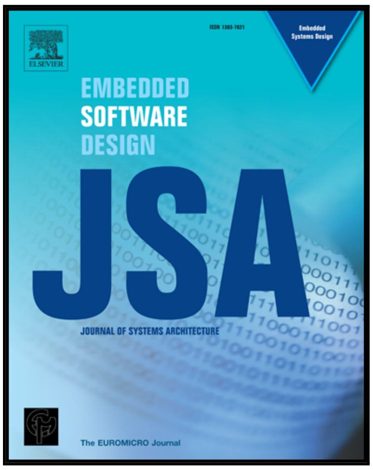
Accepted Manuscript



Integration of Data Distribution Service and distributed partitioned systems

Marisol García-Valls, Jorge Domínguez-Poblete, Imad Eddine Touahria, Chenyang Lu

PII: S1383-7621(17)30074-7
DOI: 10.1016/j.sysarc.2017.11.001
Reference: SYSARC 1471

To appear in: *Journal of Systems Architecture*

Received date: 13 February 2017
Accepted date: 7 November 2017

Please cite this article as: Marisol García-Valls, Jorge Domínguez-Poblete, Imad Eddine Touahria, Chenyang Lu, Integration of Data Distribution Service and distributed partitioned systems, *Journal of Systems Architecture* (2017), doi: 10.1016/j.sysarc.2017.11.001

Integration of Data Distribution Service and distributed partitioned systems

Marisol García-Valls[†], Jorge Domínguez-Poblete[†], Imad Eddine Touahria[†], Chenyang Lu[*]
[†]Universidad Carlos III de Madrid
Av. de la universidad 30
28911 Leganés, Madrid, Spain
{mvalls,jdominguez}@it.uc3m.es
[*]Washington University in St. Louis
1 Brookings Dr.
Saint Louis, MO 63130, USA
lu@cse.wustl.edu

***Abstract*—Avionics systems are complex and time-critical systems that are progressively adopting more flexible (though equally robust) architectural designs. Although a number of current avionics systems follow federated architectures, the Integrated Modular Avionics (IMA) paradign is becoming the dominant style in the more modern developments. The reason is that the IMA concept promotes modular designs where applications with different levels of criticality can execute in an isolated manner in the same hardware. This approach complies with the requirements of cost, safety, and weight of the avionics systems. FACE standard (Future Airborne Capability Environment) defines the architectural baseline for easing integration in avionics systems, including the communication functions across distributed components. As specified in FACE, middleware will be integrated into avionics systems to ease development of portable components that can interoperate effectively. This paper describes the usage of publish-subscribe middleware (precisely, DDS – Data Distribution Service for real-time systems) into a fully distributed partitioned system. We describe, from a practical point of view, the integration of the middleware communication overhead into the hierarchical scheduling (as compliant with ARINC 653) to allow the usage of middleware in the partitions. We explain the design of a realiable communication setting, exemplified on a distributed monitoring application in a partitioned environment. The obtained implementation results show that, given the stable communication overhead of the middleware, it can be integrated in the time windows of partitions.**

I. Introduction

Communication *middleware* and *virtualization* technologies are two main contributions to the development and maintainability of software systems as well as to machine consolidation [9]. These were initially used in mainstream applications, but are progressively entering into the critical environments and complex systems, where their role is increasingly important. In fact, in the avionics domain, the combination of IMA [24] and FACE [21] require the usage of both *virtualization technologies* to develop partitioned systems and *middleware* to ease interoperability and portability of components. This satisfies key requirement regarding cost, space, weight, power consumption, and temperature.

On the one hand, *middleware* brings in the capacity to abstract the low-level details of the networking protocols and the associated specifics of the physical platforms (e.g. endianness, frame structure, and packaging, among others). Consequently, the productivity of systems development is augmented by easing the programmability, maintanability, and debugging.

On the other hand, the penetration of *virtualization* technology has opened the door to the integration of heterogeneous functions over the same physical platform. This effect of virtualization technology has also arrived to the *real-time systems* area. The design of *mixed criticality systems* (MCS) [4] is an important trend that supports the execution of various applications and functions of different *criticality levels* [30] in the same physical machine. The term *criticality* refers to the levels of assurance over the system execution in what concerns failures. For example, in avionics systems, software design follows DO-178B [27] and DO-178C [26] that is a de facto standard for software safety; software is guided by DAL (Design Assurance Levels), and failure conditions are categorized against their consequences: from catastrophic (DAL A) to no effect (DAL E). Then, an MCS is one that has, at least, two functions of different criticalities on the same physical machines.

Over the past 30 years, middleware technology has been applied in critical domains but, mostly, in those subsystems of lower criticality levels. This is the case of, e.g., CORBA applied to combat systems [28] or, recently, DDS [19] applied to control of interoperability of unmanned aircraft and air traffic management[1], mainly for ground segment control. Still middleware is mostly used directly on bare machine deployments; yet it is not used in partitioned software systems.

This paper provides a design of a fully distributed partitioned deployment that integrates Data Distribution Service middleware. To ensure temporal isolation and communication timeliness, hierarchical scheduling is used as the execution framework. It is not the purpose of the paper to contribute new scheduling theory for this domain, but to apply it to the proposed novel distributed partioned design; this design

[1]http://www.atlantida-cenit.org