DCS for TPC MPD: Integrating DIM and MasterSCADA 4D
DCS для TPC MPD: Объединение DIM и MasterSCADA 4D

S. Sokolova[a,1], *S. Movchan*[a,2], *S. Sergeev*[a,3]

С.М. Соколова[b], *С.А. Мовчан*[b], *С.В. Сергеев*[b]

[a] JINR, Dubna, Russia

[b] ОИЯИ, Дубна, Россия

Система управления детектором (DCS) Трековой Проекционной Камеры (TPC) в эксперименте NICA/MPD [1] использует DIM (Distributed Information Management System) [2] для быстрой и надежной передачи данных между TPC и MPD DCS. Для обеспечения мониторинга мы интегрировали DIM с промышленной SCADA-системой MasterScada 4D [3]. Данное решение может быть применено к другим субдетекторам MPD.

The Detector Control System (DCS) of the Time Projection Chamber (TPC) in the NICA/MPD experiment [1] uses DIM (Distributed Information Management System) [2] for fast and reliable data transfer between TPC and MPD DCS. To provide a monitoring, we integrated DIM with the industrial SCADA system - MasterScada 4D [3]. This solution can be applied to other MPD subdetectors.

PACS: 29.40.Cs, 29.50.+v, 07.05.Hd, 07.05.Dz, 07.07.Tw

Introduction

The Multi-Purpose Detector (MPD) at the NICA (Nuclotron-based Ion Collider fAcility) complex is designed to investigate phase transitions in strongly interacting baryonic matter. The Time Projection Chamber (TPC) is a primary tracking device for the MPD experiment, demanding precise and stable operation of its associated subsystems. These include High Voltage (HV) and Low Voltage (LV) power supplies [4], cooling systems, gas mixture regulation, and calibration lasers. The reliable and coordinated operation of these systems is essential for data quality and detector safety, necessitating a sophisticated Detector Control System (DCS).

The MPD DCS uses a hierarchical tree-like architecture (Fig. 1). The central DCS, responsible for the global coordination and user interfaces, is built around the DIM protocol, chosen for its high performance and reliability in distributed environments typical of large-scale HEP experiments. The low-level control of the TPC equipment produced by vendors like CAEN

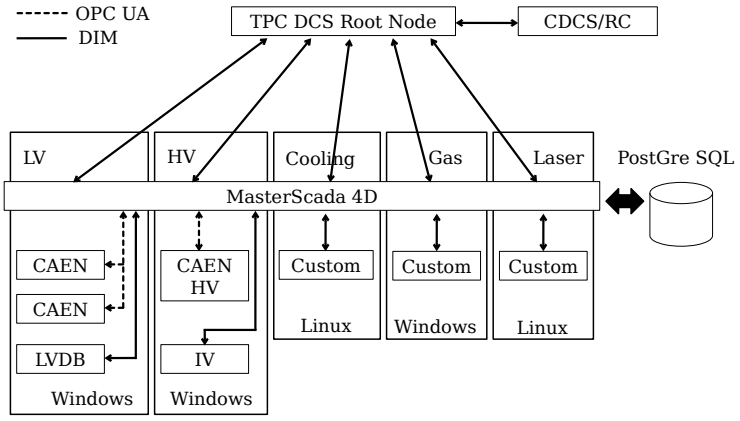

Fig. 1. Part of DCS scheme (scheme of TPC DCS), developed by Sergeev S.

is performed by the industrial SCADA system MasterSCADA 4D, running Windows platforms.

Due to that the establishing of a reliable, low-latency, and bidirectional communication channel between the DIM-based central DCS and the MasterSCADA-based hardware controllers becomes a non-trivial job. Native integration between these two systems was not available. This paper details the design, implementation, and testing of a custom middleware solution that bridges this gap, ensuring seamless integration of the TPC into the overarching MPD DCS framework.

Operational Principles of the Functional Block

The core of the integration solution is a custom C++ Functional Block that operates as a persistent service within the MasterSCADA 4D runtime environment. Its operation can be conceptually divided into two concurrent flows: the data publication flow from SCADA to DIM, and the command processing flow from DIM to SCADA (Fig. 2). Upon initialization, the block first parses its configuration to establish a mapping between internal MasterSCADA variables, which represent the real-time states and parameters of hardware like power supplies and sensors, and the corresponding DIM services that will represent this data to the external world. After that it initializes a DIM server, making these services available in the DIM name address space, and simultaneously starts a DIM command service to listen for incoming instructions. The data publication flow is cyclical and event-driven. The block continuously monitors the values of the bound SCADA variables.

[1]E-mail: sokolovasm@jinr.ru
[2]E-mail: movchansa@jinr.ru
[3]E-mail: sergeev@jinr.ru

When a change in any of these values is detected, the block does not immediately transmit it. Instead, it engages in a data conversion process. The raw value from SCADA, which might be a simple boolean, a floating-point number, or a complex structured value, is translated into a standardized format comprehensible to the central DCS. A critical part of this conversion is the state mapping, where various native hardware status codes are unified into a single state model, such as translating a collection of individual error flags into a composite «Error» state for the entire subsystem. This processed data, now formatted as a string following the *State_Message* convention, is then published by updating the corresponding DIM service. To optimize network bandwidth and processing load, the block operates with an update cycle of 100 milliseconds, but it is crucial to note that the DIM service is only updated upon a data change, preventing unnecessary network traffic during periods of stability. The command processing flow is asynchronous and interrupt-driven. It stays in a wait state till a command from the central DCS is received. After arrival of a command, such as hv_all_on, is received by the DIM command server, the block activates a validation routine. This routine checks the command for a syntactic correctness and, what is more important, for a semantic validity within the SCADA system current context, ensuring that a command to enable the high voltage should not be executed if, for example, the cooling system is reported to be failed. If the command is validated successfully, the block writes the corresponding value to a specific control variable within the MasterSCADA project. This action is the key link that closes the control loop; writing to this variable triggers the pre-programmed and often complex control sequence within the MasterSCADA logic—sequences that handle the safe ramp-up of voltages, the switching of gas lines, or the activation of laser calibration pulses. This entire operation ensures a continuous and robust bidirectional synchronization, making the states of the physical hardware instantly visible to the central DCS while allowing high-level commands to propagate down to the hardware level reliably and safely.

System Architecture and the Integration Challenge

The TPC DCS architecture is a multi-layered structure. At the top of the TCP subdetector tree resides the Subdetector Root Node Manager which communicates using DIM protocol. This node provides the main graphical user interface for TPC experts and executes the high-level control logic. At the bottom layer reside several independent MasterSCADA 4D instances, each dedicated to controlling specific hardware groups such as CAEN-based systems for LV distribution and HV generation, and custom-developed systems for cooling, gas management, and laser control, often hosted on embedded Linux or Windows computers. The primary goal was to enable a real-time data publication and command execution. The challenges were multifaceted. They included protocol heterogeneity between the industrial

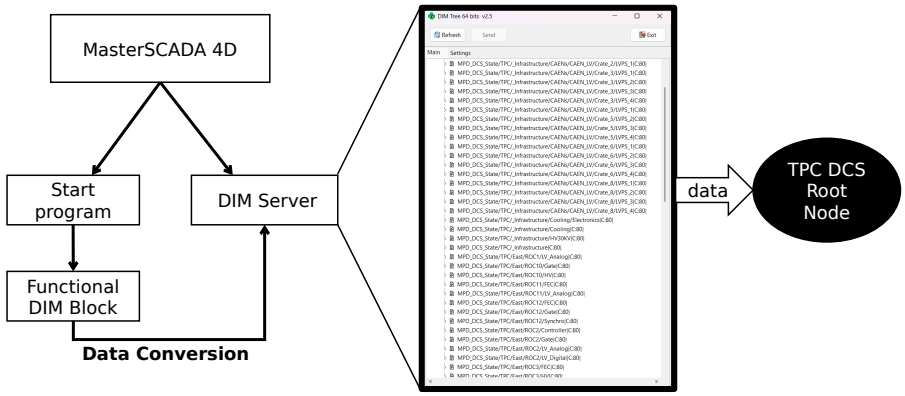

Fig. 2. C++ Functional Block operation

SCADA world and the custom HEP communication protocol. Another challenge was achieving sub-second data refresh rates for real-time control without overloading the systems. It was also crucial to define a consistent and comprehensive data model that accurately represents the state of complex subsystems. Finally, ensuring fault-tolerant operation was essential: this involved handling network interruptions and validating commands to prevent erroneous hardware control.

Data and Command Mapping

A critical contribution of this work is the definition of a unified state model for detector subsystems. This model provides a consistent semantic layer for interpreting statuses across different types of equipment. The published data string follows a strict format where a state code is paired with a descriptive message. For instance, a state code of 3 indicates a «Ready» condition, while a code of 5 signals an «Error» state. This formalization eliminates ambiguity in system status interpretation. Commands are implemented as discrete DIM services. The execution flow begins when an operator initiates an action in the Central DCS graphical interface (Fig. 3). The central DCS manager sends a DIM command to the specific MasterSCADA node's Functional Block. Upon receipt and validation, the block sets the appropriate control flags within MasterSCADA, which in turn executes the predefined sequence for that command, ultimately controlling the hardware via dedicated drivers and libraries. This seamless flow allows for centralized imperative control over distributed hardware systems.

Implementation and Performance

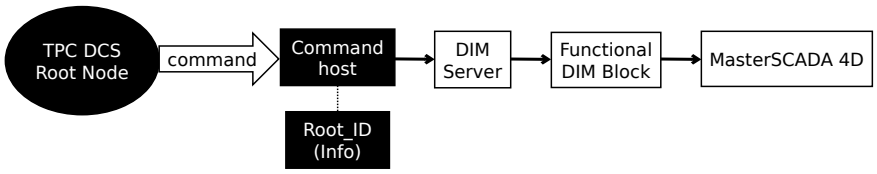

Fig. 3. Commands operation

The Functional Block was implemented in C++ using the DIM library and integrated into a MasterSCADA 4D project. The system has undergone extensive testing with TPC prototype systems. Performance metrics confirm its efficacy, with data refresh rates for critical parameters kept below 100 milliseconds and the command round-trip time from GUI to SCADA variable update typically under 50 milliseconds. The solution has demonstrated remarkable stability, operating continuously for weeks without memory leaks or crashes, even under simulated network failure conditions. It successfully handles the anticipated data volume for all planned TPC subsystems, proving its readiness for integration into the live experimental environment.

Conclusions and Future Goals

This paper presented the successful development and testing of a custom integration bridge between the MasterSCADA 4D and DIM systems for the MPD TPC DCS. The implemented C++ Functional Block provides a robust, efficient, and scalable solution for bidirectional real-time communication, fully integrating the TPC's low-level control into the unified MPD DCS framework. The significance of this work extends beyond the TPC. The architecture is designed for reusability. Therefore, our future work is directed towards finalizing integration and commissioning for all TPC subsystems, generalizing the Functional Block code into a configurable template to drastically reduce integration effort for other MPD subdetectors like the ECAL and FHCAL, and ultimately establishing this integration pattern as a standard for detector control within the NICA complex to ensure uniformity, simplify maintenance, and enhance the overall reliability and scalability of the control infrastructure.

Funding

This work was funded by the Joint Institute for Nuclear Research (JINR) within the framework of the NICA/MPD project 02-1-1065-2007/2026 budget. No additional grants were received for the conduct or management of this specific study.

Conflict of Interest

The authors declare that they have no conflicts of interest.

REFERENCES

1. V. I. Kolesnikov and A. I. Zinchenko, "Status of the NICA/MPD project," [arXiv:1312.1091 [nucl-ex]].

2. DIM (Distributed Information Management System) [Electronic resource]. – CERN. – Available: https://dim.web.cern.ch (accessed: 10.10.2023).

3. https://insat.ru/scada/scada-systema-masterscada-4d

4. Y. Ghoniem, A. Pilyar and T. Smolyanin, "Slow Control for MPD TPC LV and HV Systems Based on CAEN Equipment," Phys. Part. Nucl. Lett. **21**, no.6, 1145-1151 (2024) Doi:10.1134/S154747712470170X