Fusion Engineering and Design 84 (2009) 1049–1054

Contents lists available at ScienceDirect

Fusion Engineering and Design

journal homepage: www.elsevier.com/locate/fusengdes

Development and commissioning results of the KSTAR discharge control system

Kukhee Kim[a,*], Mikyung Park[a], Sang-Hee Hahn[a], Myungkyu Kim[a], Jaesic Hong[a], Sulhee Baek[a], Taegu Lee[a], Sang-Il Lee[a], Jin-Seop Park[a], Jaehoon Choi[a], Jayhyun Kim[a], Hyun Sik Ahn[b], Kap-Rae Park[a], Yeong-Kook Oh[a]

[a] *National Fusion Research Institute, 52 Eoeun-dong, Yuseong-gu, Daejeon 305-333, Republic of Korea*
[b] *POSCON Corporation R&D Center, 101 Korea Techno Complex Building, Korea University, Seoul 136-713, Republic of Korea*

ARTICLE INFO

Article history:
Available online 20 December 2008

Keywords:
KSTAR
Superconducting Tokamak
Discharge control
EPICS
vxWorks
Linux

ABSTRACT

The Korea Superconducting Tokamak Advanced Research (KSTAR) control system has been developed as a network-based distributed control system composed of several sub-systems. There are many local control systems for various sub-systems, and the central control system includes discharge control, machine control, and safety interlocks which aim for integrated control of the entire system. We have chosen the Experimental Physics and Industrial Control System (EPICS) as the middleware of the KSTAR control system because EPICS provides a software framework to integrate heterogeneous systems. The discharge control system, which is implemented in a part of the supervisory control system, performs the discharge sequence execution. The plasma control system, which has been implemented with general atomics and modified for KSTAR, is involved in the discharge control. The plasma control system performs real-time plasma control algorithms and provides the results of the control algorithms to the magnet power supplies. We are using a reflective memory-based real-time network for communication between the plasma control system and the magnet power supplies, thus we developed a fully digital control for the magnet power supplies. We have implemented the discharge control system with state notation language (SNL) in EPICS and also developed interface software among the sub-systems. We will present the details of the development of the KSTAR discharge control system and commissioning results.

1. Introduction

The Korea Super Conducting Tokamak Advanced Research (KSTAR) [1] is a superconducting Tokamak, which is composed of diverse sub-systems. The sub-system named the local control system contains more than 200 controllers and workstations which are distributed in the KSTAR main building and accessory buildings. The integrated control system for KSTAR [2] was developed as a network-based distributed control system using the Experimental Physics and Industrial Control System (EPICS) [3]. The KSTAR integrated control system performs the control and monitoring of more than 13,200 processing variables in the operation and communicates about 55,200 events per second between the internal systems.

The supervisory control system synthesizes the administration of the local control systems. It controls and monitors the local control systems to maintain and preserve the operational state of KSTAR and manages the computing resources to archive the operation data and to display the data. The supervisory control system can be categorized into three functions: machine control, discharge control, and data management and visualization. Fig. 1 shows the structure of the KSTAR integrated control system and the function block of the supervisory control system.

The machine control monitors the operation status of the entire system for 24 h a day and maintains and preserves the operation. KSTAR has six operation stages: maintenance, vacuum pumping, cool-down, current charge-up, plasma experiments, and warm-up. The machine control establishes the operation stage, sets the alarm and interlock configurations to match the operation stage, and intervenes during the entire period of the KSTAR operation. The discharge control is activated during the plasma experiment stage in contrast with the machine control. It performs the pre-processing of the shot, the plasma discharge, and the post-processing. The machine control and the discharge control are implemented in the central control system (CCS) which is a part of the KSTAR supervisory control system.

2. Discharge control system

In the pre-processing step, the discharge control performs the preparation of the plasma discharge shot: the operator sets the timing configuration on the time synchronization system

* Corresponding author. Tel.: +82 42 870 1616; fax: +82 42 870 1609.
E-mail address: kimkh@nfri.re.kr (K. Kim).

0920-3796/$ – see front matter © 2008 Elsevier B.V. All rights reserved.
doi:10.1016/j.fusengdes.2008.11.005

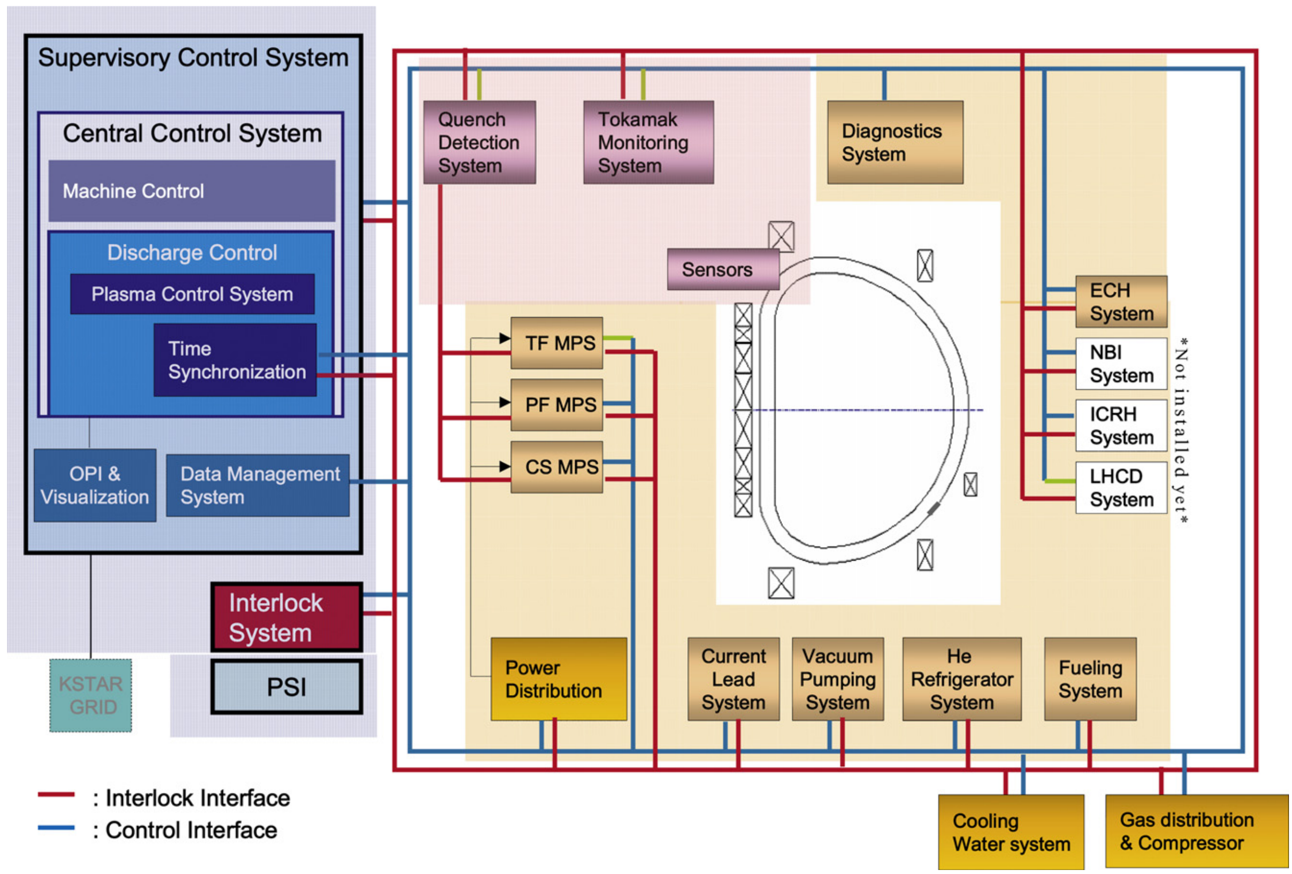

Fig. 1. The structure of the KSTAR integrated control system and the function block of the supervisory control system.

which describes timing information about diagnostics systems and heating systems. The plasma control system (PCS) [4–8] is also configured at this moment with the operation scenario which includes the magnet power supplies and gas puffing. Finally the discharge control checks all of the systems and configurations to start the plasma shot.

When the checking procedure is finished, the discharge control is moved into the shot step by the operator. In the shot, the discharge control performs the serial actions named the shot sequence. First, the discharge control places the PF magnet power supplies (PF MPS) into an operational state and then locks out the PCS. When the PCS enters into the real-time mode, the discharge control invests the control of PF MPS to the PCS and monitors the operation status of those two systems. When the plasma experiment is finished, the discharge control withdraws the control from the PCS and stops the PF MPS operation. During the plasma shot, if the watchdog processes detect some critical failure, which is implemented in the discharge control to monitor the operation status of the PF MPS and PCS, the discharge control shuts down the PF MPS and the PCS and then makes the interlock system follow it up.

After finishing the shot step, the discharge control performs the post-shot procedure. The discharge control monitors the diagnostics systems and heating systems to check completion of the data acquisition. When the discharge control detects the completion,

Table 1
The commands and the acknowledgements between the CCS, the PF MPS, and the PCS.

Shot sequence	Command					Acknowledge to CCS		Watchdog in CCS
	CCS to MPS	CCS to PCS	CCS to TSS	TSS to PCS	TSS to MPS	From MPS	From PCS	
	Operation permit					Permit Ack.		
Setting TSS			Timing Configuration					
MPS ready	Confirm Blip enable Zero crossing					Confirm Blip enable Ack. Zero crossing Ack.		
PCS standby		First lockout Final lockout					First lockout Ack. During shot	Watchdog start for PCS
MPS run	Run					Run Ack.		Watchdog sart for MPS
Shot			Shot trigger					
PCS start				PCS start Trigger				
Blip					Blip trigger	Blip Ack.		
End of shot							Writing file	Watchdog start for PCS
MPS stop	Stop					Stop Ack.		Watchdog stop for MPS

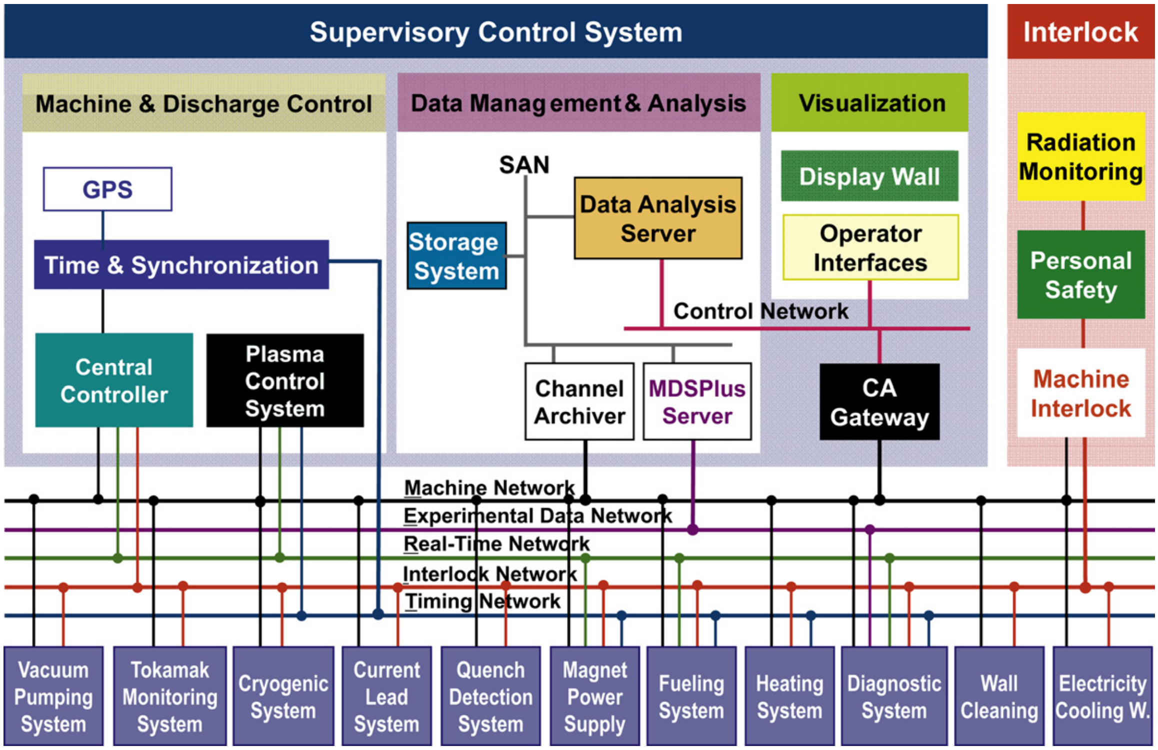

Fig. 2. The network interfaces in the KSTAR control system.

the discharge control transfers the data from the local system to MDSPlus [9].

In the procedure of the shot sequence, the discharge control exchanges a chain of commands and acknowledgements with other systems; in particular, communications with the PF MPS and the PCS are significant. The discharge control sends a command to the previous systems and waits for an acknowledgement from these systems to clear the performance of the command. After receiving acknowledgements, the discharge control sends the next command in the chain. If the acknowledgement is not received within a time limit, the discharge control makes an exception for the shot and starts a shot canceling procedure. Table 1 shows the commands and acknowledgements among the CCS, PF MPS, and PCS.

The discharge control contains watchdog processes to detect real-time violation for the PCS and the PF MPS. The PCS has a software counter on the reflective memory (RFM) and the counter is increased for every control cycle by the PCS. The software watchdog in the CCS monitors the counter. When the counter is increased, the watchdog timer, which is implemented in a part of the watchdog, is reset by the watchdog. If the time has expired, the watchdog sends a decision to the discharge control of a real-time violation on the PCS. The expiration for the time is a configuration variable; however the default value is set to 10 ms. Then the discharge control sends a 'Forced Abort' signal to the PCS via the RFM to confiscate the control and sends a fast shutdown command to the PF MPS and heating devices to protect KSTAR from a malfunction of the PCS. Finally, the discharge control reports it to the interlock system to make a follow up. The watchdog process performs in same manner with the PF MPS. Thus, there is one watchdog for the PCS and seven watchdogs for each PF MPS. The discharge control also detects several critical faults such as the plasma current fault, neutral density fault, electron density fault, PF current limit, PF voltage limit, and PF MPS fault during the shot to perform fast shutdowns of the PF MPS and heating devices. These are very important machine interlock functions to protect KSTAR and could not be performed by the interlock system, because the PLC in the interlock system cannot react on a sub-millisecond time scale. Thus, the discharge control complements the interlock system.

3. Implementations

To perform the previous functions, the discharge control was implemented in a part of the central control system because the discharge control requires various interfaces with other systems that the CCS can provide. The CCS has five networks [2] to connect with other systems and contains the central timing unit (CTU) in the time synchronization system (TSS) [10] as one of the hardware modules in the CCS. Fig. 2 shows the network interfaces in the KSTAR control system. The CCS contains a VMEbus-based PowerPC CPU board to accept various hardware modules which are designed for the VMEbus system, and is operated under the vxWorks [11] real-time OS to achieve real-time performance with the software. The CCS has two interlock interface modules and one reflective memory module on the VMEbus which are located in the VME crate and are connected with the CPU board through the backplane bus. The CTU module is a PMC mezzanine type and is located directly on the CPU board. Table 2 shows the hardware modules in the CCS.

We have implemented the software platform which is based on the EPICS R3.14.8.2. The software platform is composed of the operating system, EPICS base code, and a few device drivers. It provides hardware abstractions and an operating environment to the high-level applications. Table 3 shows the software platforms for the CCS.

As mentioned above, the discharge control and the machine control are implemented as high-level applications in the CCS which are running on the software platforms. We have minimized the portion of the EPICS database (DB) [12] programming and utilized the state notation language (SNL) [13] in the development of the high-level application. The EPICS DB cannot be modified during operation, but the SNL program can be modified during the runtime. To make that possible, we utilized the dynamic loader in vxWorks. Thus, we can make the sequencer stop, which is a part of EPICS function to exe-

Table 2
The hardware modules in the CCS.

Classification	Product name	Specification	Quantity
CPU board	VMIVME-7050	VME 6U PowerPC 1 GHz Memory 1 GB Gigabit Ethernet 2 ports	1 Module
Timing module (TSS)	In-house development	PMC type Optical network GPS time 100 MHz master clock Time accuracy < 5 μs	1 Module
Interface module for the interlock system	VMIVME-2534	VME 6U 32 Channel digital I/O	2 Modules
Reflective memory module (RFM)	VMIVME-5565	VME 6U Optical network 128 MB dual-port memory Data Thru-put 176 MB/s Delay 0.7 μs/node	1 Module

cute the SNL programs, by the EPICS IOC shell command, and then can reload the modified object into EPICS by the dynamic loader. Finally, we can restart the sequence with the modified object. We can perform it on the running control system. Thus, we can modify control software in the operation without stopping the system. We call it "Dynamic Re-loading." It should produce many advantages for commissioning and maintaining the control software. Fig. 3 shows the software function block in the CCS.

In addition, EPICS provides platform independency for the SNL program. The SNL program is executed by the sequencer which provides platform-independent execution environments and is running on EPICS channel access. Thus, the SNL program which was implemented for the PowerPC-vxWorks for the CCS can be running on other workstations even if these are Intel-x86 and Linux combinations, if the program is re-compiled on the new platform. It means a high-level application does not have to be running on the CCS, so we can move some of the software function block in the high-level application to another workstation. We call it "Migration." We can also combine the dynamic re-loading and the migration at the same time, which makes possible runtime migration. Thus, we can do the migration without stopping the system. It also has a large advantage for the commissioning and maintaining of software. We have prepared an Intel-x86/Linux workstation as the migration workstation for the CCS. This workstation executed some parts of the discharge control which were neither completed before the commissioning nor needed debugging. We fixed part of discharge control during the commissioning on the migration workstation and after the fix let it migrate into the CCS. We could reduce the down-time of the CCS using this mechanism. Fig. 4 shows the migration concept on the CCS and the migration workstation.

To realize long-term stability, we have restrained the dynamic memory allocations in the software. The repeated dynamic memory allocation and de-allocation can induce memory fragmentation and memory leakage which are a significant hindrance to long-term stability. When dynamic memory allocation is needed, we allocate memory from a pre-allocated memory pool which is allocated in the initialization step on EPICS, and when the allocated memory is no longer needed, return it to the memory pool. Thus, we have developed a memory allocation function in EPICS and use its own memory management function which is using a pre-allocated memory pool. We can avoid memory fragmentation and leakage with this method and improve the long-term stability.

4. The results of commissioning and operation

We have developed discharge control in the CCS which has around 800 records in database, around 30 EPICS native threads, around 90 communication threads, and more than 80 application threads. Thus it really became large-scale real-time control software. However it shows a very low-CPU load (<10%) in the operation because, the software is designed with an event driven mechanism to avoid the polling overhead which is based on an OS level synchronization mechanism. We also utilize dynamic re-loading and migration aggressively during the commissioning and real operation. As a result, the CCS has been stopped less than 10 times during 4000 h of operation and has minimized the down-time from a few

Table 3
The software platforms for the CCS.

Classification	Name	Specification	Remarks
Operating system	vxWorks-5.5	Host: Solaris/UltraSparc Target: PowerPC Real-Time OS	Commercial product
EPICS	Base-3.14.8.2 and extensions	vxWorks/PowerPC target	Open source
EPICS driver/device	drvVmic2534	Driver module forVMIVME-2534	EPICS bundle software (modified for the KSTAR)
	drvEtherIP	Communication driver for EtherIP DeviceNet/ControlNet	EPICS bundle software
	drvcFP20x0	Communication driver for National Instrument CompactFieldPoint	In-house development
	drvAsyn	Communication driver for the generic message exchanges	EPICS bundle software
	drvCLTU	Driver module for the TSS	In-house development
	drvVmivme5565	Driver module for VMIVME-5565	In-house development

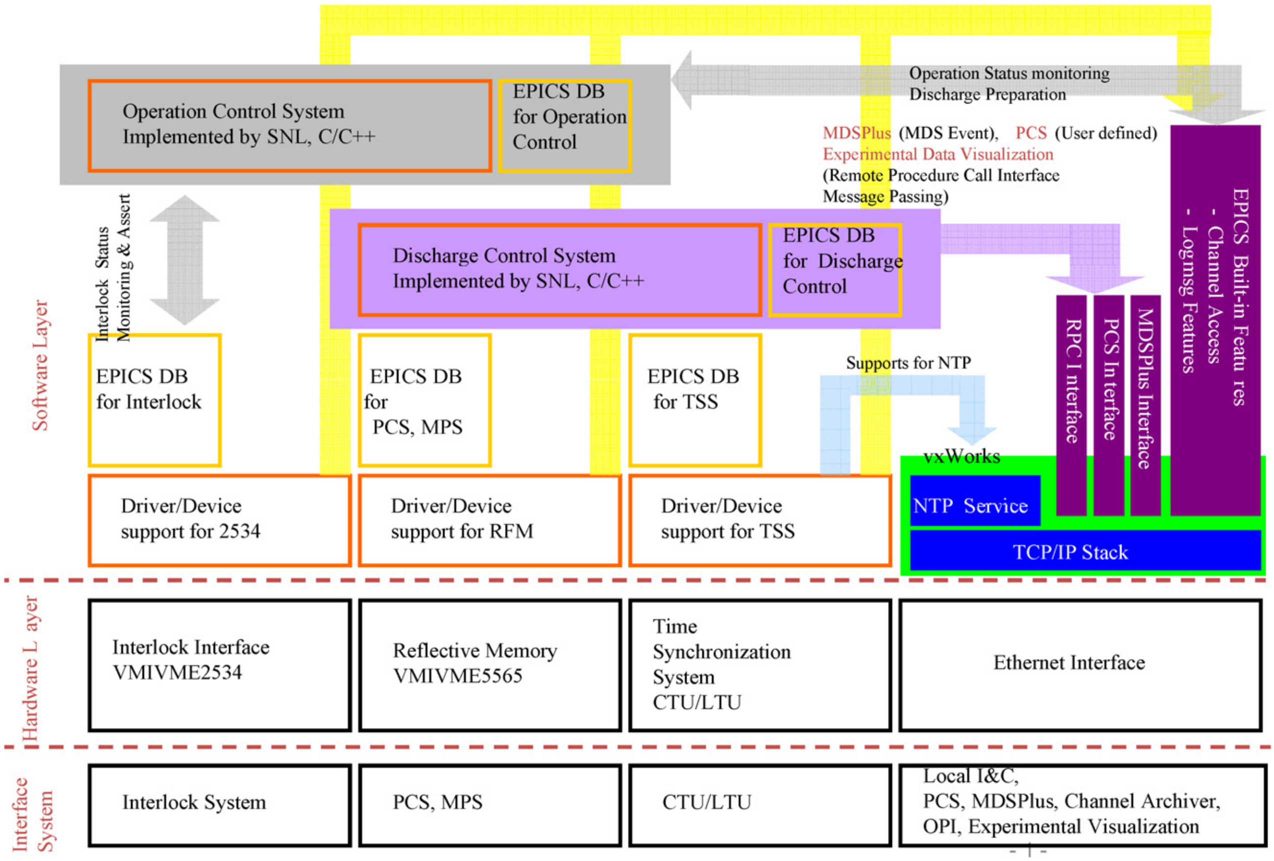

Fig. 3. The software function block in the CCS.

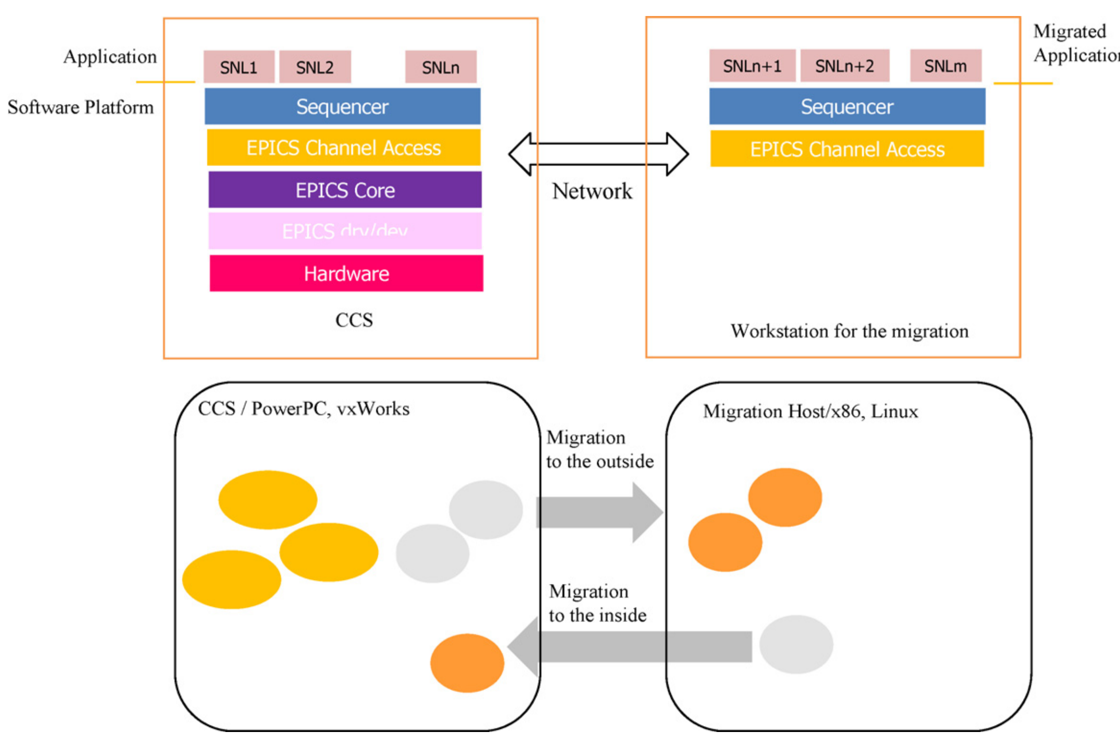

Fig. 4. The migration concept on the CCS and the migration workstation.

seconds to a few minutes, even including debugging and software modifications. There has never been a memory leakage or a system crash in the operation.

Acknowledgements

This work was supported by the Korean Ministry of Education, Science and Technology. We acknowledge substantive technical discussion with Dr. A.C. England, NFRI.

References

[1] G.S. Lee, J. Kim, S.M. Hwang, C.S. Chang, H.Y. Chang, M.H. Cho, The design of KSTAR tokamak, Fusion Eng. Des. 46 (1999) 405–411.
[2] K.H. Kim, C.J. Ju, M.K. Kim, M.K. Park, J.W. Choi, M.C. Kyum, M. Kwon, The KSTAR integrated control system based on EPICS, Fusion Eng. Des. 81 (2006) 1829–1833.
[3] R. Lange, EPICS: recent developments and future perspectives, in: Proceedings of the 9th ICALEPCS, October 13–17, 2003. http://www.aps.anl.gov/epics.
[4] H. Jhang, I.S. Choi, Design concepts for KSTAR plasma control system, Fusion Eng. Des. 73 (2005) 35–49.
[5] J.A. Leuer, R.D. Deranian, J.R. Ferron, D.A. Humphreys, R.D. Johnson, B.G. Penaflor, et al., DIII-D integrated plasma control tools applied to next generation tokamaks, Fusion Eng. Des. 74 (2005) 645–649.
[6] B.G. Penaflor, J.R. Ferron, M.L. Walker, D.A. Humphreys, J.A. Leuer, D.A. Piglowski, et al., Worldwide collaborative efforts in plasma control software development, Fusion Eng. Des. 83 (2008) 176–180.
[7] S.-H. Hahn, M.L. Walker, K.H. Kim, H.S. Ahn, B.G. Penaflor, D.A. Piglowski, et al., Plasma control system for the day-1 operation of KSTAR tokamak, SOFT2008 P3.23 and Fusion Eng. Des., submitted for publication.
[8] B.G. Penaflor, J.R. Ferron, M.L. Walker, D.A. Humphreys, J.A. Leuer, D.A. Piglowski, et al., Extending the capabilities of the DIII-D plasma control system for worldwide fusion research collaborations, the SOFT2008 P3.21 and Fusion Eng. Des., submitted for publication.
[9] J.A. Stillerman, T.W. Fredian, K.A. Klare, G. Manduchi, MDSPlus data acquisition system, Rev. Sci. Instrum. 68 (1) (1997) 939, http://www.MDSplus.org.
[10] M. Park, K.H. Kim, M.C. Kyum, M. Kwon, Development of a prototype time synchronization system for KSTAR with a VME-bus system, Fusion Eng. Des. 81 (2006) 1817–1821.
[11] vxWorks, WindRiver home page. http://www.windriver.com.
[12] EPICS Database, EPICS 3-14 Record Reference Manual. http://www.aps.anl.gov/epics/wiki/index.php/RRM_3-14.
[13] SNL, EPICS Sequencer Home page. http://www.slac.stanford.edu/comp/unix/package/epics/sequencer/.