Operations System vs. Operating System
Towards a Ground System Supporting Satellite Application Programming

University of Stuttgart
Germany

Kai Leidig, Steffen Gaißer, Susann Pätschke, Robin Schweigert, Sebastian Wenzel, Sabine Klinkner

Problem Statement

Typically, the term operating system refers to a software component that controls the resources and the processes of a computer, providing appropriate interfaces for the implementation of custom user applications. This definition is common for ordinary computer systems. Yet, what if the operating system and a corresponding application are physically separated, because the computer is within a satellite in space, while the user program is executed on ground? (Figure 1) Then, capabilities must be created to connect both, which is of course complicated by the natural boundaries in satellite communication, for example the limited satellite contact times.

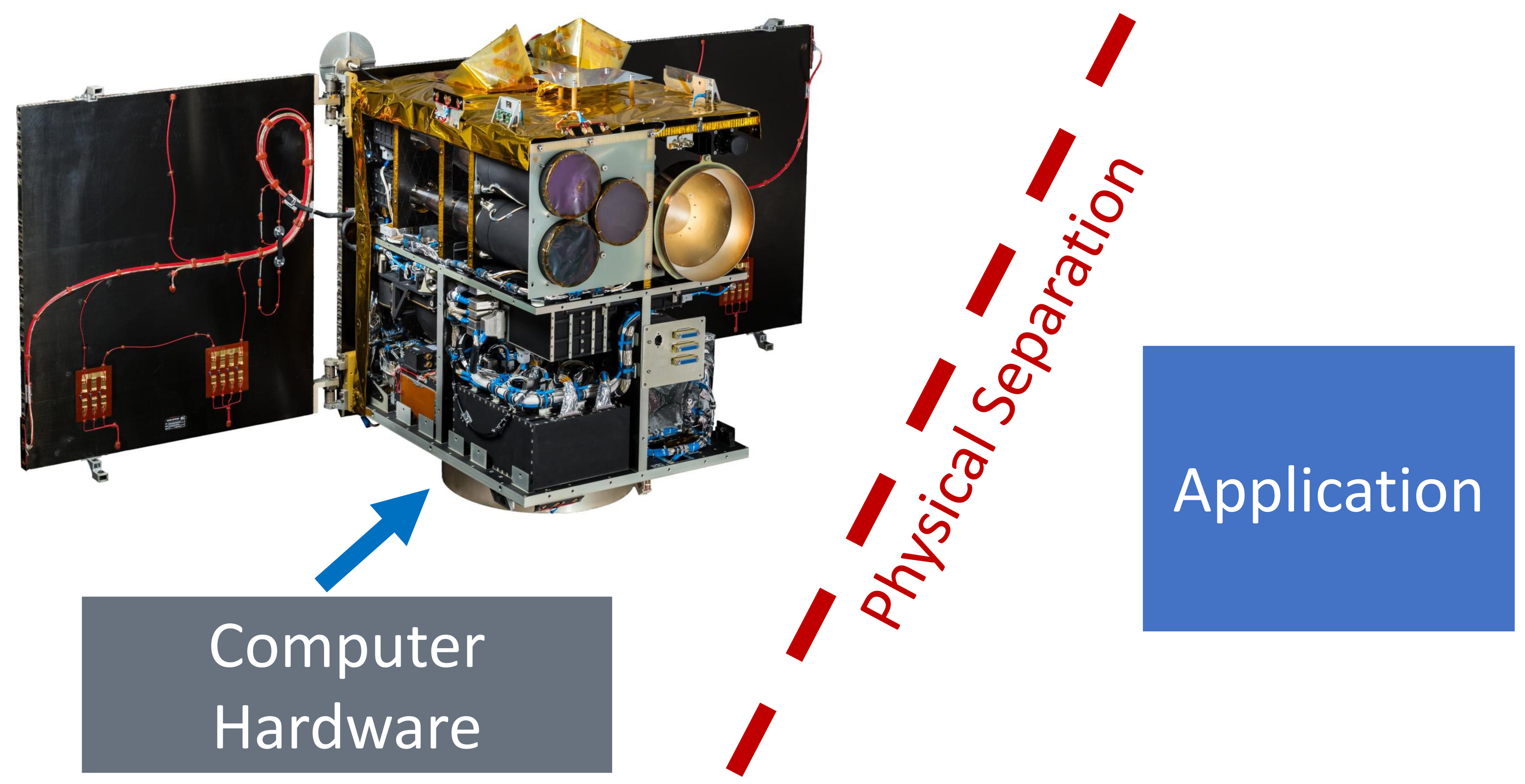

***Figure 1:** Physical Separation of Computer and App. on Ground*

Data Abstraction

Imagine the following case! A smartphone app shall be written, enabling the user to select a spot on Earth and to trigger an observation of the selected spot. Further imagine, that the respective satellite system for the observation is in place, including the appropriate ground segment. How can such an app be connected to the system?

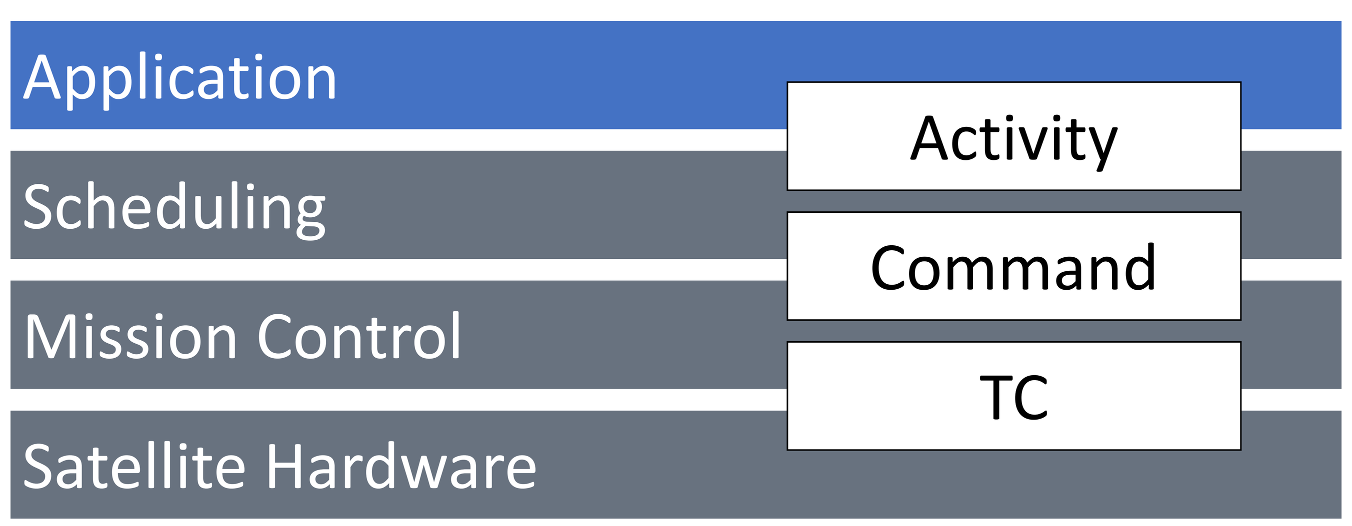

***Figure 2:** Proposed Data Abstraction Scheme (uplink only)*

An application can only be attached to an operating system, if the underlying architecture provides the appropriate abstraction layers. Current ground data handling systems are lacking of application layer protocols though. (Figure 2)

Control Process

Furthermore, in order to compensate for the limited contact times, a neat process needs to be set-up to map and control the state of the operated satellite on ground, and thus to engage an unambiguous scheduling. That control process cannot be implemented by just a single loop though. Instead, it covers nested control processes on all layers depicted in Figure 2. It begins with the release and verification of telecommands (TC) and ends with the automatic scheduling of high level processes and the verification of a successful execution based on the received data products.

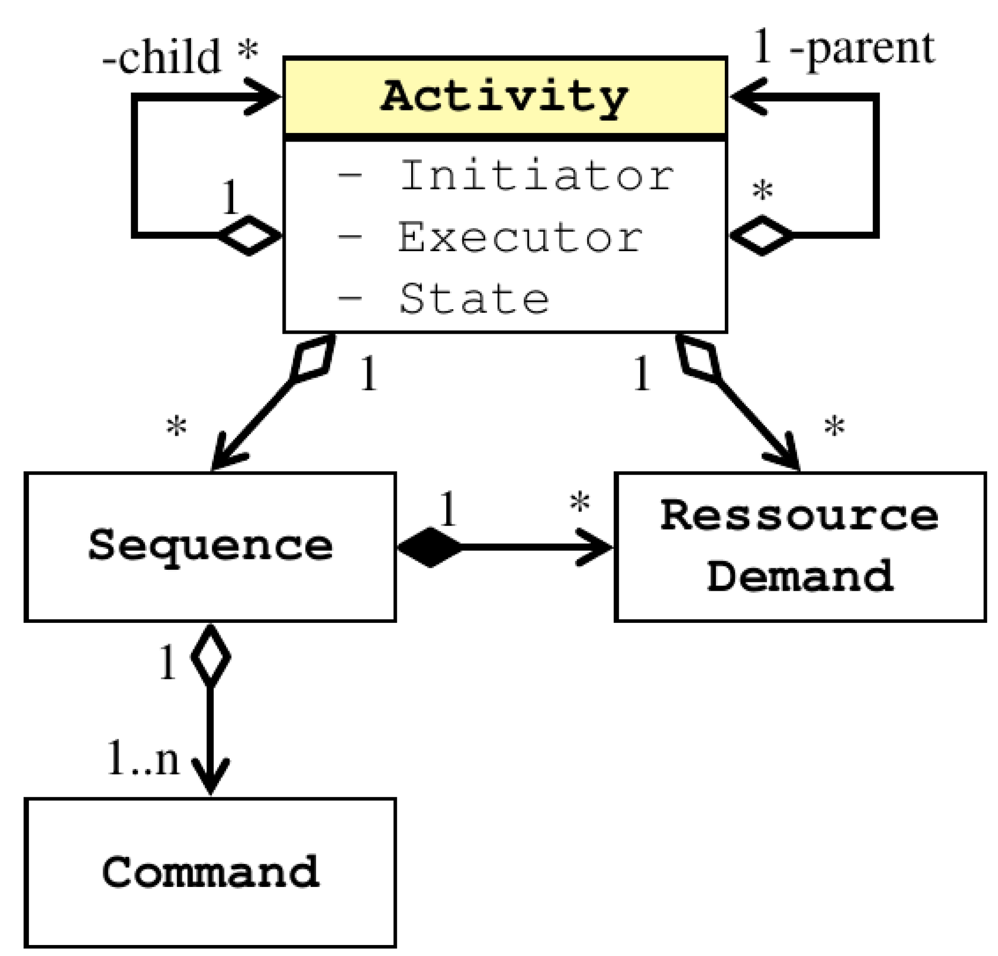

***Figure 3:** UML Diagram of an Activity Object*

The Activity

The proposed interface type allowing an application to engage a process to be executed by the satellite is the *Activity* (Figure 3). An Activity describes a process that alters the state of the operated system in a predefined manner. That alteration is quantified by the *Resource Demand* object. On the one hand, this allows for the propagation and thus for the forecast and the verification of the system state. On the other hand, this is a means of conflict resolution between activities too.

Commands to be sent to the satellite are organized in sequences. Purpose of the underlying Mission Control (Figure 2) is the release and the verification of such commands, while the Activity is verified on the basis of the resulting system state.

Nesting Activities further allows for the decomposition of tasks into subtasks. By assigning each subtask to a different executing system, satellites can collaborate in order to achieve a common objective.

Agents

Through the implementation of an abstraction layer automatically managing the resources of the operated satellite, the proposed operations system fulfills the same principle tasks as an operating system for an ordinary computer. Any application implementing the introduced interface can be connected to our system. They are referred to as *Agents*.

Catching up with the example in the beginning: The proposed system could be extended by an intuitive and user-friendly frontend to plan custom Earth observations, in which a user selects a spot on Earth and triggers the observation without getting in touch with the underlying control process (Figure 4).

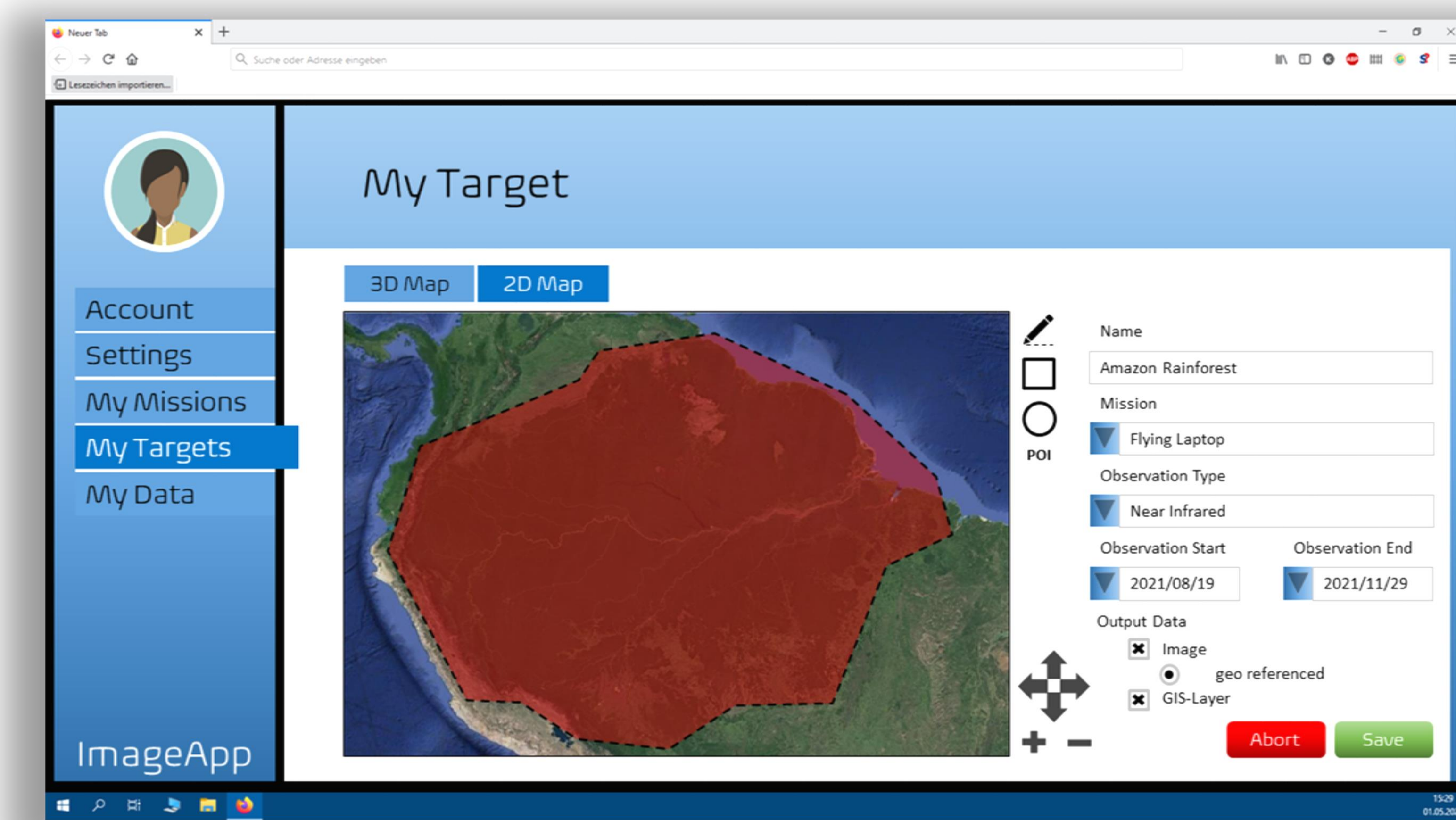

Figure 4: Example Application (Agent)

Kai Leidig

University of Stuttgart, Institute of Space Systems
leidig@irs.uni-suttgart.de
+49 711 / 685 65862