CrawlDatumStates

Note: information here is specific to Nutch 1.x - conceptually the state machine should be identical in Nutch 2.0 but implementation details are different.

Nutch 1.x maintains state of pages in CrawlDb, which is updated by various tools:

*Injector - to populate CrawlDb with new URLs
*Generator - to generate new fetchlists, and optionally mark those URLs in CrawlDb as "being in the process of fetching"
*CrawlDb update - to update the CrawlDb with new knowledge about the already known URLs (already in CrawlDb) as well as add new URLs discovered from page outlinks.
*ScoreUpdater updates the CrawlDB with LinkRank calculated URL scores.

Below is a state diagram of CrawlDatum, which is a class that holds this state in CrawlDb.

CrawlDatum.png!

(This diagram was created with UMLet, the source file in UMLet format is CrawlDatum.uxf).

Notes

After initial injection URL is in an "unfetched" state, which means it's eligible for being selected by Generator for fetching. Whether it actually becomes selected depends on Generator settings, e.g. topN or maximum allowed URLs per host in a fetchlist.

If generator.update.crawldb property is set to true, then Generator will first prepare a fetchlist and then update CrawlDb to mark the selected URLs as "in the process of being fetched". This prevents Generator from selecting the same URLs if run twice in a row without an intervening updatedb. This state is here called "generated", though in reality it's implemented as a piece of metadata and not a separate CrawlDatum status.

In order to avoid entries being stuck in this state if there is no subsequent updatedb, there is a timeout mechanism that resets back URLs to "unfetched" state after a longer period of time (default is 7 days).

Fetcher creates a record of fetching results in the form of a fetch status (incidentally, and confusingly, implemented using the same CrawlDatum class). Parsing creates more CrawlDatum records that represent outlinks, each leading to a potentially new URL. The latter are referred to in the diagram as "linked".

During the updatedb job all this data is merged into a new updated CrawlDb. Depending on the input data the outcome of this updatedb process may be different.

If there was a temporary problem in fetching (e.g. exception or time out) then this URL is left as "unfetched" but its retry counter is incremented. If this counter reaches a limit (default is 3) the page is marked as "gone". Pages that are "gone" are not considered for fetching by Generator for a long time, which is the maxFetchInterval (e.g. 180 days) - the reason for keeping them is that even gone pages may re-appear after a while, and also we want to avoid re-discovering them and giving them a status of "unfetched".

Other possible states after fetching are "truly gone" 😉 (e.g. forbidden by robots.txt or unauthorized), which get the same treatment as described above - that is after a long period of time we check again their status, which may have changed.

In case of "success" we mark this URL as "fetched". This URL is not eligible for re-fetching until after fetchInterval, at which point it's considered outdated and in need of re-fetching (i.e. the same as "unfetched").