Available online at www.sciencedirect.com

ScienceDirect

Fusion Engineering and Design 83 (2008) 413–416

Fusion Engineering and Design

www.elsevier.com/locate/fusengdes

An event-oriented database for continuous data flows in the TJ-II environment

E. Sánchez [a,*], A. de la Peña [a], A. Portas [a], A. Pereira [a], J. Vega [a], A. Neto [b], H. Fernandes [b]

[a] *Asociación Euratom/CIEMAT para Fusión Madrid, 28040 Madrid, Spain*
[b] *Associação Euratom/IST, Centro de Fusão Nuclear, Avenue Rovisco Pais P-1049-001 Lisboa, Portugal*

Available online 27 December 2007

Abstract

A new database for storing data related to the TJ-II experiment has been designed and implemented. It allows the storage of raw data not acquired during plasma shots, i.e. data collected continuously or between plasma discharges while testing subsystems (e.g. during neutral beam test pulses). This new database complements already existing ones by permitting the storage of raw data that are not classified by shot number. Rather these data are indexed according to a more general entity entitled event. An event is defined as any occurrence relevant to the TJ-II environment. Such occurrences are registered thus allowing relationships to be established between data acquisition, TJ-II control-system and diagnostic control-system actions. In the new database, raw data are stored in files on the TJ-II UNIX central server disks while meta-data are stored in Oracle tables thereby permitting fast data searches according to different criteria. In addition, libraries for registering data/events in the database from different subsystems within the laboratory local area network have been developed. Finally, a Shared Data Access System has been implemented for external access to data. It permits both new event-indexed as well as old data (indexed by shot number) to be read from a common event perspective.

Keywords: Databases; Data acquisition; Events

1. Introduction

Since start-up of the TJ-II device [1] raw experimental data have been stored in a multilayer database (MLDB) [2]. MLDB data are classified according to signal name, representing a physical magnitude, and shot number, corresponding to plasma discharge number. This classification scheme is valid for the majority of experimental data and is commonly used in current fusion devices. However shot number indexing is not well suited to next generation devices with continuous or long pulse operation. Even in the short-pulsed TJ-II device shot number tagging is unsatisfactory for data acquired continuously (e.g. coil temperature monitoring) or between discharges. An example of the latter is the neutral beam systems that need test pulses, for conditioning purposes, prior to beam injection into the TJ-II. Such data cannot be stored in the MLDB, rather they are acquired and stored/maintained using dedicated schemes for each subsystem.

The necessity of providing tools for centralized storage, classification and access for such data has motivated the development of a new database in which data are not tagged according to shot number, rather they are tagged in a more general way. Details of this development are outlined below.

2. An event-oriented database (EDB)

In the new TJ-II database an event is the key element used for data classification. An event is defined as any occurrence that is relevant for the TJ-II environment. Examples of events are alarms in control or diagnostic systems. A shot can be understood as a particular case of event. In order to facilitate the handling of events by different subsystems, events are uniquely defined for the whole system, and different events are defined for each subsystem. In this way the counting of events is the responsibility only of the subsystem that produces (or registers) the event occurrences. A definition is stored in the database for each event. Every time that an event of a given kind occurs it is registered in the database. Thus, event occurrences (EO) are used to tag the data.

The database is divided into two layers. The upper layer stores information about events as well as general information related to the data. This information is maintained in an Oracle relational

* Corresponding author. Tel.: +34 913466317; fax: +34 913466124.
E-mail address: edi.sanchez@ciemat.es (E. Sánchez).

0920-3796/$ – see front matter © 2007 Elsevier B.V. All rights reserved.
doi:10.1016/j.fusengdes.2007.11.007

Table 1
List of tables for event and data management

Purpose	Table name	Description
	EVNTS	Event definitions
Event management	EVNTS_KV	Data associated to event definitions
	EVREG	Event occurrences registry
	EVREG_KV	Data associated to event occurrences
	SIGS	Signal definitions
Data management	SIGS_KV	Data associated to signal definitions
	SIGREG	Signal registry
	SIGREG_KV	Data associated to signal registers
Data–event relations	SIG_EVNT	Signals-event occurrences relations

database [3]. The database schema is quite simple, comprising of tables for event management, data management and data–event relations (see Tables 1 and 2). In the lower layer, experimental data sets, coming from diagnostics or monitors, are stored.

Table 2 summarises the content of all the tables in the relational database. Information associated with particular event (signal) definitions, that is not always present, is stored as key-value pairs in a separated table named EVNTS_KV (SIGS_KV).

Table 2
Meaning of the different fields used in the relational database tables

Table	Field name	Description
	EUID	Event unique identifier
EVNTS	NAME	Event name
	DESCR	Event description
	EUID	Event unique identifier
EVNTS_KV	KEY	Parameter (key) name
	VALUE	Parameter value
	EO_ID	Event occurrence identifier
EVREG	EUID	Event unique identifier
	TIME	Event occurrence time
	EOC	Event occurrence counter
	EO_ID	Event occurrence identifier
EVREG_KV	KEY	Parameter name
	VALUE	Parameter value
	SUI	Signal unique identifier
SIGS	TYPE	Signal type
	NAME	Signal name (e.g.: line_density)
	DESCR	Signal description
	SUI	Signal unique identifier
SIGS_KV	KEY	Parameter (key) name
	VALUE	Parameter value
	SR_ID	Signal register identifier
	SUI	Signal unique identifier
	TIME_S	Start time of signal acquisition
SIGREG	TIME_E	End time of signal acquisition
	FORMAT	Data storage format
	SIZE	Size of the data file
	URL	Data file location
	SR_ID	Signal register identifier
SIGREG_KV	KEY	Parameter (key) name
	VAL	Parameter value
SIG_EVNT	SR_ID	Signal register identifier
	EO_ID	Event occurrence identifier

The same idea is used for storing information associated to event occurrences and signal registers. The field named TYPE in the table SIGS allows a general classification of signals and provides *a priori* knowledge about the treatment or representation that can be applied to a dataset from a given signal. Signal types include: time evolution, profile or image. The EVREG table stores one register for each event occurrence (EO). The SIGREG table stores one register for each dataset corresponding to a given signal. A dataset can have a previously undetermined number (at least one) of related EOs and new relations can be added *a posteriori*.

Raw data are stored in files on the TJ-II central server. Their location is registered in the SIGREG table (URL), so that data can be moved without modifying the data search codes. The data can be stored in different formats adapted to each particular data type. Indeed different data sets corresponding to the same signal can be stored in different formats.

The new database has been developed while trying to maintain, with minimal modifications, the MLDB database and its related software. A particular event with event identifier "SHOT" has been defined in the EDB. Each time a plasma discharge is carried out in TJ-II a SHOT EO is registered in the EDB, the shot number being its EOC. For all previous plasma discharges SHOT EOs have been registered. All software for write/read on the MLDB continues working as is [4] and is used for reading/writing shot-indexed data, also when data are searched with new tools (see Fig. 1). The EDB database extends MLDB storage capabilities for storing events and event-indexed data.

3. The EDB data access tools

Software tools have been developed to allow writing/reading of data (event-tagged) on/from the EDB from different subsystems. These tools follow a client/server architecture as previous developments [4,5]. A server program, developed in C language, runs in the TJ-II central server and allows the integration and reading of events and event-tagged data. It accesses the Oracle tables using Structured Query Language (SQL) and reads/writes raw data from/to the disk files. The conversion functions provided by Tru64 UNIX are used in this server to convert different float data formats, thus providing support for multiple client platforms.

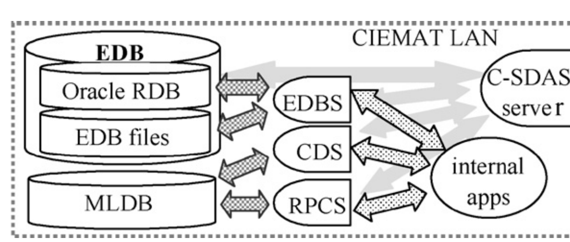

Fig. 1. EDBS server reads/writes event-tagged data. CDS server read/writes shot-tagged time evolution compressed data. RPCS reads/writes MLDB shot-tagged uncompressed data (time and profiles). The C-SDAS server allows access to all the data from outside (or inside) the LAN.