Automatic updating of a combine harvester knowledge-based system by webpages and user-uploaded files

Hongxin Liu[1], Yiming Zhang[2], Yongtao Xie[2], Yu Zhao[3*]

(1. *College of Mechanical and Electrical Engineering, Suqian University, Suqian 223800, China*;
2. *College of Engineering, Northeast Agricultural University, Harbin 150030, China*;
3. *College of Electrical and Information, Northeast Agricultural University, Harbin 150030, China*)

Abstract: Currently, knowledge-based systems are generally updated manually, resulting in long cycles, low efficiency, and difficulty in ensuring comprehensive and systematic supplementary content. To address this issue, a knowledge-based system automatic updating method based on webpage information and user files is proposed for combine harvesters. The knowledge storage structure in an electronic warehouse is analyzed, and knowledge and data types are determined. A crawler designed to locate the combine harvester-related information in the target webpage realizes the acquisition and sorting of webpage knowledge and data. The type of file uploaded by the user is set based on the knowledge base and data type; the content of the user file is extracted and filtered, and the knowledge and data of the knowledge-based system are user-defined. The organized knowledge and data are stored or updated in the knowledge base to achieve automatic updating of the knowledge-based system based on webpage information and user files. The test results show that based on user file updates, the knowledge and data of the knowledge-based system can be customized according to user needs. This meets user requirements and allows for effective automatic updates based on webpages after automatic updating of the knowledge base based on webpages. After the automatic updating of the knowledge base based on webpage information is triggered, the knowledge and data from the three webpages that were initially crawled and read are automatically updated to the knowledge base in 5.265 s. Realizing the automatic updating of knowledge and data can shorten the updating period, maintain the effectiveness and practicability of the knowledge-based system, ensure the scientific and advanced nature of the intelligent design process, and provide technical models and methods for the knowledge collection of similar knowledge-based systems.

Keywords: knowledge base, automatic updating, web-based, user upload files, data acquisition and processing
DOI: 10.25165/j.ijabe.20251803.9073

Citation: Liu H X, Zhang Y M, Xie Y T, Zhao Y. Automatic updating of a combine harvester knowledge-based system by webpages and user-uploaded files. Int J Agric & Biol Eng, 2025; 18(3): 165–175.

1 Introduction

Establishing knowledge-based systems related to combine harvesters for knowledge accumulation and platform management can improve knowledge and information management efficiency and quality for complex machine systems such as combine harvesters and is a core form of intelligent design support[1]. Using knowledge-based systems to effectively manage combine harvester knowledge helps designers use design knowledge flexibly and enhances the degree of knowledge inheritance and reuse, thus improving the design efficiency of combine harvester products[2-4]. Knowledge is the core element of knowledge-based systems. To maintain the validity and practicability of knowledge-based systems[5-7], timely updates should be carried out to ensure the immediacy of such knowledge.

A machinery and equipment knowledge-based system stores knowledge according to its characteristics. However, at present, the storing and updating of knowledge are mostly carried out through the collection and processing of knowledge by the designer or the addition of knowledge by the user[1,2,8-10]. Knowledge updates include collection, collation, identification, and storage steps. Manual updates face problems such as long data updating cycles and low efficiency in data collection. Manual updates cannot keep up with the growth rate of actual knowledge, especially for large knowledge-based systems, and thus cannot meet the demand for large-scale knowledge-based updates.

No reports have been published on the updating methods of knowledge-based system data for combine harvesters neither in China nor abroad. However, scholars in China and abroad have conducted relevant studies on knowledge-based data updates in other fields, focusing mainly on knowledge data acquisition and data management. For example, Pannu et al.[11] introduced a dark web crawler to obtain information from suspicious and malicious websites. Similarly, Peng et al.[12] designed a web crawler to collect and acquire shipping job application information in a web environment. These two kinds of data collection approaches can provide references for the automatic update of knowledge and the acquisition of data. Wang et al.[13] designed a data acquisition and integration system for a manufacturing workshop to address insufficient data utilization. The Huazhong University of Science and Technology[14] designed a computer numerical control (CNC) industrial data acquisition and storage system, which can be used to effectively associate various types of data generated by CNC equipment. These studies can fully integrate collected data with a

Received date: 2024-05-14 **Accepted date:** 2025-05-20

Biographies: Hongxin Liu, Professor, PhD, research interest: research on digital and intelligent design, Email: lcc98@163.com; **Yiming Zhang**, MS, research interest: research on digital and intelligent design, Email: zhang__yi_ming@163.com; **Yongtao Xie**, PhD candidate, research interest: organic fertilizer spreading mechanization technology and equipment, Email: 18678698606@163.com.

***Corresponding author: Yu Zhao**, PhD, research interest: agricultural artificial intelligence. Northeast Agricultural University, Harbin 150030, China. Tel: +86-13946084083, Email: yuzhao@neau.edu.cn.

system, providing new ways to apply new knowledge and data in the automatic updating process. Yang[15] mined correlations in learning content from behavioral data to form a knowledge-based network and analyzed, refined, and sorted data from learners' own information, behavioral data generated during the learning process, and relevant data from learning content to continuously adjust the knowledge-based network. Hua[16] created a domain-specific knowledge base in the form of a knowledge graph and realized semiautomatic updating of the domain-specific knowledge base by collecting user query logs and performing cluster analysis. In these studies, knowledge updates were performed based on knowledge-based systems and thus provided reference schemes for the automatic updating of a combine harvester knowledge-based system. Ganapati et al.[17] used DataHub to establish a global cooperation platform that can connect to global data centers and used the platform to perform various operations, such as searching, analyzing, cleansing, and integrating data for data management purposes. Roh et al.[18] performed data analysis after the separate storage of data extracted from the web and enterprise datasets during the data collection process, allowing users to preliminarily explore a dataset to determine its degree of usefulness. These two data management processes provide solutions for the intelligent updating of knowledge.

In summary, data updates are realized mainly by using web platforms and equipment as data sources or mining the connections within knowledge-based data, with less communication and contact with users. Knowledge-based system updates involve knowledge acquisition and the replacement or supplementation of existing knowledge data with newly acquired knowledge data. In the knowledge acquisition process, internet information has the characteristics and advantages of rich content and fast update speed, and each user has his or her own personalized knowledge needs and sources of knowledge that are tailored to his or her own characteristics. This study makes full use of the advantages of these two knowledge sources, uses the knowledge-based update form based on webpages and user-uploaded files, uses the preexisting combine harvester knowledge-based system architecture, details the in-depth analysis of the automatic updating of knowledge-based system data, and addresses problems such as those related to the lag and omission caused by the manual updating of knowledge-based systems. The proposed automatic updating of the knowledge-based system can yield the latest knowledge data in a timely manner, which shortens the knowledge data updating cycle, maintains the effectiveness and practicability of the combine harvester knowledge-based system, ensures the scientific accuracy of a series of intelligent design processes, including a model base supported by the knowledge base system, and provides a technical model and reference for similar knowledge-based system knowledge acquisition and processing methods.

2 Overall plan and development method

2.1 Overall plan

A combine harvester knowledge-based system is a functional subsystem of a product data management (PDM) system independently developed by the team of this study that supports the intelligent design of a combine harvester. The PDM system is composed of a knowledge-based system, a model-based system, an engineering analysis system, and other subsystems. The tabular data files in the knowledge-based system are stored in the SQL Server database. The primary data files (.mdf) and transaction log files (.ldf) generated upon storage, along with core files such as models, images, and documents, must comply with the unified storage path regulations managed by the electronic repository. The automatic updating of knowledge and data in the combine harvester knowledge-based system occurs in an electronic warehouse, including SQL Server data tables, picture libraries, TXT files, and PDF files. By examining these knowledge types in the existing system, this study provides new knowledge and data from webpages and user-uploaded files and introduces a research scheme for the automatic updating of the knowledge-based system. The overall scheme is shown in Figure 1.

Internet information is abundant and updated rapidly. Through the screening and processing of webpage information, the relevant knowledge and data of a combine harvester can be efficiently and quickly acquired. To acquire combine harvester knowledge and data in the web environment, a web crawler is designed in this paper to retrieve targeted information from websites, and the collected knowledge and data are processed and updated in the database and electronic warehouse.

A web crawler is essentially an information crawling program that crawls and downloads web information in a systematic, automated, and orderly manner according to specific logic and algorithm rules[19]. A web crawler accesses web information based on one or more initial uniform resource locators (URLs) and crawls web pages according to specific rules to obtain the webpage content, analyzes the structure of the webpage, parses the obtained HyperText Markup Language (HTML) files to extract the required information, and analyzes the other webpage's links through recursive retrieval to obtain the uncrawled URL queue[20]. The crawler does not stop until the URLs in the uncrawled queue are all crawled or until other predefined conditions are met. The knowledge and data crawled by the crawler cannot be used directly. The data tables need to be organized in accordance with the data table storage format specified by the combine harvester knowledge-based system in question. The data are subsequently updated in the SQL Server database[1]. For the picture and document data obtained from the webpages, after the data form is determined, the data are stored in the appropriate locations in the electronic warehouse to complete the update process.

For files uploaded by users, user preferences are retained, the types of the uploaded files are determined, and the location and content of the knowledge and data are selected by the user; then, a customized update of knowledge and data is completed.

2.2 Multilingual hybrid development

The Python language has a wealth of third-party libraries that can be conveniently referenced to quickly implement knowledge and data analysis and processing. Python is the best development language for knowledge and data acquisition and processing[21]. In this work, webpage information is used as a source of knowledge and data for the automatic updating of the knowledge-based system, and a crawler is developed using Python to access the webpage and capture and process the knowledge and data.

The Python language is used for the background process; the development program of Python is integrated with vb.net, which is the main development language of knowledge-based systems, and vb.net is used for the foreground process. Thus, multilingual hybrid programming is realized. The specific algorithms for web data sources and the processing of knowledge and data are transferred to the more advantageous Python language for execution. The problem of complex script programs in the vb.net language when developing crawlers is transformed into the problem of how to use the Python language to develop crawlers and transfer the knowledge and data

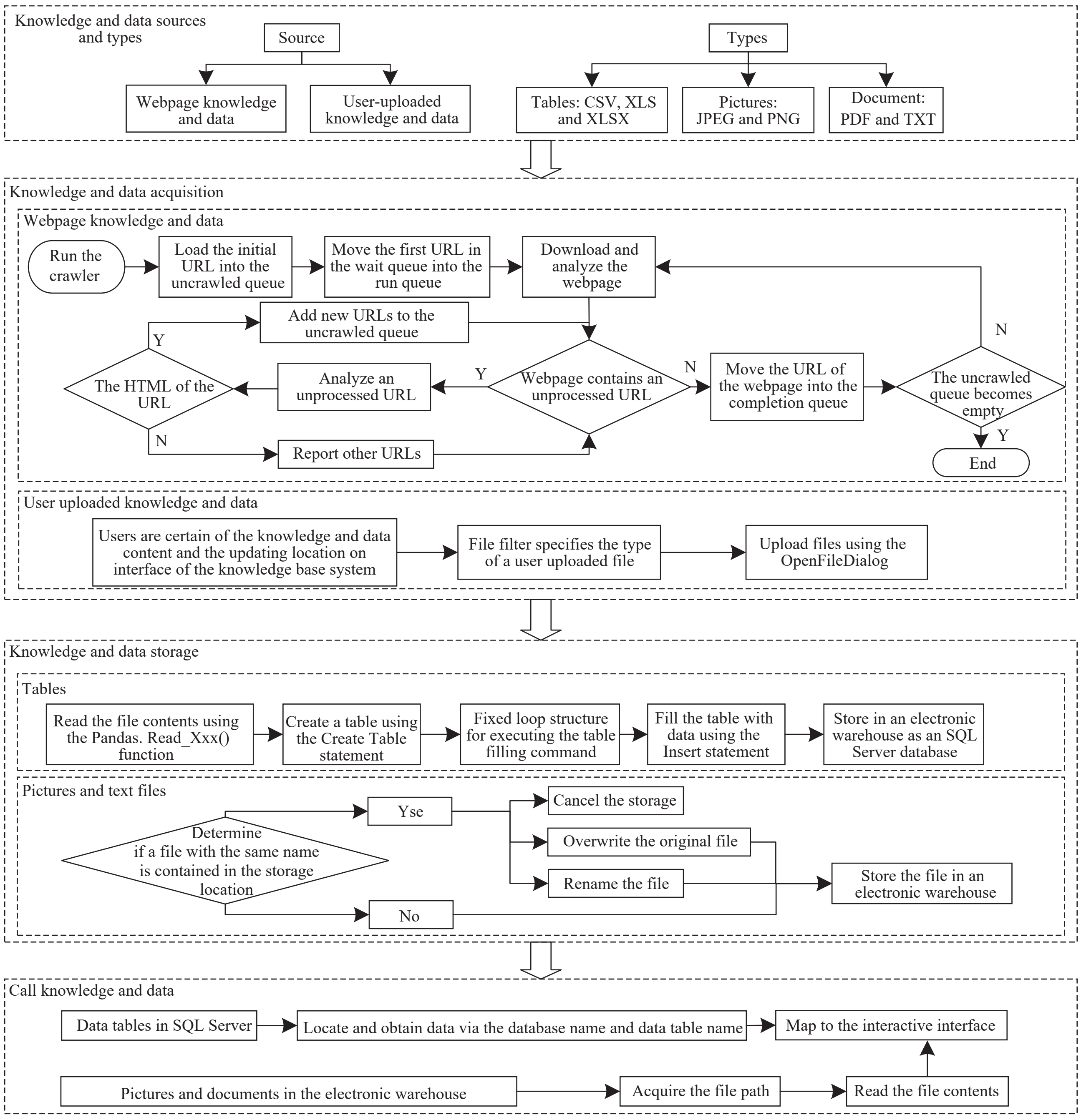

Knowledge and data sources and types
Source
Types
Webpage knowledge and data
User-uploaded knowledge and data
Tables: CSV, XLS and XLSX
Pictures: JPEG and PNG
Document: PDF and TXT
Knowledge and data acquisition
Webpage knowledge and data
Run the crawler
Load the initial URL into the uncrawled queue
Move the first URL in the wait queue into the run queue
Download and analyze the webpage
Add new URLs to the uncrawled queue
Y
The HTML of the URL
Analyze an unprocessed URL
Y
Webpage contains an unprocessed URL
N
Move the URL of the webpage into the completion queue
N
The uncrawled queue becomes empty
Y
End
N
Report other URLs
User uploaded knowledge and data
Users are certain of the knowledge and data content and the updating location on interface of the knowledge base system
File filter specifies the type of a user uploaded file
Upload files using the OpenFileDialog
Knowledge and data storage
Tables
Read the file contents using the Pandas. Read_Xxx() function
Create a table using the Create Table statement
Fixed loop structure for executing the table filling command
Fill the table with data using the Insert statement
Store in an electronic warehouse as an SQL Server database
Pictures and text files
Determine if a file with the same name is contained in the storage location
Yse
Cancel the storage
Overwrite the original file
Rename the file
Store the file in an electronic warehouse
No
Call knowledge and data
Data tables in SQL Server
Locate and obtain data via the database name and data table name
Map to the interactive interface
Pictures and documents in the electronic warehouse
Acquire the file path
Read the file contents

Figure 1 Overall scheme

acquisition and processing results to the vb.net main program. The use of the unique advantages of different languages can reduce the complexity of program design and improve the efficiency of knowledge-based system development[22-24].

Python programs can generate independent executable files in scripts, which can be integrated into the main program by calling. Specifically, the .py source program is packaged in the Python interactive environment in the command prompt window, and the generated executable file is automatically stored in the "dist" folder under the source program folder.

When running the main program, the process is referred to as the foreground process of the Python program. The Process.Start() function (used to start a separate child process and execute the target function asynchronously) of the main program executes the Python background process and passes the output result on to the main program. This process maintains the independence of each algorithm and makes full use of Python's rich third-party libraries to reduce the operational load of knowledge-based systems.

3 Acquisition and processing of new knowledge and data

3.1 Webpage knowledge and data

This study uses www.sinofarm.net, a.nongjitong.com, www.nongji360.com, and https://www.nongji1688.com/ as the target websites. These websites are specialized agricultural machinery industry service websites in China that provide comprehensive information, such as agricultural machinery industrial information and detailed product introductions. The initial URLs of these target websites and other URLs in the webpages are in HTML format, with the same basic framework and components, and the knowledge and data acquisition methods are the same.

3.1.1 Crawler presets

The re, requests, beautifulsoup, pandas, csvkit, openpyxl, xlrd, pyinstaller, and pytesseract libraries are imported into PyCharm (an integrated development environment for the Python programming language). The third-party libraries need to be installed separately. The functions of the libraries are shown in Table 1.

Table 1 Libraries and functions used in this paper

Library	Function
re (standard library)	Match strings and generate regular expressions
requests (third-party library)	Send requests
beautifulsoup (third-party library)	Operate HTML data and provide Python-style functions to process navigation, search, etc.
pandas (third-party library)	Perform data analysis, cleaning, and data preparation
csvkit (third-party library)	Manipulate CSV tabular data
openpyxl (third-party library)	Read or edit XLSX format files
xlrd (third-party library)	Read or edit XLS format files
pyinstaller (third-party library)	Generate standalone executable files from Python script
pytesseract (third-party library)	Provide a Pythonic application programming interface (API) for an executable file

During the running of a crawler, all URLs need to be checked, including by viewing the source code of the webpage and checking whether the crawl request is successful[25]. If the crawler determines that it cannot view the source code of a webpage or that the crawl request to access the webpage is denied, the crawler cannot connect to the server, which may interrupt the crawler's operation process or its ability to obtain the target information from the URL. Therefore, when requesting access to a webpage, to avoid errors, the user–agent information from the target website is initially requested through the HyperText Transfer Protocol (HTTP), and the target website is informed of the identity of the crawler. Python's exception handling mechanism is then used to avoid a crawler crash caused by the rejection of the access request to ensure that the crawler can successfully complete the remainder of the crawling task. In addition, since "\" in the URL has an escape meaning in the Python string, to ensure that a request can be sent to the URL, in this study, "r" is added before the URL character in the code to ensure that the URL retains the meaning of the original value of the character.

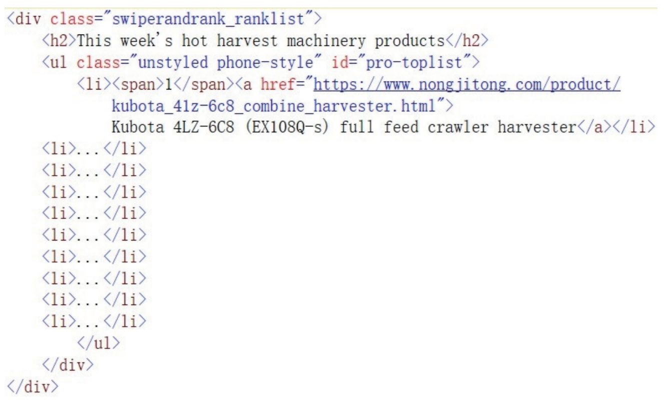
Webpage information is composed of binary 0 s and 1 s. Webpages use encoding processes to convert binary information into text. If the decoding method used by the crawler differs from that used for the webpage, then the crawler cannot obtain the correct text information, and the target information obtained by the crawler may be in the form of garbled or invalid characters. The encoding form of a webpage can be obtained from the <meta> tag in the source code, and the crawler can set its own decoding method according to the encoding form of the webpage, thus avoiding the phenomenon of unreadable information during operation.

3.1.2 Analysis and acquisition of webpage knowledge and data from the target website

Taking the crawling of knowledge and data from www.sinofarm.net as an example, https://www.nongjitong.com/product/7881.html is selected as the starting URL based on the knowledge to be updated in the combine harvester knowledge-based system. This webpage displays the parameter and product description information of the combine harvesters available on the market; therefore, this webpage is unstructured. A structured webpage is composed of data logically expressed and constructed in a two-dimensional (2D) table structure. Unlike structured webpages, unstructured webpages include not only 2D tables but also information in all formats, such as office documents, pictures, videos, and audio. To accurately collect the target information on unstructured webpages, their characteristics are identified by viewing their source codes. Common web page source code tags and their functions are shown in Table 2. Part of the source code for these webpages is shown in Figure 2.

Table 2 Common web page source code tags and their functions

Category	Tag name	Function
Structural Tags	<html>	Defines root element of HTML document containing all other elements
	<head>	Contains metadata for browsers/search engines (charset, viewport, title)
	<body>	Main content container for visible elements (text, images, links)
	<title>	Specifies browser tab/window title (text-only)
Text & Formatting	<h1>-<h6>	Header tags (h1=largest, h6=smallest) with bold and block-level display
	<p>	Defines paragraphs with automatic spacing between blocks
	<br>	Inserts single line break without extra spacing
List Tags	<li>	List item container for both ordered/unordered lists
	<ul>	Unordered list using bullet points, common for navigation
	<ol>	Ordered list with numerical sequencing (supports type attributes)
Media & Links	<img>	Embeds images (src=path, alt=alternative text, width/height=dimensions)
	<a>	Creates hyperlinks (href=URL, target=opening behavior)
	<audio>/<video>	Embeds multimedia with playback controls (src=source, controls=UI)
Table Tags	<table>	Defines tabular data structure with rows/cells
	<tr>	Table row container
	<td>	Standard table data cell
	<th>	Table header cell (bold & centered by default)
	<thead>/<tbody>/<tfoot>	Semantic table sections: header, body, footer
Form Tags	<form>	Creates interactive form for data submission (action=server endpoint)
	<input>	Generates form controls based on type (text/password/radio/etc.)
	<select>	Dropdown menu containing <option> elements
Layout & Semantics	<div>	Block-level container for layout organization (CSS integration)
	<span>	Inline container for text/element grouping
	<header>/<footer>	Semantic tags for page header/footer areas
	<nav>	Semantic navigation section
	<article>	Semantic tag for self-contained content blocks
Meta & Resources	<meta>	Configures document metadata (charset, viewport, SEO)
	<link>	Links external resources



```
<div class="swiperandrank_ranklist">
    <h2>This week's hot harvest machinery products</h2>
    <ul class="unstyled phone-style" id="pro-toplist">
        <li><span>1</span><a href="https://www.nongjitong.com/product/
            kubota_41z-6c8_combine_harvester.html">
            Kubota 4LZ-6C8 (EX108Q-s) full feed crawler harvester</a></li>
    <li>...</li>
    <li>...</li>
    <li>...</li>
    <li>...</li>
    <li>...</li>
    <li>...</li>
    <li>...</li>
    <li>...</li>
    <li>...</li>
        </ul>
    </div>
</div>
```

Figure 2 A part of the source code for the initial URL webpage

An analysis of the initial URL shows that the objective of this webpage is to obtain the URL of the specific combine harvester machinery to be crawled. According to the partial source code of the webpage in Figure 2, the target information is in the tag <a> of the tag <div>, and the URL pointed to by the tag </a> is included in the href attribute. However, from the overall source code of the webpage, multiple <div>s exist, and the position is uncertain, whereas tag <h2> adjacent to tag <div> is unique. Therefore, to accurately extract target information from the webpage source code, it is necessary to start with tag <h2> and locate tag <div> according to tag <h2>.

The BeautifulSoup library can organize and format complex webpage information by locating the HTML tags on the webpage and present extensible markup language (XML) structural information with Python objects. The BeautifulSoup library is used to treat the tag's name and attributes as key information with which to filter the HTML page to find the tag <h2> position on the initial webpage.

A navigation tree is used to find tags based on their positions on the webpage and to map part of the source code for the initial webpage. The tree structure is as follows:

```
html
 -div
       -h2
       -ul
              -li
                     -span
                     -a
              -... omit other li tag lines...
```

The navigation tree structure shows that tag <div> is tag <h2>'s parent tag; thus, when the position of tag <h2> is known, the parent method of the BeautifulSoup library can be used to determine the position of tag <div>. Under tag <div>, tag <a> can be found, and the href attribute can be crawled to locate the target information.

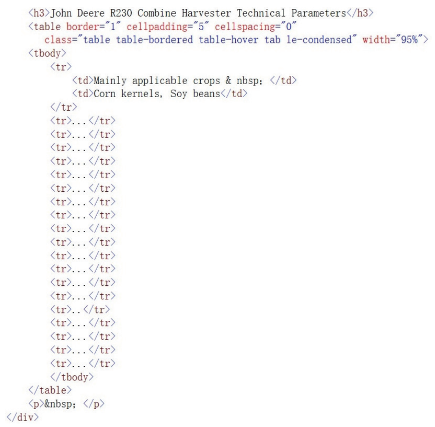
Each element in the URL queue needs to be unique to prevent the crawler from repeatedly crawling. An analysis of the URL queue shows that each URL is an unstructured webpage, and the overall layouts of the pages are similar, with the basic parameters, product introductions, and technical parameters arranged on the webpage. Taking the John Deere R230 combine harvester product webpage as an example, from the perspective of the overall source code of the webpage, relevant information about the product, including pictures, text, videos, and 2D tables, all exist in the tag <section>, with the id attribute being "section 1". The technical parameters of the John Deere R230 combine harvester obtained by the crawler are in the form of a 2D table, and some parameters are listed in Table 3. Taking the technical parameters as target information, a part of the source code describing the 2D table is shown in Figure 3.

This part of the source code of the webpage shows that the target information is in tag <tr> under tag <tbody>. However, judging from the majority of URLs in the URL queue, there may be more than one 2D table; i.e., tag <tbody> may appear multiple times in the overall source code of a webpage. Multisample webpage statistics show that 2D table information describing technical parameters is usually stored under the first-occurring tag <tbody> in the webpage source code.

Tag <tbody> consists of multiple <tr> tags, tag <tr> represents the row data in the 2D table, and tag <td> under tag <tr> describes the column data of the row. Therefore, after all <tr> tags are located, the column data of each row are obtained, the column data are combined to obtain a single data row, and the data rows are sequentially arranged to obtain the complete 2D table information.

Table 3 2D table of the technical parameters for a John Deere R230 combine harvester

Condition	Values of condition
Mainly applicable crops	Corn kernels, Soy beans
Rows	6
Applicable row spacing range/m	0.65
Working width/m	3.97
Engine model	JD6068
Engine power/HP·kW^{-1}	185/136
Gap/m	0.41
Threshing and separating type	Rasp bar+ Rod-tooth
Threshing and separating drum length× outer diameter/m	3.27×0.48
Threshing and separating area/m^2	4.06
Cleaning device	Bladed centrifugal fans
Cleaning area/m^2	4.29
Grain tank volume/m^3	5.5
Dimensions (L×W×H), m	10.05×4.23×3.84 (with corn stalk cutter)
Weight/kg	9660 (with corn stalk cutter)



```
    <h3>John Deere R230 Combine Harvester Technical Parameters</h3>
    <table border="1" cellpadding="5" cellspacing="0"
      class="table table-bordered table-hover tab le-condensed" width="95%">
    <tbody>
        <tr>
            <td>Mainly applicable crops & nbsp; </td>
            <td>Corn kernels, Soy beans</td>
        </tr>
        <tr>...</tr>
        <tr>...</tr>
        <tr>...</tr>
        <tr>...</tr>
        <tr>...</tr>
        <tr>...</tr>
        <tr>...</tr>
        <tr>...</tr>
        <tr>...</tr>
        <tr>...</tr>
        <tr>...</tr>
        <tr>...</tr>
        <tr>...</tr>
        <tr>...</tr>
        <tr>..</tr>
        <tr>...</tr>
        <tr>...</tr>
        <tr>...</tr>
        <tr>...</tr>
        </tbody>
    </table>
    <p>&nbsp; </p>
</div>
```

Figure 3 A part of the source code for the technical parameters of a John Deere R230 combine harvester

3.1.3 Webpage knowledge and data storage

The knowledge and data obtained by the crawler have HTML tags and cannot be used directly. After the content in tag <td> of the John Deere R230 combine harvester product webpage is acquired, its text information is found to contain characters "<td>" and "</td>"; in this study, regular expressions are used mainly to process knowledge and data. The symbols commonly used for these regular expressions are listed in Table 4.

Table 4 Common regular expression symbols

Symbol	Meaning
\s	Match whitespace character
*	Match the preceding character, subexpression, or characters in parentheses for 0 or multiple times
[]	Define the range of characters to match
[^]	Match any character not inside the square brackets
\	Convert characters with special meaning to their literal form
.	Match any single character
?	Indicate quantity, i.e., 0 or 1

The re.sub() function is a string replacement function in the *re* standard library and is defined below.

The tags on both sides of the text are represented by regular expressions, and the re.sub() function replaces the string matched by the regular expression with a null value; after deleting characters "<td>" and "</td>", only valid text is obtained. The specific expression is as follows:

re.sub(r'\s*\<[^<>]*\>\s*', '', td. text)

where, in the regular expression, \s* represents n spaces, \< represents character "<", [^<>]* represents characters other than < and >, \> represents character ">", and td.text represents the original string to be corrected.

After the knowledge and data are processed, they are downloaded and temporarily stored. A CSV file is a common file format for storing tabular data, which are stored in plain text. The open() function is used to read and write local files. In addition, the open() function is used to create a CSV file and write the processed knowledge and data into the file. The source code of the webpage shows that the encoding of the webpage is "utf-8"; this encoding uses a byte as the unit and has no endianness. When reading a bill of material (BOM) file and processing it as the file content, the file cannot be compiled properly, and there may be garbled characters. When writing a CSV file, it is impossible to determine whether the webpage has a BOM file; thus, encoding='utf-8-sig' is set to separately process the BOM file from the text content, the function of which is expressed as follows:

open(r"C:\….csv", 'w', newline='', encoding= 'utf-8-sig')

where 'w' is the abbreviation for mode='w', indicating that a file is opened for writing only. If the file does not exist locally, then a new file is automatically created, and if the file already exists locally, then its contents are overwritten; the use of "newline='' can avoid extra blank lines.

After the path for knowledge and data download and the file corresponding to the path are set, the processed data rows are written to the file one by one using the *for* loop.

3.2 User-uploaded files

The combine harvester knowledge-based system is divided into three modules according to the functional structure: "knowledge browsing and query", "knowledge storage and management", and "knowledge matching and model call". The "knowledge storage and management" module is divided into threshing, cleaning, header, delivery, power, drive, auxiliary, and standard modules according to the combine harvester functions. On the basis of system knowledge and data distribution, the content and location of update knowledge and data are set on the human–computer interaction (HCI) interface for each module.

Users customize the knowledge and data under the module based on its content. The types of files uploaded by users include CSV, XLSX, XLS, JPG, PDF, TXT, and so on, which are file types stored in the electronic warehouse for the combine harvester knowledge repository system.

As long as the files conform to the knowledge and data formats of knowledge bases, each file can be uploaded to the knowledge-based system for later use in the local area network. For the relevant data table files, based on the knowledge-based system data storage format, the format of the data table file uploaded by the user must conform to the established defined storage form. The encoding of the TXT file must be correct; otherwise, the content of the TXT file may be garbled when it is read and called.

Knowledge-based systems provide users with a file upload channel, prompt users to open the file, and allow users to specify the file to open. Such a system uses file extension as a tool for identifying file types and setting up file filters, and users can select only files within the specified file types on the user terminal and confirm the upload. When users continue to open the file upload channel and upload files, the system selects the last uploaded file as the latest file and applies it to the knowledge-based system.

4 Storage and retrieval of knowledge and data

4.1 Knowledge and data extraction and data table creation

For pictures and documents, after the acquisition of new knowledge and data, the files can be directly placed at the default location of the electronic warehouse. The program control can extract the file content from the specified location and display it on the HCI interface. However, the data tables cannot be directly stored as primary database files (.mdf) or log database files (.ldf), and the program cannot directly read the data table information from these files. Therefore, it is necessary to store the data table in the SQL Server database, and then the primary database files (.mdf) and log database files (.ldf) generated with the new database are stored in the electronic warehouse. The primary database files contain the startup information of the database and point to other data files in the database. The log database files save the log information used to restore the database[26].

The temporary storage form of the data tables downloaded from the webpage is the CSV, and the types of data tables that can be uploaded by users are the XLSX, XLS, and CSV. For these three types of files, this study uses the method of creating new data tables to store them in the SQL Server database. To create a data table, the column names and data types must first be set using the "create table" statement in SQL to create the data table structure. The syntax structure is as follows:

Create table <new data table name> (<column name 1> <data type 1>[null/not null],<column name 2> <data type 2>[null/not null], …)

The null and nonnull keywords determine whether null values can appear in this column, and the default keyword is null.

The content after reading is saved as a TXT file, and the content of the first row of the TXT file is independently extracted as the column name of the newly created data table. To enable the extracted content to be directly applied to the SQL language to create the structure of the newly created data table, the content must undergo the extraction process. Each column name is correlated with its corresponding data type so that they form a complete string, and the character strings are separated by the symbol ",". The processed string list can be directly integrated with the "create table" statement to create a data table.

4.2 Storage or update of new knowledge and data

After the data table structure is created, the "insert" statement in the SQL is used to fill the table structure with the data. The syntax structure is as follows:

insert [into] <new data table name> (<column name list>) values (<data value 1>, <data value 2>…)

In this syntax structure, although the "into" keyword has no real meaning, it can enhance the readability of the statement[27]. The number of values after the "value" keyword must exactly match the number of columns in the column name list; the values must have a one-to-one correspondence with the order of the column names, and the values must match or be converted to the data type corresponding to the column. Therefore, when setting the data type of a column using the "create table" statement, all possible data types should be considered.

When the data are input into the data table structure, the column names are separated with "," and concatenated to form a column name string. This string can be used as the column name list in the "insert" statement. When the data are input into the table, each "insert" statement can fill only one row of data. Therefore, this study uses a loop structure with a fixed time to repeatedly execute the data table filling command. The syntax structure is as follows:

```
For <loop control variable>=<initial value> To <final value> [Step <step size>]
    [loop body]
    Next [loop control variable]
```

The remaining content in the TXT file, except for the first row, consists of data rows, which need to be input into the newly created data table structure row by row. Therefore, in the loop structure, the initial value is set to the second data row, and the final value is set to the last data row. When the value of the loop control variable is within the range defined by the initial and final values, the loop body of the "insert" statement is executed. The Step clause is omitted, and the default step size is 1; that is, one row of data is input at a time.

The newly acquired picture and document knowledge and data are directly stored in the electronic warehouse, but the specific storage locations for different types of knowledge and data are different. The webpage-based picture and document knowledge and data use webpage tags to determine the knowledge type, and the open() function is used for storage. For the knowledge and data from user-uploaded files, the GetExtension() function in the vb.net is used to obtain the user uploaded file extensions to determine the file type of the knowledge, and copying is used to store the knowledge and data in the electronic warehouse.

According to the mapped webpage location of the electronic warehouse, the primary database files (.mdf) and log database files (.ldf) generated in the database are stored in the path "Z:\0401-2 Electronic Warehouse\Picture Library\ knowledge-based Data"; the picture files are stored in the path "Z:\0401-2 Electronic Warehouse\Picture Library"; the TXT files are stored in the path "Z:\0401-2 Electronic Warehouse\Picture Library\Knowledge Files"; and the PDF files are stored in the electronic warehouse only and are used to describe standards or patents. In knowledge-based systems, users can link to the files or folders of the standards and patents related to combine harvesters to view the detailed knowledge content; in addition to the data tables, the knowledge of standards and patents has its own unique storage location, "Z:\0401-2 Electronic Warehouse\Picture Library\Knowledge Files\Standards (Patent)".

When storing the knowledge and data of the picture and TXT files acquired on the webpage, the files are named according to appropriate strings of characters or to the file names of the files uploaded by users (system default). Before storing files, it is necessary to determine whether a file with the same name in the electronic warehouse exists. If there is no file with the same name, then the system directly stores the file in the target path; if there is already a file with the same name, then a prompt box pops up and provides the user with two choices: to directly overwrite the original file with the same name or to rename the file. The overwriting of the original file with the same name refers to the deletion of the existing file with the same name in the electronic warehouse and replacing it with a new file. In this case, the amount of data generated by the system at runtime remains the same, and the path to the file when the knowledge is called remains unchanged; however, the content of the knowledge changes. After the newly uploaded file is renamed, it is stored, and the old files are retained in the electronic warehouse. At this time, when the system is running, the default path of file mapping automatically changes, and the file storage address changes by calling a new path. The storage process of the newly acquired picture and TXT files is shown in Figure 4.

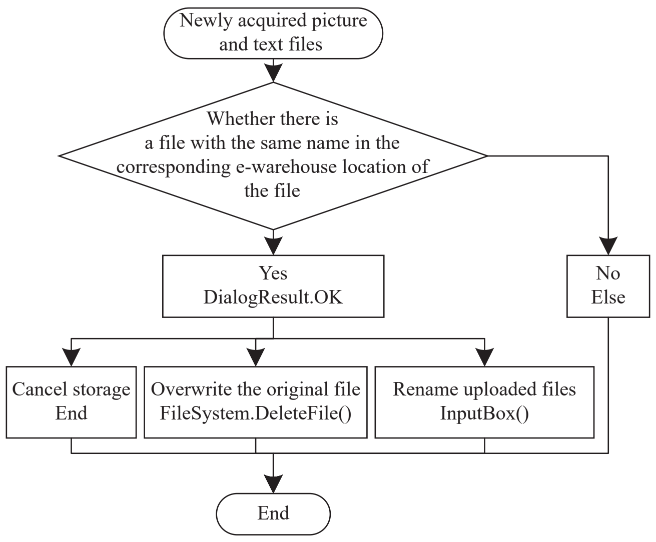

Newly acquired picture and text files
Whether there is a file with the same name in the corresponding e-warehouse location of the file
Yes DialogResult.OK
No Else
Cancel storage End
Overwrite the original file FileSystem.DeleteFile()
Rename uploaded files InputBox()
End

Figure 4 Storage process for newly acquired picture and TXT files

The naming method for PDF files involves a combination of "standard/patent code" and "standard/patent name". Before storing a newly obtained PDF file, it is necessary to determine whether the file name is standardized. The standard/patent code consists of a combination of letters, numbers, and symbols '.' and '-'. The standard/patent name is in Chinese characters. The file naming convention is expressed as a regular expression, the expression of the standard/patent code is [A-Za-z0-9\.-s], and that of the standard/patent name is [\u4e00-\u9fa5], where "\u4e00" and "\u9fa5" are unicode encodings and the start and end values of Chinese encoding, respectively[28]. The standardization of the file naming method is determined by calling the FunCheckRegular() function (used to verify that a string conforms to a specific pattern via a regular expression) in vb.net, and the statement of the determination function is as follows:

FunCheckRegular (regular expression, PDF file name).

The returned result of the FunCheckRegular() function is a Boolean value. If the file name meets the naming convention, then the function return value is "True"; otherwise, the return value is "False".

The PDF file that adheres to the naming convention is compared with the existing files in the electronic warehouse to determine whether a file with the same name in the electronic warehouse exists. When a file with the same name exists in the electronic warehouse, it means that the standard/patent document already exists in the electronic warehouse, as is the situation in this case. Users can choose to cancel the upload or overwrite the original file, and after overwriting the file, the file path called by the system remains unchanged. If the file is determined to be a new file, then the system directly stores the new file in the electronic warehouse. The storage process of the newly acquired PDF files is shown in Figure 5.

Upon adding a new document, the corresponding standard/patent information is synchronized and added to the standard/patent information database. The database information includes "Standard/Patent Code," "Standard/Patent Name," "Drafting Unit," and "Implementation Date." The first two pieces of information can be extracted directly from the new file name, whereas the last two pieces of information need to be extracted by means of text recognition and query.

PDF files can be one of two types. One type is an editable PDF file, and PDF files of this type are usually exported as WORD or

EXCEL files. The other type is a noneditable PDF file, and PDF files of this type are usually generated by the scanner or by the composition of the picture. The convert() function in Python can be used to convert PDF files to fixed layout DOCX files since such files have a fast conversion speed and can maximally restore the content of PDF files after file format conversion.

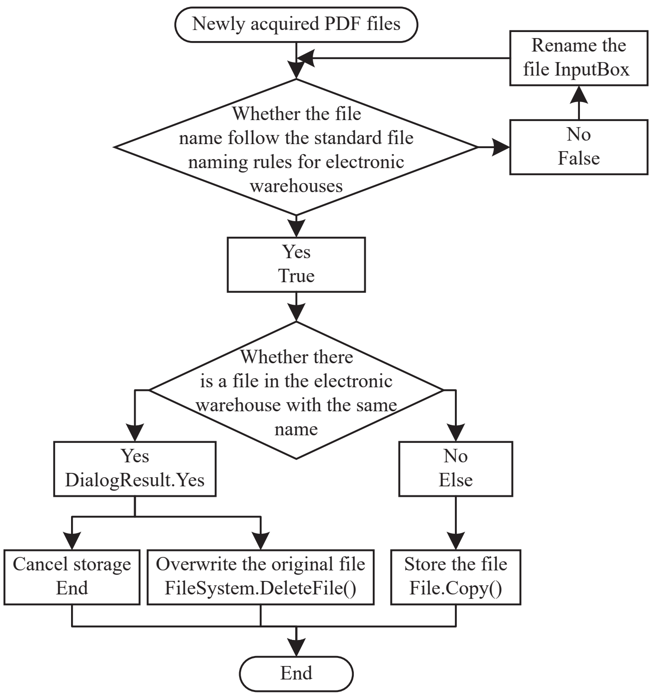

Newly acquired PDF files
Rename the file InputBox
Whether the file name follow the standard file naming rules for electronic warehouses
No False
Yes True
Whether there is a file in the electronic warehouse with the same name
Yes DialogResult.Yes
No Else
Cancel storage End
Overwrite the original file FileSystem.DeleteFile()
Store the file File.Copy()
End

Figure 5 Storage process for newly acquired PDF files

For editable PDF file format conversion, the text in the DOCX file is displayed in a separate frame, and this type of text can be directly used by the WORD query function for information retrieval.

For noneditable PDF file format conversion, the DOCX file consists of pictures. After the picture is extracted from the WORD file, optical character recognition (OCR) technology is used for text recognition to obtain the text of the picture.

A DOCX file is essentially a compressed file containing XML, picture, and knowledge and data of different formats; it can be decompressed to obtain folder_rels, docProps, words, and files [Content_Types].xml. The picture files are stored in the word folder in the media directory, and all the pictures in this directory are extracted and used as objects for text recognition.

Tesseract-OCR is an open-source text recognition engine that supports multiple languages (the default language is English) and can be applied to the script through the API. Because standard and patent files are official documents, training for the font samples in specific cases is not needed, and installing the official language packages chi_sim and chi_sim_vert in Tesseract-OCR can meet text recognition needs. Tesseract-OCR has better recognition results when the number of dots per inch (DIP) of the picture is ≥300. Therefore, the use of the resize() function (for resizing images) in Python to enlarge the picture before recognizing the text can effectively improve the text recognition rate of Tesseract-OCR.

After the text of the picture is obtained, the drafting unit and implementation date information of the standard or patent can be queried. The re.compile() function in Python (used to compile regular expression strings into reusable Pattern objects) is used to convert regular expressions into objects, and the function consists of two parameters: the regular expression and the match mode. Regular expressions can pinpoint the drafting unit and implementation date of a document, and the match mode can be set to match the format of the string. The standard document drafting unit details are located after the string "the standard drafting unit:" and before the string "main drafter of this standard", and the regular expression is as follows:

r"(the standard drafting unit:)" "(. *?)" "(the main drafter of this standard)"

The details of the implementation date of the standard document are located after the string "published" and before the string "implement mention".

In the re.compile() function in match mode, the selected re.I, re.M, and re.S require that the character case, multiline match, and any character, including line breaks, be ignored, and these three match modes can increase the level of comprehensiveness of character matching.

4.3 Call for new knowledge and data

After the structure creation and content input of the new data table are completed, the corresponding old data table is deleted based on the content of the data table, and the new data table is named according to the old data table. After the system is connected to the SQL Server database, the name of the data table to be called does not need to be changed, and the content of the newly created data table is mapped to the interactive interface.

The PictureBox control in vb.net can display picture files on the interactive interface. To enable pictures of different sizes to be completely displayed on the interactive interface, the SizeMode property of this control is set to PictureBoxSizeMode. StretchPicture can automatically adjust the picture size so that the picture always fits the size of the control. The picture property of the control is used to obtain or set the picture to be displayed in the control. After the picture file is updated, the paths of the newly stored file and original existing files are not necessarily the same. For the control to display the new files on the interactive interface in a timely manner as the files stored in the electronic warehouse are updated, the path of the picture file to be displayed is stored in the data table as the default value of the picture property to set the default picture to be displayed by the control. When the path of the new file is different from that of the original file, the former is synchronously updated to the data table using the update statement in SQL, and the original path is replaced by this new path as the default path of the system. During system operation, the picture file is called through the SQL statement to query the default path value of the picture file in the data table, and the returned result is assigned to SqlDataReader in vb.net. The system cannot directly call the query result, so the data type of the path value called by the system is String, and SqlDataReader needs to perform data type conversion before it can be used. After the Read() function in vb.net is used to read the value of the result buffer returned by the query, conversion between the two data types can be realized using the ToString() function in vb.net. String data are a type of text data that truly reflect the data stored in the data table, and only the file path under this data type can be used as the picture property value of PictureBox control. The process of calling the newly acquired picture file is shown in Figure 6.

Unlike picture files, TXT files are mapped to the interactive interface by using the RichTextBox control in vb.net. After the ReadToEnd function in vb.net is used to read the file from beginning to end, the AppendText() function in vb.net can be read to fill the read text in the control so that it can be mapped to the interactive interface. The ReadToEnd function reads the file path in the same way as it does in the picture file. After the new TXT file is stored in the electronic warehouse, its file path is synchronously updated in the data table as the value of the file path parameter of the ReadToEnd function. The process of calling the newly acquired TXT file is shown in Figure 7.

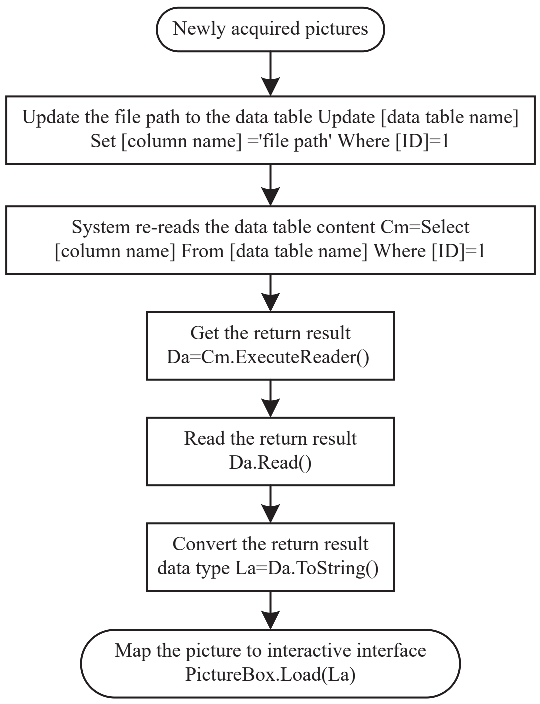

Newly acquired pictures
Update the file path to the data table Update [data table name] Set [column name] ='file path' Where [ID]=1
System re-reads the data table content Cm=Select [column name] From [data table name] Where [ID]=1
Get the return result Da=Cm.ExecuteReader()
Read the return result Da.Read()
Convert the return result data type La=Da.ToString()
Map the picture to interactive interface PictureBox.Load(La)

Figure 6 Calling process for newly acquired pictures

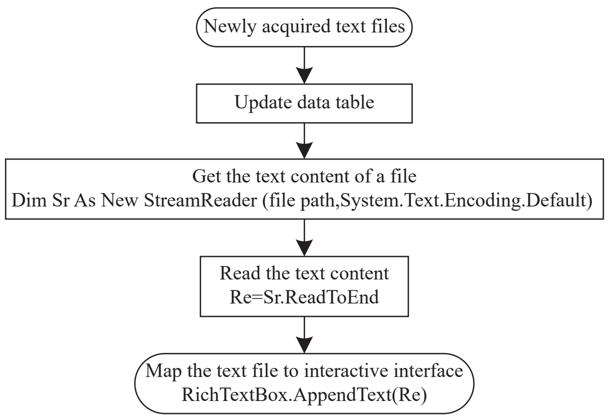

Newly acquired text files
Update data table
Get the text content of a file Dim Sr As New StreamReader (file path,System.Text.Encoding.Default)
Read the text content Re=Sr.ReadToEnd
Map the text file to interactive interface RichTextBox.AppendText(Re)

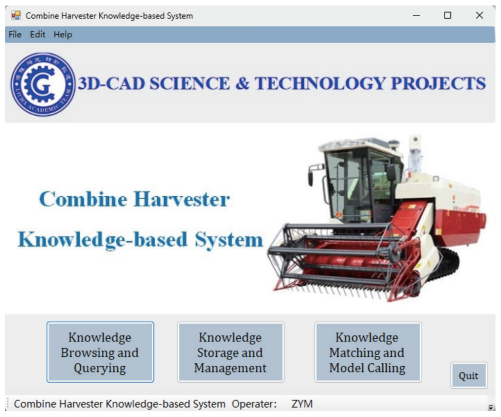
Figure 7 Calling process of newly acquired TXT files

The naming method of stored PDF files is a combination of the code column and name column in the standard/patent data table; thus, the name of the PDF file is bound to the data table. The user can determine which files to display by selecting the row data in the data table.

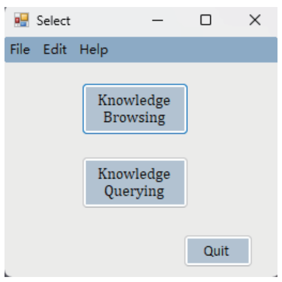
5 Technology integration and testing

5.1 Human-Computer Interaction (HCI) interface design

The automatic update (AU) function is distributed in the "knowledge browsing and query" module and the "knowledge storage and management" module under the knowledge-based system main interface. This function is located on the "knowledge browsing" main interface under the "knowledge browsing and query" module. After logging into the main interface of the knowledge-based system, "knowledge browsing" can be selected under the "knowledge browsing and querying" module, as shown in Figure 8. This function is distributed in the threshing, cleaning, header, delivery, power, drive, auxiliary, and standard part modules under the "knowledge storage and management" module. The webpage knowledge and data update and user-uploaded file update functions in the interactive interface are represented by the "webpage data update" and "local file upload" buttons, respectively. The buttons are arranged at the top of the interface.


Combine Harvester Knowledge-based System
File Edit Help
3D-CAD SCIENCE & TECHNOLOGY PROJECTS
Combine Harvester
Knowledge-based System
Knowledge Browsing and Querying
Knowledge Storage and Management
Knowledge Matching and Model Calling
Quit
Combine Harvester Knowledge-based System Operater: ZYM

a. Main interface of the combine harvester knowledge-based system



b. Knowledge browsing and query selection interface-based system

Figure 8 Location of the main interface of "Knowledge Browsing" in the system

The user specifies a node on the directory tree on the interface and selects the AU button to command the system to update knowledge and data. After the knowledge content to be updated is determined through the directory tree node, the system acquires the corresponding knowledge under the node and connects to the corresponding data table in the database to display the knowledge and data on the interface. After the user triggers the AU command, the system stores the newly acquired knowledge and data in the storage location in the electronic warehouse corresponding to the node in question.

The main interface of "knowledge browsing" covers a variety of knowledge forms, including data tables, text knowledge, picture browsing, and PDF files used to introduce standards or patents in detail. Taking this interface as an example, the knowledge and data are updated in the interface.

5.2 Example test

5.2.1 Webpage knowledge and data updating

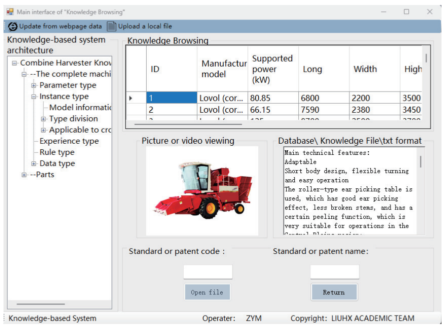
Owing to the unpredictable update frequency and quantity of the aforementioned knowledge and data source websites, to ensure the controllability, verifiability, and quantifiability of the test, the team uses the "Northeast Agricultural University National Agricultural Engineering Experimental Teaching Demonstration Center webpage," which is managed by the team, as the test subject. The testing is conducted in a regular office network environment. The test environment is simulated by posting content similar to that of the target website, and the efficiency and quality of the update of webpage knowledge and data in the knowledge-based system are analyzed, as shown in Figure 9. The latest web articles titled "Yanmar YH1180R Crawler Grain Combine Harvester" and "Lovol GM100 (4LZ-10M6) wheel grain harvester" focus on product characteristics and technical parameters to introduce combine harvester information.

Location: Home>Latest Information

▪ Yanmar YH1180R full feed harvester	2023-02-03
▪ Lovol Gushen GM100 (4LZ-10M6) longitudinal...	2023-02-03
▪ John Deere C230 Combine Harvester	2022-12-16

Figure 9 Information dynamics on the test website

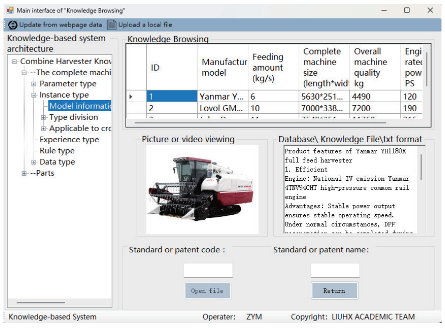
When the knowledge-based system is running, the system automatically triggers the update command and determines whether the test website is updated by detecting the title change on the website. This detection approach can find the newly added webpage articles on the test website, and a prompt box pops up asking, "Do you want to update the corresponding content in the system?", thus providing the user with an update choice. From the moment the user initiates the update until the completion of the system update, the system automatically calculates the update duration as 5.265 s. The update result is shown in Figure 10.



	ID	Manufactur model	Supported power (kW)	Long	Width	High
▸	1	Lovol (cor...	80.85	6800	2200	3500
	2	Lovol (cor...	66.15	7590	2380	3450

```
Main technical features:
Adaptable
Short body design, flexible turning
and easy operation
The roller-type ear picking table is
used, which has good ear picking
effect, less broken stems, and has a
certain peeling function, which is
very suitable for operations in the
```

a. Content on the main interface of "knowledge browsing" before the update



	ID	Manufactur model	Feeding amount (kg/s)	Complete machine size (length*wid	Overall machine quality kg	Engi ratec pow PS
▸	1	Yanmar Y...	6	5630*251...	4490	120
	2	Lovol GM...	10	7000*338...	7200	190

```
Product features of Yanmar YH1180R
full feed harvester
1. Efficient
Engine: National IV emission Yanmar
4TNV94CHT high-pressure common rail
engine
Advantages: Stable power output
ensures stable operating speed.
Under normal circumstances, DPF
```

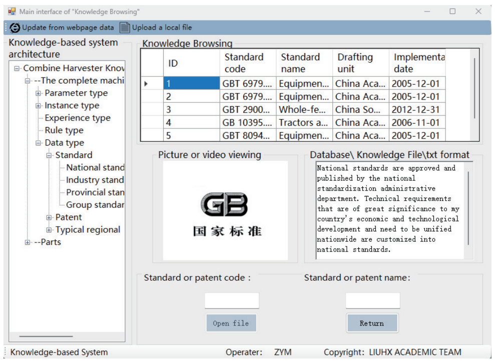
b. Content on the main interface of "knowledge browsing" after the update

Figure 10 Updates on the main interface of "Knowledge Browsing" from webpage knowledge and data

5.2.2 Knowledge and data update from user-uploaded files

Taking the "National Standards" node in the directory tree as an example, the knowledge and data under the node are updated by using the files uploaded by the user as the knowledge and data source. This node is connected to the "National Standards" data in the "K Standards and Patents" database, and the specific PDF file of the standard corresponding to each data row in the data table is stored under the path "Z:\0401-2 Electronic Warehouse\Database\Knowledge Files\Standards\National Standards". The TXT file used to describe the nodes in the interface is stored under the path "Z:\0401-2 Electronic Warehouse\Database\Knowledge Files\TXT Format\National Standards.TXT". The storage path of the picture files is "Z:\0401-2 Electronic Warehouse\Picture Library\National Standards.JPG."

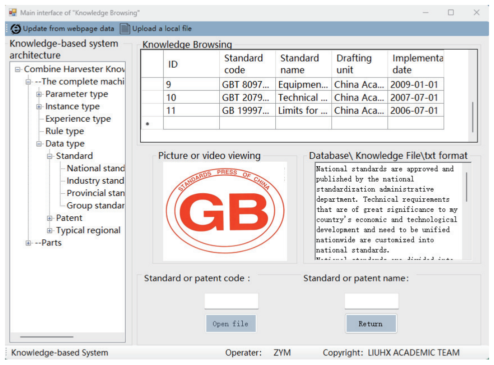
When the user triggers the "local file upload" command, a file selection box pops up; the user can select the local file to upload from the limited number of file types, and the system automatically determines the file type and uses different methods to update the content of the uploaded file on the interface, as shown in Figure 11.



	ID	Standard code	Standard name	Drafting unit	Implementa date
▸	1	GBT 6979....	Equipmen...	China Aca...	2005-12-01
	2	GBT 6979....	Equipmen...	China Aca...	2005-12-01
	3	GBT 2900...	Whole-fe...	China So...	2012-12-31
	4	GB 10395....	Tractors a...	China Aca...	2006-11-01
	5	GBT 8094...	Equipmen...	China Aca...	2005-12-01

```
National standards are approved and
published by the national
standardization administrative
department. Technical requirements
that are of great significance to my
country's economic and technological
development and need to be unified
nationwide are customized into
national standards.
```

a. Content of the main interface of "knowledge browsing" before the update



	ID	Standard code	Standard name	Drafting unit	Implementa date
	9	GBT 8097...	Equipmen...	China Aca...	2009-01-01
	10	GBT 2079...	Technical ...	China Aca...	2007-07-01
	11	GB 19997...	Limits for ...	China Aca...	2006-07-01
*					

```
National standards are approved and
published by the national
standardization administrative
department. Technical requirements
that are of great significance to my
country's economic and technological
development and need to be unified
nationwide are customized into
national standards.
```

b. Content of the main interface of "knowledge browsing" after the update

Figure 11 Updates on the main interface of "Knowledge Browsing" from user-uploaded knowledge and data

For the data table file, the system deletes the "National Standard" data table connected to this node and replaces the original data table with the newly uploaded data table. The new data table is still named after the "National Standard".

For text and picture files, the system deletes the file in the original path, names the newly uploaded file according to the old file name, and stores the newly uploaded file under this path.

For the PDF file, the "GB 19 997-2005 Limits for noise emitted by combine harvester.pdf" file is selected and added to the knowledge-based system. The system first determines that the file name conforms to the naming rule of the combination of standard codes and names and then determines that the standard file does not exist in the electronic warehouse. After storing the PDF file in the electronic warehouse, the file information is added to the "National Standards" in the data table, and the system extracts the strings representing standard codes and names in the filename, identifies the drafting unit and implementation date information in the file, and adds these four pieces of information to the data table to complete the update of the PDF file. When the user selects the directory tree node again, he or she can see that a new data row has been added to the data table, and when the "open file" command is triggered, the "GB 19 997-2005 Limits for noise emitted by combine harvester.pdf" file can be opened directly. The system records the update duration as 3.526 s, spanning from the completion of the user's document upload operation to the finalization of the knowledge base update.

6 Conclusions

1) The AU of the knowledge-based system is implemented through the hybrid programming of vb.net and Python. vb.net is used for the HCI interface to achieve communication between the user and the system, and Python is used for the data acquisition and processing of webpages, with the advantages of accurate positioning and concise codes. Hybrid programming can effectively reduce the level of complexity of program design and improve operational efficiency.

2) The knowledge-based system automatically triggers the webpage update check command after the user logs into the system, which can solve the problem of untimely system updates, thus realizing the timeliness of the data. After triggering the automatic webpage update command, the duration for crawling and updating knowledge and data from the three URLs of the test webpage to the knowledge base is 5.265 s. This capability ensures the effectiveness and practicality of the combine harvester knowledge-based system.

3) Knowledge and data updating through user-uploaded files adopts the method of users selecting and uploading files, enabling the customization of knowledge and data within the human-computer interaction framework. This approach effectively meets personalized needs and enhances user efficiency.

Acknowledgements

The authors acknowledge that this work was financially supported by the "Suqian Talent" Xiongying Program for Educational Innovation Talents Project (Grant No. SQXY202431) and the National Key Research and Development Program of China (Grant No. 2017YFD0700100).

[References]

[1] Liu H X, Li J L, Guo L F, Zhang G F. Knowledge organization and knowledge base system of combine harvester. Transactions of the CSAM, 2021; 52(2): 381–393. (in Chinese)

[2] Li Q L, Song Y Y, Yao C J, Li W B, Yue Y C. Intelligent design and optimization system for cleaning device of rice and wheat combine harvester. Transactions of the CSAM, 2021; 52(5): 92–101. (in Chinese)

[3] Liu H X, Liu Z J, Zhang G F, Zhou L L. Interactive design system of threshing device based on skeleton design. Transactions of the CSAM, 2020; 51(12): 405–416. (in Chinese)

[4] Liu H X, Wang D Y, Guo L F, Liu W W, Zhang G F. Development of advanced design technology and its application in agricultural equipment. Transactions of the CSAM, 2019; 50(7): 1–18. (in Chinese)

[5] Liu Q, Li Y, Duan H, Liu Y, Qin Z G. Knowledge graph construction techniques. Journal of Computer Research and Development, 2016; 53(3): 582–600.

[6] Sheng F, An X H, Lin H X, Zeng X Q, Hu N N, Li D, et al. A pre-evaluation method for rail transit safety based on a safety knowledge base. Journal of Tongji University (Natural Science), 2024; 52(2): 184–191. (in Chinese)

[7] Khan S, Rashid A, Rasheed R, Amirah N A. Designing a knowledge-based system (KBS) to study consumer purchase intention: the impact of digital influencers in Pakistan. Kybernetes, 2023; 52(5): 1720–1744.

[8] Li W B, Li Q L, Huang Y L, Song Y Y, Yue Y C. Construction of intelligent design platform for threshing device of combine harvester for rice and wheat. Transactions of the CSAM, 2020; 51(S2): 154–161. (in Chinese)

[9] Zhang Y A, Xin Z, Du Y F, Shao M X, Sun E X, Zhang F W. Rapid design method and software development for tractor gear box. Transactions of the CSAE, 2020; 36(21): 49–55. (in Chinese)

[10] Zhu Q, Zhao Y Z, Guo Y X, Ding Y L, Wang Q, Pan Y, et al. Distributed management method of geo-knowledge base for digital twin railway. Journal of Southwest Jiaotong University, 2023. doi: 10.3969/j.issn.0258-2724.20230389 (in Chinese)

[11] Pannu M, Kay I, Harris D. Using dark web crawler to uncover suspicious and malicious websites. In: International Conference on Applied Human Factors and Ergonomics, Florida: Springer International Publishing, 2018; pp.108–115. doi:10.1007/978-3-319-94782-2_11.

[12] Peng D, Li T, Wang Y, Chen C L P. Research on information collection method of shipping job hunting based on web crawler. In: 2018 Eighth International Conference on Information Science and Technology (ICIST), Cordoba, Granada, and Seville, Spain: IEEE, 2018; pp.57–62. doi: 10.1109/ICIST.2018.8426183.

[13] Wang Y H, Zheng L Y, Fan W. Data collection and fusion of manufacturing workshop based on standard semantic model and complex event processing underl cloud architecture. Computer Integrated Manufacturing Systems, 2019; 25(12): 3103–3115.

[14] Huazhong University of Science and Technology. CNC system industrial data collection and storage method and system. 2018; CN110858125A. (in Chinese)

[15] Yang J Y. Research on knowledge base network updating mechanism based on markov decisionprocess. Master's Dissertation. Guangzhou: Jinan University, 2020; 6. 64p. (in Chinese)

[16] Hua L Z. The design and implementation of domain knowledge base management system based on knowledge graph. Master's Dissertation. Beijing: Beijing University of Posts and Telecommunications, 2018, 3. 61p (in Chinese)

[17] Ganapati S, Ahn M, Chen Y C, Krimmer R, Pereira G V, Pliscoff C, et al. Global intelligent governance- a collaborative platform. In: DG. O2021: The 22nd Annual International Conference on Digital Government Research. 2021; pp.593–595. doi: 10.1145/3463677.3463728.

[18] Roh Y, Heo G, Whang S E. A survey on data collection for machine learning: a big data- AI integration perspective. IEEE Transactions on Knowledge and Data Engineering, 2019; 33(4): 1328–1347.

[19] Deng H P, Wu G. Discovery of topic-specific information source based on web crawler and website classification. Computer Engineering and Applications, 2016; 52(3): 59–65.

[20] Xie R R, Xu H, Zheng S W, Ma G. Web crawler-based simulation of large data grabbing method for web pages. Computer Simulation, 2021; 38(06): 439–443.

[21] Gong W W, Qi X C, Pei S L. Research and application of hybrid programming with Python and R. Computer Applications and Software, 2018; 35(1): 28–31.

[22] Liu X P, Liu H J, Zhao Y, Li G L. Time domain analysis of bridge flutter and programming based on ANSYS. Journal of Vibration and Shock, 2022; 41(13): 252–258, 293.

[23] Kochhar P S, Wijedasa D, Lo D. A large scale study of multiple programming languages and code quality. In: 23Rd International Conference on Software Analysis, Evolution, and Reengineering (SANER), Osaka, Japan: IEEE, 2016; 1: 563–573. DOI: 10.1109/SANER.2016.112

[24] Zheng C, An Y, Wang Z, Qin X, Eynard B, Bricogne M, et al. Knowledge-based engineering approach for defining robotic manufacturing system architectures. International Journal of Production Research, 2023; 61(5): 1436–1454.

[25] Liu J C, Wu B G, Chen D. Research and construction of web crawler based forest management knowledge collection system. Journal of Zhejiang A&F University, 2017; 34(4): 743–750.

[26] Choi H, Lee S, Jeong D. Forensic recovery of SQL server database: Practical approach. IEEE Access, 2021; 9: 14564–14575.

[27] Schreiner G A, Duarte D, Mello R D S. Bringing SQL databases to key-based NoSQL databases: a canonical approach. Computing, 2020; 102(1): 221–246.

[28] Wang B, Wu L, Li W J, Qiu Q J, Xie Z, Liu H, et al. A semi-automatic approach for generating geological profiles by integrating multi-source data. Ore Geology Reviews, 2021; 134: 104190.