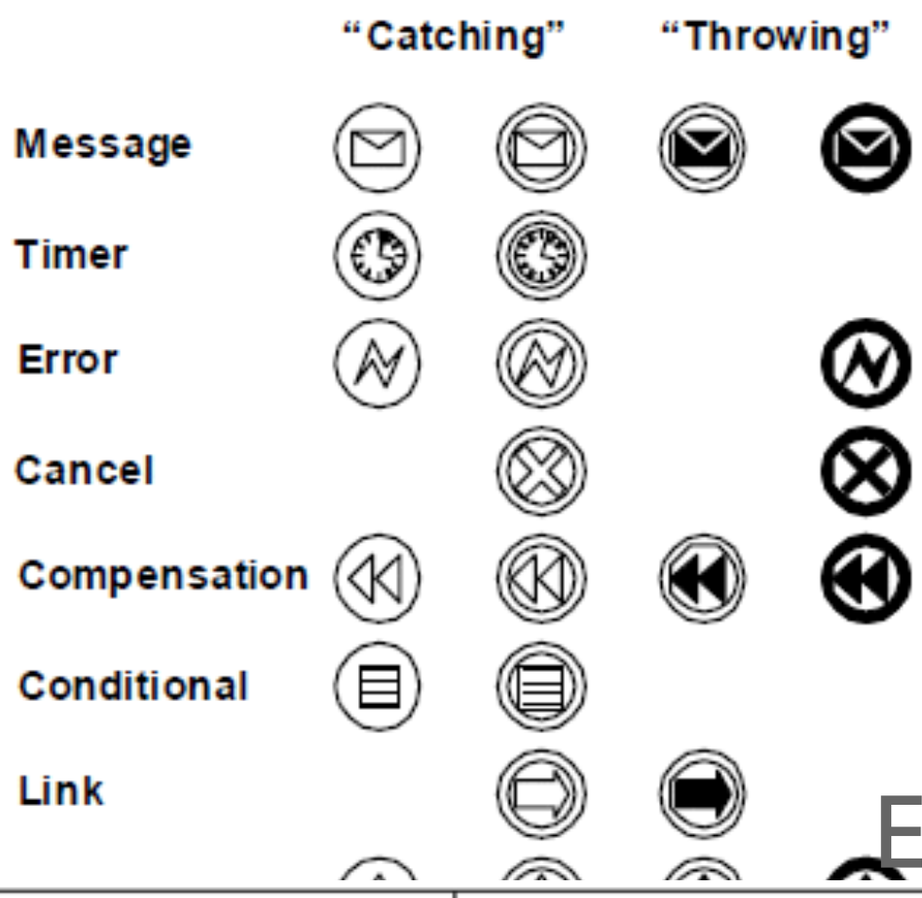
for showing the order in which activities are performed (sequence flow), ...

for describing participants in a process (pool), and to organize and categorize activities (lane);

for representing data processed/produced by activities (data object), informal grouping of activities (group), ...

Business Process Modelling Notation (BPMN)

Extended Element Set (e.g. Event types)



ErrorCode attribute for Error Event

Attributes	Description
ErrorCode : String	*For an End Event:* If the Result is an Error, then the ErrorCode MUST be supplied. This "throws" the error. [..]

Our Contribution: An ontology for BPMN

- An OWL-DL formalization of the BPMN specification

- It accurately encodes:
  - the classification of all the elements of the BPMN language
  - the formal representation of the attributes and conditions describing how the elements can be combined to obtain a “valid” BPMN business process

- The proposed formalization:
  - provides a terminological description of the language;
  - enables representing any actual BPMN diagram as a DL A-Box
    - enables several reasoning-based services

- It covers BPMN v1.1 and part of BPMN 2.02

Disclaimer

- The BPMN Ontology…
  - …is not intended to model the dynamic behaviour (behavioural semantics) of a BPMN process
    - better look at YAWL, PetriNets, …
  - …it provides an ontological formalization of BPMN as a graphical language, and not an ontological analysis in a foundational fashion
    - better look at works analysing BPMN wrt to
      - ABDESO/UFO (Guizzardi and Wagner)
      - Dolce (Sanfilippo, Borgo, and Masolo -FOMI 2014)

Modelling Process
Scope and Boundaries: Ontology Intended Uses

- Checking the compliance of a process diagram against the BPMN specification
  - e.g., the process diagram has at least one starting event and one end event, constructs are combined in the correct way
- Checking additional application-specific design guidelines
  - guidelines to guarantee process diagram readability (e.g., diagram should not contain more than ten sub-processes, every gate should have at most three out-going flows)
- Semantic description and retrieval of process diagrams (or process diagram elements)
  - e.g., to state that a certain sub-process is of type “privacy critical”, and to be able to retrieve all process diagrams that contains privacy critical sub-processes, or all privacy critical activities within a diagram
- Easy integration with organizational / domain related ontologies for enhanced semantic description and retrieval
  - e.g., check that all activities of type T performed by organization A are followed by activities of type B performed by organisation B

Modelling Process
Scope and Boundaries: Competency Questions (excerpt)

- How many flow elements does process X contain?
- What is the error code associated to error event W?
- What type of BPMN elements does sub-process Y in process X contains?
- What is the BPMN element connected by a sequence flow to activity Z?
- Is there a path of sequence flows connecting activity Z_1 to activity Z_2?
- Is process XYZ a valid process according to the BPMN specification?
- ….

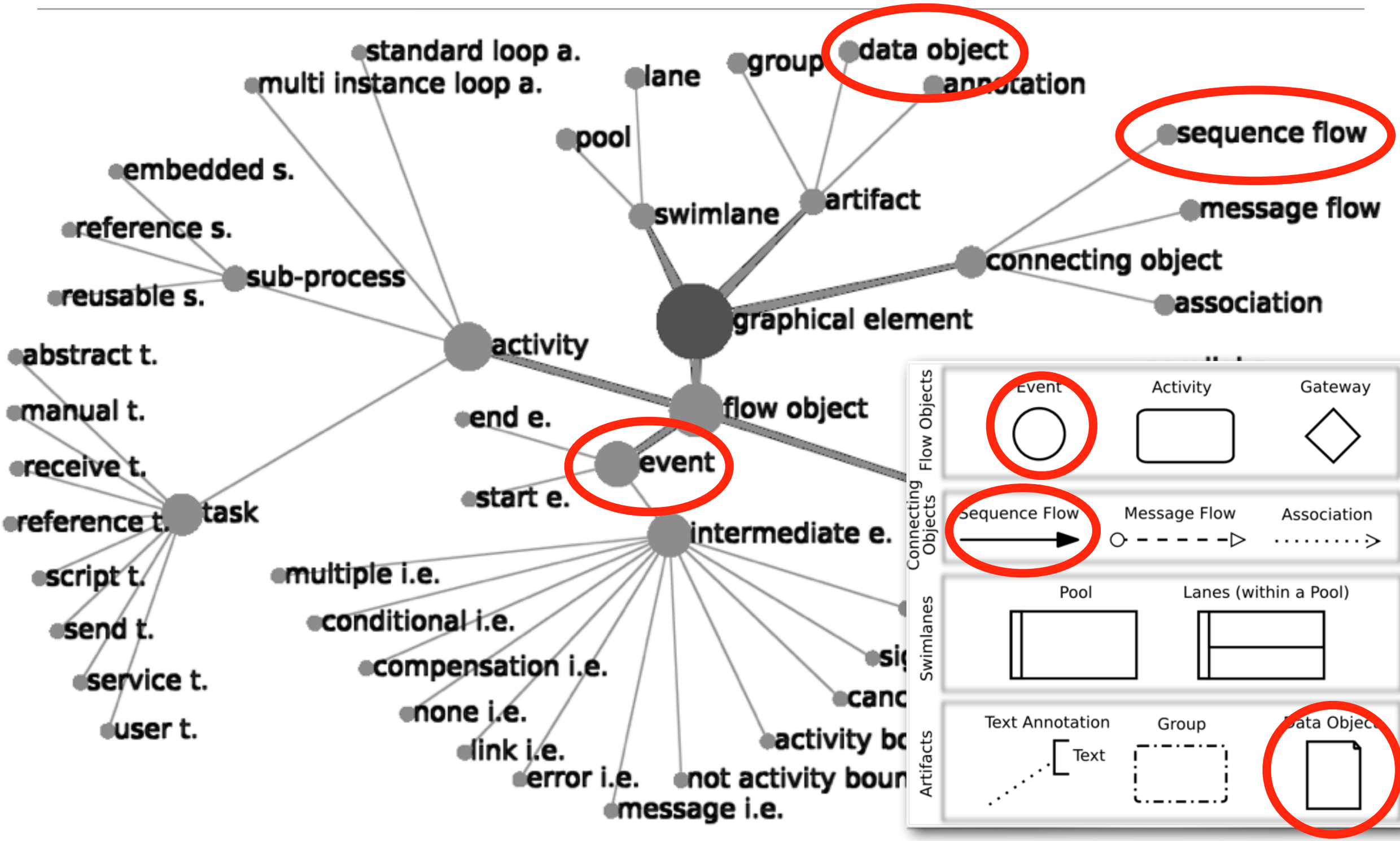
Modelling Process
Our Trusted Friend: BPMN Specification Document

For each element, it provides:

- an introductory description of the element, with some general properties and conditions
- a compact tabular description of each element's attribute
  - name, value type, multiplicity details, conditions for instantiation
- conditions holding for connecting the current element with other elements of the language
- additional details on execution level aspects of the element

Free text document, with some structure

Modelling Process
Step 1 of 3: Signature Identification



Modelling Process
Step 2 of 3: Attribute Restrictions

- An attribute is formalized either as datatype property or as an object property

- Three situations considered:

1. the value type of the attribute is another BPMN element
2. the value type of the attribute is a datatype, but only an enumerated set of options is allowed and some conditions may apply to these options
3. the value type of the attribute is a datatype with no restriction

Modelling Process
Step 2 of 3: Attribute Restrictions (cont'd)

- Case I: The value type of the attribute is another BPMN element
- Example:
  - Target attribute of Intermediate Event [p47]

Target (0-1) : Activity	A Target MAY be included for the Intermediate Event. The Target MUST be an activity (Sub-Process or Task). This means that the Intermediate Event is attached to the boundary of the activity and is used to signify an exception or compensation for that activity.

- Formalization: as object property
  - domain: the class having the attribute
  - range: the class of the element mentioned as value type of the attribute

$$\exists hasIntermediateEventTarget.\top \sqsubseteq IntermediateEvent$$

$$\top \sqsubseteq \forall hasIntermediateEventTarget.Activity$$

Modelling Process
Step 2 of 3: Attribute Restrictions (cont'd)

- Case II: The value type of the attribute is a datatype, but only an enumerated set of options is allowed and some conditions may apply to these options

- Example:
  - AdHocOrdering attribute of Embedded SubProcess [p47]

[AdHoc = True only] **AdHocOrdering** (0-1) (Sequential | Parallel) Parallel : String	If the Embedded Sub-Process is Ad Hoc (the AdHoc attribute is True), then the AdHocOrdering attribute MUST be included. This attribute defines if the activities within the Process can be performed in Parallel or must be performed sequentially. The default setting is Parallel and the setting of Sequential is a restriction on the performance that may be required due to shared resources.

- Formalization: as object property
  - domain: the class having the attribute
  - range: a new class enumerating all possible values of the attribute

$$\exists hasESPAdHocOrdering.\top \sqsubseteq EmbeddedSubProcess$$
$$\top \sqsubseteq \forall hasESPAdHocOrdering.AdHocOrderingType$$

Modelling Process
Step 2 of 3: Attribute Restrictions (cont'd)

- Case III: The value type of the attribute is a datatype with no restriction

- Example:
  - Text attribute of Text Annotation [p95]

Text : String	Text is an attribute which is text that the modeler wishes to communicate to the reader of the Diagram.

- Formalization: as datatype property
  - domain: the class having the attribute
  - range: a datatype compatible with the value type of the attribute

$$\exists hasTextAnnotationText.\top \sqsubseteq TextAnnotation$$
$$\top \sqsubseteq \forall hasTextAnnotationText.DT\{string\}$$

Modelling Process
Step 2 of 3: Attribute Restrictions (cont'd)

- For each attribute, we formalized its multiplicity details as an OWL cardinality restriction on the class having the attribute
  - (0..1) multiplicity is encoded as "at most one" OWL cardinality restriction
  - (1) multiplicity is encoded as "exactly one" OWL cardinality restriction
  - (1..n) multiplicity is encoded as "at least one" OWL cardinality restriction
  - (0..n) multiplicity is not encoded at all

- Example:
  - State attribute of Data Object [p94]

State (0-1) : String	State is an optional attribute that indicates the impact the Process has had on the Data Object. Multiple Data Objects with the same name MAY share the same state within one Process.

$$DataObject \sqsubseteq (\leq 1)hasState$$

Modelling Process
Step 2 of 3: Attribute Restrictions (cont'd)

- For each attribute, we also encode additional conditions ruling the usage of the attribute

- Example:
  - ErrorCode attribute of Error, in case the Error is a result of an End Event [p.94]

ErrorCode : String	For an End Event: If the Result is an Error, then the ErrorCode MUST be supplied. This "throws" the error.

- Formalization: case by case

$$EndEvent \sqsubseteq \neg\exists hasResult.Error \sqcup$$
$$\exists hasResult.(Error \sqcap \exists hasErrorCode)$$

Modelling Process
Step 3 of 3: Structural Constraints Formalization

- Formalization of the conditions concerning the usage of the elements of the language to compose a BPMN diagram

- Example: [p48]
  - A Start Event MUST be a source for Sequence Flow.

$$StartEvent \sqsubseteq \exists hasConnectingObjectSource^{-1}.SequenceFlow$$

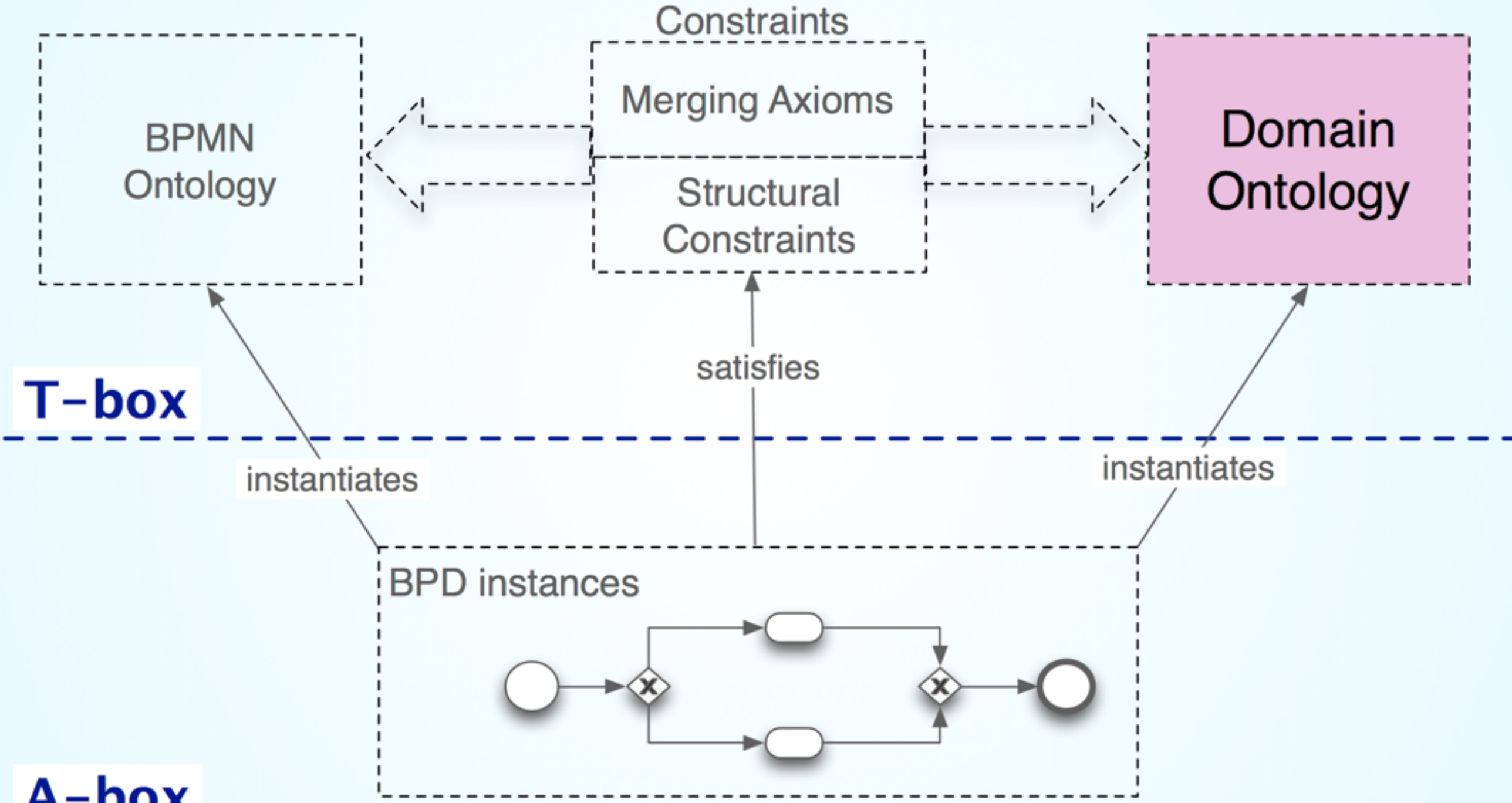
- Formalization: case by case

The BPMN Ontology
Limitations

- A few documented properties and conditions are not encoded in the BPMN Ontology:
  - Execution level properties (behavioural)
  - Attribute default values
  - “Undecidable” conditions

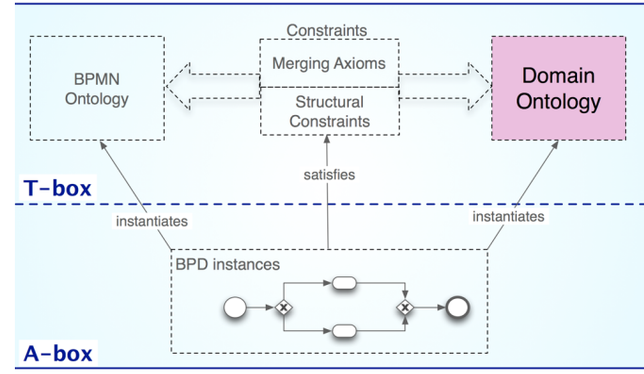
The BPMN Ontology
Ontology Metrics

Feature	Value
DL Expressivity	$\mathcal{SHOIN}(\mathcal{D})$
Classes	117
Object Properties	123
Datatype Properties	48
Individuals	104
Class Axioms	463
Object Property Axioms	236
Datatype Property Axioms	96
Individual Axioms	250
Annotation	504



The domain Ontology

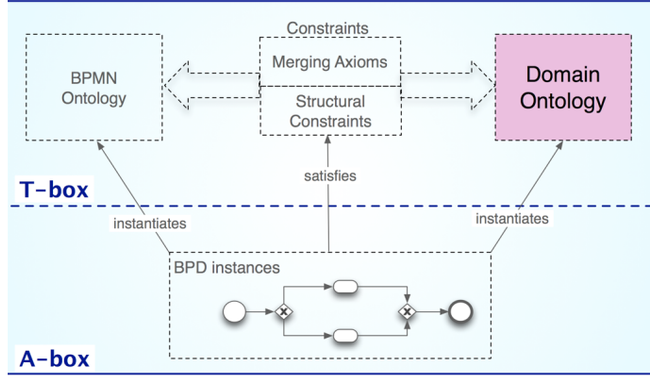
The domain ontology

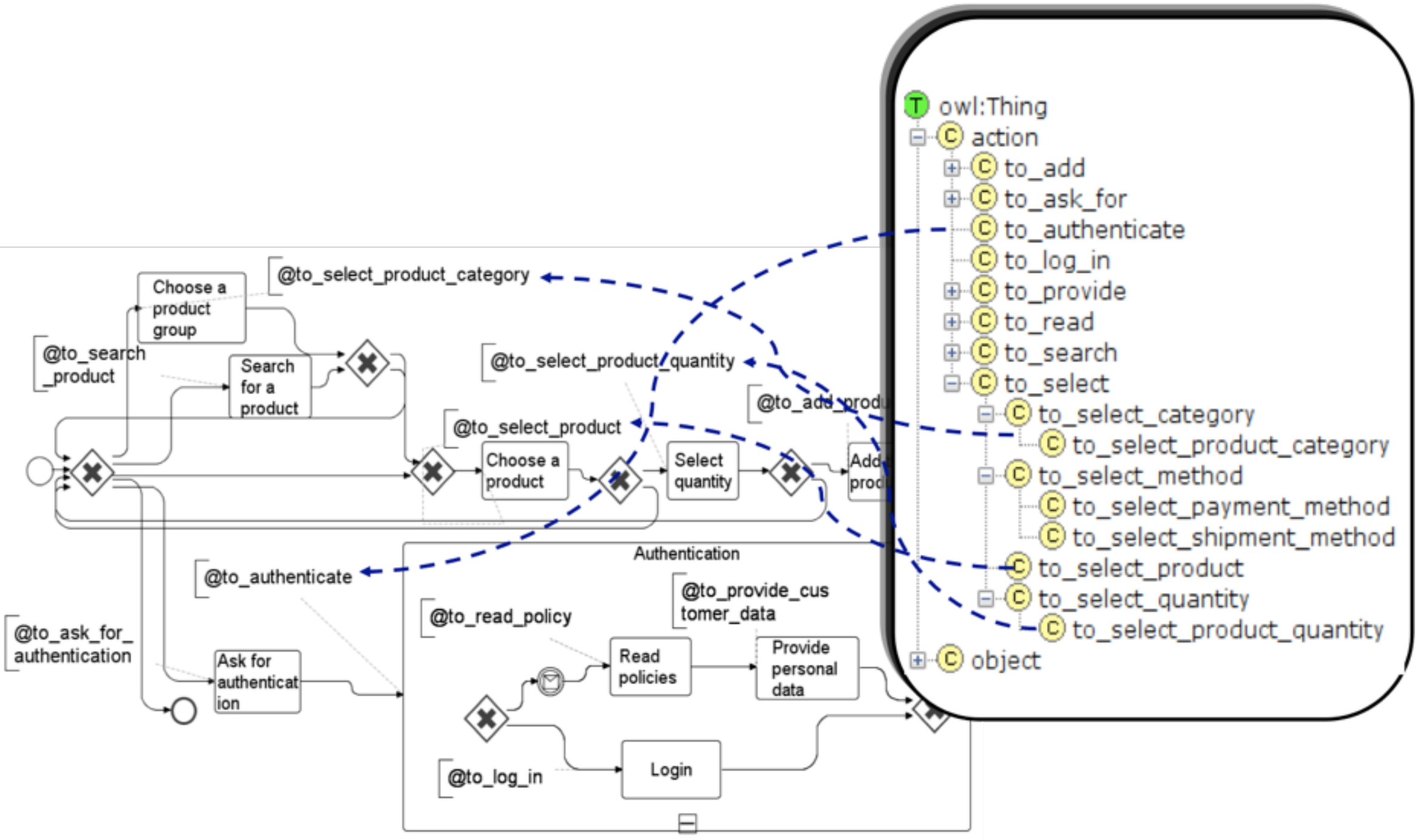


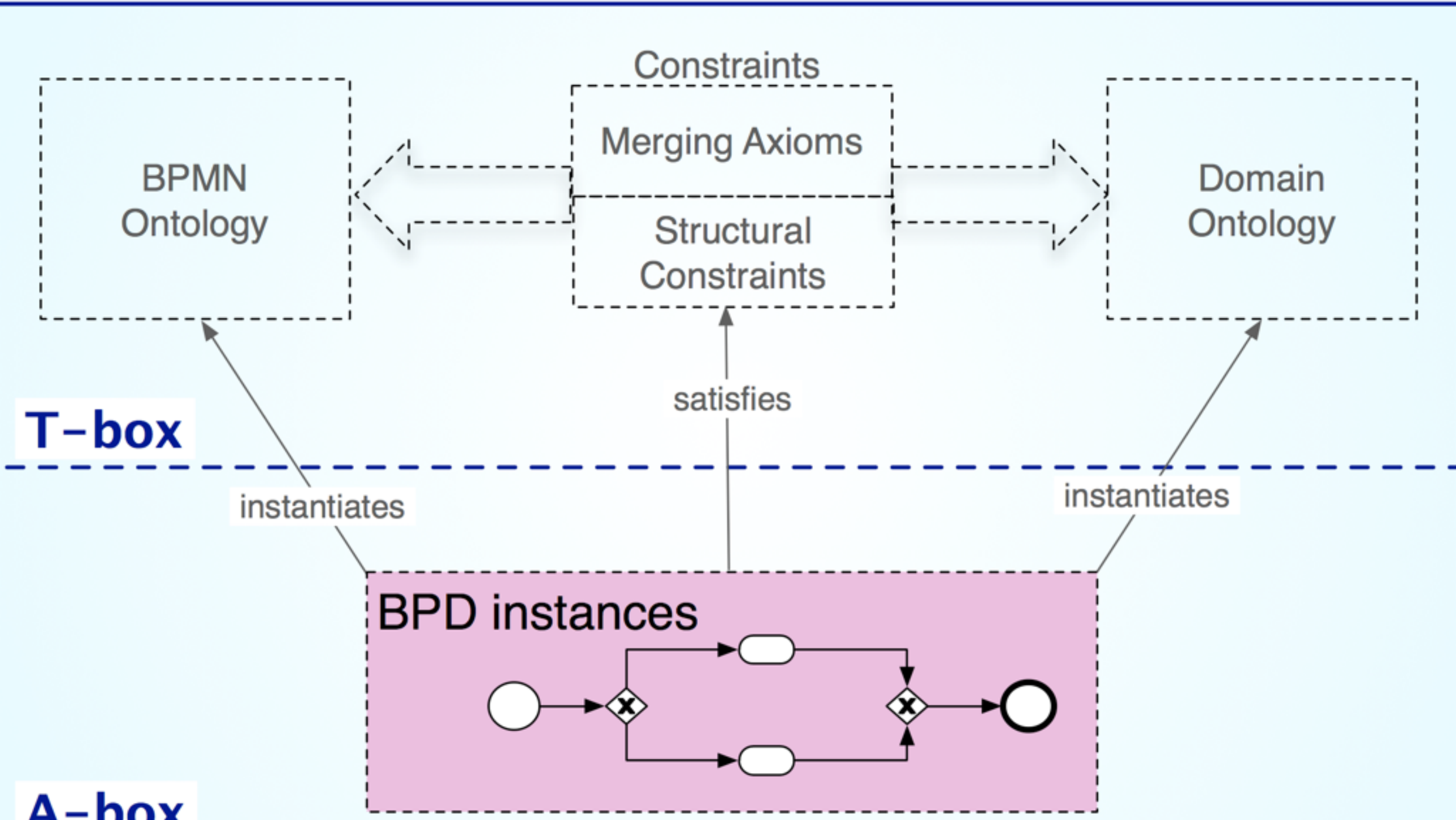
- Represents the (specific) business domain:
  - *Organizational hierarchy*
  - *Data objects*
  - *Documents classification*

- Used to annotate the elements of the BPD; Can be composed of:
  - Top level ontologies, such as DOLCE;
  - Domain-specific ontologies.

The domain ontology



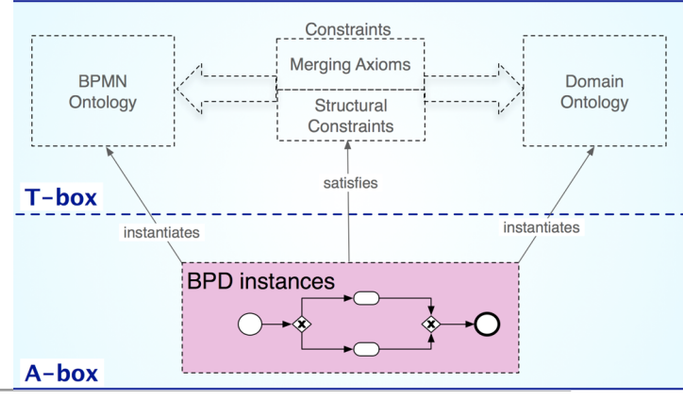




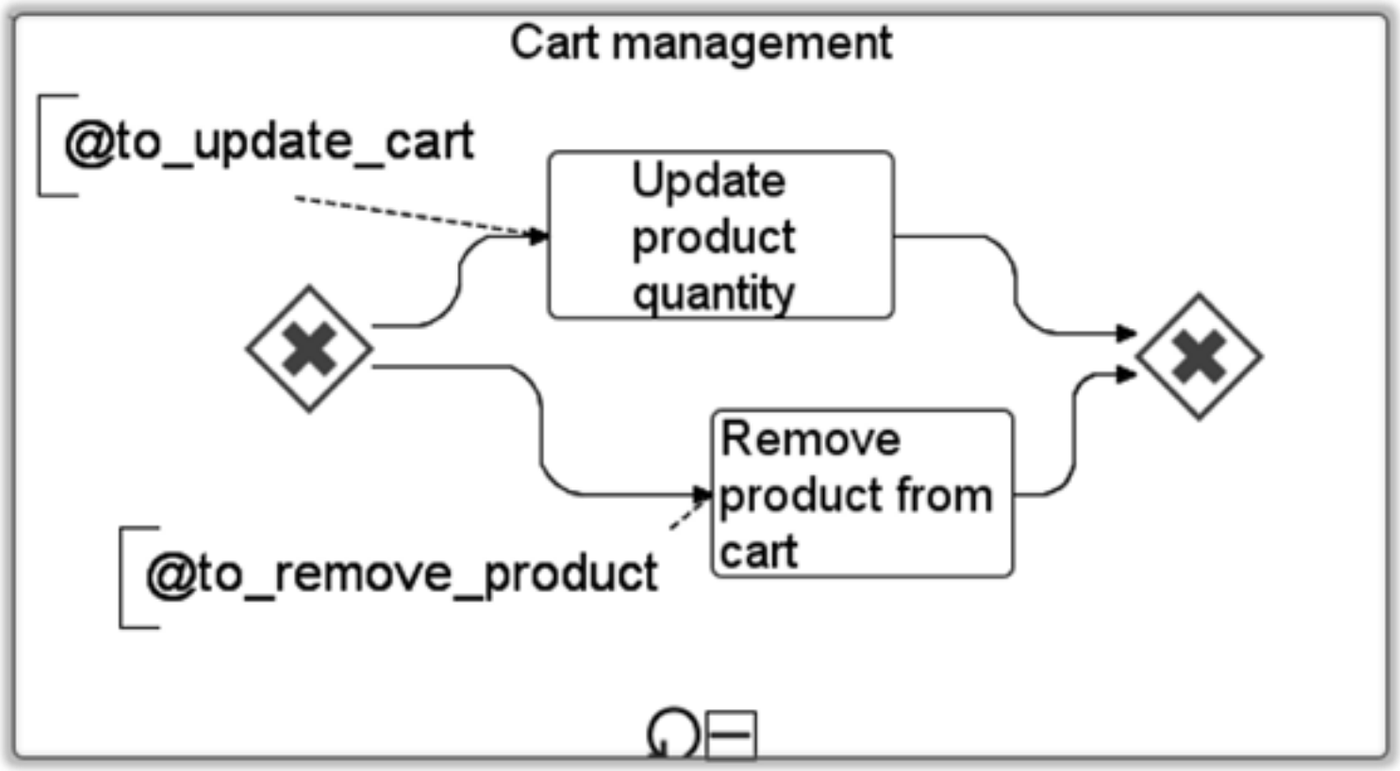
Transform a BPMN diagram

into OWL

Instantiating the BPMN Ontology

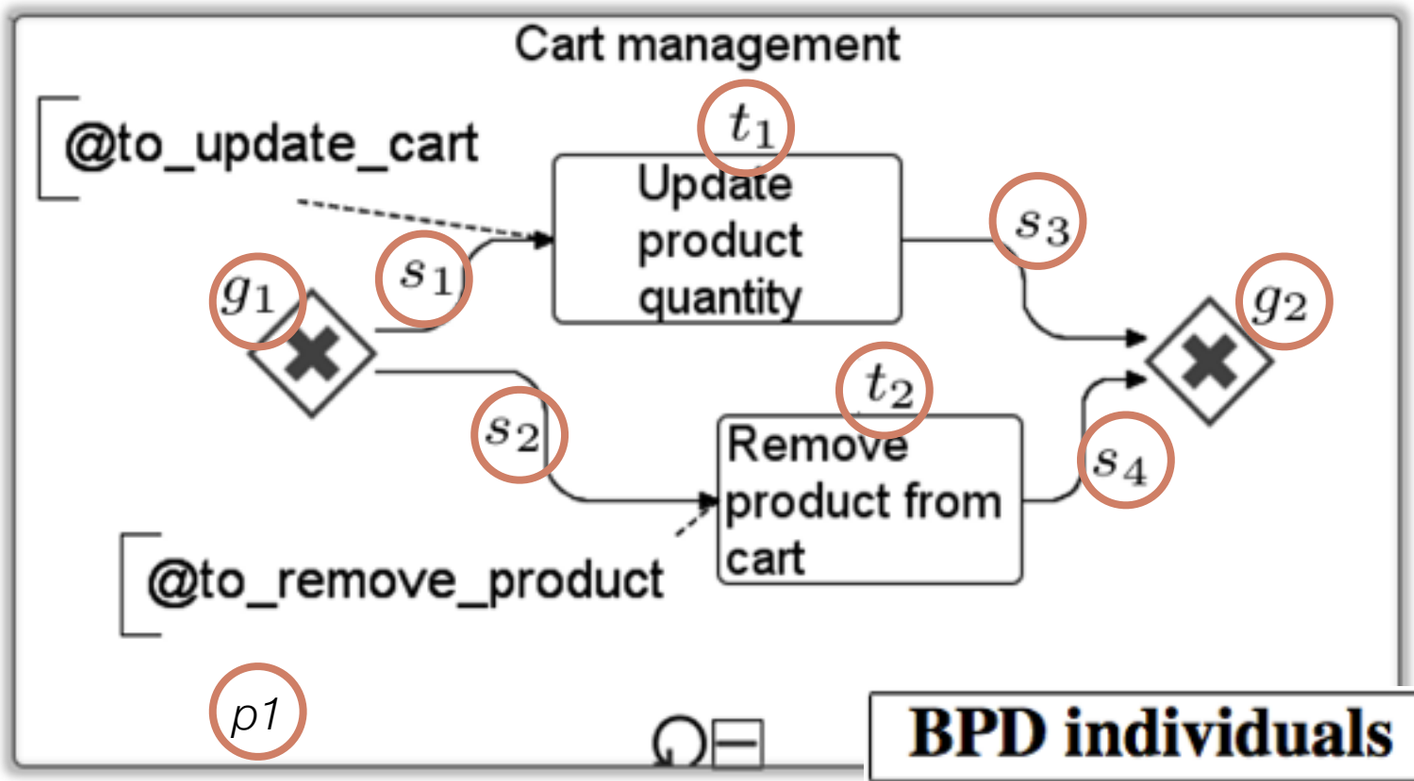


- Given a BPMN business process diagram (BPD), it is possible to represent it as an A-box in the language of the BPMN Ontology



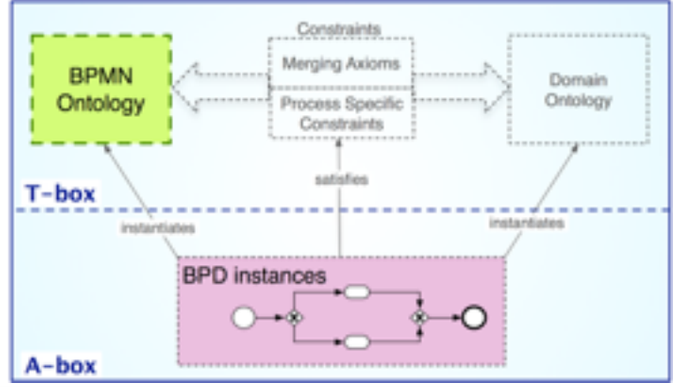
Instantiating the BPMN Ontology

- Given a BPMN business process diagram (BPD), it is possible to represent it as an A-box in the language of the BPMN Ontology

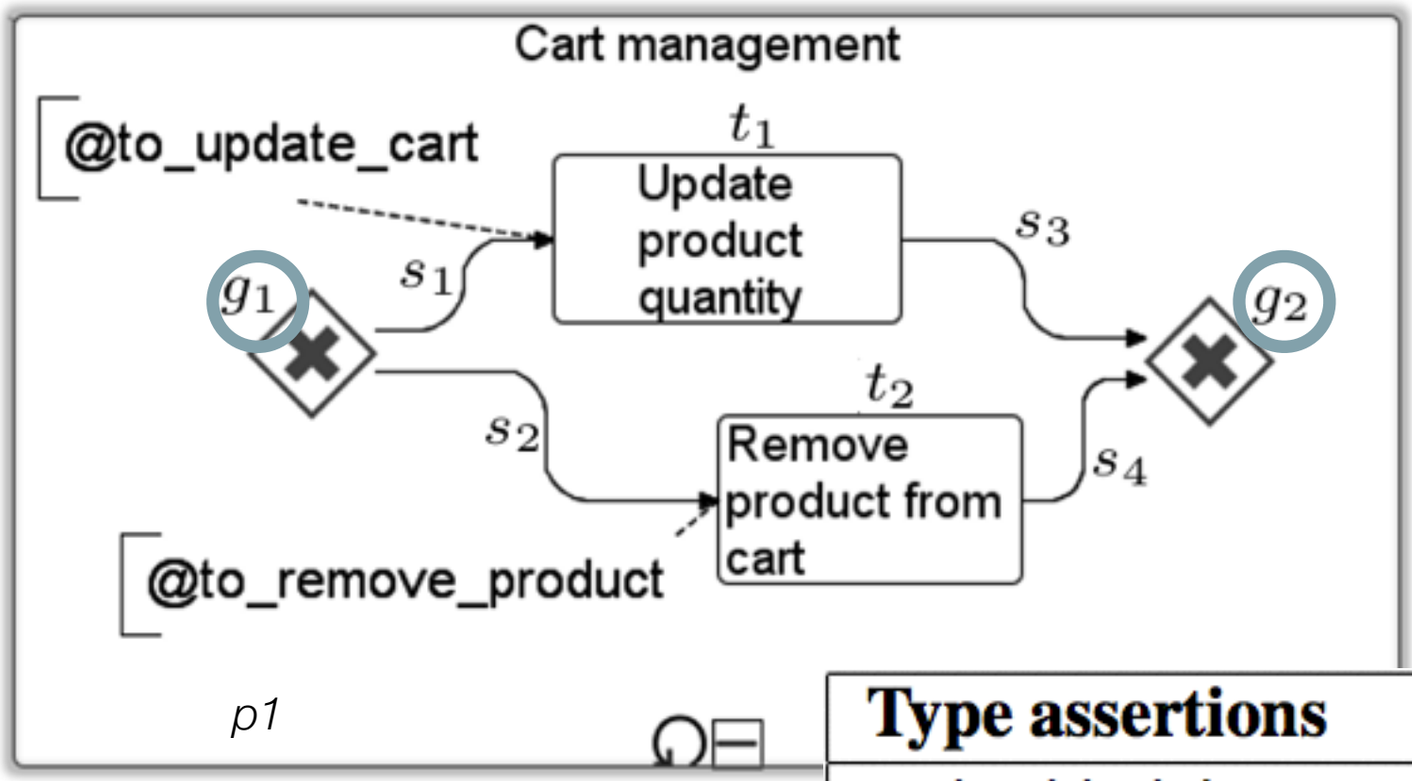


BPD individuals
p_1 corresponds to the entire subprocess
$s_1, \ldots, s_4$ correspond to the four sequence flow
g_1 and g_2 correspond to the left and the right gateways
t_1 and t_2 correspond to the top and bottom atomic task

Instantiating the BPMN Ontology

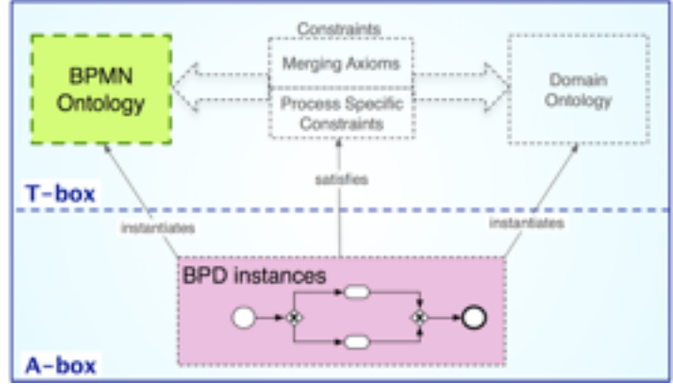


- Given a BPMN business process diagram (BPD), it is possible to represent it as an A-box in the language of the BPMN Ontology

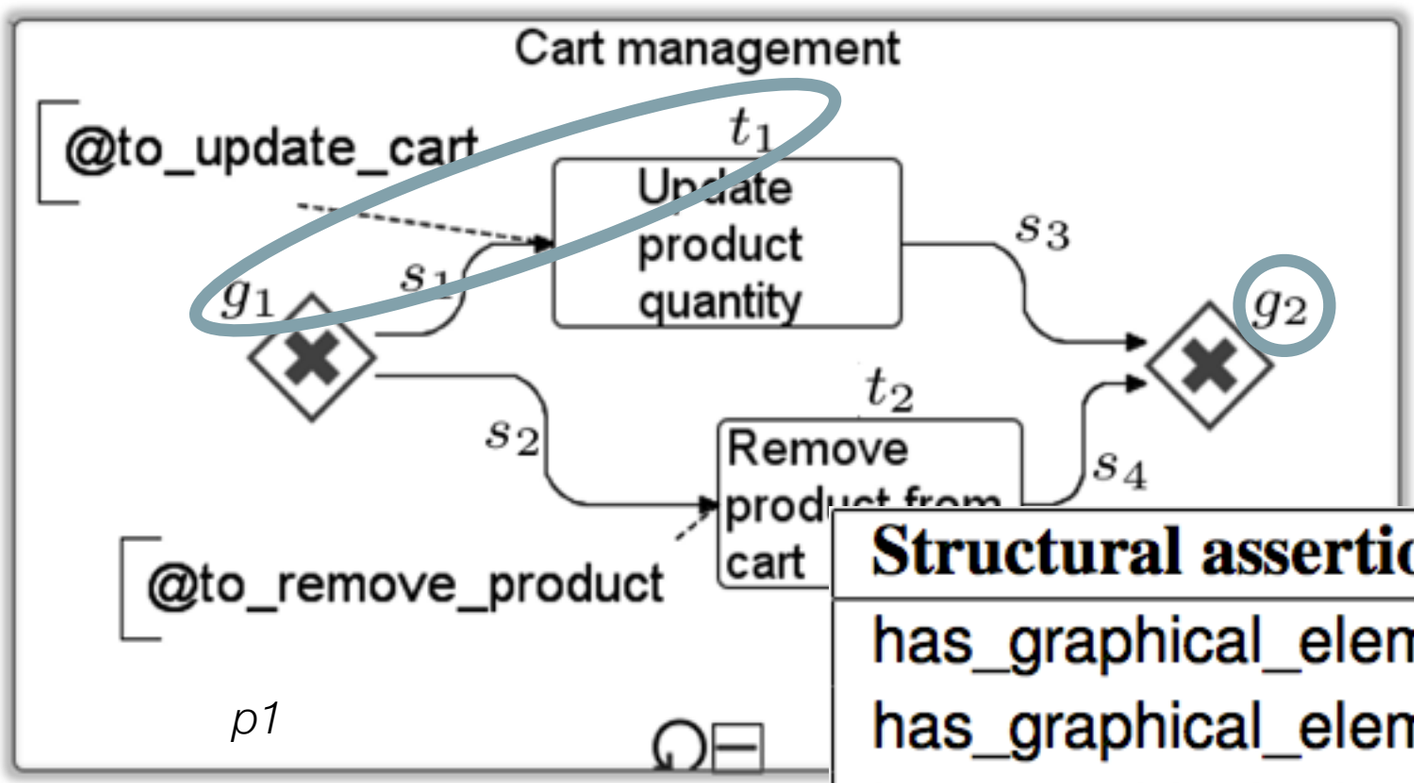


Type assertions	
embedded_loop_sub_process(p_1)	
data_based_exclusive_gateway(g_i)	$i = 1,2$
sequence_flow(s_j)	$j = 1,..,4$
task(t_k)	$k = 1,2$

Instantiating the BPMN Ontology



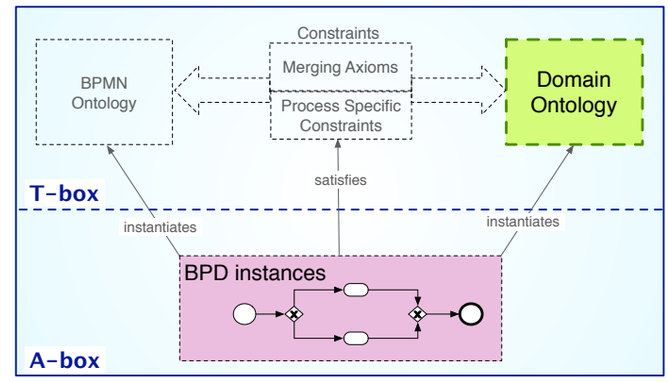
- Given a BPMN business process diagram (BPD), it is possible to represent it as an A-box in the language of the BPMN Ontology



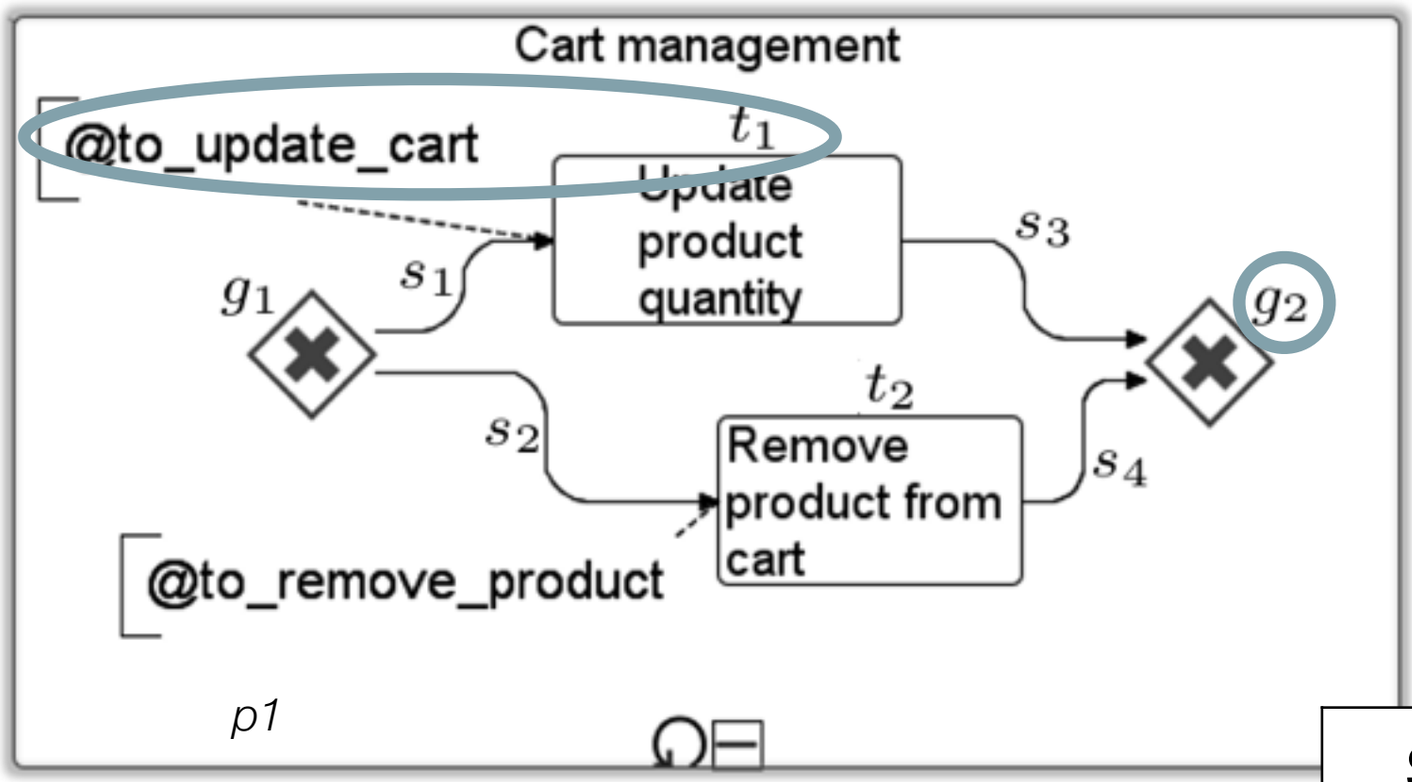
Structural assertions

has_graphical_elements(p_1, s_i)	$j = 1,..,4$
has_graphical_elements(p_1, g_i)	$i = 1,2$
has_graphical_elements(p_1, t_i)	$k = 1,2$
has_sequence_flow_source_ref(s_1, g_1)	
has_sequence_flow_target_ref(s_1, t_1)	
has_sequence_flow_source_ref(s_2, g_1)	

Instantiating the BPMN Ontology

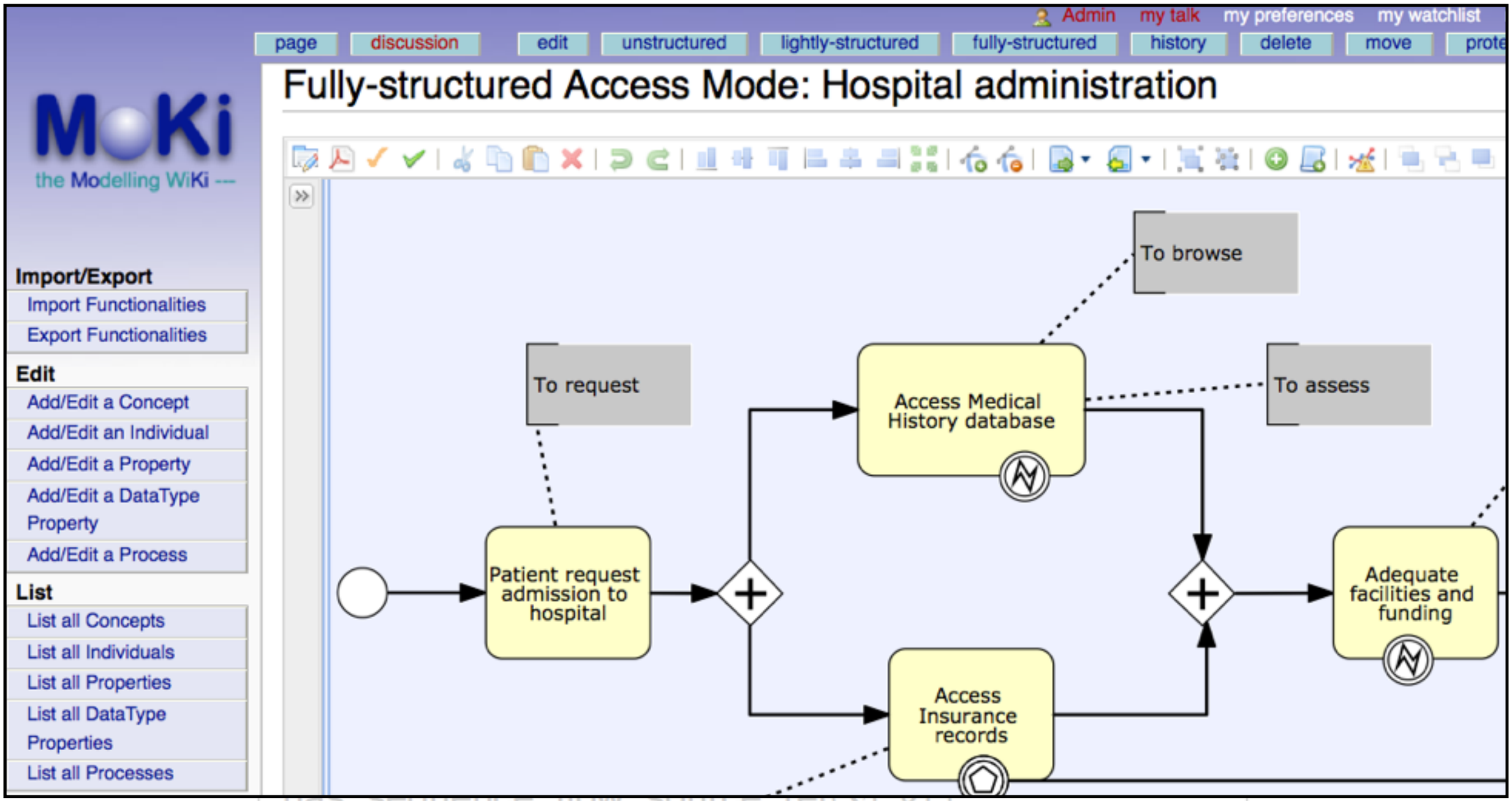


- Given a BPMN business process diagram (BPD), it is possible to represent it as an A-box in the language of the BPMN Ontology

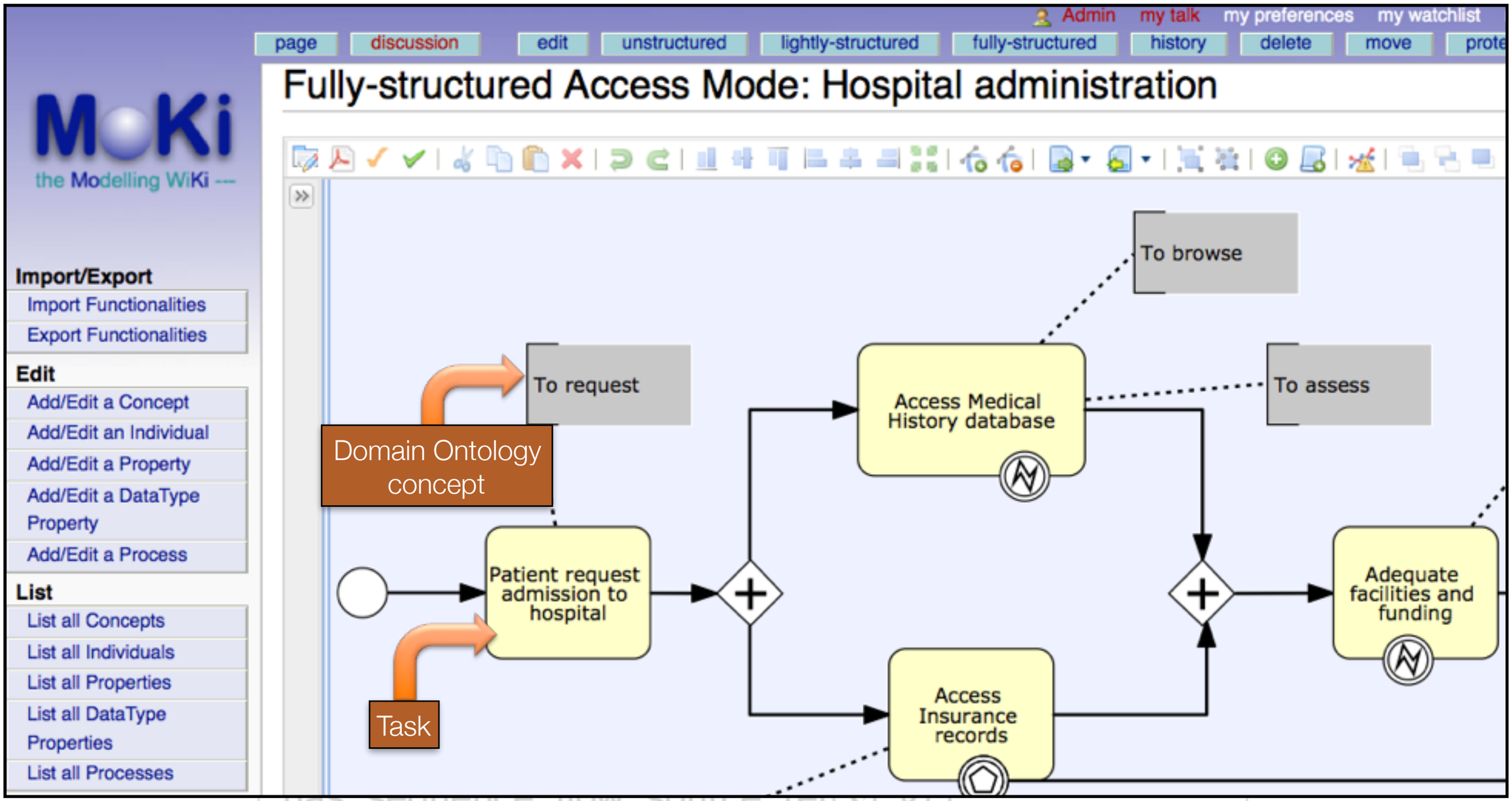


Semantic assertions
to update cart(t_1)
to remove product (t_2)

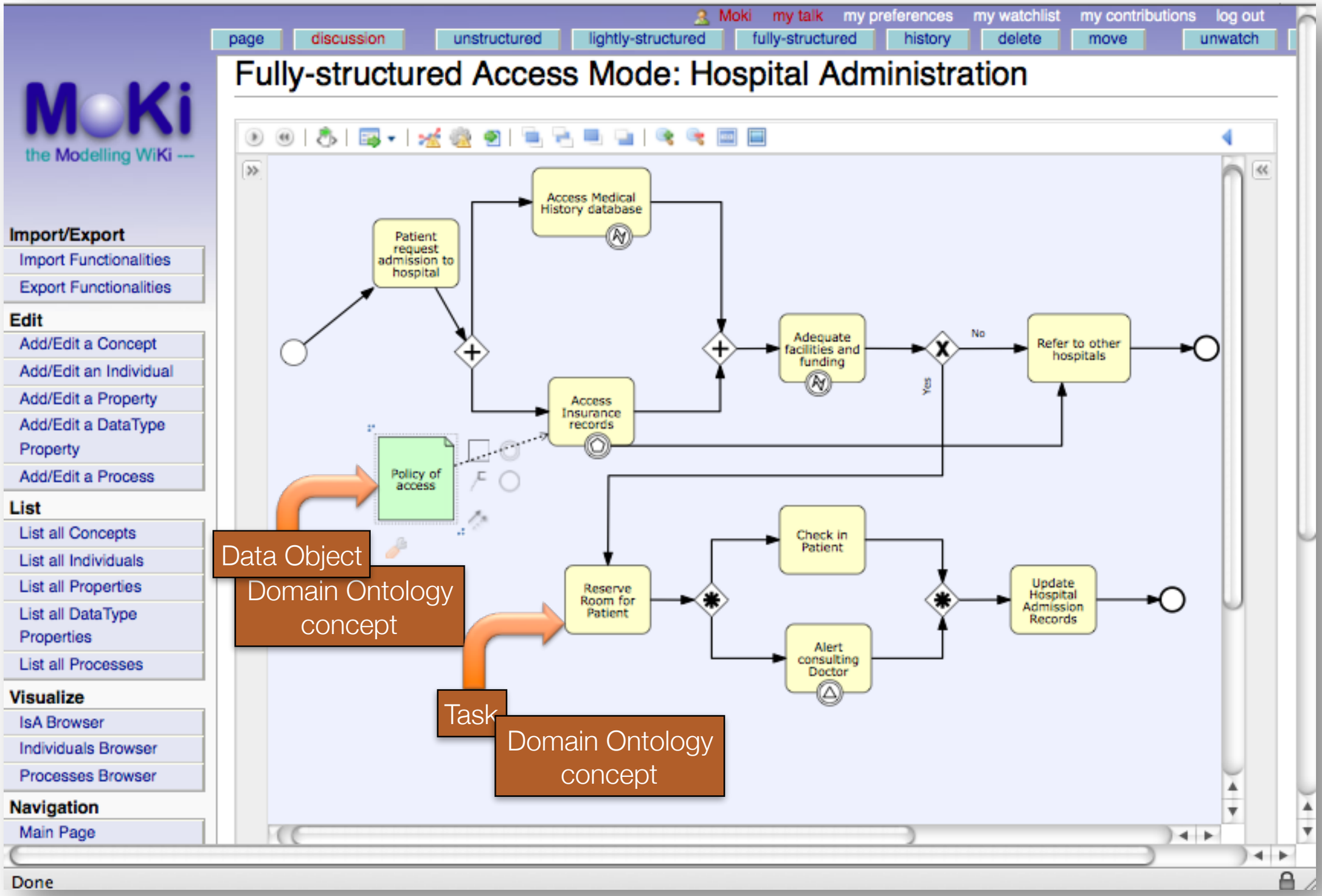
Tool support: “Compose” the diagram structure in the A-box via **MoKi**



Tool support: “Compose” the diagram structure in the A-box via **MoKi**



Tool support: “Compose” the diagram structure in the A-box via **MoKi**



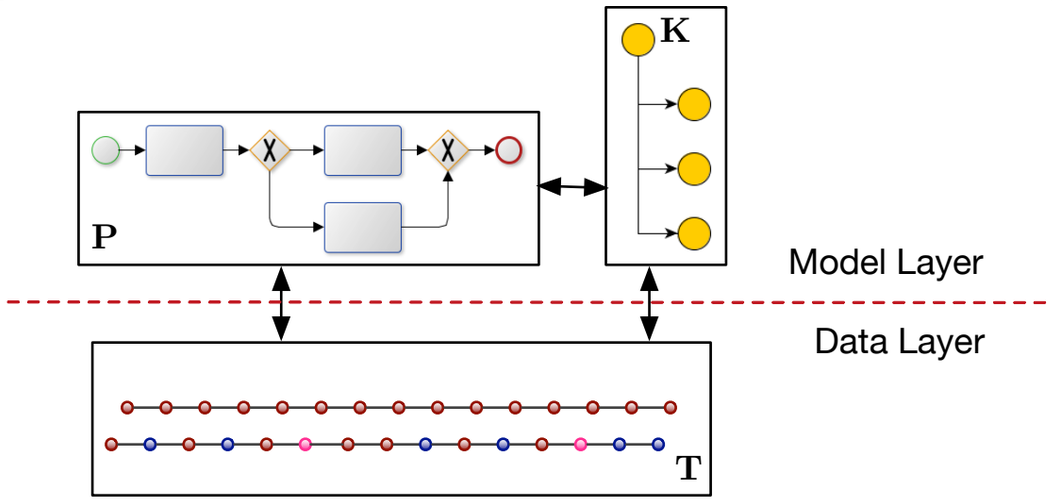
Instantiating the BPMN Ontology
Reasoning over an instantiated BPMN Ontology

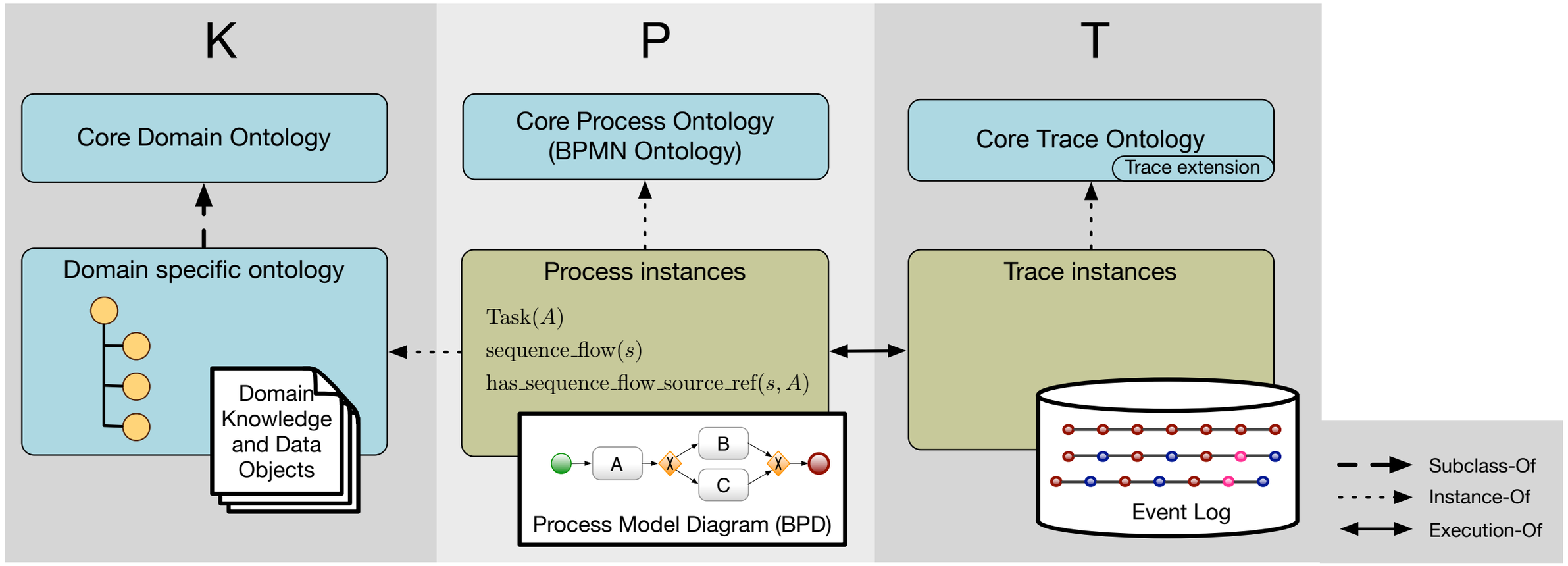
- Query answering on BPMN diagrams (via SPARQL)
  - "Which are the activities which follows gateways and produce a data object?”
  - "Are there sub-processes which do not contain start/end events?"
- Compliance checking of a BPMN diagram against the BPMN Specification
  - e.g.:

$$Gateway \sqsubseteq (\geq 2) hasSequenceFlowTarget^{-1} \sqcup ((\leq 1) hasSequenceFlowTarget^{-1} \sqcap (\geq 2) hasGatewayGate)$$

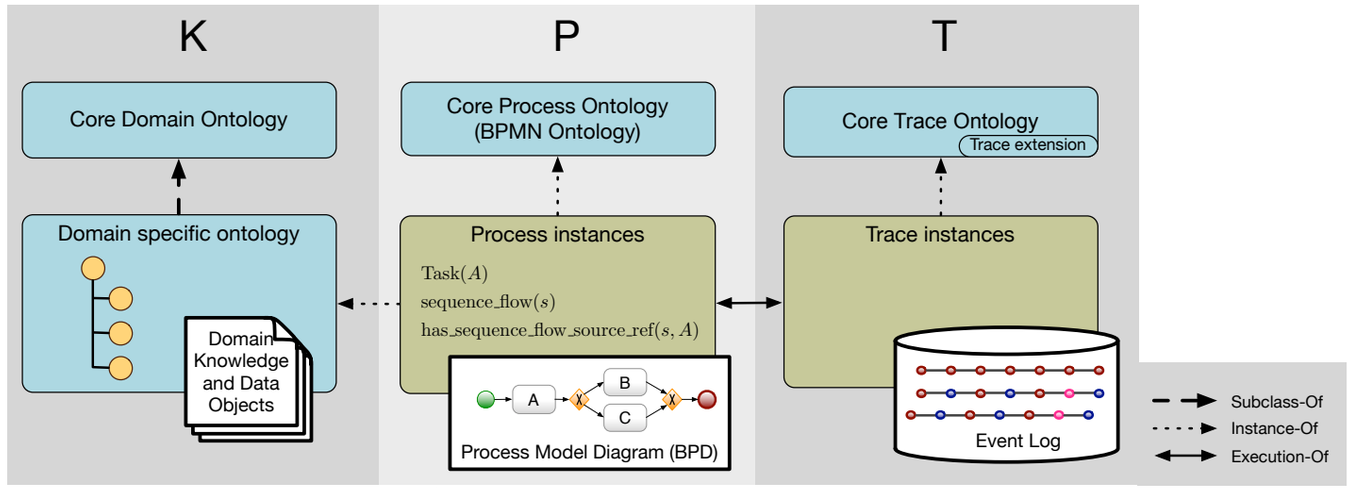
  - doable, but in closed-world assumption!

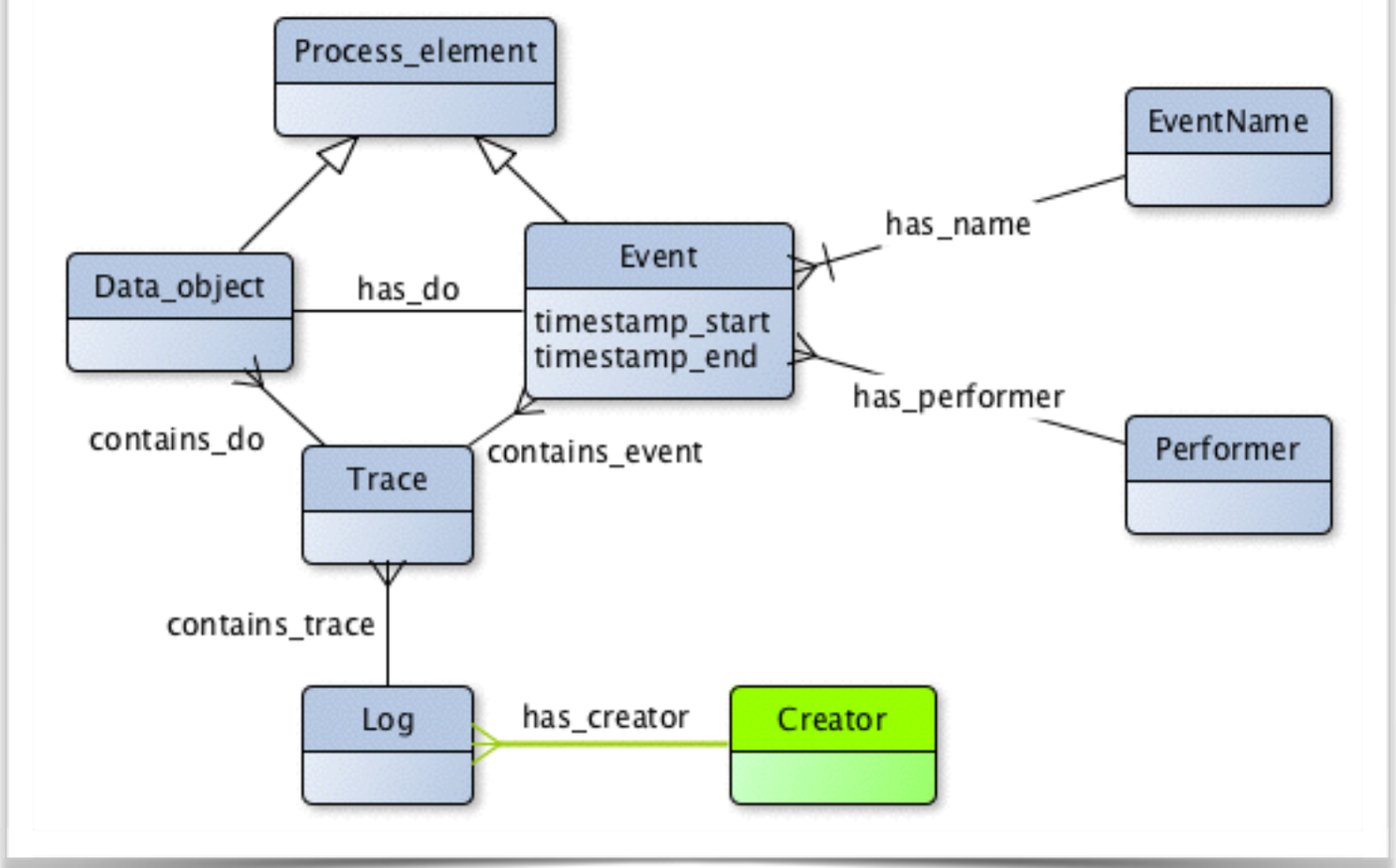
A recent extension: the execution dimension





A recent extension: the execution dimension





Based on the
XES IEEE standard

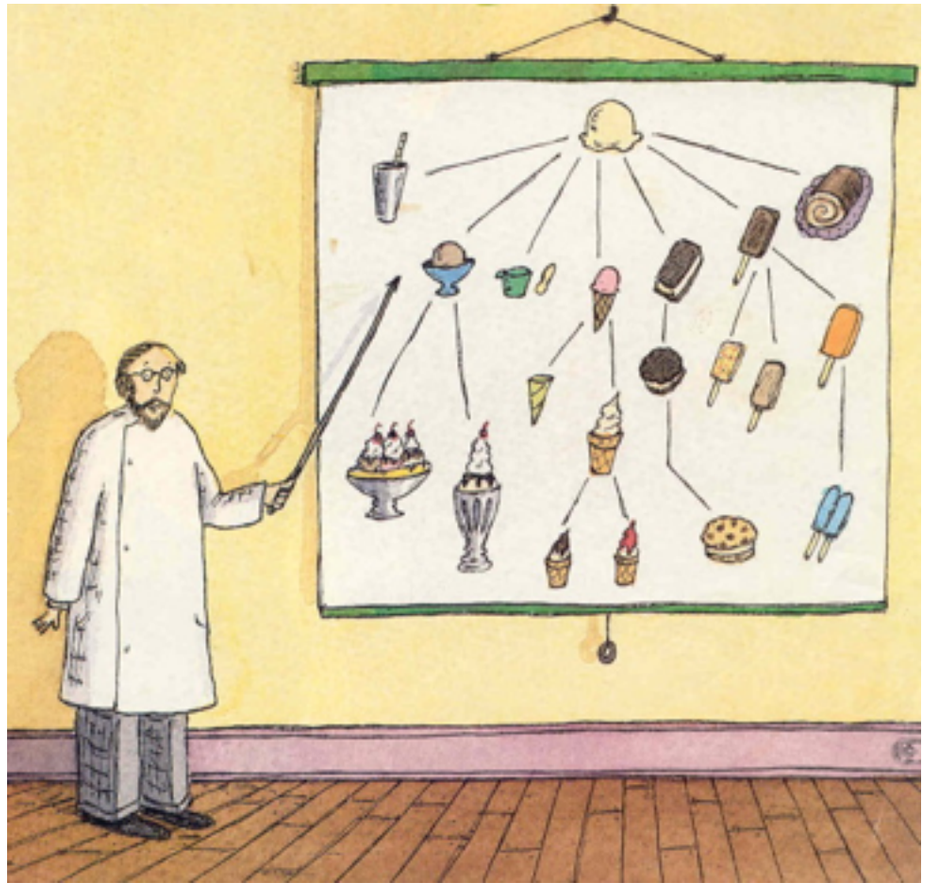
Why?

- **Process Performance Indicators**
  - PPI.1 the average time per process execution spent by the municipality of Trento;
  - PPI.2 the total number of Registration Request documents filled from January, 1st, 2014;
  - PPI.3 the percentage of times in which the flow followed is the one which passes first through the APSS pool and then through the Municipality one;
  - PPI.4 the number of cases and the average time spent by **each public office involved in the birth management procedure** for executing optional activities (i.e., activities which, taken a path on the model, can be either executed or not);

Exploiting ontologies

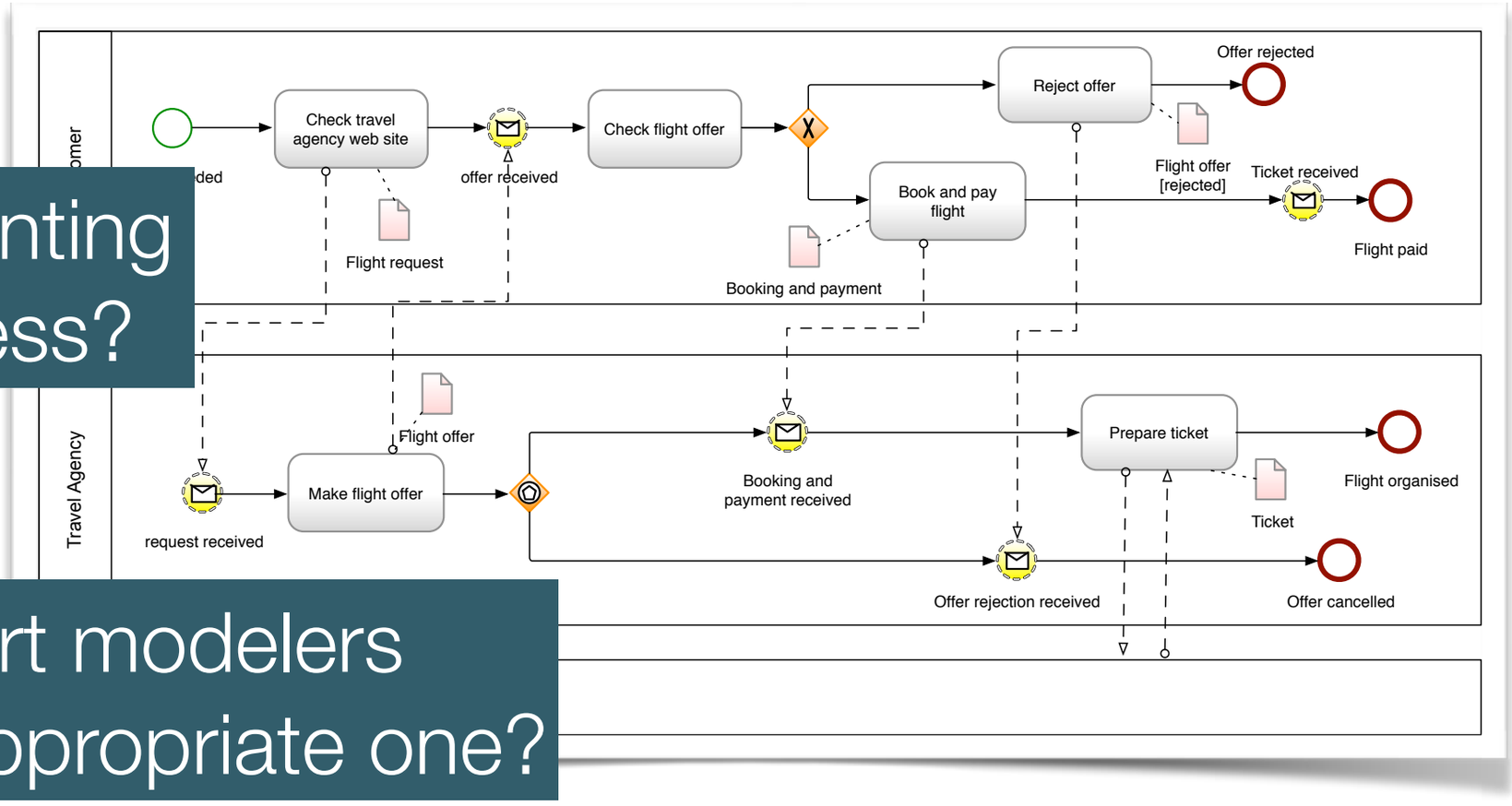
- To to compare and clarify BPM languages
- Idea 2: compare different process notations and identify challenges/problems

Joint work with Greta Adamo, Stefano Borgo,
Chiara Di Francescomarino, Nicola Guarino, Emilio Sanfilippo

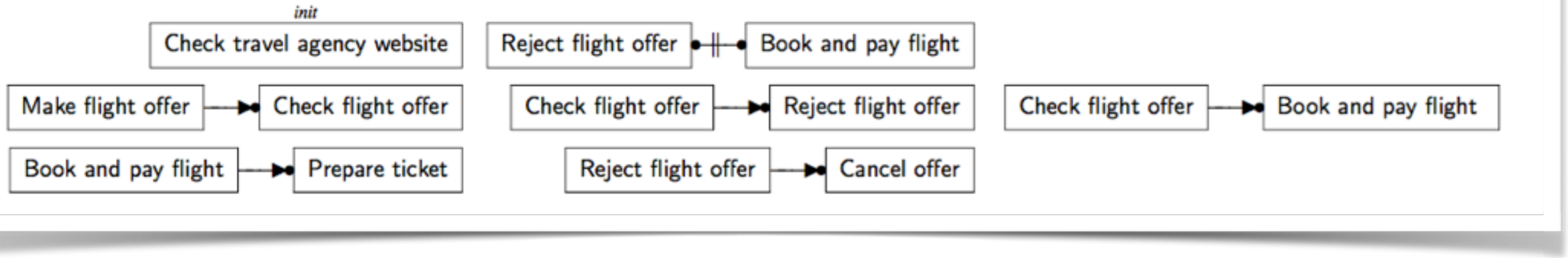


Exploiting ontologies: long term challenges

Are they representing the same process?

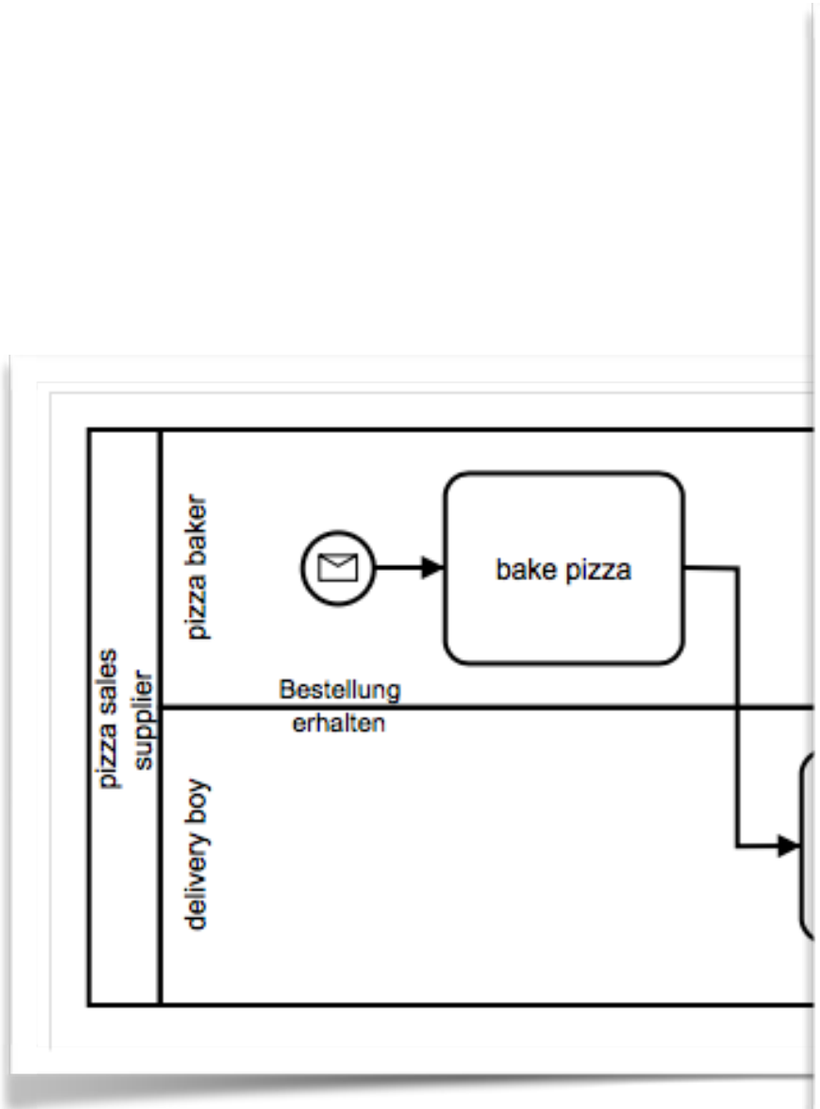


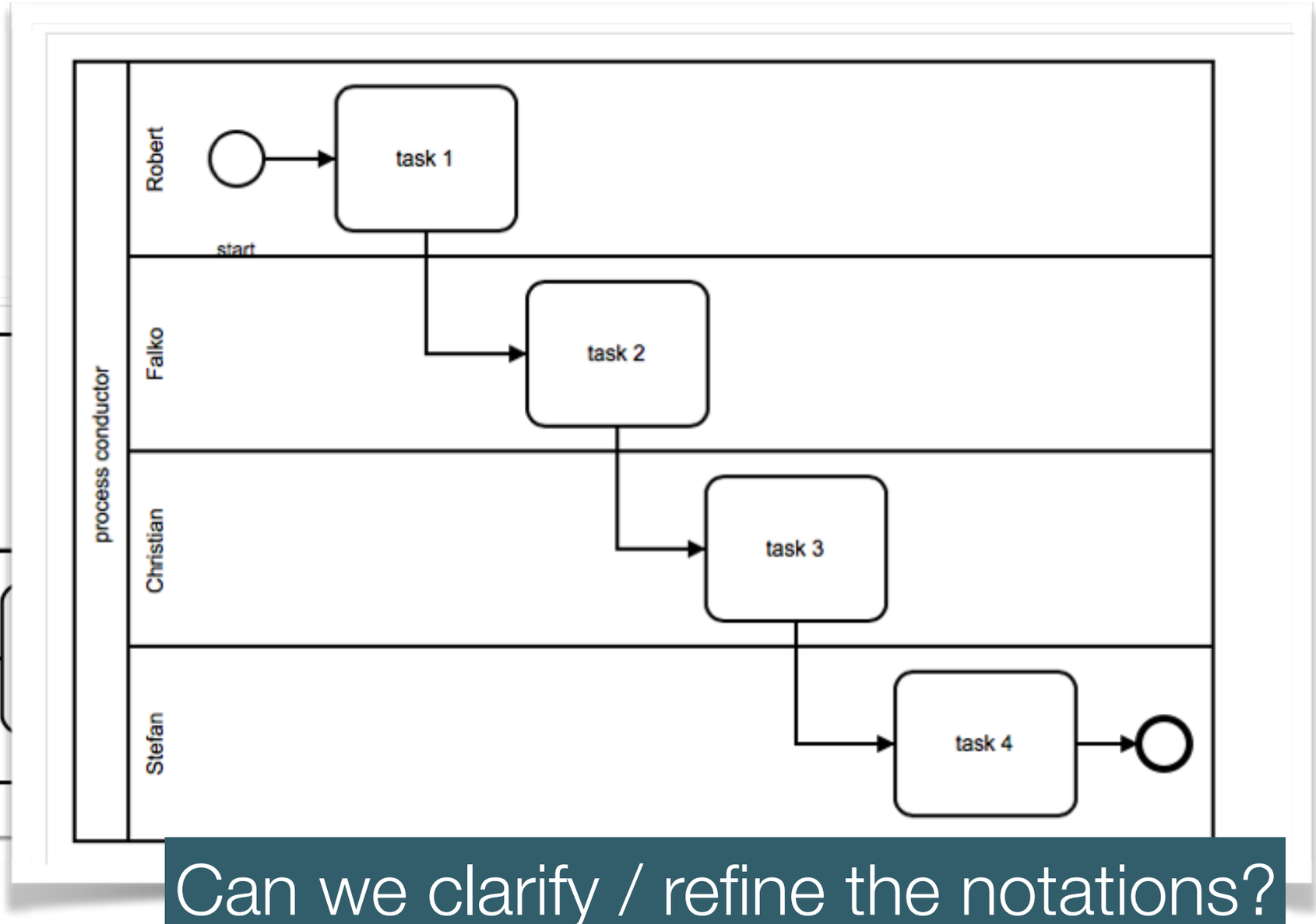
Can we support modelers to pick the most appropriate one?



Exploiting ontologies: medium term challenges

- What is the meaning of the different constructs?



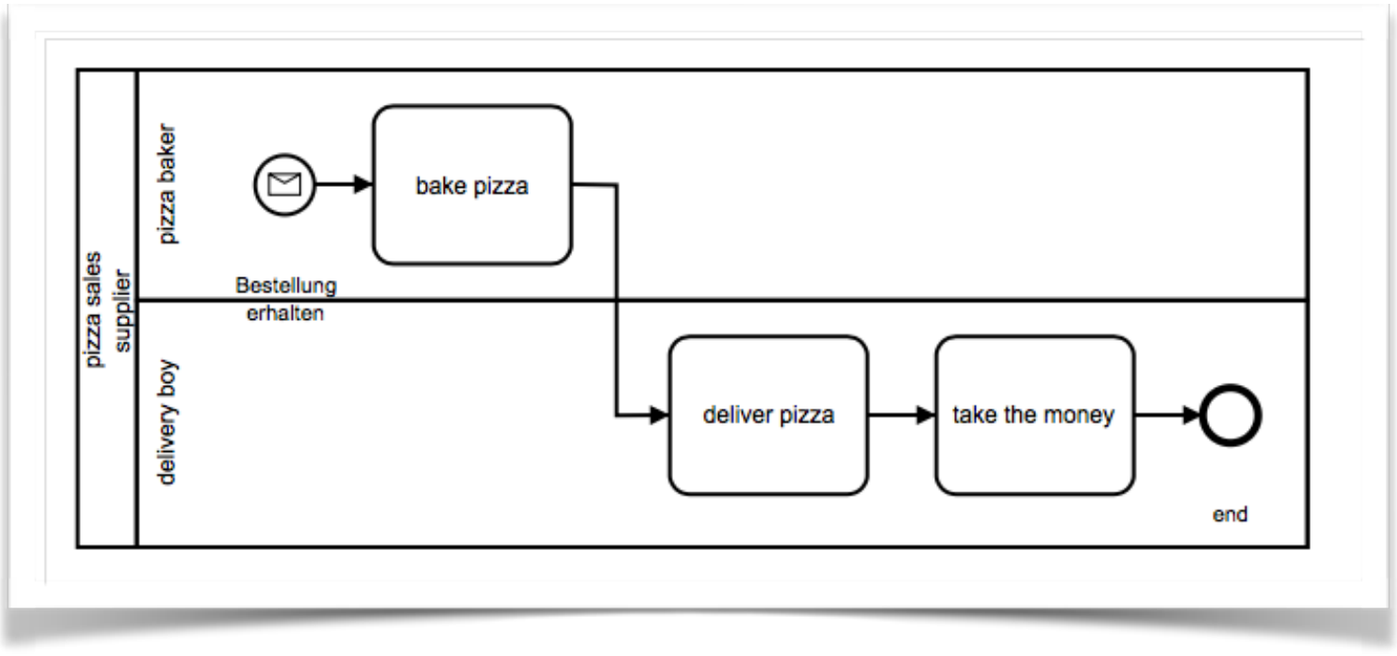


Can we clarify / refine the notations?

[See https://camunda.org/bpmn/reference]

Exploiting ontologies: medium term challenges

- What is the meaning of the different constructs?



Shall my algorithm of process repair swap bake and deliver?

[See https://camunda.org/bpmn/reference]

Work done so far

- **Aim:** starting an ontological analysis of various kinds of kinds of *process elements* and their *properties*:
    - Relation between activities (arrows)
    - Representation of the world's states (explicit or implicit)
    - Types of participants (objects, roles, data…)

Work done so far

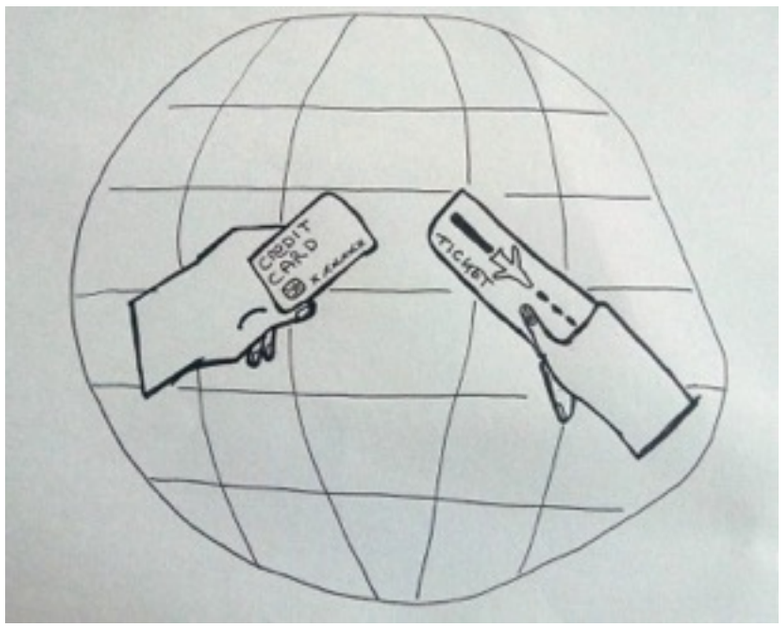
Five popular languages in B2C:

3 imperative (BPMN, UML-AD, EPC)

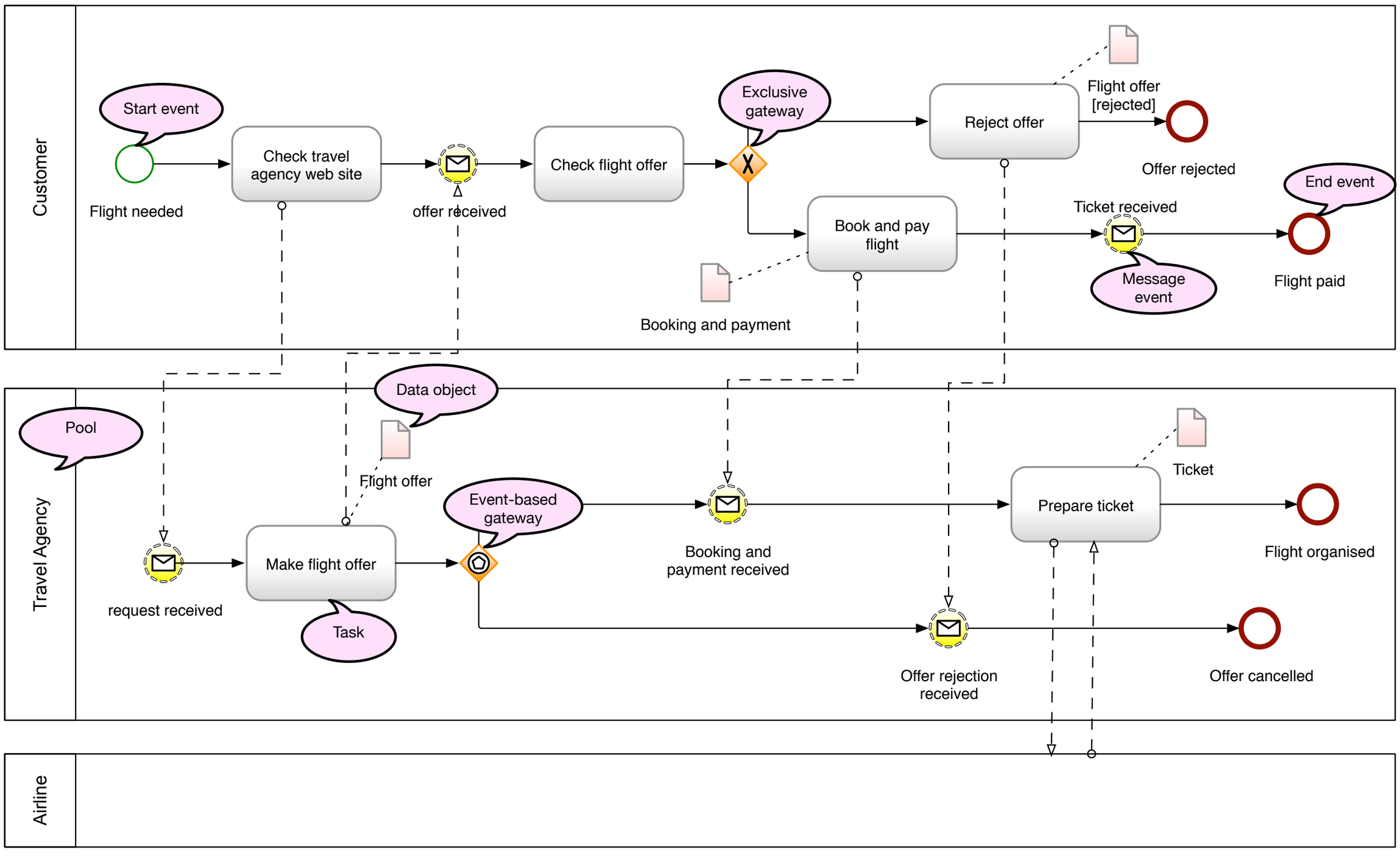
2 declarative (CMMN and DECLARE)

Simple scenario:

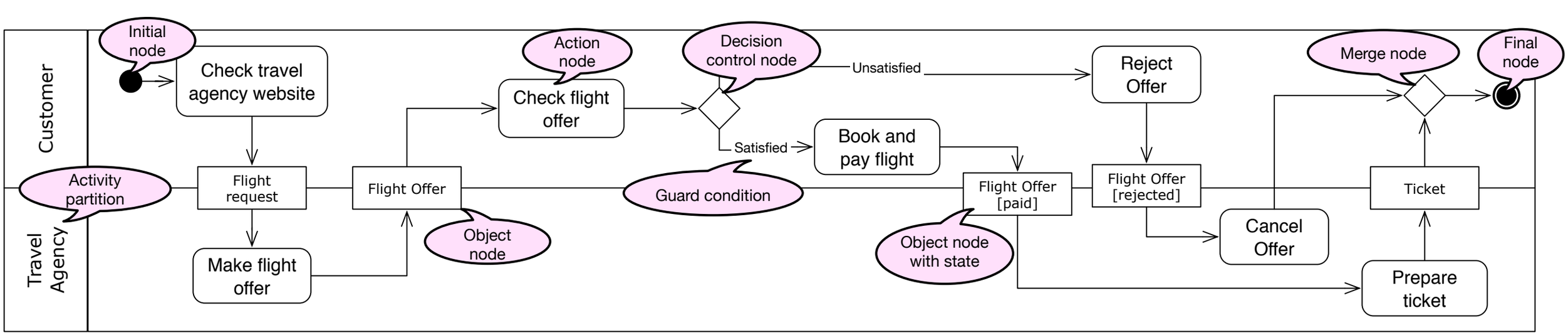
A customer buying a flight ticket from a travel agency



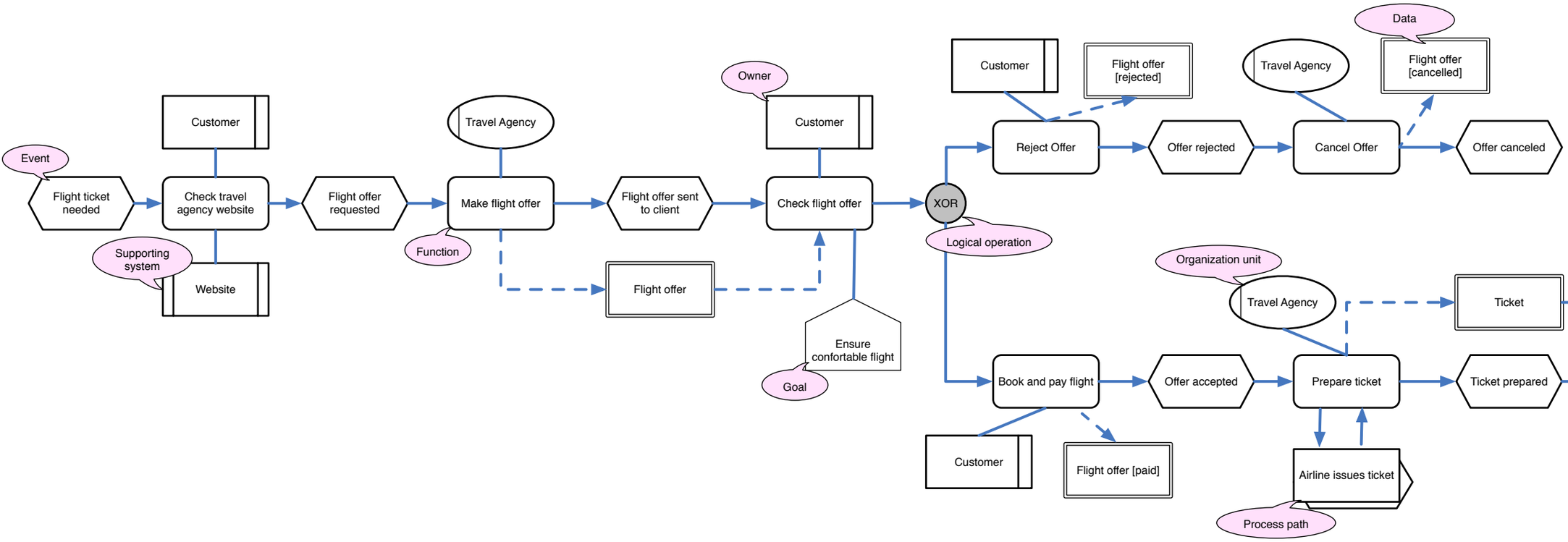
BPMN 2.0


Customer
Start event
Flight needed
Check travel agency web site
offer received
Check flight offer
Exclusive gateway
Reject offer
Flight offer [rejected]
Offer rejected
Book and pay flight
Booking and payment
Ticket received
Message event
End event
Flight paid
Travel Agency
Pool
Data object
Flight offer
request received
Make flight offer
Task
Event-based gateway
Booking and payment received
Prepare ticket
Ticket
Flight organised
Offer rejection received
Offer cancelled
Airline

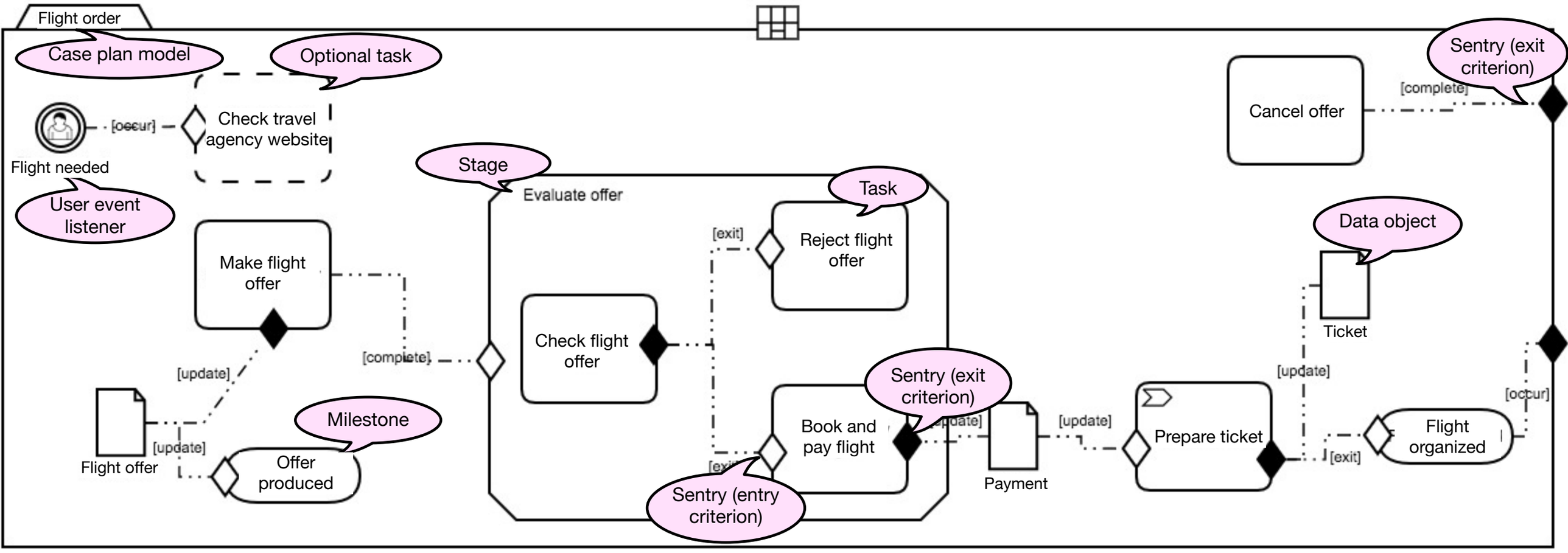
UML-AD


Customer
Travel Agency
Initial node
Check travel agency website
Activity partition
Flight request
Make flight offer
Flight Offer
Object node
Action node
Check flight offer
Decision control node
Unsatisfied
Satisfied
Guard condition
Book and pay flight
Reject Offer
Flight Offer [paid]
Object node with state
Flight Offer [rejected]
Cancel Offer
Merge node
Final node
Ticket
Prepare ticket

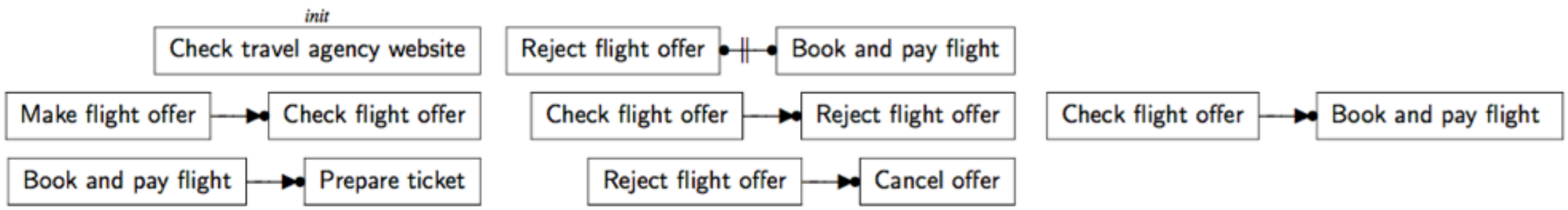
EPC


Event
Flight ticket needed
Customer
Check travel agency website
Supporting system
Website
Flight offer requested
Travel Agency
Make flight offer
Function
Flight offer sent to client
Owner
Customer
Check flight offer
Flight offer
Ensure confortable flight
Goal
XOR
Logical operation
Customer
Flight offer [rejected]
Reject Offer
Offer rejected
Travel Agency
Cancel Offer
Data
Flight offer [cancelled]
Offer canceled
Organization unit
Travel Agency
Ticket
Book and pay flight
Offer accepted
Prepare ticket
Ticket prepared
Send ticket to customer
Flight organised
Customer
Flight offer [paid]
Airline issues ticket
Process path
Travel Agency

CMMN


Flight order
Case plan model
Optional task
[occur]
Check travel agency website
Flight needed
User event listener
Make flight offer
[update]
[complete]
Flight offer
[update]
Milestone
Offer produced
Stage
Evaluate offer
Task
[exit]
Reject flight offer
Check flight offer
Book and pay flight
Sentry (exit criterion)
Sentry (entry criterion)
[update]
Payment
[update]
Prepare ticket
Cancel offer
Sentry (exit criterion)
[complete]
Data object
Ticket
[update]
[exit]
Flight organized
[occur]

DECLARE


init
Check travel agency website
Reject flight offer
Book and pay flight
Make flight offer
Check flight offer
Check flight offer
Reject flight offer
Check flight offer
Book and pay flight
Book and pay flight
Prepare ticket
Reject flight offer
Cancel offer

Comparison between language elements

The three basic categories of process modelling languages:

- **Behavioural** (BEV): Functional, Event, Flow and State
- **Data** (DT)
- **Organizational** (ORG)

Comparison between language graphical elements

		BPMN	UML-AD	EPC	CMMN	DECLARE
BEV	Func	Task Subprocess	Action node Activity	Function Process path	Task Stage	Task
	Event	Start/End Intermediate Send/receive	Start/End node Accept event action Send signal action	–	Timer User Event Listener	–
	Flow	Gateway Sequence Flow Message Flow	Control node Control Flow Object Flow	Logical operators Control Flow Info Flow	Connector Sentry	Connector Pattern
	State	Guard on gateway	Guard on control node Pre- Post-condition on activity	Event Start/End event	Sentry Milestone	–
DT		Data input data output data store	Object node	(I/O) data object	Case file item	–
ORG		Pool, Lane	Activity Partition	Organization Activity Owner	–	

Comparison between language graphical elements

		BPMN	UML-AD	EPC	CMMN	DECLARE
BEV	Func	Task Subprocess	Action node Activity	Function Process path	Task Stage	Task
	Event	Start/End Intermediate Send/receive	Start/End node Accept event action Send signal action	–	Timer User Event Listener	–
	Flow	Gateway Sequence Flow Message Flow	Control node Control Flow Object Flow	Logical operators Control Flow Info Flow	Connector Sentry	Connector Pattern
	State	Guard on gateway	Guard on control node Pre- Post-condition on activity	Event Start/End event	Sentry Milestone	–
DT		Data input data output data store	Object node	(I/O) data object	Case file item	–
ORG		Pool, Lane	Activity Partition	Organization Activity Owner	–	

(Explicit) start / end

Comparison between language graphical elements

		BPMN	UML-AD	EPC	CMMN	DECLARE
BEHAV	Func	Task Subprocess	Action node Activity	Function Process path	Task Stage	Task
	Event	Start/End Intermediate Send/receive	Start/End node Accept event action Send signal action	–	Timer User Event Listener	–
	Flow	Gateway Sequence Flow Message Flow	Control node Control Flow Object Flow	Logical operators Control Flow Info Flow	Connector Sentry	Connector Pattern
	State	Guard on gateway	Guard on control node Pre- Post-condition on activity	Event Start/End event	Sentry Milestone	–
DT		Data input data output data store	Object node	(I/O) data object	Case file item	–
ORG		Pool, Lane	Activity Partition	Organization Activity Owner	–	

(Explicit) start / end

Atomic Activities

Comparison between language graphical elements

		BPMN	UML-AD	EPC	CMMN	DECLARE
BEV	Func	Task Subprocess	Action node Activity	Function Process path	Task Stage	Task
	Event	Start/End Intermediate Send/receive	Start/End node Accept event action Send signal action	–	Timer User Event Listener	–
	Flow	Gateway Sequence Flow Message Flow	Control node Control Flow Object Flow	Logical operators Control Flow Info Flow	Connector Sentry	Connector Pattern
	State	Guard on gateway	Guard on control node Pre- Post-condition on activity	Event Start/End event	Sentry Milestone	–
DT		Data input data output data store	Object node	(I/O) data object	Case file item	–
ORG		Pool, Lane	Activity Partition	Organization Activity Owner	–	

(Explicit) start / end

Atomic Activities

Complex Activities

Comparison between language graphical elements

		BPMN	UML-AD	EPC	CMMN	DECLARE
BEV	Func	Task Subprocess	Action node Activity	Function Process path	Task Stage	Task
	Event	Start/End Intermediate Send/receive	Start/End node Accept event action Send signal action	–	Timer User Event Listener	–
	Flow	Gateway Sequence Flow Message Flow	Control node Control Flow Object Flow	Logical operators Control Flow Info Flow	Connector Sentry	Connector Pattern
	State	Guard on gateway	Guard on control node Pre- Post-condition on activity	Event Start/End event	Sentry Milestone	–
DT		Data input data output data store	Object node	(I/O) data object	Case file item	–
ORG		Pool, Lane	Activity Partition	Organization Activity Owner	–	

(Explicit) start / end

Routing nodes

Atomic Activities

Complex Activities

Comparison between language graphical elements

		BPMN	UML-AD	EPC	CMMN	DECLARE
BEV	Func	Task Subprocess	Action node Activity	Function Process path	Task Stage	Task
	Event	Start/End Intermediate Send/receive	Start/End node Accept event action Send signal action	–	Timer User Event Listener	–
	Flow	Gateway Sequence Flow Message Flow	Control node Control Flow Object Flow	Logical operators Control Flow Info Flow	Connector Sentry	Connector Pattern
	State	Guard on gateway	Guard on control node Pre- Post-condition on activity	Event Start/End event	Sentry Milestone	–
DT		Data input data output data store	Object node	(I/O) data object	Case file item	–
ORG		Pool, Lane	Activity Partition	Organization Activity Owner	–	

(Explicit) start / end

Routing nodes

Atomic Activities

Connectors

Complex Activities

Comparison between language graphical elements

		BPMN	UML-AD	EPC	CMMN	DECLARE
BEV	Func	Task Subprocess	Action node Activity	Function Process path	Task Stage	Task
	Event	Start/End Intermediate Send/receive	Start/End node Accept event action Send signal action	–	Timer User Event Listener	–
	Flow	Gateway Sequence Flow Message Flow	Control node Control Flow Object Flow	Logical operators Control Flow Info Flow	Connector Sentry	Connector Pattern
	State	Guard on gateway	Guard on control node Pre- Post-condition on activity	Event Start/End event	Sentry Milestone	–
DT		Data input data output data store	Object node	(I/O) data object	Case file item	–
ORG		Pool, Lane	Activity Partition	Organization Activity Owner	–	

(Explicit) start / end

Routing nodes

Atomic Activities

Connectors

Complex Activities

State

Comparison between language graphical elements

		BPMN	UML-AD	EPC	CMMN	DECLARE
BEV	Func	Task Subprocess	Action node Activity	Function Process path	Task Stage	Task
	Event	Start/End Intermediate Send/receive	Start/End node Accept event action Send signal action	–	Timer User Event Listener	–
	Flow	Gateway Sequence Flow Message Flow	Control node Control Flow Object Flow	Logical operators Control Flow Info Flow	Connector Sentry	Connector Pattern
	State	Guard on gateway	Guard on control node Pre- Post-condition on activity	Event Start/End event	Sentry Milestone	–
DT		Data input data output data store	Object node	(I/O) data object	Case file item	–
ORG		Pool, Lane	Activity Partition	Organization Activity Owner	–	

(Explicit) start / end

Atomic Activities

Complex Activities

Routing nodes

Connectors

State

Guards

Comparison between language graphical elements

		BPMN	UML-AD	EPC	CMMN	DECLARE
BEV	Func	Task Subprocess	Action node Activity	Function Process path	Task Stage	Task
	Event	Start/End Intermediate Send/receive	Start/End node Accept event action Send signal action	–	Timer User Event Listener	–
	Flow	Gateway Sequence Flow Message Flow	Control node Control Flow Object Flow	Logical operators Control Flow Info Flow	Connector Sentry	Connector Pattern
	State	Guard on gateway	Guard on control node Pre- Post-condition on activity	Event Start/End event	Sentry Milestone	–
DT		Data input data output data store	Object node	(I/O) data object	Case file item	–
ORG		Pool, Lane	Activity Partition	Organization Activity Owner	–	

(Explicit) start / end

Routing nodes

Guards

Atomic Activities

Connectors

Data objects

Complex Activities

State

Comparison between language graphical elements

		BPMN	UML-AD	EPC	CMMN	DECLARE
BEV	Func	Task	Action node	Function	Task	Task
		Subprocess	Activity	Process path	Stage	
	Event	Start/End	Start/End node	–	Timer	–
		Intermediate	Accept event action		User Event Listener	
		Send/receive	Send signal action			
	Flow	Gateway	Control node	Logical operators	Connector	Connector
		Sequence Flow	Control Flow	Control Flow	Sentry	Pattern
		Message Flow	Object Flow	Info Flow		
	State	Guard on gateway	Guard on control node	Event	Sentry	–
			Pre- Post-condition on activity	Start/End event	Milestone	
DT		Data input	Object node	(I/O) data object	Case file item	–
		data output				
		data store				
ORG		Pool, Lane	Activity Partition	Organization	–	
				Activity Owner		

(Explicit) start / end

Routing nodes

Guards

Atomic Activities

Connectors

Data objects

Complex Activities

State

Organizational/ownership

Comparison between language graphical elements

		BPMN	UML-AD	EPC	CMMN	DECLARE
BEV	Func	Task Subprocess	Action node Activity	Function Process path	Task Stage	Task
	Event	Start/End Intermediate Send/receive	Start/End node Accept event action Send signal action	–	Timer User Event Listener	–
	Flow	Gateway Sequence Flow Message Flow	Control node Control Flow Object Flow	Logical operators Control Flow Info Flow	Connector Sentry	Connector Pattern
	State	Guard on gateway	Guard on control node Pre- Post-condition on activity	Event Start/End event	Sentry Milestone	–
DT		Data input data output data store	Object node	(I/O) data object	Case file item	–
ORG		Pool, Lane	Activity Partition	Organization Activity Owner	–	

(imperative) Languages are rich in symbols!

and so….

- are these symbols what is needed to describe a process?
- is their intended semantics clear?

What does the ontological analysis tell us of them?

What is a business process?

a structured, measured set of **activities** designed to produce a specific **output** for a particular customer or market. [...] A process is thus a **specific ordering** of work **activities** across time and space, with a **beginning** and an **end**, and clearly defined **inputs** and **outputs**

T. Davenport. Process Innovation: Reengineering work through information technology. 1993.

a collection of **activities** that takes one or more kinds of **input** and creates an **output** that is of **value** to the customer

M. Hammer and J. Champy. Reengineering the Corporation: A Manifesto for Business Revolution. 1993.

a set of linked **activities** that take an **input** and transform it to create an **output**. Ideally, the transformation that occurs in the process should add **value** to the input

H. J. Johansson, P. McHugh, A. J. Pendlebury, and W. A. Wheeler.
Business Process Reengineering: Breakpoint Strategies for Market Dominance. 1993.

What is a business process?

a set of **activities** that are performed in coordination in an **organizational** and technical environment. These activities jointly realize a **business goal**. Each business process is enacted by a single **organization**, but it may interact with business processes performed by other **organizations**

M. Weske. Business Process Management. Concepts, Languages, Architectures. 2012.

What is a business process

a set of **activities** that are performed in coordination in an **organizational** and technical environment. These activities jointly realize a **business goal**. Each business process is enacted by a single **organization**, but it may interact with business processes performed by other **organizations**

M. Weske. Business Process Management. Concepts, Languages, Architectures. 2012.

1. Activities
2. Input
3. Output
4. Value
5. Organisational boundaries

What does the ontological analysis tell us of them?

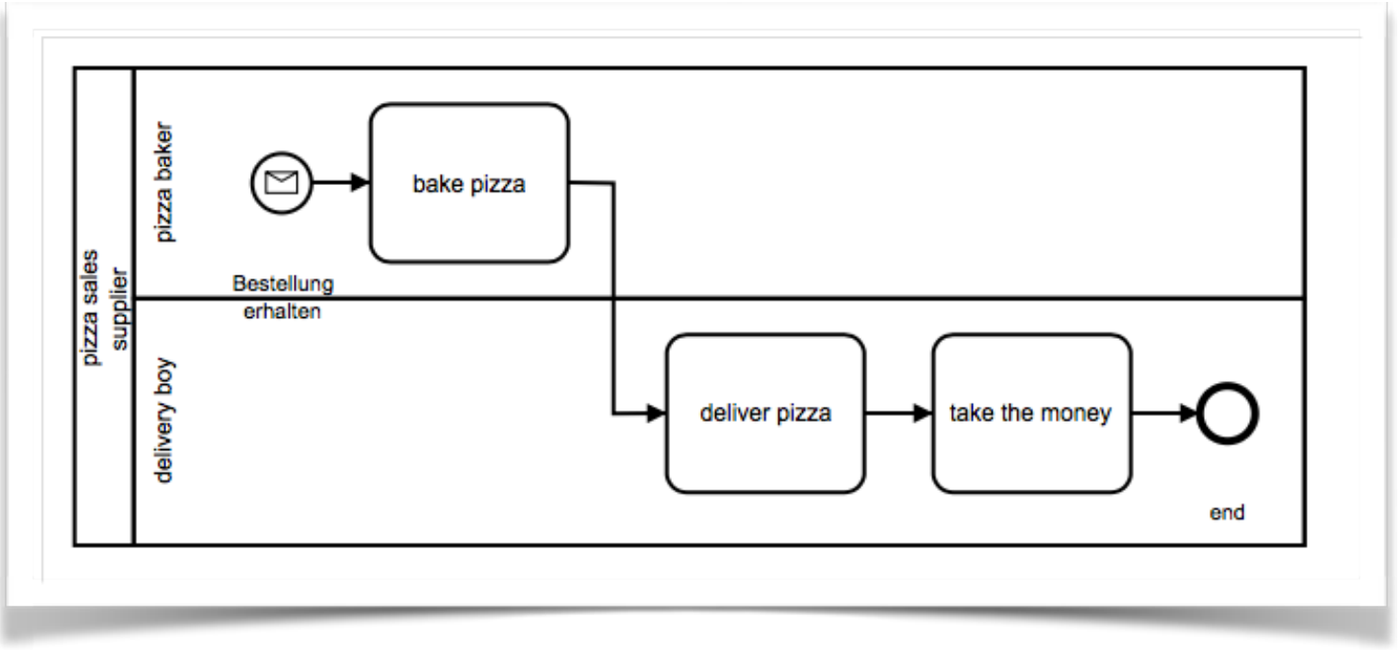
Activities

- In BPM activities are (atomic or compound) actions, consisting of intentional transformations from some initial state (the input) to some other state (the output). The participants to such actions are the entities that take part in these transformations.

- In ontological analysis actions are (specific kinds of) *events*, while their participants are *objects.*

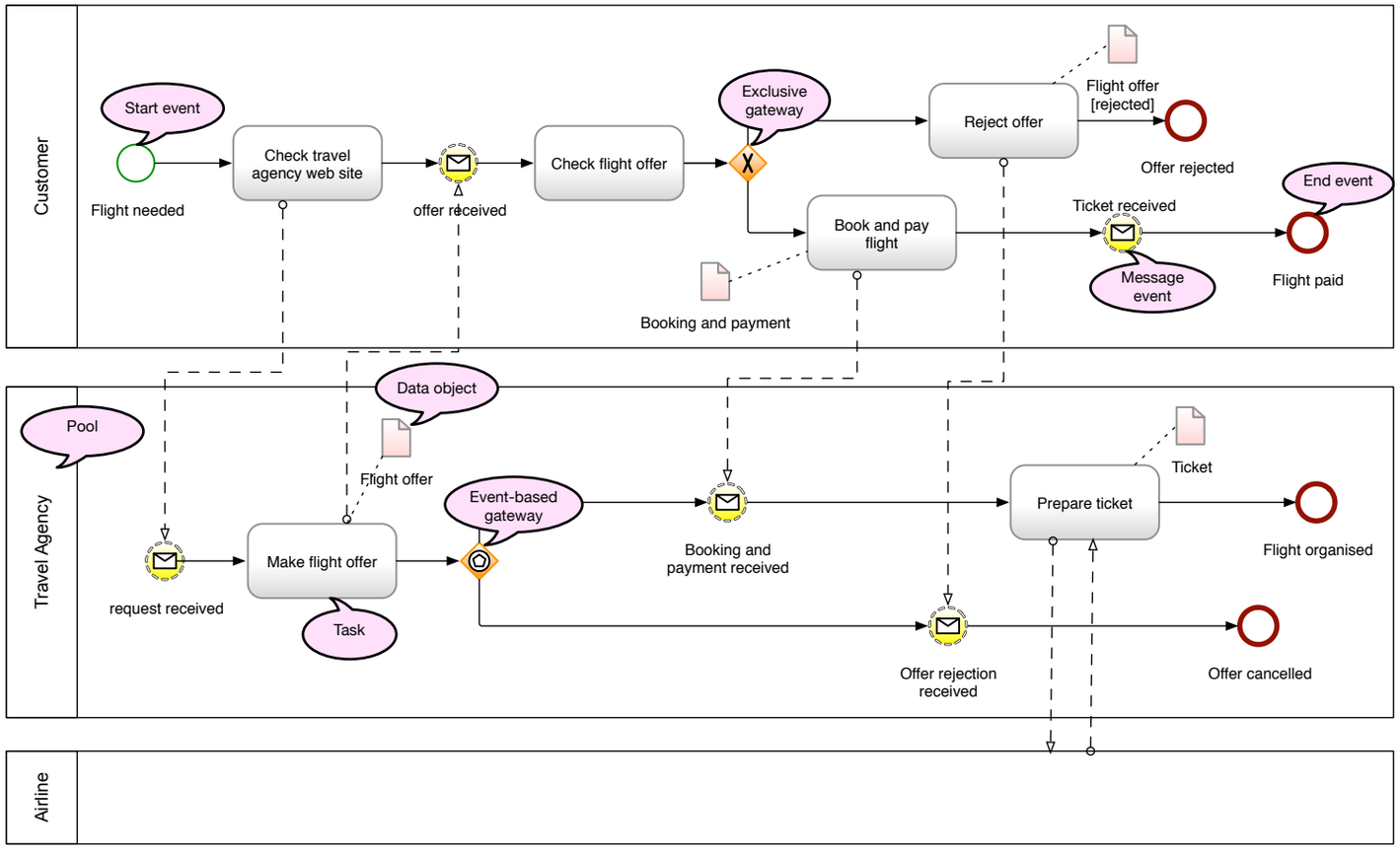
Activities - challenges

- relations between activities
    - temporal, causal, constraints, … ?
      Can the ontological analysis help us distinguish?



Activities - challenges

- State of the world
    - is the (implicit or explicit) representation of the state of the world necessary to fully characterise a process (model)?



Participants

1. **Physical** participants: located in the physical space (e.g. *person*, *computer*)

2. **Non-physical** participants: lack physical locations (e.g. *information object*)

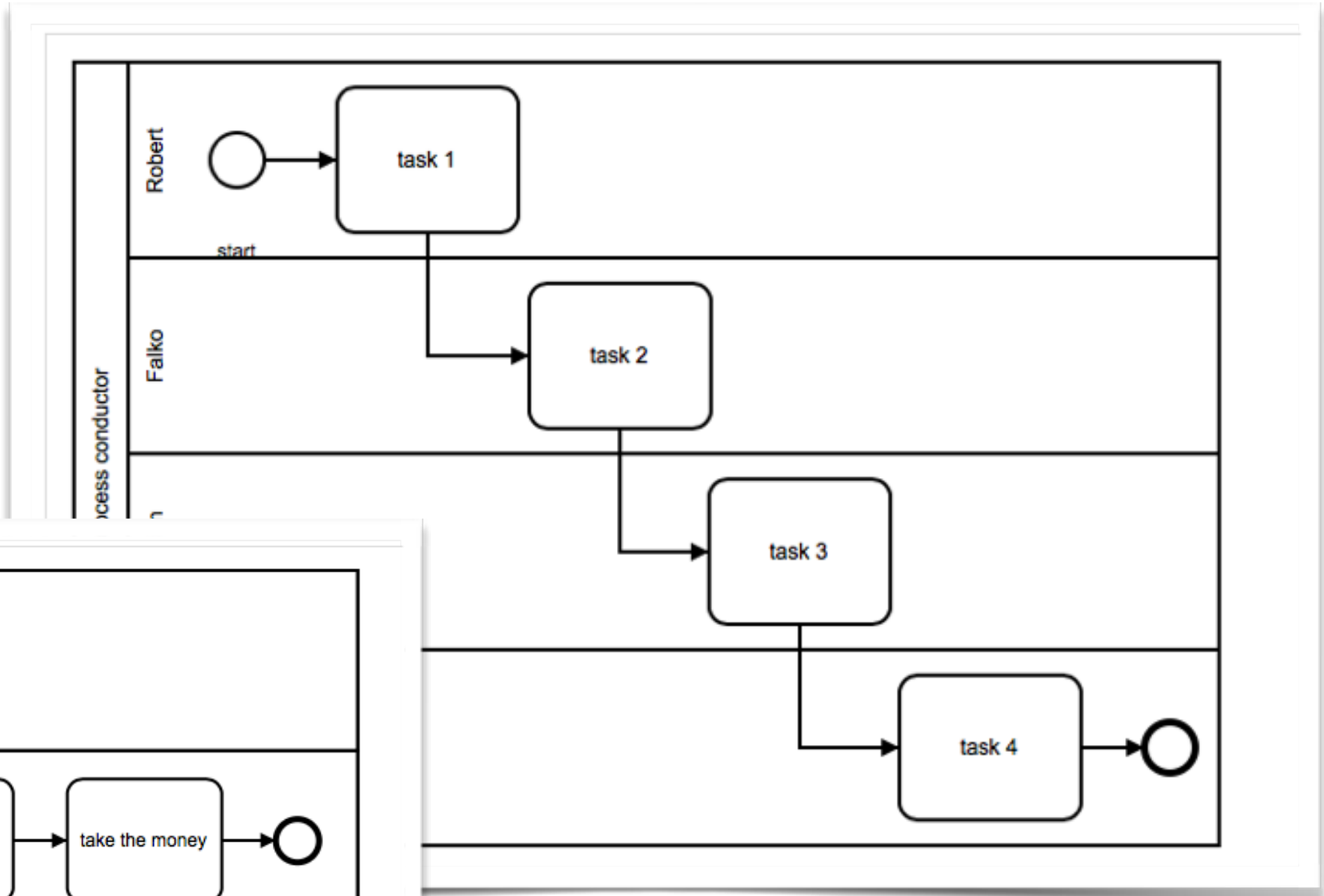
3. **Agentive**: (e.g., the *customer* paying for the flight)
    - Acting behaviour
    - Intentions, Beliefs Desires

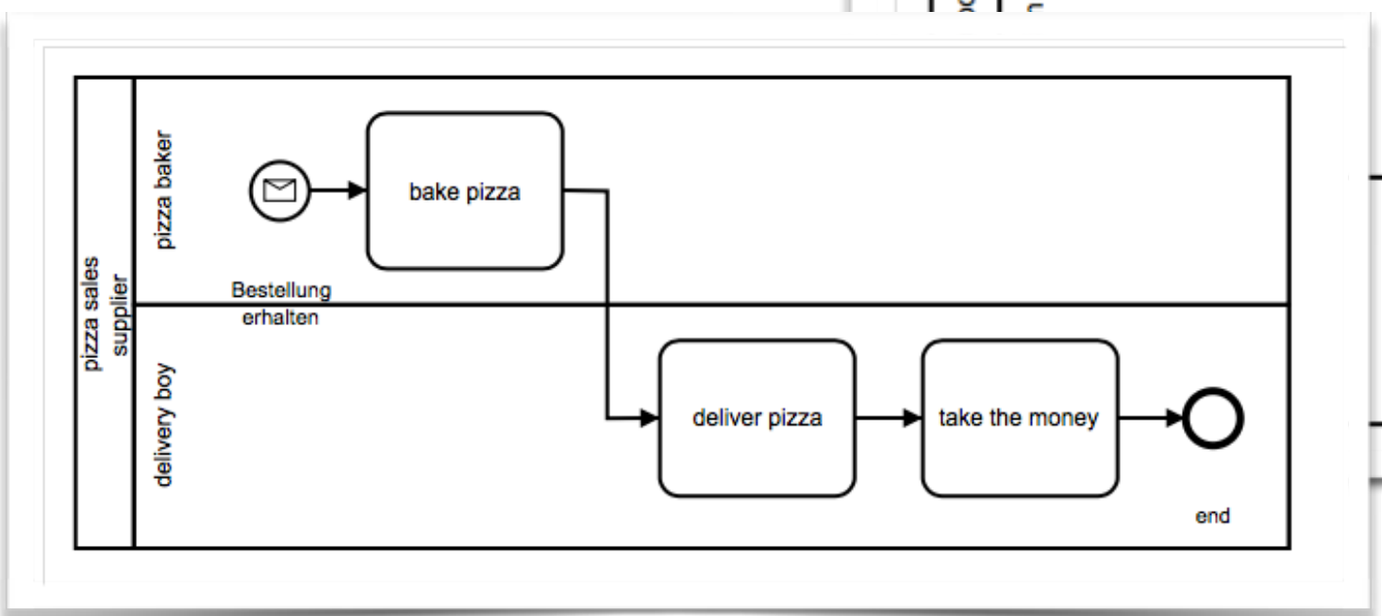
4. **Non-agentive**: Patient of the action (e.g. the *offer* whose status changed from created to rejected)

5. **Roles** of participants: properties that objects only **contingently** satisfy within certain contexts, (e.g. to be *customer* of Amazon, to be a *resource* during the booking of a flight)

Participants - roles

1. **Roles** of participants: properties that objects only **contingently** satisfy within certain contexts





What do our languages actually represent?

	CHARACTERISTIC	BPMN	UML-AD	EPC	CMMN	DECLARE
PROCESS	Set of activities	Yes	Yes	Yes	Yes	Yes
	Clear Input/Output	Yes	Yes	Yes	Somehow	Somehow
	Goal/Value	No	No	Somehow	Somehow	No
	Organizational boundaries	Yes	Yes	Yes	No	No
ACTIV.	Different types of relations between activities	No	No	No	No	Somehow
	State of the word	Somehow	Somehow	Yes	Somehow	No
PART.	Agentive vs non agentive	Somehow	No	Somehow	No	No
	Information vs carrier	No	No	No	No	No
	Object vs role	Somehow	Somehow	Somehow	No	No

Conclusions….

- To represent diagrams is definitely “simpler”
- To inject characterisations from the ontological analysis into BPM languages is necessary also at the time of data.