HPR Scheduling System - Design Notes

Dave Morriss

Last Updated: 2012-11-09 19:59

Discussion Points

1. How will the scheduler be invoked?
   - It might be desirable to run the script in a web-based mode responding to REST requests.
   - It might be easier just to call it out of a superior script.

2. What will the scheduler be required to return?
   - In single show mode:
     - UUID/GUID of next show
     - show number allocated
     - date of release
   - In queue reporting mode (assuming there is a need):
     - as above but repeated
     - further details such as title, host, etc.

3. How will the queue of pending shows be notified to the scheduler?
   - Stored in a database?
   - Stored in a YAML file? (**Note** the current system under development can read and write YAML)

4. How will the pending queue (in database or YAML file) be managed?
   - Presumably a separate tool will scan the accumulated ATOM files/fragments and construct this

5. What attributes of a show will be needed to perform the required actions? At the moment the types of show attributes catered for are:
   - UUID/GUID: allocated elsewhere as an unique show designator
   - Tag(s): used to indicate show characteristics of interest to the scheduler. Tag values are fairly arbitrary, they are only of relevance to the code elements which look for them. They would need to be defined somewhere to ensure consistency when adding show details. Examples:
     - COMNEWS: The show is a Community News show for the first Monday of the month
     - DEFAULT: The show is a standard user-contributed show to be scheduled whenever there is a free slot
     - PRIORITYn: A way of giving a higher priority to a show should it need to be boosted above a default show, for example
     - DEBUT: First show from a new host
     - LITS: Linux in the Shell, every second Tuesday
     - TGTM: Talk Geek To Me, every Friday
     - THURSDAY: Syndicated Thursday show
     - DATE-20121201: Show to be released on a specific date
     - SHOW-2048: Show to be released when a certain show number has been attained

- It might be desirable to provide a means of translating the show UUID/GUID into displayable details if the scheduler is to be used to produce a pending queue summary.

6. What other contextual elements will the scheduler need to be aware of?
   - The current show number so that it can compute the next
   - The current date so that it can compute the next release date
   - The names/handles of all show hosts so that a new one can be detected and the 'DEBUT' tag allocated (debatable)
   - Recent show history so that it is possible to detect that a host has already had a show released this week.
   - Full show/host history to enable automatic re-prioritisation if (for example)
     - a show has been delayed more than n days in the queue
     - a *dormant* host has submitted a show after a long hiatus

7. How will the scheduling algorithms be represented?
   - *Triggers* representing *if-then-else* expressions will be processed by the software in a priority order
   - Triggers will be held in a database or a YAML file between scheduler invocations
   - Triggers will need to be managed and maintained, some of which could be partly automated. For example, the trigger which checks for SHOW-2048 should be purged after that show number has passed. Triggers will also need to be added, edited and removed manually, so a tool for doing this is needed.

8. How will the tests used by the triggers be represented?
   - These will be coded as expressions based on a series of functions (actually object methods) which are defined within the scheduler library
   - Example:
     - `dayisweekday($dt) && tagExists($shows,'DEBUT')` - this determines whether the date being scheduled for is a weekday and whether a DEBUT show exists

9. How will the trigger actions be represented?
   - These will be coded as function calls as above
   - Example:
     - `findShowWithTag($shows,'DEBUT')` - this finds the next show marked with a DEBUT tag

10. Might it be desirable to allocate all hosts UUID/GUID numbers to avoid confusion with varying spellings, etc?