

Stanford Linear Accelerator Center



EPICS

Synchronized Actions

Ideas on How to Achieve Synchronized Gets
and Puts in an EPICS Environment

Matthias Clausen

SLAC/ESD

05/2001



Introduction

Synchronized actions in the EPICS environment have been an issue for a long time. Looking at the origins of EPICS, it is closer designed for generic process control than for individual machine controls with all their special needs and features.

Of course this is also the attractive side of EPICS with all its tools and building blocks which help to adapt it easily to various kinds of applications. Only some special applications remain to be solved because they fall through the cracks.

One of them being the synchronization of actions on individual IOC's.

Of course there are ways to synchronize actions on IOC's based on hardware timing systems. Record processing can be stimulated by individual settings in the timing system. But it is extremely difficult (if not impossible) to initiate synchronized actions (gets and puts) across IOC's from client programs.



Available Mechanisms

Synchronized Actions on an IOC:

- **On hardware interrupt**

Record processing can be activated based on a hardware interrupt. Records can be linked together in clusters of records which get processed together. Interrupt can be generated by individual I/O boards or by dedicated timing boards.

Timing boards typically also allow the setting of special masks in order to generate the interrupt on selected events.

- **On (soft) event**

The on-event processing in the IOC's provides a nice tool to trigger record processing in the IOC based on events which can be generated in various ways. (by event records, but also by any type of program on the IOC)

Since the event number in record processing can be also changed, source and destination of event processing can be modified on the fly and allow high flexibility.

The disadvantage is that event processing runs at a lower priority, cannot be combined with time based processing and cannot be synchronized over IOC boundaries.



Requirements I

- Writing new (desired) settings to records spread over several IOC's and activate these settings with a single synch command.
Applications:
 - Settings of magnets
- Reading data from various records in several IOC's on a synchronized event
(this might also require allocation of hardware for special readouts)
Applications:
 - Reading data from BPM's for a specific bunch
 - Correlate data taken at a precise time
 - Get data for a certain time at a certain rate



Requirements II

Writing new (desired) settings to records spread over several IOC's and activate these settings with a single synch command. The same command will also trigger the data acquisition.

Applications:

- Correlation plots.
 - Stepping of magnets through a certain range while.
 - The data acquisition of the synchronized channels takes place.



Requirements - Derived

CA clients will need to the ability to register with the desired new values for several records and an event number in several IOC's

Special CA server tasks will be necessary on the IOC

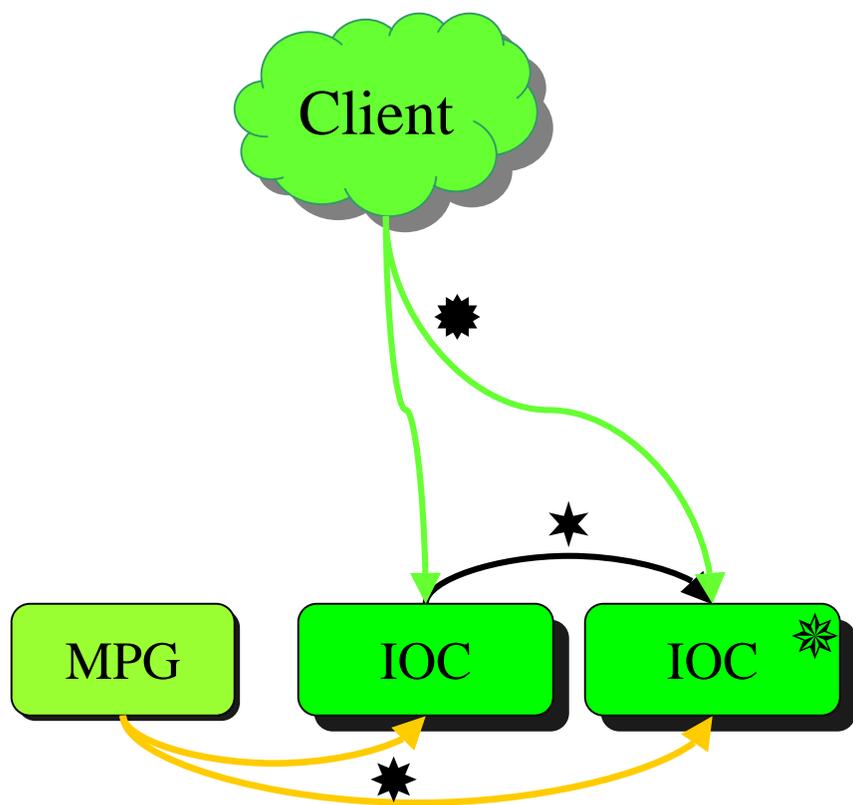
- The new portable CA server will help to ease this task

The handling of 'global' events will need to be embedded in the CA protocol

- The next version of CA protocol will support new data types and thus help to solve also this requirement



Trigger

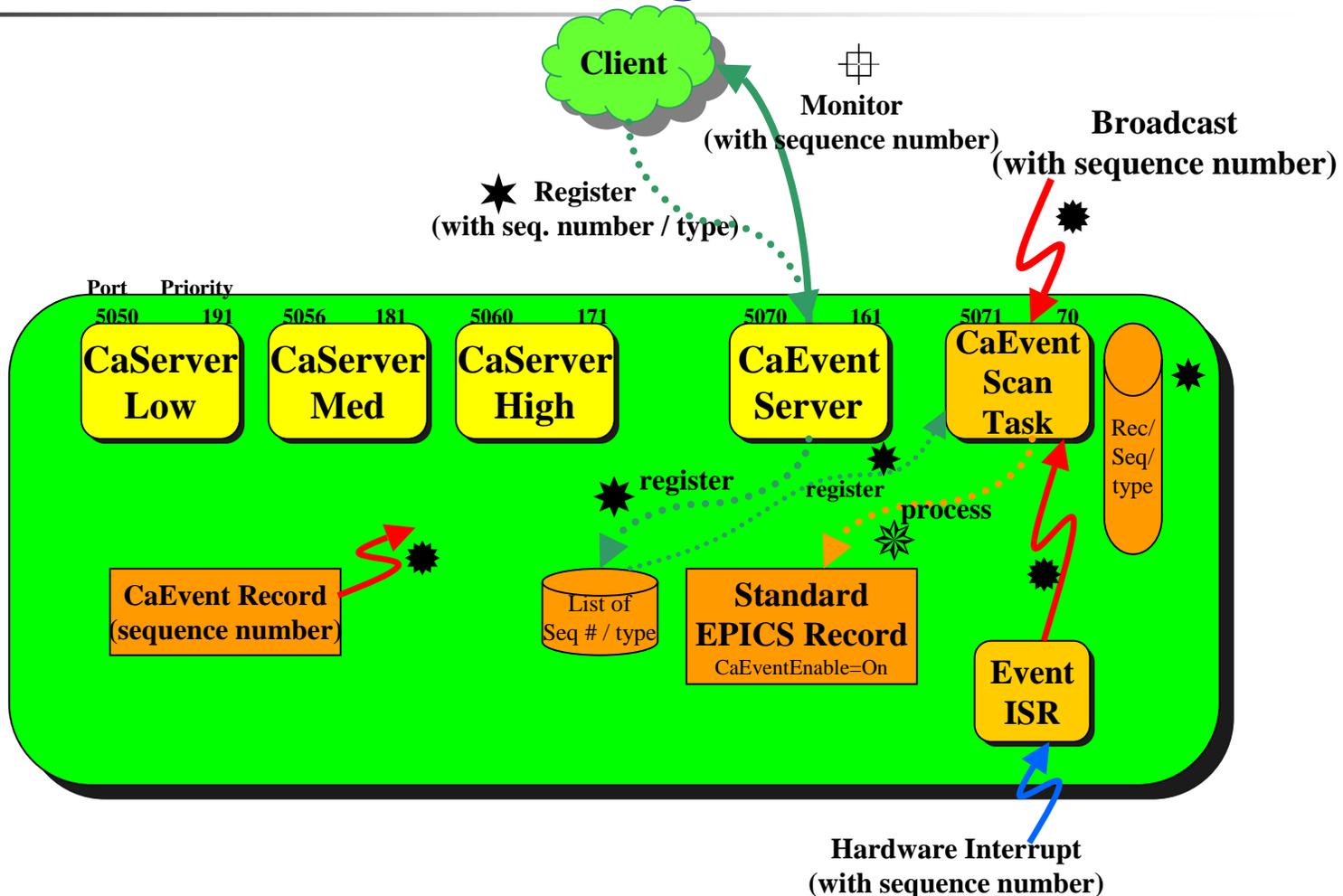


- ★ From IOC to IOC's
- ★ From MPG to IOC's
- ★ From client applications
- ★ Inside of IOC's

(MPG:= Master Pattern Generator)



IOC Processing



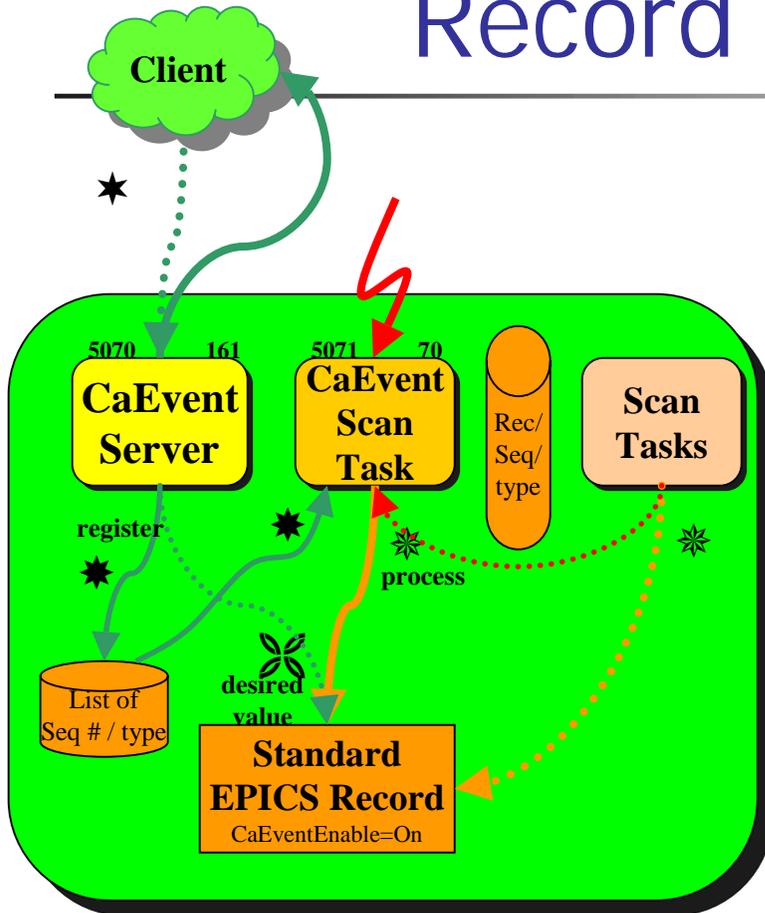


IOC Processing

- * Client performs caSearch to special CaEvent port and connects to record with special (unique) sequence number and (his)special processing (one of the standard EPICS record processing).
 - (one special **-new-** processing type is the synchronous put – the desired value will be processed as value on the indicated sequence number)
- * Special entries in record and CaEvent task register sequence number with special processing for this record.
- * CaEvent scan task gets activated by on of: broadcast, ISR, record
- * Records with the selected sequence number get processed
 - (more than one record can have the same sequence number)
- ⊕ The client receives the data from processing including the sequence number. (This way correlations with several records can be performed)



Record Processing



- * The caSearch will only be successful if the CaEventEnable flag is set to true. This prevents records from being activated which drivers do not allow processing by the CaEvent task.
- * The new desired value field for output records allows to perform synchronous puts if the sequence number is associated with the 'put' action.
- * Whenever the CaEvent task is activated it will look for:
 - Records registered with a certain sequence number
 - Records registered with a certain processing type (i.e. scan_30)
- * Records can be processed either by the CaEvent task or by the Scan tasks. If a record receives a sequence number for processing by the CaEvent task in addition to the already defined Scan time, the Scan Task will trigger the CaEvent task to process. This type of processing will only trigger the 'normal' monitors and not the monitors for the CaEvent server.
- * Records activated with the 'set desired value' processing type will all use the desired value as the new output value. Since the CaEvent task is running at a high priority, all the 'sets' will be performed at the 'same' time.



Next Steps

Looking for allies

- Who has similar requirements?
 - Is there something to change/add?
- Who would be interested to collaborate?
 - SLAC is willing to coordinate and throw in some manpower
- Time Schedule:
 - Start prototyping: 3rd quarter '2001
 - Preliminary version (no CA support): ~ 1st quarter '2002
 - Final version: depending on CA developments

Coordination see:

<http://www.slac.Stanford.edu/~clausen/EPICS/SynchronizedActionsIndex.html>