MIGRATION OF CONTROL SYSTEM OPERATOR INTERFACES FROM EDM TO PHOEBUS AT TPS

L. P. Hsu*, C. Y. Liao, C. Y. Wu, J. K. Liao, Z. Q Wu, C. K. Yang
National Synchrotron Radiation Research Center, Hsinchu, Taiwan

Abstract

The Taiwan Photon Source (TPS) control system originally adopted the Extensible Display Manager (EDM) as its primary display framework. However, this architecture has gradually become insufficient in terms of maintainability, scalability, and system integration. To address these limitations, the TPS control system migration introduces Phoebus as the next-generation operator interface framework and redesigns the overall HMI architecture around it. Phoebus is based on a modular and layered JavaFX architecture and integrates middleware services such as alarm management, data archiving, and save-and-restore functionality. The migration process includes automated batch conversion and manual refinement, together with centralized version control to ensure deployment consistency. Using TPS as a case study, this work establishes a complete upgrade workflow and architectural solution. The introduction of Phoebus significantly improves system maintainability and scalability, while also providing a solid foundation for future modernization of the control system. This paper describes the migration process of reconstructing the TPS operator interface and related system architecture with Phoebus as the core platform.

INTRODUCTION

In the Taiwan Photon Source (TPS) control system, the Human-Machine Interface (HMI) plays a critical role. It not only serves as the primary interface between operators and accelerator equipment, but also directly affects fault diagnosis and overall operational efficiency. The TPS control system is primarily built on the EPICS framework [1], whose distributed architecture enables real-time data exchange and control among subsystems through the network. For more than a decade, TPS has relied on the Extensible Display Manager (EDM) [2] as its primary display and operation interface. During the early stages of system deployment, EDM provided a low-barrier and efficient graphical development environment, allowing engineers to rapidly construct control pages and integrate accelerator equipment. However, as the TPS system continued to expand, hardware platforms evolved, and operational requirements became increasingly complex, the architectural and technical limitations of EDM gradually emerged and began to constrain further system development.

From the perspective of display management, TPS has accumulated a large number of EDM display pages over years of system expansion. These displays were developed by different engineers across different periods, often without unified design standards or naming conventions. As a result, the display files exhibit inconsistent structures and visual styles, reducing the overall maintainability of the system.

Regarding data model support, as the EPICS V4 architecture has matured, control systems have gradually evolved from the traditional Channel Access (CA) mechanism toward support for structured data models such as PVStructure and high-dimensional data types [3]. However, EDM only supports CA communication and lacks native support for modern EPICS data structures, preventing it from directly handling complex data structures. From the perspective of software lifecycle and technology evolution, EDM development has largely stagnated, with limited community support and feature updates. The underlying technologies on which it depends, such as Motif and X11, are also gradually being phased out by modern operating systems, introducing long-term compatibility and maintenance risks. In addition, newer generations of engineers are increasingly unfamiliar with these legacy technologies, creating potential sustainability challenges in technical expertise and manpower.

The challenges TPS faces with EDM are therefore not limited to the display tool itself, but rather stem from the gap between the legacy display architecture and modern control system requirements. These issues involve display performance, maintainability, data integration capability, scalability, and long-term supportability, all of which become increasingly significant as the system continues to grow. Consequently, the modernization effort focuses not only on replacing the display tool, but also on reexamining the HMI architecture and technology selection at the system level to ensure long-term maintainability and scalability under highly reliable accelerator operation.

Requirements

Phoebus [4], as the successor to CS-Studio, is based on standard Java and JavaFX technologies and provides a modular, layered architectural framework together with integrated middleware services such as alarm management, archiving, save-and-restore, and logging. Phoebus has already been used in EPICS-based control applications. These capabilities enable operator interfaces to integrate more naturally into modern control system architectures. The installation and deployment of Phoebus and display panels must remain highly simplified for end users. The solution should not depend on accelerator-specific software orchestration frameworks and must support macOS, Windows, and Linux platforms without requiring pre-installed third-party software.

* hsu.lp@nsrrc.org.tw

OVERVIEW OF THE TPS CONTROL SYSTEM GUI

The TPS control system is built on the EPICS platform and includes multiple subsystems such as RF systems, vacuum systems, beam diagnostics, and magnet power supplies. The system contains thousands of PVs and hundreds of operator interfaces. Existing operator interfaces are primarily based on EDM (.edl) display files, while some functions are implemented using MATLAB-based applications. However, MATLAB introduces version dependency issues, which may lead to compatibility and maintenance difficulties across different execution environments. In addition, the system depends on numerous scripts that operate collaboratively. These resources are centrally stored on a dedicated server and accessed through a Network File System (NFS). Because the MATLAB applications in the system were compiled using older 32-bit versions, their runtime environments exhibit strong dependencies on the underlying operating system platform. As a result, operator consoles also rely on relatively old operating systems to maintain compatibility and preserve legacy functionality. However, such outdated software stacks have become increasingly incompatible with modern development environments, particularly when integrating newer tools.

The TPS main control GUI remains one of the most frequently used operator interfaces. A comparison between the legacy EDM version and the migrated Phoebus version is shown in Fig. 1. The Phoebus version preserves the overall control structure while adapting the interface to the new display environment.

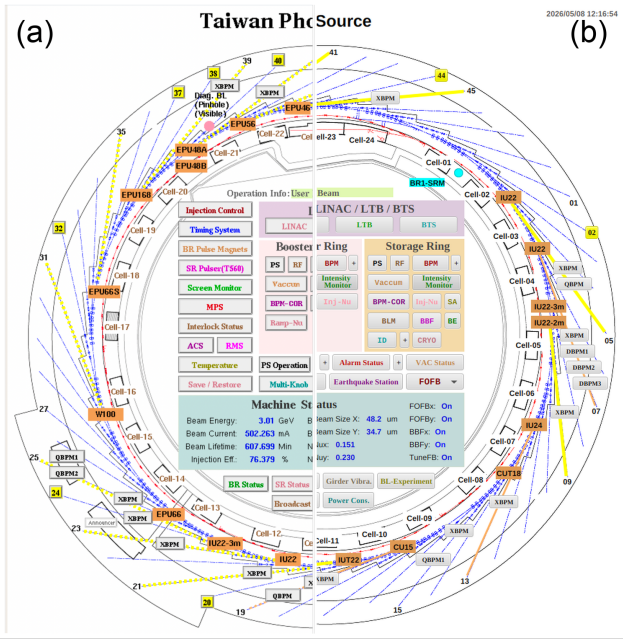

Figure 1: Comparison of the TPS main control GUI in (a) EDM and (b) Phoebus.

For the migration survey, 556 EDM pages were inventoried across the TPS control system. By May 2026, 505 pages had been converted to Phoebus, corresponding to an overall completion rate of 90.8%. Storage-ring-related displays account for the largest portion of the migration workload. The distribution of converted and remaining pages by merged subsystem category is shown in Fig. 2, and the overall migration status is summarized in Table 1. This distribution was used to determine the conversion priority and validation sequence.

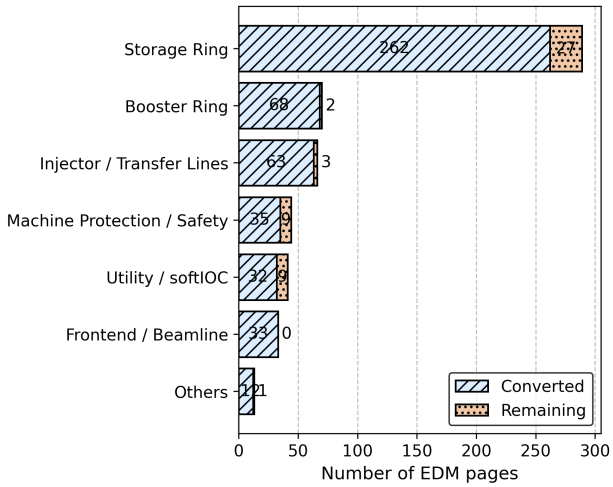

Figure 2: Distribution of inventoried EDM pages by subsystem category. The stacked bars indicate converted pages and remaining pages in each subsystem.

Table 1: Summary of the EDM-to-Phoebus Migration Status

Item	Value
Total EDM pages inventoried	556
Converted Phoebus pages	505
Remaining pages	51
Overall completion rate	90.8%
Pages requiring macro handling	28
Legacy startup scripts identified	42
Legacy command blocks identified	653

PHOEBUS SYSTEM UPGRADE STRATEGY

Phased Migration

To reduce the risks associated with large-scale one-time conversion, a phased migration strategy was adopted. The process began with establishing a source-control baseline to ensure conversion consistency. Existing EDM displays were then inventoried and analyzed, including statistics on display quantity, complexity, and PV usage. Representative displays were selected for pilot testing. Through multiple stages of small-scale implementation, the effectiveness of the conversion tools could be continuously validated and the migration methodology iteratively refined before gradually extending the process to the entire system.

Conversion Pipeline

The Phoebus-provided conversion utility was used to batch-convert the inventoried EDM (.edl) display files into

Phoebus (.bob) display files. The Display Builder automatically mapped most basic widgets. Most common components could therefore be converted automatically, significantly reducing repetitive engineering effort. Conversion reports were also generated during the process. Some display pages failed to convert successfully because EDM used composite layouts containing nesting errors, malformed attributes, or format errors that the conversion tool could not correctly parse and reconstruct. These cases required manual intervention and correction.

Manual Refinement and UI Optimization

Since EDM and Phoebus differ in widget architecture and display behavior, manual inspection and refinement were required after automatic conversion. Typical tasks included improving widget mappings, adjusting paths and macros, rewriting scripts and refining layouts. After functional equivalence was achieved, additional UI optimization was performed, including standardizing color schemes and icons, improving layout readability, and reorganizing navigation workflows. These refinements ensured that the final interfaces were not only functionally complete but also provided a better user experience.

The overall conversion workflow is illustrated in Fig. 3. The TPS migration effort involved more than direct file-format conversion. From the surveyed legacy resources, 42 startup scripts and 653 command blocks were identified. Many operator interfaces were coupled to external shell-based actions, launcher scripts, and subsystem-specific runtime behaviors.

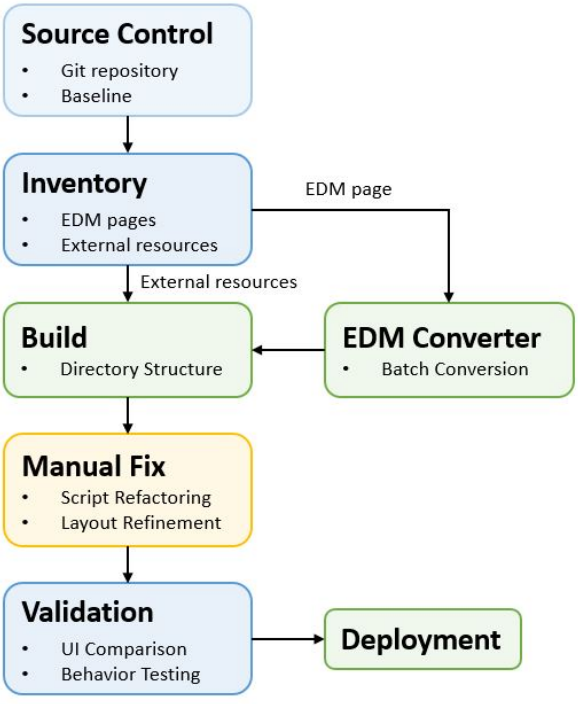

Figure 3: Workflow used for the EDM-to-Phoebus migration at TPS. The process includes display inventory, batch conversion, manual refinement, validation of PV bindings and scripts, and final deployment.

During the migration process, substantial manpower was required to validate the large number of existing display pages and EPICS Process Variables (PVs), ensuring that display logic, data bindings, and control behavior remained fully consistent with the original system. In addition, 28 pages required explicit macro-related handling to preserve runtime parameter passing, file references, or display-link behavior after conversion. Many legacy custom scripts and plugins also relied on specific runtime environments or deprecated APIs and therefore could not be directly reused on the new platform. These components had to be rewritten or replaced with equivalent solutions. Typical issues identified in this stage are summarized in Table 2. Furthermore, the migration also introduced changes to HMI operation workflows, requiring operators to become familiar with the redesigned interfaces and interaction models.

Table 2: Typical issues identified during migration and the corresponding handling methods.

Issue Type	Typical Cause	Resolution
Unsupported nested widgets	EDM composite structures could not be fully parsed by the automatic converter	Manual reconstruction and widget remapping in Phoebus
Macro and path inconsistency	Legacy relative paths and runtime macro usage varied across displays	Path cleanup and macro normalization after conversion
Script and command incompatibility	Shell-based actions and launcher scripts were tightly coupled to the old environment	Replaced with JavaScript, external tools, or revised invocation logic
Layout degradation after conversion	EDM and Phoebus differ in widget geometry and rendering behavior	Manual UI refinement to recover readability and navigation

CONCLUSION

The EDM-to-Phoebus migration at TPS was carried out through a staged workflow combining automated conversion and manual refinement. By May 2026, 505 out of 556 inventoried EDM pages had been converted, corresponding to a completion rate of 90.8%. The conversion results indicate that automatic tools can handle a large fraction of standard displays, but macro handling, legacy command blocks, startup scripts, and layout differences remain major sources of manual engineering work. These results support the continued migration of TPS operator interfaces toward a more maintainable Phoebus environment.

REFERENCES

[1] Y.-S. Cheng *et al.*"Development Status of the TPS Control System", *Proc. ICALEPCS'13*, San Francisco, CA, USA, Oct. 2013, paper MOMIB02

[2] John Sinclair, Extensible Display Manager, 2007. https://controlssoftware.sns.ornl.gov/edm/eum.html

[3] M. Kraimer, K. Isupov, and M. M. Petrov, "EPICS V4 and PVAccess", *Proc. ICALEPCS'11*, Grenoble, France, Oct. 2011, paper WEAAUST04.

[4] K. Shroff *et al.*, "PHOEBUS: An open ecosystem for control system applications and services", in *Proc. ICALEPCS'25*, Chicago, IL, USA, Sep. 2025, pp. 116–121. doi:10.18429/JACoW-ICALEPCS2025-MOCR002