Machine-Assisted Design of Business Processes Using Descriptor Space Analysis

Maya Lincoln, Technion – Israel Institue of Technology
Avigdor Gal, Technion – Israel Institue of Technology
Mati Golani, Ort Braude College

Background/1

- Business Process Repositories describe the “know-how” of organizations
- Business Process Repositories can be used for:
  - Management of regulations and compliance enforcement
  - Management and control of IT systems
  - Analysis and improvement of processes
  - Documentation and training
  - Mergers and acquisitions planning
  - Performance monitoring

Motivation /1

- Process modeling is considered a manual, labor intensive task
  - The outcome depends on personal domain expertise
  - Errors or inconsistencies can lead to bad process performance and high process costs
- Hence, automating the reuse of constructs, gathered from predefined process models does not only save design time but also supports non-expert designers in creating new business process models

Motivation /2 – An Example

- Consider an airport process model that incorporates processes related to passengers check-in before boarding an airplane
- Now, suppose that the airport management desires to extend the services provided to its customers by offering a new service: "check-in from home"
- In addition, it is also desired to outline the "check-out" process model as an extension of the current repository
- The existing repository encapsulates know-how and business logic that are relevant and useful for the creation of these new models
  - e.g. passenger check-in policies and procedures regarding security, luggage handling, passenger handling and document validation

Motivation /3 – An Example

- In the above scenario, it would have been helpful for the process designer to design the new processes using a supporting system that relies on the reuse of previous know-how instead of creating the model manually from scratch
- To illustrate our methodology we use a real-world case study for airport process design
- Based on a "check-in" process that already exists in the repository, we demonstrate how it is possible to design the two, above mentioned, new business processes

Research objective

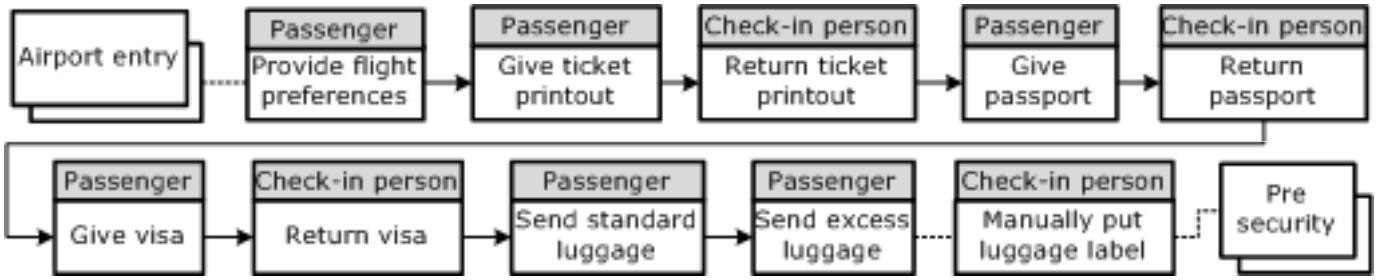
- Propose an effective method for designing new business process models related to any functional domain, without limiting the focus to a specified functional area
- Delineate new business process models according to the organization's specific business logics and business rules

Related work

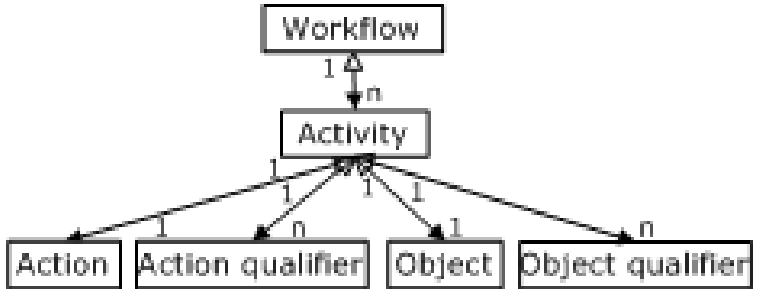
- Most previous work focused on supporting the design of alternative process steps within existing process models
- Less work has been carried out on the design of new process models
- The few works that addressed the design of new models were limited to a specific domain such as production management

The Descriptor Model /1

An example: the passenger check-in process



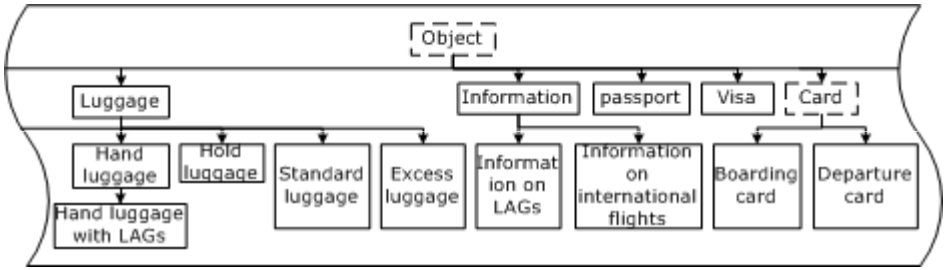
The process descriptor model



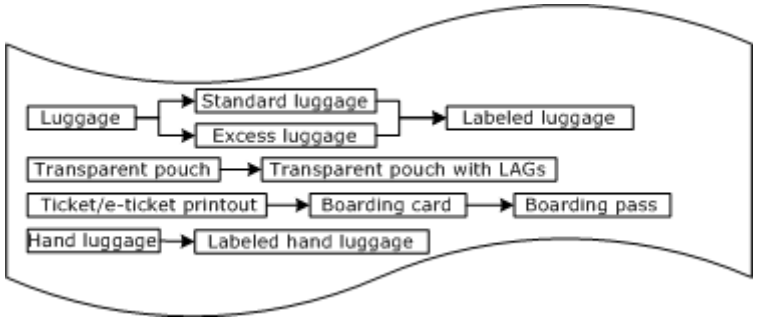
A Descriptor Model for Process Design /1

Object taxonomies

- An object hierarchy model



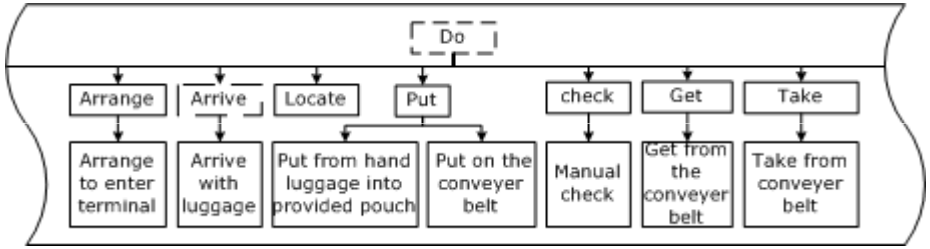
- An object lifecycle model



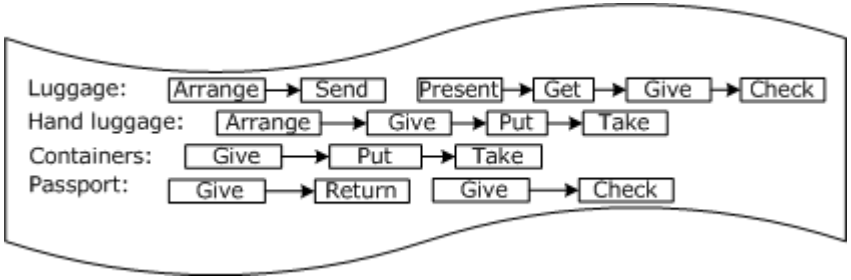
A Descriptor Model for Process Design /2

Action taxonomies

- An action hierarchy model



- An action lifecycle model



The Descriptor Space - Definition

- A quad-dimensional space of activities
  - Each space coordinate represents an activity as a quadruple AC = <O,OQ,A,AQ>
  - Some coordinates represent “real” activities from the process repository, while others represent “virtual” activities
- The distance between every two coordinates

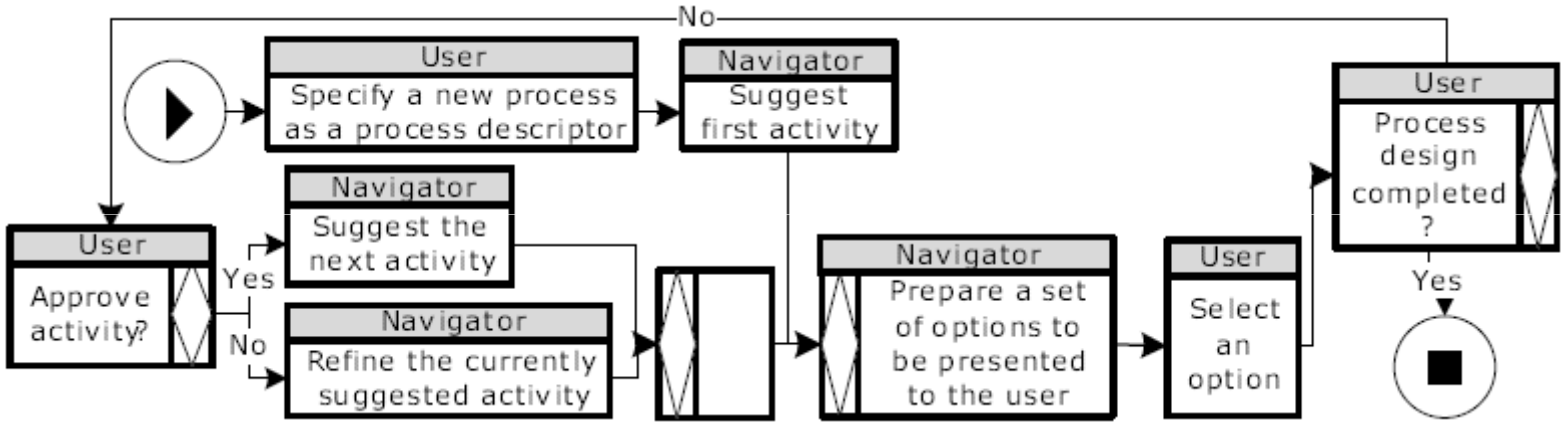
$$Dist\,(AC_i, AC_j) = OD_{ij} + AD_{ij} + OHD_{ij} + AHD_{ij}$$

  - OD_{ij} – the object distance: the minimal number of steps connecting Oi and Oj in the object lifecycle model
  - AD_{ij} - the action distance: the minimal number of steps connecting Ai and Aj in the action sequence model
  - OHD_{ij} - the object hierarchy distance: the minimal number of steps connecting Oi with Oj in the object hierarchy model
  - AHD_{ij} – the action hierarchy distance, defined similarly to OHD_{ij}
  - A “no-connection” distance is used when OD/AD are undefined

The Descriptor Space – An Example for Calculating Distances

- Consider the two descriptors:
  - AC_i = (luggage,hand,check,null) and
  - AC_j = (luggage,null,get,from the conveyer belt)
- To navigate from AC_i to AC_j:
  - We move one step up in the object hierarchy (OHD = 1) from the object Hand luggage to the object Luggage
  - Then, we recede two steps back from the action Check in the action sequence (AD = 2), resulting with the action “Get”
  - Finally, we drill down one step within the action hierarchy (AHD = 1), and retrieve the action “Get” from the conveyer belt, and by that we reach the target descriptor
  - The total distance between the two above coordinates is 4

The Process Navigator /1



The Process Navigator /2

- Suggesting the First Process Activity
  - Goal
  - Search the target object and its more specific objects within the object hierarchy model
  - Match it with an initial action that can be acted on this object
  - Compose first activity suggestions
    - Retrieved objects and the first action that acts upon them
  - Sort and flag results

The Process Navigator /3

- Refining the Currently Suggested Process Activity (e.g. “Get luggage”)
  - Action and Object Refinement
    - E.g. “Get luggage from the conveyer belt”, “Get hand luggage”
  - Action and Object Generalization
  - Advance an Object's State or an Action
    - The object “Standard luggage” represents a more advanced state of the object “Luggage”
    - The action “Give” follows “Get” in the action sequence applied on “Luggage”
    - => The following refinement suggestion is constructed: “Get standard luggage”,and “Give luggage”

The Process Navigator /4

- Refining the Currently Suggested Process Activity (continue)
  - Recede to a Less Processed State of the Object or to a Former Action
    - E.g. the action “Present” is acted on “Luggage” before this object is taken (before the action “Get” is applied), hence creating the refinement option: “Present luggage”
  - Move to a Sibling Action or Object
    - E.g. a navigation to sibling actions to “Get” retrieves a list of activities that includes: “Check luggage” and “Take luggage”
    - In the same manner, a search for sibling objects, retrieves a list of activities, that includes: “Get passport” and “Get visa”

The Process Navigator /5

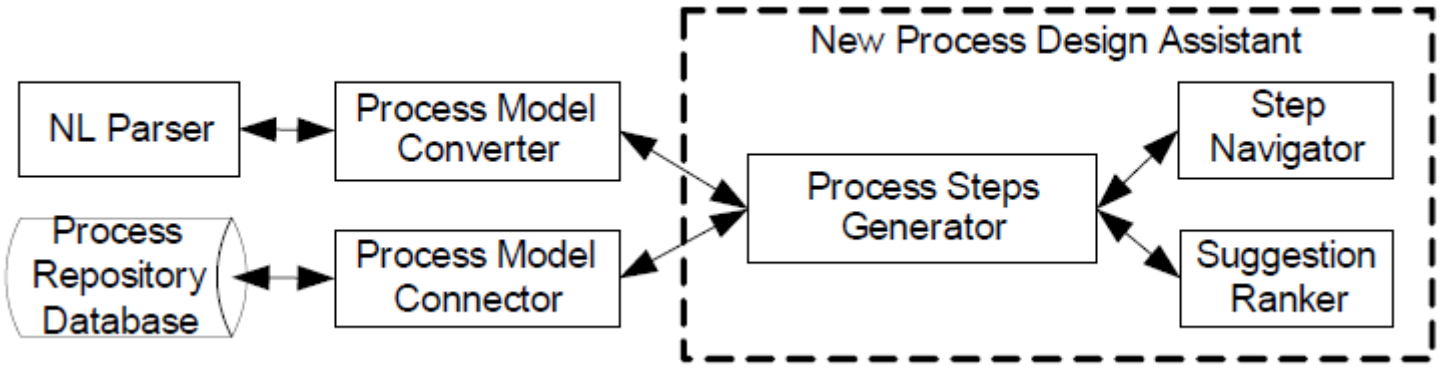
- Suggesting the Next Process Activity
  - Goal: take the process execution flow one step forward
  - Two alternative ways:
    - Advancing to a later action that acts on the currently accepted object
      - E.g. “Give passport”-> “Check passport” / ”Return passport”
    - Proceeding to a sibling object combined with the reference activity's action
      - Rationale: in some process flows the same action is operated on sibling objects in order to fulfill a certain process goal (e.g. Send standard luggage -> Send excess luggage)
      - E.g. “Give passport”-> “Give visa” / ”Give luggage” / ”Give information”

The Process Navigator /6

- Preparing a Set of Output Options
  - Sort by Proximity to the Reference Activity
    - By calculating distances
  - Internally Sort by Similarity to Processes in the Repository
    - *No change* - the suggested activity is represented “as is” within the underlying business process repository. No mark
    - *Slight modification* - there is an actual activity in the underlying business process repository containing the same object and action with different qualifiers. Marked with “~”
    - *Major change* - the object and action within the suggested activity were not coupled in any of the activities within the underlying business process repository. Marked with “M”.
  - Add a Random Option
  - Flag Each Option
    - E.g. “[1,~]”

Implementation

- Server side logic is implemented in PHP using a MySql database
- The client runs within an Internet browser and is implemented in HTML and JavaScript, with AJAX calls to the server

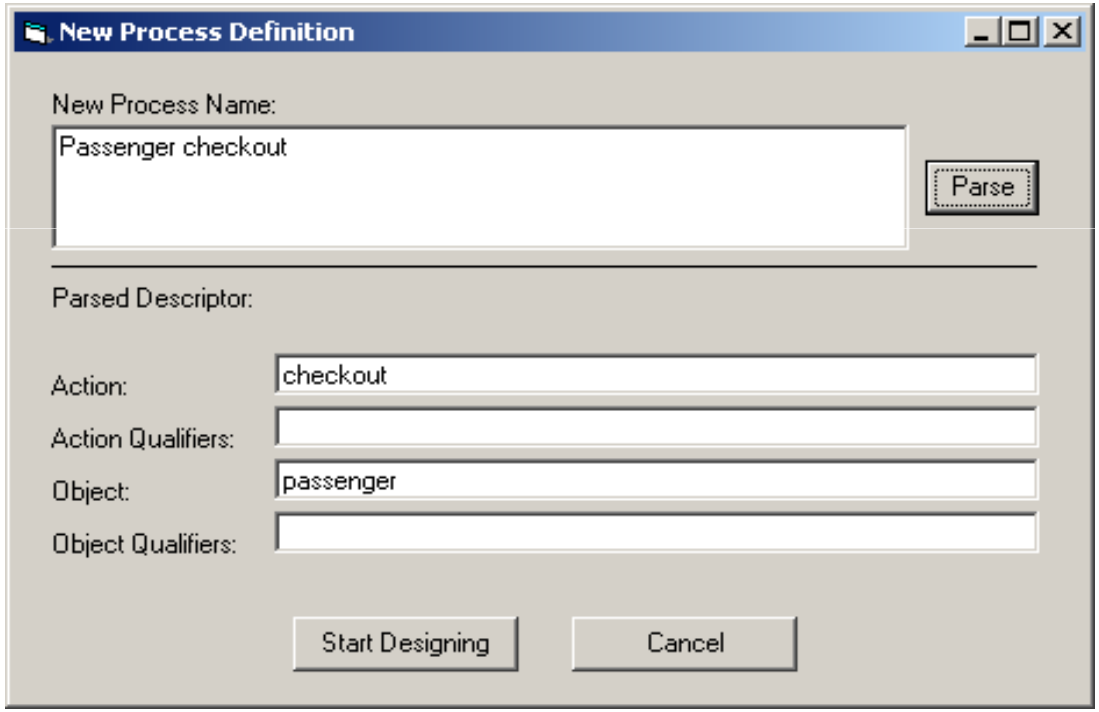


Case Study /1

- Based on the aviation process repository
- Designing a new process: “Passenger Checkout”
  - Extends the process repository by handling passenger related activities conducted after an airplane arrives at its destination
  - Final design output:

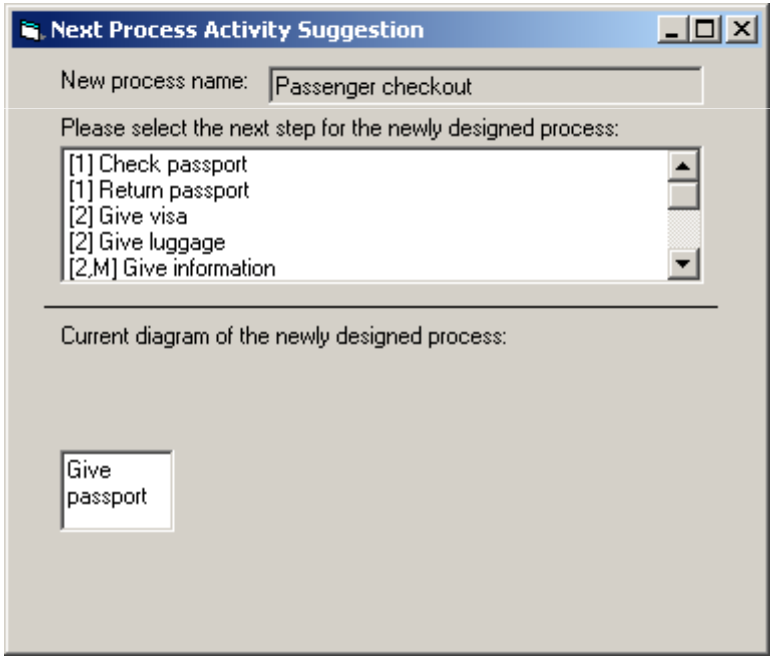
Case Study – process generation system/2

- Step 1: The process designer's input



Case Study process generation system/3

- Step 2: First activity (defined by the designer) is: “Give passport”
- Step 3: Next activity suggestions:



Case Study /4

- Step 4: The designer selects the option “Check passport”
- Step 5: The designer selects the option “Give luggage” as a next future activity (will be required at the customs point)
- Step 6: The designer then asks the process navigator to provide next step options and receives:
  - [1] Check luggage, [2] Give visa, [2,M] Give information
- Step 7: The designer selects the first option, “Check luggage”

Case Study /5

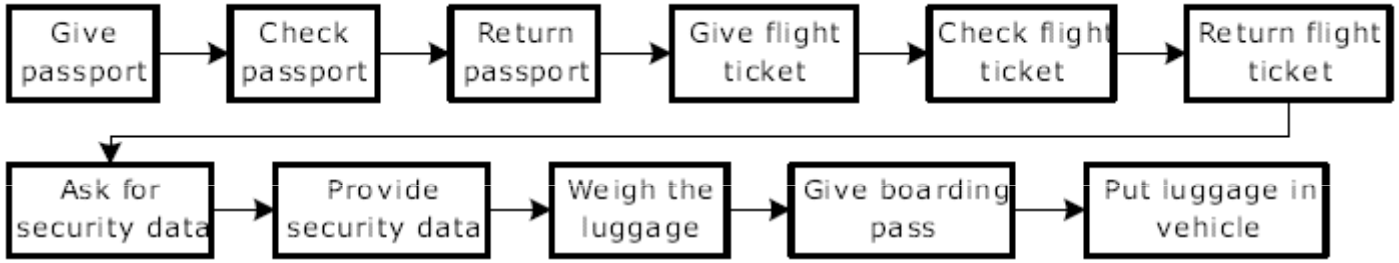
- Step 8: The designer asks for previous activity suggestions to “Give luggage”
  - Rationale: by reviewing the newly designed process, she realizes that an activity may be missing before Give luggage, since the passenger may not have carried his luggage with him to the airplane.
- Step 9: Retrieved previous step suggestions (by navigating backwards in the action sequence)
- Step 10: The designer selects the option: “Get luggage” and asks the process navigator to refine it
  - Reason: it seems to lack sufficient details to express the activity required in this context

Case Study /6

- Step 11: The process navigator presents refinement suggestions
- Step 12: The designer selects the option: “[1,~] Get luggage from the conveyer belt”
  - Note that this activity was selected although it was not represented “as is” in the business process repository

Case Study /7

- Designing the new process: “Send luggage from home”
- Output:



- An interesting observation is the usage of the activity “Put luggage in vehicle”
  - While the original business process repository contained the action “Put in vehicle” applied only to the object “Baby carriage”, the terminating activity combines this action with the object: “Luggage”

Experiments - Data /1

- We chose a set of 14 processes from the Oracle Business Model (OBM)
  - nine business processes from the Procurement category (96 activities)
  - five business processes from the Inventory category (31 activities)
- The Procurement data set contains related, sequential activities and therefore encapsulates a focused operational area
- The Inventory data set encapsulates a loosely coupled business logic regarding an extended business area

Experiments - Evaluation Methodology /2

- At each experiment, a single process was removed from the database and was reconstructed using the “New Process Design Assistant” software (NPDA)
- This way, the missing process serves as the final design goal, enabling us to measure the method's effectiveness in an objective manner
- Each experiment was conducted according to the following steps:
  - Remove one of the processes from the database so that the database will not contain any of its activities
  - Run the NPDA and select at each phase the option (activity) compatible with the removed process
    - Handle cases in which no option represents the goal activity

Experiments - Methodology /2

- At each experiment, a single process was removed from the database and was reconstructed using the “New Process Design Assistant” software (NPDA)
- This way, the missing process serves as the final design goal, enabling us to measure the method's effectiveness in an objective manner
- Each experiment was conducted according to the following steps:
  - Remove one of the processes from the database so that the database will not contain any of its activities
  - Run the NPDA and select at each phase the option (activity) compatible with the removed process
    - Handle cases in which no option represents the goal activity

Experiment Results/1

Table 1. Experiment results.

Column #	1	2	3	4	5	6	7
Column name	# of total processes in DB	# of total activities in DB	% of goal activities represented in the DB	Avg. # of steps per design phase	Avg. location of correct option in 'next activity'	Avg. location of correct option in 'refine activity'	Avg. location of the correct option per design phase
Avg.-all	14	127	89.0%	2.0	1.2	2.8	2.6
Avg.-Procurement	9	96	90.6%	1.9	0.8	3.0	2.8
Avg.-Inventory	5	31	83.9%	2.1	1.9	2.4	2.3

Experiment Results/2

Table 2. Distribution of successful predictions vs. the number of required refinements.

# of refinements	0	1	2	3	4	5	6	7	8	9
% of successful predictions	12%	35%	27%	12%	4%	2%	2%	1%	1%	3%
Cumulative	12%	48%	75%	88%	92%	94%	96%	96%	97%	100%

Conclusions/1

- The proposed method, software tool, and experiments provide a starting point that can alreadybe applied in real-life scenarios, yet several research issues remain open, including:
- (1) an extended empirical study to further examine the quality of newly generated processes;
- (2) an extended activity decomposition model to include
- an elaborated set of business data and logic (e.g., roles and resources); and
- (3) defining a learning mechanism that will take into account previous designer preferences and adjusting (in real time) the process delineator mechanism.

Conclusions/2

- As a future work we intend to investigate further language semantics by using more advanced natural language processing techniques, as well as semantic distances between words.
- Finally, we intend to apply the techniques we have
- developed to create new methods for workflow validation

Thank you !