ARINC 653 In Space
Dasia 2005, EUROSPACE, Edinburgh, Scotland

N. Diniz
Skysoft Portugal, S.A.
Av. Conselheiro Fernando de Sousa, 19-12º, 1070-072 Lisboa - Portugal
Phone: +351 21 382 93 66
Fax: +351 21 386 64 93
nuno.diniz@skysoft.pt

J. Rufino
Faculdade de Ciências da Universidade de Lisboa
Edificio C8, Campo Grande, 1749-016 Lisboa - Portugal.
Phone: +351 217500254
Fax: +351 217500084
ruf@di.fc.ul.pt

Abstract

In this paper it is proposed that the ARINC 653 avionics standard is a good candidate to take the role of the standard operating system interface building block included in the space on-board software framework defined within the scope of the ESA Harmonization effort. Within the paper, we present a comprehensive overview of the ARINC 653 standard; some needed adaptations with regard to its application to space and future research directions to define the best suited adaptations.

1 INTRODUCTION

The ARINC specification 653 [1] stands in the area of safety critical RTOS for the civil aviation world in the same position as the POSIX [2] specification stands in the area of open operating systems for the business Information Technologies' world. Its purpose is to provide a standardized interface between the Real Time Operating System and the Application Software running over it.

The ARINC 653 standardized interface provides portability to the applications, eases the integration tasks and will open the aeronautics avionics market to software companies providing specialised Commercial Off-The Shelf (COTS) components.

ARINC 653 provides to civil aviation application developers a dependable and fault-tolerant, certifiable, hardware and Operating System (OS) independent, common interface to access resources like memory (through partitioning services), execution time slots, process management, time, process communication and process synchronization in safety critical Real Time Operating Systems (RTOS).

The space world is looking for a standardized interface for the Operating Systems located on board the spacecrafts. Most of the requirements from the civil aviation world that led to

the definition of ARINC specification 653 are also requirements from the space world. The adaptation of the specification to the specific space world needs can thus be performed with minor changes to the original specification and keeping its basic principles.

The ARINC653 concept instantiates one of the building blocks identified in the Data Handling Framework defined in the scope of the Technology Harmonization effort conducted by ESA.

Why should the space community look to the ARINC653 specification and to its concepts?

ARINC 653 and Integrated Modular Avionics (IMA) are the answers provided by the civil aviation world to problems that are also identified in the space world. ARINC 653 is a specification developed by the major air framers and avionics providers, under the umbrella of the ARINC Corporation.

NASA has also been paying attention to the subject and an important reference report is the one from J. Rushby [4]. It discusses the problems related with partitioning (Spatial and Temporal) in avionics systems.

Now it is the time to look to the results of the R&D effort made by the aviation world and use it in the scope of ESA's Technology Harmonization plan.

The building blocks identified for ESA, in the scope of the Technology Harmonisation program, are described in [5]. Among these, one concerns the development of an OS independent interface for space on board applications, to access the resources of the computational node (memory, communication ports, scheduler, synchronization, etc.).

Additionally, the technologic interest of ESA in the ARINC 653 standard was expressed in [6].

The adoption of the ARINC 653 concept in space on board software will provide the application developers with an environment that is standard and independent from the underlying hardware and OS implementation. It will provide to the integrators an easier integration environment together with portable applications. The partitioning concept makes it adequate to software with different degrees of criticality, taking into account all the criticality levels described in [5].
It will allow the re-usage of R&D effort already spent in the scope of another industry domain and will increase the synergies in the development of software for the parallel domains of civil aviation and space with the subsequent reduction in the development costs of on-board software.

2 ARINC 653 In Space

2.1 ARINC653 Overview

ARINC 653 provides to civil aviation application developers a hardware and OS independent, dependable, fault-tolerant, certifiable, common interface to access resources like memory (through partitioning services), execution time slots, process management, time, process communication and process synchronization in safety critical Real Time Operating Systems (RTOS).

ARINC specification 653 is a main component from the Integrated Modular Avionics concept described in [3]. It is hardware and OS independent because the underlying computational resources are accessed through the use of standardized services that must be made available by the Operating System and have a well defined, abstract and language independent interface.

The concept was first deployed in the Boeing 777, using avionics supplied by Honeywell. Its standardization has originally begun with the publishing of the ARINC Report 651 [3]. The first draft of ARINC 653 was published in 1997. Currently, the Airbus A380, the A400M and the Boeing 787 aircrafts will use an IMA architecture with its modules providing an ARINC 653 interface. Being so, the future seems assured in the world of civil aviation for IMA and ARINC 653. The concept proofed its advantages and will be spread into the world of general aviation and helicopters.

Dependability in ARINC 653 is provided by spatial and temporal partitioning whilst Fault-Tolerance in provided by Health Monitoring mechanisms.

2.2 Spatial Partitioning

Spatial partitioning consists in ensuring that it is not possible that an application writes into the memory or data of an application running on a different partition.

In ARINC 653, spatial partitioning is conceptually ensured by the ARINC 653 partitions. A partition in ARINC 653 is like a program in a single application environment, with its own data, context and configuration attributes. It is restricted to use only ARINC 653 services to interface the system. ARINC 653 makes no provisions whatsoever on the method used by the OS to ensure spatial partitioning. It only conceptually requires and assumes it.

2.3 Temporal Partitioning

Temporal partitioning consists in ensuring that the activities in one partition do not affect the timing of the activities in other partition.

In ARINC 653, temporal partitioning is ensured by a fixed, cycle based scheduling. The OS maintains a major time frame (MAF) of fixed duration, which is periodically repeated throughout the module's runtime operation. Partitions are activated by allocating one or more partition windows within this major time frame. The order of the partition activation is

defined off-line at configuration time using configuration tables. This provides a deterministic scheduling methodology since partitions are furnished with a predefined amount of time to access processor resources.

2.4 Health Monitoring

Health Monitoring (HM) is the group of the mechanisms implemented by the OS to monitor and report hardware, application and OS software faults and failures. The HM helps to isolate faults and prevent failures from propagating.

There levels of Health Monitoring are defined within the scope of ARINC 653: process, partition and module Health Monitoring. The first level occurs at the process level. Each partition includes an Error Handler process, which is an ARINC 653 process with the highest priority. All the problems that are not dealt at process level are dealt at partition level. The system integrator predefines health monitoring tables that describe the actions associated to a given failure in the partition. Typically, if the failure is not handled by the Error Handler process, it will be handled by the Health Monitoring actions, defined at partition level. But the error can also be passed to the Module or even System level Health Monitoring. System level Health Monitoring is outside the scope of ARINC 653.

2.5 ARINC 653 Services

The complete set of basic services is described in the part I of the ARINC 653 specification and was defined to meet the requirements that an application implemented using them will be certifiable up to the Level A of airworthiness certification, described in the RTCA DO-178B recommendation [7]. ARINC 653 Part I services are mandatory in any Operating System implementation claiming compliance with the ARINC specification 653.

The services provided are divided in six main classes: Partition Management, Process Management, Time Management, Inter-Partition Communication Services, Intra-Partition Communication Services and Health Monitoring services.

Partition Management deals with the ARINC 653 fundamental concept of partition. A large application can use multiple partitions to run but one single partition instance cannot run on two different processors or in two different computational resources. Partitions are subject to robust space and time partitioning and application partitions interface the system using ARINC 653 services only, allowing, thus, that different applications simultaneously run on the same computational resource without affecting one another on any way.

Process Management deals with processes. Processes are the execution unit of ARINC 653. Only processes have code associated with it. A partition must contain at least one process. The intra partition process scheduler works in a pre-emptive, priority based, algorithm. Processes may be periodic or aperiodic and are never visible outside the partition.

Time Management provides to the partitions the means to control the execution of periodic and aperiodic processes.

Inter-Partition Communication services include inter-partition communication services that are hardware independent. These include Sampling Ports and Queuing Ports. These objects are defined at integration time and created during initialization time. Sampling Ports are communication objects allowing a partition to access a channel of communication configured to operate in sampling mode, that is, each new occurrence of a message overwrites the previous one. Queuing Ports are communication objects allowing a partition to access a channel configured to communicate in queuing mode.

Intra-partition communication services include process communication means, blackboards and buffers, and synchronization means, semaphores and events. These mechanisms are created at (partition) initialization time but don't need to be configured at integration time. Buffers are like queuing ports for intra-partition communications whilst blackboards are like sampling ports for intra-partition communication.

Finally, process Health Monitoring services are provided, allowing the handling of errors at process level, including partition shutdown and restart, if needed.

2.6 Future Development

Up to today, only part 1 of ARINC 653 has been released. However, the ARINC 653 Working Group (WG) has established a roadmap for building a more comprehensive specification including other parts and features.

The part II of the standard will describe optional extensions to the services; namely, file management, logbook management and, possibly, multiple schedule management.

It is expected that ARINC 653 and IMA allow the introduction of the concept of incremental or modular certification. One approach for modular certification is described in [8]. The subject is also under study by the EUROCAE WG-60/SC-200 [9] and was investigated in the scope of several European Commission Framework Programme projects namely, GASCA, NEVADA, PAMELA and VICTORIA [10].

In addition to the concept of modular certification, the concept of compliance is also needed. Since ARINC 653 provides a standard interface, there is the need for the implementers to proof compliance with the standard interface. The part III of the ARINC 653 specification describes the set of compliance requirements that must be fulfilled plus the associated test scenarios. This section is being developed by Skysoft Portugal, under an invitation made by Airbus Deutschland.

2.7 The place of ARINC 653 in Space

In the scope of the Technology Harmonization effort, conducted by ESA, a reference architecture was defined for the Data Handling Software. This architecture definition based itself in the concept of building block(s). The building blocks needed to achieve the desired data handling functionality were identified in [5]. Among these, one concerns the interface with the Operating System.

In order to avoid making the choice of the Operating System a major constraint of the software design, it is important that a standard interface is defined between applications and the underlying Operating System. This was identified already by the civil aviation world and the ARINC specification 653 was defined with the intention to solve it.

We believe that the best approach to define the OS interface building block is to root it on the ARINC 653 (layer). This approach will save significant R&D effort and will allow the development of synergies between the aeronautical and the space domains.

Finally, it is important to highlight that the standardized interface with the OS is just one of the requirements of the approach to deal with Data Handling Software Architecture. This approach identifies the need for standardization of other functions. Computer initialization and self-testing, OS interface, patch and dump functionality, application interpreter and the development framework are those functions.

2.8 Problems solved and answers needed

ARINC 653 provides an answer for the standardization of the OS interface, only. An adaptation of ARINC 653 would also need to address specific space requirements such as computer initialization, patch and dump functionality and space software development framework.

The standard provides a restrictive support to computer initialization, given the specific requirements of civil aviation systems, where most of the work is left to the system integrator and is not included within the scope of the specification. It involves a high degree of offline configuration and testing. For an utilization in space on board software, the standard would need to be extended or, alternatively, procedures would need to be established on how to use the functionality currently supported during computer initialization, making use of configuration tables and/or of the initialization process of the partitions.

Configuration is made using offline configuration tables that describe the partitions, inter-partition communication ports and partition timeslots in the fixed scheduling definition. These configurations are currently fixed during the period of operation of the systems. This would need to be adapted to space needs, including the multiple scheduling option, which could be used, for example, to schedule the partitions and processes in safe mode. It is also important to identify required hardware support for a safe memory partitioning service. This might be implemented as hardware memory block protection. This is the kind of implementation issue that is outside the scope of the standard itself but needs to be solved by any implementation claiming compliance to it.

Concerning patch functionality, a key difference of the requirements from both worlds lies in the fact that the operational cycle of the software in civil aircrafts consists in alternate timeslots of execution and maintenance (when the aircrafts are on-ground), which means that no provisions are needed to support the software upgrade of the systems while they are in operational mode.

Obviously, a completely different situation occurs with the software running in the spacecrafts. In this case, upgrade of the existing applications, or even the upload of new applications, is a mandatory requirement. In ARINC 653, the lifecycle of the partitions running in one computational resource is completely defined by the integrator (the air framer) at integration time, via a configuration table, where, among other things, the timeslots of execution of the partitions are defined. This approach has to be more flexible, in order to be adapted to the space needs. A direction that was already discussed on the ARINC 653 WG is to include multiple schedule management. This can contribute to solve this problem.

Another important missing capability is the automatic generation of the configuration tables based in the information about the applications/partitions that will run on the system, and according to their needs in terms of communication resources and timeslots. It is not certain whether this problem can be solved by the on board software and these calculations might need to be performed on the ground facilities. This problem also raises the question on how and where scheduling analysis will be performed.

Dump functionality will be supported directly by the file and logbook management services that will be published in part II of the standard. It is important to notice that the logbook functionality is mostly a restriction of the file system functionality that can be certified up to Level A of the RTCA DO-178B recommendation, unlike the file system functionality.

Finally, it is clear that ARINC 653 already provides a significant improvement in the development framework available for both application developers and integrators and for the general portability of the developed software.

3 ARINC 653 SUPPORT IN RTEMS

This section describes our ideas on how the Real-Time Executive for Multiprocessor Systems (RTEMS) [11] can be adapted in order to offer the application interface and the functionality required by the ARINC 653 specification. The RTEMS is considered a robust multitasking operating system kernel and it is today a very interesting design component for space applications, given its recent qualification for airborne systems [12].

The RTEMS is an open-source product exhibiting a modular architecture. It supports a wide range of processors through the encapsulation of hardware dependent features in an adaptation layer, known as board support package (BSP), which includes processor architectures as diverse as: the Intel IA-32 architecture, including adaptation to personal computer (PC) platforms; SPARC architectures, such as LEON and ERC32.

3.1 Support to ARINC 653 spatial partitioning

An effective support to ARINC 653 spatial partitioning requires the use of specific memory protection mechanisms usually implemented in a hardware memory management unit.

Such kind of unit is included in the design of Intel IA-32 and it can be added to other processors, such as the SPARC LEON processor. However, the native RTEMS distribution does not exploit the use of memory management unit mechanisms and therefore does not offer, in general, protection against uncontrolled memory accesses.

The RTEMS native distribution uses a flat memory addressing scheme fully compliant with the ARINC 653 process memory access requirements: no use of process private addressing spaces, i.e. sharing of the partition address space with any other process within the partition.

Thus, one fundamental challenge in the implementation of the ARINC 653 specification on RTEMS concerns: the identification of how a memory protection scheme compliant with the ARINC 653 partitioning should be integrated in the RTEMS modular architecture; definition and design of such modules and its interface with each specific hardware platform.

3.2 Support to ARINC 653 temporal partitioning

The native RTEMS distribution dynamically schedules processes for execution using a priority-based pre-emptive scheduling algorithm. These are also the requirements of the ARINC 653 specification for the scheduling of both periodic and aperiodic processes within each partition.

It is worthwhile noticing that the RTEMS native kernel is currently being enhanced with additional temporal protection mechanisms aiming preservation of the timeliness attributes of the system, in the presence of disturbances such as overload and/or faults [13].

The scheduling of the ARINC 653 partitions, which is static, cyclic, based, is not directly supported by RTEMS and must be implemented as additional modules.

3.3 Support to ARINC 653 Health Monitoring

This module should be designed as a set of extensions to RTEMS existing mechanisms and managers, such as: user extensions manager, which execution is scheduled in relevant points of process execution, such as process switching; fatal error manager, instantiated upon de detection of abnormal operating conditions.

3.4 Support to ARINC 653 services

The provision of an application programming interface in conformity with the ARINC 653 specification based on the services provided by the RTEMS kernel concerns different sets of primitives: process management, time management, inter-partition communication services and intra-partition process communication and synchronization services.

The implementation of inter-partition partition communication requires the modification in a given extend to the services provided by the RTEMS kernel. Though different RTEMS native mechanisms, such as the provisions to dual-ported memory mapping (dual-ported memory manager) or multiprocessing environments (multiprocessor manager) exist, these

have to be combined with space partitioning and with the operation of the underlying memory management unit.

The provision of process management, time management and intra-partition process communication and synchronization services is less complex, in the sense that the services offered by the native RTEMS kernel can be wrapped, in general, to provide the application programming interface required by the ARINC 653 specification.

The main extension to RTEMS that needs to be implemented in order to fully support the services provided by ARINC 653 is the Partition Management set of services, which are tightly coupled with the implementation of the temporal and spatial partitioning mechanisms.

4 CONCLUSIONS

In this paper, we have supported the ARINC 653 concept as one of the most suited candidates to be the core of the OS interface building block identified in the Data Handling Framework defined under the scope of the Technology Harmonization effort conducted by ESA.

The main argument is that most of the requirements from the civil aviation world that led to the definition of ARINC Specification 653 are also requirements from the space world. The adaptation of the specification to the specific space world needs can thus be performed with minor changes to the original specification and keeping its basic principles.

The adoption of the ARINC 653 concept by the space world is bound to bring benefits in terms of reduced costs, due to modular certification and lower integration effort. Unique economic advantages can be obtained if the adoption of such concept contributes to the development of synergies between space and civil aviation worlds in terms of R&D efforts and use of COTS components.

The first step that is needed to implement the ARINC 653 concept is to compare the requirements that presided to the definition of ARINC 653 against the similar requirements that are raised by the ESA, in the scope of the Technical Harmonization effort. More specifically, it is necessary to compare the operational context in which the software runs, in the two different worlds, including the operational lifecycle, maintenance and upgrade mechanisms, etc.

Despite the possible differences existing in terms of needs between the two domains, the core of the specification can be directly used in space environments.

This is the way forward to bring standardization to Operating System access in space on board software.

5 REFERENCES

[1] Airlines Electronic Engineering Committee (AEEC), Avionics Application Software Standard Interface (ARINC Specification 653-1), ARINC Inc., 2003.
[2] American National Standards Institute Technical Committee X3J11, Information Technology – Portable Operating System Interface (POSIX), Institute of Electrical and Electronics Engineers Inc., 1996 Edition.
[3] Airlines Electronic Engineering Committee (AEEC), Design Guidance for Integrated Modular Avionics (ARINC Report 651), ARINC Inc., 1991.
[4] J. Rushby, Partitioning in Avionics Architectures: Requirements, Mechanisms, and Assurance, Technical Report NASA / CR-1999-209347, SRI International, 1999.
[5] P. Plancke, P. David, Technical Note on On Board Computer & Data Systems, European Space Technology Harmonisation, Technical Dossier on Mapping, ESA, 2003
[6] J-L. Terraillon, K. Hjortnaes, Technical Note on On-board Software European Space Technology Harmonisation, Technical Dossier on Mapping, ESA, 2003
[7] RTCA SC- 167 / EUROCAE WG- 12, Software Considerations In Airborne Systems and Equipment Certification (RTCA/DO-178B), RTCA Inc., 1992.
[8] J. Rushby, Modular Certification, Technical Report NASA / CR-2002-212130, SRI International, 2002.
[9] www.eurocae.org
[10] http://www.euproject-victoria.org
[11] RTEMS C Users Guide. On-Line Applications Research Corporation, 2003.
[12] C. Cantenot, J. Seronie-Vivien, RTEMS Operating System Qualification is completed. DASIA 2005, EUROSPACE, Edinburgh, Scotland.
[13] M. Coutinho, J. Rufino, C. Almeida. Control of Event Handling Timeliness in RTEMS, (paper submission in preparation), 2005.