

Beschleunigerkontrollen (Benutzerperspektive)

R.Bacher, MCS

Beschleunigerbetriebsseminar 2013

19.3.2013

Benutzerperspektive

- Benutzer:
 - BKR: Alle Personen, welche die BKR-Applikationen benutzen
 - Front-End: Alle Personen, welche für Subsysteme und Komponenten verantwortlich sind und Front-End Hardware mit dem Kontrollsystem verbinden bzw. selbst Device-Serverprogramme schreiben
- Welche Möglichkeiten bietet das Kontrollsystem für Benutzer?
- Was wird vom Benutzer erwartet? Auf was muss man sich als Benutzer einstellen?

(siehe auch "Wünsche an Automatisierung / Kontrollen")

Kontrollsysteminfrastrukturen (1/2)

- **Zwei unterschiedliche, weitgehend getrennte Kontrollsysteminfrastrukturen (wird die nächsten Jahre auch so bleiben):**
 - Betreiber:
 - MCS1 (in Zusammenarbeit mit MSK): **“TINE-Welt”**
 - MCS4: **“DOOCS-Welt”**
 - **Vollwertige** Kontrollsysteme:
 - Datenkommunikationsmöglichkeiten
 - Dienste (Archive, Alarme, Data Acquisition, Logbücher, ...)
 - Werkzeuge (Programmierschnittstellen, Kontrollsystem-Browser, Editoren / Wizards, Templates, Scripte, Bibliotheken,)
 - Universelle und spezielle Schnittstellen zur Beschleunigerhardware
 - Timing-System
 - EDV-Infrastruktur (Remote-Überwachung, Repositories, File-Systeme, Web-Server, Hardware- / Software-Support, Netzwerk-Support...)

Kontrollsysteminfrastrukturen (2/2)

- **Zwei unterschiedliche, weitgehend getrennte Kontrollsysteminfrastrukturen:**
 - **Brücken / Gemeinsamkeiten** zwischen beiden Kontrollsystemtypen*
 - **Ziel:** Bei gemischter Infrastruktur sollen die Benutzer (BKR) nicht wissen müssen, von welchem Kontrollsystemtyp gerade die Daten stammen
 - **Betrifft:**
 - Datenkommunikationsmöglichkeiten
 - Dienste (Archive, Alarmer, Logbücher)
 - **Unterschiedliches Kundenmodell:**
 - Was ist Aufgabe des Betreibers?
 - Was ist Aufgabe des Benutzers?

* Es gibt noch weitere Kontrollsysteme bei DESY (elektrische Infrastruktur, Wasserkühlung, Klimatisierung, Kälteversorgung, Beamlines und Experimente), mit denen die Beschleunigerkontrollsysteme kommunizieren können müssen.

Kontrollsystemlandkarte

| | PETRA3-Komplex, REGAE | FLASH, XFEL |
|---|---|---|
| Betreiber | <ul style="list-style-type: none"> • MCS1*, MCS4 | <ul style="list-style-type: none"> • MCS4 *, MCS1 |
| BKR-Konsolen <ul style="list-style-type: none"> • Betriebssystem • Programmiersprachen • GUI (Graphical User Interface) | <ul style="list-style-type: none"> • Windows* • Java*, VB/VB.NET**, MATLAB, LabView, HTML/JavaScript, Scripting, Konfigurieren • programmierte Applikationen*, konfigurierte Applikationen | <ul style="list-style-type: none"> • Mac OS*, Solaris** • Konfigurieren*, C++**, MATLAB, Java, HTML/JavaScript, Scripting • konfigurierte Applikationen*, programmierte Applikationen |
| Device-Server <ul style="list-style-type: none"> • Betriebssysteme • Server-Modell • Programmiersprachen • Device-Schnittstellen | <ul style="list-style-type: none"> • Windows*, Linux, VxWorks • TINE-artig*, DOOCS-artig • Java*, VB/VB.NET**, C / C++, LabView • CANopen*, SEDAC, TCP/IP, PC104*, PCI**/PXI, VME, μTCA, (Soft)-SPS | <ul style="list-style-type: none"> • Linux*, Solaris** • DOOCS-artig*, TINE-artig • C++*, Java • μTCA*, VME**, CANopen, SEDAC, TCP/IP, (Soft)-SPS |

Legende: * hauptsächlich / ausschließlich , ** wird abgelöst / nicht (mehr) empfohlen

Kontrollsystemlandkarte

| | PETRA3-Komplex, REGAE | FLASH, XFEL |
|--|--|---|
| Betreiber | MCS | MCS1 |
| BKR-Konsolen: <ul style="list-style-type: none"> Betriebssystem Programmiersprachen GUI (Graphical User Interface) | <ul style="list-style-type: none"> V J L k p k | DS*, Solaris** guriere *, C++**, MATLAB, HTML/JavaScript, Scripting gurierte Applikationen* , ammierte Applikationen |
| Device-Server: <ul style="list-style-type: none"> Betriebssysteme Server-Modell Programmiersprachen Bussysteme und Überrahmen | <ul style="list-style-type: none"> V T J C PCI**/PXI, VME, µTCA | *, Solaris** CS-artig* , TINE-artig , Java *, VME**, CAN, SEDAC, TCP/IP |



Legende: * hauptsächlich / ausschließlich, ** wird abgelöst / nicht (mehr) empfohlen

BKR-Applikationen / Displays (1/2)

- Displays zur **Bedienung bzw. Überwachung von Subsystemen oder Komponenten** (z.B. Magnete, Vakuum, Kicker)
@XFEL (FLASH)*:
 - Autoren: Subsystem- bzw. Komponentenexperten
 - Typ: jddd-Displays
- Displays zur **Beschleunigerbedienung** (z.B. Übersichtsdarstellungen, Catalog / Sequenzer bzw. Save/Restore-Tool)
@XFEL (FLASH): *(siehe auch “jddd Features für den Betrieb”)*
 - Autoren: Operateure, Koordinationsteam, MCS
 - Typ: jddd-Displays, Java Applikationen

Legende: * XFEL (FLASH) spezifisch

BKR-Applikationen / Displays (2/2)

- Displays für **komplexe Betriebsaufgaben** (z.B. SASE Control, Top-Up Betrieb)

@XFEL (FLASH): *(siehe auch “High-Level Anwendungen bei XFEL/FLASH”)*

- Autoren: MCS, MPY, Koordinationsteam
- Typ: jddd-Displays, Java Applikationen, MATLAB Applikationen

- Displays zur **Off-Line-Analyse** (z.B. History-Plots, Archive-Viewer, DAQ-Viewer)

@XFEL (FLASH):

- Autoren: MCS
- Typ: jddd-Displays (i.d.R. in andere Displays integriert), Java Applikationen

Kontrollsystemserver (1/2)

- Alle DOOCS-artigen Kontrollsystemserver sind zukünftig grundsätzlich über die **TINE-Kommunikationsprotokolle** erreichbar
- **@XFEL (FLASH):** Server die direkt **bunchsynchroner Daten** (“fast stream”) aus der **Beschleunigerhardware** auslesen (z.B. BPM-Server):
 - sind **grundsätzlich DOOCS-artige** Server
 - müssen in **C++** programmiert werden
 - liefern die bunchsynchroner Daten über ein spezielles DAQ-Protokoll an die DAQ-Systeme
 - laufen auf einem Rechner in einem μ TCA-Crate
 - Autoren: MCS, Subsystem- bzw. Komponentenexperten anderer Fachgruppen (z.B. MSK)

Kontrollsystemserver (2/2)

- **@XFEL (FLASH):** Server, die **bunchsynchroner Daten (“fast stream”)** aus den **DAQ-Systemen** verarbeiten (z.B. Orbit-Feedback):
 - sind **grundsätzlich DOOCS-artige** Server
 - müssen in **C++** programmiert werden
 - laufen auf dem gleichen Rechner wie das zugehörige DAQ-System, das die bunchsynchronen Daten bereitstellt (assoziierte DAQ-Server)
 - Autoren: MCS, Subsystem- bzw. Komponentenexperten anderer Fachgruppen (z.B. MSK, FLA)
- **Andere Server (“slow stream”):**
 - können **DOOCS-artige oder TINE-artige Server** sein (z.B. Getterpumpen, Magnetserver)
 - Autoren: MCS, Subsystem- bzw. Komponentenexperten anderer Fachgruppen (z.B. MHF-p)

μTCA @ XFEL (FLASH) (1/2)

- μTCA ist der **neue High-End Standard** für die Schnittstelle zur Beschleunigerhardware *(siehe auch “μTCA-Status”)*
 - Kommt auf jeden Fall dort zum Einsatz,
 - wo **bunchsynchroner Daten** verarbeitet werden
 - wo ein **hoher Datendurchsatz** gewährleistet werden muss
 - In jedem μTCA-Crate steckt ein Rechner mit Netzwerkanbindung
 - In jedem μTCA-Crate steckt bei Bedarf ein Timing-Empfängermodul; die Trigger / Events sind über einen internen Bus bzw. über dedizierte Kabelverbindungen verfügbar *(siehe auch “Timing System bei XFEL/FLASH”)*
 - Ein User-Slot in einem μTCA-Crate besteht aus **einem universellen bzw. speziellen AMC-Modul** (Vorderseite) und einem **speziellen RTM-Modul** (Rückseite), die in der Crate-Mitte zusammengesteckt werden

μTCA @ XFEL (FLASH) (2/2)

- μTCA ist der **neue High-End Standard** für die Schnittstelle zur Beschleunigerhardware
 - AMC-Module (**von MCS integrierte und unterstützte Standardmodule**):
 - **DAMC2**:
 - Digital I/O Board mit on-board FPGA-Chip, 4 optische Links, Speicher und Clock-Verteilung
 - Hinweis: Teile des FPGA-Code müssen aufgabenspezifisch implementiert werden (muss i.d.R. **von den Entwicklern der Front-End Elektronik** bereitgestellt werden)
 - **SIS8300**: 10-Kanal ADC, 16 bit, 125 MSamples/s
 - RTM-Module:
 - Müssen i.d.R. **von den Entwicklern der Front-End Elektronik** bereitgestellt werden
 - Enthalten die aufgabenspezifische Front-End Elektronik bzw. Schnittstellen (analog, digital) zur eigentlichen Front-End Elektronik, die sich auch ausserhalb des μTCA-Crates befinden kann

DAQ @ XFEL (FLASH)

- Hat eine **zentrale Bedeutung** bei XFEL (FLASH):
 - **Zusammenführung bunchsynchroner Daten (“fast stream”)** aus verschiedenen Datenquellen (z.B. RF, BPM) zu einem gemeinsamen, synchronisierten Datensatz
 - **Bereitstellung (im Speicher, nicht über das Netzwerk) bunchsynchroner Datensätze (“fast stream”)** für assoziierte DAQ-Server, welche diese Daten für ihre Aufgaben benötigen (z.B. Software-Feedbacks)
 - **Abspeichern der bunchsynchronen Datensätze** in Files für eine später Off-Line Analyse
- Es wird voraussichtlich **mehrere DAQ-Systeme** mit unterschiedlichen bunchsynchronen Datensätzen geben
- Nicht jedes DAQ-System muss assoziierte DAQ-Server haben

Historien und Archive

- Es gibt unterschiedliche Möglichkeiten / Archive, Historien von Daten (“slow stream”) mitzuschreiben
- Diese unterscheiden sich im Detail in ihren Eigenschaften
- Es gibt **nicht unbedingt für alle Daten** Historien / Archiveinträge
- Fehlende Historien / Archiveinträge werden **bei Bedarf zusätzlich** eingerichtet (die Mitarbeit der Benutzer wird erwartet)

- Möglichkeiten der Visualisierung bzw. Analyse von Historien:
 - jddd-Displays:
 - Historien werden mit einer Plot-Komponente dargestellt
 - Archive-Viewer:
 - Der dazugehörige Archive-Server behält seine Historien “für immer”
 - Statt der ursprünglichen Property-Namen können Alias-Namen für das Browsing benannt werden
 - **@XFEL (FLASH)**: Kennt auch DAQ-Kanäle (wird zur Zeit implementiert)
 - **@XFEL (FLASH)**: Funktionsumfang des Trend-Plots und des Correlation-Plots wird erweitert
 - Multi-Channel Analyzer

Snapshots

- Es gibt die Möglichkeit, bei Vorliegen bestimmter Trigger-Bedingungen server-seitig **Snapshots ausgewählter Daten** zu erzeugen
- Die Snapshot-Erzeugung kann erfolgen durch:
 - **@XFEL (FLASH)**: DAQ-System
 - Event-Archive System
- Die Snapshot-Kanäle werden **bei Bedarf** eingerichtet (die Mitarbeit der Benutzer wird erwartet)

Schlussbemerkungen

- **Offene Punkte:**
 - **@XFEL (FLASH):** Behandlung und Visualisierung von Alarmen
 - Absolute zeitliche Vergleichbarkeit der Updates bei Servern (“slow stream”):
1s?, 100ms?, 10ms?
 - Organisation des “1st-Level Supports” (Ansprechpartner) bei gemischten Systemen
 - ...
- **Herausforderungen:**
 - Hohe Erwartungshaltung an die Bedienbarkeit, Übersichtlichkeit, Automatisierung, Zuverlässigkeit etc.
 - **@XFEL (FLASH):** Rechtzeitige Verfügbarkeit von BKR-Applikationen / Displays mit den erwarteten Eigenschaften
 - **@XFEL (FLASH):** Rechtzeitige Verfügbarkeit der μ TCA – Komponenten (inklusive Firmware) mit den erwarteten Eigenschaften
 - **@XFEL (FLASH):** Leistungsfähigkeit des Kontrollsystems bei hohem Datenaufkommen
 - ...