

ADVANCED CREW TERMINAL - "A GENERIC FRAMEWORK FOR THE REALISATION OF SYSTEM AND PAYLOAD CREW TERMINALS"

Leslie P. Gale * & Mikael Wolff **

* *ORIGIN/Nieuwegein B.V, P.O. Box 1444, NL 3430 BK Nieuwegein, The Netherlands*
fax: +31 30 6060577, email: Leslie.Gale@nlwgfsc.origin.nl

** *ESA/ESTEC, P.O. Box 299, NL 2200 AG Noordwijk,*
fax: +31 71 5655419, email: wolff@wg.estec.esa.nl

ABSTRACT. Building upon experience gained in a number of ESA studies, ORIGIN as prime with partners CRI (DK), SAS(B) and NLR (NL) have developed a concept and a product called the Advanced Crew Terminal (ACT). The work was funded by ESA and the Netherlands Agency for Aerospace Programmes (NIVR). ACT provides a generic solution for the development and utilisation of crew and ground terminals for system and payload operations. ACT will reduce the risk and development time for developers by removing the need for each developer to develop the space and ground segment interfaces. It will provide for the users of the terminals a cognitive approach to terminal operation, where, all but the dedicated payload or system applications are common to all the terminals. ACT is part of a generic framework for crew operations which is promoted and developed by ESA/ESTEC. This paper presents the background to establishing ACT and the ACT project which has led to the provision of generic services and demonstrated its use. Also the planned extensions which include using the DMS-R/COF crew procedure language (CPL) in ACT (completed in parallel to ACT) and Speech Recognition will be discussed, together with a look at future trends.

1. BACKGROUND

The need for ACT arose as an output of ESTEC Crew Workstation Test-Bed (CWS TB) activities for Columbus (see [RD1]) which identified the benefits of standardising crew terminals.

Two types of crew terminals were studied for Columbus:

- The 'Portable Terminal' (PT). This is a system item belonging to the data management subsystem (DMS), to support the crewman for system operations and for high level payload operations. The PT is connected to the onboard network, and provides control/monitoring access to subsystems and payloads at a level similar to that from mission control from the ground (see: [RD2]).
- The 'Crew Portable Computer' (CrewPC). This is a general purpose crew terminal belonging to the payload complement, to support the crewman in his role of on-site scientist as assistant / representative of the principal investigator (PI) and his team on the ground (see [RD3]).

However, the CrewPC and the PT were developed for specific pre-defined missions demonstrations and simulations in order to demonstrate and study crew and user support concepts. Modified versions (as shown in figure 1) have been tested during actual flight opportunities (D2-CTE and EUROMIR-CPC).

As the successor of the CrewPC Mark II and the Portable Terminal ACT is designed to cater for systems, subsystem, and payload operation. ACT, and unlike, CrewPC and PT, ACT is a generic system that can be adapted and configured to fulfil mission-specific needs.

To standardise and meet the requirements of customers for the ACT, three classes of ACT derivatives have been identified: onboard payload terminal (OPT); onboard system terminal (OST); and a ground payload terminal (GPT). This does not however restrict the potential of ACT. Its generic approach will allow its use for ground activities (e.g. pre-flight testing, checkout, crew training, and on-line mission operation) and even unmanned flights where ACT derivatives can be used for ground-based telescience activities (remote monitoring and control).

Disclaimer: Opinions expressed are those of the authors and need not necessarily represent those of ORIGIN, ESA/ESTEC or the partners CRI, NLR and SAS.

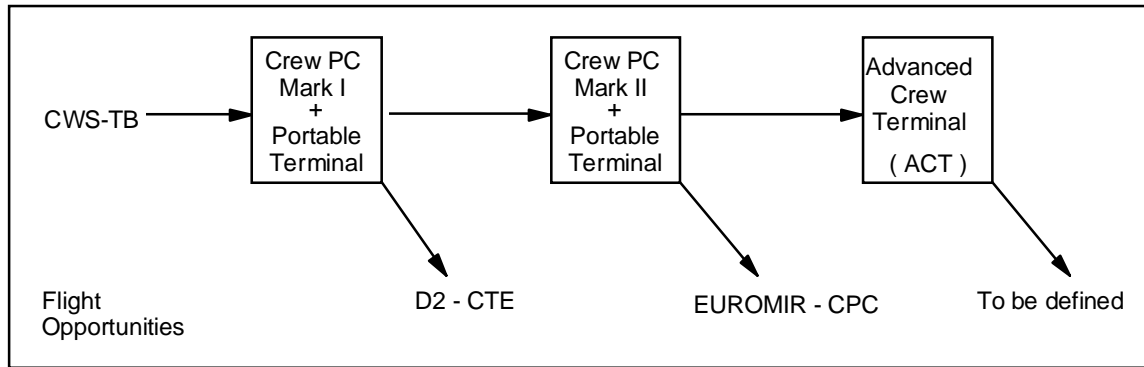


Figure 1: CWS-TB Developments and Flight Opportunities

Note: NASA inspiration sources to the ACT project are the Crew Personal Support Computer (CPSC) project and the Payload General Support Computer (PGSC) project.

ACT is also part of a larger framework which ESA/ESTEC wishes to establish in order to support and demonstrate the development and use of laptops as terminals for use by astronauts, principal investigators and development teams. ACT is therefore a concept and a product. A major goal of ACT is to become an integral part of a development kit which can be distributed to developers and users of system and payload terminals.

2. THE ACT PROJECT

The ACT project itself has been carried out in two phases. Phase 1 (completed January 1995) was the specification phase in which the user requirements, architectural concepts and COTS selection took place. Phase 2 (completed September 1996) was the implementation phase of the ACT project and contained four main processes;

1. Development of the generic ACT core;
2. Development of the ACT laboratory demonstration configuration (applications and databases) using the generic ACT framework;
3. Demonstration (exercising) of the ACT framework, and the ACT laboratory demonstration configuration.
4. Evaluation of the ACT framework, and the ACT laboratory demonstration configuration.

The ACT project delivers more than a software product to the ESA/ESTEC Framework. It provides style guidelines, authoring and configuration tools to assist developers.

In order to demonstrate the ACT concept and product a demonstration mission has been adopted. The mission calls for an instance of each of the three ACT derivative classes: OST, OPT and GPT. A laboratory environment has been established using as near-to-real as possible equipment. Scenarios have been developed which replicate as close as possible in-orbit and ground operations during a manned space mission whilst illustrating clearly the features of ACT. The environment and scenario form the ACT laboratory demonstration configuration.

3. THE OPERATIONAL CONCEPT

To cater for the needs of the ACT users an operational concept has been defined which brings into context the operational environment and the facilities required to allow the users of ACT to complete their tasks. The concept takes into account the mission lifecycle, which can be a short duration or increment of the space station utilisation.

Two distinct operational environments can be identified for which the ACT provides facilities:

- 1 Mission Preparation Environment
- 2 Mission Operation Environment

The operational environment is defined by the various facilities, and by the communication network between those facilities. The low bandwidth communication between ground and space imposes severe constraints on the operational capabilities during mission operation. Analysis of these facilities and network has been used to establish the design requirements and constraints on the development of the ACT.

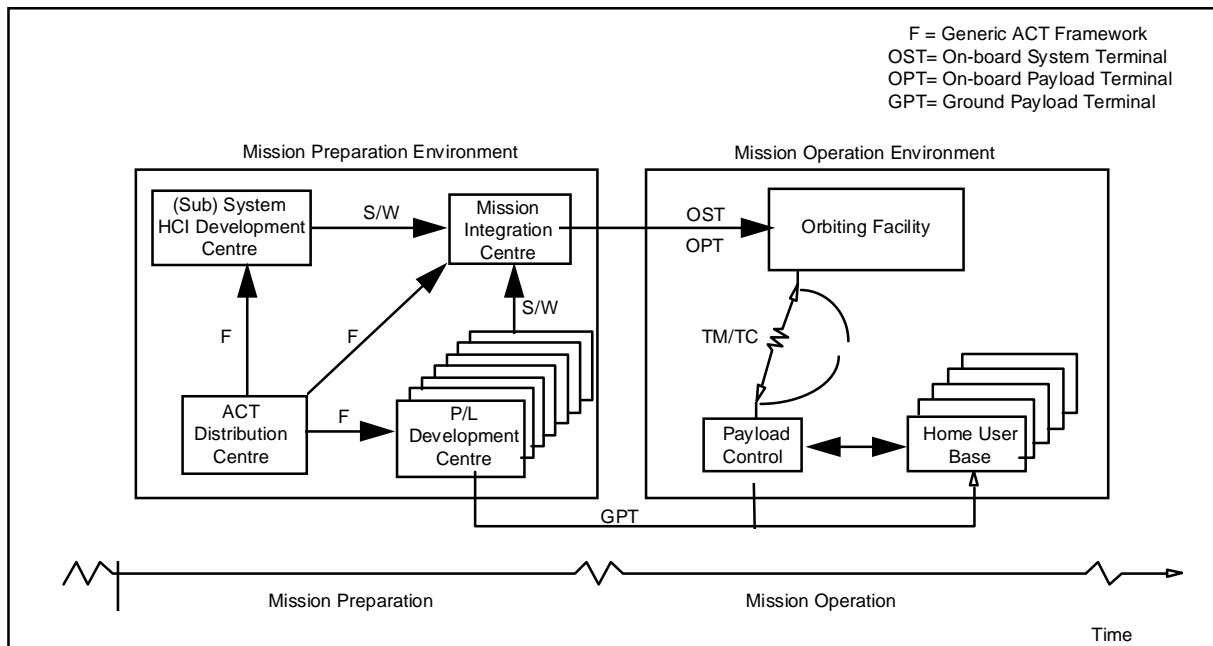


Figure 2: Mission Preparation and Mission Operation

3.1. ACT USERS

The ACT derivatives will have different users which can be subdivided into 'ACT mission preparation users' and 'ACT mission operation users'.

The mission preparation users will develop mission-specific dedicated system or payload applications using the generic core services and other products such as Visual Basic, Labview or even coding in the C programming language. The mission operation users, on the other hand, will use the developed mission-specific dedicated applications and the core services. All core services are available for users during both mission preparation and mission operation phases.

3.1.1. ACT MISSION PREPARATION USERS

ACT mission preparation users can be divided into developers and crew undergoing training. The developers that create dedicated services using a subset of generic core services, can be subdivided into (sub)system HCI developers and payload software developers.

The (sub)system HCI developers are responsible for developing the HCI for the ACT OST-derivative. The OST is the derivative in which the ACT plays the role of the system terminal. Core services such as authoring tools for multimedia documentation and procedure execution can be used by these users.

The payload software developers are responsible for preparing all the software parts which are used during mission operation for a particular payload running on OPT and GPT derivatives. The OPT is the derivative in which the ACT plays the role of the Crew PC.

Payload software developers can be subdivided into:

- Operations support authors use the authoring tools provided by ACT. They are responsible for configuration and editing/filling of the operational database, which includes procedures, reference documents and help information.
- Software application developers who are responsible for dedicated payload applications (such as virtual control panels (VCPs) and data processing applications). Core services that are used are configuration tools to integrate the developed mission-specific applications with the ACT core services. The interface between core services and dedicated services is handled by the ACT communication services.

3.1.2. ACT MISSION OPERATION USERS

ACT mission operation users, who are users of the mission-specific dedicated services and a subset of generic core services, can be subdivided into the *crew* and *ground experts*:

- Crew are the ‘in-orbit’ users of ACT derivatives (OPT and OST).
- Ground experts are all the people involved with supporting tasks executed during the mission with respect to system and payload operations using GPTs. Ground experts include:
 - System Engineer
 - Principal Investigators
 - Payload Facility Experts
 - Support Engineers

Both crew and ground experts make use of the dedicated services (e.g. VCPs) and the core services such as the multimedia documentation system (a browser) and the procedure execution system. Due to the fact that OPT and GPT can be synchronised, the quality of space-ground communication and team collaboration is much better when compared with communication by voice link only.

Further, a common “look and feel” for crew can be achieved by adopting the ACT concept across the various subsystem and payloads in major programmes (e.g. COF). This is an important mission benefit, increasing flight operations effectivity and efficiency while reducing training effort.

3.1.3. PROJECT PROOF OF CONCEPT APPROACH

Utilising the operational concept the ACT project adopted a proof-of-concept approach. Two teams were established with the following role distribution:

Team	Role
ACT providers	Core services developers ACT distributors Help desk Mission (Laboratory) Integrators
ACT derivative developers	(Sub)System HCI developers P/L Developers Mission (Laboratory) Integrators

3.2. FACILITIES AND SERVICES PROVIDED

For mission ACT is directed towards supporting crew in the performance of their tasks and provides a number of features which include:

- multi-media documentation and help facilities;
- shared operation between onboard and ground terminals;

- multi-media interaction between crew onboard and the ground based experts for telesupport;
- remote payload operation;
- role, task and procedure management (Timeline schedule);
- multi-media supported procedure ;
- updating of ground prepared procedures and documentation during the mission;
- (image) capture support, between ground and flight environment, and vice-versa;
- support services, such as, status information, logging, and navigation.

For mission preparation, the flexibility and capabilities offered by ACT provide a cost-effective solution to terminal development. Payload and system terminal developers can concentrate on the mission-specific applications. ACT provides tested and reliable services to simplify development activities, reducing development time, risks and costs. Services include:

- communication protocols and management;
- payload interfacing;
- logging;
- housekeeping;
- graphical annotation;
- procedure authoring; and
- multi-media support.

Further, ACT is based upon low cost technology that emphasises the use of commercial laptops for flight computers, commercial desktop computers for ground-based (mission preparation) and COTS (commercial off the shelf) products.

However, it should be noted that the low bandwidth communication capabilities between the space and ground environments has had an impact on the selection and use of COTS products in ACT. An example of this is the shared whiteboard facility allowing astronauts and payload specialists to communicate (telesupport, teletraining). This can be supported by a COTS product providing the communication bandwidth is high and reliable. The space network is low bandwidth and subject to loss-of-signal (LOS) making the use of a current COTS product impossible.

4. ACT ARCHITECTURAL CONCEPT

Technology improves in a development cycle shorter than most space missions. To take advantage of technology improvements, without incurring cost penalties for redesign, the ACT architecture should ideally be platform-independent (i.e. hardware, operating system, database system, communication).

ACT utilises the experience and lessons learnt by ORIGIN in the CrewPC and PT developments. In order to develop a generic system, services have been classified into core services and dedicated services. The core services are generic services which can or will be part of all ACT derivatives. The dedicated services are those services that are specific to a certain mission and a certain derivative.

The core services contain:

- core communication services;
- core applications;
- core database services.

The dedicated services which are under responsibility of ACT users, will contain:

- dedicated applications
- dedicated databases

Two platform dependent layers can be identified:

- database platform dependencies layer
- HW & OS dependencies layer

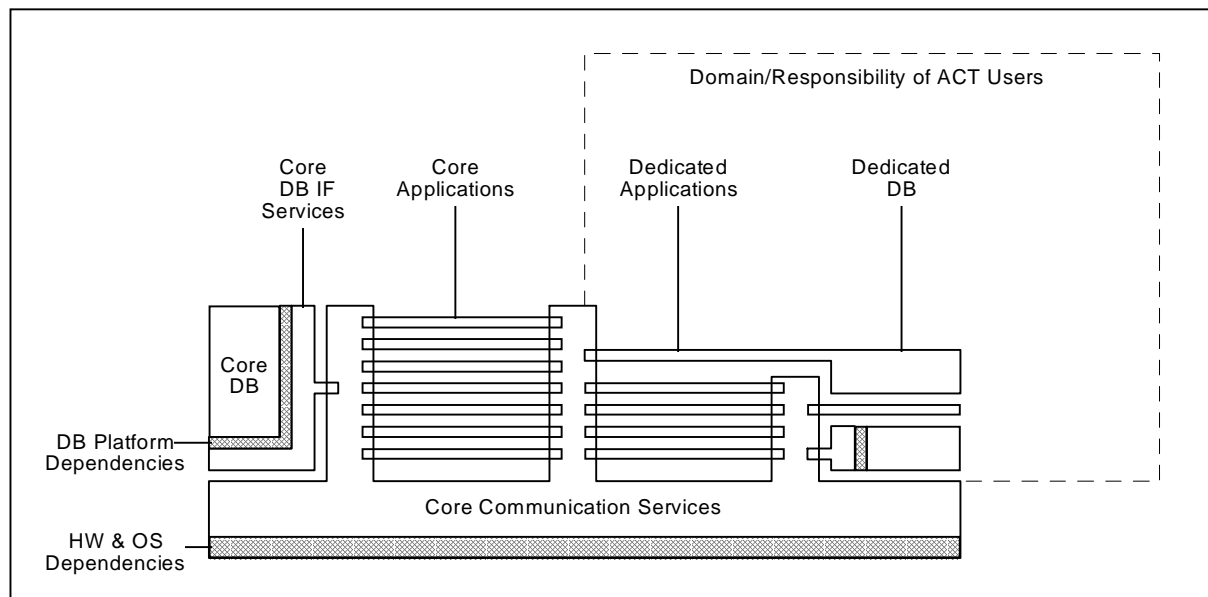


Figure 3: ACT Architecture Layers and Platform Independence

As the architecture shows, applications (core and dedicated) can easily be added or removed. All applications make use of the same communication service.

All core services and dedicated services should be structured such that the function (applications) is clearly separated from data (stored in datafile(s)). This will allow for separate applications for mission preparation and mission operation related to a certain service. However, other structures are possible, as depicted in “Figure 3: ACT Architecture Layers and Platform Independence”.

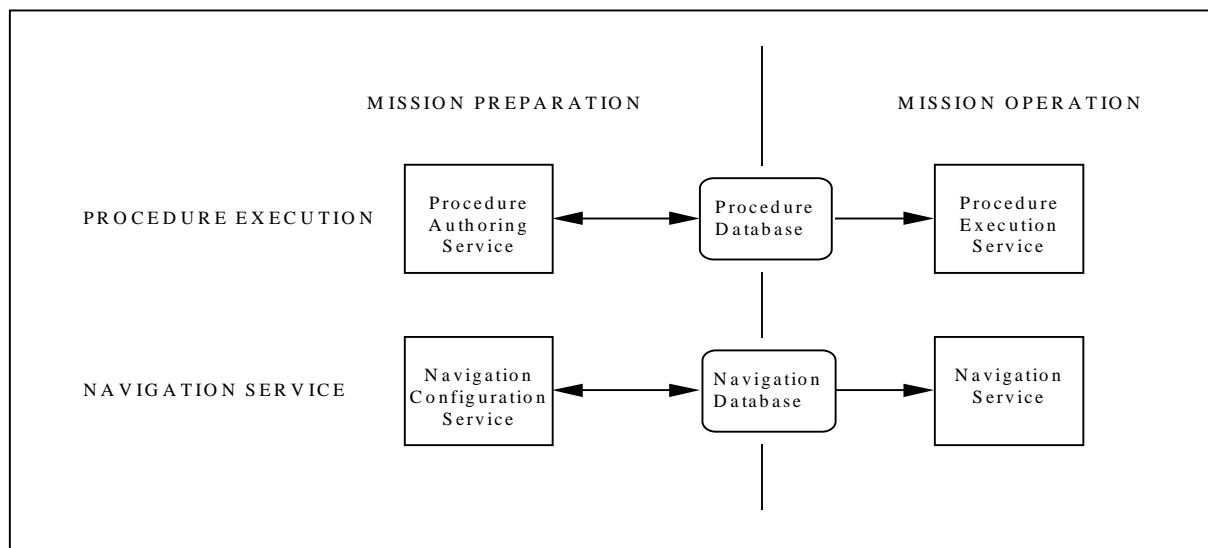


Figure 4: Example of the service structure

Note: The ACT architecture does not preclude that some databases can be updated during the mission operations phase.

Support to the user is an important feature for successful introduction and acceptance. ACT provides an integrated solution providing support facilities for development. So, for instance, the procedure service (PS), provides a procedure authoring service (PAS) for use during mission preparation to create the

mission-specific procedures. This authoring tool will allow users to edit and modify procedures that will be stored by the authoring tool in one or more datafiles. During mission operation these datafiles will be used by the procedure execution service (PES).

5. PROJECT EXPERIENCE

At the start of the project a choice was made to include into the ACT as many COTS products as possible. A number of decisions, and with hindsight crucial decisions, were made. Firstly, the chosen platform was a Thinkpad 750 CD running Windows for Workgroups (3.11). Secondly, documentation and the procedure language were to be based on HTML. Netscape was chosen as the documentation browser.

Also, there is a strong requirement to have shared applications, that is, applications which run synchronised between the OPT and one or more GPTs. This facility is essential for telesupport and teletraining. To facilitate shared applications, a synchronisation interface has been developed and provided as part of the ACT core services, allowing, in principle, any component to become a shared application.

Development of the interface was not problem free. A shared application works by capturing (menu) actions and events and transmitting them to its partner(s). For instance, a crew member may start-up PES with a given procedure on the OPT. Providing the GPT and OPT are sharing the application, the GPT will follow by invoking PES, starting the procedure, and going to the step selected by the crew member who is using the OPT. ACT, however, uses multi-media procedures and documentation. Not all COTS products provide an interface to capture the actions and events needed by ACT and the interface can be complex, e.g. a video sequence requiring a frame-by-frame synchronisation. Also, experience has shown that the COTS products interfaces are not completely reliable. This resulted in extra development in the core services to reach the desired result.

Another drawback was the inability of some COTS products to communicate using an OLE2 interface. Since a goal was to get the dedicated applications as close to being plug-and-play as possible a number of extra interfaces, e.g. DDE-OLE, have been built to simplify the dedicated application interfacing.

Further, not all the COTS products use memory in a satisfactory manner. ACT has a relatively large number of active components causing memory problems. This have been solved by paying careful attention to configuration and the use of a memory management COTS tool.

The ACT proof-of-concept approach provided the ACT providers (the core development team) with useful feedback on the needed documentation content. By allowing the derivative developers the freedom to choose the development tool for application development the interfaces required of the core components were tested, problems, such as the OLE2 problem described above were quickly highlighted and resolved.

ACT has achieved a positive result. It has been realised using a low-cost approach, the unit price of the ACT is low. All space segment issues such as communication using CCSDS [RD4], payload interfacing, use of multi-media material (HTML, GIF, Video) have been resolved. The procedure service, comprising a procedure authoring service (PAS) and procedure execution service (PES), provide the terminal developer and user with a comprehensive but easy-to-use interface. Also PES is capable of managing both the ACT procedure protocol and the protocol from the crew procedure language (CPL) of DMS-R and COF. Payload and system developers can concentrate on their problem: building experiment databases and virtual control panels.

6. FUTURE TRENDS

Already strong signals are coming from astronauts to increase the intelligence in the onboard terminals. Knowledge based systems, AI oriented approaches and interfacing devices such as speech are being requested. To investigate, and extend the capabilities, an Applied Speech Recognition/Synthesis (ASR/S) project has been started to allow terminal operation whilst both hands are involved with the execution of an experiment, e.g. when using the glovebox facility. Initial study has shown that suitable products are only available under Windows 95 or Windows NT. Therefore, there are now candidates for the ACT operating system. Also, the hardware platform development has moved on providing new opportunities in terms of screen size and resolution, memory and speed. A move to Windows 95 and more probably Windows NT is expected, which will retain the current approach to HCI which takes advantage of the astronaut's familiarity with Microsoft Windows products.

7. CONCLUSION

ACT is available now, offering to system and payload developers and to the crew functional capabilities which are compatible with or in excess of those offered by either the DMS-R or COF laptops. It is recognised that the ACT development has not been subject to the rigours of the software and design requirements imposed upon DMS-R and COF. The use of the mission database (MDB) facilities is not included in ACT, but the ACT consortium would certainly welcome the opportunity to demonstrate ACT in the development of a payload terminal in the context of the International Space Station. ORIGIN and its partners CRI, NLR and SAS thank ESA/ESTEC and NIVR for the opportunity to develop ACT. The assistance, advice and criticism from ESA project teams was welcomed and constructive.

References:

- [RD1] ESA/ESTEC. Columbus Utilisation Simulations In the ESTEC Crew Workstation Test-Bed. ESA/ESTEC/WGO/93.350.
- [RD2] ERNO Raumfahrttechnik, ROT134. Crew Work Station Reference System; Software User Manual for the Portable Terminal, ROT134-AMMI&CSII-93/01-Draft
- [RD3] BSO. Man Machine Interface Elements for Crew Services Final Report. BSO/1703294, April 1994.
- [RD4] CCSDS. Consultative Committee for Space Data Systems, Advanced Orbiting Systems, Networks and Datalinks: Architectural Specification, CCSDS 701.0-B-2, Blue Book, Chapters 2.5 and 3, October 1992.