On the Application Post-Print of the Shanghai Newspaper Group Manuscript Distribution Monitoring System Solution

Authors: Hua Xiuwen

Date: 2023-10-08T00:00:00+00:00

Abstract

The news gathering, editing, and distribution system reduces workflow procedures, improves work efficiency, shortens publication time, and enhances news timeliness, while also imposing higher requirements on technical personnel. This paper primarily focuses on implementing a monitoring solution for the Shanghai United Media Group' s distribution system, specifically addressing how to ensure real-time and effective monitoring of the system' s operational status and improve technical personnel' s efficiency in handling missed articles.

Full Text

Preamble

Incoming information is decoded and written to the shared folder of the manuscript receiving server. The editorial server polls this shared directory every 15 minutes, traversing files generated since the last check based on the acceptance timestamp of the most recent file. When a new file is detected, the system opens its index to identify the required distribution type set, recording it into the comparison database using the filename as the key and the type set as the attribute. The system then reads the new manuscript, classifies it according to its attributes, processes its distribution, and makes it available to users. As shown in Figure 2 [Figure 2: see original paper], for text files, the system proceeds to the next file upon completing these operations; for image files, it additionally verifies whether the image download is complete.

2.2.2 Main Functions Overview

According to Figure 3 [Figure 3: see original paper], this section provides a brief introduction to the system' s primary functions.

2.2.2.1 Manuscript Information Statistics

This function comprises three components: information recording, information publishing, and timed polling.

Information Recording: The workflow is illustrated in Figure 4 [Figure 4: see original paper]. The monitoring system first traverses the comparison database, extracting records marked as unclassified since the last polling cycle. It then searches the editorial distribution database for distribution records of these files, comparing the distribution type set against the distribution records to check for matching elements. If all elements match, the record in the comparison database is marked as classified. If no record is found or the found record does not match the elements in the type set, the unmatched elements are inserted into the comparison database as attributes with "unclassified" as the result.

Information Publishing: As shown in Figure 5 [Figure 5: see original paper], this component displays recorded data information on web pages. It primarily uses WebLogShowService, WebPhotoService, and WebTextService to retrieve data, stores the retrieved data using WebLogShow, WebPhotoClass, and WebTextClass, then uses ViewControl to integrate and display the data, with Default presenting the integrated data on the web interface.

Timed Polling: Implemented through cyclic programs that periodically initiate traversal routines, as depicted in Figure 6 [Figure 6: see original paper]. The timed polling mechanism consists of three cyclic programs: matchtest, matchWenHui, and matchXinMin. The matchtest program calls the matchWenHui and matchXinMin subroutines, which in turn invoke Check() to traverse and verify classification status.

2.2.2.2 Alarm Reminders

Alarm reminders consist of three components—web alarm, sound alarm, and message push—providing multi-sensory alerts to operators for processing alarm information.

Web Alarm: When unclassified or unmatched situations occur, the system calls the Color() function in the program, changing the background color of "Last Received File Time" and "Last Distribution Time" from green to red. This prominent visual cue enables technicians to promptly identify missed manuscripts and undistributed content in the editorial system.

Sound Alarm: Based on the web alarm, when a page alarm is triggered, the system calls the Sound() function to play the Sound.mp3 file from the database. This allows technicians to receive monitoring system alerts through audio cues when they cannot view the monitoring page, significantly improving work efficiency.

Message Push: Primarily implemented through SMS, message push focuses on delivering real-time alerts about missed manuscripts, including specific times-

tamps and content descriptions, to ensure information timeliness, accuracy, and effectiveness. SendMessage operates as an independent thread within MessageService. While all Service.Server processes utilize a unified set of configuration information, different types of Message information are extended using the Adapter pattern, with ServiceServer' s Adapter being dynamically loaded through configuration.

2.2.2.3 Log Recording

Log recording primarily tracks times when files are not received, periods when manuscripts remain undistributed, and details of undistributed manuscripts. When manuscripts are not distributed, the system can automatically identify missed manuscripts based on editorial distribution rules and record and display them in the log system, eliminating the need for manual retrieval and comparison. This significantly enhances technician productivity.

2.3 System Interfaces

For enterprise developers, the system provides comprehensive business operation interfaces to facilitate integration and invocation. The key interfaces are as follows:

- **ReturnRecord**: A custom class for returning records
- **FolderClass**: A custom class for manuscript folder traversal
- **PhotoFolderClass**: A custom class for image manuscript traversal, inheriting from FolderClass
- **TravelDataBase**: Function to traverse records of undistributed image manuscripts in the database
- **TravelDirectory**: Function to traverse all date folders and determine write times for newly received files
- **CheckPhotoDirectory**: Function to traverse folders of newly received files and identify required text manuscripts based on file type and write time
- **Material**: Base class for matching entities
- **FoundFile**: Function to find the last write time of a file based on its path
- **FoundTitle**: Function to access XML files via full path and return strings without single quote replacement
- **OneToTwo**: Function to add leading zeros to single-digit numbers
- **PhotoClass**: Image manuscript instance class
- **CheckPhoto**: Function to check whether a manuscript has been distributed and enter it into the database
- **InsertOrUpdateSQL**: Function to submit database requests based on custom class instances
- **FoundRecord**: Function to find records based on parameters; if search fails and file creation time is after 23:00, performs an expanded query based on title

- **LoadObjectByPhotoName**: Function to check the cache for manuscripts with the same name based on filename
- **AddCache**: Function to add new PhotoClass instances to the cache
- **GetPhotoName**: Function to extract manuscript names from the database based on filename
- **TypeTransition**: Classification matching instance
- **PhotoQuestion**: Image manuscript entity
- **SQL function custom class**: Custom class for SQL function execution
- **ExecuteInsertSQL**: Function to submit insert requests to the database based on PhotoClass instances and record in log table
- **ExecuteUpdateSQL**: Function to submit update requests to the database based on PhotoClass instances and record in log table
- **GetUnMatch**: Function to find lists of image manuscript records that have not completed distribution

Note: Figure translations are in progress. See original paper for figures.