# Mission Planning and Scheduling Services Outline

### 6.3.1 Planning Request Service [PRS]

| Operation | MAL Pattern | Description |
| --- | --- | --- |
| SubmitRequest | REQUEST | Send Planning Request to Provider, returns the Request ID and the Request Instance ID of Planning Request created.  Returns the IDs of all created objects (ID of the Request, ID of the RquestVersion)  The status is not returned. The MonitorRequestStatus shall be used.  If an error happens no instance is created and no reporting on status is performed accordingly. |
| UpdateRequest | REQUEST | Send updated Planning Request to Provider. It results in a new Request Instance.  Returns the new RequestVersionID.  The input structure is the RequestVersion Object.  It is required to send a complete RequestVersion Object for the update, rather a “delta”.  In case of error, the error message can contain the reason for failure  In future (next version of the BB) we can consider supporting the “delta”. |
| CancelRequest | SUBMIT | Send cancellation of Planning Request to Provider.  Input the Request ID. |
| GetRequestStatus | REQUEST | Returns current state of Planning Requests. Subject to filter: All, by Domain, ...)  Returns a list of <ID of the RequestIdentity, current ID of Request Version, Status> |
| MonitorRequestStatus | PUBSUB | Based on four keys, allows subscribing to the status updates for requests  Key 1: RequestIdentity ID  Key 2: UserID  Key 3: RequestStatus  Key 4: domain  Notification Structure: RequestUpdate (As it is in the current model amended with Request Instance ID) |
| getRequest | PROGRESS | Subject to filter: All, by Domain, by Source, RequestIdentity ID  Input: filter  Returns: List<RequestID,RequestVersion,RequestStatus> |
|  |  |  |

### 6.3.2 Plan Distribution Service [PDS]

Note that certain operations on Plans (ListPlans, GetPlan, GetPlanStatus and Monitor Plan) also apply to PatchPlans.

| Operation | MAL Pattern | Description |
| --- | --- | --- |
| ListPlans | REQUEST | Returns a list of available PlanIDentity IDs and PlanVersion IDs [or PatchPlan IDs], their type [Plan or PatchPlan] and their latest Status, using PlanInformation structure  Subject to filter: All, by Domain, by Status, by Predecessor, by Type [Plan or PatchPlan]. Note that for PatchPlans the Predecessor = the Source Plan. |
| GetPlan | REQUEST | INPUT: List (PlanIdentityID, Type [Plan or PatchPlan], PlanVersionID or PatchPlanID [optional, if not provided, then the latest one])  OUTPUT: List<PlanID, type [Plan or PatchPlan], PlanVersion or PatchPlan (including the PlannedItems)> |
| GetPlanStatus | REQUEST | Input: List <PlanIdentity ID, Type [Plan or PatchPlan], PlanVersion ID or PatchPlanID (optionl, if not provided means the latest)>  Output: List<PlanIdentity ID, Type [Plan or PatchPlan], Plan Status using the PlanInformation Structure (PlanVersionID or PatchPlanID to be added to it)>. |
| MonitorPlan | PUBLISH-SUBSCRIBE | Consumers register interest in a filtered set of Plans or PatchPlans (receiving the full content of the plan) When published by the Provider, these are then forwarded to all registered Consumers. Consumers deregister to stop receiving Plans. Subscription shall be based on   * (Domain is not a key, is always in the subscribtion). * PlanIdentity ID * Originator * Type [Plan or PatchPlan] * Plan Status (e.g. committed, ….) |
| MonitorPlanStatus | PUBLISH-SUBSCRIBE | Notifies the subscriber about the changes in plan status for the subscribed Plans (receiving just the ID and Status). Consumers deregister to stop receiving Plans. Subscription shall be based on   * PlanIdentity ID * PlanVersion ID * Originator * Plan Status (e.g. committed, ….) |
| MonitorPlanItemStatus | PUBLISH-SUBSCRIBE | Notifies the subscriber about changes in plan item [planning activity, planning event, planning resource] for the subscribed plans (receiving the corresponding Activity Updates, Event Updates and Resource Updates). Consumers deregister to stop receiving Plan Item Status. Subscription shall be based on:  PlanIdentity ID; PlanVersion ID; Currently Executing Plan  PlanningRequestID  Domain  Originator [User]  Item Types (Activity, Event or Resource)  Specified Activity Instances, Event Instances or Resources |
| QueryPlan | PROGRESS | Query for retrieving the plans  INPUT: Filter (PlanIdentityID,Filter on PlanInformation, Filter on EventType, Filter on ActivityType) not sure if also on EventInstance, ActivtyInstance?  OUTPUT: List<PlanID,PlanVersion(including the PlannedItems)>  This should be a separate Capability Set |

### 6.3.3 Planning Process Management Service [PMS]

To be deleted. Use M&C service to provide required interfaces for allowing automation of the planning processes.

### 6.3.4 Plan Execution Control Service [PEC]

| Operation | MAL Pattern | Description |
| --- | --- | --- |
| LoadOrMergePlan | REQUEST | Requests that the Plan Execution Provider loads or merges a specified Plan into the currently executing Plan (or Schedule).  The operation has two option flags: Load/Merge and Return Execution Status.  Load implies clearing and replacing the currently executing Plan (or Schedule) with the specified Plan.  Merge implies applying changes [relative to the predecessor Plan] to the currently executing Plan only [insertions, updates and deletions].  The provider acknowledges synchronously and, if the Return Execution Flag is set returns a final Plan Status once the Plan has been executed. |
| SubmitAction | SUBMIT | Sends Plan Execution Control directive to the Provider. The set of supported Actions is implementation specific, but is expected to include directives to Start/Stop and Pause/Resume execution of the Plan (or potentially a sub-plan by Planning Domain). |
| MonitorValue | PUB/SUB | Monitor value of Plan Execution function parameters. |
| GetValue | REQUEST | Returns the value of a Plan Execution function parameter. |
| SetValue | SUBMIT | Set the value of a Plan Execution function parameter. |
| ApplyPlanPatch | REQUEST | Apply a Patch Plan |
|  |  |  |

### 6.3.5 Plan Information Management Service

| Operation | MAL Pattern | Description |
| --- | --- | --- |
| AddRequestDef AddEventDef AddActivityDef AddResourceDef UpdateRequestDef UpdateEventDef UpdateActivityDef UpdateResourceDef | REQUEST    REQUEST | Manage current definitions for Planning Requests, Events, Activities and Resources (and potentially self-standing Constraints) with operations to Add, Update and Remove definitions. |
| ListRequestDefs  ListEventDefs  ListActivityDefs  ListResourceDefs  GetRequestDef GetEventDef GetActivityDef GetResourceDef | PROGRESS  REQUEST | Returns list of current definitions [Templates] for Planning Requests (also Planning Events, Planning Activities and Planning Resources).  Returns the specified Definition object. |
| RemoveRequestDef RemoveEventDef RemoveActivityDef RemoveResourceDef | SUBMIT |  |

### 6.3.6 Plan Edit Service

| Operation | MAL Pattern | Description |
| --- | --- | --- |
| InsertActivity UpdateActivity DeleteActivity  InsertEvent UpdateEvent DeleteEvent  UpdateResource | SUBMIT | Edit the currently executing Plan (or Schedule). Operations are provided to Insert, Update and Delete occurrences of Planning Activities and Events. Note that inserting a hierarchical Activity will result in the creation of an entire tree of Activities. When deleting an Activity, deletion of any sub-tree of Activities is optional. Resources do not have occurrences, and so may only be Updated. |
| UpdatePlanStatus | REQUEST | Update of Plan status by a third party. |
| applyPlanPatch | INVOKE | Apply a plan diff |