Fusion Engineering and Design 88 (2013) 3041–3045

Contents lists available at ScienceDirect

Fusion Engineering and Design

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

J-TEXT-EPICS: An EPICS toolkit attempted to improve productivity

Wei Zheng[a,b], Ming Zhang[a,b,*], Jing Zhang[a,b], Ge Zhuang[a,b]

[a] State Key Laboratory of Advanced Electromagnetic Engineering and Technology, Huazhong University of Science and Technology, Wuhan 430074, China
[b] College of Electrical and Electronic Engineering, Huazhong University of Science and Technology, Wuhan 430074, China

HIGHLIGHTS

- Tokamak control applications can be developed in very short period with J-TEXT-EPICS.
- J-TEXT-EPICS enables users to build control applications with device-oriented functions.
- J-TEXT-EPICS is fully compatible with EPICS Channel Access protocol.
- J-TEXT-EPICS can be easily extended by plug-ins and drivers.

ARTICLE INFO

Article history:
Received 5 March 2013
Received in revised form 22 July 2013
Accepted 25 July 2013
Available online 22 August 2013

Keywords:
EPICS
CODAC
Tokamak
Control system

ABSTRACT

The Joint Texas Experimental Tokamak (J-TEXT) team has developed a new software toolkit for building Experimental Physics and Industrial Control System (EPICS) control applications called J-TEXT-EPICS. It aims to improve the development efficiency of control applications. With device-oriented features, it can be used to set or obtain the configuration or status of a device as well as invoke methods on a device. With its modularized design, its functions can be easily extended. J-TEXT-EPICS is completely compatible with the original EPICS Channel Access protocol and can be integrated into existing EPICS control systems smoothly. It is fully implemented in C#, thus it will benefit from abundant resources in.NET Framework. The J-TEXT control system is build with this toolkit. This paper presents the design and implementation of J-TEXT EPICS as well as its application in the J-TEXT control system.

1. Introduction

The Joint Texas Experimental Tokamak (J-TEXT) facility, formerly TEXT-U, is an experimental platform for magnetic confinement fusion research [1]. Many plant systems serve this tokamak, and as the number of plant systems keeps growing, a flexible control system is needed. Inspired by the ITER COntrol, Data Access and Communication (CODAC) [2] J-TEXTs control system is being updated using Experimental Physics and Industrial Control System (EPICS).

EPICS is very popular among large experimental physics facilities for its high performance. However, before ITER made the choice for EPICS, this tool and its use was practically unknown in the fusion community. EPICS is a signal-oriented control system framework [3], and developing an EPICS input/output controller (IOC) application is not a quick process. Instead of using the original EPICS IOC [4,5], we decided to develop our own EPICS software toolkit, named J-TEXT-EPICS, which is capable of building EPICS Channel Access (CA) server and client programs. J-TEXT-EPICS focuses on rapid development of applications. It is designed to improve the development efficiency of control applications. Inspired by object-oriented programming, J-TEXT-EPICS contributes device-oriented functions. With its well defined structure, its abilities can be extended by plug-ins easily.

This paper presents the advantages of using J-TEXT-EPICS, the design and implementation of J-TEXT-EPICS, and real-world applications of using J-TEXT-EPICS.

2. Purposes of J-TEXT-EPICS

ITER provided a great tool named ITER CODAC Core System a useful tool to build the EPICS-based ITER control system. However, the CODAC Core System is built specifically for ITER requirements. J-TEXT has many controllers that do not support the CODAC Core System. Although the CODAC Core System supports connecting non-EPICS hardware into the EPICS control network conveniently [6], the supported hardware is Siemens S7 PLC, which J-TEXT does not have. Moreover, J-TEXT, as a newly rebuilt tokamak, has an increasing number of plant systems. J-TEXT is also much smaller

* Corresponding author at: Room 307, Building East 15,Huazhong University of Science and Technology, Wuhan 430074, China. Tel.: +86 27 87793005x8307; fax: +86 27 87793060.
E-mail addresses: zhangming@hust.edu.cn, zheng.jtext@gmail.com (M. Zhang).

0920-3796/$ – see front matter © 2013 Elsevier B.V. All rights reserved.
http://dx.doi.org/10.1016/j.fusengdes.2013.07.008

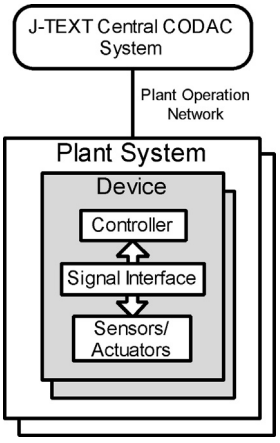

Fig. 1. Simplified control system diagram of J-TEXT and the concept of "device".

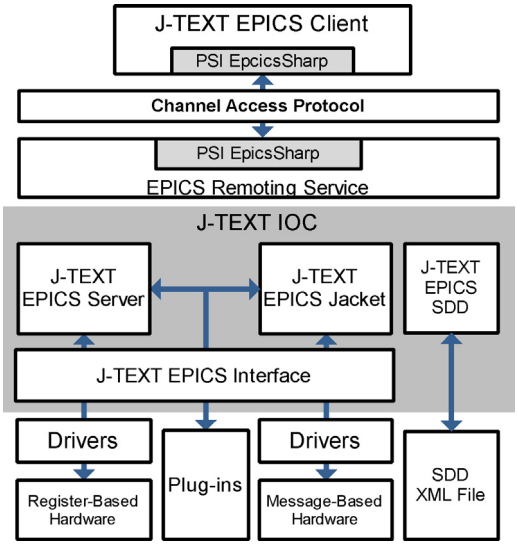

Fig. 2. Structure of J-TEXT-EPICS. The grey part represents J-TEXT IOC.

in scale compared with ITER. A tool is needed to build control systems for J-TEXT and integrate newly delivered plant systems into it quickly.

J-TEXT-EPICS is a device-oriented tool for plant system developers to build control systems within a short period. Fig. 1 shows a simplified diagram of a J-TEXT control system. J-TEXT is composed of many plant systems that are supervised by the Central CODAC System. In original EPICS-based control systems, reading process variables (PV) will obtain the status of a controller, whereas setting PVs can change the configuration of a controller or send a command to it. By using J-TEXT-EPICS, controllers, along with their connected sensors and actuators, are seen as "devices" from the Central CODAC point of view, shown as the gray filled part in Fig. 1. A "device" in J-TEXT-EPICS is a concept abstracted from a set of instruments and controllers implementing a given technical function. A "device" has status that indicates its states, configurations that operators can change, and methods that command the "device" to perform given tasks. Instead of manipulating individual PVs, only device names are needed to monitor, configure, and control the plant systems, thereby simplifying the process of deploying, integrating, and managing plant systems. J-TEXT-EPICS is compatible with EPICS CA. Therefore, other EPICS-based controllers can be integrated into the J-TEXT control system smoothly.

Also, J-TEXT has various types of controllers with different hardware and software, some of them are custom-build. Using the original EPICS IOC requires developing driver and device support for them. And implementing some very specialized task using the original EPICS IOC requires developing special record supports which needs in-depth knowledge about programming and EPICS. Besides being device-oriented, J-TEXT-EPICS offers very simple ways of creating extensions and control applications. Following very brief interfaces, drivers and plug-ins can be developed in a short period. And the plug-ins and drivers for J-TEXT-EPICS are loaded at runtime while the original EPICS IOC needs recompiling to used the extensions. Control applications can be configured with XML files, and control tasks can be written in Python script language. J-TEXT-EPICS is fully implanted using Microsoft.NET framework and thus it is supported by abundant.NET resources. J-TEXT-EPICS is designed to improve the productivity, not the performance. With the.NET garbage collecting mechanism, it is impossible to make J-TEXT-EPICS real-time capable. There are prior works that deal with the real-time capabilities of EPICS. A common way is to separate EPICS and real-time processes and using MARTe as real-time execution environment; the real-time processes use CA to communicate with EPICS [7]. Another way is implementing EPICS device support which manages the real-time threads [8].

3. Design and implementation of J-TEXT-EPICS

3.1. Structure of J-TEXT-EPICS

J-TEXT-EPICS contains a set of.NET programming libraries and software called J-TEXT IOC. Fig. 2 shows the structure of J-TEXT-EPICS. J-TEXT-EPICS libraries (JELibs) can be used to build CA server (CAS) or client software. J-TEXT IOC, which is built with these libraries, is similar to the original EPIC IOC. It is flexible and can be configured for various applications.

PSI EpicsSharp [9] is a CA library implemented in.NET. It is developed by the Paul Scherrer Institute (PSI). J-TEXT-EPICS uses this CA library to perform all types of CA communication.

At the top of Fig. 2 is an EPICS client built with JELibs. It is usually a type of human–machine interface (HMI) software. In a control system built with J-TEXT-EPICS, any EPICS compatible software can be used to build an HMI, such as Control System Studio or MEDM.

3.2. EPICS Remoting Service

The PSI EpicsSharp library only allows creation of one CAS on one controller. However, it is necessary to have multiple IOCs running on a single controller. J-TEXT-EPICS addresses this limitation by adopting the singleton design pattern. EPICS Remoting Service (ERS) is a background service program that serves a similar purpose as does EPICS standard caRepeater service. ERS provides each J-TEXT IOC a CAS proxy as if it has created its own, when in fact only one CAS is being managed by ERS. J-TEXT IOC accesses the ERS using.NET Remoting [10] Inter-Process Channel, which is based on Named Pipes. All CA protocol communications on the same controller are handled by a single ERS.

3.3. J-TEXT-EPICS Server Library

J-TEXT-EPICS Server Library (JEServer) included in the JELibs manages the J-TEXT-EPICS Record (JERecord) collection. Although J-TEXT-EPICS is device oriented, it has implemented records that support some fields common to all standard EPICS record types. JEServer also manages drivers and plug-ins.