ISSN: 2414-1895 DOI: 10.6919/ICJE.202611_12(11).0007

Automatic Parameter Extraction and Verification Method for Nuclear Power Plant DCS Control Stations

Zhaoxu Kang[a]

Liaoning Hongyanhe Nuclear Power Co., Ltd., Dalian 116300, China

[a]lovecaf@163.com

Abstract

Pre/post-download parameter consistency verification is a very import task in the operation and maintenance of DCS in nuclear power plant. The conventional manual verification approach suffers from heavy workload, long duration, and risks of missing items, data misalignment, and misjudgment. This paper proposes a method for automatic extraction of control station algorithm lists and online parameter verification. The method parses control station configuration project files to automatically filter algorithm blocks according to user-specified block names and database types, and generates standardized list files containing Dynamic Data Exchange (DDE)-compliant parameter reading expressions. Online parameters are batch-read and compared via the DDE interface before and after program download based on the same list file, guaranteeing one-to-one correspondence between the two acquisitions. Engineering application on CPR1000 nuclear power plants shows that the proposed method significantly reduces manual effort and improves the efficiency, accuracy, and reliability of parameter verification.

Keywords

Parameter Verification; Configuration File Parsing; Dynamic Data Exchange.

1. Introduction

The DCS is the core of the instrumentation and control (I&C) system of a nuclear power plant, undertaking key tasks such as reactor protection, process control and operation monitoring [1]. During long-term operation, control strategy optimization, defect modification and outage commissioning involve the modification of control logic and program downloading of control stations, and the parameter consistency verification before and after downloading is a necessary step to confirm that the downloading has not caused unexpected parameter changes. IAEA safety standards and standards such as IEC 61513 impose strict requirements on the modification control and verification of safety-important I&C systems [2-3], and parameter verification is a fundamental part of modification verification activities.But with the increasing scale of modern DCS systems (such as the CPR1000 project, which has a total of 35 control stations), manual parameter comparison is no longer sufficient to meet on-site requirements.

Therefore, this paper proposes an automatic algorithm-list extraction and online parameter verification method for control stations. The main contributions include: (1) an automatic algorithm-list extraction method for control station configuration project files ;(2) a paired online-parameter batch acquisition mechanism before and after downloading;(3) automatic item-by-item comparison with difference highlighting;(4) a complete engineering implementation verified in the control station maintenance of CPR1000 plants.

ISSN: 2414-1895 DOI: 10.6919/ICJE.202611_12(11).0007

2. Scheme Design

2.1 Overall Framework and Problem Formalization

As shown in Fig. 1,the method consists of three successive steps : ① list extraction-parsing the configuration project files, automatically filtering qualified algorithm blocks and extracting associated parameters to generate a standardized list file; ② paired acquisition-batch-reading online parameters through the DDE interface before and after downloading, and writing them into the pre-download and post-download data areas respectively; ③ automatic comparison-comparing item by item according to the list correspondence and identifying differences. All three steps use the standardized list file as the only data carrier, so that the parameter identification remains consistent throughout the verification process, mechanically guaranteeing the one-to-one correspondence of the two acquisitions.

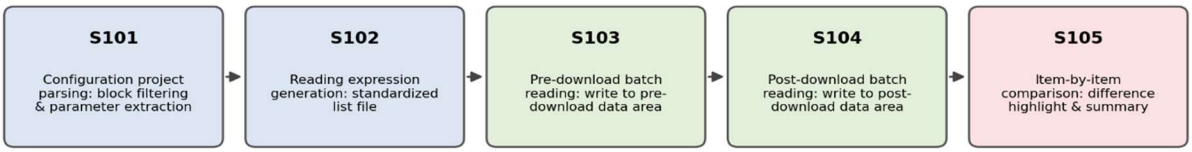


Fig. 1 Overall steps of the automatic algorithm-list extraction and online parameter verification method for control stations

2.2 Configuration Project Parsing and Algorithm List Extraction

The control station configuration project files are organized as "algorithm program + configuration data file": each control station has a separate folder named by its station number, which contains both the control algorithm program written by CFC(Continuous Function Chart)[4] and the XML-format configuration data file corresponding to the algorithm program one to one. The configuration data file contains the complete content of the algorithm program, including the algorithm program name, algorithm block definitions, parameter names, parameter paths and database types. The control station configuration algorithm program and the function block connection relationship are shown in Fig. 3.

Fig. 2 shows that the list extraction process is divided into three steps: ① parsing XML-format configuration data file to identify the algorithm program name; ② conditionally matching the algorithm block definitions according to the user-specified algorithm block names and database types, and extracting the qualified algorithm block names, control station information and associated parameter names; ③ associating the algorithm block names, owning control station information and parameter names to obtain the algorithm blocks to be verified and their associated parameter information. The interface of the list extraction tool is shown in Fig. 4.

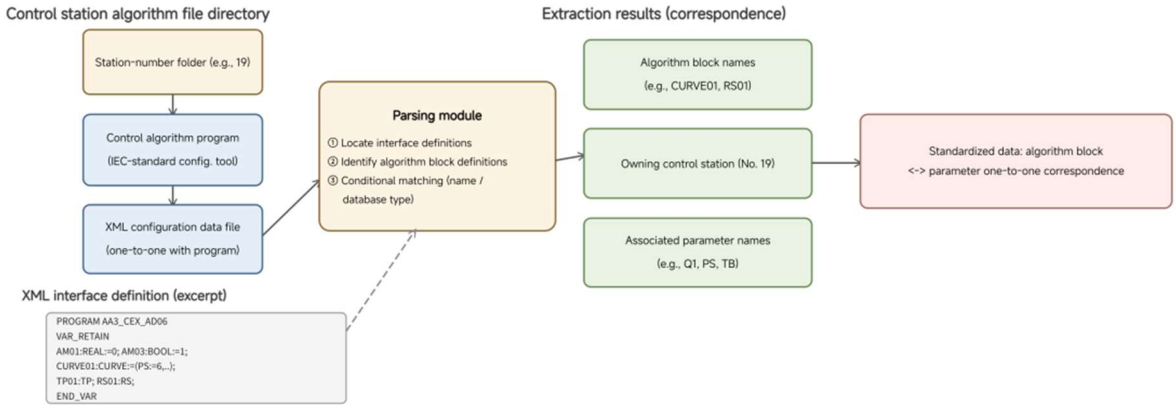


Fig. 2 Data processing flow of automatic algorithm list extraction

ISSN: 2414-1895 DOI: 10.6919/ICJE.202611_12(11).0007

Taking the project path D:\HOLLiAS_MACS\ENG\user\HYH_AA3\19 as an example, the extraction tool parses the configuration project files under this path, identifies the algorithm program name (e.g., AA3_CEX_AD06), extracts the algorithm block names, the control station number (19) and the parameter names (e.g., RS01.Q1), and establishes their correspondence. A typical fragment of the interface definition part of the configuration data file is as follows:

<interface><![CDATA[PROGRAM AA3_CEX_AD06
VAR RETAIN
AM01:REAL:=0; AM02:REAL:=0; AM03:BOOL:=1;
CURVE01:CURVE:=(PS:=6,TB:=0,…); TP01:TP; SFT01:SFT:=(RT:=2); RS01:RS;
END_VAR]]></interface>

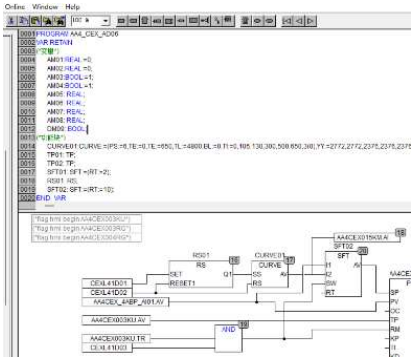


Fig. 3 Control station configuration algorithm program and function block connections

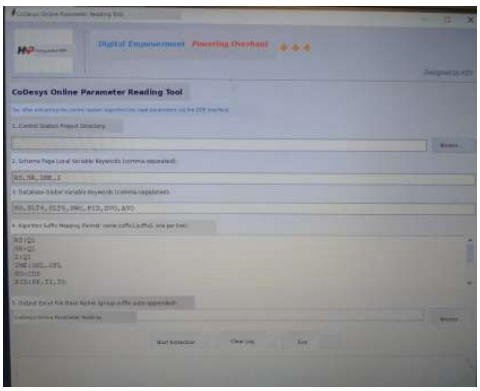


Fig. 4 Interface of the algorithm list extraction tool

2.3 Parameter Reading Expression Generation and the Standardized List

After list extraction, the algorithm block and parameter information needs to be converted into parameter reading expressions directly callable by the DDE interface[5] format "application|topic!item", whose general form is

E = oDeSys|′ < project file path > ′! < algorithm program name >.< algorithm block name >.< parameter name > Taking the parameter Q1 of algorithm block RS01 in algorithm program AA3_CEX_AD06 of station No. 19 as an example, the generated parameter reading expression is:

=CoDeSys|'D:\HOLLiAS_MACS\ENG\user\HYH_AA3\19\HYH_AA3_19.pro'!AA3_CEX_AD06.RS01.Q1

An independent Excel list file named by the control station number, with the naming rule "prefix_algorithm program project name_station number.xlsx". The generated standardized list file is shown in Fig. 5, and the composition of the parameter reading expression and the structure of the list file are shown in Fig. 6.

ISSN: 2414-1895 DOI: 10.6919/ICJE.202611_12(11).0007

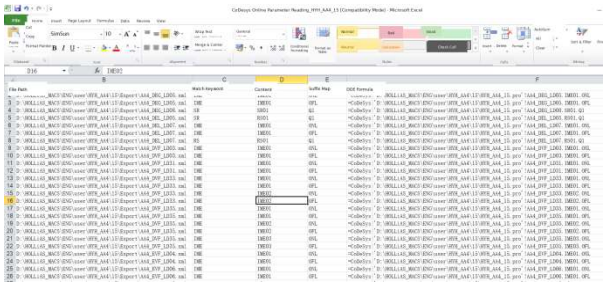
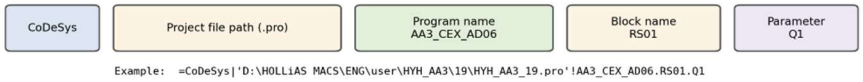


Fig. 5 Generation result of the standardized algorithm list file

(a) Composition of the parameter reading expression (DDE interface format)



Example: =CoDeSys|'D:\HOLLiAS MACS\ENG\user\HYH_AA3\19\HYH_AA3_19.pro'!AA3_CEX_AD06.RS01.Q1

(b) Structure of the standardized list file (Excel, named by station number, grouped by control station)

Block name	Control station	Database type	Parameter	Reading expression	Pre-download	Post-download	Result
CURVE01	19	Input variable library	PS	=CoDeSys|'...pro'!AD06.CURVE01.PS	6	6	Consistent
RS01	19	Input variable library	Q1	=CoDeSys|'...pro'!AD06.RS01.Q1	0	1	Inconsistent

Fig. 6 Composition of the parameter reading expression (a) and structure of the standardized list file (b)

2.4 Paired Acquisition before and after Downloading and Automatic Comparison

As shown in Fig. 7,Online parameter acquisition is performed twice based on the same list file. Before downloading, batch parameter reading requests are initiated to obtain the real-time current values of the parameters through DDE interface, and the values are written into the pre-download data area at the positions corresponding to the parameter identifications in the list file. After the control station program is downloaded and resumes operation, the same list file is loaded, batch reading requests are initiated again, and the obtained real-time current values are written into the post-download data area.The reading time of each acquisition is recorded for traceability.

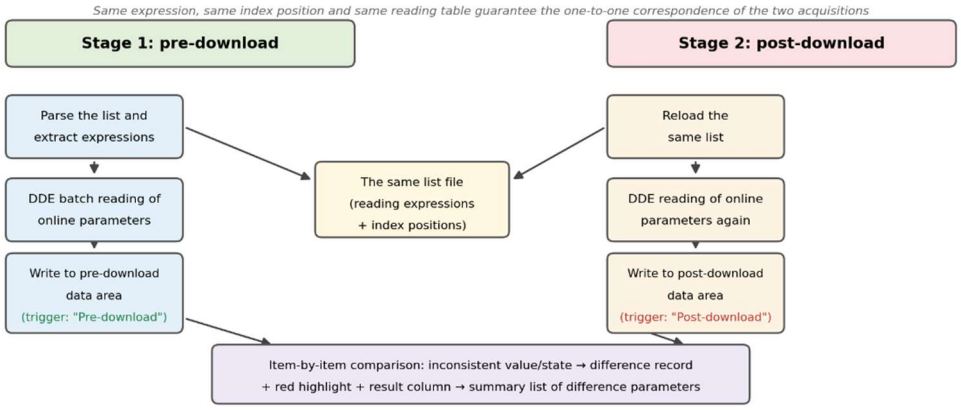


Fig. 7 Execution flow of paired acquisition and automatic comparison before and after downloading

In the automatic comparison step, the real-time current values of the same record in the pre-download and post-download data areas are extracted and compared item by item: if the value is inconsistent, the record is judged as a difference record, and the corresponding cells in the list file are highlighted (red fill) with the result column outputting "Consistent/Inconsistent". The difference judgment and identification rules are shown in Table 1.

ISSN: 2414-1895 DOI: 10.6919/ICJE.202611_12(11).0007

Table 1. Parameter difference judgment and identification rules

Comparison case	Judgment	Identification and output
Pre-download value consistent with post-download value	Consistent	The result column outputs "Consistent"; no identification
Pre-download value inconsistent with post-download value in value	Difference record	Corresponding cells highlighted in red; the result column outputs "Inconsistent"
DDE reading failure or abnormal return value	Difference record	Manual check

The online parameter comparison tool is implemented in Excel VBA, and its interface and comparison results are shown in Fig. 8. After loading the list file, the user clicks the "Pre-download" button to complete the batch acquisitions; After clicking the "Post-download" button, the tool automatically complete the batch acquisitions again and completes item-by-item comparison and highlighting. It should be noted that the software itself does not participate in the downloading of the control station program and is only responsible for parameter reading and comparison; the downloading operation is still performed by on-site personnel according to the prescribed process.

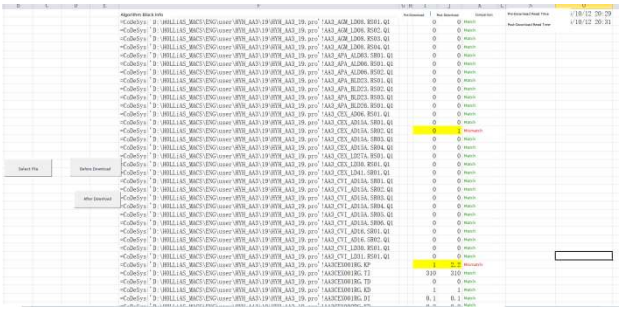


Fig. 8 Interface and comparison results of the online parameter comparison tool

3. Analysis and Verification

3.1 System Implementation and On-site Operation Process

The proposed method is implemented with two lightweight tools, the "algorithm list extraction tool" and the "online parameter comparison tool", which work together without an independent database or complex deployment and are convenient for on-site engineers. The system implementation environment is shown in Table 2. The list extraction tool is developed in Python [8] and is responsible for configuration project file parsing, algorithm block filtering and standardized list generation; the comparison tool is developed in Excel VBA [9] and is responsible for list loading, DDE interface invocation, online parameter acquisition, item-by-item comparison and difference identification.

Table 2. System implementation environment

Item	Configuration / description
Operating system	Windows XP
DCS platform	HOLLiAS MACS[10]
Configuration software	CoDeSys (compliant with IEC 61131-3)[7]
Data interface	DDE (Dynamic Data Exchange) interface
List extraction tool	Python program
Online parameter comparison tool	Excel VBA program
List / verification record file	Excel (.xls) format
Application object	CPR1000 nuclear power plant DCS control stations

ISSN: 2414-1895 DOI: 10.6919/ICJE.202611_12(11).0007

In on-site application, the complete operation process is: ① input the extraction conditions, the algorithm block names, database types, selecting project file path and result saving path; ② the tool automatically parses the configuration project and generates the standardized list named by station number; ③ open the comparison tool and load the list; ④ click the "Pre-download" button to batch-collect the online parameters before downloading; ⑤ on-site personnel perform the control station program downloading; ⑥ after the downloading is completed and operation resumes, click the "Post-download" button to batch-collect the parameters after downloading; ⑦ the tool automatically compares, highlights differences and saves the verification record; ⑧ the staff reviews the difference items, judges whether they belong to the intended modification, and outputs the difference parameter summary list.

3.2 Engineering Application Verification

The method has been applied to the maintenance of DCS control stations of CPR1000 nuclear power plants. Taking the verification of algorithm program AA3_CEX_AD06 of a control station (station No. 19) as an example: the tool automatically extracted all parameters to be verified, including function blocks such as RS, TP, SFT and CURVE, according to the set keywords, and generated the standardized list file named by station number; the two batch acquisitions before and after downloading were both completed in one pass, and the comparison results are shown in Table 3. Among them, the value of parameter RS01.Q1 was 0 before downloading and 1 after downloading, which was judged as a difference record and automatically highlighted in red, and the staff quickly located and confirmed the source of the change accordingly;

Table 3. Example of parameter verification comparison results (fragment)

Block	Parameter	Pre-download value	Post-download value	Result
AM01	AV	0	0	Consistent
CURVE01	TB	105	105	Consistent
RS01	Q1	0	1	Inconsistent (highlighted)

3.3 Efficiency Comparison and Complexity Analysis

The manual approach relies on manual operation in all steps of parameter acquisition, collection, correspondence and comparison, with high error risk, and the time consumed grows linearly with the scale of algorithm blocks; the proposed method automates all the above steps with low error risk, and the operation time mainly depends on the reading rate of the DDE interface, which is weakly correlated with the scale of algorithm blocks.

To further quantitatively characterize the efficiency difference of the two approaches, let the number of parameters to be verified be N, the manual per-item operation time for query, transcription and comparison be τ, the DDE per-item reading time be tDDE, and the fixed overhead of the automatic process (list loading, file saving, etc.) be T0; then the workloads of the two approaches can be expressed as

$$T_{man}(N) = \tau N,\ T_{auto}(N) = T_0 + t_{DDE} \cdot N \tag{1}$$

Accordingly, the proportion of time saved is

$$\eta(N) = \frac{T_{man}(N) - T_{auto}(N)}{T_{man}(N)} \times 100\% \tag{2}$$

ISSN: 2414-1895 DOI: 10.6919/ICJE.202611_12(11).0007

The model parameters and their illustrative values are shown in Table 4 (τ and tDDE take illustrative values according to typical on-site operation). Fig. 9 shows the workload growth trends of the two approaches with the scale of parameters to be verified and the time saving ratio curve: the workload of the manual approach grows linearly with N (a convex curve on the logarithmic axis), reaching about 33 h when N reaches the order of 2000 items; the workload of the proposed method is mainly determined by the fixed overhead and grows very slowly with N-for 2000 parameters, the two acquisitions and comparison take only minutes, and the time saving ratio exceeds 99%, with greater advantage for larger N. This is consistent with the "minute-level completion and greatly shortened operation time" effect observed in engineering applications.

Table 4. Complexity model parameters and illustrative values

Parameter	Meaning	Illustrative value
N	Number of parameters to be verified	10–2000 items
τ	Manual per-item operation time for query, transcription and comparison	1 min/item
tDDE	DDE per-item reading time	0.05 s/item
T0	Fixed overhead of the automatic process (list loading, file saving, etc.)	2 min

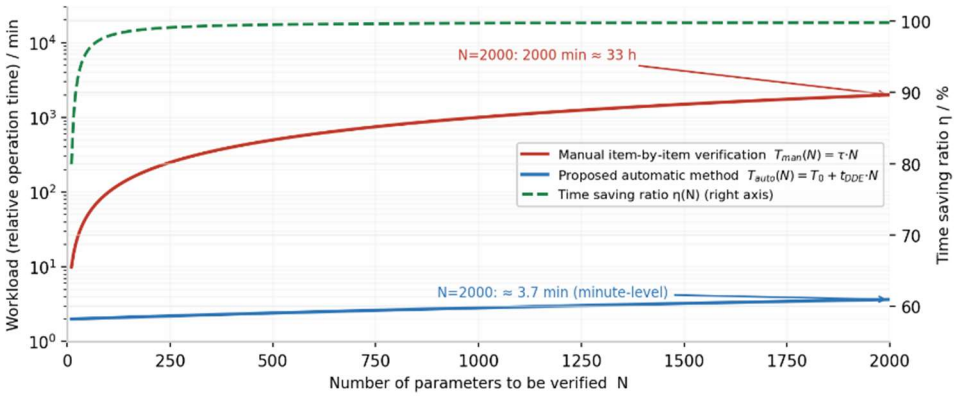


Fig. 9 Workload growth trends of the two approaches with the parameter scale and the time saving ratio (theoretical complexity illustration)

The proposed method still has two limitations: first, it relies on the DDE interface provided by the configuration software-if the platform interface form changes (e.g., upgraded to OPC UA)[6], the parameter reading expression generation step needs to be adapted accordingly; second, parameters for which DDE reading fails still require manual check, and fully unattended operation cannot yet be achieved.

4. Conclusion

The paper proposes an automatic control station algorithm-list extraction and online parameter verification method.Online parameters are batch-collected through the DDE interface by a standardized file which generated automately by parsing configuration project files. Then item-by-item automatic comparison is performed according to the list correspondence with differences highlighted, forming an integrated automated verification process of "extraction–acquisition–comparison".

The limitation of the method lies in its dependence on the availability of the DDE interface of the configuration software, and parameter extraction and comparison are scattered in two tools. Future work will study the adaptive expression generation mechanism for new-generation industrial data interfaces such as OPC UA, and explore integrating two tools to further enhance convenience.

ISSN: 2414-1895 DOI: 10.6919/ICJE.202611_12(11).0007

References

[1] Wang, C., & Luo, A. (2010). *Design and application examples of distributed control systems (DCS)* (2nd ed.). Publishing House of Electronics Industry.

[2] International Atomic Energy Agency. (2016). *Design of instrumentation and control systems for nuclear power plants: IAEA safety standards series No. SSG-39*. IAEA.

[3] International Electrotechnical Commission. (2011). *IEC 61513:2011 Nuclear power plants-Instrumentation and control for systems important to safety-General requirements for systems*. IEC.

[4] International Electrotechnical Commission. (2013). *IEC 61131-3:2013 Programmable controllers-Part 3: Programming languages*. IEC.

[5] Microsoft. (2019, August 24). *Dynamic data exchange*. https://learn.microsoft.com/en-us/windows/win32/dataxchg/dynamic-data-exchange

[6] OPC Foundation. (2003). *OPC data access custom interface standard, Version 3.00*. OPC Foundation.

[7] 3S-Smart Software Solutions GmbH. (n.d.). *CODESYS development system documentation*. https://www.codesys.com/

[8] Python Software Foundation. (2026, November 13). *Python 3 documentation*. https://docs.python.org/3/

[9] Microsoft. (2022, July 12). *Excel VBA reference*. https://learn.microsoft.com/en-us/office/vba/api/overview/excel

[10] HollySys Group. (n.d.). *HOLLiAS MACS distributed control system technical manual*. HollySys Group.