SCOS-2000
Time Correlation
Technical Note

Document Reference:	EGOS-MCS-S2K-TN-1030
Document Status:	Version 1.0
Prepared By:	SCOS-2000 GIMUS Team
Date:	2018-10-31

DOCUMENT CHANGE LOG

Issue/ Revision	Date	Modification Nb	Modified pages	Observations
Version 1.0	10/08/2018		All	Creation of the document for S2K-7699 Chapter 1,2,3 are mostly new; chapter 4 is taken from the EUD4S2K SUM; Section 5.2 is taken from a GAIA TCO document; the rest refers to ADD-0017.

Document Change Record

DCR No:	1.0	Originator:	GIMUS Team
Date:	12/17/2018	Approved by:	
Document Title:			
Document Reference:			
Page	**Paragraph**	**Reason for Change**	
n/a	n/a	Editorial changes only	

1 ABSTRACT

This document describes how the SCOS-2000 Time Correlation works, its purpose, mechanism, systems and architecture.

1.1 TABLE OF CONTENTS

1.2 LIST OF FIGURES

1.3 LIST OF TABLES

2 INTRODUCTION

2.1 PURPOSE

This document describes the time correlation in detail. It is intended as a general guide to the persons working with the Time Correlation in SCOS 2000.

2.2 SCOPE

This document describes the SCOS-2000 time correlation components, processes and other details.

When applicable, a reference to an existing document is given instead of a full description of the interface.

2.3 ACRONYMS AND ABBREVIATIONS

A complete list of the Acronyms and Abbreviations used within the SCOS-2000 Project is available in [RD1].

2.4 REFERENCES

2.4.1 Applicable Documents

The following documents are considered applicable. The applicable issues of the SCOS-2000 documents are identified in the latest issue of the SCOS-2000 Document Control List [CNF-01] or the EUD4S2K Document control List [CNF-02].

Doc.	Reference	Title
AD1	EGOS-EUD-S2K-SUM-1001	EUD4S2K Software User Manual
AD2	EGOS-MCS-S2K-ADD-0017	TM Packetiser and Packet Displays ADD

2.5 Reference Documents

The following documents are referenced. The applicable issues of the SCOS-2000 documents are identified in the latest issue of the SCOS-2000 Document Control List [CNF-01].

Doc.	Reference	Title
CNF-01	CIDL-0001	SCOS-2000 Document Control List
CNF-02	DCL-1001	EUD4S2K Document control List
RD1	GLO-0001	SCOS-2000 Glossary, Definitions and Acronyms
RD2	ICD-0001	SCOS-2000 Database Import Interface Control Document
RD3	SUM-0007	SCOS-2000 MSG Software User Manual

3 OVERVIEW

3.1 GENERAL CONTEXT

The **Time Correlation** establishes a correlation between the **On-Board Time** of an event (OBT, the actual on-board time accordingly to the on-board clock) and its time seen by SCOS-2000 (what would be the on-board time according to the ground clock). The reason behind this is ensuring the universality of the time measurements. This is mainly needed by the Packetiser, for time-stamping the packets, and by the Releaser, for encoding TT (Time-Tagged) commands.

To do so, the conversion is performed by using the OBT contained in a special telemetry packet called Time Packet, which holds the on-board time at which the previous frame[1] transmission occurred (see Figure 1Time Correlation Shema). Time Packets are usually generated every some fixed number of frames. When the frame preceding that one containing the Time Packet, arrives on ground, the Earth Reception Time[2] (ERT), a field inside the frame (see **Error! Reference source not found.**), it is used to derive the time at which the frame was generated (by subtracting a few transmission delays). The inferred time (UTC) is probably different by the OBT arriving with the next frame (and the reporting the time of the same event), because calculated from a different clock (a different time reference), likely not synchronised with the on-board one.

To describe the function to convert the local time into the on-board one (UTC to OBT), the TCO Server creates the **Coefficient Packets** storing the gradient and the offset (a_1 and a_0) of the linear transformation function[3]. To calculate them, it uses the **Time Couples** (OBT and UTC) and optionally one of the two methods (see also Figure 1Time Correlation Shema):

- ✓ If the on-board clock is synchronized (like with a GPS device), then we know the time is kept correctly except for a fixed difference with the ground clock ($a_1 = 1$).
- ✓ If the on-board clock is not synchronized, then a least-squared fit[4] is performed over a set of Time Couples to determine the coefficients and the gradients.

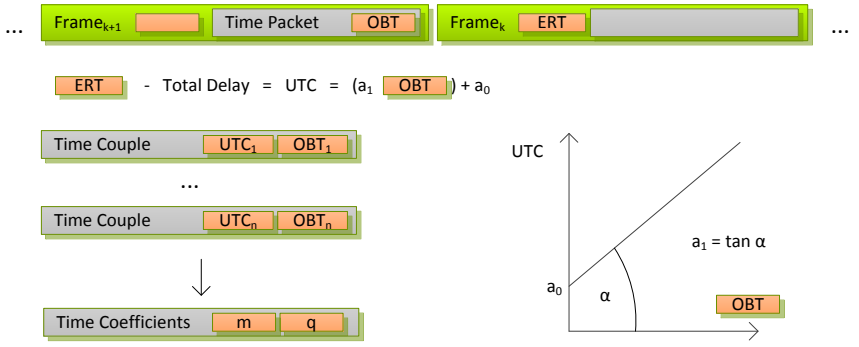


Figure 1Time Correlation Shema

[1] It is physically impossible to save the time at which the frame was generated inside itself. That is why it will be delivered with the next one.

[2] This value is inserted inside the frame by the Ground Station once the frame is received on ground.

[3] A linear function is described by $f(x) = a_1 x + a_0$.

[4] A least-squared fit is a mathematical procedure to find the best curve fitting a series of points. It minimizes the sum of the squares of distances between each point and the candidate curve until it finds the "perfect" one.

It is allowed to dynamically switch between these two methods using the **Time Correlation application**.

Every time a particular Coefficient Packet (a_1 and a_0) is used, the subsequent Time Couples are constantly monitored in order to verify its reliability.

The **deviation** is the difference between the OBT and the UTC after the coefficients have been applied to it (after the transformation). In principle it should be always zero but, in practice, it might change at some point, due to the non-constant transmission delay or the on-board clock drift.

The deviation is used to define the **accuracy** and the **validity**, two properties, identified by two limits (accuracy ≤ validity), that define the reliability of the Time Correlation. It can be:

- ✓ ACCURATE and VALID (deviation ≤ accuracy)
- ✓ INACCURATE and VALID (accuracy ≤ deviation ≤ validity)
- ✓ INACCURATE and INVALID (validity ≤ deviation)

Both the accuracy and the validity limits are configurable in the Time Correlation application.

In automatic mode (the time correlation run autonomously), the coefficients are calculated using the latest Time Couples anytime the deviation exceeds the specified thresholds for the validity.

Time Coefficients are usually valid for a specific period and this period is taken into account when the data is retrieved from the archive, to correctly re-calculate the reception times.

If we don't have real **TCO** data, we can use simulations to test time correlation functionality. Packets can be injected in **SCOS** through the **TM-TC Test Tool** under the **Simulation** tab. After setting up the simulation tool, go to the **TM Packet Injector** tab, right click and **Add Packet**. Now you need to specify the **SPID** of the packet. When it comes to time correlation there are 3 types of packets that are of interest, as we can see in figure 5:

Time Report (SPID 1000005) -> Time Couple (SPID 1000098) -> Time Coefficients (SPID 1000099)

(Time Report == Time Packet)

The **Time Correlation Client Library** provides the services to perform the time conversions as well as the functionalities to access the **Time Correlation Server** which is built around the singleton class **TCOconcreteInstance**. The TCO Library is configured using the **TCOconfig** class which creates an instance of the **TCOobt** class for each clock on-board. The time format of the clock is represented by the **TCOtimeFormat** class, at the moment implementing only the CUC and CDS formats. It is possible to define new formats if needed.

Based on the configuration provided by the TCOconfig class, the TCOconcreteInstance creates a **TCOclientCache** object for each OBT clock which needs to be monitored by the system. Each request to the TCOconcreteInstance, in fact, needs to specify the OBT clock identifier depending on which, the request is forwarded to the proper TCOclientCache which will in turns autonomously decides to either perform the operation by itself (using a cache of recent coefficients) or send the request to the TCO Server.

3.2 GENERAL STRUCTURE

The purpose of the time correlation server and client library is to establish a relation between the time measured by an On Board Clock and the time as measured on the ground. This is done by generating a set of time correlation coefficients based on the history of stored OBT/UTCTC pairs (time couples) created when time packets are received from the spacecraft which are used to permit conversion between on-board and on-ground times based on either the current or historical coefficients.

The time couples relate the time assigned to an event by an onboard clock to the time assigned to the same event by ground based measurements. The calculation of these two times is somewhat complex and requires the consideration

of various delays which are configurable in the system. Because it is important to understand these and other concepts in order to correctly consider the system, the following will be presented:

- Time Correlation Mechanisms
- Time Correlation Packetizer
- Time Correlation System & Architecture Design

Apart from the handling and processing of time packets the key elements of the time correlation functionality are provided by the TCOclient library and the TCOserver process. There is no access to the TCOserver other than through the TCO client library interface. The user interface to time correlation is provided via a dedicated panel on the Time Correlation application. The Time Correlation application communicates with the TCO via the client library, not directly.

The full functionality provided by the time correlation function is listed below.

- Time correlation calculation from on-board time / ground time (OBT / UTC) couples.
- Time correlation conversions between on-board time and ground time, and vice versa.

The following time correlation management issues are also taken into account:

- Time correlation conversions using old applicable coefficients.
- Management of absolute-time parameters.
- Continuous update of the time correlation validity, and accuracy status.
- Propagation Delay estimation.
- Storage of OBT / UTC time couples in telemetry packets.
- Storage of time correlation coefficients and associated status in telemetry packets.

4 TIME CORRELATION MECHANISMS

In order to explain this it is necessary to have a basic understanding of the mechanisms of time correlation as supported by SCOS-2000 whose fundamental aim is to determine the relation between the time as recorded by a clock (or clocks) on board the spacecraft and UTC time on earth and to permit conversion between UTC time and on-board time and vice versa.

The mechanism for the computation of this relation is as follows:

The spacecraft periodically samples the time as measured by the on-board clock and reports this to the ground via a time packet inserted in a telemetry frame. The triggering event for the sampling of the on-board time is the transmission of a telemetry frame with frame counter equal to a multiple of N where N is a power of 2 between 1 and 256.

The time packet itself contains the on-board time "latched" as a response to this triggering event and is downlinked to ground in a subsequent frame. The MCS therefore is able, with a knowledge of the value of N, to determine which frame triggered the sampling of the time within the time packet. The UTC reception time of this frame is known from the time stamped in the NIS header in which the frame was received at the MCS but in order to identify the UTC time to which the on-board time in the time packet corresponds it is necessary to take account of a number of delays. These are in reverse order:

- Any processing delays in the ground station. (ground delay)
- The propagation delay from of the signal from the spacecraft to reception at the ground station.
- Any on-board delay between the triggering of the storage of the time packet and the actual radiation of the corresponding frame the radiation delay. (on-board processing/radiation delay)

Subtracting these delays from the ERT, the Frame Transmission Time (FTT) is obtained as following:

FTT = ERT – ground delay – propagation delay – on-board processing delay

In addition to these delays that should be subtracted from the Earth Reception Time there is an additional additive delay which is any delay between the event that triggers the sampling of the on-board clock and the actual time when the OBT is latched in the relevant on-board register (on-board latching delay). These delays are illustrated in the following figure:

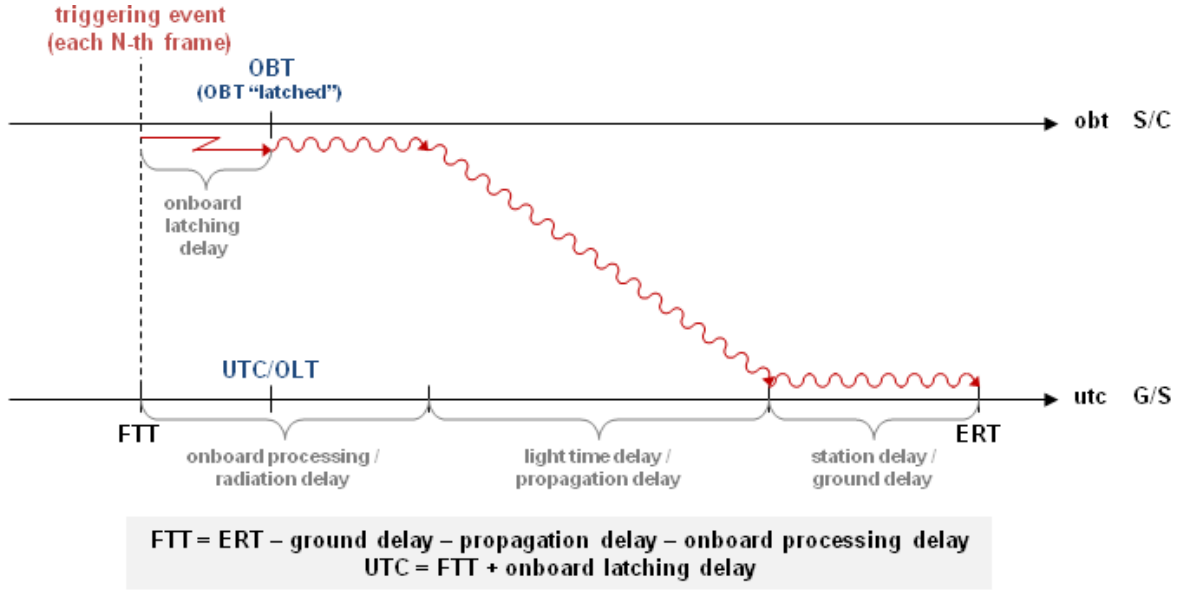


Figure 2 Delays involved in calculating the relation between OBT and UTC

From these four delays the UTC time – termed the On-board Latching Time – can be determined via UTC = FTT + on-board latching delay which corresponds to the on-board time reported in the time packet. Thus we have an on-board time and a ground UTC time for the same event – the latching of the current clock value on board. These two times are referred to as a time couple and a telemetry packet is created by SCOS-2000 to store the time couple; from one or a succession of time couples the relation between the on-board time and UTC time can be derived.

However, because the time packet is not received in same frame to which it refers it is necessary for SCOS-2000 to match the time packet to the frame whose transmission triggered the latching of the time in the time packet. This is done by configuring an interval following the reception of a frame with a frame counter that is a multiple of N where it is expected to receive a time packet. If a time packet is received in this interval then a time couple is created. If a time packet is received outside of this interval then it will be ignored and no time couple generated. This is illustrated in the following figure:

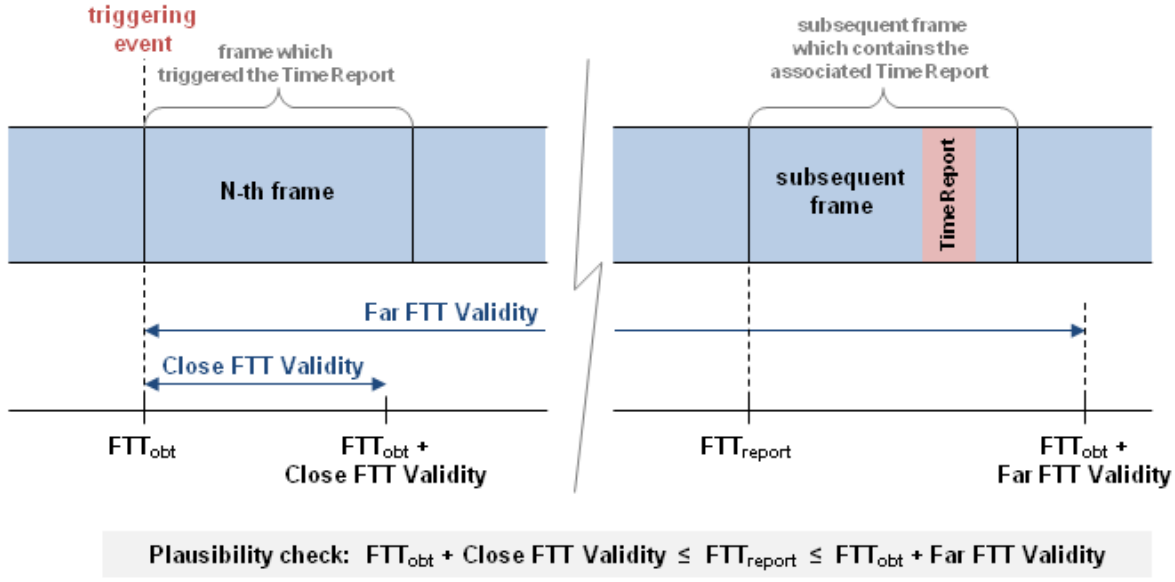


Figure 3 Plausability check of an incoming time report

SCOS-2000 supports two methods for the computation of the time correlation relation. One is based on performing least squares fit over a set of time couples in order to derived two coefficients (gradient and offset) that characterise the relationship. The other mechanism is expected to be used when the on-board time is subjected to synchronisation with an independent provider of a time signal such as GPS. In this case the gradient can be assumed to have the value 1 and it is only necessary to calculate the offset, for which only one time couple is required.

Every time that a time couple packet is created a check is made by calculating the difference between the UTC of the current time couple and the UTC that is obtained from using the current correlation coefficients to convert the time couple OBT to a UTC. This difference is termed the deviation and is used to derive the accuracy and validity status of the coefficients according to whether it is less than or greater than the configured limits for accuracy and validity with the accuracy limit representing the tighter limit – hence coefficients can be declared to be accurate and valid, inaccurate but still valid or inaccurate and invalid.

More formally expressed:

TCO Deviation = UTC (of current time couple)

- UTC (from OBT of current time couple using the current correlation coefficients)

UTC (of current time couple) = ERT - ground delay

- propagation delay - on-board processing delay + on-board latching delay

UTC (from OBT of current time couple using the current correlation coefficients) =

UTC_N + Gradient * (OBT – OBT_N) + Offset *or*

UTC_N + OBT – OBT_N – Offset OBT + Offset

(depending on the algorithm, see 4.1)

INVALID & INACCURATE <=> 'TCO Accuracy' < 'TCO Validity' < **TCO Deviation**

VALID & INACCURATE <=> 'TCO Accuracy' < **TCO Deviation** < 'TCO Validity'

VALID & ACCURATE <=> **TCO Deviation** < 'TCO Accuracy' < 'TCO Validity'

When coefficients are not valid and accurate this is normally a trigger to the user to manually trigger a recalculation of the coefficients based on the latest received time couples. However, the system also supports an automatic mode which will trigger an automatic recalculation of the coefficients whenever the deviation exceeds half the accuracy limit. In this way the system attempts to ensure that there are always valid and accurate coefficients available. When operating in automatic mode the system reacts to the detection of an invalid time correlation by initially discarding the time couple as a rogue. After a configurable number of invalid detections the system decides that the time couples are not rogues and should be used as the basis for a new automatic correlation.

In addition to monitoring the accuracy and validity status of the coefficients it is also possible to configure the system to monitor the synchronisation status. This is for use where the on-board clock is synchronised against an independent time source such as GPS. Such time sources often have an expected offset to UTC time. For example GPS time does not take leap seconds into account so that over time GPS time and UTC have built up an offset of a number of seconds which is a known value. If the on-board clock is synchronised in this way then we expect that the calculated offset correlation coefficient is equal (within a tolerance taken to be the accuracy limit) of this known value. If it is not then this indicates that the on-board clock is no longer synchronised with the time source – this is the meaning of the monitoring of the synchronisation status. Thus it is possible for the time correlation coefficients to be valid and accurate (meaning that they can be reliably used to convert UTC times to on-board times and vice versa) but desynchronised (meaning that the on-board clock is no longer synchronised with its time source).

4.1 ALGORITHM

There are two possibilities:

- **FREE RUNNING CLOCK.** This algorithm performs a calculation based on a least squares fit over a set of ground and on-board time couples and computes a gradient and offset for the best-fit line through them.
- **SYNCHRONISED ON BOARD CLOCK.** This algorithm simply calculates the offset between the ground and on board time of the last time couple.

Thus the gradient only has meaning in "Free running clock" mode and various fields on the display are greyed out when using the "Synchronised clock" algorithm in consequence. This is explained for each affected field later.

The time correlation relation for the "free running clock" (least square algorithm) is:

UTC(OBT) = UTC_N + Gradient * (OBT – OBT_N) + Offset

And for the "Synchronised clock" (difference algorithm) it is:

UTC(OBT) = UTC_N + OBT – OBT_N – Offset

Note that the conversion is done in a relative coordinate system relative to the last time couple used for calculating the time correlation coefficients. This is done to reach a higher time resolution. The relation between the used relative system and an absolute system is shown in the following figure. The given "display offset" corresponds to the offset used throughout SCOS and this documentation. It is stored in a time correlation coefficient packet.

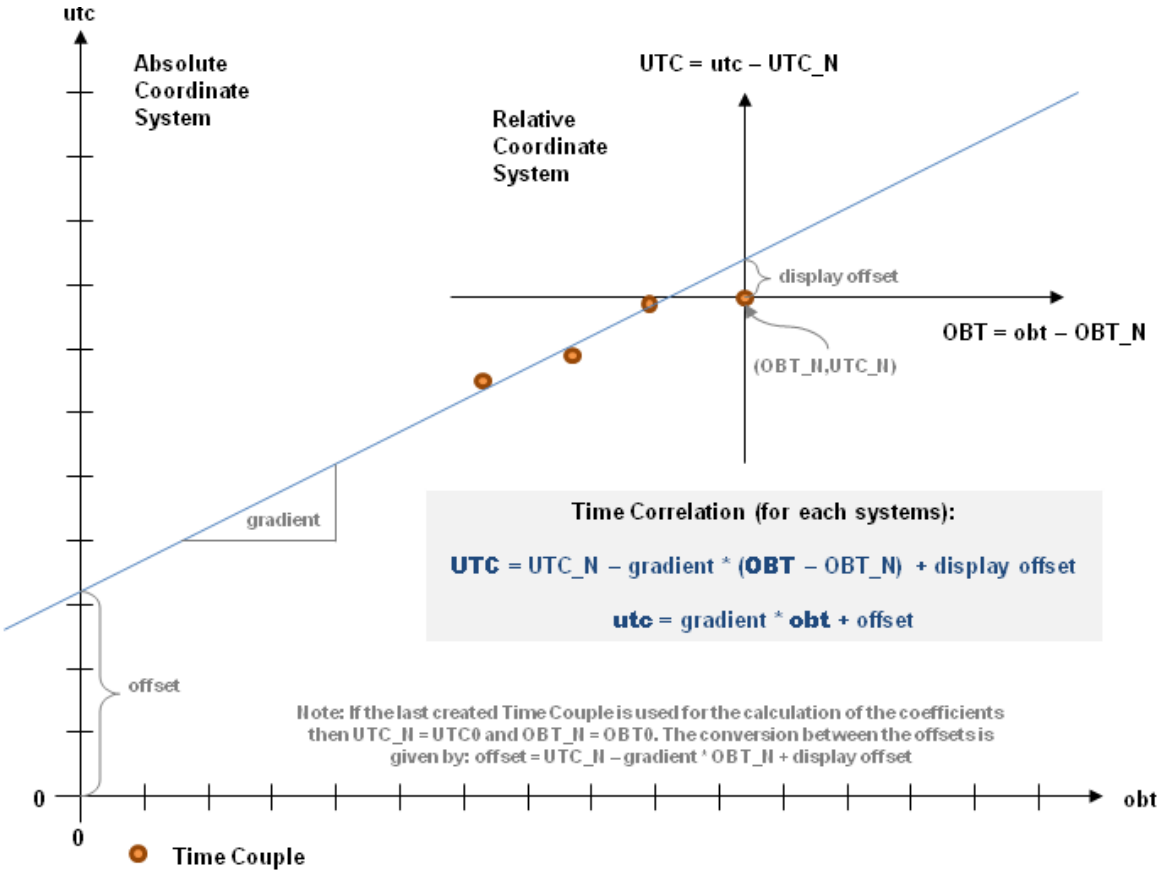


Figure 4 Time correlation conversion in an absolute and relative coordinate system

5 TIME CORRELATION PACKETIZER

The packetiser provides an engine which drives the calculation of time correlation coefficients based on the reception of time packets from the spacecraft as well as requiring time correlation services as described below. All time correlation services are accessed via the time correlation API. Within the API there is an interface to a TCO server process although this is not visible to the packetiser.

Each TM Transfer Frame and TM Source Packet to be filed and/or distributed is associated to a time stamp, which is used as prime filing key in the history files. The SCOS-2000 Packetiser is supporting three different options of TM time-stamping:

- Based on the time when the SCOS-2000 TM Packet is filed in the archive. This option is useful during testing activities e.g. when playing back files of TM Transfer Frames which are time-stamped in the past
- Based on the time when the Transfer Frame containing the header of the unit was transmitted to the ground (FTT, Frame Transmission Time).This is evaluated based on the Earth Reception Time minus the applicable propagation delays (ground delay, propagation delay and onboard processing delay).
- Based on the time when the Packet was generated on-board. This only applies to TM Source Packets containing the On-Board Time (OBT) as part of its data field. SCOS-2000 provides an API for the conversion of the OBT into UTC. The actual conversion routine between OBT and UTC (based on the ground maintained ground correlation) is performed using an API defined in the TCO subsystem.

The mission can be configured to use one of these time-stamping options.

5.1 TIME CORRELATION ACCESS CLASSES

TCOconcreteInstance

The TCOconcreteInstance is a singleton object for the time correlation.

The following public methods relevant to the packetiser exist:

This function gets the singleton object reference.

```
static TCOconcreteInstance & instance();
```

This function destroys the singleton object.

```
  static void destroyInstance();
```

The function obtUtt converts on-board time to ground processing time using the current coefficients. It returns false if the conversion parameters have not been set.

```
  virtual bool obtUttAccuracy(unsigned int obtId,
                              const char* obtRawTime,
                              TDSsunTime_pico &utt) const;
```

The function obtUttAccurary converts on-board time to ground processing time. It returns false if the conversion parameters have not been set. In case of refUtt is omitted, the latest function will be used. It returns as an output parameter the accuracy of the time correlation coefficients used.

```
virtual bool obtUttAccuracy(unsigned int obtId,
                            const char* obtRawTime,
                            TDSsunTime_pico &utt,
                            bool & accuracy,
                            TDSsunTime* refUtt = NULL) const;
```

The function obtUttValid converts on-board time to ground processing time using the OBT as a reference to find the correct coefficients. If coefficients are not valid uses the next valid coefficients and returns false. It returns as an output parameter the accuracy of the time correlation coefficients used.

```
virtual bool obtUttValid(unsigned int obtId,
                         const char* obtRawTime,
                         TDSsunTime_pico &utt,
                         Bool & accuracy) const;
```

The function propagationDelay calculates the propagation delay based on ground UTC and optional auxiliary identifier, which can be used by missions for specifying the ground station ids etc.

It returns false if the calculation fails in any way. The default implementation will just return a propagation delay set by call to 'setPropagationDelay'.

```
virtual bool propagationDelay(double& returnedDelay,
                              double & lightTimeDelay,
                              double & onBoardDelay,
                              double & stationDelay,
                              const TDSsunTime_pico & utc,
                              const unsigned & auxId = 0);
```

The function setCorrelationStatus supports setting the status of the current time correlation calculated when a new time packet is received.

```
virtual bool setCorrelationStatus(unsigned int obtId,
                                  CorrelationStatus validity,
                                  CorrelationStatus accuracy,
                                  CorrelationStatus synchronisation,
```

```
                                    MessageSeverity severity,
                                    const TDSsunTime & createTime = ZERO_TIME,
                                    const TDSsunTime & filingTime = ZERO_TIME
                                    ) const;
```

The function calculateTCOparameters requests the calculation of time correlation coefficients.

```
virtual bool calculateTCOparameters(unsigned int obtId,
                                      const TDSsunTime & toTime = ZERO_TIME,
                                      const TDSsunTime & fromTime = ZERO_TIME
                                     )const;
```

The function confirmTCOparameters requests the confirmation of the last calculated parameters.

```
virtual bool confirmTCOparameters(unsigned int obtId,
                                    const TDSsunTime &validityTime= ZERO_TIME
                                    ) const;
```

The function resetTCOcalculation requests that the current time correlation be reset.

```
virtual bool resetTCOcalculation(unsigned int obtId,
                                  const TDSsunTime &validityTime= ZERO_TIME
```

The function createTimeCouple allows the creation of a time couple packet and triggers the processing related to the reception of a time couple.

```
virtual bool createTimeCouple(unsigned int obtId,
                              bool updateBuffer,
                              const char * obt,
                              const TDSsunTime_pico & ftt,
                              const TDSsunTime_pico & ert,
                              const TDSsunTime & filingTime,
                              const double lightTimeDelay,
                              const double onboardDelay,
                              const double stationDelay,
                              const unsigned int auxId,
                              double deviation = 0.0
                             );
```

The function checkTAIsynchronised checks the synchronisation status of the current coefficients.

```
virtual bool checkTAIsynchronised(unsigned int obtId,
                                  CorrelationStatus& synchronisation,
                                  Double accuracyInterval
                                  );
```

5.2 TIME CORRELATION PACKETS

The time correlation processing produces two types of telemetry packets containing the time couples and associated data and the time correlation parameters. See [AD1] time correlation sections for more details on the meaning and relation of these parameters.

The meaning of physical moments of time and time delays involved in Time Couple Packets (Earth Reception Time, Frame Transmission Time, OBT Latching Time, Light Time Delay, On-board Frame Radiation Delay, and Ground-Station Delay) are described in [AD2], in the time correlation section. The definition of the content of the fields and their relation to the physical moments of time and delays are given below.

Note that *TCO_ESA_EPOCH* is a MISC variable, while the *DEFAULT_ESA_EPOCH* is a default epoch currently hardcoded in the source code, in case the previously mentioned MISC variable cannot be read.

5.2.1 Time Couple Packet

Name	Length in Bits	Position	Description
OBT	64	0	*OBT expressed as TDSsunTime (4 bytes seconds, 4 bytes microseconds), which is a Unix time with an epoch of TCO_ESA_EPOCH (fallback is DEFAULT_ESA_EPOCH).* *On-Board Time (OBT).* ***This field is used internally by the MCS.***
OLT	96	8	*OLT expressed as TDSsunTime_pico(4 bytes seconds, 4 bytes microseconds, 4 bytes picoseconds), which is a Unix time with an epoch of TCO_ESA_EPOCH (fallback is DEFAULT_ESA_EPOCH).* *OBT Latching Time (OLT): is the time when the OBT corresponding to a given Frame Transmission Time is sampled and latched (i.e. stored) in the relevant on-board register.* *This is the value which is coupled with the on-board time, in order to produce 'points' for a least squares algorithm.*
FTT	96	20	*FTT expressed as TDSsunTime_pico(4 bytes seconds, 4 bytes microseconds, 4 bytes picoseconds), which is a Unix time with an epoch of TCO_ESA_EPOCH (fallback is DEFAULT_ESA_EPOCH)..* *Frame Transmission Time (FTT): is the time when the transmission of a telemetry frame is initiated on-board by the Transfer Frame Generator (TFG). For the telemetry frames used for time correlation purposes, this time is signalled by the TFG in order to request the sampling of the On-board Time (OBT) in a dedicated register.*
ERT	96	32	*ERT expressed as TDSsunTime_pico (4 bytes seconds, 4 bytes microseconds, 4 bytes picoseconds), which is a Unix time with an epoch of TCO_ESA_EPOCH (fallback is DEFAULT_ESA_EPOCH).* *Earth Reception Time (ERT): this is the UTC time associated by the relevant equipment in the ground station to a received frame (i.e. the telemetry frame time-stamp). In the case of SLE Service Providers, this is required to correspond exactly to the Frame Reception Time.*

Light Time Delay	64	44	*Speed of light transmission delay (in seconds) expressed as a double, also known as OWLT (one-way light time). Propagation delay of the signal from the spacecraft to the reception at the ground station.*
On board Frame Radiation Delay	64	52	*On-board Frame Radiation Delay or On-board processing (in seconds) expressed as Double.* *This is the time interval between the generation of the signal requesting the* *OBT latching and the actual start of the radiation of the telemetry frame (i.e. it represents the difference between* *the Frame Transmission Time and the Frame Radiation Time)*
OBT Latching Delay	64	60	*OBT Latching Delay (in seconds) expressed as Double.* *This is the time interval between the generation of the signal requesting the OBT latching and the time when the corresponding OBT is sampled and latched in the relevant on-board register (i.e. it represents the difference between the Frame Transmission Time and the OBT Latching Time)*
Groundstation Delay	64	68	*Ground station processing delay (in seconds) expressed as a double*
Groundstation ID	32	76	*Groundstation ID expressed as integer*
Deviation	64	80	*Time Correlation deviation (in seconds) expressed as double.* *Difference between the UTC time of this Time Couple (ERT – Groundstation Delay – Light Time Delay – On-board Frame Radiation Delay + OBT Latching Delay) and the UTC time calculated with the current correlation coefficients from the OBT time of this Time Couple*

Table 1 Time Couple Packet

5.2.2 Time Correlation Packet

Name	Length in Bits	Position	Description
Version	8	0	*Version number. Permits packet structure to be changed in the future. Set to 1 in the original implementation.*
Algorithm	32	1	*Algorithm used to calculate the coefficients.* *1 = DIFFERENCE* *2 = LEAST SQUARE* *The difference algorithm is used for the 'Synchronized clock' and the least square algorithm is used for the 'Free running clock'.*
Synchronisation Check	8	5	*Byte value indicating whether synchronisation status is checked or not 1 = checked, 0 = not checked.*
Validity	8	6	*Validity Status* *0 = INVALID* *1 = VALID* *Invalid means that the deviation has exceeded the user controlled hard threshold*
Accuracy	8	7	*Accuracy Status* *0 = INACCURATE* *1 = ACCURATE* *Inaccurate means that the deviation has exceeded the user controlled soft threshold*
Synchronisation	8	8	*Synchronisation Status* *0 = DESYNCHRONISED* *1 = SYNCHRONISED*
Spare	24	9	*Padding*

Gradient	64	12	*Time Correlation gradient expressed as an IEEE 64 bit double (in UTC seconds/OBT seconds).* *Gradient in the time correlation equation (least squares algorithm). For the difference algorithm the gradient is equal to 1.* *This is the first coefficient in the formula y = mx + c, i.e.* *UTC = Gradient * (OBT – OBT_N) + Offset + UTC_N*
Offset	64	20	*Time Correlation offset expressed as an IEEE 64 bit double.* *The offset in the time correlation equation previously used when the time correlation was calculated in an absolute way via:* *UTC = Gradient * OBT + Offset* *This was the second coefficient in the formula y = mx + c* *Now OBSOLETE since the time correlation is now calculated in a relative way and the Display Offset is used as shown below.* *The conversion between Offset and Display Offset is given by:* *Offset = UTC_N – Gradient * OBT_N + Display Offset*
OBT_N	64	28	*Time Correlation OBT_N expressed as an IEEE 64 bit double number of OBT seconds since the start of the epoch.* *OBT time of the last time couple used in the calculation of the time correlation coefficients (Gradient & Display Offset) stored in this packet.*
UTC_N	64	36	*Time Correlation UTC_N expressed as an IEEE 64 bit double number of UTC seconds since the start of the epoch.* *UTC time of the last time couple used in the calculation of the time correlation coefficients (Gradient & Display Offset) stored in this packet.*

Display Offset	64	44	*Time Correlation display offset expressed as double* *The offset in the time correlation equation currently used where the time correlation is done in a relative way via:* *UTC = UTC_N – Gradient * (OBT – OBT_N) + Display Offset*

Table 2 Time Correlation Packet

6 TIME CORRELATION SYSTEM & ARCHITECTURAL DESIGN

6.1 TIMES AND DELAYS USED IN TIME CORRELATION

a) Frame Transmission Time (FTT): this is the time when the transmission of a telemetry frame is initiated on-board by the Transfer Frame Generator (TFG). For the telemetry frames used for time correlation purposes, this time is signaled by the TFG in order to request the sampling of the On-board Time (OBT) in a dedicated register

b) OBT Latching Time (OLT): this is the time when the OBT corresponding to a given Frame Transmission Time is sampled and latched (i.e. stored) in the relevant on-board register

c) Frame Radiation Time: this is the time at which the signal event corresponding to the leading edge of the first bit of the Attached Sync Marker that immediately precedes this telemetry frame is presented at the phase center of the on-board antenna used to radiate the frame

d) Frame Reception Time: (definition taken from SLE) this is the time at which the signal event corresponding to the leading edge of the first bit of the Attached Sync Marker that immediately precedes this telemetry frame is presented at the phase center of the antenna used to acquire the frame

e) Earth Reception Time (ERT): this is the UTC time associated by the relevant equipment in the ground station to a received frame (i.e. the telemetry frame time-stamp). In the case of SLE Service Providers, this is required to correspond exactly to the Frame Reception Time

f) On-board Frame Radiation Delay: this is the time interval between the generation of the signal requesting the OBT latching and the actual start of the radiation of the telemetry frame (i.e. it represents the difference between the Frame Transmission Time and the Frame Radiation Time)

g) OBT Latching Delay: this is the time interval between the generation of the signal requesting the OBT latching and the time when the corresponding OBT is sampled and latched in the relevant on-board register (i.e. it represents the difference between the Frame Transmission Time and the OBT Latching Time)

h) Propagation Delay: this is the One-Way-Light-Time (OWLT) between the spacecraft and the ground antenna. It represents the time required by the frame on the space link to reach the ground antenna following radiation from the spacecraft (i.e. the difference between the Frame Radiation Time and the Frame Reception Time)

i) Ground Delay: this is the difference between the ground UTC time-stamp of the telemetry frame (i.e. the Earth Reception Time) and the time when the first bit of the Attached Sync Marker of a frame is started to be received by the ground antenna (i.e. the Frame Reception Time). In the case of SLE Service Providers, this delay is by definition 0 since the correction is already performed by the provider;

j) UTC Time of Transmission (UTT): this expression is commonly used when discussing time correlation and is used to indicate the UTC time corresponding to a given OBT. The origin of the term assumes that the latching delay is

zero and so the OBT time in a time report relates to the FTT expressed in UTC – hence UTT. With the inclusion of the latching delay the term is erroneous and somewhat misleading; it is nevertheless common.

This is shown in the following figure:

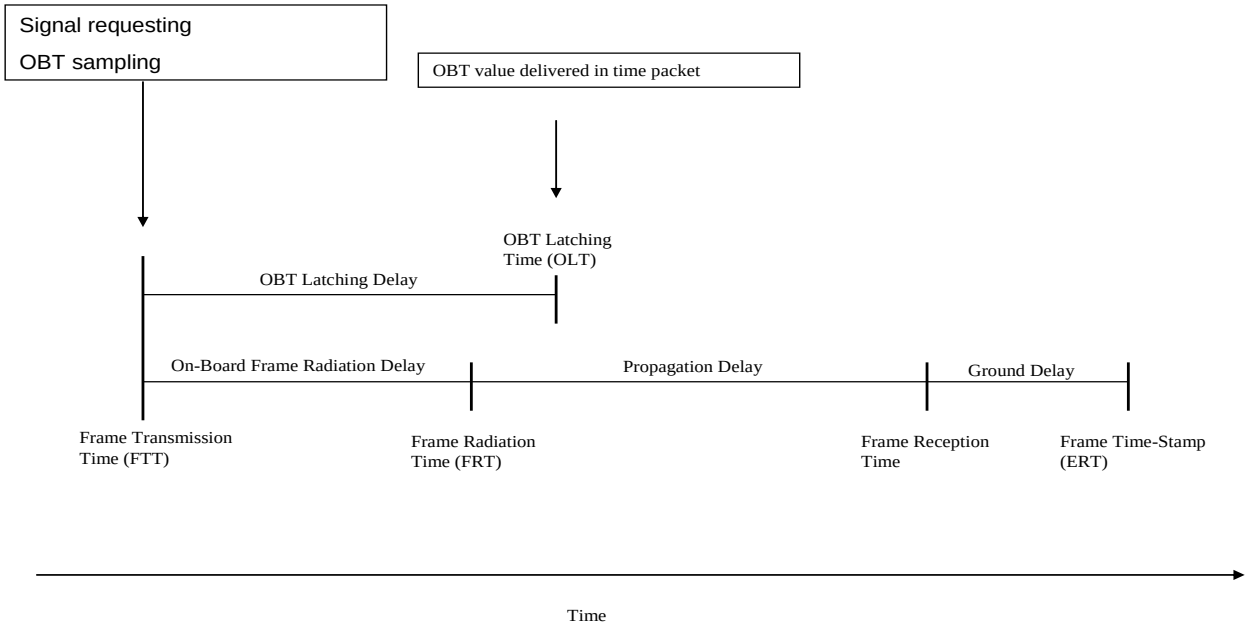


Figure 5 TCO Time and Delays used in Time Correlation Calculation

The time couples used in the time correlation calculation therefore relate the OBT reported by the spacecraft within time packets with the OLT time which is derived from the ERT stamped on the telemetry frames at the groundstation. The OLT time is derived based from the ERT by subtracting the ground delay, the propagation delay and the on-board frame radiation delay and adding the latching delay.

6.2 METHODS OF TIME CORRELATION

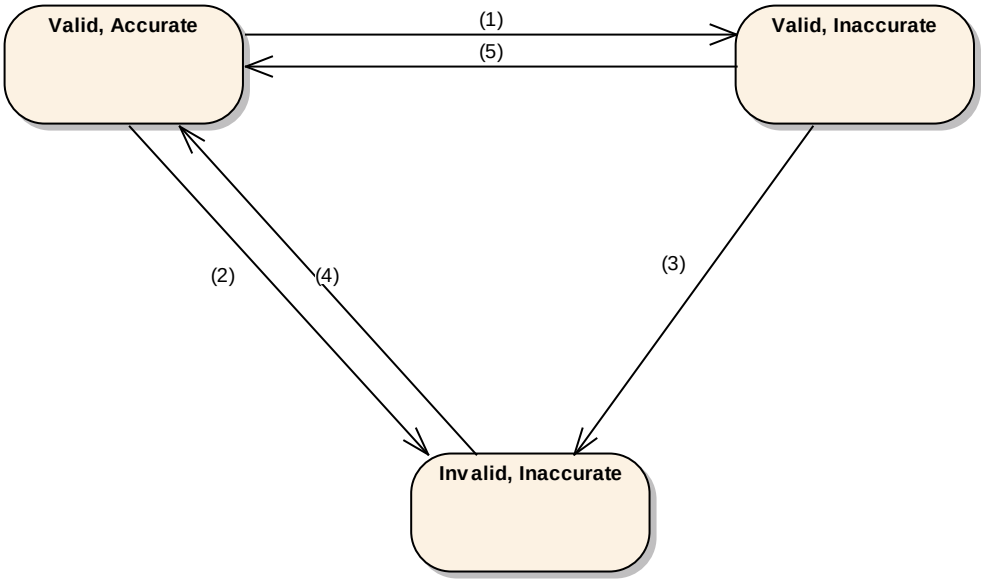
SCOS-2000 supports two methods of time correlation based on whether the mission maintains an independently synchronised on-board clock or not. If so the object of the time correlation function is to calculate the simple correction between the on-board and on-ground time and to monitor the synchronisation status. If the on-board clock is not independently synchronised than a least-squares fit is used to compute time correlation coefficients. The “difference” mode and “least-squares” mode can be switched between at run-time and the monitoring of the synchronisation status can also be performed in “least-squares” mode.

Time correlation coefficients are calculated based on the latest available time couples with the calculation able to be triggered automatically or manually. The status of the current time correlation coefficients is monitored for each time couple received to determine whether defined accuracy or validity thresholds are violated. This is computed by comparing the deviation (the difference between the OLT time within the time couple and the UTC time obtained by applying the current time coefficients to the OBT within the time couple) with configurable limits for accuracy and validity.

In automatic mode new time correlation coefficients are calculated whenever the accuracy exceeds half the accuracy limit to attempt to maintain an accurate correlation in the event of drifts of the onboard clock. If the “least squares” mode is used this implies that a least squares fit is performed over the last set of time couples stored in a buffer.

The validity threshold is intended to determine when the current coefficients are so bad as to be no longer applicable – for instance if the onboard clock experiences a jump. In this case, in automatic mode, a configurable number of invalid time couples triggers a reset of time correlation. This means that any time couple buffer is emptied and the system will collect time couples received after the reset and perform a new calculation at the first opportunity.

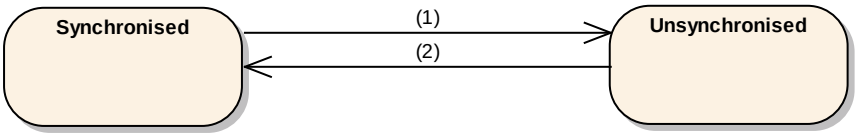
The following state diagram shows the possible states of the time correlation in terms of the validity and accuracy of the coefficients and shows the events that trigger transitions between these states.



(1) Time Couple received whose deviation exceeds accuracy limit
(2) Time Couple received whose deviation exceeds validity limit OR time correlation reset performed either manually or automatically
(3) Time Couple received whose deviation exceeds validity limit OR time correlation reset performed either manually or automatically
(4) Time correlation performed and new coefficients committed.
(5) Time correlation performed and new coefficients committed.

Figure 6 TCO Accuracy & Validity State Diagram

Each time that time correlation coefficients are newly generated the synchronization status of the new coefficients is assessed. Dependent on the external time reference there is an expected offset between the onboard clock and UTC. If the offset calculated by the time couples is within the accuracy limit of this value then the coefficients are said to be synchronized. If not they are unsynchronized. The following state diagram shows the transitions between the synchronization states. Note that unlike the accuracy and validity case the synchronization status is only assessed when new correlation coefficients are generated and not based on the reception of time couples.



(1) Time correlation performed resulting in difference between offset and expected offset greater than accuracy limit
(2) Time correlation performed resulting in difference between offset and expected offset less than accuracy limit

Figure 7 TCO-synchronization State Diagram

6.3 TCO ENGINE

The purpose of the TCO engine is to support the time correlation functionality of the telemetry packetiser in a highly configurable way.

There are two main elements to the time correlation functionality within the packetiser. The first is the detection of the frames received that trigger the generation of time-packets by the on-board software and whose time therefore provides the on-ground time corresponding to the on-board time report in the subsequently received time packet. The second is the processing of time packets and the creation of time couple packets to be stored in the archive for the purposes of a later time correlation calculation.

A typical mission produces time packets containing the sampled time of the leading edge of each Nth telemetry frame. The packetiser thus needs to detect this frame and store the corresponding time. When the subsequent time packet is received this time and the time contained in the time packet form the time couple.

In order to handle more complex scenarios than that described above such as multiple on-board clocks the TPKTMtcoEngine class is assisted by two additional classes. One describes the rules for detecting the frames for which the time packet applies. This is typically done by specifying the VC IDs and data channels that such frames are received on and by specifying the corresponding sampling frequency This class may be sub-classed to define different rules than those present in the default class. The other helper class describes the on-board clock including the format of the time reported in the time packet and the location of the time for the clock within the time packet. Again this may be sub-classed to cope with, for instance, scenarios where different time packets are used for different on-board clocks.

The rules and the on-board clocks are mapped together such that each on-board clock is associated to a single rule but one rule may be associated to many on-board clocks.

Thus every time a frame is received it is checked to see if any of the defined rules indicates that the frame time must be stored and each time a time packet is received it is checked whether this is related to a defined rule and if so time couples are created for every on-board time mapped to that rule.

A TPKTMtcoConfig singleton class is used to configure the time correlation system – creating and specifying the rules and on-board clock definitions and mapping them together. Thus in most cases mission configurations of time correlation functionality will consist of replacing this class only. For more complex cases subclassing, particularly of the class defining the rules, will be necessary.

6.3.1 System Design

6.3.1.1 TPKTM

6.3.1.2 TPKTMtcoTimePktRule

This class is used to specify the rules by which to recognise telemetry frames for which time packets containing the OBT of their generation is expected.

6.3.1.3 TPKTMtcoObt

This class is used to characterise the on-board times in use by the mission

6.3.1.4 TPKTMtcoConfig

This class is used to configure the time correlation functionality. Defining the rules and the on-board clocks and mapping them together.

6.3.1.5 TPKTMtcoEngine

This class contains the mains bones of the Packetiser time correlation functionality. It contains methods that are called for each telemetry frame received in order to identify whether any of the defined rules identify that this frame is one for which a time is sampled on-board.

A method is supplied to be called each time a time packet is received. In this case it is checked whether it relates to a defined rule and if so a plausibility check is performed to determine that the packet does correspond to the last frame identified by the rule. If so, for each OBT mapped to the rule the OBT is extracted from the packet in the applicable format (CUC and CDS formats are supported) and a time couple packet generated and sent to the PARC. The detailed functionality is described in section 5.

It is possible to create a TPKTMtcoEngine in “offline” mode. In this mode all functions related to the detection of time packets and the creation of time couples is available but none of the functionality which relates to the current time correlation coefficients (such as tracking the number of deviations received and triggering an automatic reset of coefficients etc). It is expected that this mode be used for the processing of historical telemetry – for instance the generation of more accurate time-couples taking into account accurate propagation delays.

6.3.2 Component Description

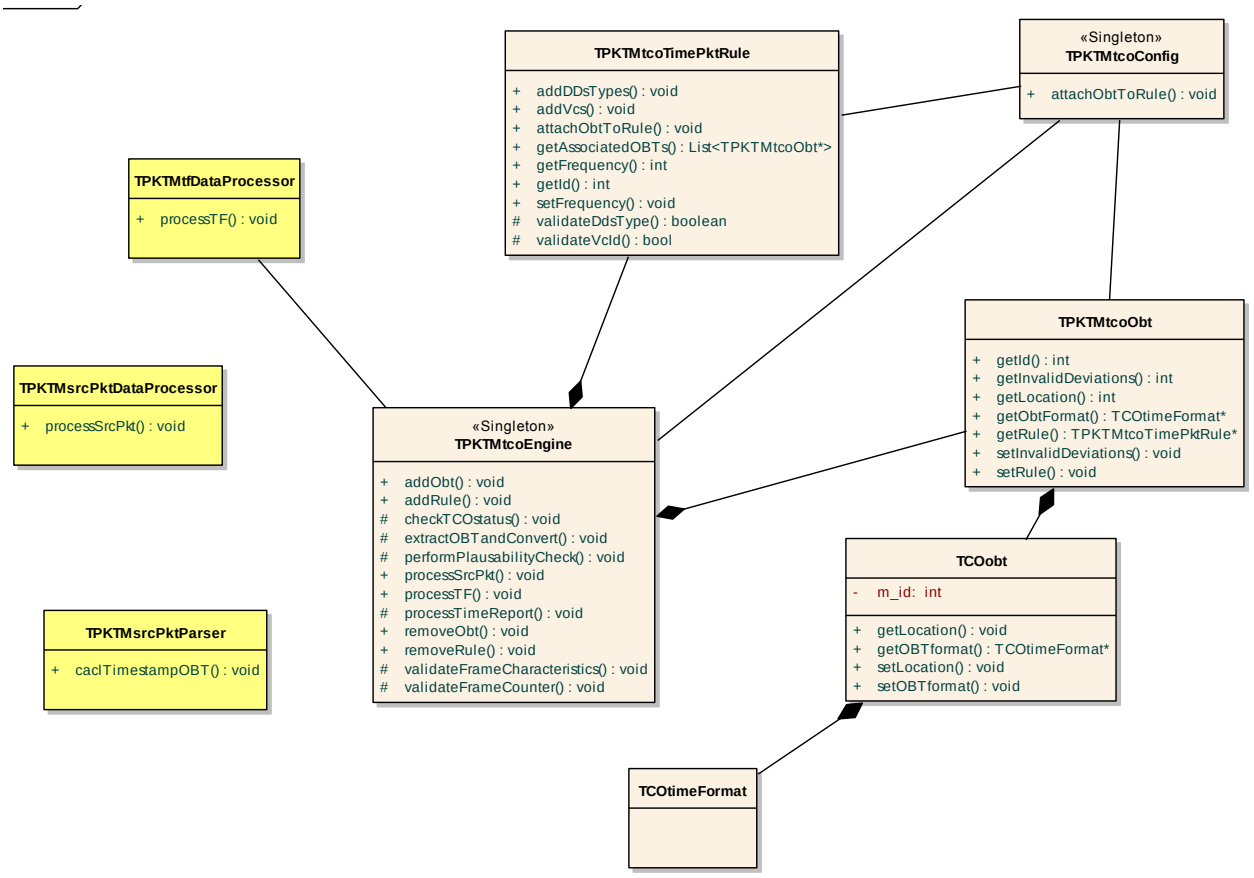


Figure 8 TCO Engine Component Diagram

6.3.2.1.1 TPKTMtcoEngine

The TCO engine is a class which is used within the packetiser to drive the time correlation processing. It works in co-ordination with the TCO server. The distinction between tasks undertaken by the TCO engine and the TCO server is explained in each step below.

Every time the TPKTMtfDataProcessor::processTF() method is called the processTF() method is called within the TPKTMtcoEngine. This checks whether the frame satisfies the rules used to detect the frames used on board the spacecraft for timestamping (e.g. real time VC0 data with VCFC % 32 = 0). If so the frame time is stored within for each rule matched for later use when the associated time packet is received. There is no communication with the TCO server at this stage.

The processSrcPkt() method is called each time TPKTMsrcPktDataProcessor::processSrcPkt() detects a time packet. For each of the defined rules it is checked that the time-packet corresponds to the rule and if so the following functionality is executed for each OBT mapped to the rule:

(1) Apply the Plausibility check to the FTT associated to the OBT of the Time Report packet received. This check consists of determining whether the candidate FTT lies within a dynamically configurable interval with respect to the FTT containing the current Time Report packet. More formally expressed:

FTTreport – FarLimit <= FTTobt AND FTTreport – CloseLimit >= FTTobt

- FTTobt: Candidate FTT (associated to the applicable rule).
- FTTreport: FTT of the frame containing the Time Report source packet.
- FarLimit: MISCdynamic var 'TCO_FAR_FTT_PLAUSIBILITY_LIMIT'.
- CloseLimit: MISCdynamic var 'TCO_CLOSE_FTT_PLAUSIBILITY_LIMIT'.

This is performed internally by the engine with no reference to the TCO server.

(2) If the FTT plausibility check is not passed, then no further action is taken (neither the Time-Couple is generated, nor the Time Correlation coefficients and status are updated). Whether or not the plausibility check is applied is determined by a MISC variable set via the Time Correlation application for each OBT.

(3) Extract the OBT from the Time Report packet, in the defined format. Create the OBT/UTC time couple using the function 'createTimeCouple' provided by the 'TCOconcreteInstance' class, passing the extracted OBT and its associated Onboard Latching Time. TCOconcreteInstance supplies the client side interface to the server. All the information required for the time couple is generated by the TCO engine. This is supplied to the TCO server which stores the time couple packet.

(4) Convert the extracted OBT into UTC using the 'obtUtt' member function provided by the 'TCOconcreteInstance' class to convert OBT into UTC, with no reference time. This is performed by the TCO engine with no reference to the server.

(5) Compare the obtained UTC with the Onboard Latching Time and check whether the difference (deviation) falls within the validity and accuracy time intervals defined in the 'TCO_VALIDITY_INTERVAL' and 'TCO_ACCURACY_INTERVAL' MISCdynamic variables respectively. If any change in the status of the time correlation is detected, call the function 'setCorrelationStatus' provided by 'TCOconcreteInstance' to set and file the new status, as well as raising an alarm / info message depending on the transition (go out or inside of limits). This implies a communication with the TCO server.

Note that when the Time-Correlation Auto-Update feature is enabled ('TCO_AUTO_UPDATE' MISCconfig variable), the validity and accuracy checks will only be performed if the 'TCO_SUSPEND_VALIDITY_CHECK' MISCdynamic variable is set to false (the former checks are suspended when a Reset Time Correlation operation is executed). For the manual time correlation calculation mode, those checks will always be done.

(6)In case Time-Correlation Auto-Update feature is enabled ('TCO_AUTO_UPDATE'), and the validity check has not been suspended ('TCO_SUSPEND_VALIDITY_CHECK') apply the Auto-Update check:

- Auto-Update If (Deviation <= Validity Interval) AND (Deviation > Accuracy Interval). If passed, trigger the automatic calculation of time correlation coefficients and status update. The triggering is performed via a call to the TCO server which is responsible for the actual calculation of the coefficients and the storage of the coefficient packet.

(7) In case Time-Correlation Auto-Update feature is enabled ('TCO_AUTO_UPDATE'), and the validity check has not been suspended ('TCO_SUSPEND_VALIDITY_CHECK'), check for the Auto-Reset condition:

- Auto-Reset If (Deviation > Validity Interval) for 'N' consecutive times

If this condition holds true, then Time-Couples will continue to be processed (calculating the deviation of the current TCO) but won't go into the buffer for recalculation purposes. After a configurable number 'N' of consecutive invalid deviations, defined by means of 'TCO_AUTORESET_INVALID_DEVIATIONS', the TCO will be automatically reset (by calling the 'resetTCOparameters' function of the 'TCOconcreteInstance' API). As a result, the Time-Couples buffer will be emptied, and the Auto-Update feature will be temporarily disabled (by setting variable 'TCO_SUSPEND_VALIDITY_CHECK' to true), just until two new Time Reports have been received and their associated Time-Couples put into the buffer, proceeding then to the automatic calculation of the Time Correlation. Note that the user can also manually perform the reset of the Time-Couples buffer if in the automatic time correlation calculation mode.

The triggering of the reset is also performed via a call to the TCO server.

In brief the TCO engine stores the time stamps of telemetry frames used on board for timestamping and when subsequence time packets are received which certain plausibility checks the engine requests the TCO server to create a time couple packet. For each time packet that meets the plausibility criteria the engine determines the accuracy and validity of the current parameters and informs the TCO server if these differ from the previous values. If the deviation exceeds half of the accuracy limit and automatic mode is being used the engine requests the TCO server to recalculate the coefficients. Also in automatic mode, if a greater than configured number of invalid time packets are received the engine requests the server to reset the time correlation. Figure 9 TCO Engine & TCO Server Sequence Diagram shows the interactions between the TCO Engine and the TCO server in the form of a sequence diagram.

Note that each MISCconfig variable referenced in this section is to be understood as being duplicated for each OBT supported by the system.

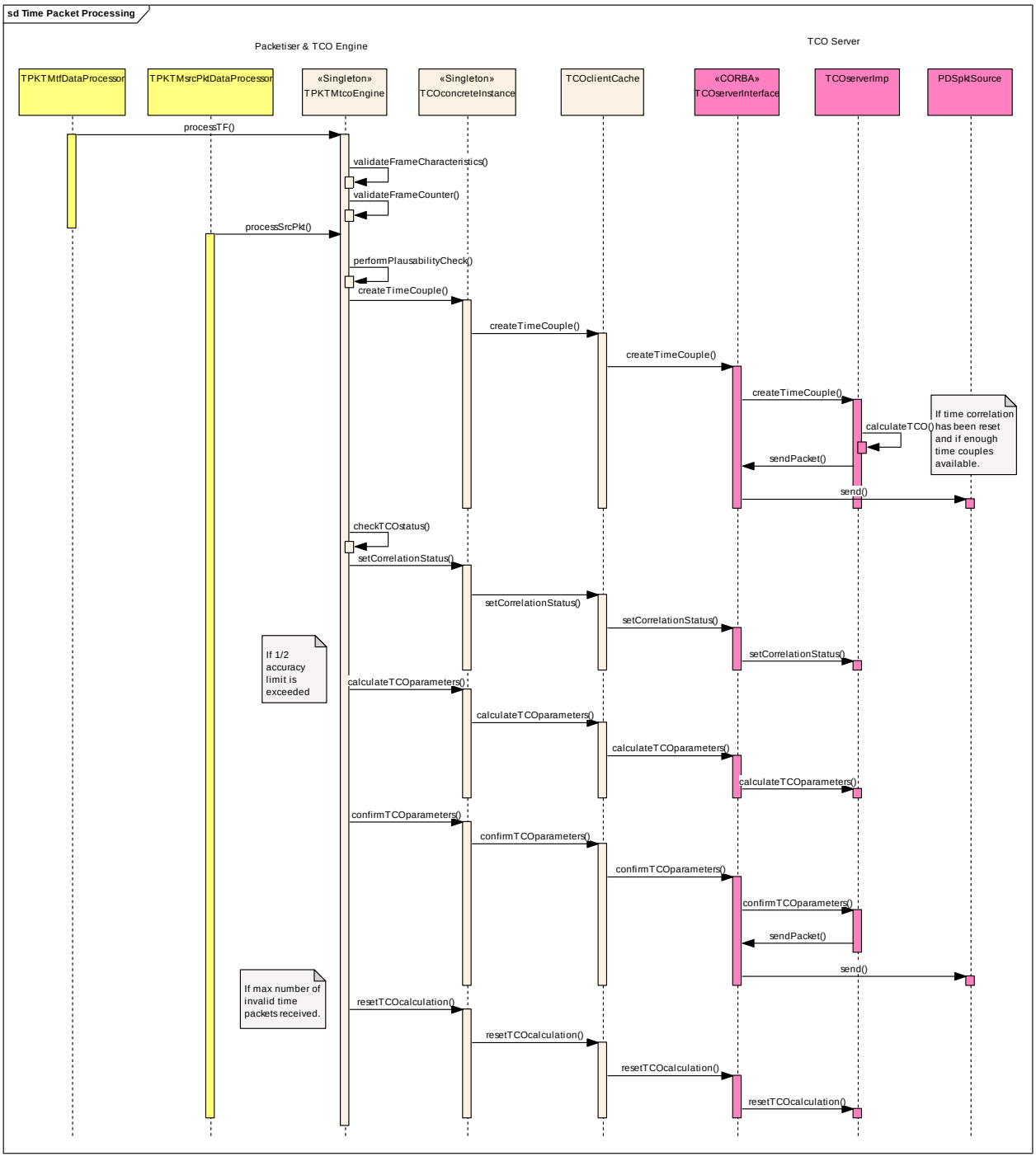


Figure 9 TCO Engine & TCO Server Sequence Diagram

6.4 TIME CORRELATION CLIENT (LIBRARY)

6.4.1 System Design

The time correlation library is based on a singleton class TCOconcreteInstance that provides time conversion services for client applications as well as providing the means for communication with the TCO server – for instance requesting the storage of a time-couple or a computation of TCO coefficients.

Since time correlation services are used throughout the system the default functionality of this class implemented in an abstract class TCOfunction whereas TCOconcreteInstance is the default implementation used throughout SCOS-2000 code.

The TCO system on the client and the server side is configured using a configuration class TCOconfig which creates TCOobt objects for each on-board clock in use on the spacecraft. Each OBT is represented using a TCOtimeFormat object that represents the time format used by the clock. SCOS-2000 supports CUC and CDS formats – missions can create their own formats based on this class.

Based on the configuration specified in the configuration class TCOconcreteInstance creates one TCOclientCache object for each OBT for which time correlation is monitored by the system. Each request for time correlation services must specify the identifier of the OBT and is thus forwarded to the appropriate TCOclientCache object which either forwards the request on to the TCO server or services the request based on its own cache. The reason for the name “clientCache” is that each object maintains a cache of recent coefficients in order to minimize requests to the TCO server. This mechanism is discussed further later.

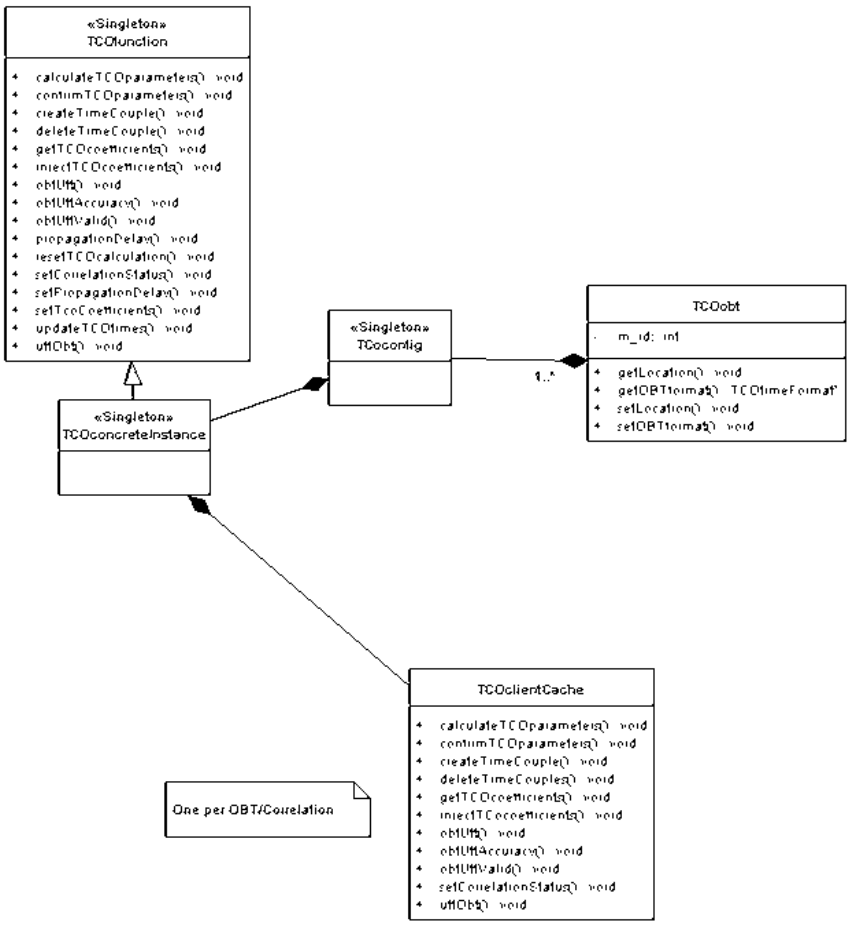


Figure 10 Classes involved in TCO client-side processing

6.4.2 Component Description

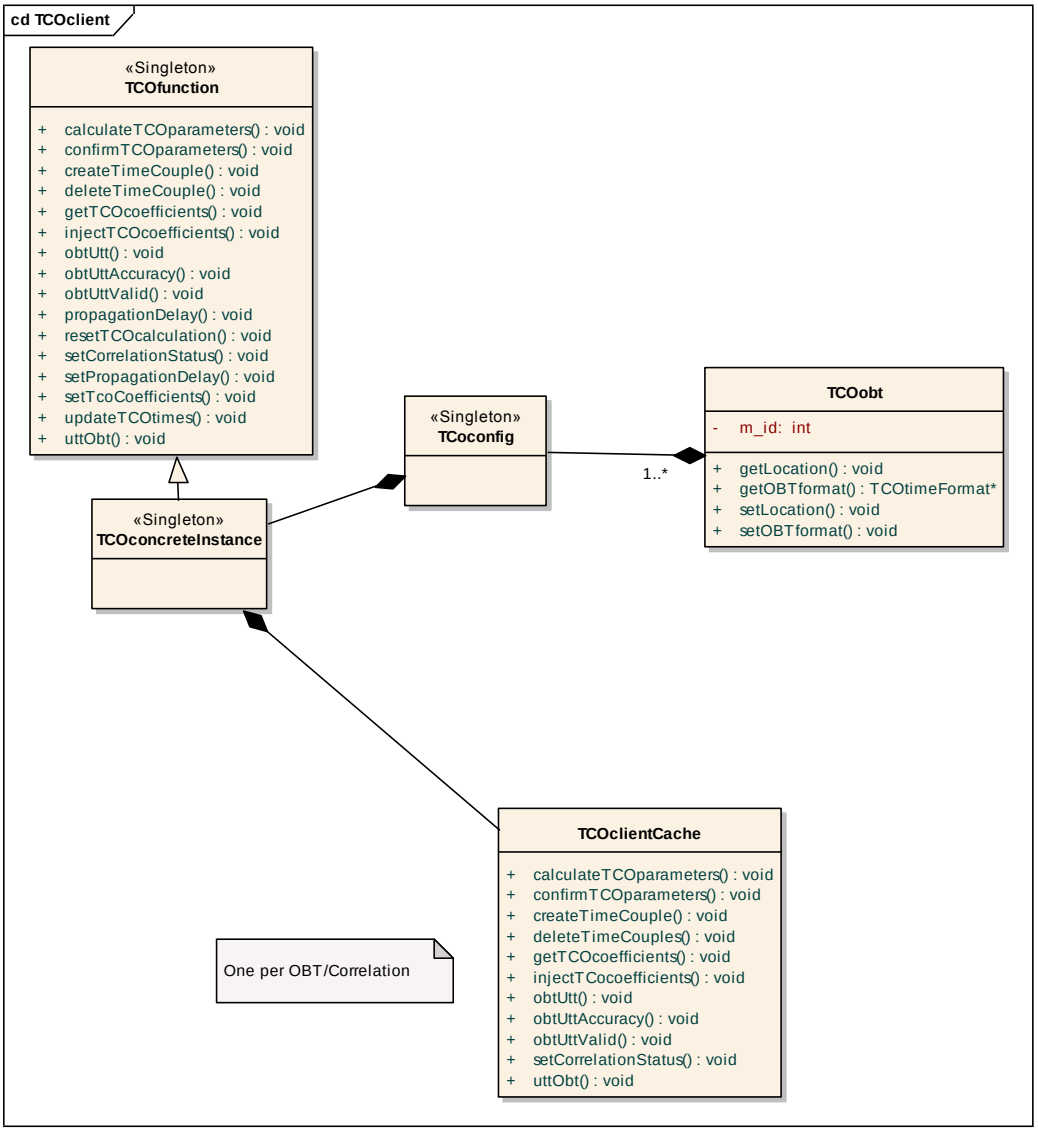


Figure 11 Classes involved in TCO client side processing

6.4.2.1 TCOconcreteInstance/TCOfunction

TCOfunction is an abstract class which defines the interfaces for time correlation conversion operations. The 'TCOconcreteInstance' class provides the SCOS-2000 implementation of these methods. Both these classes are implemented as singleton classes.

TCOconcreteInstance is only used within SCOS-2000 by classes which are directly involved in the time correlation computations. Classes which merely require time conversions reference an instance of TCOfunction. Within the SCOS-2000 implementation TCOfunction::instance() use the method TCOfactory::createTCO to which creates or returns a pointer to a TCOconcreteInstance object. If missions wish to implement an alternative time correlation strategy then they need to create their own child class of TCOfunction and ensure that this is create by the TCOfactory::createTCO method.

Based on the configuration defined in the TCOconfig class the TCOconcreteInstance singleton object creates one TCOclientCache object per OBT.

The main functions implemented by the class are:

TCOconcreteInstance& **instance**(): It returns a singleton object's reference.

void **destroyInstance**(): It destroys the singleton object.

bool **isTcoServerAvailable**(): It tells whether the TCO CORBA Server is available or not.

The other functions are discussed in the context of the TCOclientCache class. Function calls to TCOconcreteInstance specify the identifier of the OBT. The functionality is then delegated to the relevant TCOclientCache object.

6.4.2.2 TCOclientCache

The TCOclientCache provides an OBT specific interface to the TCOserver processes.

It keeps the last as well as the last "next valid" (time correlation coefficients retrieved from the TCO Server, along with their validity intervals. This information is maintained in order to minimise the number of retrievals from the server. If a request is done for a historical time correlation coefficient that is within the interval of the last accessed one, then it is satisfied without asking the TCO server. This is necessary for performance reasons.

The main functions implemented by this class are described in the TCOserver section. Apart from where local caching is used the function is simply to forward the request to the TCOserver.

6.4.2.3 TCOobt

The TCOobt class provides information for each OBT maintained by the spacecraft including the format of the on-board clock and the location of the OBT in time packets.

6.4.2.4 TCOtimeFormat

This is an abstract base class describing time formats used for the onboard clocks. It defines functions related to the clock format – for instance convert the binary time in the defined format into UTC time. This class can be sub-classed by mission to support formats that are not supported by SCOS-2000.

6.4.2.5 TCOtimeFormatCUC

This class is a specialisation of the TCOtimeFormat for handling CUC time formats. The class is parameterised by the PFC from which the different CUC formats can be derived.

6.4.2.6 TCOtimeFormatCDS

This class is a specialisation of the TCOtimeFormat for handling CDS time formats.

6.5 TIME CORRELATION SERVER

6.5.1 System Design

The time correlation server task is responsible for all the computations required for the computation of time correlation for all the OBTs supported by the spacecraft. The task provides a CORBA interface for requests from client applications which is maintained by the class TCOserverInterface. Within SCOS-2000 this interface is not accessed directly but via the TCO client library.

In a similar way to the mechanism used for the client library and described above the TCOconfig defined configuration is used to create one TCOserverImpl object for each OBT on the system which provides services for that OBT only.

The full functionality of the TCOserver is discussed later but includes:

- The production and distribution of time couples packets
- The calculation and confirmation of time correlation coefficients - either current or historical
- The deletion of no longer valid time-coefficients
- The injection of time correlation coefficients
- The retrieval of time correlation coefficients.

6.5.2 Component Description

The TCO server is a resident process that accepts requests through the ‘TCOserver’ CORBA interface from client applications via the TCO client library.

The process does not operate standalone but in coordination with the TCO Engine described in **Error! Reference source not found.** The TCO server is responsible for the production and storage of all telemetry packets related to the time correlation – the time couples and time correlation coefficients packets and is responsible for all calculations of TCO coefficients based either on the contents of time couples supplied to it and stored in its internal buffer or retrieved from the archive in case a request is made to perform the calculation using historical coefficients.

The TCOserver also plays a role in the conversion on board times and UTC where historical coefficients are required since these must be retrieved from the archive.

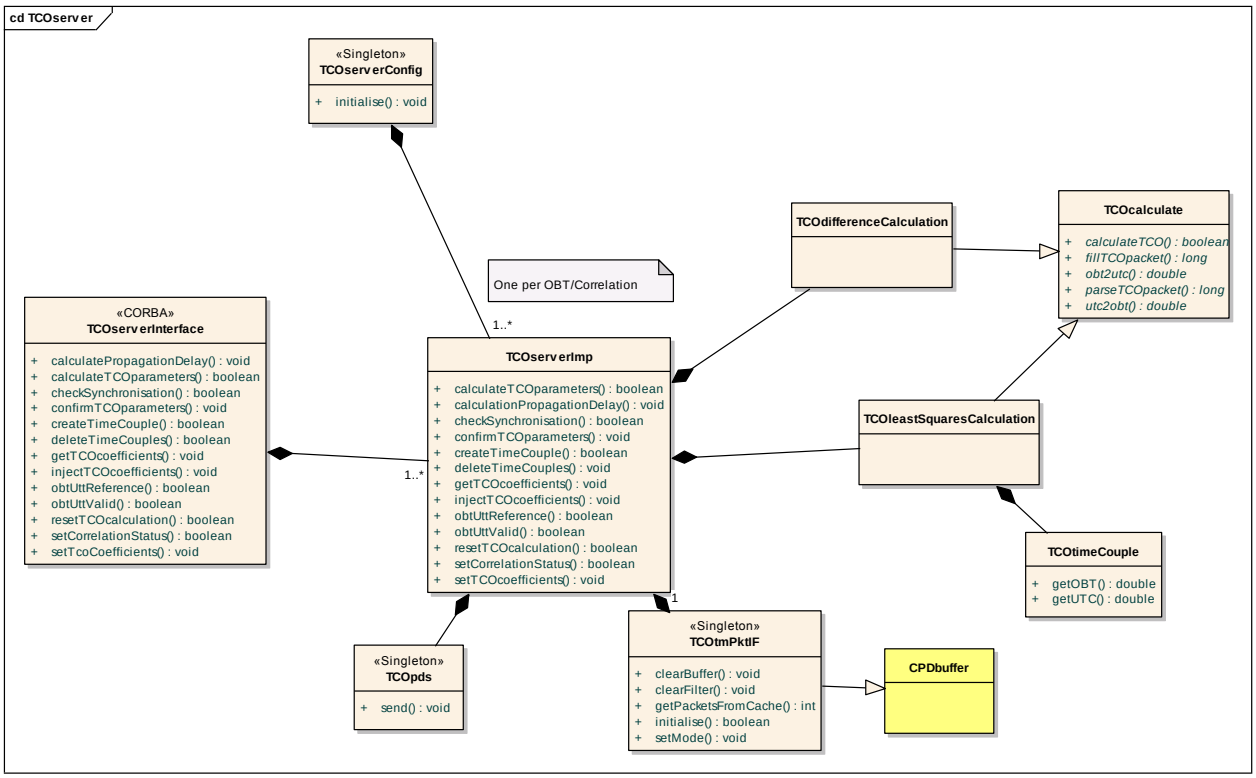


Figure 12 Class diagram for TCO server process

6.5.2.1 TCOserverInterface

This class provides the CORBA interface to the TCOserver process. Each interface function specifies the identifier of the on-board clock to which the request pertains. A TCOserverImpl object is created for each OBT supported by the spacecraft and the implementation of the interface functions are delegated to the specific TCOserverImpl object for that OBT.

6.5.2.2 TCOserverImpl

This class provides the server TCO processing for a single OBT.

The time correlation process supports two types of correlation that can be switched between dynamically. These are a simple correction and a least squares mechanism. In the first case it is envisaged that the on-board clock is synchronized, for instance by a GPS device. In this case it is only required to compute the correction between the time to which it is synchronized and UTC time. (These are not necessarily the same, for instance GPS time does not observe leap seconds). If the on-board clock is not synchronized in this way it is usual to perform time correlation based on a least squares fit over a set of recently received time couples.

In both cases a gradient and an offset is calculated for the correlation coefficients although in the case of the correction mode the gradient is fixed to one. The coefficients are inserted into a SCOS-2000 packet and archived.

Each time a new time couple is received the validity and accuracy of the current coefficients are monitored. This is performed by using the current coefficients to convert the OBT within the time couple to a UTC and comparing this

to the UTC within the time couple. Thresholds are configured to determine whether the coefficients are valid and accurate where accurate is a stronger condition then valid.

It is also possible to configure via a MISCconfig variable whether to monitor the synchronization status of the on-board clock. It is expected that the on-board clock differs from UTC by a fixed amount where the clock is synchronized to another time source. Thus even though time correlation coefficients have been computed and checked to be valid and accurate it is possible that the OBT is not actually synchronized. This is therefore independently monitored. (Note that this functionality is present within the TPKTMtcoEngine of the telemetry packetiser).

The TCOserverImpl maintains a buffer of time couples which are stored in the buffer as they are created following the reception of a time packet containing data for the OBT in question. The contents of the buffer can then be used to calculate the current time correlation coefficients either based on a request received via CORBA or automatically according to the configuration defined by MISCconfig variables.

If a manual request is made to calculate time correlation coefficients then in the most straightforward case this is performed in the same way as the automatic case although an explicit confirm is required before the new coefficients are stored. In other words the manually triggered calculation produces a set of intermediate coefficients which are activated by an explicit confirmation. In the automatic case this confirmation is also automatic.

However in the manual case it is also possible to specify a time range for the couples to be used for the computation. If only an end time is supplied then the configured number of couples prior to that time is used for the calculation. If an end time and a start time is specified then all the couples present in the archive between those two times are used. Clearly in difference mode only the couple immediately prior to the end time is relevant. This calculation uses a dedicated time couple buffer – i.e. the “live” buffer is maintained for the default case where coefficients are to be generated based on the latest applicable data.

Also in manual mode it is possible to specify the validity time when confirming the coefficients. This defines the filing time of the coefficients packet. In automatic mode the validity time is “now”.Haved a

In the automatic case it is defined how many invalid time couples will trigger an automatic calculation of time correlation. The validity of the current time coefficients are determined by whether the difference between the UTC contained in a time couple packet differs by more than a configurable amount from the value obtained by converting the OBT to UTC using the current coefficients. If so the current time correlation is declared invalid. Once more than a configurable number of invalid time couples is received the buffer is reset and the time couples collected once more. Once there are sufficient (as defined by another MISCvariable) entries in the buffer the new time correlation coefficients are recomputed.

Apart from the calculation of time correlation coefficients the TCO server is responsible for the conversion of OBT to UTC and vice versa for historical data. This requires accessing the archive to obtain the time-coefficients applicable for a reference time specified in the request. The look up is based on the an OBT which can lead to problems if the OBT is reset – for instance during simulations. This can result in many coefficients packets valid for similar OBT periods. In order to avoid this scenario, whenever a new coefficient packet is created all coefficient packets in the archive which relate to an OBT within the period where the last generated coefficients were valid will be deleted.

The functionality of the TCOserverImpl class is further discussed through the main operations that it provides are:

bool **initialiseTCOparameters**(bool cacheInit): It is called in the start-up of the server, and it retrieves the current and the last valid time correlation coefficients along with their associated status, either from the cache / history files (setting the value of the corresponding MISCconfig variables) or from the MISCconfig variables, depending on the value of the *cacheInit* parameter. It should be noted that MISC variables are duplicated for each OBT defined on the system with the suffix "_n" being used to label the variable for the nth OBT.

bool **obtUttReference**(const char *obt, TDSsunTime &utt, const TDSsunTime &refUtt, long double &gradient, long double &offset, TDSsunTime &obtN, TDSsunTime &utcN, TDSsunTime &startTime, TDSsunTime &endTime, bool &accurate, bool &synchronised, int searchMode): This function uses 'TCOtmPktIF' to retrieve the time correlation coefficients that are valid for the period corresponding to the refUtt time, performing the search, either by filing or creation time, as indicated by searchMode. The search is done backwards in time only when earliest TCO coefficients known are found before current time, otherwise the retrieval is always done forwards. The function then correlates obt into UTC, and returns the converted value through utt. It also returns the retrieved time correlation coefficients as well as their associated validity interval, accuracy and synchronisation status through the corresponding output parameters. If the applicable coefficients are invalid the conversion is nonetheless done, although it returns false. When the applicable coefficients are not found, it sets utt to zero and returns false.

bool **obtUttValid**(const char* obt, TDSsunTime &utt, long double &gradient, long double &offset, TDSsunTime &obtN, TDSsunTime &utcN, TDSsunTime &startTime, TDSsunTime &endTime, bool &accurate, bool &synchronised, int searchMode, int directionMode): This function is similar to the former, but in case the applicable coefficients for the reference time are invalid, it looks forward in time to retrieve the next valid ones. In this case, it returns false, but the conversion is done. If no conversion is done at all, for no valid coefficients have been found, it returns utt = 0. Besides, the time correlation retrieval can be configured by means of the searchMode (filing or creation time) and directionMode (forwards or backwards in time) parameters.

bool **calculateTCOparameters**(): It calculates the new time correlation coefficients (setting their related validity and accuracy status parameters to true), updates the last valid ones to the new coefficients, and stores all of these values in their corresponding MISCconfig variables as well as into a local cache within the object, allowing for a fast retrieval when requested through function 'getTcoCoefficients'. The time correlation coefficients calculation is based on the time couples stored in a local buffer by the 'createTimeCouple' member function.

bool **confirmTCOparameters**(): This function confirms manually calculated TCO parameters. Only after this method has been called are the coefficients activated and the packet containing the coefficients archived.

void **resetTCOparameters**(): This operation has the following effects: It sets the time correlation status to invalid, clears the Time-Couples buffer, and suspends the validity check (through the 'TCO_SUSPEND_VALIDITY_CHECK' MISCdynamic variable). Eventually, it forces the automatic update of the time correlation coefficients (more precisely, as soon as there are two new Time-Couples inside the buffer).

bool **createTimeCouple**(const char *obt, const TDSsunTime &frameTransmissionTime, const TDSsunTime &filingTime, double deviation, bool inPast): This function creates a telemetry packet with the received parameters in the format applicable to the OBT. It can be called specifying a filing time in the past in which case the couple is created but is not added to the buffer used for current computations. This is for the purpose of reprocessing data to generate more accurate time couples.

bool **deleteTimeCouple**(const TDSsunTime from, const TDSsunTime to): this function makes a request to the PARC to delete time couple packets.

void **setTcoCoefficients**(long double gradient, long double offset, const TDSsunTime &obtN, const TDSsunTime &utcN, bool validity, bool accuracy, TcoCoeffRequest requestType): Sets either the current or the last valid TCO coefficients (as stated by the 'requestType' parameter), as well as their validity and accuracy status.

bool **getTcoCoefficients**(long double &gradient), long double &offset, TDSsunTime &obtN, TDSsunTime &utcN, bool &accuracy, int requestType): This function returns either the current or the last valid (according to the requestType parameter) time correlation coefficients along with their associated accuracy status, which are all obtained from the locally-cached values in order to avoid accessing MISCconfig variables. The function's return value is the validity status of the time correlation coefficients retrieved.

bool **checkSynchronised**(unsigned int &synchronisation, unsigned int accuracyInterval [optional]): It calculates the expected TAI / UTC difference ('TCO_TAI_UTC_DIFFERENCE') minus the current time correlation correction, and checks whether its absolute value is within the accuracy interval (accuracyInterval, or 'TCO_ACCURACY_INTERVAL' if the optional argument is not provided) in order to calculate the synchronisation status, which is returned.

STL::Optional<TDSsunTime **findEarliestTime**(TCO_IDL::SearchMode searchMode): This method returns earliest valid OBT and UTC time correlation coefficients which can be retrieved from PARC. The two coefficients may not necessarily be coupled, as one can inject valid time coefficients which "intersect" the earliest known couple. For example, one can inject valid time coefficients with an OBT later than the earliest and known couple and a UTC later than the earliest known couple.

6.5.2.3 TCOtmPktIF.

This class represents the interface to the CPD. It is implemented in a similar way to other CPD interfaces (for instance the 'CMDHtcPktIf" class in the command handler). It also provides the functions to retrieve the requested time correlation coefficients and their validity time (interval until the time correlation is calculated again). It constructs filters in order to retrieve the required data from the CPD. The SPIDs of the time correlation data packets are obtained from MISCconfig variables.

6.5.2.4 TCOcalculation

This is an abstract base class defining the functionality of classes providing the calculation of time correlation coefficients. It defines as well as the calculation methods for storing the calculated coefficients to a packet and reading the values from a packet.

6.5.2.5 TCODifferenceCalculation & TCOleastSquaresCalculation

These are specialisations of the TCOcalculation base class providing functionality for the correction/difference and least squares method of time correlation.