

# Minutes of the 1<sup>st</sup> CSDT meeting, ESTEC Jan. 25, 2001

**Ref: FIRST/FSC/MOM/0170**

**Date: Jan 29, 2001**

**Issue: 1.0**

**Author: Stephane Veillat**

## **Participants:**

Odile Coeur-Joly (CESR)

Peer Zaal (SRON)

Hassan Siddiqui (ESA/HSCDT)

Do Koster (SRON)

Kevin Galloway (ESA/HSCDT)

Erich Wiezorrek (MPE/PACS)

Jean Marie Wallut (CNES)

Jon Brumfitt (ESA/HSCDT)

Stev Guest (RAL/SPIRE)

Rik Huygen (KUL/PACS)

Peter Roelfsema(SRON)

Johaness Riedinger (ESA/HSCDT) (afternoon only)

Stephane Veillat (ESA/HSCDT)

## **Introduction**

New members of the CSDT, namely Peter Zaal, Do Koster and Steve Guest, were specially welcome to the meeting and to the team.

PR explained that Odile Coeur-Joly and Jean Marie Wallut were attending the meeting at his invitation and as observers.

PR let us know that Albrecht De Jong could not attend the meeting for private reasons. SV mentions that Mark Thomas would certainly not join the meeting as he was sick the day before.

It was clarified that CSDT meeting should take place on a monthly basis.

**RH took an action to set-up a mailing list for the CSDT**, this mailing list will replace and complete the current use case one.

## **TOP 1: Approval of agenda**

See agenda in appendix 1:

The agenda was approved with the following exception: due to the absence of both Albrecht and Mark, it was agreed that the agenda item on CUS could not be addressed.

EW clarified that there would be no presentation on his side regarding the CCM.

**PR took the action to trigger Albrecht for contacting Mark to discuss CUS issue.**

## **TOP 2: IA/QLA framework**

JJM presented the status of the work he has been performing so far in this area and in particular the prototype, see slides appendix 2.

VG #7 shows all the classes for the prototype, i.e the way the framework can be used to develop QLA

VG #8 shows the framework . It is limited to the CCM. This is what would be provided to the ICCs to develop their IA/QLA

Specific comments triggered by JJ presentation

- QLA only makes sense if this is a RT process, i.e. if QLA displays are updated as TM arrived and not waiting for TM of a complete BB to be received (PR). The prototype should be modified in this respect.
- In QLA/IA one should have the possibility to change the data processing on-line (PR)
- TM (for PACS+HIFI) packets: all science and HK TM are tagged with OBSid and BBid, versus only start/end block being marked. The prototype can easily be adapted to that. **SG took an action to investigate if the marking is similar for SPIRE TM, by next CSDT meeting.**
- TM can arrive in different order not necessarily block/block. QLA should be able to cope with that. This will be different with IA where TM will be retrieved on a block/block basis.

EW stressed that PACS is expecting an interface to the FCSS whereby the QLA can register for receiving TM (product TM) as they are made available, and nothing more.

This led to questions on what is meant exactly by the IA/QLA framework and what it brings on top of TM ingestion . TM ingestion goes up to the point where it generates and stores product TM in BB , it could/should also therefore serve product TM to QLA.

EW mentioned that Ekkie Wieprecht was also developing a prototype for the IA/QLA framework.

SV/PR/JJ told their surprise that the activity from Ekkie were not known to the rest of the team and in particular to JJ, who was clearly nominated responsible for this WP. SV/PR expressed strong concerns that we were seeing a lack of communication and coordination in the team. SV pointed that, as a minimum, status of the PACS development in this area should have been presented at the meeting. PR expressed his feeling that PACS was moving its own way with little consideration for the common development.

EW and other attendees saw these parallel activities more as an opportunity (2 solutions to choose from instead of one) than a threat.

It was however agreed to have a coordination meeting asap to discuss the above issues on the scope of the IA/QLA framework, to compare analysis and design done by Ekkie and JJ and to analyse the impact of this work on the CCM. **JJ took the action to organize the meeting.** The meeting is planned on the 13/02/01. It was agreed that this meeting should include only relevant SW developers.

### TOP 3: CCM

JB presented the CCM, related potential updates and issues (see slides Appendix 3)

It was agreed that the class "target" should be renamed "Pointing mode" and that pointing mode should be associated to observing mode rather than observing request. In this respect, it was clarified that an observing mode can operate with several pointing modes (PR + EW).

As a side remark on the CCM, SG pointed to the fact the proposer/person inheritance relationship should be reviewed.

JB pointed the BB definition in the CCM as being still immature although it is at the heart of the CCM. JB has investigated four approaches to implement BB (see slides). From the meeting discussion, it emerged that the option 3 (BB defined using a scripting language as JPython, JPython script is an attribute of the BB mode class defined at the time of instantiation of the class BB mode, a new BB mode is an object ) was the most promising one. JB has already made a prototype which shows the feasibility of this option. EW pointed that this option seems in line with the outcome of the ILT splinter group. It was also noted that this option is not exclusive from option 1 (BB mode defined as a Java class, see also JJ prototype) which could be used for stable BB mode definition, e.g. in AOT. It was agreed that this issue should be further discussed at the technical meeting on the 13/02, as the issue of BB implementation is also very much interconnected with the IA/QLA framework.

A general comment was made by EW that the ILT technical note had led to a lot of development in terms of class modelling which should be taken into account for the CCM. **JB took an action to review carefully the note (v1.3) and identify potential impact on the CCM by 13/02.**

#### **Controlling changes to the CCM:**

See JB slide (appendix 2)

It was agreed that when real coding would start, the changes to the CCM should be formally controlled (i.e. through a CCB) . At this stage of the development all changes to the CCM (yellow classes) should be co-ordinated by JB, i.e. he should be aware of any change proposal and co-ordinate actual implementation of the change.

#### **TOP 4: DB access framework**

See JB slides (appendix 3). JB presented current status on design architecture and prototype to interface application SW with database.

In this context, SG raised the issue of 3 tier versus 2 tier architecture. SG argued that it would be more appropriate for application SW to access the DB through an object server (3-tier architecture). In particular because SPIRE ICC will have several sites accessing one physical database. EW also pointed that a 3 tier architecture would make it easier to deploy new versions of the core classes, as only the object server would need to be updated, versus all SW applications.

JB sees the 3-tier architecture relevant only for applications accessed outside the FSC (e.g. PHS) and argued that they are other solutions to deploy new versions of the core classes.

JB will further address this issue in the TN resulting from the IA WP.

It was however clarified that this is not an issue for ILT and that the choice of 2-tier versus 3 tier architecture can be postponed to after FCSS v0.1

Finally, it was precised that developers do not yet need to consider the DB access for going on with their anlysis and design WP. Access to DB will not impact their architecture.

#### **TOP 5: Open issues**

See appendix 4: report from KG

#### **TOP 6: ILT use cases:**

JRR and SV stressed the importance of keeping the UCs and in particular the ILT UCs up to date as they are the staring point of the on going development. Currently ILT UCs still include a large number of open issues which needs to be tied up.

**EW took an action to update the ILT use cases by 21/02**

**For EW to be able to do the job, KG will convert UC in HTML by Wed next week 31/01.**

#### **AOB:**

Next meeting date: It was agreed that the last Wed. of every month should be reserved for CSDT meeting. CSDT meeting place will alternate between ESTEC and institutes. The next meeting will take place on the 22/02 in RAL (breaking the general rule above)..

JRR stressed the agreement at the FCSS-MG that ICCs should report monthly on WPs in the same fashion as FSCDT.

Following FCSS SRR recommendation , it was agreed that FINDAS could be dropped from the FCSS documentation, i.e. FCSS

Together upgrade. All ICCs have running Together 4.2. beyond this point, it was agreed that SW COTS set-up should be a standard agenda item for the CSDT meetings

**Action items (recap):**

<b>AI#</b>	<b>Actionee</b>	<b>Action</b>	<b>Due date</b>
CSDT#1-01	RH	Set up CSDT mailing list	asap
CSDT#1-02	PR	Trigger AdJ to meet MT on CUS issues	asap
CSDT#1-03	SG	Investigate TM Obsid and BBid tagging for SPIRE	Next meeting 22/02
CSDT#1-04	JJ	Organize IA/QLA framework meeting on 13/02	asap
CSDT#1-05	JB	Review CCM in light of ILT UC analysis TN (V1.3)	13/02
CSDT#1-06	EW	Review/update ILT UCs	21/02
CSDT#1-07	KG	Convert UCs into HTML	31/01
CSDT#1-08	SG	Determine SPIRE APID allocation	02/02
CSDT#1-09	EW	Answer new open issues related to open issue 2.12.2	09/02

## **Appendix 1: Agenda**

## Proposed Agenda for CSDT-meeting #1/2001 24-Jan-01, ESTEC, Room Bf228

- 10:00 Welcome, Introduction, Agenda SV
- Under this agenda item we will also appoint the secretary to this meeting, i.e. the unlucky person who gets to write the minutes. It would be great if someone volunteered (hint: all teams except SPIRE have previously done this, but Steve might not feel entirely comfortable at taking minutes of the first meeting of this group which he attends...).
- 10:15 Demonstration and discussion of a QLA infrastructure prototype JJM
- Jean-Jaques will demonstrate initial results from his prototyping effort to give everyone a feeling for what can be done. We should not forget that this is an *investigative prototype* that very likely does not reflect *how* things will be done in the HCSS later on (so people should not go away with the source code and believe they can now start building around the classes this prototype implements!); indeed Jon and JJ have only started to analyse which of the 100 or so classes implemented in this prototype could be useful in the Core Class Model...
- 11:15 CCM JBr
- In his architectural investigation in general, and more recently in interaction with JJ and the QLA infrastructure prototype, Jon has come across a few proposed/potential changes to the CCM that need to be discussed. At least one of the potential changes is closely linked with an "open issue" (item 2.2.21 in the "open issues" document) which we have to discuss later at this meeting. More generally, we should firmly agree on how we propose and disposition changes to the CCM in the future, when more and more people will start to implement prototypes of various "infrastructure items"...
- E.g. what about the presentation Erich gave at the clarification meeting, which contained classes not currently being represented in the CCM ? Has any further progress been made which could be presented ?
- 12:00 Progress on DB abstraction JBr
- To the FSCDT, Jon occasionally gives a talk on things he wants us to be aware of, such as design patterns. He'll give us a bit of insight into what he has done to make it completely transparent to client applications whether they are using a Versant or an Objectivity DB and what *your* application needs to do (i.e. how you have to use this infrastructure) to make the same JAVA code run even if tomorrow we decided to replace Versant by something different.
- 12:30 Lunch
- 13:30 CUS MT
- Getting started on the CUS WPs, the feeling is growing on Mark that certain aspects of the CUS are not yet sufficiently well defined/agreed, in particular which kind of logical and arithmetic constructs the CUS needs to support. In the plenary part of the CUS discussion we need to agree a course of action how we pin down and agree this (most likely in a series of actions) before Mark and Albrecht take other CUS issues off-line in an office as this is their only chance to spend a bit of time together before Albrecht comes off his EGSE work in February.
- 14:00 Open Issues KG
- On the day before this meeting, we will have gone through the list of open issues with the PST, but some of the issues that will soon become urgent are of no interest to the PST and have to be dealt with by the CSDT. According to our records, these are:

- 2.2.21 Observing mode scope. Clearly it will make a difference to the CCM if we define an “observing mode” as something which is purely instrument-related or as something which has instrument *and* spacecraft related components.
- 2.9.1 Missing consolidated telemetry. Remember that we assume that we can detect missing TM using the sequence count, which is per APID. What progress has been made in assigning APIDs by the instrument developers ? ICCs please report...
- 2.9.2 Observation measurement/building block identifiers in science TM. Can Erich please report on the current status of PACS plans to assign BB IDs on-board during execution of an observation (which is incompatible with our current CCM, in which the HCSS knows about these identifiers) ? Any changes in approach for the other instruments ? ICCs please report...
- 2.9.3 Detection of non-observation telemetry ? Can anything new be said to illuminate this issue or does MOC have to get involved through the HGSSE ?
- 2.12.2 Generate and transfer ILT test procedure. This open issue goes back, I believe, to the time when it wasn't clear whether test procedures would be prepared on Test Control or (in the form of “schedules”) on HCSS. Can this be closed now that we have a better idea of the I/Fs between HCSS and Test Control ?

15:00 ILT Use-cases all

Recommendation R18-2 from issue 1 of the SRR/v0.1 PDR Board Report reads “Complete all ILT use-cases urgently” and RID PD1-CCB-59 (on the ILT UC document) reads “It is not clear what is the real purpose of this document, or in other words, what should really be presented for each use case (activity diagram, sequence diagram, class diagram, etc ). Some use cases are said to be missing up-front. [...]”.

I don't believe anyone of us really disagrees with these statements, which clearly indicate that not all is well in this area and that something needs to be done. The purpose of this agenda item is to agree *who* does *what* by *when*. If we stay by what we have said often enough in the past (that “the common development is driven by use-cases”) we can't just leave the ILT UCs in the state they are in at present...

16:00 AOB all

- We need to find a day for our next CSDT meeting in February where most of the people can attend.
- Also, I want to remind everyone of the agreement reached at the last HCSSMG telecon concerning how every member of the CSDT reports on workpackages for the monthly progress reports (read FIRST/FSC/MOM/0169 if you haven't done so yet!).
- Board recommendation R08-4 states “FCSS development teams to decide whether or not to retain the name 'FINDAS' and to make all documentation consistent with that decision”. What is the CSDT's opinion ?

16:30 End (hopefully most of you are still here then)

## **Appendix 2: JJ slides**



# QLA prototype

Jean-Jacques Mathieu



# Plan

---

- Purpose of the prototype
- Prototype and CCM
- Lessons learned
- Open issues
- What do we expect from users
- What next?

# Purpose of the prototype

---



- Concentrate on QLA aspects
  - Telemetry flow (acquisition and dispatching)
  - Product and Process (from TM to Product)
- Recommendations for TM
- Provide early a partial implementation of the functionality provided by the main classes of the CCM for test and triggering of ideas
- Assess difficulty of implementing observing modes

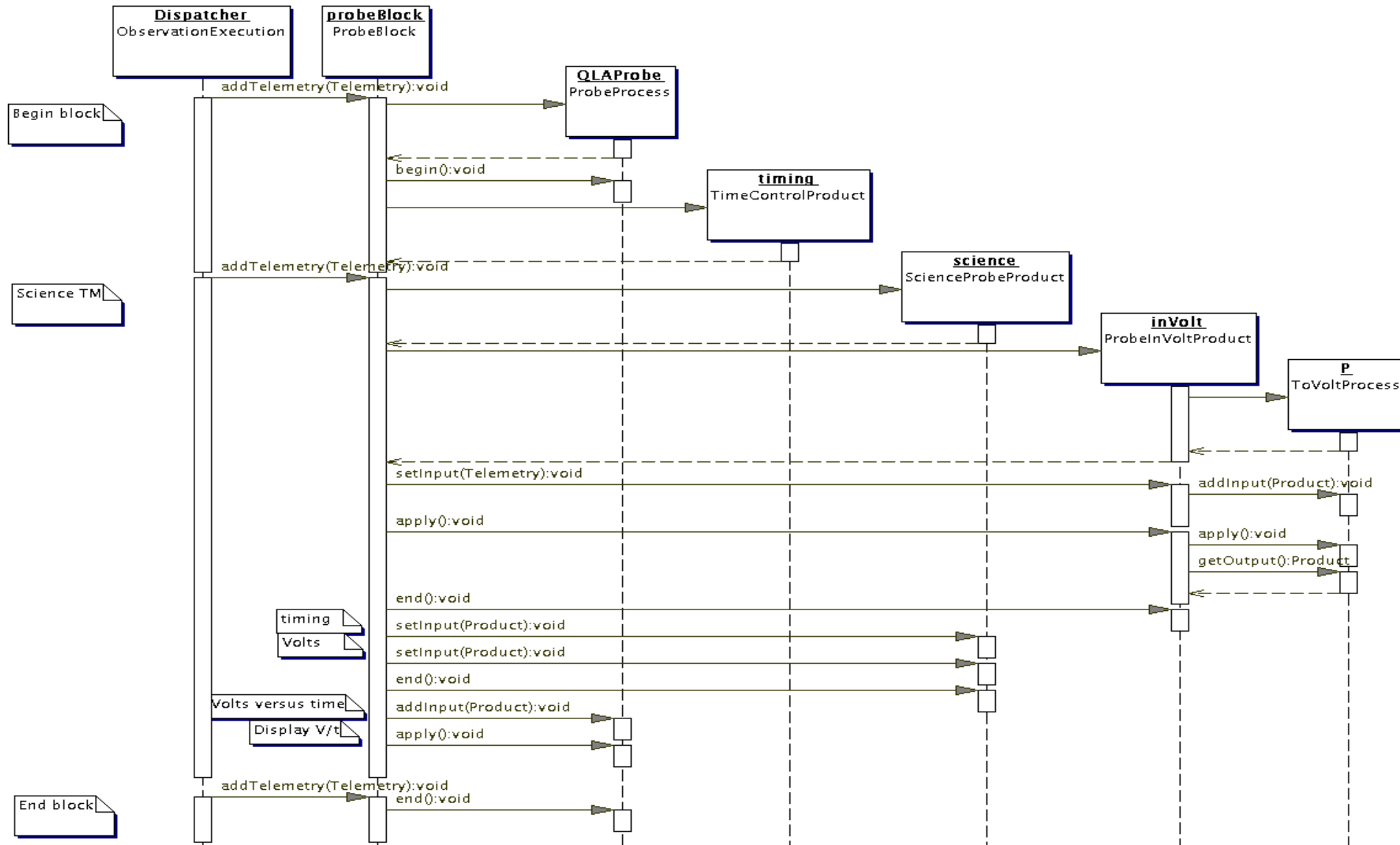
## Observing mode example

---

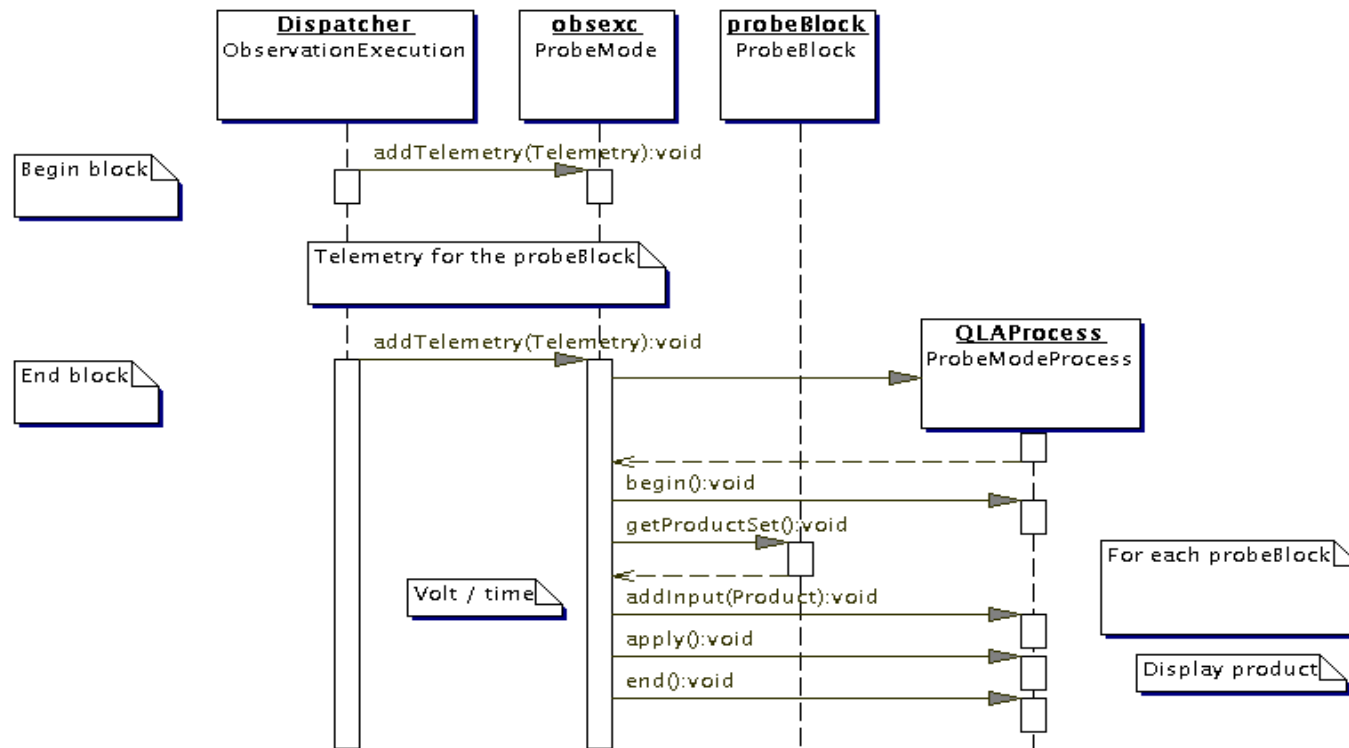


- Tension measured at irregularly known intervals
- Tension is converted to bytes
- Result is set in a telemetry packet
- Number of measurements is given in a telecommand
- There are two levels of blocks:
  - Observing mode level
  - Building block level
- Display every measurement in Volts/time
- Display all measurements in one scrolling window

# Telemetry acquisition and QLA (1)

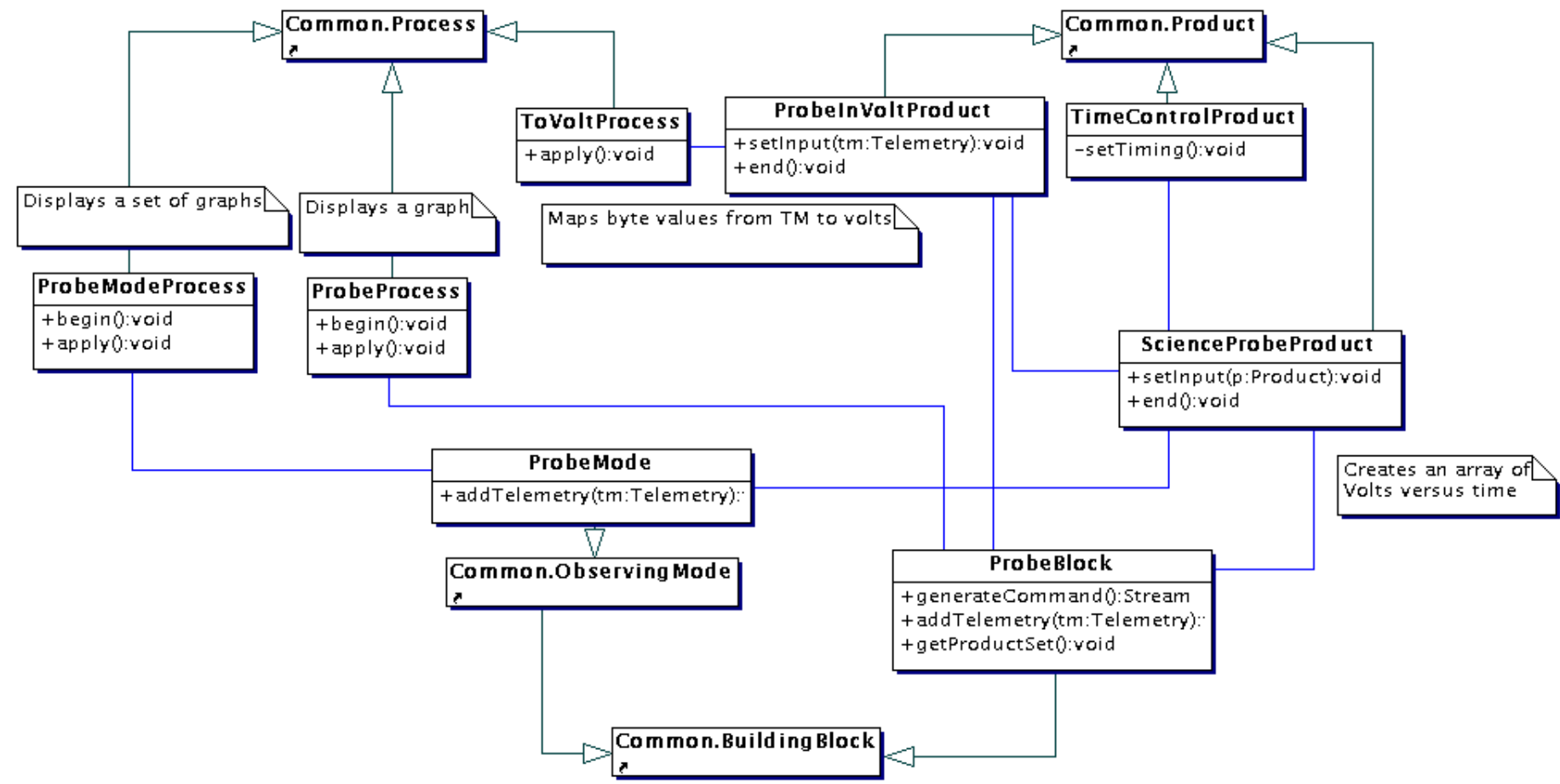


# Telemetry and QLA (2)

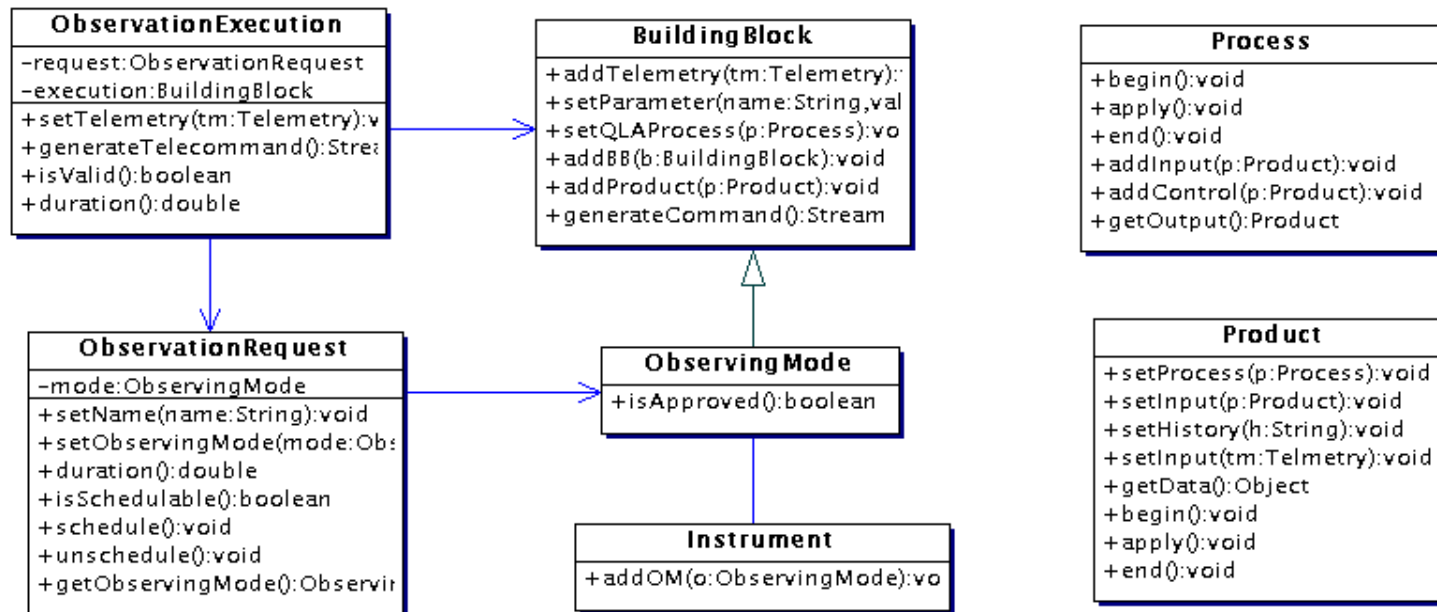




# Mode classes



# Common classes





# Telemetry aspects

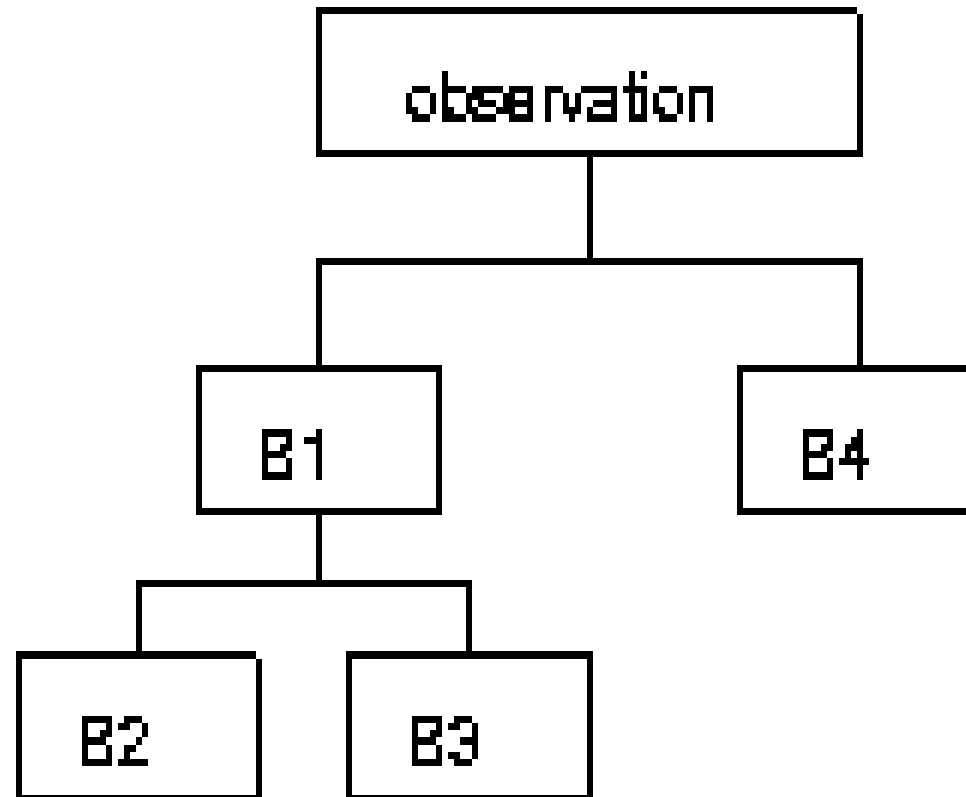


- 
- Marking is immutable (fixed during instrument command generation)
  - Mark begin and end of observation and block
  - Other telemetry should not need marking
  - The structure implied by the marks reflects that of building blocks
  - Telemetry forwarded to Observation Execution
  - Observation Execution dispatches telemetry to subordinate blocks
  - There is only one block 'active' at any given time

# Example



begin observation  
begin b1  
tm  
tm  
begin b2  
tm  
tm  
end b2  
tm  
tm  
begin b3  
tm  
end b3  
end b1  
begin b4  
tm  
end b4  
end observation



# Core functionalities

---



- Retrieving elements by name (ie `getProductWithName("inVolt")`)
- Parameter handling:
  - askable or not
  - demotion to subordinate blocks and promotion to upper block
  - Uniqueness by name
- Default behavior for begin/end block command generation
- Default behavior for telemetry dispatching

# Lessons learned

---



- CCM:

- is a analytic view of the problem not implementation
- classes must be well described (responsibilities unclear)
- should be challenged and changed when needed

- Prototype:

- Have a clear goal(s) before prototyping
- Do not write "quick and dirty" code (you might use more of it than you thought)
- For evolutionary prototype, conform to agreed models (still can challenge and change them)
- Be wary of having to write quite some code which is 'helping code' for the prototype (ie not core or essential)



# Open issues

---

- Parameter handling
- Where do the pointing belong
- What really is the definition of "block execution", "observation result" (may be they do not belong here?)
- What are the responsibilities of each of the 'main' blocks: observation request, observing mode, building block and observation execution
- What are the functionality each have and what is the default behavior for each of them

# Feedback and next actions

---



- Missing functionality
  - Default behavior
  - Commands/telemetry/BB marking
  - Parameter handling
  - Applicability of Product/Process to other systems than QLA
  - Scripting (what, where, how)
- 
- Update CCM
  - Write responsibilities
  - Code the core classes (or packages containing them)

### **Appendix 3: JB slides (CCM)**

# Core Class Model

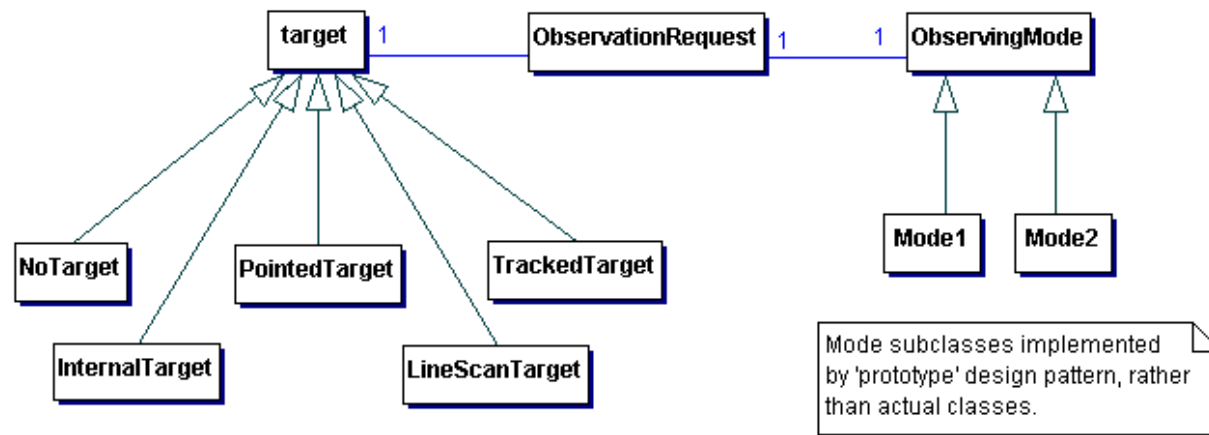
Jon Brumfitt





# Issues

Target and ObservingMode are closely related



- ◆ Do we have this part of the model right?
- ◆ Only certain ObservingModes are relevant for a given Target type
- ◆ Model should be driven by problem domain
- ◆ 'Target' should perhaps be 'PointingMode'



# Building Blocks

---

BuildingBlocks are the biggest area of uncertainty in the CCM

## Approaches:

- ◆ Hard-coded sub-classes of BuildingBlock
  - ◆ sufficient for routine ObservingModes
- ◆ BuildingBlock sub-classes for language constructs (e.g. LoopBlock)
  - ◆ parser generates AST using these sub-classes
- ◆ JPython interpreter executing script
  - ◆ store script as attribute of block
- ◆ Compile new Java classes on-the-fly
  - ◆ serialize and store classes as attribute of persistent block

## Needed:

- ◆ requirements for language constructs
- ◆ exploratory prototypes of major candidate approaches



# Minor changes

---

- ◆ CalibrationTarget renamed InternalTarget (RID)
- ◆ ObservationExecution needs 'generateCommand' method



# Controlling changes

---

How do we propose and accept changes to the CCM?

- ◆ Initially, need flexibility, during architectural investigation
- ◆ Later, need more control over changes
  
- ◆ One person responsible for the CCM package
  - ◆ maintain a coherent concept
  - ◆ coordinate changes



## **Appendix 4 JB slides (ODBMS)**

# Progress on database abstraction

Jon Brumfitt



# Status

---

## Database evaluation (complete)

- complete (procurement in progress)

## Architectural investigation (in progress)

- goal: to remove dependencies on a particular ODBMS
- investigated abstraction of various features
- prototype implemented for Versant and Objectivity
  - simple clients which work with either database
- producing more complete version for Versant only
  - integrated with some simple clients
- extending to support multiple databases
- support for sandboxes to be investigated
- technical note in progress





# Approach

---

## Aspects of abstraction

- ◆ abstraction of databases, transactions, queries, etc
- ◆ abstraction of persistent objects

## 'Pluggable' component architecture using abstract factories:

- ◆ 'ObjectStore' class provides generic API for an OO database
- ◆ factory to create ObjectStore configured by Java properties file
  - ◆ to specify concrete implementation
  - ◆ to specify database(s) etc
- ◆ factory to create instances of persistence-capable classes
- ◆ ObjectStore acts as a query factory
  - ◆ uses properties for query resources

## Appendix 5: Open Issues

### Open Issue 2.2.21: Observing mode scope

*1. Does a single observing mode address both the use of instrument and spacecraft or is there an observing mode associated with the instrument and an observing mode associated with the spacecraft (rasters etc)*

It was noted that on ISO an AOT contained pointing information  
Pjotr gave an example of a “peak-up” observation which would use 2 subsystems.  
Erich mentioned “Raster-scans”.

Yes, an observing mode is applicable to more than one subsystem (instrument, s/c, ...)

*2. Do building blocks contain telecommands relating to one subsystem or to more than one subsystem.*

Non-issue as it is purely a matter of definition.

Open issue will be closed

### Open Issue 2.9.1: Missing consolidated telemetry

*The following requirement is not captured in any requirement documents. It is posted in this "open issues" document until is captured in a particular document or the underlying issue gets solved.*

*It shall be possible for the FSC to detect missing consolidated TM data. TM data lost during the space-ground transmission will be missing in the consolidated archive from MOC. One way for the FSC to detect missing TM data is to use the TM packet sequence count. TM packet sequence count is done per APID.*

*Considering that the FSC will be handling science data separately from HK data, this should mean that there is at least an APID specific to science TM data for each instrument.*

HIFI: 4 independent data sources with separate APIDs. Housekeeping has its own APIDs.

PACS: Erich stated that at the very least one APID has been allocated to science and one to housekeeping

SPIRE: Details not known

**Action:** Steve Guest – determine the details of SPIRE APID allocation.

Due date: 02/02/01

The open issue will be closed following completion of the action.

### Open Issue 2.9.2 Observation measurement/ building block identifiers in science telemetry

*The following requirement is not captured in any requirement documents. It is posted in this "open issues" document until is captured in a particular document or the underlying issue gets solved.*

*The instrument science TM data shall include the necessary information for the FSC to be able to associate, when relevant, each TM data to the context of an observation measurement. (i.e. BB).*

PACS will make extensive use of on-board procedures which may potentially switch between building blocks (assumed to mean a choice of building block could be made based on previous events)

The real problem was stated to be: Making sure the allocated building block identifiers are correct. If they are generated on board they must match what would have been generated on the ground.

This information will be added to the open issue but the open issue remains open

### **Open Issue 2.9.3: Detection of non-observation telemetry**

*The following requirement is not captured in any requirement documents. It is posted in this "open issues" document until is captured in a particular document or the underlying issue gets solved.*

*The instrument and spacecraft TM data shall include the necessary information for the FSC to be able to detect when TM data have been generated outside the context of an observation. TM may be generated following manual commanding of satellite from the MOC. It is therefore expected that instrument manual commanding from MOC will lead to special tagging of resulting TM, allowing FSC to relate these TM to the manual commands. This will most certainly mean that the command sequences delivered to MOC by ICCS for manual commanding of the instruments will have to include special obsid setting.*

Erich stated that the ILT working group technical note addresses this issue.

A reference to this technical will be added to the open issue

As few people had read the TN the open issue remains open until confirmation of satisfactory solution at the next open issues meeting.

### **Open Issue 2.12.2 Generate and transfer ILT test procedure**

- 1. An extension use case is needed to describe the case where the EGSE-ILT does not follow the usual sequence of events; for example, EGSE-ILT dies halfway through a test procedure and does not generate a log.*
- 2. The current scenario does not allow nested procedure templates. Test procedure templates may use nested constructs as the scripting language allows, but the current model requires each procedure to be a self-contained script.*

Erich stated that the ILT working group technical note addresses this issue.

A reference to this technical will be added to the open issue

As few people had read the TN the open issue remains open until confirmation of satisfactory solution at the next open issues meeting.

Erich requested a new open issue should be added to the open issues document: "How to associate a test procedure script with a test procedure execution object"

**Action:** Erich Wiezorrek – To provide an answer to this new open issue.  
Due date: 09/02/01

## Open Issue 2.12.9: Unconsolidated telemetry ingestion

Note: Not currently in Issue 1.0 of the Open Issues document – will appear in the next release. This open issue records the open issues identified in UCF-758.

*1. The contents of the telemetry packets can be compressed. If the observing mode/ building block identifiers/ markers in the telemetry are compressed also then the ingestor will have to decompress to obtain this information.*

Yes – decompression must be performed

*2. Is it necessary to log the ingestion of the telemetry packets.*

Yes – as a general design approach clients etc should log their activities and detected anomalies.

*3. Telemetry packets will be generated that do not belong to an observing mode. What happens to these. This is definitely true for housekeeping and event packets and is possible for science packets.*

See open issue 2.9.3.

If the packet does not contain the necessary identifiers it goes into a container.

*4. The detection of missing packets: mimic the behaviour of consolidated telemetry*

Agreed. Use of caching to enable the telemetry packets to be ordered and missing packets to be determined.

*5. TM ingestion should support a scenario where tests on different models run in parallel - for example ILT on the HIFI flight model may be run in parallel with IST on the qualification model. At least there should be no problem when the data are merged into a shared/ distributed database.*

Note: Not to be understood as 2 sources operating in parallel.

Before starting a test the source of the data must be identified to the system.

*6. EGSE-APIDs may not be unique. A number of APIDs is reserved for EGSE purposes. This number is so small that different instruments will probably use the same APID for different purposes. Ingestion must cope with that.*

Remains open.