YETE: Distributed, Networked Embedded Control Approaches for Efficient, Reliable Mobile Systems

Sergio Montenegro
Aerospace Information Technology
University of Würzburg,
Germanay
sergio.montenegro@uni-wuerzburg.de

Klaus Schilling
Robotics and Telematic
University of Würzburg,
Germanay
schi@informatik.uni-wuerzburg.de

***Abstract*—An approach for integration of individual processor capacities to an overall distributed mobile system via a network will be presented. Applications considered include robotic cars as well as satellite systems in space applications (both will be called "Vehicles")**

Keywords—Avionics; Communications; Distributed Systems; Task Migration; Cooperative Units; Formation Fly; RODOS Real Time Operating System;

Introduction

A typical space vehicle (space robot, orbiter, lander, rover) integrates many specific computers, which are limited and optimized to their specific tasks. Instead of multiple specialized computers, a distributed data processing concept is elaborated in this project with the objective of optimal use of distributed computing resources including embedded control tasks at component and instrument level. Thus functionalities can be performed at different processors to achieve robust system performance also in case of defects. Such cooperative behaviors require closed control loops via communication links. Challenges relate to the combination of digital, packet oriented, event driven methods for communication with continuous or at least on fixed sampling intervals based approaches from control engineering in particular in real-time critical situations. The solution is based on a minimalistic basic communication system and a “Building Blocks Execution Platform (BBEP)”, running on all computers of the system. For that Purpose we developed the open source dependable real time operating System RODOS ("http://en.wikipedia.org/wiki/Rodos_%28operating_system%29"). Thus applications can be migrated between computers in this system. The basic system measures the load of each included processor in real-time to distribute loads between all processors (central computers, payload and front-ends for devices). In the extreme these cooperating embedded controllers are distributed at several networked vehicles.

The feasibility of these concepts is independent of any communication media and protocols therewith we can use existing wire and wireless communication links even long distance links like for example interplanetary communication. In this way we may implement a network of devices and computers inside of a vehicle. The network my extend dynamically to other vehicles as soon as a communication link is established and even it is possible to build a very wide network where some nodes are in orbit, on the surface of other planets and some nodes on ground (on the earth).

Approach

A typical spacecraft has many devices, each one has its own front end computer. Even if a device is idle, the computing resources of its front end computer are not accessible to other functionalities and similar, if a device requires more computing resources than it is provided by its front end computer, no one else can help.

We aim to banish this border. Instead of having many sometimes very powerful and sometimes very small front end computers, we aim to implement a computer cluster which has enough resources to perform all spacecraft applications and all required functionality for the attached devices. Each device has now just a small front end interface to the computing cluster. The interconnection between front end interfaces and computing cluster elements will be implemented in the first step using wired and in the next step using wireless communication links.

At the end, instead of many specialized IO-computers plus board computer, we will have only a few powerful computers in a cluster. This computer pool offers scalable computing power and reliability. By turning units on and off we can adapt the system to the current system requirements, including power consumption, computing performance and grade of redundancy.

After we have wire less links between devices and computing cluster elements, for the communication and cooperation, there is no border or barrier between several spacecrafts or rovers (lets call them "vehicles") as long as they are close enough. Now vehicles can support each other with computing power and by providing direct access to any sensor or actuator.

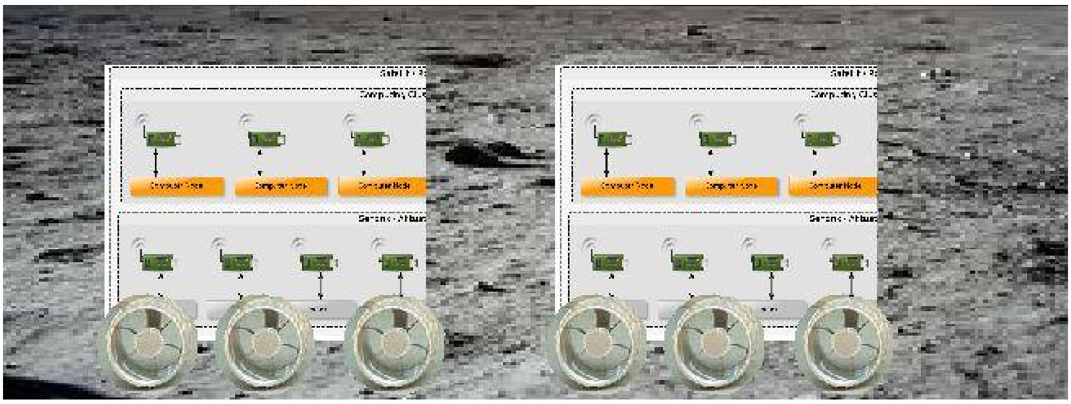
Example Scenario

In this scenario the whole system consists of multiple cooperating space vehicles which are communicating with each other by relatively powerful radio signals. This inter-vehicle-communication link can have a range of several kilometers.

In each vehicle the modules - computer, actuators, sensors – are communicating by low range radio signals in a range of only a few meters (max. 10). But there is also an additional more powerful transmission device for several kilometers (up to 100 km for satellites, up to 2 km for robot vehicles) to enable data transfer between units of one formation.

We work with several cooperating space vehicles. If one (A) has a problem, a second vehicle (B) may approach, so that B can receive sensor data from A and trigger/control its actuators directly without requiring any interaction with the board computer of A.

B can not only diagnose the state of A but it can also control A with its own sensors and actuators for example to move it to another place.



Two Vehicles where all internal devices and computer may be accessed by both.

This strategy will allow:

1. Several vehicles together can perform computations which need a higher computational power than the one existing in a single vehicle.

2. Even if the board computer of one vehicle fails completely, another vehicle can diagnose the failed one and maybe trigger a recovery.

3. Any vehicle can use sensor data from any other vehicles, e.g. sensor fusion.

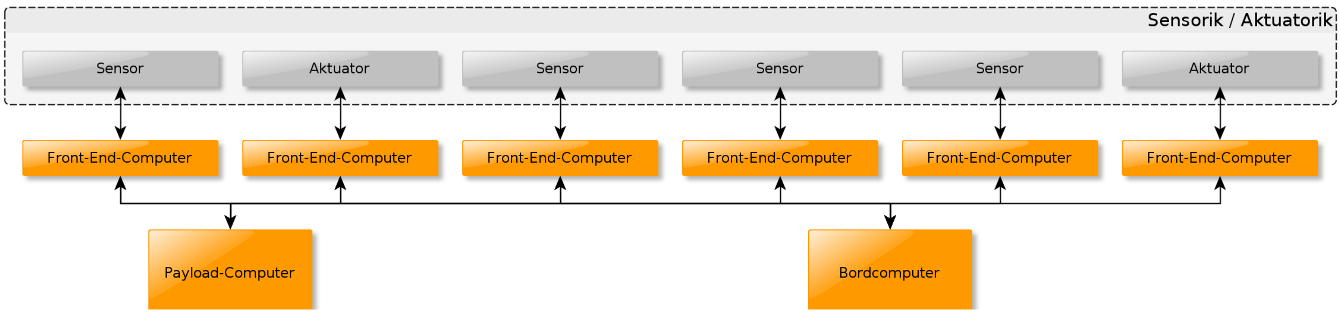
4. If one vehicle is damaged completely, the still working sensors and actuators can be used by the other vehicles.

5. If the board computer of a vehicle is overloaded, some of the software tasks can be executed on other computers on other vehicles.

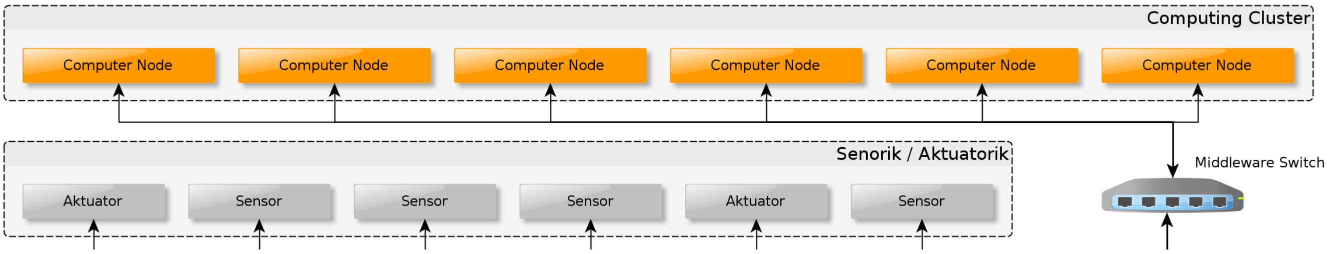
Development Steps

The final functionality of YETE will be implemented step by step in 5 steps or models:

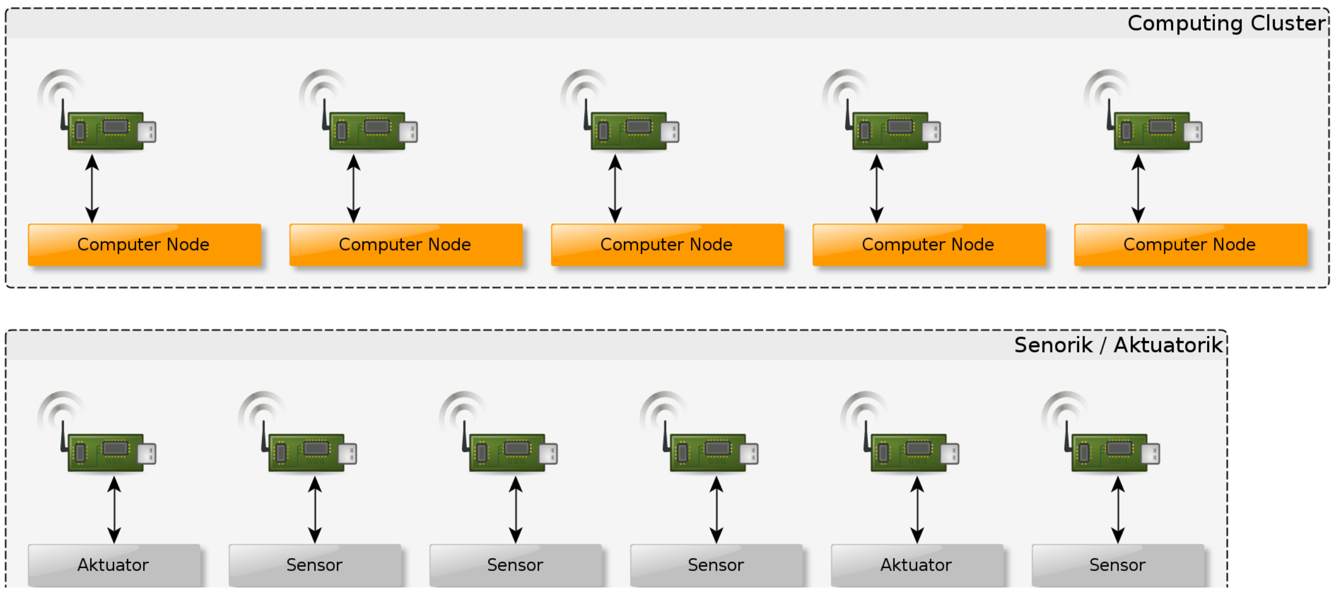
1. In our first model (June 2014) we will replicate the typical space craft hardware: We have a board computer plus few devices each with its own front end computer. From the Hardware point of view nothing is new, but the software is different. All Computers (Board computers and device front end computers) have a common software infrastructure (RODOS) which allows task to migrate from any computer to any other. Now the software sees no boundaries between device front ends and other computers. Applications may be executed anywhere where we have free computing power.



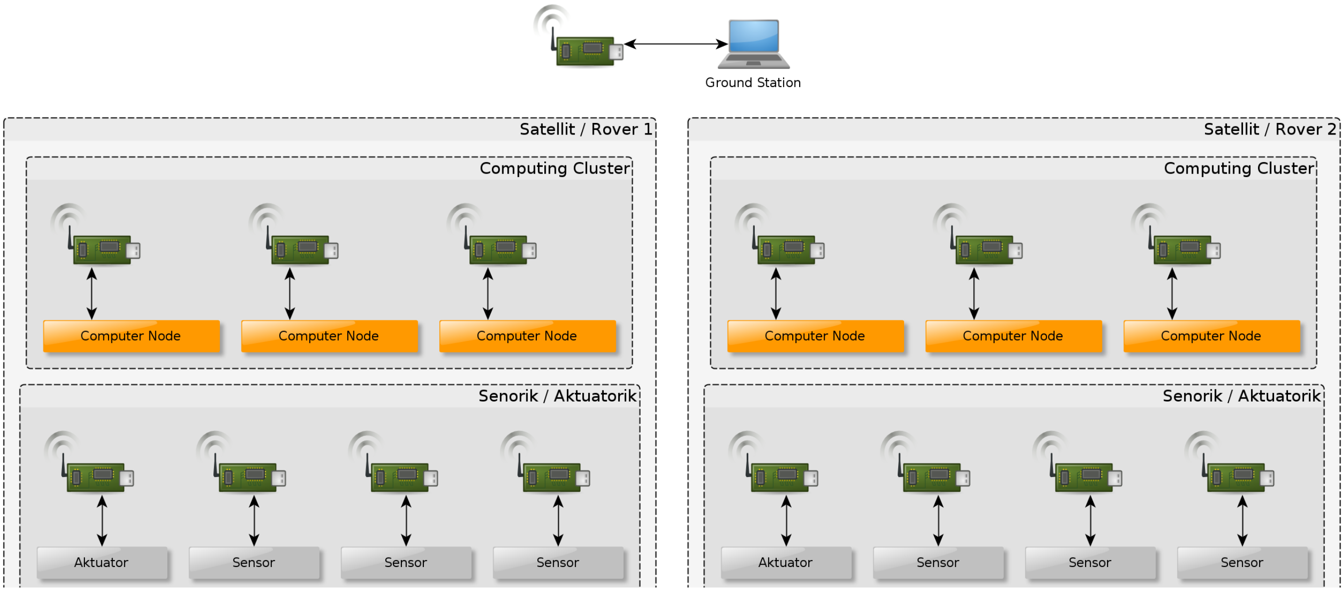
2. In our second model we remove the front end computers from the devices. Instead of such, we will have only very simple SOC (System On One Chip) Interface to the computer network. Instead of a single board computer we have now a cluster of several computers. Like in Model 1 software applications may migrate from any node to any other. Now we have fewer computers and higher functionality and flexibility. The tasks required by the devices may be executed now on any computer in the network.



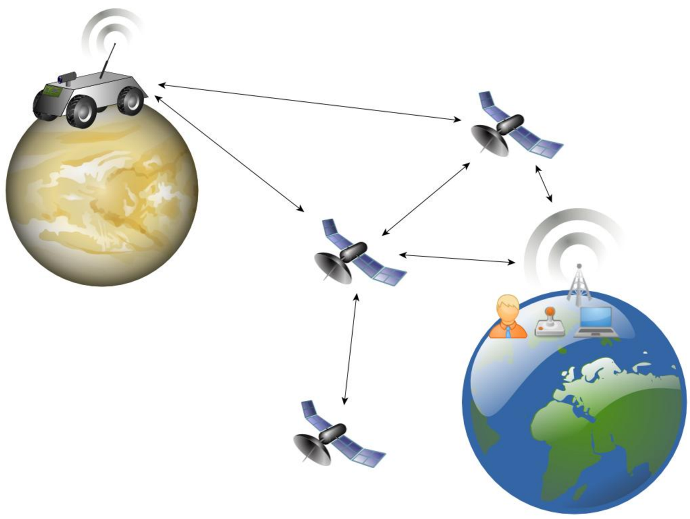
3. Until Model 2 we used wires to interconnect devices and computers. In model 3 we remove the wires and use instead of wires a wireless protocol like for example WiFi. Devices will get an SOC interface to the wireless network. In this SOC we will implement the wireless protocol as well as the required interface to the corresponding device. Now we can just attach new devices to the vehicle without having to change software or hardware. A very big drawback of current space computers is that their hardware has to be tailored to the attached devices. If we have another vehicle configuration with other devices, we have to redesign the board computer. But now, in Model 3, a computer needs only a single interface to the wireless network and it may be developed independently of the attached devices.



4. The step from model 3 to model 4 is very small. Now we just have to bring two or more of the vehicles from the model 3 in proximity, and now any computer and any device from any vehicle may communicate with any other. Software tasks may now be distributed among vehicles and any vehicle may use any device from any other.



5. In model 5 we bring the operator into the loop (Man in the Loop MIL). We aim to allow an operator real time control of remote vehicles using estimator and predictor models to overcome long communication delays.



Acknowledgment

Funded by the Space Agency of the German Aerospace Center (DLR) with federal funds of the German Federal Ministry of Economics and Technology (BMWi) under 50RA1330