PROPOSED SOIS PLUG-AND-PLAY ARCHITECTURE AND RESULTING REQUIREMENTS ON SPACEWIRE MAPPING

Session: SpaceWire Networks and Protocols

Long Paper

Stuart D. Fowell

SciSys UK Ltd,
Clothier Road,
Bristol,
BS4 5SS,
UK

Chris Taylor

European Space Technology Evaluation Center,
2200 AG Noordwijk,
The Netherlands

E-mail: stuart.fowell@scisys.co.uk, chris.taylor@esa.int

ABSTRACT

This paper describes the initial concept of the proposed Plug-and-Play architecture of the SOIS area of the CCSDS, and the resulting requirements on the mapping of SOIS onto SpaceWire. Firstly, this paper defines the term "plug-and-play" in the scope of SOIS and a number of SOIS use cases for "plug-and-play", so as to derive requirements that must be met by a SOIS Plug-and-Play architecture. Secondly, this paper proposes a tentative SOIS Plug-and-Play architecture, based on initial analysis of the use cases and consideration of existing plug-and-play technologies, e.g. USB 2.0, IEEE 1451, 1-wire as well as proposals such as those already made for SpaceWire. Finally, this paper proposes draft requirements of the mapping of the SOIS Plug-and-Play architecture onto SpaceWire. As this is an initial concept paper, it is hoped that it will generate debate and feedback on the SOIS Plug-and-Play initiative that will, of course, be gratefully received and taken into account.

1 EXISTING SOIS ARCHITECTURE

The Consultative Committee for Space Data Standards (CCSDS) [1] was founded in 1982 by the major space agencies in the world to discuss and define common space communications issues, to enhance governmental and commercial interoperability and cross-support, while also reducing risk, development time and project costs.

As part of the CCSDS work, the Spacecraft Onboard Interface Services (SOIS) area is developing standards to radically improve the spacecraft flight segment data systems design and development process by defining generic services that will simplify the way flight software interacts with flight hardware and permitting interoperability and reusability both for the benefit of Agencies and Industrial contractors. As part of the standardisation process for SOIS, a subnetwork-neutral architecture of services has been defined [2], as illustrated in Figure 1. Mappings of these services to capabilities of specific subnetworks are then defined, e.g. protocols on SpaceWire, MIL-STD-1553B and CAN. This allows, amongst other benefits, for satellite architectures to be re-used across different busses and standardised off-the-shelf devices and subsystems to be developed.

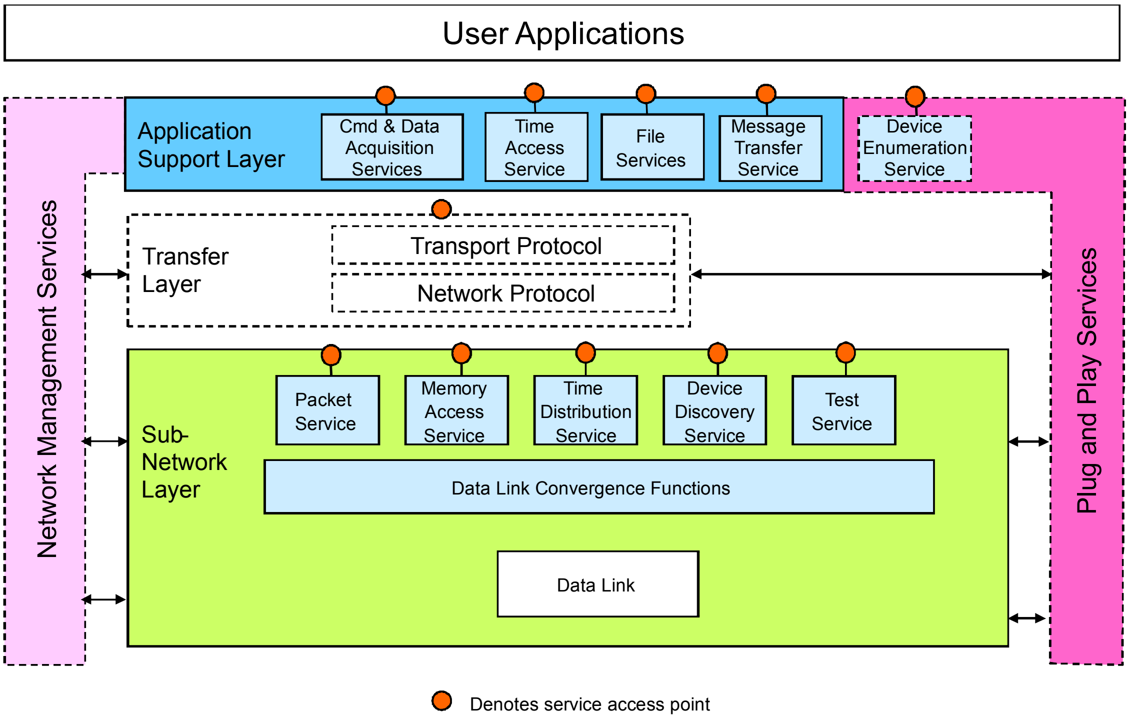

Figure 1: CCSDS SOIS Architecture

Of relevance here are the following services:

- **Command and Data Acquisition Services**, that provide mechanism for commanding of and acquiring data from devices within a spacecraft;
- **Message Transfer Service**, that provides transfer of messages between software applications within a spacecraft;
- **Packet Service**, that provides transfer of packets between data systems[1] within a subnetwork of a spacecraft;
- **Memory Access Service**, that provides access to memory locations of a data system from another data system within a subnetwork of a spacecraft.

The first set of standards is currently being reviewed by the various Space Agencies. The ECSS are currently developing protocols to provide the mappings onto

---

[1] Within ISO standards, communication is defined as being between "data systems", a generic term that can be taken, within the SpaceWire context, as mapping onto a node within a SpaceWire network. A node is defined as any addressable SpaceWire network entity, be it a processing, IO, or memory module, a transducer, an instrument etc.

SpaceWire and MIL-STD-1553B and a similar exercise is planned in 2008 for CAN. For SpaceWire, the RMAP protocol is suitable for the SOIS Memory Access Service and the SpaceNet project is prototyping a SpaceWire packet protocol for the SOIS Packet Service.

However, these standards only address a static or top-down configured communications architecture. In addition, support for "wireless" capabilities is being considered and by their nature can result in a more dynamic communications architecture.

To address these issues, it has been identified that the SOIS architecture needs extending to support "plug-and-play" concepts and a "Birds-of-a-Feather" (BoF) grouping has been organised to address this. This paper summarises the current position of the SOIS Plug-and-Play BoF, building on previous positional papers [3, 4].

2 PLUG-AND-PLAY REQUIREMENTS

Before we can define a SOIS Plug-and-Play architecture and its mapping onto SpaceWire, it is important to clarify the term "plug-and-play" and define appropriate use-cases within the SOIS domain. From these can be extracted requirements on the SOIS plug-and-play architecture.

2.1 Definition of "Plug-and-Play"

The Wikipedia definition of "plug-and-play" is as follows:

> "**Plug and play** is a computer feature that allows the addition of a new device, normally a peripheral, without requiring reconfiguration or manual installation of device drivers. … Modern plug-and-play includes both the traditional boot-time assignment of I/O addresses and interrupts to prevent conflicts and identify drivers, as well as hotplug systems such as USB and Firewire."

Translated to a spacecraft domain, "peripheral" should include onboard computing modules such as processing, IO and mass memory modules, as well as devices traditionally associated with avionics, from this simple (e.g. thrusters, magnetometers, thermistors) to the more complex (e.g. star trackers), and simple instruments. It has been considered that perhaps plug-and-play should extend to including the integration of whole sub-systems, but has been discounted as this is where the line between where device integration and software system integration occurs, and so is beyond the scope of SOIS. This limits SOIS Plug-and-Play to the establishment of communication of the SOIS Command and Data Acquisition Services of the Application Support Layer and the Subnetwork Layer Services.

Therefore, we limit the definition of "plug-and-play" in a SOIS context to:

> "the mechanisms necessary to establish communication services between two data systems in a spacecraft's onboard (sub-)network, without requiring reconfiguration or manual installation of device drivers by any user (higher-level service or OBSW application)."

2.2 SOIS Plug-and-Play Use Cases

In conjunction with defining the term "plug-and-play", the use cases to be solved by plug-and-play within the SOIS domain need to be captured. The following is a summary list of those use cases identified to date:

- **Dynamic Spacecraft Network Reconfiguration** – activation of redundant devices upon a flying spacecraft in response to faults. A Fault Detection, Isolation and Recovery (FDIR) system application simply powers up replacement. Reconfiguration happens automatically (bottom-up), rather than hierarchical (top-down);

- **Spacecraft Integration & Test** – Electrical Ground Support Equipment (EGSE) connection to Spacecraft under test using wireless technologies;

- **Rapid Spacecraft Assembly of Devices** – to reduce/eliminate the need for aspects of Spacecraft database for configuring OBSW;

- **Biometric Health Monitoring of ISS/Orbiter crew** – characterised as facilitating the incorporation of heterogeneous sensing and control devices in a wireless, heterogeneous communications network [8].

It is important to also identify the use cases that are out-of-scope for SOIS Plug-and-Play (though that is not to say that SOIS Plug-and-Play may not have a role to play within them):

- **Onboard Software Upgrade or Reconfiguration** – covering mode changes or software updates. This is purely a software change with no new data systems introduced, so there is no reconfiguration of the SOIS communication services required, and so out of scope of SOIS Plug-and-Play.

- **Rapid Spacecraft Assembly of Subsystems** – while SOIS Plug-and-Play will simplify at the subnetwork layer the integration of subsystems with other subsystems and/or complex instruments, integration also requires a complex exchange of information using perhaps a software framework or middleware that is beyond the present scope of SOIS. However, such a software framework would exchange messages using the SOIS Message Transfer Service. Therefore SOIS Plug-and-Play greatly aids but itself does not fully solve this use case.

2.3 SOIS Plug-and-Play Requirements

From these uses cases, a tentative set of requirements of SOIS plug-and-play can be extracted: the SOIS Plug-and-Play architecture shall:

1. support a mechanism to discover new data systems added to (powered up, mechanically inserted, sending announcement packet) a SOIS subnetwork;

2. support a mechanism to discover old data systems removed from (switched off, failed, mechanically removed, out of range, electing to withdraw) a SOIS subnetwork;

3. support a mechanism to discovery of capabilities of added data systems;

4. support a mechanism to reconfigure SOIS communication services to allow communication to and from added data systems;

5. support a mechanism to reconfigure SOIS communication services to remove information about removed data systems;

6. support a mechanism to notify users (applications and higher layer services) of added and removed data systems and their capabilities.

These are a preliminary set of requirements, which are expected to be expanded and refined over the process of defining the SOIS Plug-and-Play architecture within the multi-agency discussion and debate that is inherent in the CCSDS approach.

3 TENTATIVE SOIS PLUG-AND-PLAY ARCHITECTURE

Having established the requirements for a SOIS Plug-and-Play Architecture, the architecture itself must be designed. Within SOIS, the “adopt-adapt-innovate” approach is used, i.e. adopt an existing standard or technology if it meets the requirements, otherwise adapt (i.e. modify) an existing standard or technology if it can be changed to meet the requirements, and failing either, innovate (i.e. develop) a novel technology to meet the standards.

To do this then, a survey of existing standards, technologies, and prototypes is required.

3.1 Existing Plug-and-Play Technologies and Studies

From work sponsored and directly performed by NASA and ESA, the following plug-and-play technologies have been identified:

- **USB 2.0** – ubiquitous serial bus for data exchange between host computers (typically PCs) and a wide range of simultaneously accessible peripherals. The bus allows peripherals to be attached, configured, used and detached while the host and other peripherals are in operation. USB provides most of the characteristics required for plug-and-play. However, it has a rigid topology and allows for only a single host computer [5].

- **IEEE 1451** – a standard for a Smart Transducer Interface for Sensors and Actuators, designed to ease connectivity of sensors and actuators into a device or field network. Uses a common object model for smart transducers along with a smart transducer interface module (STIM), a transducer electronic data sheet (TEDS), and a digital interface to access the data. A variety of standards within this set define access to analogue, digital and smart transducers across a variety of communication protocols, including wireless [6].

- **1-wire** – a device communication bus system from Dallas Semiconductor that provides low-speed data, signalling and power over a single wire. The bus includes a mechanism for recovering the address of every device on the bus. As the device address includes the device type so that this can produce an inventory of all devices on the bus [7].

- **wireless technologies** – by their very nature of integrated independent data systems, wireless technologies support plug-and-play concepts. The architectures of wireless technologies are still being considered for their inclusion within SOIS Plug-and-Play.

In addition, a number of projects have produced prototypes of plug-and-play architectures and/or technologies:

- **BioNet** – BioNet is a network-transparent device-driver and application-client framework. It is a general middleware solution for the integration of disparate data-producing endpoints over heterogeneous wired and wireless networks. The BioNet architecture is a highly-scalable, decentralized, asynchronous, publish/subscribe message bus [8].

- **SpaceWire Plug-and-Play Prototyping** – within the SpaceWire community, a working group coordinated by NASA GSFC has been established and a number of prototyping initiatives for supporting plug-and-play within SpaceWire networks, primarily focussed on network mapping and router (re-) configuration [9]

Of course, it is a requirement on SOIS that it is independent, but mappable onto, network specific plug-and-play technologies.

3.2 Proposed SOIS Plug-and-Play Architecture

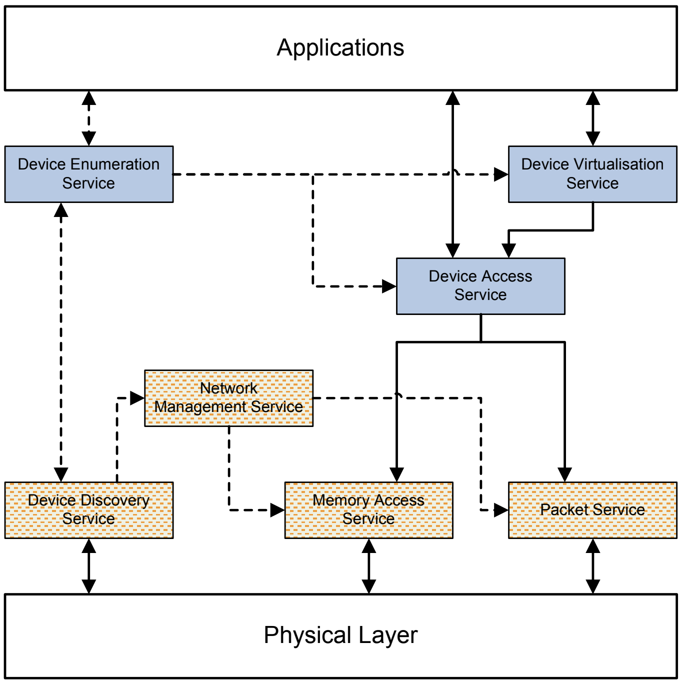

Figure 2: Proposed CCSDS SOIS Plug-and-Play Architecture

The following is a tentatively proposed SOIS Plug-and-Play Architecture, see figure 2. It is primarily focussed on device plug-and-play with particular consideration for wired devices. It should be expected that this will evolve as plug-and-play of more generic data systems and wireless "busses" are considered.

At the heart of the SOIS Plug-and-Play architecture is the **Device Enumeration Service**, which is responsible for managing the discovery of a new device and its insertion into the SOIS communications architecture.

A Subnetwork **Device Discovery Service** is used to discover new devices. This may implement a specific discovery mechanism, e.g. by broadcasting for new devices, or react to a subnetwork-specific event, e.g. a trigger that a new device has been powered up or inserted into the subnetwork. It is assumed that this service is also responsible for allocating or obtaining a subnetwork-specific address for the new device. This allows the Subnetwork Layer Services to be reconfigured to allow communication from within the SOIS communications architecture with the new device.

It is expected that a Subnetwork **Network Management Service** (to be provided as part of subnetwork protocol definitions) is responsible for any reconfiguration of the Subnetwork Layer services that may be required, e.g. updates to SpaceWire Router GAR Tables.

Another key concept is the provision by each SOIS Plug-and-Play-enabled device of an **Electronic Data Sheet** (EDS), that defines the device type and capabilities (e.g. functions, protocols and classes-of-service supported). This, together with the subnetwork-specific address of the new device, allows the Command and Data Acquisition Services and/or Message Transfer Service to be reconfigured to allow application-level communication with the new device. It is assumed that either the Subnetwork Memory Access or Packet Service will be used to obtain the device's EDS.

So, to summarise the steps required to "plug in" a new device (i.e. power up):

1. Device is powered up.

2. Subnetwork specific Device Discovery Service discovers the device and either discovers its address or allocates it.

3. The subnetwork specific Network Management Service performs any necessary configurations to allow communication with the device from any other data system in the subnetwork.

4. The Device Enumeration Service is notified of the new device, including its address. The Device Enumeration Service uses the subnetwork specific Memory Access Service to read the device's EDS to discover its capabilities, e.g. device class. The device may be configured to a default setting.

5. The Device Enumeration Service configures the Device Virtualisation Service and Device Access Service so that users may command and acquire data from the device.

6. Finally the Device Enumeration Service notifies a registered OBSW application that a new device has been plugged into the system.

This approach allows for both static and dynamic systems to be deployed, where the dynamic SOIS Plug-and-Play services manipulate the configuration of the potentially static services.

4 Requirements on SpaceWire Mapping

Given the current status of the SOIS Plug-and-Play initiative, anything written in this section of the paper must be considered speculative!

To date the SpaceWire community have focused on network mapping and discovery of attached and detached nodes. These map nicely to the functionality of the SOIS Device Discovery Service.

Beyond this, the Device Enumeration Service will require a standardised mechanism to discover the capabilities of a node is required, i.e. provision of the device's EDS. RMAP provides a generalised method that can be used to access the EDS data. However, where is EDS located, what address is it at? Is this standardised or perhaps dynamically obtained? In addition, the content of the EDS (structure, types, etc.) remains to be defined. Probably it will not be specific to SpaceWire.

However, one thing is certain. The SOIS Plug-and-Play Architecture has to date primarily focussed on SpaceWire. As it develops, it must encompass other bus types, both wired and wireless. The diverse requirements and technologies must be taken into account in the difficult balancing act between the contradictory requirements of the genericity of the SOIS architecture and the resource constraints of actual spacecraft.

References

1. Consultative Committee for Space Data Standards, www.ccsds.org.

2. "Spacecraft Onboard Interface Services – Informational Report", CCSDS 850.0-G-1, Green Book, Issue 1.0, Washington, D.C, June 2007.

3. David P, "Online Remote Maintenance", CCSDS Fall 2003 Working Meeting, Location, 20th October 2003 (available from author).

4. Plummer C, "Plug and Play for Spacecraft Onboard Systems", CCSDS Spring 2005 Working Meeting, Athens, Greece, March 2005 (available from author).

5. "USB 2.0", http://www.usb.org/home.

6. IEEE 1451 Standard for a Smart Transducer Interface for Sensors and Actuators, http://ieee1451.nist.gov/intro.htm.

7. "1-Wire", http://www.1wire.org/.

8. Gifford K, Kuzminsky S, Williams S, Saiz J, "BioNet: a developer-centric middleware architecture for heterogeneous devices and protocols", IEEE Wireless Communications and Networking Conference, Las Vegas, NV, USA, 3rd-6th April 2006.

9. McGuirk P, "SpaceWire Plug-and-Play", 2006 MAPLD International Conference, Washington, D.C, USA, September 2006.