Comparative Analysis of EPICS IOC and MARTe for the Development of a Hard Real-Time Control Application

A. Barbalace, A. Luchetta, G. Manduchi, C. Taliercio, Consorzio RFX, Corso Stati Uniti 4, 35127 Padova, Italy
B. B. Carvalho, D. F. Valcárcel, IPFN, Av. Rovisco Pais, 1049-001 Lisboa, Portugal
Presenting author email: antonio.barbalace@igi.cnr.it

Going to develop a Real-Time Control Application?

Hard or Soft Real-Time? Which latency is acceptable? What about jitter?

Strict determinism or occasional event loss are tolerated?

What about exploiting the last hardware technologies on the market?

- Both are programming environments to code a **real-time control system** suited for **scientific experiments**.
- Both are **portable** on different operating systems and architectures.
- Both are written in **C/C++ language**.

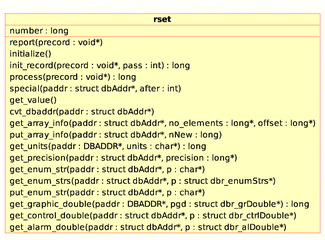
- Both are **component based** (components can be connected to carry out the required control algorithm).
- Both are **configurable at initialization** and **reconfigurable at runtime**.
- Both create during **initialization** all data **structures** that are **required at runtime**.

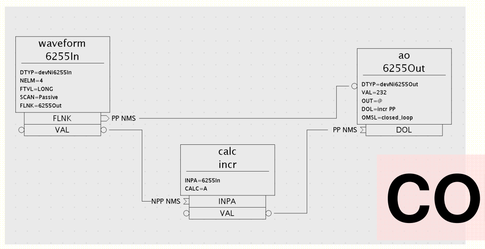
EPICS IOC
EPICS version 3.14.11

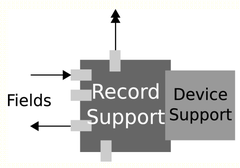
MARTe
MARTe/BaseLib CVS June 2011

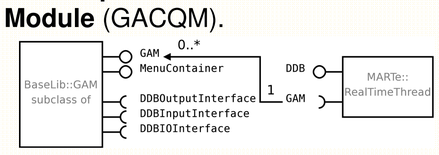
COMPONENT BASED APPROACH

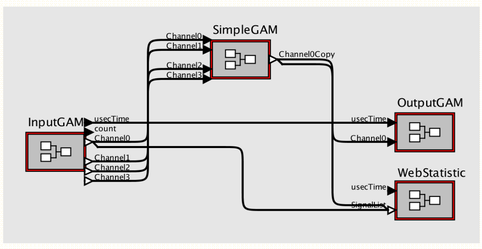
- Components are implemented by **records.**
- Records are logically grouped in **databases**.
- New record types (called **Record Support**) can be created by providing a set of routines and data structures.
- For **every record** type it is possible to define a new **Device Support** that enables the record to interact with an I/O device.

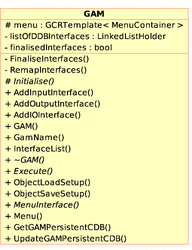












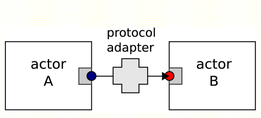
CONFIGURATION CAPABILITY

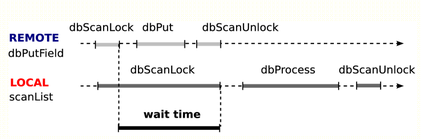
- The configuration of an EPICS IOC application spans across **different files.**
- The database configuration file is populated by **a list of records**.
- In EPICS a configuration file is used to **generate IOC source code** before compilation.
- When a new record is ready to be tested the developer has to **create** a new project, **insert** the record with ancillary files in the project, **create** the configuration files **compile all** and **run** the project.
- Value and compound data of a record can be **tuned at runtime**.
- To load or remove a record a user has to **change the configuration** files and **recompile** all the application from scratch before **running** it again.

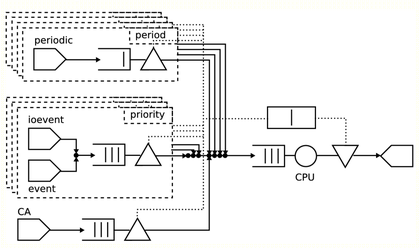
- The configuration of a MARTe application is held by a **single file**.
- A configuration file contains **different lists of GAMs** and **instance descriptors** of C++ classes.
- In MARTe a configuration file **is loaded** by a running MARTe system.
- When a new GAM is ready to be tested the developer has to **compile it**, **write** a test configuration file (modifying a template) and **run** the project.
- In order to change a GAM parameter it is necessary to **stop the execution** of every MARTe activity and **reload** the framework.
- To load or remove a GAM just **reload** the configuration file: it is not necessary to recompile the application.

CONCURRENT MODELS OF COMPUTATION

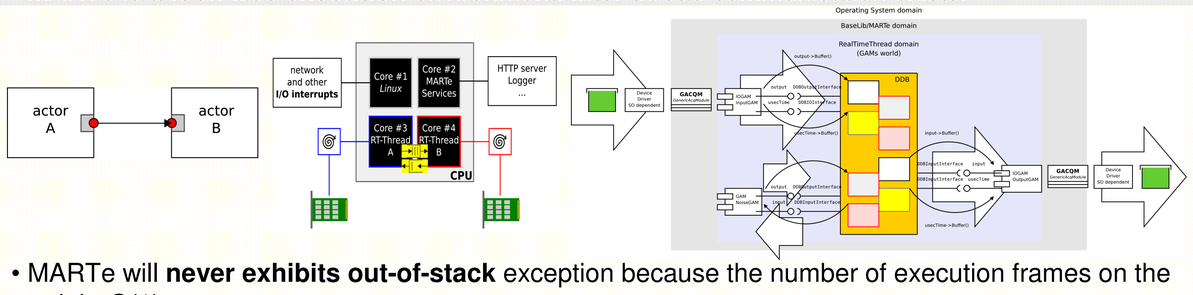
- A single record in an EPICS IOC can be **accessed concurrently** for **reading**, **writing** or **processing**. Reading and writing can trigger processing, before fetching a value (**demand-driven**) and after updating a value (**data-driven**).
- The processing of a chain of records takes place within a **scan** (an Operating System's thread).
- A scan can be locally triggered periodically (**periodic scan**) or by software and hardware events (**event scan**, **IO event scan**); remotely by **caGet** or **caPut** operations.
- After the processing has been triggered on the first record of a chain those records that have to be executed next are determined by the **associated link** in the database. A link can carry data and processing (INLINK, OUTLINK) or simply processing (FWDLINK).
- Since a single record can be concurrently accessed by different scans, **per-record locks exist**.







- A long EPICS' chain of passive records with many links can produce an **out-of-stack exception**, the number of execution frame on the stack is *O(n)* where *n* is the number of passive chained records.



- MARTe will **never exhibits out-of-stack** exception because the number of execution frames on the stack is *O(1)*.

- EPICS supports data diversity by means of a rich set of **protocol adapters**.
- In EPICS it is possible to have many INLINKs to the same record's field but it is **not possible** to have an OUTLINK or a FWDLINK **connected to more than one** record's field.

- MARTe does **not support data diversity**: two GAMs can communicate only if they both adhere to the same data interface.
- In MARTe one output signal from a GAM can be connected to as **many** GAMs as needed.

CONCURRENT MODELS OF TIME

- **No time model** can be assumed, a record can be processed at any time by different (concurrent) scans.
- A discrete algorithm will be executed in a non-uniform sampling environment. The smart way choosen in the PID record is to define a **minimum amount of time between record processings**.

- Each GAM is triggered **once per period** and the time elapsed between calls to GAMs is equivalent to period *T*.
- **Periodic timing** is provided by a GACQM interfaced with a TimeInput GAM. Algorithmic GAMs are aware of the absolute time and of the execution period T (via ExternalTimeTriggering service).

REAL-TIME SUPPORT

- The developer can not choose how to wait for an event, it fully **depends** on the **device driver**.
- Every event is **queued** for execution and **never lost**. There are different queues of execution and the execution policy is FIFO. One queue handles CA requests; three queues (low, medium, high priority) are shared between events and IO events and there is one queue per periodic scan.
- The EPICS Operating System Interface (OSI) layer does **not support multiprocessor environments**. A static set of queues is defined for each instance of EPICS IOC.

- The way in which a device driver (GACQM) waits for a hardware event is **selectable** (polling or interrupt) and defined in the **configuration file**.
- MARTe does not queue any event: events can be lost but there is **no jitter due to events enqueuing**.
- MARTe allow the developer to **assign** threads and IRQs to specific processors. The **affinity mask** of threads and IRQs on the processors is a parameter in the MARTe **configuration file**.

13th International Conference on Accelerator and Large Experimental Physics Control Systems
October 9 - 14, 2011 - Grenoble, France

Aknowledgments
This work was set up with financial support by Fusion for Energy.