Table 1: Initial annotations on BPMN elements

| BPMN Element | | Game Element (Based on the proposed meta-model) | Annotation |
| --- | --- | --- | --- |
| Business Process | | A game level: The level number and the needed score to upgrade to it, depends on the business process complexity and importance, which are defined in Table 1. | GameLevel |
| Pool | | A game object for the game level corresponding to the process  (it is usually a place in the game which can have inner objects) | GameObject |
| Lane | Organizational role | Character: The role that the game scenario is written to teach the process to him is the player, and others are non-player. | PlayerCharacter  NonPlayerCharacter |
| System/Application | Game Object: Usually a PC | GameObject |
| Organizational unit | * Game Object: A place for the organizational unit * Characters for the organizational unit roles (one of them is player, and the others are non-player) | GameObject  PlayerCharacter  NonPlayerCharacter |
| Activity | Task | A new state for the game object corresponding to the task owner`s lane, and actions for it, or new actions for the previous state of the game object.  The designer can also define challenges for the task. | State  Action  Challenge |
| Sub-process | A game level: this level is activated inside another level which is corresponding to the sub-process owner`s business process. | GameLevel |
| Event | Start | The start state for the game object corresponding to the innermost event owner`s lane, and actions for the state, associated with the start of the game level. | State  Action |
| Intermediate | New state for the game level corresponding to the innermost event owner`s lane. | State |
| End | A new state for the game object corresponding to the innermost event owner`s lane, and actions for the state, associated with the end of the game level. | State  Action |
| Event Trigger Type | | A challenge on knowledge of the player corresponding to the event owner`s lane | Knowledge Challenge |
| Gateway | | A challenge on knowledge of the player corresponding to the gateway owner`s lane. The player should recognize the only correct output in Exclusive gateway, all the correct outputs in Inclusive gateway, and the correct output based on the received event in Event-based gateway. | Knowledge Challenge |
| **Note**: The challenge can be defined only on the gateways of the player owner`s lane. | | | |
| Flow | Sequence | New states and actions for the game elements assigned to the flow parties.  A specific action should be defined for connection with Process component of the game. The value of relatedElementType attribute is “bpElement” and the target business process element should be determined in the relatedBPelement attribute. | State  Action   * relatedElementType =“bpElement” |
| Data Association | A new state for the game object corresponding to the Resource.  Interaction between the owner`s object of the state corresponding to the task and the game object corresponding to the Resource | State  Interaction |
| Message | A path should be defined between the places corresponding to the pools. | GameObject |
| Resource | | Game object: A paper or an object in UI pages for Data Object, a shelf, or a document folder for Data Store. | GameObject |

Table 2. Secondary annotations

| Initial Annotation | Secondary Annotation | Description |
| --- | --- | --- |
| GameLevel | Scene | A scene to display the game level and its game objects in the screen. |
| GameObject | innerObjects | Such as elements of a UI page. |
| Components | Required components for the game object. |
| States | New states (if needed): the data of “State” row should be customized for them. |
| PlayerCharacter | A player character is a game object, so the data of the “GameObject” row should be customized for them. | |
| Capabilities | Static capabilities which are assigned to the player from the first and dynamic capabilities that will be assigned to the player based on the status of gameplay. |
| playerLog | An element for each player to store the player's game results. |
| NonPlayerCharacter | A non-player character is a game object, so the “GameObject” row data should be customized for them. | |
| State | actions | New actions (if needed) |
| changeRules | Rules for changing the state and the target state after applying the rule |
| Action | Constraints | Constraints of running the action |
| Triggers | Which action of a state should be executed so as to this action could be run?  **Note**: Setting the trigger of action is necessary, as the operations call chain is needed to generate sequence diagram at the next level. |
| relatedElementType | For each action, only one of the following 5 attributes can be set. In order to avoid conflicts, the type of action-related element should be determined in this attribute. |
| relatedLog | If the Action duty is to connect with PlayerLog element, the element should be defined in this attribute. |
| relatedChallenge | If the Action duty is to connect with Challenge element, the related challenge should be defined in this attribute. |
| relatedBPelement | If the Action duty is to connect with the Process component, the BPMN element should be defined in this attribute. |
| relatedScreen | If the Action duty is to show a UI page, the page should be defined in this attribute.  Design of UI pages takes place during the design process, whenever it is needed. |
| relatedArt | If art components should be presented for action execution, the components should be defined in this attribute. |
| isPlayerRequest | If the action is related to the player request, this feature should be set to “True”. |
| Challenge  (of any kind) | newObjects | The information of “GameObject” row should be customized for them. |
| States and actions for the challenge and for existing game objects | States and actions for challenge execution. An action named ‘runScoringRule’ should be defined for the challenge to run its scoring rule. The data of “State” and “Action” rows should be customized for them. |
| scoringRule   * ruleContext * rewardConditions | The scoring rule of the challenge and the context of rule execution  The reward-winning conditions are: 1) the reward policy; 2) the type of rewards that can be obtained; 3) the dynamic capabilities that are activated as a reward. |

Table 3. Setting previous and next elements for combined fragments

|  |  |  |
| --- | --- | --- |
| Combined Fragment | Sequence Diagram Picture | Rules for ordering the elements of B lifeline |
| Alt |  | * next (1) = Alt * previous (Alt) = 1 * previous (1.1) = previous (1.3) = Alt * next (Alt) = { operand1: 1.1, operand2: 1.3, !operand1 and !operand2: 1.5} * previous (1.5) = { operand1: 1.2, operand2: 1.4, !operand1 and !operand2: Alt} * next (1.2) = next (1.4) = 1.5 |
| Par |  | * next (1) = Par * previous (par) = 1 * next (Par) = {1.1a, 1.1b} * previous (1.1a) = previous (1.1b) = Par * next (1.1a) = next (1.1b) = 1.2 * previous (1.2) = {1.1a, 1.1b} |
| Opt |  | * next (1) = Opt * previous (Opt) = 1 * next (Opt) = { operand: 1.1, ! operand: 1.3} * previous (1.1) = Opt * next (1.2) = 1.3 * previous (1.3) = { operand: 1.2, ! operand: Opt} |
| Loop |  | * next (1) = Loop * next (1.2) = Loop * previous (Loop) = {1, 1.2} * next (Loop) = {operand: 1.1, ! operand: 1.3} * previous (1.1) = previous (1.3) = Loop |