Multiple Instances in BPMN Collaborations

Barbara Re

University of Camerino
School of Science and Technology

Motivations

We focus on **modelling** of **collaborative systems**

Motivations

We focus on **modelling** of **collaborative systems**

- In **collaborative systems**, multiple participants cooperate and share information via message exchange to reach a shared goal
- Interaction scenarios may become quite complex, especially when **multiple instances** of the interacting participants are involved

Motivations

We focus on **modelling** of **collaborative systems**

- In **collaborative systems**, multiple participants cooperate and share information via message exchange to reach a shared goal
- Interaction scenarios may become quite complex, especially when **multiple instances** of the interacting participants are involved
- A clear **model** of interactions and data exchanges is necessary
  - **BPMN collaboration diagrams** are an effective way to model such **collaborative scenarios**

Motivations

We focus on **modelling** of **collaborative systems**

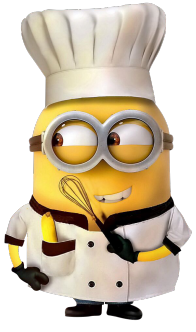
- In **collaborative systems**, multiple participants cooperate and share information via message exchange to reach a shared goal
- Interaction scenarios may become quite complex, especially when **multiple instances** of the interacting participants are involved
- A clear **model** of interactions and data exchanges is necessary
  - **BPMN collaboration diagrams** are an effective way to model such **collaborative scenarios**

Why Multi-instance Collaborations?

Cake Delivery Collaboration: Single-Instance

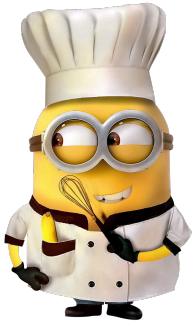
Daughters

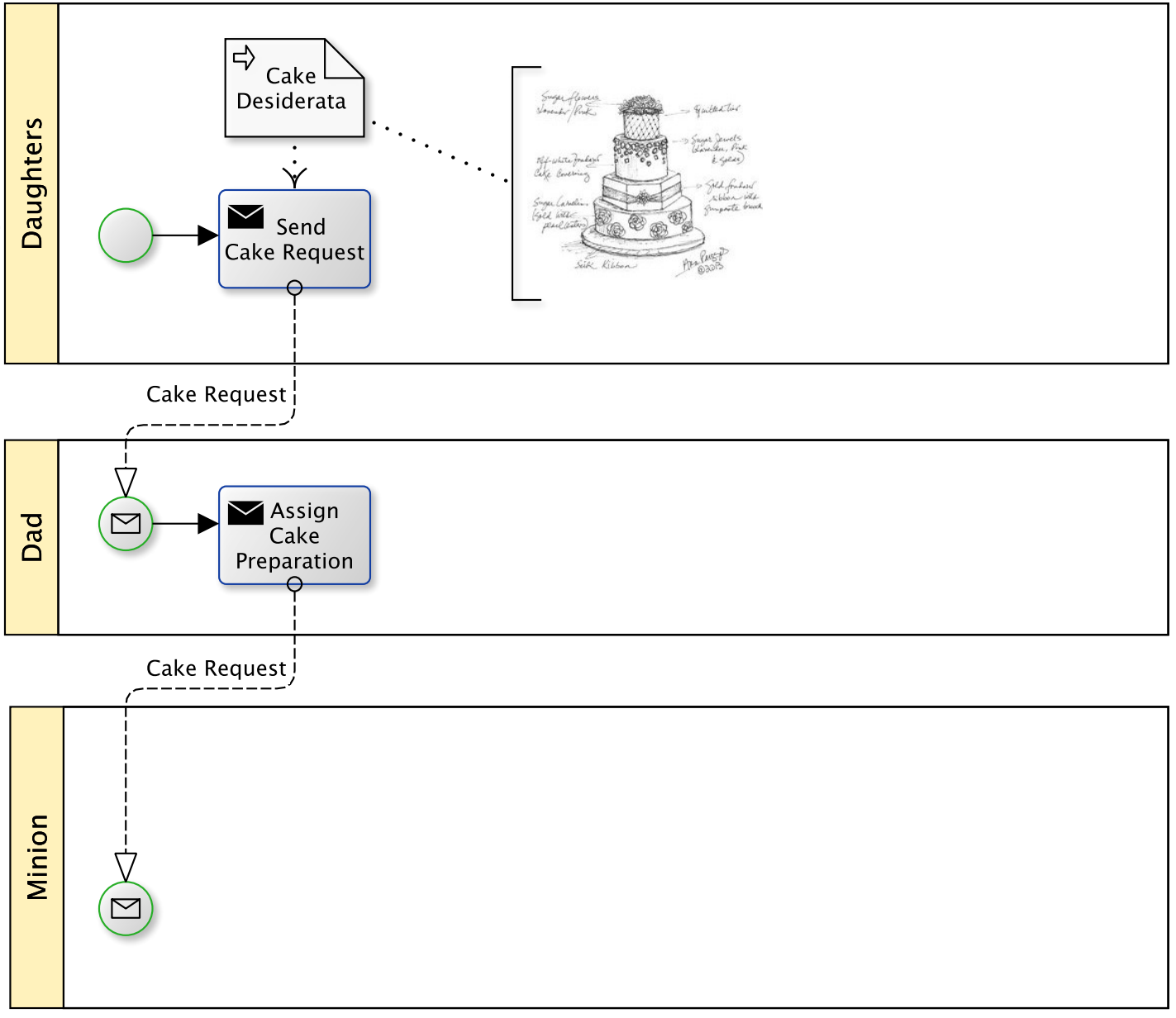
Dad



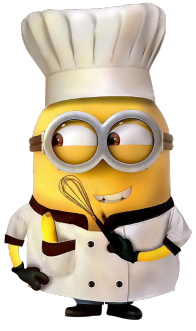
Minion

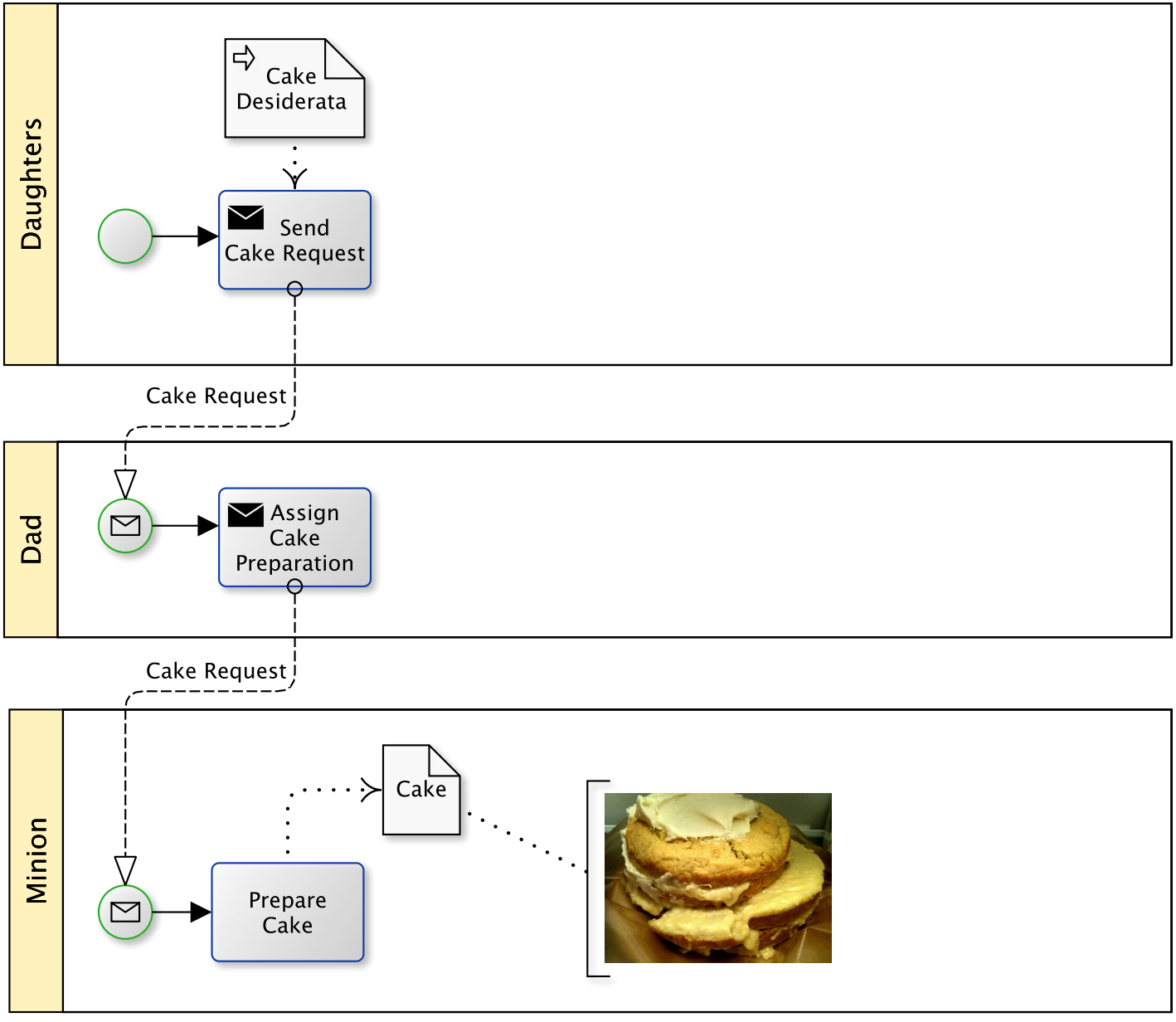
Cake Delivery Collaboration: Single-Instance



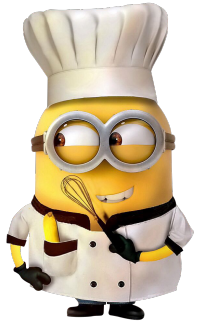


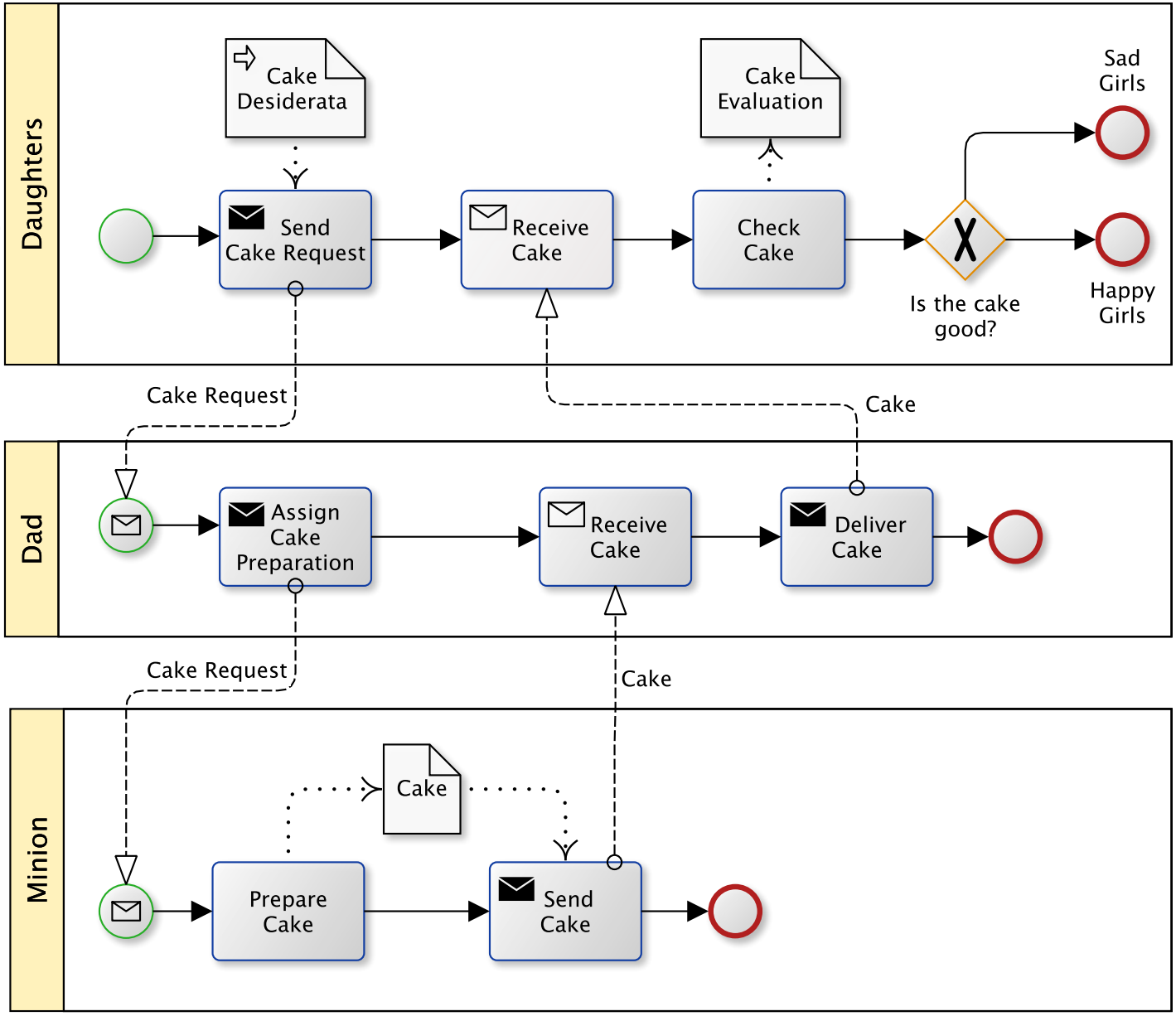
Cake Delivery Collaboration: Single-Instance



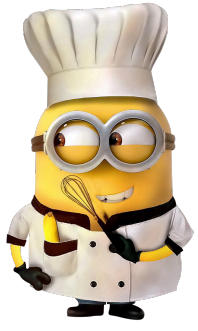


Cake Delivery Collaboration: Single-Instance





Cake Delivery Collaboration: Single-Instance



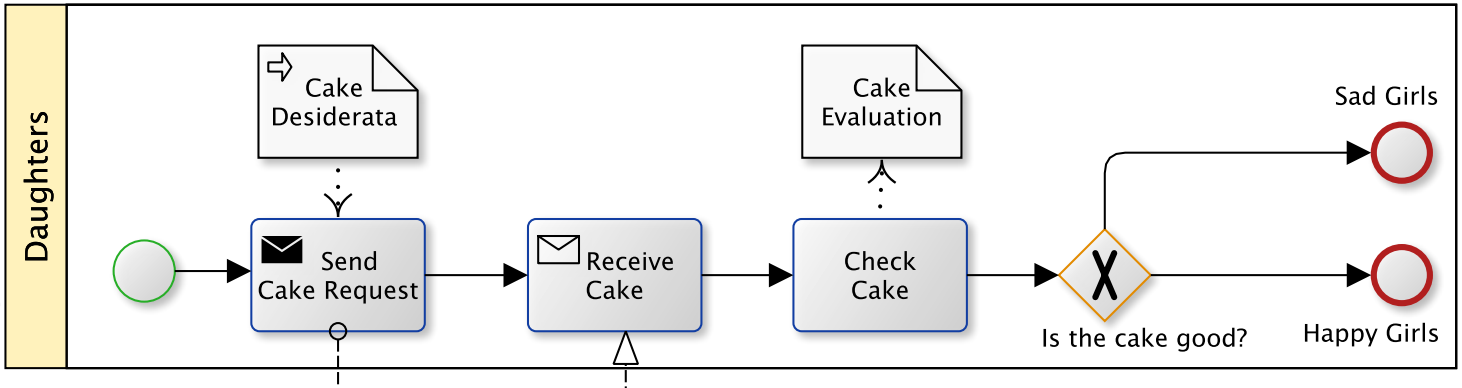
Cake Delivery Collaboration: Multi-Instance

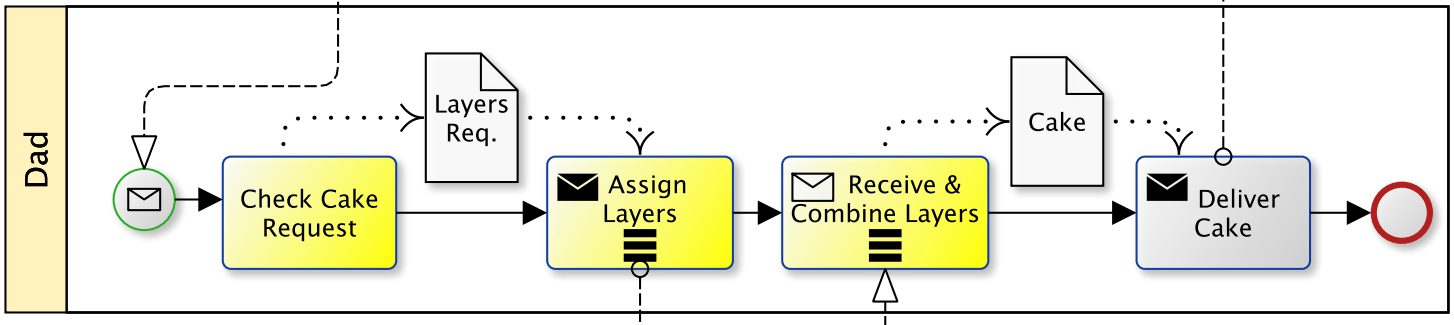
Daughters

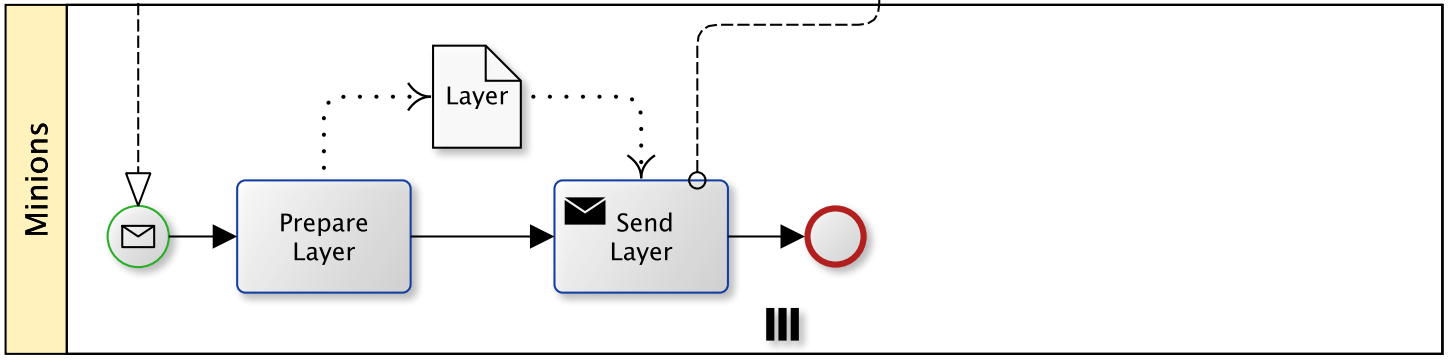
Dad

Minions

Cake Delivery Collaboration: Multi-Instance







Cake Delivery Collaboration: Multi-Instance

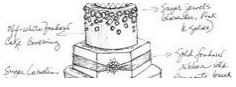
Cake Delivery Collaboration: Multi-Instance

Cake Delivery Collaboration: Multi-Instance

Cake Delivery Collaboration: Multi-Instance

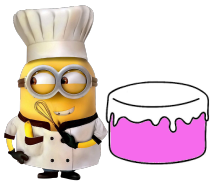
Cake Delivery Collaboration: Multi-Instance

Cake Delivery Collaboration: Multi-Instance



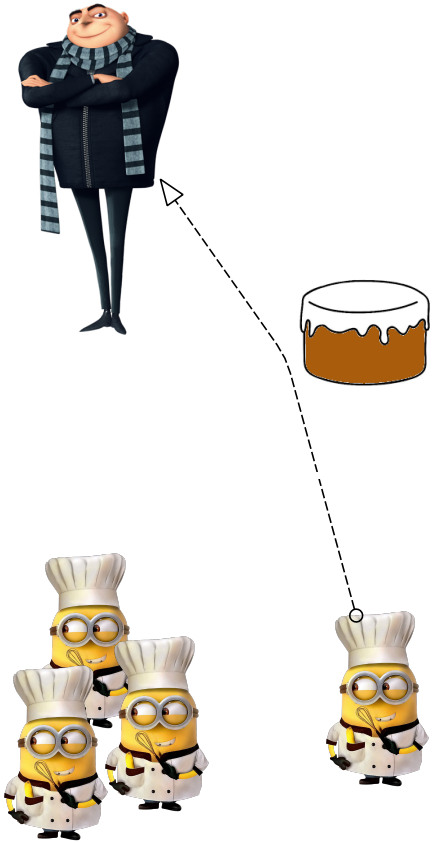
Cake Delivery Collaboration: Multi-Instance

Cake Delivery Collaboration: Multi-Instance



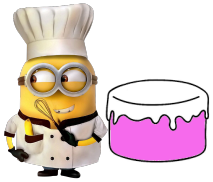
Cake Delivery Collaboration: Multi-Instance

Cake Delivery Collaboration: Multi-Instance



Cake Delivery Collaboration: Multi-Instance

Cake Delivery Collaboration: Multi-Instance

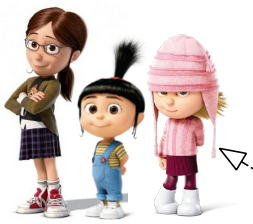


Cake Delivery Collaboration: Multi-Instance

Cake Delivery Collaboration: Multi-Instance

Cake Delivery Collaboration: Multi-Instance

Cake Delivery Collaboration: Multi-Instance



Cake Delivery Collaboration: Multi-Instance

Motivations: multi-instance collaborations

Many practical scenarios involve multi-instance participants
to improve the business process performance
or due to the nature of the problem

- scientific paper review process
- travel booking services
- manufacturer-suppliers scenarios
- ...

Motivations: multi-instance collaborations

Many practical scenarios involve multi-instance participants
to improve the business process performance
or due to the nature of the problem

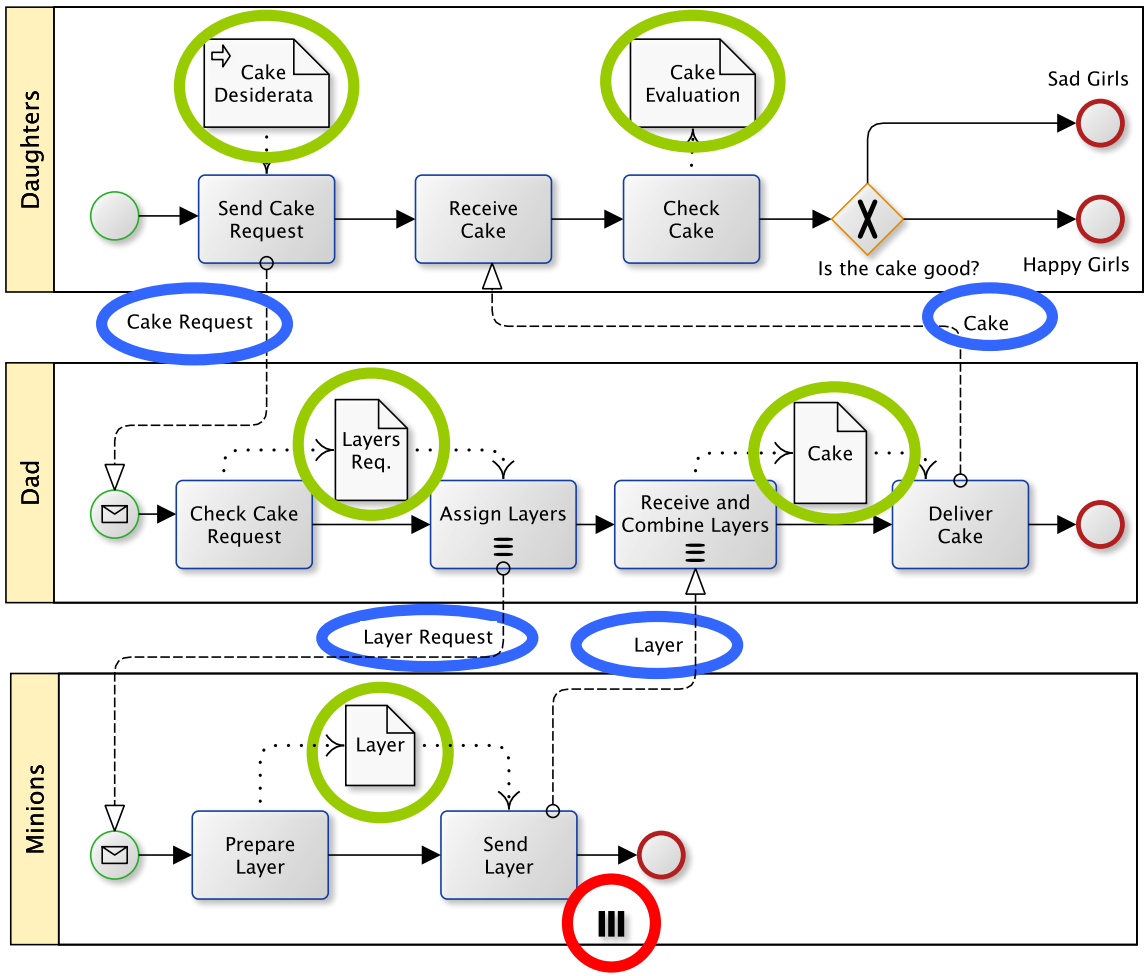
- scientific paper review process
- travel booking services
- manufacturer-suppliers scenarios
- ...

Form the technological point of view, IT supports for collaborative systems make large use of multiple instantiation

- e.g. SOA architectures

Motivations: multi-instance collaborations

The **interplay** between multiple instances, messages, data objects and control flow constructs **complicates the model semantics**



Motivations: multi-instance collaborations

The **interplay** between multiple instances, messages, data objects and control flow constructs **complicates the model semantics**

Modelling multi-instance collaborations is **error-prone**

There can be issues concerning

- bad data handling
- malformed, missing or unexpected messages
- message correlation

Motivations: multi-instance collaborations

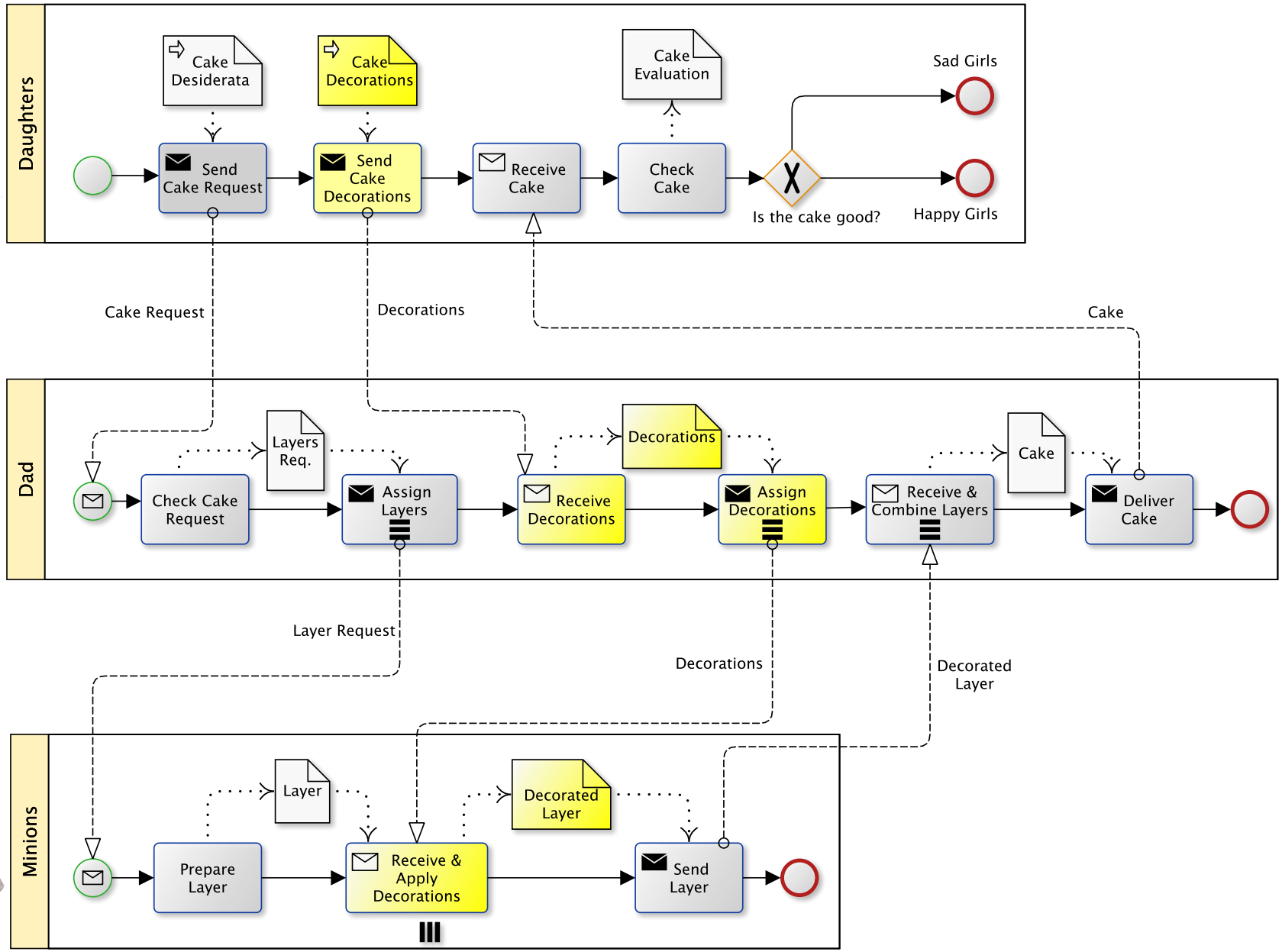
The **interplay** between multiple instances, messages, data objects and control flow constructs **complicates the model semantics**

Modelling multi-instance collaborations is **error-prone**

There can be issues concerning

- bad data handling
- malformed, missing or unexpected messages
- **message correlation**

Cake Delivery Collaboration: Correlation Issues

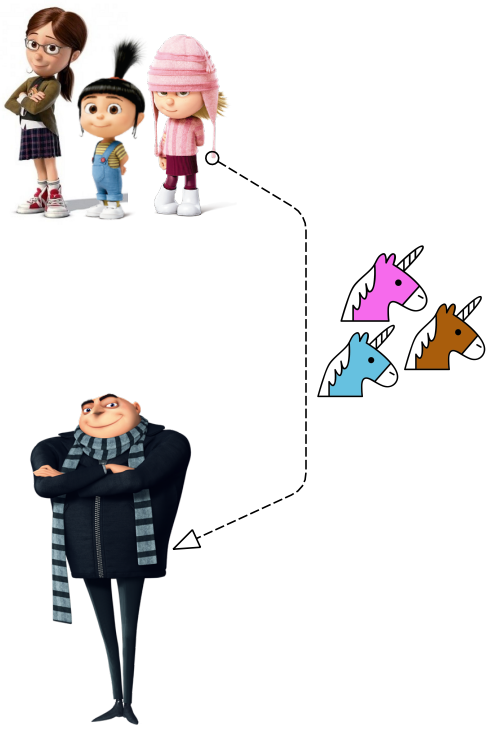


Cake Delivery Collaboration: Correlation Issues

The initial interactions are
like in the previous scenario...

Cake Delivery Collaboration: Correlation Issues

Cake Delivery Collaboration: Correlation Issues

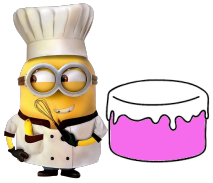


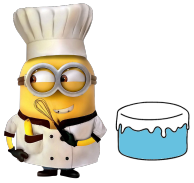
Cake Delivery Collaboration: Correlation Issues

Each decoration must be
delivered to the proper minion

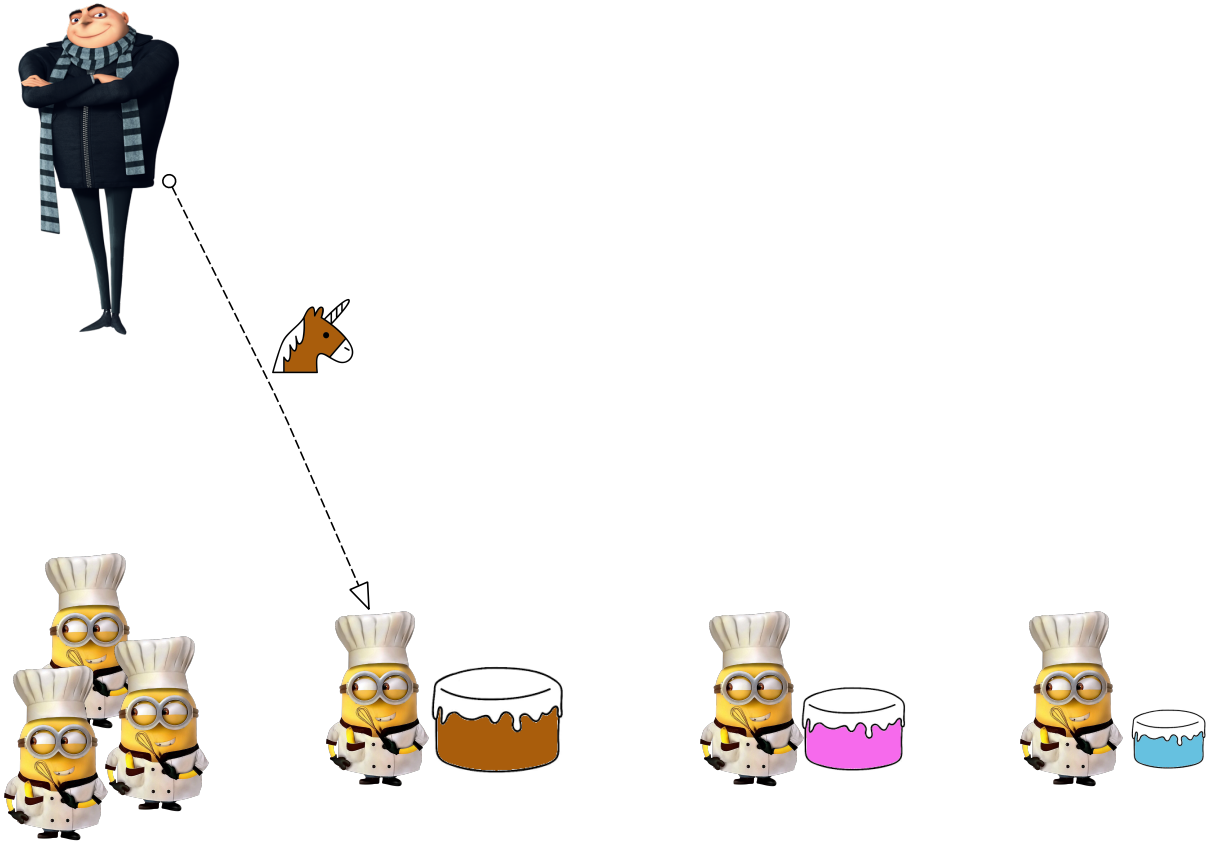
Each Dad's message must be
correlated
to the proper Minons' instance

BPMN fosters the use of
the content of the messages
to correlate them with
the corresponding instances

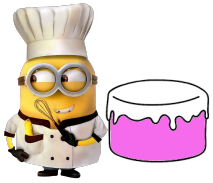


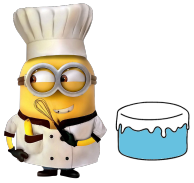


Cake Delivery Collaboration: Correlation Issues

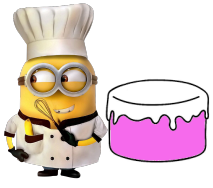


Cake Delivery Collaboration: Correlation Issues





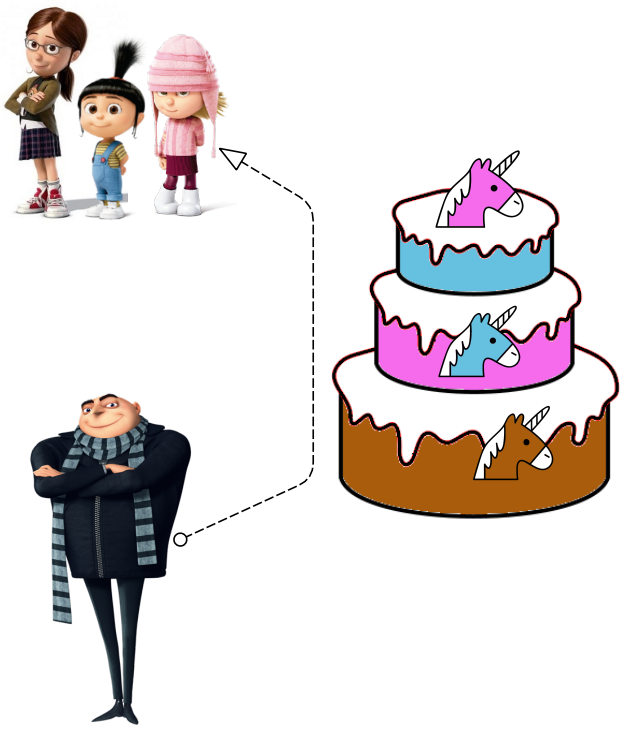
Cake Delivery Collaboration: Correlation Issues



Cake Delivery Collaboration: Correlation Issues

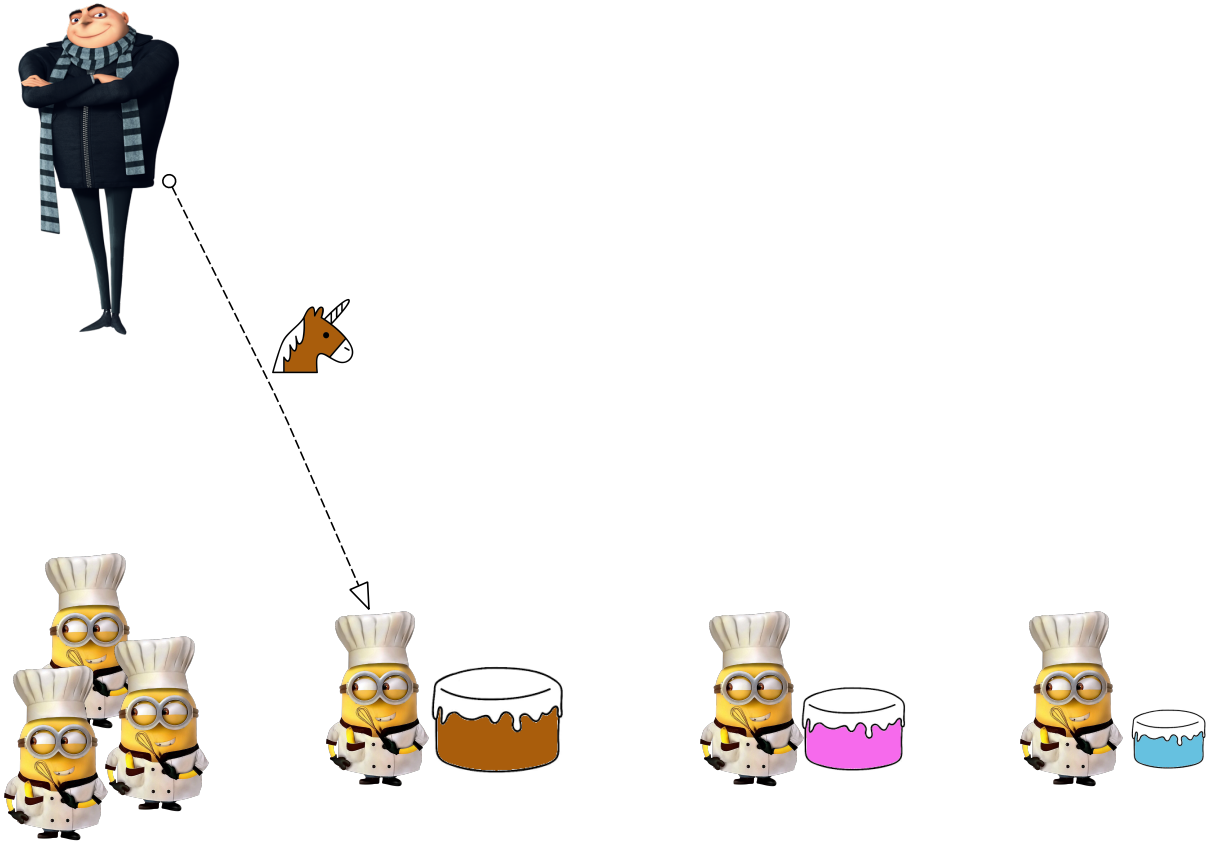
The result will be...

Cake Delivery Collaboration: Correlation Issues

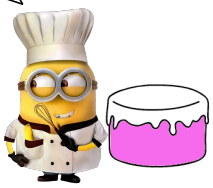


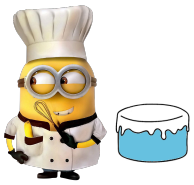
Cake Delivery Collaboration: Correlation Issues

Cake Delivery Collaboration: Correlation Issues



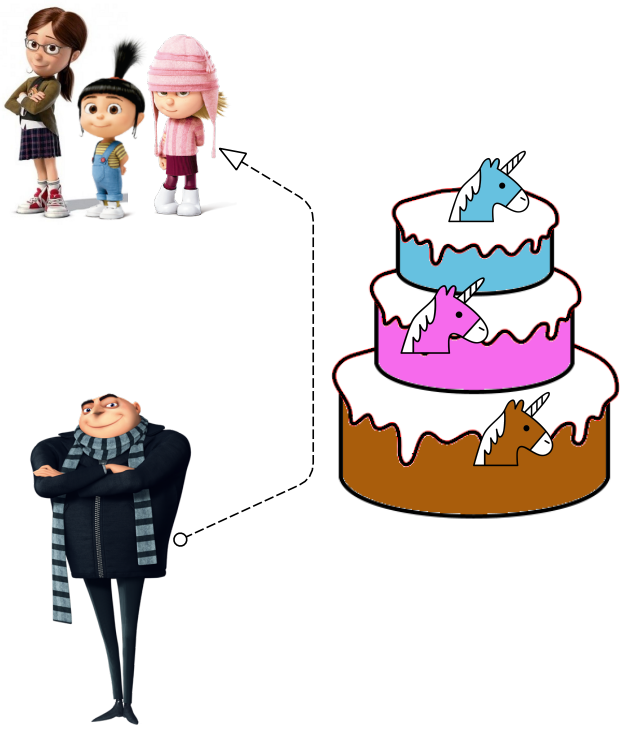
Cake Delivery Collaboration: Correlation Issues





Cake Delivery Collaboration: Correlation Issues

Cake Delivery Collaboration: Correlation Issues



Cake Delivery Collaboration: Correlation Issues

Interplay between Multiple Instances, Messages and Data Objects

To deal with **multiple instances** in BPMN collaboration models, it is necessary to take into account the **data flow**

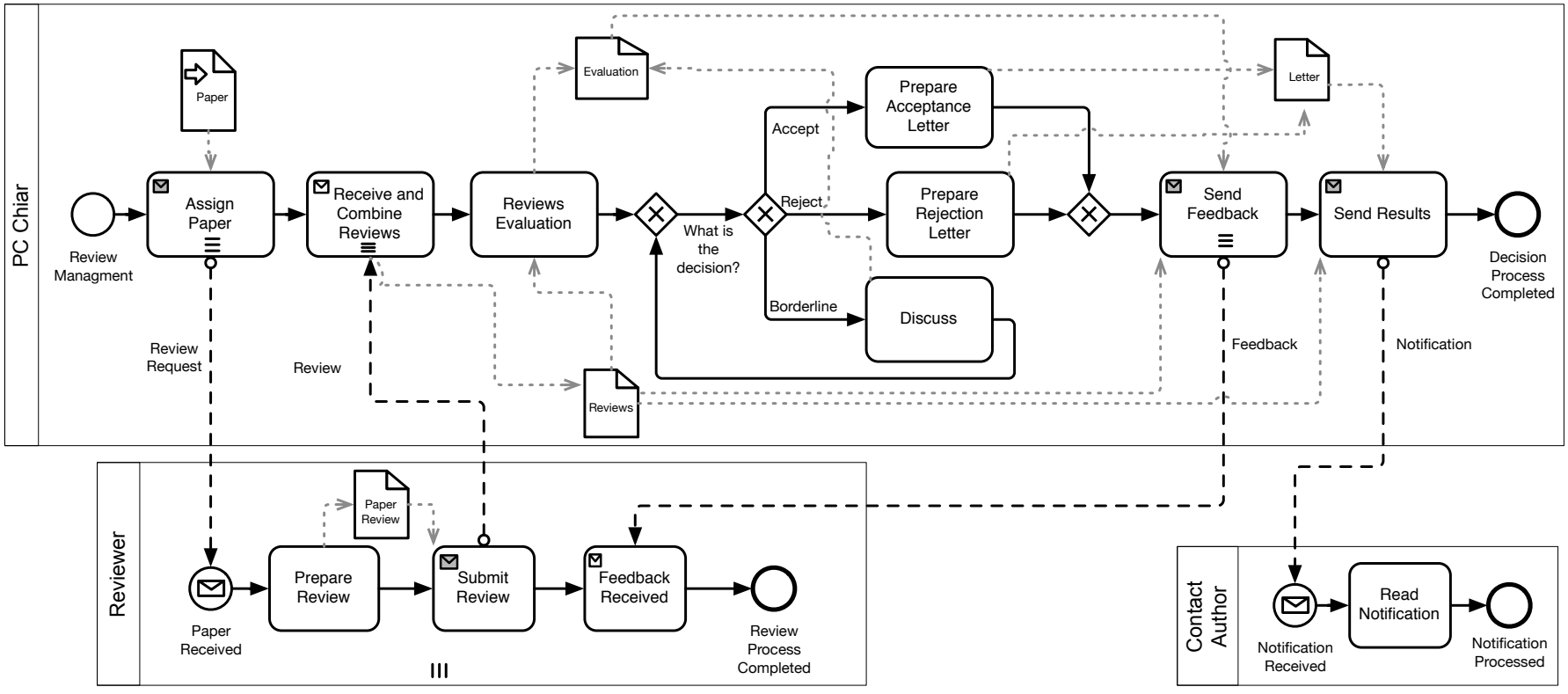
- **The creation of process instances** can be triggered by the arrival of messages, which contain data
- **Within a process instance**, data is stored in data objects, used to drive the instance execution

Values of data objects can be used to fill the content of **outgoing messages**, and vice versa, the **content of incoming messages** can be stored in **data objects**

Messages and process instances must contain enough information to determine, when a message arrives at a pool, **if a new process instance is needed or, if not, which existing instance will handle it**

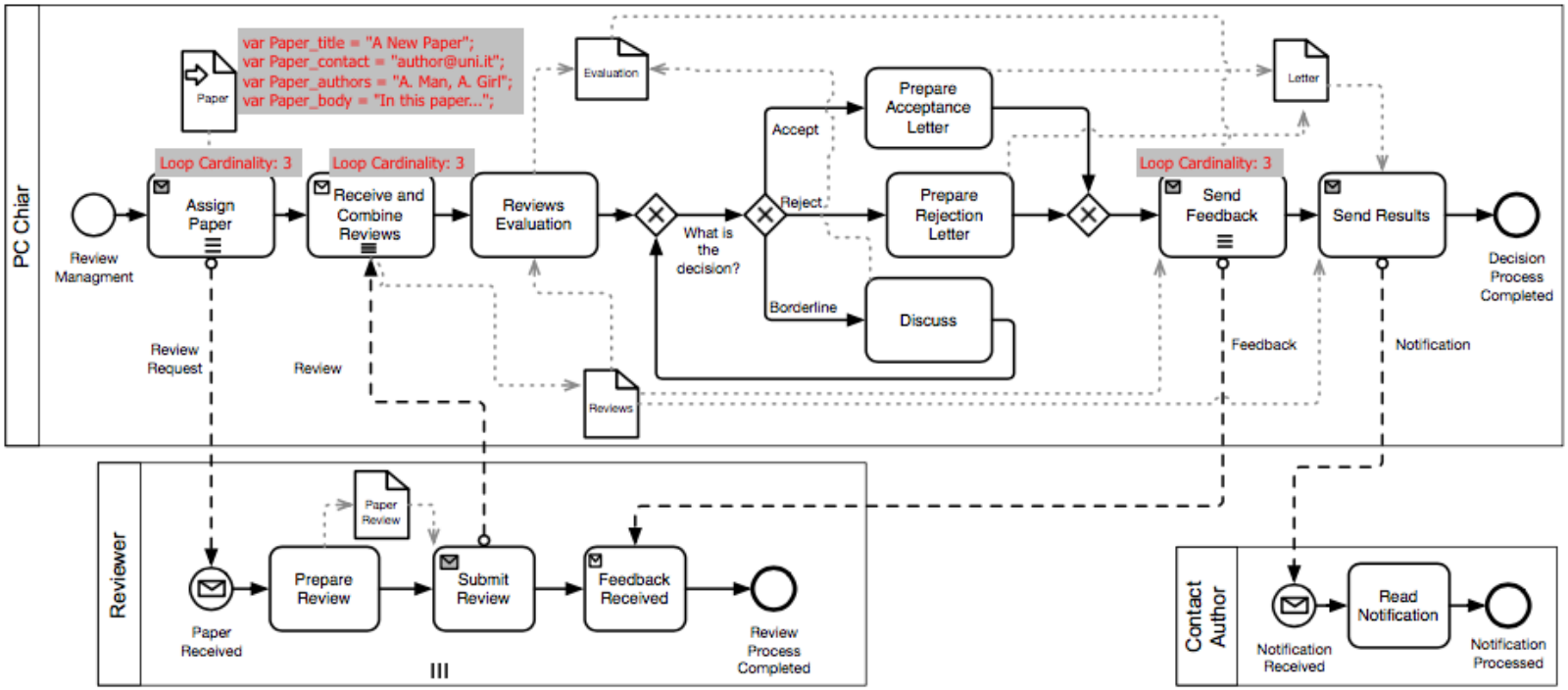
- To this aim, BPMN makes use of the concept of **correlation**, this is achieved by embedding values, called **correlation data**, in the content of the message itself
- **Pattern-matching** is used to associate a message to a distinct receiving task or event

Paper Reviewing Collaboration Model (I)



Messages coming into Reviewer pool might start a new process instance, or be routed to existing instances already underway

Paper Reviewing Collaboration Model (II)



(a) Paper {title, contact, authors, body} Reviews {title, reviewers, scores, bodies}
Evaluation {title, decision} Letter {title, evaluation} PaperReview {title, score, body}

(b) ReviewRequest {title, body} Notification {title, contact, authors, evaluation, scores, bodies}
Review {reviewerName, title, score, body} Feedback {reviewerName, title, evaluation}

In order to keep the notation independent from the kind of data structure required from time to time the standard left underspecified data objects structure

We consider here a generic record structure, assuming that a data object is just a list of fields, characterised by a name and the corresponding value. Of course, a field can be used to represent the state of a data object.

Paper Reviewing Collaboration Model (III)

According to the BPMN standard, **data objects do not have any direct effect on the sequence flow or message flow of processes**, since tokens do not flow along data associations (see p. 221 BPMN Standard)

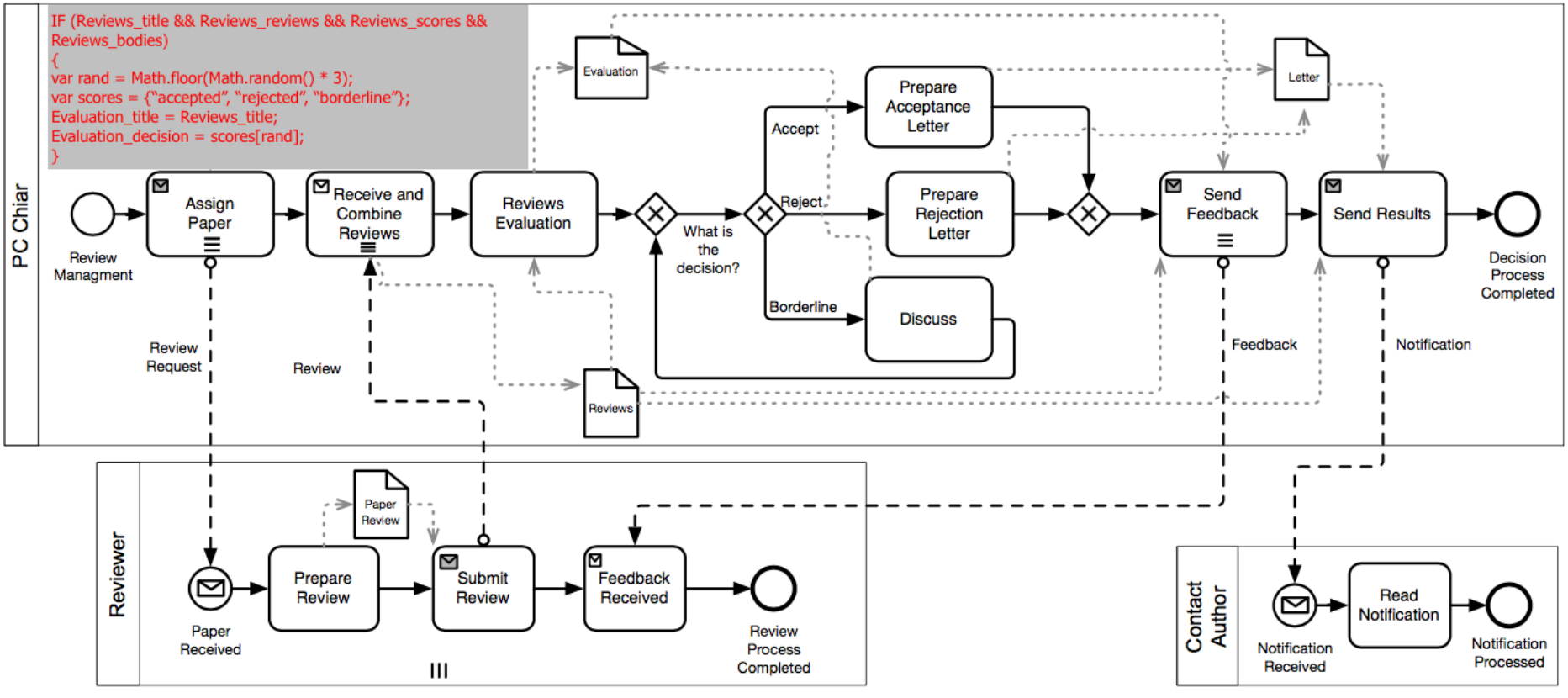
However, this statement is questionable:

- On the one hand, **the information stored in data objects can be used to drive the execution of process instances**, as they can be referred in the **conditional expressions of XOR gateways to take decisions** about which path should be taken
- On the other hand, **data objects can be connected in input to tasks** and the standard states that *"the **Data Objects as inputs into the Tasks act as an additional constraint for the performance of those Tasks**. The performers [...] cannot start the Task without the appropriate input"* (see p. 183 BPMN Standard)

A data object has an implicit indirect effect on the execution, since it can **constrain the decision taken by a XOR gateway or act as a guard condition on a task** that can **constrain and manipulate** data object fields values **via assignments**

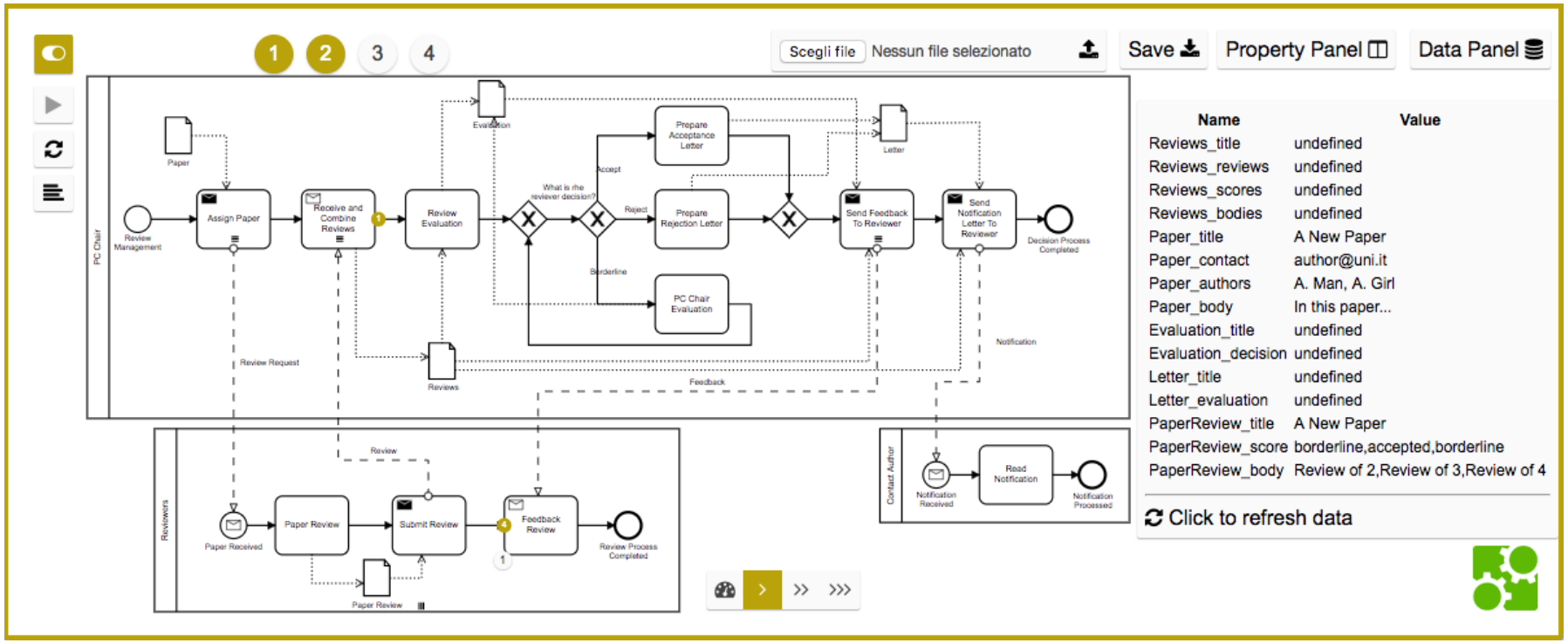
Paper Reviewing Collaboration Model (IV)

$$if\,(guard)\{assignements\}$$



MIDA Animator

MIDA: Multiple Instances and Data Animator



http://pros.unicam.it:8080/Mida/modeler.html

The MIDA tool can **support designers in debugging their multi-instance collaboration models**, as it permits to check the evolution of data and process marking while executing the models. **The identification of the bug is still in charge of the human user**.

Lessons Learned

The BPMN standard has the flavour of a framework rather than of a concrete language, because some aspects are not covered by it, but left to the designer

> The standard left **underspecified the internal structure of data objects**: "*Data Object elements can optionally reference a DataState element [...] The definition of these states are out of scope of this specification*" (see p. 206 BPMN Standard)

We consider **a generic record structure for data objects**

> The **expression language** operating on data is left unspecified by the standard

This is not an issue for the formalisation, but the expression language has to be instantiated in the concrete implementation of the animator

In **MIDA**, for the sake of simplicity, **we resort to the expression language of JavaScript**, as this is the programming language used for implementing the tool

> The **lack of a formal semantics** in the standard may lead to different interpretations of the tricky features of BPMN

We aim at clarifying the **interplay between multiple instances, messages and data objects**