47821O

Adaptive Management of Computing and Network Resources for Spacecraft Systems[1]

Barbara Pfarr[2], Lonnie R. Welch[3], Ryan Detter[3,4], Brett Tjaden,[3] and Eui-Nam Huh[3]

It is likely that NASA's future spacecraft systems will consist of distributed processes which will handle dynamically varying workloads in response to perceived scientific events, the spacecraft environment, spacecraft anomalies and user commands. Since all situations and possible uses of sensors cannot be anticipated during pre-deployment phases, an approach for dynamically adapting the allocation of distributed computational and communication resources is needed. To address this, we are evolving the *DeSiDeRaTa* adaptive resource management approach to enable reconfigurable ground and space information systems. The DeSiDeRaTa approach embodies a set of middleware mechanisms for adapting resource allocations, and a framework for reasoning about the real-time performance of distributed application systems. The framework and middleware will be extended to accommodate (1) the dynamic aspects of intra-constellation network topologies, and (2) the complete real-time path from the instrument to the user. We are developing a ground-based testbed that will enable NASA to perform early evaluation of adaptive resource management techniques without the expense of first deploying them in space.

The benefits of the proposed effort are numerous, including the ability to use sensors in new ways not anticipated at design time; the production of information technology that ties the sensor web together; the accommodation of greater numbers of missions with fewer resources; and the opportunity to leverage the DeSiDeRaTa project's expertise, infrastructure and models for adaptive resource management for distributed real-time systems.

The majority of real-time computing research has focused on the scheduling and analysis of real-time systems whose timing properties and execution behavior are known *a priori* (i.e., their derivation is based, in part, on theory instead of on experience or on experiment). However, there are numerous applications that must execute in highly dynamic environments (such as Earth and Space Science Missions), thereby precluding *accurate* characterization of the applications' properties by static models. In such contexts, temporal and execution characteristics can only be determined accurately by empirical observation or experience (i.e., *a posteriori*). In most real-time computing models the execution time of a "job" is used to characterize workload *a priori* as an integer "worst-case" execution time. Characterizing workloads of real-time systems using *a priori* worst-case execution times can lead to poor resource utilization, particularly when the difference between WCET and normal execution time is large. The DeSiDeRaTa project is addressing these shortcomings. One of the fundamental

---

[1] Funded by NASA Office of Earth Science, NASA Research Announcement 99-OES-08.
[2] Real-Time Software Engineering Branch; NASA Goddard Space Flight Center; barbara.pfarr@gsfc.nasa.gov.
[3] Laboratory for Intelligent, Real-Time, Secure Systems; School of Electrical Engineering and Computer Science; Ohio University; {welch|tjaden|detter|huh}@ohio.edu.
[4] This work was performed in part while Ryan Detter was in a co-op position in the Real-Time Software Engineering Branch at the NASA Goddard Space Flight Center.

innovations of the DeSiDeRaTa project is the dynamic path paradigm, which is employed for modeling and resource management of large-grain, distributed real-time mission-critical systems.

The effectiveness of adaptive resource management will be demonstrated in a ground based real-time system called ITOS. ITOS, the Integrated Test and Operations System, is a suite of software developed by a group in the Real-time Software Engineering Branch at the Goddard Space Flight Center for control of spacecraft and spacecraft components during development, test, and on-orbit operations. In the future, some portion of this configuration will likely be contained within an autonomous constellation of satellites.

ITOS is a good candidate for adaptive resource management. First, the workload of an ITOS-based system is environment-dependent, and thus varies at run-time in several ways, including:

- The telemetry data rate varies
- The number and types of displays vary
- The number of telemetry streams varies
- The types of elements within a particular telemetry stream varies
- The number of workstations being served with telemetry data varies

Another reason that ITOS is a good candidate is that it has the ability to be dynamically adapted to handle varying workloads via utilization of a pool of distributed computing resources.

By studying the sensors, actuators, applications, and communication relationships within ITOS, we have constructed a model of the dynamic real-time paths of ITOS. Future work includes identifying fault tolerance requirements, inserting probes into ITOS applications (e.g., for events & resource usage), specifying the above properties in D-Spec (the DeSiDeRaTa QoS specification language), obtaining resource usage profiles for the ITOS applications and paths, modifying the applications to be scaleable & fault tolerant, and deploying adaptive resource management into ITOS. The effectiveness of the approach will be determined via extensive experimentation. The components of ITOS will be distributed over a collection of computers, and Network Time Protocol (NTP) will be used to provide global time. Dynamic scenarios will be run and real-time QoS will be recorded. The scenarios will be challenging enough to force resource reallocations. The quality of the adaptive resource management strategy will be assessed by collecting several assessment metrics that we have developed. Ideally, QoS will be improved with each reallocation action. Thus, we will measure the *absolute* and *percentage improvement* in QoS per reallocation action. Other metrics that we will gather are (1) *QoS violation rate* (QVR) – the number of QoS violations per unit of time, (2) the *sensitivity* of chosen allocations to increased tactical load, and the time to perform resource reallocations.

Adaptive Management of Computing and Network Resources for Spacecraft Systems[1]

Barbara Pfarr[2], Lonnie R. Welch[3], Ryan Detter[34], Brett Tjaden,[3] and Eui-Nam Huh[3]

It is likely that NASA's future spacecraft systems will consist of distributed processes which will handle dynamically varying workloads in response to perceived scientific events, the spacecraft environment, spacecraft anomalies and user commands. Since all situations and possible uses of sensors cannot be anticipated during pre-deployment phases, an approach for dynamically adapting the allocation of distributed computational and communication resources is needed. To address this, we are evolving the *DeSiDeRaTa* adaptive resource management approach to enable reconfigurable ground and space information systems. The DeSiDeRaTa approach embodies a set of middleware mechanisms for adapting resource allocations, and a framework for reasoning about the real-time performance of distributed application systems. The framework and middleware will be extended to accommodate (1) the dynamic aspects of intra-constellation network topologies, and (2) the complete real-time path from the instrument to the user. We are developing a ground-based testbed that will enable NASA to perform early evaluation of adaptive resource management techniques without the expense of first deploying them in space.

The benefits of the proposed effort are numerous, including the ability to use sensors in new ways not anticipated at design time; the production of information technology that ties the sensor web together; the accommodation of greater numbers of missions with fewer resources; and the opportunity to leverage the DeSiDeRaTa project's expertise, infrastructure and models for adaptive resource management for distributed real-time systems.

The majority of real-time computing research has focused on the scheduling and analysis of real-time systems whose timing properties and execution behavior are known *a priori* (i.e., their derivation is based, in part, on theory instead of on experience or on experiment). However, there are numerous applications that must execute in highly dynamic environments (such as Earth and Space Science Missions), thereby precluding *accurate* characterization of the applications' properties by static models. In such contexts, temporal and execution characteristics can only be determined accurately by empirical observation or experience (i.e., *a posteriori*). In most real-time computing models the execution time of a "job" is used to characterize workload *a priori* as an integer "worst-case" execution time. Characterizing workloads of real-time systems using *a priori* worst-case execution times can lead to poor resource utilization, particularly when the difference between WCET and normal execution time is large. The DeSiDeRaTa project is addressing these shortcomings. One of the fundamental

[1] Funded by NASA Office of Earth Science, NASA Research Announcement 99-OES-08.
[2] Real-Time Software Engineering Branch; NASA Goddard Space Flight Center; barbara.pfarr@gsfc.nasa.gov.
[3] Laboratory for Intelligent, Real-Time, Secure Systems; School of Electrical Engineering and Computer Science; Ohio University; {welch|tjaden|detter|huh}@ohio.edu.
[4] This work was performed in part while Ryan Detter was in a co-op position in the Real-Time Software Engineering Branch at the NASA Goddard Space Flight Center.

innovations of the DeSiDeRaTa project is the dynamic path paradigm, which is employed for modeling and resource management of large-grain, distributed real-time mission-critical systems.

The effectiveness of adaptive resource management will be demonstrated in a ground based real-time system called ITOS. ITOS, the Integrated Test and Operations System, is a suite of software developed by a group in the Real-time Software Engineering Branch at the Goddard Space Flight Center for control of spacecraft and spacecraft components during development, test, and on-orbit operations. In the future, some portion of this configuration will likely be contained within an autonomous constellation of satellites.

ITOS is a good candidate for adaptive resource management. First, the workload of an ITOS-based system is environment-dependent, and thus varies at run-time in several ways, including:

- The telemetry data rate varies
- The number and types of displays vary
- The number of telemetry streams varies
- The types of elements within a particular telemetry stream varies
- The number of workstations being served with telemetry data varies

Another reason that ITOS is a good candidate is that it has the ability to be dynamically adapted to handle varying workloads via utilization of a pool of distributed computing resources.

By studying the sensors, actuators, applications, and communication relationships within ITOS, we have constructed a model of the dynamic real-time paths of ITOS. Future work includes identifying fault tolerance requirements, inserting probes into ITOS applications (e.g., for events & resource usage), specifying the above properties in D-Spec (the DeSiDeRaTa QoS specification language), obtaining resource usage profiles for the ITOS applications and paths, modifying the applications to be scaleable & fault tolerant, and deploying adaptive resource management into ITOS. The effectiveness of the approach will be determined via extensive experimentation. The components of ITOS will be distributed over a collection of computers, and Network Time Protocol (NTP) will be used to provide global time. Dynamic scenarios will be run and real-time QoS will be recorded. The scenarios will be challenging enough to force resource reallocations. The quality of the adaptive resource management strategy will be assessed by collecting several assessment metrics that we have developed. Ideally, QoS will be improved with each reallocation action. Thus, we will measure the *absolute* and *percentage improvement* in QoS per reallocation action. Other metrics that we will gather are (1) *QoS violation rate* (QVR) – the number of QoS violations per unit of time, (2) the *sensitivity* of chosen allocations to increased tactical load, and the time to perform resource reallocations.