Unclassified, Draft

A very abstract model of a Group log

Steve Powell

February 12, 2009

We model the log or history structure of a group used by a Group Coordinator in keeping a collection of group members synchronised.

This model is frozen (12 February 2009) pending the reintroduction of the requirement for group configuration synchronisation.

Contents

1 Intro

This simple model appears to capture the essential properties of the group state that a coordinator needs in order to understand how to direct the group members what to do to synchronise with the group configuration.

2 Basic Types

There are *Runtime Artefact Models* (*sic*), commands that change them, and group members that (may) execute the commands:

$$[RAM, COMMAND, MEMBER]$$

and there is an initial command, used in initial states:

$$\mid\ initCOMMAND : COMMAND$$

The group log is essentially an (alternating) sequence of models and command, each model being associated with (a set of) member responses. We therefore need to have a set of possible responses:

$$[RESPONSE]$$

the only feature of which is that there are two distinct responses which we might encounter...

$$\begin{array}{|l} ok, fail : RESPONSE \\ \hline ok \neq fail \end{array}$$

A member response is thus a response associated with a member, and a partial function records these for a collection of members:

$$Responses == MEMBER \rightarrow\!\!\!\!\!+ RESPONSE$$

Entries in the log (*transitions*) need to be identified by external parties, so we require that each has an identifier:

$$[TRANSITIONID]$$

and an initial identifier is required for the initial state:

$$\mid\ initTRANID : TRANSITIONID$$

3 Group log state

The Group log records a series of *RAM* states, with a command between each state, and a collection of responses from group members concerning their synchrony with that state.

Figure 1 shows a diagram which inspired this model.

There is no *a priori* reason that the sets of member responses need correlate although, in any particular case, the members of the group would appear here, and the responses indicate if they are (or were) synchronised with that state (based upon responses received by the coordinator).

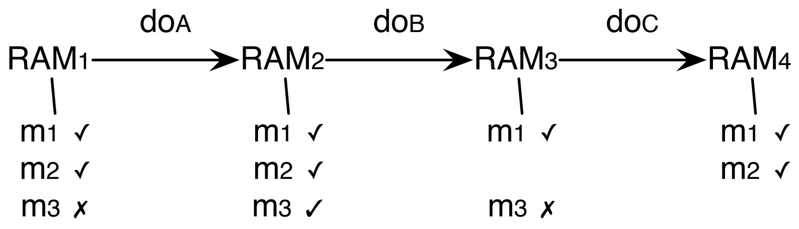

Figure 1: Diagram of a Group log

3.1 Parts of the log

We first model the *RAM* and *Responses* pairs, these are, effectively, a record of the group state – so we call it:

$$\begin{array}{|l}
\hline \mathit{GroupState} \\
\mathit{ram} : \mathit{RAM} \\
\mathit{responses} : \mathit{Responses} \\
\hline
\end{array}$$

The responses are a record of whether group members have reached that state.

We can say something (but not everything) about an initial group state:

$$\begin{array}{|l}
\hline \mathit{InitGroupState} \\
\mathit{GroupState}' \\
\hline
\mathit{responses}' = \varnothing \\
\hline
\end{array}$$

and leave the *ram* unspecified here. It turns out we need to use the initial group state both when recording new transitions into a log, and when initialising a new log. These have differing *RAM* values.

We now formulate a state transition, which is a command and (the resulting) group state, associated with a transition identifier:

$$\begin{array}{|l}
\hline \mathit{Transition} \\
\mathit{tranid} : \mathit{TRANSITIONID} \\
\mathit{fromCmd} : \mathit{COMMAND} \\
\mathit{GroupState} \\
\hline
\end{array}$$

The idea is that the command *led to* or *resulted in* the *ram* in the state.

The initial group transition can be specified as:

$$\begin{array}{|l}
\multicolumn{1}{l}{\_\_\ \mathit{InitTransition}\ \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_} \\
\mathit{Transition}' \\
\mathit{InitGroupState} \\
\hline
\mathit{fromCmd}' = \mathit{initCOMMAND} \\
\mathit{tranid}' = \mathit{initTRANID} \\
\hline
\end{array}$$

where again the RAM value is not constrained, since even the first transition in a new log will have to be formulated with whatever RAM is supplied initially.

3.2 Full log

We define a *group log* as a sequence of group state transitions:

$$\begin{array}{|l}
\multicolumn{1}{l}{\_\_\ \mathit{GroupLog}\ \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_} \\
\mathit{glog} : \mathrm{seq}\ \mathit{Transition} \\
\hline
\forall i_1, i_2 : \mathrm{dom}\ \mathit{glog} \mid (\mathit{glog}\ i_1).\mathit{tranid} = (\mathit{glog}\ i_2).\mathit{tranid} \bullet i_1 = i_2 \\
\hline
\end{array}$$

ensuring that all the transitions in the sequence have distinct identifiers, and that is all there is to it.

3.3 Initial log

The initial value of a group log is defined to record the (presumed) initial state of the group. This is a (given) intial RAM, no responses and the initial command. Thus:

$$\begin{array}{|l}
\multicolumn{1}{l}{\_\_\ \mathit{InitGroupLog}\ \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_} \\
\mathit{initRam}? : \mathit{RAM} \\
\mathit{GroupLog}' \\
\hline
\mathit{glog}' = \langle (\mu\, \mathit{InitTransition} \mid \mathit{ram}' = \mathit{initRam}? \bullet \theta\, \mathit{Transition}') \rangle \\
\hline
\end{array}$$

Note that this last constraint ties down the initial state completely, as a function of the initial RAM parameter.

4 Group log operations

After initialisation, the following operations are envisaged:

RecordTransition where a group transition is added to the log, with a new identifier;

RecordResponse where a member response is recorded in the log at the identified transition;

LogProjection where a log is (potentially) reduced to account for a given set of group members;

LogTruncation where a log is truncated so that only relevant transitions for a given set of members is retained.

The assertion is that we can define these operations purely in terms of the state provided, accomodating the major coordinator functions without having to know of what the *RAM* consists, or what the *COMMAND*s actually do.

The transition identifier introduced above becomes relevant in response recording, but requires refinement if automatic resynchronisation is required. The identifiers then have to form an unbroken sequence in order to have a common understanding of the coordination events that have flowed between the members.