TOWARDS A CLASS LIBRARY FOR MISSION PLANNING

Olivier Pujo[1]
Simon T. Smith[2], Paul Starkey[2], Thilo Wolff[1]

[1]European Space Operations Centre (ESOC)
Robert Bosch Str. 5
D-64293 Darmstadt

[2]Cray Systems Ltd.
DAS House
Quayside
Temple Back
UK-Bristol BS1 6NH

Abstract

The PASTEL Mission Planning System (MPS) has been developed in C++ using an Object-Oriented (OO) methodology. Whilst the scope and complexity of this system cannot compare to that of an MPS for a complex mission one of the main considerations of the development was to ensure that we could re-use some of the classes in future MPS. We present here PASTEL MPS classes which could be used in the foundations of a class library for MPS.

Key words: Mission Planning, Object-Oriented, Class Library

Introduction

PASTEL is an experimental optical terminal to be flown on as a passenger on SPOT-4, the earth observation spacecraft developed and operated by the Centre National d'Etudes Spatiales (CNES). A corresponding optical terminal will be flown on the ARTEMIS spacecraft, which is part of the Data Relay and Technology Mission programme (DRTM) of the European Space Agency (ESA). PASTEL will have a separate Mission Control System (MCS), which will be operated by ESA. The PASTEL Mission Planning System (MPS) is part of the MCS and has been developed in C++ using an Object Oriented (OO) methodology.

In ESA, the area of Mission Planning is one in which the use of generic systems has been considered only recently, in sharp contrast to spacecraft control systems for which ESA has been using configurable multi-mission systems for nearly twenty years. ESA's mission planning systems have been project specific development and there has been very little carry over of the expertise, tools or software from one project to another. A recent ESA study showed that there are areas of commonality between different mission planning systems (ESA, 1992). The use of traditional software technologies (FORTRAN, PASCAL, functional decomposition...) for MPS development has certainly been one of the hindrances to provision of re-usable components.

One of the strong claims of OO approach is the possibility of re-use. Re-use in an OO system is usually implemented through the Class Library concept, where a Class Library is a collection of general purpose components that can be used as a basis for further refinement on specific projects. The concept is similar to that of standard graphics or numerical libraries, with the important difference that by using the OO principle of inheritance the behaviour of the components can be modified (to add, remove or alter features).

Whilst the scope and complexity of the PASTEL MPS cannot compare to that of a mission planning system for a complex science or earth observation mission, it is used in this paper to highlight certain possibilities for re-use. One of the steering factors in the development of the PASTEL MPS was to try to ensure that some of the classes would be re-usable in future mission planning systems. An immediate motivation being potential re-use in other areas of the DRTM programme.

We start with an introduction to the PASTEL mission. Then we look at various aspects of the PASTEL MPS: the development process and object class hierarchy. A discussion on the re-use potential of the PASTEL MPS Timeline and Reservation Plan area follows.

PASTEL and the SILEX experiment

PASTEL and its counterpart terminal, OPALE, mounted on the ARTEMIS satellite form the SILEX (Semiconductor Inter-Satellite Link Experiment) mission which will be used to downlink high rate data generated by SPOT's optical camera, using ARTEMIS as a data relay. For technological purposes PASTEL will also be able to point to stars.

The PASTEL terminal will be operated by ESA from the PASTEL Mission Control System (MCS) located in the ESA Redu station. Control and monitoring information will transit through the SPOT-4 Control Centre, located at Toulouse, in a cross-support scenario. The planning of the SILEX experiment is under the responsibility of the PASTEL MCS, which will coordinate with the SPOT-4 Control Centre and the ARTEMIS Control Centre.

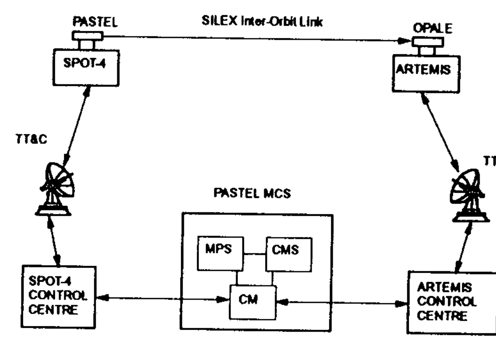

Figure 1: SILEX experiment and PASTEL MCS

As shown in Figure 1, the PASTEL MCS comprises three principal subsystems, (1) a Control and Monitoring System , (2) a Mission Planning System, and (3) a Communications Monitor. Within the PASTEL MCS, the MPS is in charge of all planning activities.

PASTEL MPS

The PASTEL MPS main functions (ESA, January 1993) are:

(A) to allow the MPS operator to coordinate the production of the Reservation Plan, (which defines the periods in which PASTEL can communicate with OPALE, and the periods where star tracking can be performed by PASTEL),

(B) to produce an Operations Timeline, containing all the details including telecommands of the operations to be scheduled from the PASTEL MCS, under MPS operator control.

The Reservation Plan holds SILEX communications sessions, which are also called "windows". At the first stage of the planning process, the communications sessions are called visibility windows and are derived from flight dynamic information provided by SPOT-4 Control Centre. The following steps involve a number of iterations between PASTEL MPS, SPOT-4 CMP and ARTEMIS MCS to allow each centre to reserve or cancel the windows according to their operational constraints.

PASTEL MPS development

PASTEL MPS has been developed using C++ and OO methodology following the Object Modelling Technique (Rumbaugh *et al*, 1991). A traditional waterfall life cycle process model (ESA, 1991)was adopted with the following adaptations. The user interface was prototyped at an early stage. The design documentation was simplified: a unique design document replaced the traditional Architectural Design and Detailed Design Documents. And finally integration of components was performed from very early design stages. The overall effort for the development was in the area of 30 man-months, and 24000 lines of codes were produced.

The object-oriented approach was primarily adopted for this development in view of the potential re-use of it in the frame of the ARTEMIS MCS development to support the scheduling of OPALE. PASTEL MPS will be the first OO system delivered in ESOC for operational usage.

PASTEL MPS Object Classes

The overall object classes hierarchy for the core of PASTEL MPS is provided in figure 2 (ESA, June 1993). Two parallel structures appear in this hierarchy: the Reservation Plan and the TimeLine. For simplification purposes, this figure does not cover the class hierarchy for the user interface objects, which were introduce to dissociate application objects from the user interface. We will first provide a short outline of the Reservation Plan and the Timeline before discussing their potential re-use.

The Reservation Plan (and its associated user interface objects) contains, from a user perspective, the list of all windows for a planning period (typically of five weeks). Each window has a status which determines whether the corresponding communication session is reserved or cancelled.

The operator interacts with the Reservation Plan through a specific display, called the Reservation Plan Mode Display, which consists of two areas:

- The Reservation Plan Index, which allows the operator to navigate through the weeks and days of the

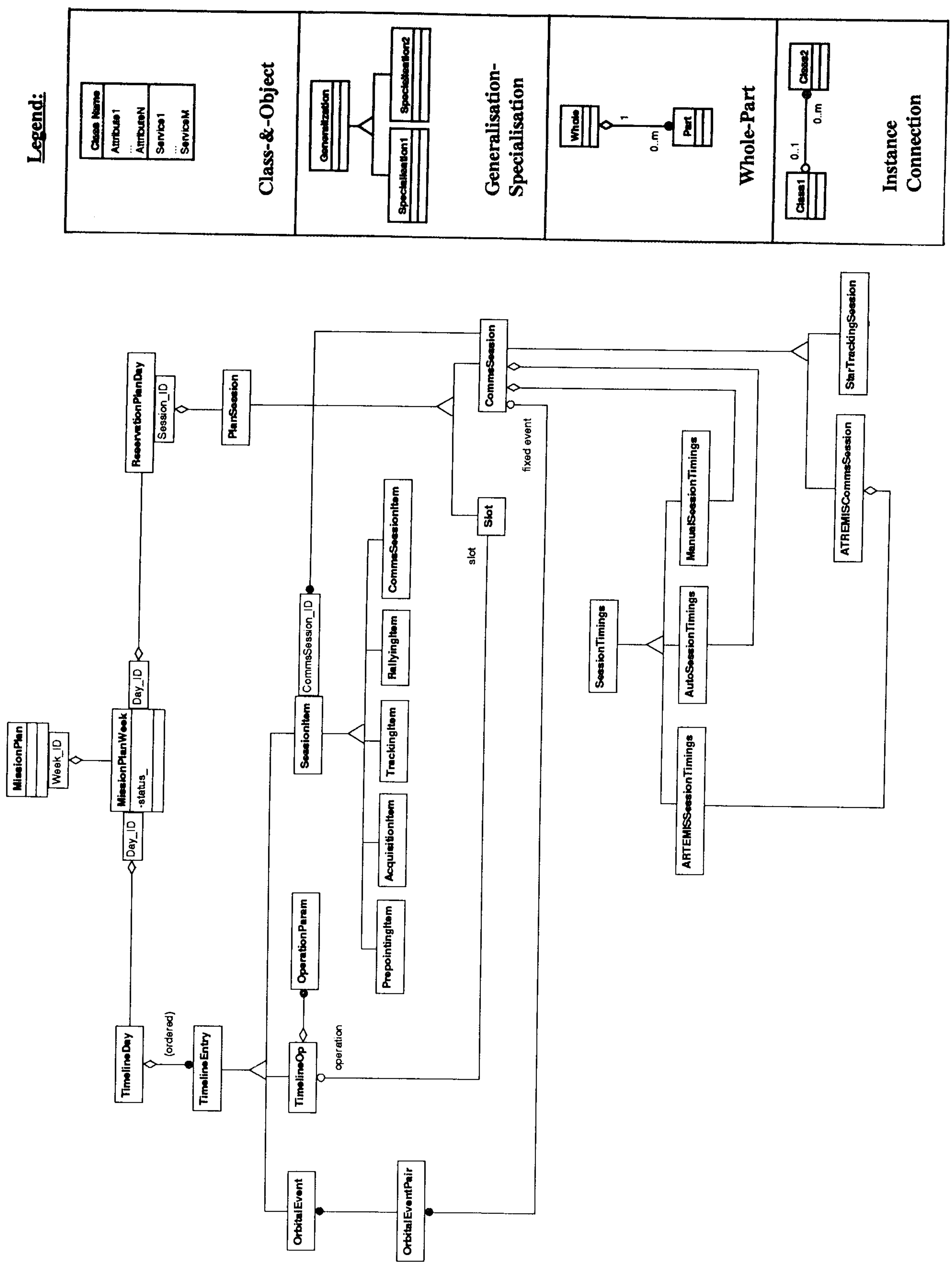

Figure 2: Overall Object Hierarchy

Plan and to select the window to work with.

- The Window Display, which displays all information relevant to a single window and provides the operator with a set of options for controlling the planning of the window.

A parallel data structure to the Reservation Plan, the Timeline, is available to the operator from the start of the planning cycle. The Reservation Plan and the Timeline provide two different views of approximately the same information describing the operations to be scheduled on-board. Where, in the Timeline, the information is formatted in templates close to command sequences, in the Reservation Plan, the information is formatted in templates which are closer to the intuitive user perception of the planning. In other words, the Reservation Plan provides a macroscopic view of the planning, whilst the Timeline provides a microscopic view.

The Timeline contains mainly sessions items,which describe the operations to be performed to establish a communications session. The Timeline may also contain other operations, such as Laser diodes calibrations. In principle the operator is free to enter operations into the Timeline at any stage, although in practice most of the operations will probably be entered at a late planning stage once the Reservation Plan has been more or less finalized.

Re-use of PASTEL MPS classes

The PASTEL MPS is simple compared to that of other ESA mission planning systems for the following reasons:

- it operates within a fixed set of resources and constraints,
- scheduling tasks are handled manually,
- the communications sessions scheduled result in a fairly fixed pattern of operations in the Timeline,
- it is an off-line system, i.e. there is no requirement to perform real-time re-scheduling of the mission.

Despite these restrictions, the areas covered by the PASTEL MPS are equivalent to parts of more complex planning systems. Thus some of the classes identified in the PASTEL MPS Object hierarchy can be considered for re-use.

Timeline

The most promising area for re-use in the PASTEL MPS is certainly the Timeline area. It includes several classes: TimelineDay, TimeLineEntry, SessionItem, OrbitalEvent, TimelineOp, etc., and for each of these classes we foresee potential for genericity. However, we will focus in our discussion on the timeline itself, which is implemented in PASTEL MPS in the TimelineDay.

One fundamental component of any spacecraft mission planning system is the timeline. The timeline is the basic structure to store information required to plan a mission:

- scheduled operations such as time tagged operations to be executed on-board the spacecraft or at the ground station,

- events pertinent for spacecraft operation such as eclipse entry and exit, ground station visibility period.

- spacecraft on-board status changes such as instruments mode switching or on-board tape recorder activity.

PASTEL MPS *TimelineDay* class is an interesting starting point because it highlights very generic features, namely:

- time-ordered list with protected insert mechanism (in order to force entries to be inserted in chronological order),

- support of heterogeneous list items, i.e. all elements forming it do not necessarily belong to the same class,

- support of active display filtering, i.e. it is possible to select list items of the selected types.

The Timeline is in essence a chronological list of items, which correspond either to operations to be executed on-board or to events such as eclipse times, etc. Whenever an item is added to the list, it is essential to check that this is done according to a correct time order.

To meet the genericity objective a timeline clearly needs to support a mix of objects. There are some good reasons to distinguish between timeline inputs such as operations or orbital events. We use the OO inheritance mechanism to solve this problem as shown below in figure 3. By defining *operation* and *eclipse* as subclasses of a more general class, *timeline_entry*, we can construct a timeline with entries of type *timeline_entry* provided that the programming language supports the use of the subclasses *operation* and *eclipse* in place of the general *timeline_entry* contained in the class definition for timeline.

Editing the timeline is, by nature, a highly interactive task and the support of active display filtering is a common requirement. In fact any combination of elements types should be displayable. This has been implemented simply by adding a dedicated attribute, *type*, to the *timeline_entry* class, which is then used by the mission timeline user interface object to implement the active filtering.

Reservation Plan

Another potential area of re-use in the PASTEL MPS is the Reservation Plan area. On the overall object hierarchy (figure 2), the parallel between the classes *TimelineDay* and ReservationPlanDay is quite striking. They are formed respectively of TimelineEntry and PlanSession, which are generic classes for parallel structures such as *SessionItem* and *CommsSession* or *TimelineOp* and *Slot.*

Schematically, one could say that the

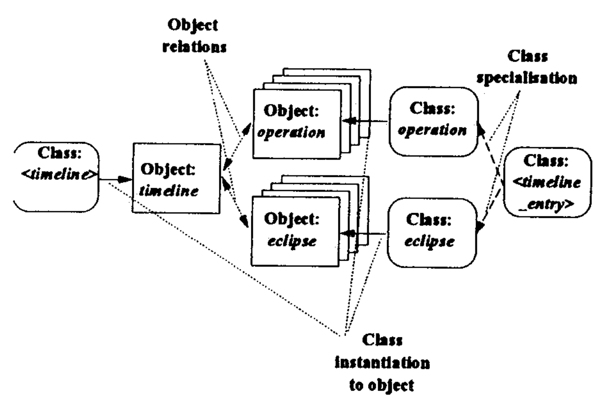

Figure 3: Mixing objects within the timeline

Reservation Plan is used in early planning phases, while the Timeline is used in the last planning phase. However, there is no general rule forbidding the operator to edit the Timeline at an early stage or the ReservationPlan at a late one. In fact, they both contain more or less the same information and what distinguishes them is more the way in which an operator wants to interact with them.

The Reservation Plan is geared more to specific planning aspects of PASTEL; it holds e.g. information such as terminal constraints, which are useful only to compute communication sessions timings, or window status attributes (reserved|cancelled|...), which are not held in the PASTEL timeline parallel structure.

In order to keep the Reservation Plan manageable, three levels of hierarchy are defined: week, day and window. This breakdown maps the events managed in the Reservation Plan and the planning cycle. Although it is clearly specific to PASTEL, it could be very simply generalized through the OO inheritance mechanism and/or by renaming some classes.

This breakdown is also useful to provide external users with some "snapshots" of the Plan or to update the Plan according to new data from external users. This is performed in PASTEL MPS by specific callbacks and methods, which could be re-written for any application. It is anticipated that the definition of external user interface is in any case specific to each mission. All that a generic mission planning class library needs to provide is the anchor points to these external interfaces, which are provided, in PASTEL MPS, in the methods of the Reservation Plan items (ReservationPlanDay, CommsSession...).

Finally, the possibility to manage various versions of the Reservation Plan is believed to be of interest to a number of missions. The PASTEL MPS allows two versions of the Reservation Plan to co-exist, one being the reference and the second corresponding to an update generated when receiving inputs from external users. The two versions can be compared and the operator can select the update to apply to the reference Reservation Plan. This feature is used only for temporary purposes but could be used in a broader way to allow multiple operators working concurrently.

To summarize, the following features are, we believe, quite generic in the PASTEL MPS Reservation Plan area:

- the breakdown of a reservation plan into smaller units, which is a mandatory requirement to keep the plan manageable;
- the requirement to provide to external users "snapshots" of the plan to synchronize planning activities.
- the configuration management of several plan versions.

Conclusions

At ESOC a number of object-oriented developments have recently taken place or are in progress, PASTEL MPS being one of the very first. Whilst the prime objective of the PASTEL MPS development was not to provide a generic class library for mission planning, there was a strong motivation to achieve some genericity in view of the potential for re-use on ARTEMIS and DRS satellites.

It has been shown that there is a good expectations that certain PASTEL MPS classes can be re-used. The Timeline and Reservation plan areas seem very promising starting points. The on-going Generic Mission Planning Facilities for Operations Study should confirm these expectations by consolidating some aspects of these PASTEL MPS classes to make them more generic and by using them to model other ESA mission planning systems.

It is anticipated that the result of this work is fed into SCOS II, ESA's new infrastructure for spacecraft control, which is currently under development and which is being built as a C++ class library for spacecraft control.

References

ESA (1991) ESA Software Engineering Standards, ESA PSS-05-0 Issue 2.

ESA (1992) Generic Mission Planning Toolset, *ESA Report CR(P) 3600*.

ESA (1993, January) PASTEL Mission Control System Software Requirements Document.

ESA (1993, June) PASTEL Mission Control System Architectural Design Document.

J. Rumbaugh *et al.* (1991) *Object-Oriented Modelling and Design*. Prentice-Hall International .

Acknowledgements

The PASTEL MPS was developed by Cray Systems Ltd. under an ESOC contract 8793/90/D/IM. The re-use of the PASTEL MPS classes is currently under study in the frame of ESOC Contract 10694/93/D/IM(SC) - Study on Generic Mission Planning Facilities for Operations - performed by Cray Systems Ltd. and Matra MarconiSpace.