How to build a Space Robot; ERA Lessons Learned

Marc Oort, Frank Meiboom, Cock Heemskerk
Fokker Space
Newtonweg 1, 2303 DB Leiden, NL. m.oort@fokkerspace.nl, phone +31715245448, fax +31715245499

Abstract
The **European Robotic Arm** [ERA] is being built for use on the Russian Segment of the International Space Station. The project is commissioned by ESA as part of their manned-space program, with Fokker Space as Prime Contractor, and 23 companies from 7 European countries participating in the development of the arm. The ERA is scheduled to be launched as part of the SPP by Space Shuttle to the ISS, and is planned to operate on the ISS for ten years. Testing of the Flight Model is currently underway. This paper focuses on some of the lessons learned from this project.

1. Introduction to ERA
The ERA system (Figure 1) consists of an arm, an EVA Man Machine interface , an IVA Man Machine Interface, a Refresher Trainer [RTR] and a Mission Preparation and Training Equipment [MPTE]

The ERA arm is a 11 meter, 6 Degree-of-freedom arm , whose most striking feature is the ability to cover large distances on the ISS by “hopping” from one basepoint (which supplies the power and communication interface) to another. For an overview of the operational aspects of the arm, see [SO].

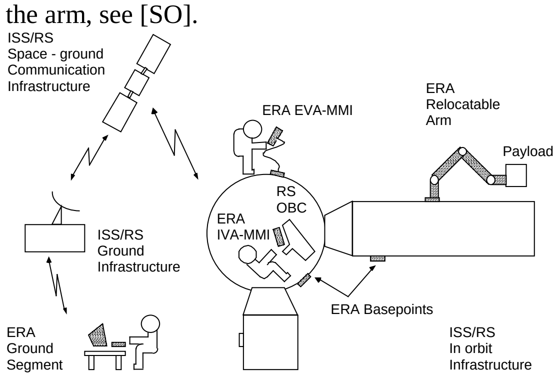

Fig 1: The ERA system

The arm itself contains a multitude of processors. The ERA Control Computer (ECC) is the central nexus for communication with the sub-systems on the arm on one side, and communication with the MMIs (through the Russian Segment Central Post Computer) on the other side. It consists of a main ERC32 processor and three smaller communication and housekeeping processors. The Manipulator Joint Systems (MJS), End Effectors (EE) and Camera and Lighting Units (CLU) each contain two or more processors of their own.

2.Lessons Learned
The ERA program successfully passed the Critical Design Review in the fall of 1999. Although the ERA still has to undergo Final Acceptance, we can already look back and compile lessons learned from the development of this , both technically and organizationally, complex system. With a project which has taken so long to complete, there are almost no areas which do not have elements which (in retrospect) should have been done differently. From early on n the project, we have recorded these lessons before they were forgotten once the arm was delivered. Some of these have already been discussed previously, most notably the dramatic change of the ECC processor from a Thor to an ERC32 (see [PB]).
In this paper we will focus on two aspects, Firmware design and the Man Machine interfaces.

3. There is no such thing as simple Software
The ERA contains close to ten different software systems, developed by as many companies all over Europe. ERA is even dependent on a critical interface with, and functionality contained within, the Russian Segment, the development of which ERA has had little control. Many of the subsystems were initially thought to contain very simple Firmware (or in some cases early on none at all!). Note that the term “Firmware” is used to signify non-maintainable SW. ESA initially required all SW in ERA to be maintainable in flight, not only to allow upgrading, but only to take into account that the performance of the arm in space conditions could not be fully verified on ground, and thus could require modification. This requirement was waived for SW components which were regarded to contain simple functionality, the parameters of which could be modified through the 1553 interface.
The problem with this SW/FW split in practice is threefold: First of all, almost all the FW items became more complex as the sub-system design evolved. Figure 2 shows the average increase in the memory estimate of the FW elements in ERA, starting from system PDR (i.e. when the design was already well underway). The 10% increase over 3½ years does not seems much, but this includes FW items of which the design was straight-forward from the start (e.g. 1553 interface boards). For a

comparison, the trend of the most fluctuating sub-system is shown. The current size is almost 2½ of that initially estimated.

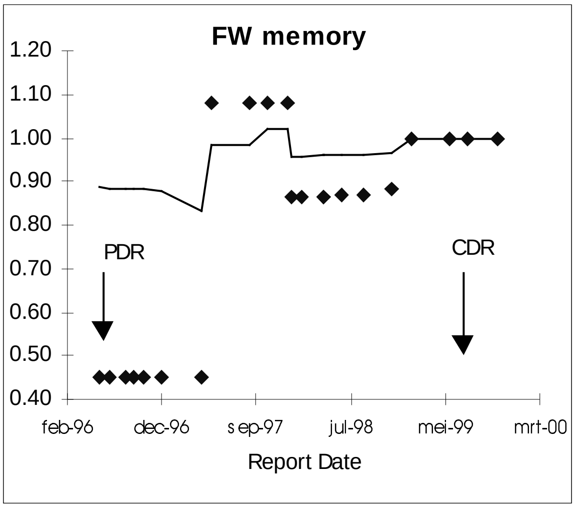

Figure 2: Estimated FW memory, scaled to the final figure. Line: average estimate, diamonds: worst-case sub-system

Secondly, once a decision is made to declare a component FW, it is very difficult to change the design into a maintainable SW system later on. Nevertheless, this was partially done for one sub-system, with significant consequences on both the ECC design and operations. ERA now contains FW with up to 3500 lines of code (based on 10 bytes = LOC), see Figure 3. In some cases, the FW is neither small nor simple with state machines which are more intricate than the ECC.

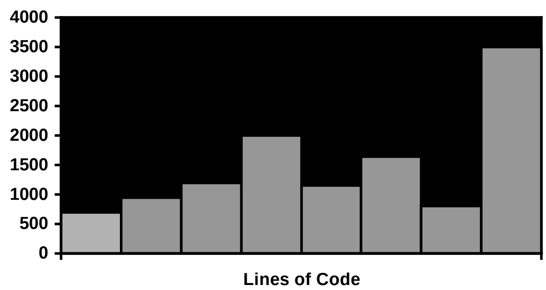

Figure 3.Current estimated Lines of code for FW items

Finally, declaring a component FW contains a dangerous trap. The argument "the code is simple, therefore it requires less rigorous testing and documentation, and therefore less shadow engineering" is both easy to believe (especially when the FW is embedded within a complex HW system which requires much attention) and fallacious. Because the FW is non-maintainable, it requires extra attention, verification and documentation. In ERA (also for maintainable SW) it was frequently not clear just how much and what level of coverage was required for adequate verification. In addition, the development of the systems were constantly out of phase. The management and interface control of these different baselines, the associated shadow engineering efforts, and the problems in achieving workable intermediate integrated systems for EQM testing proved to be quite a challenge. Especially, it was extremely difficult to verify at an early stage that the functional interface between the ECC and the sub-system SW corresponded in detail to the system level concepts. No amount of detail in ICDs can guarantee in advance that two systems are developed such that they can function together to their full extent. Even when the Flight Units were fully developed, detailed tests at system level uncovered features in the S/S FW which clashed with the ECC SW design, and thus required modification of the latter. With so many sub-systems connected, there is always the danger that a correction necessary to achieve correct functioning of one interface results in a problem in another interface.

The lesson to be learned is that even the smallest SW component deserves full attention from the higher-tier contractors. Given the understandable limitation that one person cannot shadow-engineer all aspects of a sub-system, a full time shadow-engineer should be appointed who has the difficult task of monitoring all SW development, making sure that the lower-tier FW developers understand the context of their component with in the system, and making sure that a consistent functional interface is established which allows the entire system not only to function, but also to be operable.

4. The Human Element

The human element, the ERA operator, added an extra complexity. Several reviews by the astronaut community of the Man Machine Interfaces resulted in significant changes. It has to be realized that the ERA design preceded the ISS-level standardization efforts. With almost no precedents (the SSRMS being sufficient different in design not to allow reuse of concepts developed there), Fokker and their MMI subcontractors basically had to invent most MMI related aspects themselves. A good example of this is the design of the EVA Man-Machine Interface (EMMI). As ERA is part of the Russian Segment, the early design of the EMMI was based on discussions with Russian experts. The resulting concept (Figure 4) allowed little monitoring and intervention capabilities during ERA automatic operations, and a number of isolated manual operations. The resulting layout contains a large number of switches with a single function, small display capabilities, and a large Execute handle to confirm automatic commands.

When the decision was made that astronauts from all ISS user nations should be able to operate all robots on and in the station, in as much as possible standard way, as well as more detailed Human Factors Analyses (which resulted in the rejection of the Execute handle because of the excessive strain on the operator), significant changes had to be made. Both the number and complexity of the operations which should be possible with the ERA without ground-planned automatic sequences increases, and the required monitoring capabilities had to be extended. This all had to be achieved within the physical limitations of the existing EMMI box. The result is shown in Figure 5. The number of displays has not increased significantly, but the information which can be displayed has been increased dramatically by allowing the operator to select several display modes. Note that there are still Russian experts which prefer the original layout!

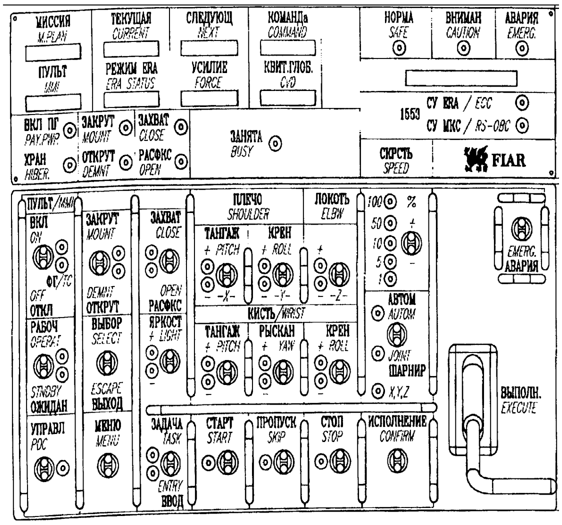

Figure 4: The original EMMI Layout

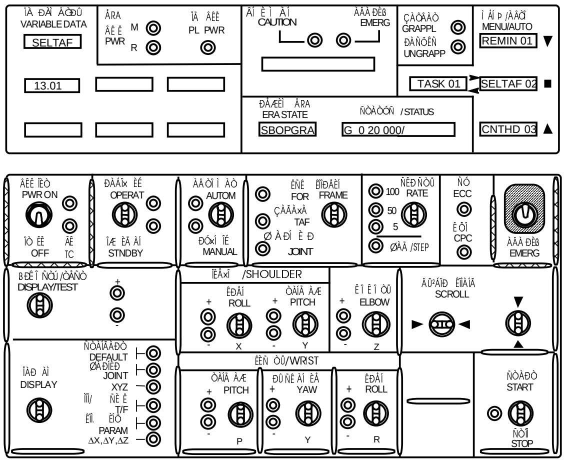

Figure 5: The final EMMI Layout

The MMI layout is now considered frozen unless there is full consensus in the astronaut community that a certain aspect is unacceptable, and even then modifications will only be made if they are feasible (i.e. by modifying SW only). Efforts are underway for several years now to write ISS-level MMI display standards, but they mainly focus on IVA MMIs, are slow in reaching consensus, and all already designed MMIs are excluded from the standard. ERA would have benefited greatly from an already existing mature standard on space-robot MMIs, but understandably these did not exist yet. Hopefully, future MMI designers can profit from the guide-lines originating from the intense scrutiny of the ERA MMIs by the astronaut community.

5. Conclusions

In a long and complex projects like ERA mistakes are made. To avoid similar mistakes in future, it is important to already realize and document lessons learned during the development, not only afterwards. Two important lessons learned in ERA:

Even the smallest and simplest SW component requires full attention, to avoid it becoming large and complex.

In Man Machine Interface design, expect radical changes when the users start using the MMI. Strive for good MMI design guidelines.

References:

[PB]: Beerthuizen, P: “ERA- The Processor Challenge”, DASIA 1999

[SO]: Schoonejans and Oort, “ERA the Flexible Robot”, ISAIRAS 99