

Indizieren mit BPID oder BPCID?

Ludwig Hechler

19. September 2012

Inhaltsverzeichnis

1. Einleitung	3
1.1. Betrachtete Geräte und Properties	3
2. Standard	4
2.1. Gepulster Magnet	4
2.2. Elektrostatisches Septum	5
3. MMI ins SIS18	6
3.1. Gepulster Magnet	6
3.2. Elektrostatisches Septum	7
3.3. Beam Production Chain Timestamp	8
4. SIS18 Booster Mode	8
4.1. Gepulster Magnet	8
4.2. Elektrostatisches Septum	9
4.3. Beam Production Chain Timestamp	10
5. Gleicher Beam aus SIS18 aber verschiedene Targets	11
5.1. Gepulster Magnet	11
5.2. Elektrostatisches Septum	12
6. Soll/Ist-Vergleich	12
6.1. Gepulster Magnet	12
6.2. Elektrostatisches Septum	13
7. Alternativen	13
7.1. Gepulster Magnet	14
7.2. Elektrostatisches Septum	15
7.3. BPCID und alternative Produktionskette	16
8. Gerät bedient mehrere Maschinen	17

9. Therapiebetrieb	17
9.1. Gepulster Magnet	17
9.2. Elektrostatisches Septum	18
10. Allgemeines	18
10.1. Commit Event	18
10.2. State Machine	19
10.3. Property ACTIVE	20
10.4. Istwerte aus mehreren BPs	20
10.5. Positionierbare Pressluftantriebe	21
11. Zusammenfassung: Wo klemmt's?	21
11.1. Was geht (bisher) nicht mit BPID?	21
11.2. Was geht (dann) doch mit BPID?	21
11.3. Was geht (bisher) nicht mit BPCID?	22
11.4. Was geht (bisher) allgemein nicht?	22
A. Funktion getBp()	22
B. Event-Format	24
C. Gateway zwischen GMT und GSI-Timing	24
C.1. Therapiebetrieb	25
Literatur	25

1. Einleitung

Das Papier ist ein Versuch, die Indizierung der Soll- und Istwerte im lokalen Speicher des Frontends nur mit der BPID oder nur mit der BPCID zu lösen. Der Index zum Zugriff auf die Daten muss sowohl von der Property (via API) als auch von der Realtime-Action (via Timing-Event) geliefert werden.

Die Idee, die BPCID als Index zu benutzen, kommt aus der Diskussion, ob wir eine zusätzliche *Sequence ID* (SID) benötigen (oder einfach nur haben sollten), die weitgehend der bisherigen VrtAcc-Nummer entspricht. BPCID und SID sind (bzw. wären) beide BPID-übergreifende oder -zusammenfassende Kennungen. Und vielleicht geht ja das, was man mit der SID erreichen möchte, auch mit der BPCID.

Einige Überlegungen zu BPID und BPCID beziehen sich nicht direkt auf die Indizierung der Daten, sondern auf deren Eignung für verwandte Zwecke.

Hiesige Kapitel beziehen sich zum Teil auf gleichnamige Kapitel in [2].

Punkte, bei denen es (noch) ungelöste Probleme gibt, werden am Rand mit !? gekennzeichnet.

!?

1.1. Betrachtete Geräte und Properties

1.1.1. Gepulster Magnet

Ein gepulster Magnet setzt zunächst einen Sollwert und liest später einen Istwert (und räumt am Ende des Zyklus auf).

1.1.2. Elektrostatisches Septum

Ein elektrostatisches Septum hat zwei Properties für verschiedene Sollwerte (S1, S2) und zwei Properties für die entsprechenden Istwerte (I1, I2). Setzen und Lesen geschieht zu verschiedenen Zeiten im Zyklus und in folgender zeitlicher Reihenfolge: S1, I1, S2, I2.

1.1.3. Standard-Properties

In die Überlegungen einbezogen sind die Standard-Properties SETTING und ACQUISITION, die jedes Gerätemodell haben muss. In der FESA Development Guideline [4] ist festgelegt, dass SETTING alle Sollwerte für einen Zyklus und ACQUISITION alle Istwerte (und wahrscheinlich auch Sollwerte) eines Zyklus enthält.

Allerdings ist es fraglich, ob 'Zyklus', also das, was bisher mit einer VrtAcc-Nummer

!

gekennzeichnet wird, bezüglich dieser beiden Properties in Zukunft noch eine vernünftige Kategorie ist. Wenn nicht, wird man die Guideline an dieser Stelle noch einmal überdenken müssen. Siehe z. B. Kapitel 2.2.1.

2. Standard

Use-Case-Nummer U-001.

2.1. Gepulster Magnet

2.1.1. BPID

1. Der Magnet bekommt den Sollwert S für BPID x und liefert den Istwert I für BPID y . Es sei nochmal darauf hingewiesen, dass x und y unterschiedliche BPIDs sind!
2. Die Property `SETTING` bekommt den Sollwert S für BPID x .
3. Die Property `ACQUISITION` liefert den Istwert I für BPID y (und z. B. den Gerätestatus, der BP-unabhängig ist).
4. Ich möchte mit einem Werkzeug für Gerätespezialisten (Python, FESA Navigator, ...) auf `TK1MU1` zugreifen. Woher weiß ich überhaupt, welche BPIDs ich zur Indizierung der Daten nehmen muss? Der Operateur kann mir sicher die BPCID sagen. Damit kann ich mit Hilfe der in [1], Kapitel *Sollwerte an das Gerät schicken* vorgeschlagenen Funktion

```
vector<BPID> bp = getBp("TK1MU1", bpcId);
```

die BPIDs für die Produktionskette herausbekommen. Die Funktion würde in diesem Fall die BPIDs x, y zurückliefern in der zeitlich korrekten Reihenfolge, was bedeuten würde: Sollwert für x setzen, Istwert für y lesen. Ja, man muss das Gerätemodell kennen, um das so interpretieren zu können. Siehe dazu auch Anhang A.

5. Ich kenne den Istwert von `TK1MU1`, sowie seinen Multiplexing Context (seine Stamps) und somit die zugehörige BPID. Woher weiß ich, dass dieser Istwert $I(y)$ zum Sollwert $S(x)$ gehört? Dazu könnte die in [1], Kapitel *Istwerte vom Gerät lesen* vorgeschlagene Funktion

```
vector<BPID> bp = getBpsOfBpc("TK1MU1", y);
```

dienen. Die würde in diesem Fall ebenfalls die BPIDs x, y in der zeitlich korrekten Reihenfolge zurückliefern.

2.1.2. BPCID

1. Der Magnet bekommt den Sollwert S und liefert den Istwert I für BPCID x .
2. Die Property SETTING bekommt den Sollwerte S und die Property ACQUISITION liefert den Istwert I für BPCID x .
3. Das entspricht dem Vorgehen mit VrtAccs im bestehenden Kontrollsystem.

2.2. Elektrostatisches Septum

Dieser Punkt erledigt auch die *Use-Case-Nummer U-015*.

2.2.1. BPID

1. Das Septum bekommt den Sollwert S1 für BPID u und liefert den Istwert I1 für BPID v . Und es bekommt den Sollwert S2 für BPID w und liefert den Istwert I2 für BPID x . Für S1/I1 und S2/I2 gibt es jeweils eine eigene Property. Die BPIDs kommen über das Timing in der zeitlichen Reihenfolge u, v, w, x .
2. Wie kann ich der Property SETTING mitteilen, dass Sollwert S1 für BPID u und Sollwert S2 für BPID w gilt? *Das geht in FESA bisher nicht!* !?
3. Wie kann ich der Property ACQUISITION mitteilen, dass ich den Istwert I1 für BPID v und den Istwert I2 für BPID x haben möchte? *Das geht in FESA bisher nicht!* !?

FESA erlaubt es allerdings, dass einzelne Felder einer Istwert-Property jeweils einen eigenen Kontext, also Timestamp, Eventstamp usw., haben. Damit könnte I1 mit Kontext K1 und I2 mit K2 zurückgeliefert werden.

4. Woher weiß ich, dass der Istwert I1(v) zum Sollwert S1(u) und der Istwert I2(x) zum Sollwert S2(w) gehört? Dazu könnte man wiederum die Funktion

```
vector<BPID> bp = getBpsInBpc("S12ME1I", v);
```

nutzen (siehe Kapitel 2.1.1, Punkt 5). Die würde in diesem Fall die BPIDs u, v, w, x in der zeitlich korrekten Reihenfolge zurückliefern, was bedeuten würde: Sollwert 1 für u setzen, Istwert 1 für v lesen, Sollwert 2 für w setzen und Istwert 2 für x lesen. Auch hier muss man das Gerätemodell kennen, um das so interpretieren zu können.

2.2.2. BPCID

1. Das Septum bekommt die Sollwerte S1, S2 und liefert die Istwert I1, I2 für die BPCID x .
2. Die Property SETTING bekommt die Sollwerte S1, S2 und die Property ACQUISITION liefert die Istwerte I1, I2 für BPCID x .
3. Das entspricht dem Vorgehen mit VrtAccs im bestehenden Kontrollsystem.

3. MMI ins SIS18

*Use-Case-Nummer U-002, erster Teil.*¹

3.1. Gepulster Magnet

Zur Zeit wird für eine MMI der Sollwert eines gepulsten Magneten nur *einmal* für alle Injektionen ins SIS18 gesetzt und der Istwert nur *einmal*, bei der ersten Injektion, gelesen. Ansonsten wird nur die Anzahl der MMIs (im Istwert-EQM) mitgezählt.

?

Wenn ich das richtig sehe, dann kommt für eine MMI das Sollwertsetz-Event im SIS-Timing 1-mal und das Istwertlese-Event n-mal.

3.1.1. BPID

1. Für die derzeitige Realisierung (1-mal Sollwert setzen, 1-mal Istwert lesen) gilt zunächst das in Kapitel 2.1.1 Gesagte.
2. Möchte man die Istwerte *aller* Injektionen einer MMI speichern, gibt es zwei Lösungsmöglichkeiten.
 - a) Alle Injektionen einer MMI werden mit dem selben Timing-Event² und damit mit der selben BPID ausgeführt. Dann muss die Geräte-Software so gestaltet sein, dass sie alle Istwerte speichern kann, ohne ältere Werte zu überschreiben (Stichwort State Machine).

!?

Das wird nicht mit dem FESA-Istwert-Ringpuffer gehen, wenn man die Istwerte *aller* Injektionen einer MMI mit *einem* Property-Zugriff lesen möchte. Dann wird man ein zweidimensionales Istwert-Array brauchen, das

¹Laut Petra Schütt (13. Feb. 2012) wird es keine zweifache Extraktion aus SIS100 oder SIS300 geben. Damit entfällt *U-002, zweiter Teil*.

²Mit Timing-Event ist das komplette Event-Pattern gemeint. Es enthält u. a. die Event-Nummer und die VrtAcc-Nummer im alten System bzw. die BPID und die BPCID im neuen System.

zunächst mit der BPID und dann noch mit der (von der GSW mitgezählten) MMI-Nummer indiziert wird.

- b) Jedes Timing-Event, das das Istwert lesen triggert, hat eine eigene BPID (die Event-Nummer bleibt natürlich gleich).

Das beeinflusst die State Maschine, falls man eine implementiert hat. Nach einem Sollwertsetz-Event können, im Falle von MMI, mehrere Istwertlese-Events kommen.

- i. Was würde in diesem Fall die Funktion

```
vector<BPID> bp = getBp("TK1MU1", bpcId);
```

liefern? Für fünf MMIs bekäme man z. B. u, v, w, x, y, z in der zeitlich korrekten Reihenfolge zurück: Sollwert für BPID u setzen, Istwert der MMI 1 mit v lesen, Istwert der MMI 2 mit w lesen usw.

- ii. Was machen Geräte, die auch auf das Istwertlese-Event hören, aber nicht an der MMI beteiligt sind? Geht nicht! Die müssen entweder auf ein anderes Event hören, oder deren GSW muss die MMI berücksichtigen. ?

3.1.2. BPCID

1. Für die derzeitige Realisierung (1-mal Sollwert setzen, 1-mal Istwert lesen) gilt zunächst das in Kapitel 2.1.2 Gesagte.
2. Möchte man die Istwerte *aller* Injektionen einer MMI speichern, muss die Geräte-Software so gestaltet sein, dass sie alle Istwerte speichern kann, ohne ältere Werte zu überschreiben. Alle Events kommen nämlich mit der selben BPCID.

Auch das wird, wie bei der BPID, nicht mit dem FESA-Istwert-Ringpuffer gehen, wenn man die Istwerte aller Injektionen einer MMI mit einem Property-Zugriff lesen möchte. Man müsste wieder mit einem zweidimensionalen Istwert-Array arbeiten. Siehe Punkt 2a in Kapitel 3.1.1. !?

3.2. Elektrostatisches Septum

Die derzeitige Behandlung eines Septums ist analog zu der in Kapitel 3.1 beschrieben, weil nur *ein* Sollwert und der dazugehörige Istwert MMI-relevant sind.

Ist ein Gerät denkbar, dass zwei oder mehrere Sollwerte und die entsprechenden Istwerte hat, die MMI-relevant sind und die zudem in verschiedenen BPs realisiert bzw. gelesen werden müssen? Wenn ja, muss man hier noch mal denken! ?

3.3. Beam Production Chain Timestamp

Die BPCTS ist in Anhang B so beschrieben: Die BPCTS *sagt, wann die Beam Production Chain gestartet wurde, also in etwa der Zeitpunkt, wann die Reise der Ionen an der Ionenquelle begann*. Bei der MMI werden aber mehrere Unilaczyklen in einen SIS18-Zyklus injiziert. Welche BPCTS haben die Unilac-Zyklen einer MMI? Die Antwort hat Auswirkungen auf die Implementierung von State Machines. Siehe Kapitel 10.2.

Ein vergleichbares Problem hat man bei SIS18-Boosterzyklen. Siehe Kapitel 4.3.

Antwort: *Alle MMI-Zyklen einer Injektion vom Unilac ins SIS18 haben die selbe BPCTS!*

4. SIS18 Booster Mode

Use-Case-Nummer U-003.

Das SIS18 injiziert 4 Beams in das SIS100. Im SIS18 hat Beam 1 andere Eigenschaften als die Beams 2 bis 4.

4.1. Gepulster Magnet

4.1.1. BPID

1. Magnet bekommt den Sollwert $S1$ für BPID u und liefert den Istwert $I1$ für BPID v . Und er bekommt den Sollwert $S2$ für BPID w und liefert den Istwert $I2$ für BPID x .
 - a) Die BPIDs im SIS18 für die 4 Füllungen ins SIS100 kommen in der zeitlichen Reihenfolge u, v, w, x, w, x, w, x .
 - b) Auf der Realtime-Ebene wird der Sollwert $S2$ dreimal zur Hardware geschickt, nämlich für die Beams 2, 3 und 4. Entsprechend wird der Istwert dreimal gelesen.
 - c) Nutzt die Realtime-Ebene Wechselpuffer, dann wird $I2_n$ von $I2_{n+1}$ überschrieben! Das Subskript n steht für die Beams 2 bis 4.
 - d) Nutzt die Realtime-Ebene einen Ringpuffer für die Istwerte, bleiben zunächst alle $I2_n$ erhalten (bis die Füllung des Ringpuffers einmal rum ist). Wie man (via FESA) an ältere Werte drankommt und wie man sicherstellen kann, dass man genau die Werte der Beams 2 bis 4 aus *einer* MMI bekommt, ist (noch) nicht klar.

?

2. Die Property SETTING bekommt den Sollwert S1 für BPID u bzw. den Sollwert S2 für BPID w .

3. Die Property ACQUISITION liefert den Istwert I1 für BPID v und, je nach Realisierung,

a) entweder einen Istwert $I2_n$ für BPID x oder

b) 3 Istwerte $I2_2, I2_3, I2_4$ für BPID x (wobei hier die Anzahl der Istwerte je nach Anzahl der Boosterzyklen variiert).

4. Woher weiß ich, welche BPCIDs ich zur Indizierung der Daten nehmen muss? Der Operateur kann mir wieder die BPCID nennen. Die bereits bekannten Funktion

```
vector<BPID> bp = getBp("TK1MU1", bpcId);
```

würde mir etwa u, v, w, x zurückliefern. Das sind aber BPCIDs für *zwei* Zyklen, nämlich u, v für Soll-/Istwert des Beams 1 und w, x für die Soll-/Istwerte der Beams 2 bis 4, da alle zur selben BPCID gehören. Die Funktion könnte aber genau so gut w, x, u, v liefern. Damit könnte ich nur davon ausgehen (wenn ich das Gerätermodell kenne), dass u, v bzw. w, x in der zeitlich korrekten Reihenfolge sind. Die Reihenfolge der beiden Zyklen kann womöglich nur zufällig sein.

5. Woher weiß ich, dass der Istwert $I1(v)$ zum Sollwert $S1(u)$ gehört? Dito $I2_n(x)$ und $S2(w)$? Im Prinzip wieder mit der Funktion

```
vector<BPID> bp = getBpsInBpc("TK1MU1", v);
```

und den gleichen Problemen wie im vorhergehenden Punkt.

!?

!?

4.1.2. BPCID

1. Da der Boostermode des SIS18 für *eine* Produktionskette arbeitet, wird es auch nur *eine* BPCID geben. Stimmt das?

2. Die 4 Boosterzyklen des SIS18 sind *nicht* identisch. TK1MU1 braucht für den Beam 1 eine andere Einstellung (Sollwert S1) als für die Beams 2 bis 4 (Sollwert S2). Damit kann eine *einzelne* BPCID, die für alle Boosterzyklen gilt, *nicht* als Index für die Daten benutzt werden.

?

!?

4.2. Elektrostatisches Septum

4.2.1. BPID

1. Im Prinzip gilt für diesen Fall das in Kapitel 4.1.1 Gesagte, mit dem Unterschied, dass das Septum je 2 Soll- und 2 Istwerte für den Beam 1 und die Beams 2 bis 4 hat.

2. Die Funktion

```
vector<BPID> bp = getBp("S12ME1I", bpcId);
```

würde dann schon leicht unübersichtliche *acht* verschiedene BPIDs liefern, nämlich s, t, u, v, w, x, y, z mit: BPID s ist Index für Sollwert S1 für Beam 1; t ist Index für Istwert I1 für Beam 1; u ist für S2, Beam 1; v für I2, Beam 1; w für S1, Beam 2; x für I1, Beam 2; y für S2, Beam 2; z für I2, Beam 2. Die Funktion könnte aber genau so gut w, x, y, z, s, t, u, v liefern.

Hätte *jeder* der Beams 1 bis 4 andere Einstellungen, dann würde die Funktion *16* verschiedene BPIDs zurückliefern.

3. Und dito

```
vector<BPID> bp = getBpsInBpc("S12ME1I", v);
```

mit den gleichen Problemen.

4.2.2. BPCID

1. Da der Boostermode des SIS18 für *eine* Produktionskette arbeitet, wird es auch nur *eine* BPCID geben. Stimmt das?
2. Die 4 Boosterzyklen des SIS18 sind *nicht* identisch. S12ME1I braucht für den Beam 1 eine andere Einstellung (Sollwerte S1, S2) als für die Beams 2 bis 4 (Sollwert S3, S4). Damit kann eine *einzelne* BPCID, die für alle Boosterzyklen gilt, *nicht* als Index für die Daten benutzt werden.

4.3. Beam Production Chain Timestamp

Die BPCTS ist in Anhang B so beschrieben: Die BPCTS *sagt, wann die Beam Production Chain gestartet wurde, also in etwa der Zeitpunkt, wann die Reise der Ionen an der Ionenquelle begann*. Im Beispiel für den Booster-Mode begannen aber *viermal* Ionen ihre Reise in einen SIS100-Zyklus. Welche BPCTS haben die vier SIS18-Boosterzyklen? Die Antwort hat Auswirkungen auf die Implementierung von State Machines. Siehe Kapitel 10.2.

Ein vergleichbares Problem hat man bei der MMI vom Unilac ins SIS18. Siehe Kapitel 3.3.

Antwort: *Alle* Booster-Zyklen *einer* Injektion vom SIS18 ins SIS100 haben die *selbe* BPCTS!

5. Gleicher Beam aus SIS18 aber verschiedene Targets

Use-Case-Nummer U-022.

Cave X und Cave Y werden, eventuell über weitere Beschleuniger, mit Strahl versorgt. Für beide Targets ist es erforderlich, dass im SIS18 U^{73+} auf 123,4 MeV/n beschleunigt wird. Durch die nachfolgend betrachteten Geräte fliegen *beide* Strahlen.

5.1. Gepulster Magnet

5.1.1. BPID

1. Es gilt zunächst das in Kapitel 2.1.1 gesagte.
2. Vorstellbar sind dann zwei Szenarien:
 - a) Der Magnet hat für den Strahl nach Cave X die BPIDs v für den Sollwert und w für den Istwert, sowie für den Strahl nach Cave Y die BPIDs x für den Sollwert und y für den Istwert. Das sind also komplett getrennte Einstellungen. Die Sollwerte für v und x sind aber identisch.

Möchte man eine Strahlmanipulation vornehmen, die für beide Strahlen gilt, muss man das *zweimal* machen, also den Magneten zweimal mit einem neuen (identischen) Sollwert versorgen (für die BPIDs v und x).

- b) Der Magnet hat für *beide* Strahlen die BPIDs x für den Sollwert und y für den Istwert (nur die BPCIDs sind unterschiedlich).

Eine Strahlmanipulation wirkt automatisch auf beide Strahlen.

Eine echte Energieumstellung für nur einen der beiden Strahlen, z. B. von 123,4 MeV/n auf 234,5 MeV/n, geht nur mit der Einführung weiterer BPIDs und der Versorgung aller betroffenen Geräte mit Sollwerten für diese BPIDs.

5.1.2. BPCID

Da der Strahl nach Cave X eine andere BPCID haben wird als der Stahl nach Cave Y, gilt für jeden der beiden Strahlen genau das in Kapitel 2.1.2 gesagte. Die Einstellungen des Magneten für die beiden Strahlen sind komplett unabhängig voneinander.

5.2. Elektrostatisches Septum

5.2.1. BPID

Es gilt das in Kapitel 2.2.1 gesagte, einschließlich der noch ungelösten Probleme bezüglich der Properties SETTING und ACQUISITION. Die Ergänzungen des Kapitels 5.1.1 gelten im Prinzip auch für das Septum.

5.2.2. BPCID

Da der Strahl nach Cave X eine andere BPCID haben wird als der Stahl nach Cave Y, gilt für jeden der beiden Strahlen genau das in Kapitel 2.2.2 gesagte. Die Einstellungen des Septums für die beiden Strahlen sind komplett unabhängig voneinander.

6. Soll/Ist-Vergleich

Use-Case-Nummer U-025.

Punkt 2 aus den Use-Cases, nämlich eine Regelung über zwei Geräte, wird hier noch nicht behandelt.

6.1. Gepulster Magnet

6.1.1. BPID

1. Siehe zunächst Kapitel 2.1.1.
2. Auf der Realtime-Ebene gibt es bei gepulsten Magneten am Ende des Zyklus noch ein sogenanntes Aufräumen. Das würde in einer RT-Action während des Beamprozesses mit der BPID z passieren. Wenn ich darin z. B. einen Soll/Ist-Vergleich machen wollte, müsste ich auch auf der Realtime-Ebene wissen, mit welchen BPIDs Soll- und Istwert im Speicher abgelegt sind.
3. Das gleiche Problem habe ich natürlich, wenn ich den Soll/Ist-Vergleich bereits in der RT-Action, die den Istwert liest, machen wollte: Darin kenne ich zwar die Istwert-BPID, aber mit welcher BPID muss ich den zu vergleichenden Sollwert indizieren?
4. Kann es nicht so gehen?

!?

?

- a) Man kann davon ausgehen, dass Sollwert setzen und Istwert lesen zeitlich immer aufeinander folgen. Dann merkt sich die RT-Action zum Sollwert setzen den gerade ans Gerät geschickten Sollwert in einem Speicher unabhängig von der BPID. Später vergleicht die RT-Action zum Istwert lesen den gerade gelesenen Istwert mit dem gemerkten Sollwert.
- b) Das geht entsprechend auch in einer RT-Action zum Aufräumen, wenn sich die beiden RT-Actions vorher Sollwert *und* Istwert gemerkt haben.
- c) Eine State Machine (siehe Punkt 1 in Kapitel 10.2) kann (in der Regel) sicher stellen, dass Sollwert und Istwert tatsächlich zusammen gehören.

6.1.2. BPCID

Da alle RT-Actions mit der selben BPCID aufgerufen werden, gibt es keine Probleme die zu vergleichenden Soll- und Istwerte zu indizieren.

6.2. Elektrostatisches Septum

6.2.1. BPID

Auch hier habe ich das in Kapitel 6.1.1 erwähnte Problem, dass ich in der Istwert-RT-Action zwar die Istwert-BPID kenne, aber nicht weiß, mit welcher BPID ich den zu vergleichenden Sollwert indizieren muss. !?

Eine Lösung wäre auch hier die in Punkt 4, Kapitel 6.1.1 vorgeschlagene.

6.2.2. BPCID

Da alle RT-Actions mit der selben BPCID aufgerufen werden, gibt es keine Probleme die zu vergleichenden Soll- und Istwerte zu indizieren.

7. Alternativen

Use-Case-Nummern U-006, U-012, U-023.

Während eines laufenden Zyklus' wird entschieden, dass der Standardzyklus abgebrochen und statt dessen mit der Alternative fortgefahren wird.

Ein Beispiel wäre, dass tatsächlich ein alternativer Strahlweg ausgewählt wird. Der Weg des Strahls wäre dann z. B. nicht mehr Quelle → SIS18 → SIS100 → Cave X

sondern Quelle \rightarrow SIS18 \rightarrow Cave Y. Und der Abzweig zum Cave Y wäre in der Transferstrecke zwischen SIS18 und SIS100 am Magneten MU³.

Eine weiteres Beispiel wären alternative Downramps am Ende eines Zyklus je nach dem Injektionslevel des folgendem Zyklus.

7.1. Gepulster Magnet

Eigentlich wie Kapitel 2.1. Der Magnet muss eben nur spät genug seinen Sollwert bekommen, also *nach* der Entscheidung, ob es dahin oder alternativ dorthin gehen soll.

7.1.1. BPID

Der Magnet braucht natürlich *zwei* Sollwertsätze, einen Satz für den Standard, einen für die Alternative. Der Standard-Sollwert wird mit der BPID s , der Alternativ-Sollwert mit der BPID a indiziert.

Wird der Standardzyklus über das General Machine Timing (GMT, neues Timing-System) gefahren, trägt das für gepulste Magnete relevante Sollwertsetz-Event die Event-Nummer e und die BPID s . Beim Alternativzyklus ist es die Event-Nummer e und die BPID a . Die Event-Nummer ist also in beiden Fällen die gleiche, weil die FESA-RT-Action zum Sollwert setzen an ein bestimmtes Event, hier eben an e , gekoppelt ist.

! Zu beachten ist, dass natürlich *alle* Geräte einer Maschine⁴ das *selbe* Event hören. Im Alternativfall reagieren also mindestens *alle* gepulsten Magnete auf das Event mit der BPID a ! Damit müssen *alle* Magnete einen zweiten, alternativen Datensatz haben.

Angenommen, die Alternative im obigen Beispiel bedeutet, dass der Strahl mit den selben Eigenschaften einfach nicht nach Cave X sondern nach Cave Y soll. Dann heißt das:

1. Alle gepulsten Magnete nach dem SIS18 und vor dem Magneten MU haben für die BPIDs s und a zwei Datensätze, die identisch sind.
2. MU hat zwei unterschiedliche Datensätze für die BPIDs s und a .
3. Alle gepulsten Magnete nach MU bis zum SIS100 benötigen eigentlich nur den Datensatz für die BPID s , weil mit a kein Strahl durch sie fliegt. Sie hören aber auch auf das Event mit der BPID a . Was sollen sie damit tun?

³Im FAIR gibt es nur noch MH und MV

⁴Bisher hätte man Timing-Domain gesagt. Das legt aber mehrere Timing-Master nahe, was wir ja nicht haben werden.

4. Alle gepulsten Magnete nach MU bis zum Cave Y benötigen eigentlich nur den Datensatz für die BPID a , weil mit s kein Strahl durch sie fliegt. Sie hören aber auch auf das Event mit der BPID s . Was sollen sie damit tun?

?

Die Frage „Was sollen sie damit tun?“ könnte man vielleicht mit „Magnet für s bzw. a deaktivieren“ beantworten. Das würde aber erfordern, dass Events des Standardzyklus eine andere BPCID tragen als die Events des Alternativzyklus, zumindest solange sich eine mögliche Property ACTIVE auf die BPCID bezieht. Siehe dazu die Kapitel 7.3 und 10.2.

7.1.2. BPCID

Indizieren mit BPCID geht nur dann, wenn die Alternative eine *eigene* BPCID hat. Siehe dazu Kapitel 7.3. Dann gälte das im vorigen Kapitel Gesagte auch für diesen Fall.

7.2. Elektrostatisches Septum

Ein Septum ist hier ein ungeeignetes Beispiel, weil es an der Injektion oder der Extraktion beteiligt ist und damit alle Soll- und Istwerte entweder komplett zum Standardzyklus oder komplett zur Alternative gehören.

Betrachten wir also ein erfundenes „Octum“, das die selben Soll- und Istwerte hat wie das Septum, dessen zeitlich früherer Soll-/Istwertsatz (mit u und v indiziert) aber keine Alternative kennt, dessen späterer Soll-/Istwertsatz (mit w und x indiziert) aber an der Alternative beteiligt ist.

Für eine Alternative muss das Octum also mit *zusätzlichen* Sollwerten versorgt werden und es liefert entsprechend zusätzliche Istwerte.

7.2.1. BPID

Damit ist es vorstellbar, dass für das Octum BPIDs in einem Zyklus sowohl in der Reihenfolge u, v, w_s, x_s im Falle des Standardzyklus, als auch in der Reihenfolge u, v, w_a, x_a im Falle des Alternativzyklus auftreten können (siehe Kapitel 2.2.1). Es gibt also insgesamt sechs verschiedene BPIDs.

Zur Datenversorgung würde man zwei Properties zur Versorgung der Sollwerte haben. Der zeitlich frühe Sollwert wird mit einem Datensatz versorgt, der zeitlich späte mit zweien, für den Standard- und den Alternativzyklus. Dann gilt für die beiden 'späten' Datensätze das in Kapitel 7.1.1 Gesagte entsprechend.

Hinweis: Das geht nur, wenn das Octum-Netzgerät es erlaubt, dass der Sollwert für w unabhängig vom Sollwert für u eingestellt werden kann.⁵

7.2.2. BPCID

Indizieren mit BPCID geht nur dann, wenn die Alternative eine *eigene* BPCID hat. Siehe dazu Kapitel 7.3. Dann gälte das im vorigen Kapitel Gesagte auch für diesen Fall.

7.3. BPCID und alternative Produktionskette

Die normale Kette sei Quelle \rightarrow SIS18 \rightarrow SIS100 \rightarrow Cave X. Die bekommt eine BPCID. Die alternative Kette sei Quelle \rightarrow SIS18 \rightarrow Cave Y⁶. Der Abzweig zum Cave Y sei in der Transferstrecke zwischen SIS18 und SIS100 ab dem Magneten MU.

?

Welche BPCID hat die Kette MU \rightarrow Cave Y?

Es gibt prinzipiell drei Möglichkeiten, wobei aber nur zwei praktikabel sind:

1. Die (Teil-) Kette MU \rightarrow Cave Y hat die selbe BPCID wie die normale Kette.

Korrelation von Daten von Geräten entlang *beider* Ketten ist mit Hilfe der BPCID problemlos möglich.

2. Die (Teil-) Kette MU \rightarrow Cave Y hat eine eigene BPCID. Bezüglich der verschickten Events heißt das aber, dass die BPCID mitten im Zyklus wechselt, wenn die Alternative gefahren wird.

Korrelation von Daten von Geräten entlang der Kette Quelle \rightarrow SIS18 \rightarrow Cave Y ist problematisch, weil die BPCID am Magneten MU wechselt.

3. Die komplette Kette Quelle \rightarrow SIS18 \rightarrow Cave Y hat eine eigene BPCID.

Das geht gar nicht, weil man womöglich erst zum Zeitpunkt SIS18-Flattop entscheiden will, ob es nach Cave X oder nach Cave Y weiter geht. Zu diesem Zeitpunkt ist der Strahl aber schon durch die davor liegenden Strahlzweige und Beschleuniger hindurch. Die (Teil-) Kette Quelle \rightarrow MU kann also nur *eine* BPCID haben.

Wahrscheinlich muss man beim Nachdenken über Beam Production Chains (BPC/Kette) noch sauberer trennen zwischen

1. der räumlichen Sicht, also dem Strahlrohr entlang dessen sich der Strahl bewegt, und

⁵Ich denke da an den Kicker und das Problem, dass ein Sollwert S1 kleiner ist als der (womöglich zeitlich vorhergehende) halbe Sollwert S2.

⁶Wobei Cave Y auch ein Dump sein könnte.

2. der zeitlichen Sicht, also dem Zyklus bzw. den Zyklen, die der Strahl durchläuft.

8. Gerät bedient mehrere Maschinen

1. Die Anforderung hat man nur bei Geräten in Strahltransportstrecken. Richtig? Wenn diese eigene, von den Beschleunigern unabhängige Maschinenkennungen spendiert bekommen, stellt sich das Problem nicht.
2. Ein Gerät hört auf mehrere Maschinenkennungen. So könnte z. B. HTCMU1 auf SIS- *und* ESR-Timing hören, weil der Strahl mal vom SIS und mal vom ESR kommen kann.

BPID bzw. BPCID müssen eindeutig sein, dürfen also nicht gleichzeitig im SIS-Timing *und* im ESR-Timing vorkommen. Ansonsten müsste man die Daten im Front-end zusätzlich mit der Maschinenkennung indizieren.

Das Event, das die RT-Action zum Sollwert setzen triggert, sollte in der SIS- und der ESR-Maschine das selbe sein (muss aber nicht unbedingt).

9. Therapiebetrieb

Use-Case-Nummer U-009.

9.1. Gepulster Magnet

9.1.1. BPID

Jede EFI-Kombination wird mit eigenen BPIDs realisiert. Es gilt das in Kapitel 2.1.1 Gesagte.

9.1.2. BPCID

Jede EFI-Kombination wird mit einer eigenen BPCID realisiert. Dann gilt das in Kapitel 2.1.2 Gesagte.

Ob das allerdings Sinn macht, sei mal dahingestellt.

9.2. Elektrostatisches Septum

9.2.1. BPID

Jede EFI-Kombination wird mit eigenen BPIDs realisiert. Es gilt das in Kapitel 2.2.1 Gesagte.

!?

Damit hat man natürlich auch die Probleme mit den Properties SETTING und ACQUISITION. Siehe die Punkte 2 und 3 im genannten Kapitel.

9.2.2. BPCID

Jede EFI-Kombination wird mit einer eigenen BPCID realisiert. Dann gilt das in Kapitel 2.2.2 Gesagte.

?

Wie gesagt, ob das Sinn macht, ist die Frage.

10. Allgemeines

10.1. Commit Event

[Bisher nur Stichworte:]

1. Commit-Event zeitversetzt entlang der Kette für die verschiedenen Beschleuniger.
2. Commit-Event mit Transaktionsnummer (TAN).
3. Commit-Event mit BPCID als Kennung? Ansonsten müssten viele BPIDs mitgeschickt werden.

?

?

Wenn die Daten mit BPID indiziert sind, wie kann ich mit der BPCID des Commit-Events auf die korrekten Daten kommen? Die Daten selbst wurden ja schon zusammen mit einer BPID geschickt und im Zwischenpuffer abgelegt. Brauche ich dann die BPCID überhaupt noch? Reicht nicht einfach die TAN?

?

4. Gibt es mehrere Zwischenpuffer für Sollwerte, die auf ihr Commit warten oder nur einen? Wenn es mehrere gibt, wie werden sie unterschieden?

10.2. State Machine

1. Nahezu alle am GSI-Timing teilnehmenden Gerätemodelle implementieren in ihrer RT-Software (den EQMs) eine *State Machine*. Diese erlaubt es u. a. festzustellen, wenn Events *innerhalb eines VrtAcc*, also innerhalb eines Zyklus, fehlen oder in einer falschen Reihenfolge kommen, was zu einem *Sequence Error* führt. Man stellt damit nicht nur (selten vorkommende) fehlerhafte Programmierungen der Pulszentrale fest, sondern z. B. auch defekte TIFs oder Timing-Receiver. Das funktioniert, weil sich die VrtAcc-Nummer innerhalb eines Zyklus' nicht ändert.

In Zukunft wird es innerhalb eines Zyklus' mehrere BPIDs geben. Damit ist es zunächst nicht mehr möglich zu entscheiden, ob das Istwertlese-Event mit der BPID y zum vorhergegangenen Sollwertsetz-Event mit der BPID x gehört oder nicht. Die RT-Actions können also ohne weitere Informationen über den Zyklus Sequence Errors nicht, oder nur sehr eingeschränkt, erkennen.

Kann es beispielsweise nicht so gehen?

- a) Mit dem Sollwertsetz-Event kommt u. a. die BPCID mit. Die merke ich mir.
- b) Mit dem Istwertlese-Event kommt auch die BPCID mit. Wenn diese mit der gemerkten übereinstimmt, bin ich noch im selben Zyklus und kann die State Machine weiter abarbeiten.

Das geht nicht ganz korrekt, wenn zwei Zyklen der selben BPCID direkt aufeinander folgen, oder im Booster Mode (siehe Kapitel 4.1.1), wenn alle SIS18-Boosterzyklen die selbe BPCID haben. Aber dieses Loch gibt es schon im bestehenden System.

Statt der BPCID könnte man vielleicht auch die BPCTS (siehe Anhang B) nehmen. Die ist eindeutiger, da sie sich in der Regel mit jedem Zyklus ändert. Zwei aufeinander folgende Zyklen mit der selben BPCID haben zwei unterschiedliche BPCTSs. Das sollte auch für Alternativen (siehe Kapitel 7) gelten, solange Standard- und Alternativzyklus die selbe BPCTS haben.

Ausnahmen bilden die MMI vom Unilac ins SIS18 und die Booster-Zyklen bei der Injektion vom SIS18 ins SIS100.⁷ Dort haben alle MMIs und alle Booster-Zyklen einer Injektion die *selbe* BPCTS.

2. [tbs] Verschachtelte Zyklen im Unilac bei Hoch- $B\rho$ -Betrieb.

⁷Wo gibt es noch ähnliche Injektionen oder Betriebsarten?

10.3. Property ACTIVE

1. Geräte, die am GSI-Timing teilnehmen, kann man für einen virtuellen Beschleuniger (einen VrtAcc, also einen bestimmten Zyklus) aktivieren und deaktivieren. Das müsste man im GMT mit der BPCID machen können.⁸
2. Hat ein Gerät eine State Machine, muss man darauf achten, dass nur dann (de-)aktiviert wird, wenn die State Machine in ihrem Ausgangszustand ist. Das wurde bisher im Gap zwischen zwei Zyklen realisiert.

FESA (-Core) guckt *vor* dem Aufruf einer RT-Action, ob im (Sollwert-) Wechsellpuffer neue Daten vorliegen und stellt dann diese der RT-Action bereit. Wenn man das mit dem ACTIVE-Sollwert und einem Timing-Event vergleichbar dem heutigen Evt-Start-Cycle macht, bräuchte man wahrscheinlich kein Gap dafür.

Muss der echte Sollwert aber auch schon mit Evt-Start-Cycle gesetzt werden, müsste man das (De-) Aktivieren womöglich ans Ende des Zyklus setzen. Damit bekommt man aber noch einen aktiven Zyklus mehr als gewünscht. Das ist unschön, wenn z.B. eine BPC nur alle 3 Minuten läuft, ich in der ersten Minute ein Gerät deaktiviere und dann nach zwei Minuten das Gerät doch noch einmal einen Sollwert erhält.⁹

10.4. Istwerte aus mehreren BPs

[Bisher nur Stichworte:]

Siehe Juttas Email vom 25.7.12

Drei Möglichkeiten Istwerte zu lesen von einem (Diagnose-) Gerät, das in den BPs 1 bis 5 gemessen hat:

1. ISTWERT(bpid)
2. ISTWERT(bpid1, bpid2, bpid3)
3. ISTWERT(bpcid)

Erfordert eine Property ISTWERT, die

1. eine variable Anzahl von (Daten-) Indizes erlaubt;

⁸Allerdings sollte man bedenken, dass 'inaktiv' z. B. bei gepulsten Magneten ziemlich irreführend ist. Es heißt nämlich nicht, dass der Magnet den Strahl nicht beeinflusst. Der Magnet bekommt nur keinen Sollwert geschickt in dem VrtAcc, für den er deaktiviert ist. D. h., dass er den zuletzt geschickten Sollwert behält. Und der beeinflusst den Strahl sehr wohl.

⁹Ansätze einer Lösung dieses Problems in FESA habe ich schon mit Alex diskutiert. Man könnte z. B. aktivieren und Sollwert setzen an das selbe Event hängen, das Sollwert setzen aber mit einem minimalen Delay versehen. Das würde garantieren, dass das (De-)Aktivieren immer *vor* dem Sollwert setzen kommt. Das muss aber alles noch genauer durchdacht werden.

2. zwischen BPID und BPCID als Index unterscheiden kann;
3. eine variable Anzahl vom Messwerten zurückliefern kann;
4. die Messwerte jeweils passend mit dem korrekten Kontext (der korrekten BPID usw.) kennzeichnen kann.

Zu beachten ist allerdings, dass das (wahrscheinlich) nur geht, wenn die verschiedenen Messungen in den BPs 1 bis 5 jeweils mit einem eigenen Event gestartet wurden. Eine Messung über fünf aufeinander folgende BPs, die mit *einem* Event zu Beginn des ersten BPs gestartet wurde, wird in der Regel auch nur *einen* Kontext, nämlich den aus BPID 1, zurückliefern können.

10.5. Positionierbare Pressluftantriebe

[Bisher nur Stichworte:]

Gerätemodell PPOS.

1. Zwei Events im Zyklus zugeordnet. Mit zeitlich früherem Event reinfahren, mit späterem rausfahren (oder auch umgekehrt).
2. Die Events zum Rein- und Rausfahren werden als Sollwert via Property ans Gerät geschickt. Also keine feste Zuordnung Event - RT-Action.

11. Zusammenfassung: Wo klemmt's?

11.1. Was geht (bisher) nicht mit BPID?

1. Property SETTING mit Sollwerten für verschiedene BPIDs. Siehe Kapitel 2.2.1 und 9.2.1.
2. Property ACQUISITION mit Soll- und Istwerten aus verschiedenen BPIDs. Siehe Kapitel 2.2.1 und 9.2.1.

11.2. Was geht (dann) doch mit BPID?

1. Commit-Event mit BPCID und Zuordnung zu mit BPID indizierten Sollwerten. Siehe Kapitel 10.1.
2. Soll/Ist-Vergleich, wenn Sollwert und Istwert mit unterschiedlichen BPIDs indiziert werden und man nur via BPID indizieren möchte/kann. Lösungsvorschlag siehe Kapitel 6.1.1 und 6.2.1.

3. Feststellen von Sequence Errors mit Hilfe einer State Machine. Es geht wahrscheinlich mit der BPCID. Siehe Punkt 1 in Kapitel 2.1.1.

11.3. Was geht (bisher) nicht mit BPCID?

1. Boosterzyklen im SIS18 mit unterschiedlichen Sollwerten für die einzelnen Zyklen. Alle Zyklen haben die selbe BPCID. Dito Istwerte. Siehe Kapitel 4.1.2 und 4.2.2.
2. Sollwerte für Standardzyklus und (zugehörigem) Alternativzyklus, wenn beide die selbe BPCID haben. Dito Istwerte. Siehe Kapitel 7 und Kapitel 7.3.

11.4. Was geht (bisher) allgemein nicht?

1. Lesen aller Istwerte einer MMI mit *einem* Zugriff, wenn FESA mit Istwert-Ringpuffern arbeitet. Siehe Kapitel 3.1.1 und 3.1.2.
2. Die Funktionen `getBp()` und `getBpcInBpc()` liefern bisher keine eindeutigen Informationen. Siehe u. a. die Kapitel 3.1.1, 4.1.1, 4.2.1 sowie Anhang A.

A. Funktion `getBp()`

Die Idee der Funktionen

```
vector<BPId> getBp(string nomen, BPCId bpcId);  
vector<BPId> getBpsInBpc(string nomen, BPId bpId);  
  
vector<BPId> getBp(MId machId, int eventNum, BPCId bpcId);  
  
vector<BPId> getBp(string nomen, string property, BPCId bpcId);  
vector<BPId> getBpsInBpc(string nomen, string property, BPId bpId);
```

ist noch nicht komplett ausgereift.

1. Bisher gehe ich davon aus, dass die Funktionen nur auf der Anwendungsebene realisiert werden, und man dort auf irgendeine intelligente Art und Weise an die benötigten Informationen kommt (LSA, Datenbank, ...). Eventuell könnten die Funktionen `getBp()` auch auf der FESA-Ebene benötigt werden, möglicherweise sogar in RT-Actions. Im Moment sehe ich allerdings keine konkrete Anforderung. Wie das zu realisieren wäre, ist mir auch noch komplett unklar.

!?

2. Ist es überhaupt möglich, die für ein Gerät *relevanten* Events und daraus die aktuell zugehörigen BPIDs zu erhalten? Welche Instanz im Kontrollsystem hat diese Information? Oder sind es mehrere Instanzen mit Teilinformationen, die entsprechend kombiniert werden müssen?
3. Kann ich davon ausgehen, dass die zurückgelieferten BPIDs in der zeitlich korrekten Reihenfolge sind? Wenn die Funktion für das Septum z. B. u, v, w, x liefert (siehe Kapitel 2.2.1), kann ich sicher sein, dass immer vier Werte geliefert werden und der dritte Wert (w) die BPID für den zweiten Sollwert ist?
4. Sollte die Funktion besser eine Kombination von BPIDs und Event-Nummern zurückliefern? Da die Event-Nummern an RT-Actions geknüpft sind, hätte ich darüber einen Zusammenhang zwischen BPID und RT-Action (z. B. Sollwert S1 setzen, Istwert I2 lesen) hergestellt.
5. Was ist mit alternativen Abläufen, z. B. um den Beam zu dumpen? Für diese Alternativen muss das Septum mit *zusätzlichen* Sollwerten versorgt werden und es liefert entsprechend zusätzliche Istwerte.

!?

Damit ist es vorstellbar, dass für das Septum sowohl die BPIDs u, v, w_1, x_1 als auch die BPIDs u, v, w_2, x_2 gelten, also insgesamt sechs verschiedene BPIDs. Was liefert in diesem Fall `getBp()`? Braucht `getBp()` einen weiteren Parameter, um den Normalfall und die Alternative unterscheiden zu können? Kann es mehr als eine Alternative geben?

Siehe dazu auch Kapitel 7.2.

6. Liefert `getBp()` eine Kombination von BPIDs und Event-Nummern zurück (Punkt 4), dann bekäme man im Falle von Alternativen (Punkt 5) sowohl für x_1 als auch für x_2 das selbe Event e zurückgeliefert.
7. Man könnte – eventuell ist das sinniger – auch folgende Funktionen deklarieren

```
vector<BPID> getBPs(string nomen, BPCID bpcId);
BPCID getBPC(string nomen, BPID bpId);
```

und obige `getBpsInBpc()` durch

```
vector<BPID> bpId getBPs("TK1MU1", getBPC(42));
```

implementieren.

8. Auch

```
vector<BPID> bpId getBp(SIS18, EVT_RAMP_START, 7);
```

(mit $7 == BPCID$) kann keine eindeutige BPID liefern, wenn das Event innerhalb einer Maschine mehrfach in der BPC vorkommt.

9. Und

```
vector<BPID> bpId getBp("S11MU2", "RAMPVALS", 7);
```

ebenso nicht.

B. Event-Format

Dieses Format eines Events wurde im Timing-Workshop im Dezember 2011 diskutiert.

Header

MID : Machine ID, eventuell unnötig

PID : Pattern ID, nicht mehr nötig! ???

tbs... : Andere Info. Vorschlag: Versionsnummer bzw. Message-Typ für Message Format

BPCID : Beam Production Chain ID

BPCTS : Beam Production Chain Timestamp. Sagt, wann die Beam Production Chain gestartet wurde, also in etwa der Zeitpunkt, wann die Reise der Ionen an der Ionenquelle begann.

BPID : Beam Process ID

EVTNO : Event-Number

Payload

Zeitstempel : 8 Bytes

Bei der Diskussion sollte man das Format, das die Geräte-Software, also die FESA-Klasse sieht, betrachten. Was über GMT-Netzwerk kommt ist hier mehr oder weniger uninteressant.

C. Gateway zwischen GMT und GSI-Timing

[Bisher nur Stichworte:]

Geplant ist ein Gateway zwischen General Machine Timing (GMT) für FAIR und altem GSI-Timing zur (online) Übersetzung der Timing-Events vom neuen ins alte System.

Alle BPIDs eines (z. B. SIS18-) Zyklus im GMT müssen in *eine* VrtAcc-Nummer im GSI-Timing übersetzt werden.

Alternativen, die nur einen Teil eines Zyklus betreffen, sind *nicht* möglich! Geräte im alten System haben eben nur einen Soll- und einen Istwertsatz pro VrtAcc.

Alternativen, die einen kompletten Zyklus betreffen, sind eigentlich keine, könnten also über eine eigene VrtAcc-Nummer abgebildet werden.

Siehe auch Kapitel 7 und Anhang A, Punkt 5.

C.1. Therapiebetrieb

Use-Case-Nummer U-009.

[Bisher nur Stichworte:]

Falls therapie-ähnlicher Betrieb im GMT (sog. multi-parameter beam) nur mit BPIDs gemacht wird, z. B. mit 79 verschiedenen BPID-Sätzen¹⁰, wie kann man das übersetzen in 79 verschiedene EFICD-Events, alle für den VrtAcc 0?

!?

Sie auch [3], Kapitel *Therapiebetrieb*.

Literatur

- [1] Controls Core Team, *BPCID, SID, BPID – Common*.
- [2] Controls Core Team, *BPCID, SID, BPID – Use Cases*.
- [3] Controls Core Team, *BPCID, SID, BPID – Solutions*.
- [4] FESA FEC Integration Team, *Guideline for Designing FESA Equipment Software at the GSI and the FAIR Facility*.

¹⁰37 + 42 = 79 ☺