PIPELINE: Scheduling Extension

October 4, 2005

Contents

1 Overview

Up to now, the features of PIPELINE have been oriented around the methodologies of accomplishing a complex task involving the collaboration of many artists and automated procedures. With the exception of check-in log messages, PIPELINE has been largely agnostic as to how artist communicate with one another, their supervisors and the production staff. This document proposes an extension to PIPELINE which would integrate scheduling, the approval process, time sheets and other administrative features with the existing functionality of PIPELINE.

2 Tasks

The process of managing a production can be thought of as the process of breaking the project down into clearly achievable tasks and assigning these tasks to the available personnel in an order which minimizes the overall time needed to complete the project. There may also be client driven constraints which require certain tasks to be performed in an order needed to meet milestones along the way as well.

A Task is somewhat analogous to a PIPELINE Node in that they both represent a stage in a process. However, there isn't a direct one-to-one relationship between Nodes and Tasks. Tasks can be thought of as a more abstract and high level representation of the production. Typically, Tasks are usually created early in the planning stages of the production process. As the production progresses, Nodes may later created to implement some or all of the goals of an existing Task. Tasks may even be used to schedule work which never has any associated Nodes.

Due to these differences, Pipeline represents Tasks separately from Nodes but provides connections between these two systems where they interact.

2.1 Properties

2.1.1 Name

Each Task is given a unique name within the Project. This name should be a short identifier similar to the last component of a Node name.

2.1.2 Supervisor

The name of the user who has supervisory responsibilities for the Task. This could be a Lead Artist, a Producer or the Director for the Project. The Supervisor has special privileges and responsibilities related to the Task. Initially, the Supervisor is the user who created the Task but this can be changed later.

2.1.3 Owner

The name of the Artist to whom the Task is currently assigned. This does necessarily mean that the artist is currently working on the Task, only that the currently plan is for them to eventually complete the Task.

2.1.4 Start Date

The estimated date and time when work on the Task should have begun. This can be different from when work actually does begin. This information contributes to the process which determines the order Tasks are assigned and acts as a metric for evaluating Project progress.

2.1.5 Completion Date

The estimated date and time when all work on the Task should be complete and approved. Similar to the Start Date, this information is provided for scheduling and information purposes and may not exactly correspond to the actual completion date.

2.2 Tags

Privileged users may define an arbitrary set of Tags for each Project. Each Tag has an identifying name and a set of possible values used to classify Tasks.

For instance, a CGI project might define a Tag called TYPE with values such as: DESIGN, RESEARCH, TEXTURING, MODELING, RIGGING, ANIMATION, LIGHTING, COMPOSITING, TRACKING, etc... The same project might also define a SHOT Tag with values corresponding to the scene and shot breakdown of the project.

Each Task may select one or more of the defined values for each Tag. Tags may be either can be created in one of two modes: Exclusive or Multiple. Exclusive Tags require that the Task selects zero or one of the values for which make up the Tag. Multiple Tags allow several values to be selected at once.

The TYPE example Tag above would probably be an Exclusive Tag, while the SHOT example Tag would be a Multiple Tag. For instance, consider a modeling task which produces a model used in several shots. It might use the TYPE[MODELING] and SHOT[12_10,12_13,23_4] Tags.

Tags can be used to filter which Tasks are shown in the Task Timeline and to sort and group the shown Tasks according to Tag commonalities. Tags can also aid in the process of assigning Jobs to Artists capable of accomplishing the Task.

2.3 Constraints

Some Tasks cannot (or should not) be started or actively worked on until other Tasks have already been completed (CBB or FINAL Event) or reached its first intermediate state of completion (APPROVE Event). Until all constraints have been met for a Task, its START DATE cannot be moved to a time earlier than the Event on the constraining Task. In the case where no Events yet exist on the constraining Task, the START DATE cannot be moved to a time earlier than the START DATE of the constraining Task (for APPROVE constraints) or the COMPLETION DATE (for CBB or FINAL constants). While constrained, the Task will not accept START Events.

2.4 Current Nodes

Each task may be associated with a set of PIPELINE Nodes. Only checked-in versions of Nodes may be associated with a Task. The ADD NODE, REMOVE NODE and UPDATE events described below alter which Nodes and/or which version of a Node are associated with a Task. The most up-to-date set of checked-in Node versions associated with the Task make up the Current Nodes.

The history Task Events which change the associates node provide a way of determining which nodes and/or node versions where associated with the Task at any point in time.

2.5 Event History

Each Task maintains a log of the Task Events which have occurred for the Task. The Event History provides a detailed accounting of the life of a Task from creation to completion including all interaction between the Supervisors and Owners of a Task.

2.6 Task States

The Task State represents the currently state of progress in completing the Task. Task Events can be used to change the state of a Task. At any point in time, a Task will have one of the following states.

2.6.1 INACTIVE

No work is currently being done to accomplish the Task by the task OWNER if any exists. This is the initial state.

2.6.2 ACTIVE

The Task has an Owner which is actively working to complete the Task.

2.6.3 NEEDS APPROVAL

The Owner of the Task has completed an intermediate step toward the completion of the task which must be approved by the Supervisor before work can continue. The Owner will not actively work on the Task until the step is approved or otherwise commented upon by the Supervisor.

2.6.4 APPROVED

The Supervisor has reviewed and approved the previous intermediate step submitted by the Owner.

2.6.5 CHANGES REQUIRED

The Supervisor has reviewed the previous intermediate step submitted by the Owner and has submitted a message giving additional direction concerning the Task. The Owner of the Task should not continue working on the task until they have read the associated message.

2.6.6 HELD

The Supervisor or Owner has decided that all work on this Task should be temporarily suspended until further notice.

2.6.7 COULD BE BETTER

The Supervisor has reviewed the last approval request and has decided that the current state of the Task is good enough to be considered FINALLED. However, there are some remaining issues that could be addressed once all other Tasks reach a COULD BE BETTER or FINALLED state in order to improve the quality of the Project. No work should progress on this Task until further notice.

2.6.8 Finalled

The Supervisor has decided that the Task has been completed and no further changes should take place. Once a Task has been Finalled, only the Supervisor may request further changes to the Task.

3 Task Events

Task Events represent the occurrence of a significant change in the Task Properties, Dependencies or Current Nodes associated with a Task. Each Task Event is logged in the Event History for the specific Task which is the target of the Event.

All operations which modify the scheduling system involve generating a Task Event. Some of these events are generated by the direct interaction with the scheduling user interface and some are generated as a side effect of operations on the Nodes associated with a Task.

3.1 Event Properties

Task Events have a set of properties associated which differ depending on the kind of event. However, all Task Events contain at least a Name, Author and TimeStamp property.

The full set of properties are:

- Name: The symbolic name of type of Event which has occurred.
- Author: The name of the user who generated the Task Event. Usually this is either the Owner or Supervisor for the Task.
- Time Stamp: When the Task Event was generated.
- Message: A text message explaining the reason for the event and/or additional information or requirements to be associated with the Task.
- Node: The fully resolved name and revision number of specific checked-in Node who's association with the Task is being changed.

3.2 Event Types

The following details meaning and usage of the complete set of Task Events. There are restrictions on who may generate some types of Events. Each of the following Task Events begins with an Access section which lists the classes of users which are allowed to generate the Event. The possible values for Access are:

- Privileged: Only users with privileged status may generate the Event. Typically, only supervisory and administrative users have privileged status when using Pipeline.
- Supervisor: Only the user who's name matches the Supervisor property of the Task may generate the Event. Note that this user need not have privileged status.
- Owner: Only the user who's name matches the Owner property of the Task may generate the Event.
- Any: Any user may generate the Event.

3.2.1 Create

Access: Privileged

This Event causes a new Task to be created. The initial Name, Supervisor properties must be specified for the Task. In addition, an initial Message must be given which outlines the requirements for completion of the Task. The new Task will be created with a Task State of Inactive.

3.2.2 Assign

Access: Privileged, Supervisor

This Event changes the Owner of the Task. The Task State is unaffected. An optional Message may be given for this Event to explain the change of assignment and any new direction for completing the task.

3.2.3 Manage

Access: Privileged, Supervisor

This Event changes the Supervisor of the Task. An optional Message may be given for this Event to explain the change of supervision. Task State is unchanged by this Event.

3.2.4 Start

Access: Owner

This Event signals that the Owner of the Task has begun actively working on the Task and changes the Task State to Active. An optional Message may be given for this Event to explaining the intended course of action. This Event is generated when the Owner begins work on the Task for the day or switches to this Task from another Task.

3.2.5 Stop

Access: Owner

This Event signals that the Owner of the Task has ceased actively working on the Task and changes the Task State to Inactive. An optional Message may be given for this Event to explaining the reason work as ceased. This Event is generated when the Owner switches to work on another Task or leaves for the day.

3.2.6 Add Node

Access: Privileged, Supervisor, Owner

This Event adds one or more checked-in Node versions to the set Current Nodes associated with the Task. If a different version of any of the given Nodes are already associated with the Task an error will be generated. An optional Message may be given with this Event to explaining the reason for adding the Node to the Task. Task State is unchanged by this Event.

3.2.7 Remove Node

Access: Privileged, Supervisor, Owner

This Event removes one or more Nodes from the Current Nodes associated with a Task. An optional Message may be given for this Event to explaining the reason for the disassociation of the Nodes with the Task. Task State is unchanged by this Event.

3.2.8 Update

Access: Any

This Event signals the completion of an intermediate step toward the completion of the Task which should be brought to the attention of the Supervisor of the Task. However, this step does not require the Supervisor's approval. A Message must be provided explaining the progress which has been made. Task State is unchanged by this Event.

This Event may be generated manually by the Owner to log progress on a Task which is unrelated to any Node. More commonly though, this Event is generated automatically as part of a check-in operation for one or more of the Current Nodes associated with the Task. In this case, the Message property of the Event is copied from the check-in log message. The revision numbers stored for the Current Nodes are also updated.

Note that it is possible for users other than the Owner to generate this Event. It may be cause for concern if this Event was not expected by the Owner and Supervisor of the Task. The more liberal access to Any user is allowed so that renegade check-ins for Current Nodes associated with the Task will be brought to the Owner's and Supervisor's attention as soon as possible.

3.2.9 Submit

Access: Owner

This Event signals the completion of an intermediate step toward the completion of the Task which requires the Supervisor's approval. This Event is identical to the Update Event with the exception that it also changes the Task State to Needs Approval and can only be performed by the Owner of the Task.

When a Node is checked-in, if the user performing the check-in is the Owner of a Task associated with the Node, they are given the option of choosing whether an Update Event or Submit Event will be generated. If the user performing the check-in is not the Owner, an Update Event is always generated.

3.2.10 APPROVE

Access: SUPERVISOR

This Event signals that the Supervisor has reviewed and approved the intermediate step marked by a previous Update or Submit Event. An optional MESSAGE may be given for this Event to further elaborate on the approval. This event sets the Task State to APPROVED.

3.2.11 CHANGE

Access: SUPERVISOR

This Event signals that the Supervisor has reviewed the Task and has added additional directions and/or requirements which must be satisfied for the Task. A MESSAGE must be given which explains the required course of action. This event sets the Task State to CHANGE REQUIRED.

3.2.12 HOLD

Access: SUPERVISOR, OWNER

This Event signals that work on the Task should be temporarily suspended until further notice. An optional MESSAGE may be given for this Event to explain the hold. This event sets the Task State to HOLD.

3.2.13 CBB

Access: SUPERVISOR

This Event signals that the Supervisor has reviewed the previous intermediate step and decided that the current state of the Task is good enough to be considered complete. However, there are some remaining issues which could be addressed once all other Tasks have reached a COULD BE BETTER or FINALLED STATE. A MESSAGE is required to explain the remaining issues to be addressed. This event sets the Task State to COULD BE BETTER.

3.2.14 FINALLED

Access: SUPERVISOR

This Event signals that the Supervisor has reviewed the previous intermediate step and has decided that the Task is complete. An optional MESSAGE may be given for this Event as a final comment. This event sets the Task State to FINALLED.

3.3 Task Events for each Task State

The following tables describe the types of Task Events which can be generated based on the current Task State. The structure of the tables in as follows:

Initial State		
Event	*Access*	*New State*

Where the *Initial State* is the Task State before the event is generated. The *Event* is the generated Task Event. The *Access* section consists of a set of letter representing the types of users which may generate the Task Event: O=Owner, P=Privileged, S=Supervisor, A=Any. The *New State* is the Task State after the event has been generated.

(none)		
CREATE	P	UNFINISHED

UNFINISHED		
ADD NODE	PSO	(unchanged)
REMOVE NODE	PSO	(unchanged)
UPDATE	A	(unchanged)
SUBMIT	O	NEEDS APPROVAL
DIRECT	S	UNFINISHED
HOLD	SO	HELD

NEEDS APPROVAL		
APPROVE	S	APPROVED
HOLD	SO	HELD
DIRECT	S	UNFINISHED
CBB	S	COULD BE BETTER
FINAL	S	FINALLED

Approved		
Hold	SO	Held
CBB	S	Could Be Better
Final	S	Finalled

Changes Required		
Start	O	Active

Held		
Direct	S	Changes Required

Could Be Better		
Direct	S	Changes Required
Final	S	Finalled

Finalled		
Direct	S	Changes Required

(any)		
Manage	PS	(unchanged)
Assign	PS	(unchanged)

4 Projects

All Tasks and their associated Events are contained within a Project. A Project represents a logical unit of work which is largely independent. Examples include: a commercial, a film project or a video game. Within a particular Project, a Task must be unique and any specific Node may only be associated with one Task. However, different Projects may have Tasks which share association with a Node. For instance, a model may be used in one Project and then later be reused in another Project. It is even possible that more than one Project active at the same time may share a Node.

5 User Interface

The previous sections have concentrated on the mechanics of how Tasks and Event are represented and how they interact behind the scenes. This section describes the design of the interface presented to the user and tries to give some idea of a typical work flow using this system.

5.1 Task Timeline Panel

The Timeline is the primary visualization of a Project and its Tasks. Essentially, the Timeline is a graph with time as the horizontal axis and the list of Tasks as the vertical axis. Each Task is represented by a horizontal bar which spanning the period of time that the Task is active. This bar is segmented at Task Events and colored according to the Task State during the interval between Events.

The set of Tasks displayed may be filtered using a criteria involving the Owner and Supervisor names and any combination of Tag values. These same criteria are also used to sort and organize the displayed Tasks. For instance, it would be possible to view only those Tasks assigned to a particular user (or users) and arrange them in a nested hierarchy first by Shot and then by Type Tag values. Alternatively, only those Tasks which set the Model value of the Type Tag could be displayed sorted by Task Owner. The current Task State can also be used as part of the filter and sort criteria.

5.2 Task Details Panel

The Timeline can be used to select a specific Task or an Event for more detail. The Task Details panel provides access to all Task Properties, Tags, Constraints, State and Event History information associated with the Task. This information is presented in a form similar to the Node Details and History panels. The Timeline and Node Details panels will be linked so that selecting an Event from the Task's bar in the Timeline will cause the Task Details panel to scroll to the corresponding Event in the history.

5.3 Task Nodes Panel

This panel displays the set of Current Nodes for the Tasks selected in the Task Timeline. Nodes may be displayed in a textual table form or as a set of thumbnails images associated with the Node. The Edit operation is available for these Nodes within this panel. Also, if there is a Node Browser panel which shares the same panel group, the Nodes may be added to the selection of the Node Browser from this panel.

5.4 Event Generation

Task Events can be generated from the Timeline and Task Details panel using a context sensitive pop-up menu similar to that used to perform Node operations such as check-in and check-out. Only the set of events which are legal for the current user and Task State will be enabled in this pop-up menu.

5.5 Node Interface Changes

In order to keep Tasks in sync with the Node side of PIPELINE, some changes will be needed to the user interface related to Nodes.

Whenever a Node has been associated with a Task, the State of the Task and the relationship of its Owner to the current user should be represented whenever a Node is visualized. The most important place for this to occur is in the Node Viewer panel. One idea is for some small iconic modifier to be rendered near to the Node's symbol in the graph which would alert the user and provide an anchor for the the Task Event pop-up menu and a way to jump to the task in the Timeline panel.

When a Node operation is performed for a Node which has an associated Task some additional actions should be taken.

In the case of a Check-In operation where the current user is the Owner of the Task, an option should be added to the Check-In panel which allows the user to specify whether an UPDATE or SUBMIT Event should be generated. If the current user is not the Owner, a warning dialog should be displayed advising the user that they are not the Owner of the Task and should not proceed, if they choose to Check-In the Node anyway, an Update Event should be generated automatically. Regardless of the current user and Owner, a warning message should be generated if a Check-In is attempted if the associated Task is not in an ACTIVE state. In all cases the check-in log message will be used for the Task Events.

Any Node operation which would modify the working version of the Node (such as Edit, Link, Unlink, Renumber or Node property changes) should cause a warning dialog to be displayed before performing the operation if there exists an associated Task and the current user is not the OWNER of the Task or the Task is not in an ACTIVE state.

These warning shouldn't stop the user from performing the operation, but hopefully make them think about the consequences before proceeding.