Specifications for a BPMN 2 Process Editor and Workflow Engine for GLPI

Notes:

1. In grey: optional features (= that would be nice to have, but not mandatory)
2. Below the mention, “GLPI object” references an object that can own GLPI tasks: a Ticket, a Change or a Problem. Could it be a Project?
3. Terminology:
    a. A “Process” is a workflow definition, it contains Tasks, linked together with Gateways
    b. A Task is a set of Steps
    c. A Step, can be a form or a generated PDF document
    d. A Gateway is a conditional gate linking tasks. Gateway types are inclusive OR (OR), exclusive OR (XOR) and parallel AND (AND).
    e. A “Case” is a running instance of a Process
    f. Case variables: variables that scope is the running case.
4. Beware that the feature below are the minimal requirements!

Technologies

Must be using the same technologies than GLPI:

- PHP
- HTML/CSS Javascript
- MySQL (or MariaDB) (better if using PDO)
- Being open source, and having an active community!

Process Editor features

- Web application (no client installation), (integration into GLPI core?)
- Process management,
    - To add new processes,
    - To delete existing one,
    - To manage process versions,
    - etc…
    - includes hook to execute scripts on special events:
        - start of a case
        - delete of a case
        - cancel of a case
        - re-assign of a case (in fact re-assign of at least one task in the case)

BPMN 2 diagram (=process map) edition:

- Tasks:
    - Include one or more Forms that will be “executed” by the Workflow Engine (like a steps by steps wizard).

- Include conditions that are used to show or not the Forms in the steps by steps execution.
- Include assignment rules to compute which task(s) will be executed after the current one (called routing). These assignment rules can be static or dynamic (see below task assignment rules)
- Include hooks to execute scripts (PHP scripts):
  - Before forms: to do computations to impact what is shown in Forms,
  - After forms: to do computations that are using what has been input in Forms.
  - Before assignment: to do computations that are used to decide assignment for next tasks.
  - Before Routing: to do computations that are used to decide which task(s) will be the next task(s).
  - After Routing: to do computations that are based on the results of the routing (newly created tasks).
- Task types:
  - Form task: contains a list of forms (steps by steps wizard)
  - Service task: dedicated to call SOAP or REST API and return results.
  - Script task: PHP script to be executed in the process flow.
- Marker types (a task can be marked as follow):
  - Loop: the task is repeated until a condition is true.
  - Parallel: the task is instantiate and is assigned to people specified (static or dynamic). It is working more or less like a template task that is instantiate several times and each instance is assigned to a different person.

- Gateways (OR, AND, and XOR), divergences and convergences:
  - Inclusive (OR) gateways: one or more tasks are activated after current one (depending on conditions).
  - Parallel (AND) gateways: all tasks following current one are activated (no conditions). These // tasks are executed in // ☺.
  - Exclusive (XOR) gateways: one and only task, in the list of tasks following the current one, is activated.
- Pools and swim lanes: used to clarify the process map
- Start and end events
- Sub-process (a process by itself that can be used in several processes like a subroutine). When called, variables can be passed to the sub-process:
  - Synchronous sub-process: execution of the sub-process in parallel with the main process. The synchronous sub-process will return variables to main process when it finishes. The main process will wait for the sub-process to finish (depending on the diagram).
  - Asynchronous sub-process: execution of the sub-process in parallel with the main process. The asynchronous sub-process will not return variables to main process when it finishes.

- Database connections: to be able to connect to any kind of DB (mysql, postgresql, oracle, MSSql,…), and to use these connections in scripts or in Forms.
- A variable dictionary: it can be used to group all the variables used in the process (Case variables).

- Output documents: possibility to generate PDF using a localized template defined using the Case variables. This is great to generate PDF documents like “confidential agreements”, or else.
- Form editor:
    - Field types:
        - Textbox,
        - Textarea,
        - Dropdown (like select2), options may come from a static definition or a dynamic SQL request (made via database connections),
        - Checkbox,
        - Radio buttons,
        - DateTime picker,
        - Titles (H1, H2, …),
        - Images,
        - Buttons (normal or submit),
        - Tables (grids with add, delete, pages features),
        - …
    - Panels (to be able to add HTML code)
    - Sub-forms (to be able to include existing Forms in a main Form, like templates)
    - Fields must be mapped to Case variables (variables with scope in running case)
    - Dependent fields (ex: dropdown options that depend on a value of another field)
    - Field values: static or dynamic from SQL request (made via database connections),
    - Field default value: static or dynamic from SQL request (made via database connections),
    - Field validation: JavaScript formula and/or regex
    - JavaScript scripts:
        - To be able to include external JavaScript libraries
        - To be able to write JavaScript code that will be included in the Form during execution time.
    - Form sandbox-rendering display. It will permit to view a form without running the complete process.
- Localization must be available for user interface objects:
    - Forms,
    - Fields,
    - Tasks. In Tickets, Changes and Problems, tasks must be dynamically localized depending on the language of the GLPI user who is logged in.
    - Output documents,
    - Process map, …

Localization must be also available when PDF exports are generated for Tickets, Changes and Problems, using the “Print to pdf” export plugin.

- Time zone management must be available

Task assignment rules

Possibility to assign a task to a user or to a group of users: static (at task edition) or dynamic (during process execution via a PHP script).

- Cyclical assignment: the task is automatically assigned to one person within the list of available people.
- Manual assignment: the task is manually assigned to one person within the list of available people.
- Reports to: the next task is automatically assigned to the manager of the user who terminates the current task
- Value based assignment: next task is automatically assigned to a user via a Case variable (a PHP hook/script can be used to set the value of the variable).
- Self-service: next task is automatically assigned to a group, and anyone in the group can “claim” the task to execute it.
- Self-service value based assignment: next task is automatically assigned to a group via a Case variable (a PHP hook/script can be used to set the value of the variable), and anyone in the group can take the task to execute it.
- A timeout timer can be used to set time limit to execute the task. Then actions can be:
    - To send a reminder to the assigned user (or assigned group)
    - To escalade the task to the manager of the user (or to the manager of group)
    - To re-assign the task to another user.
    - To automatically execute the task with specified values for the form fields, and to derivate (route) the case to the next task(s).
- Possibility to define “super-users” in task properties. Those “super-users” can be re-assigned to the task in case of emergency (for example: assigned user can’t execute the task, he/she may be ill, or in holidays).

Workflow Engine

This part describes workflow engine capabilities and interactions with GLPI objects.

- Start a case (running instance of a process) from/for any GLPI object
- Delete or cancel a case (with rights)
- Start a case from mailgate (depending on GLPI business rules and/or information in the email).
- Start a case on a defined schedule, either recurring (i.e.: every Monday start a case from process XXX), either one shot (on next Monday, start case from process XXX).
- Start a case from simplified interface (post-only/self-service profile) using ticket type (incident or request) and ticket category.
- Start a case (process selection) with rights over entities/profiles, and/or business rules, and/or object category
- Possibility to start one or more processes (same or different processes) in a GLPI object (must be possible to limit the quantity of started processes).
- In a GLPI object, it should be possible to see the current running:
    - case task list, composed of zero or more tasks (is empty when case is completed = no more task to do):
        - case task assigned to current GLPI user:
            - possibility to re-assigned task to someone having rights on this task (or to a super-user)
            - possibility to assign task to a group with rights on the task (i.e.: the task will go “to be claimed”)

      - execution of the steps (forms, or output PDF document) in the task (steps by steps),
    - case task assigned to someone else:
      - view of a sum-up
      - possibility to re-assigned task to someone having rights on this task (or to a super-user),
      - possibility to assign task to a group with rights on the task (i.e.: the task will go “to be claimed”)
    - case task assigned to a group (to be claimed task):
      - possibility to claim this task if the current user is in the assigned group
      - possibility to assigned task to someone having rights on this task (or to a super-user),
      - possibility to assign task to a group with rights on the task (i.e.: the task will go “to be claimed”)
    - case sub-process: when task is executing a sub-process, show a sub-process sum-up with a link to it, to be able to execute sub-process.
  - case information sum-up,
  - case map: shows a read-only graphical map of the process to show the finished tasks, running tasks and future tasks.
  - case history: shows a list of the already done tasks with history (which tasks, who and when) with re-assignments, claims, etc…
  - case variable history (case variables that have been modified per each task)
  - case form history (view of what has been input in a finished form), but in read-only (no modification)
- In GLPI object linked to the case, evolution of the GLPI object tasks matching the process map:
  - Creation of new tasks matching the new tasks of the case (depending on the process evolution)
  - Changing the technician of the task if the case task has been re-assigned
  - Changing the technician of the task if the case task has be claimed (case task assigned to a group and claimed by a user belonging to the group)
  - Changing the status of the task when the case task has been executed
  - Changing the task to information when the case has been cancelled
- Possibility to create follow-ups in the GLPI object from a case hook (PHP script).
- Possibility to set the GLPI object task content from a case hook (PHP script)
- Possibility to set the GLPI object properties (like title, description, status, category, etc…) from a case hook (PHP script). For extended details: Case variables
  - General variables: to read/write info about GLPI
  - General item variables: to read/write info about GLPI object
  - Item actors: to get info about GLPI object actors
  - Next task: to write info about next GLPI object task
  - Current task: to write info to current GLPI object task
  - Follow-ups: to write follow-ups to GLPI object
  - Solution: to write solution info to GLPI object
- Possibility to start a new case in a GLPI object (current one or another existing one) from a case hook (PHP script)

- Possibility to interact between cases (send information from a case hook to another running case).
- Possibility to call SOAP or REST API from a case (synchronously or asynchronously). To be able to interact with other applications (send data to them or get data from them).
- Possibility to execute a task from an email via mailgate. The idea, behind this, is the possibility to approve a request (for example, a manager should approve a purchase order) via an email answer.
- Process versioning:
  - Possibility to have a new version of a process running in parallel with the former one. Explications: if needed former process running cases may continue to run using the former process definition.
  - Possibility to activate/deactivate start of new cases for a process

Notifications

Notifications must be by default the GLPI task notifications.

It must also be possible to define notification template that will be used by notifications dedicated per process task and per events:

- new task
- update task (this one can be used to send re-assignment notification to former assigned technician).
- task done

Also possibility to send spontaneous notifications from a case hook (PHP scripts). These are based on dedicated templates.

Legal point of view about task content modifications

Mandatory point: when done (i.e. case task has been executed by a user), the content of the GLPI object task MUST not be modified. For examples:

- purchasing requests when manager approval is needed,
- "rights request" for accessing sensitive data in an ERP.
- …

Profiles/rights management

- General rights should be managed per profile and per module:
  - BPMN editor module:
    - Read process
    - Create process
    - Update process
    - Delete process (this right is a bit special, as we must take into account that possibly there are Cases created from this process…)
  - Per profile, we may have rights on running cases:
    - Read: view the case with all its data

        - Cancel: this is a technical cancellation (not the same as if a manager refused to approve a purchase order). Doing this must change the TO_DO case tasks into informational task. DONE tasks are not touched.
        - Delete: if case has been started (created) by mistake, then it should be possible to delete it. Doing this must delete all case tasks in GLPI object (for this case ☺)!.
- Process rights should be managed per process, per profile and per entities (recursive or not)
    - Start new cases