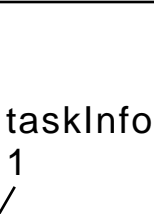
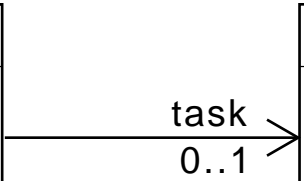


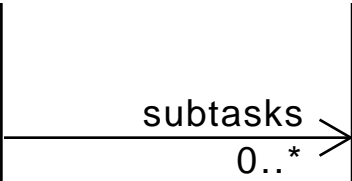
Task Runner
RESPONSE: String TIMESTAMP: String ERROR: String GENERIC_ERROR_RESPONSE: String STACKTRACE: String ALERT_HTML: String INFO_PRIVATE: String -taskInfo: TaskInfo
TaskRunner(aTaskInfo: TaskInfo) runTask() -addTimestamp() -addTimestamp(subtask: Subtask)



Task Info
-vocabulary: Vocabulary -version: Version -task: Task -dbTask: db.entity::Task -em: EntityManager -nowTime: LocalDateTime -modifiedBy: String
TaskInfo(aDbTask: db.entity::Task, aVocabulary: Vocabulary, aVersion: Version) TaskInfo(aTask: Task, aVocabulary: Vocabulary, aVersion: Version) getVocabulary(): Vocabulary getVersion(): Version getTask(): Task setDbTask(aDbTask: db.entity::Task) getDbTask(): db.entity.Task setEm(anEm: EntityManager) getEm(): EntityManager setNowTime(nowTime: LocalDateTime) nowTime(): LocalDateTime setModifiedBy(modifiedBy: String) modifiedBy(): String persist() process()



Task
-vocabularyId -versionId -subtasks: SortedSet<Subtask> -results: Map<String, String>
getVocabularyId(): Integer setVocabularyId(aVocabularyId: Integer) getVersionId(): Integer setVersionId(aVersionId: Integer) setSubtasks(aSubtasks: List<Subtask>) addSubtask(aSubtask: Subtask) addSubtasks(subtaskList: List<Subtask>) getSubtasks(): List<Subtask> getResults(): Map<String, String> setResults(aResults: Map<String, String>) addResult(key: String, value: String)



Subtask
-subtaskProviderType: SubtaskProviderType -operation: SubtaskOperationType -priority: Integer -provider: String -subtaskProperties: Map<String, String> -status: TaskStatus = TaskStatus.NEW -results: Map<String, String>
Subtask(aSubtaskProviderType: SubtaskProviderType, anOperation: SubtaskOperationType, aProviderClass: Class<? extends WorkflowProvider>) setSubtaskProviderType(aSubtaskProviderType: SubtaskProviderType) getSubtaskProviderType(): SubtaskProviderType setOperation(anOperation: SubtaskOperationType) getOperation(): SubtaskOperationType setPriority(aPriority: Integer) getPriority(): Integer setProvider(aProvider: String) setProvider(aProviderClass: Class<? extends WorkflowProvider>) getProvider(): String setSubtaskProperties(aSubtaskProperties: Map<String, String>) getSubtaskProperties(): Map<String, String> determinePriority() getStatus(): TaskStatus setStatus(aStatus: TaskStatus) getResults(): Map<String, String> setResults(aResults: Map<String, String>) addResult(key: String, value: String) compareTo(other: Subtask): int equals(other: Object): boolean hashCode(): int

«enumeration» TaskStatus
NEW PROGRESS SUCCESS ERROR

«enumeration» SubtaskProviderType
BACKUP HARVEST IMPORTER PUBLISH TRANSFORM

«enumeration» SubtaskOperationType
INSERT DELETE PERFORM

Workflow Methods
insertAccessPoint(em: EntityManager, existingAccessPoint: AccessPoint, __taskInfo: TaskInfo, isDraft: boolean, modifiedBy: String, nowTime: LocalDateTime, __schemaAP: schema.AccessPoint) deleteAccessPoint(ap: AccessPoint): List<Subtask> deleteVersionArtefact(va: VersionArtefact): List<Subtask> -addConceptTransformSubtasks(subtaskList: List<Subtask>) createHarvestPoolPartySubtask(operation: SubtaskOperationType, __vocabulary: Vocabulary): Subtask createImporterSesameSubtask(operation: SubtaskOperationType): Subtask createPublishSissvocSubtask(operation: SubtaskOperationType): Subtask