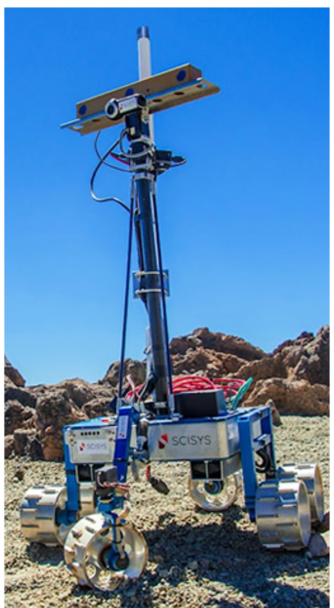
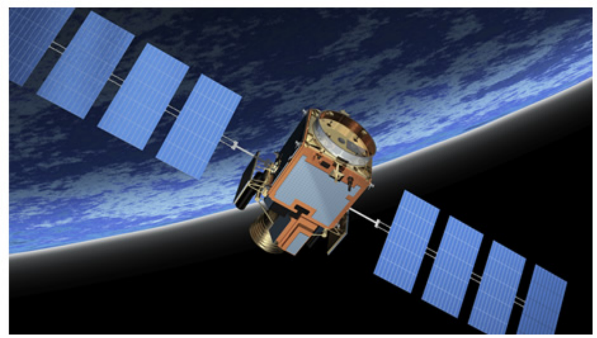
Network Discovery Protocols

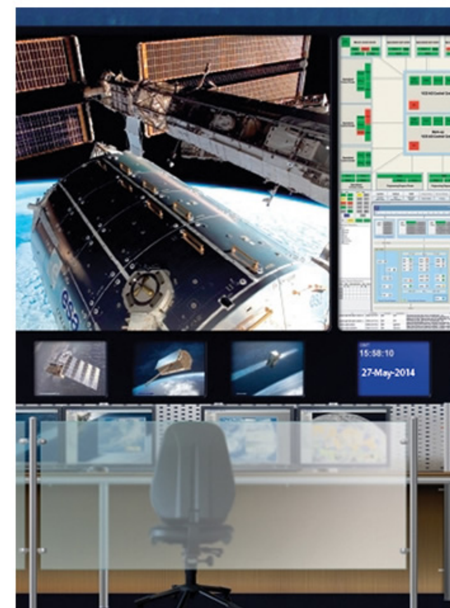
Final Presentation

Stuart Fowell

10th December 2014







Overview

- Consortium and Project Objectives
- User Requirements
- Protocol Design and Specification
- Protocol Implementation
- NDP Demonstrator
- Recommended Future Directions
- Conclusions

Consortium and Project Objectives

Consortium

- SCISYS
  - Prime
  - Requirements Input and Collation
  - Protocol Specification
  - Software Protocol Implementation
  - Demonstrator Development
- STAR-Dundee
  - Requirements Input
  - Hardware Protocol Implementation
- Thales Alenia Space – France
  - Requirements Input

Project Objectives

- SpaceWire standard defines aspects of highly flexible and capable communication system
- However, does not offer a standard mechanism for discovering the topology of a network, or what SpaceWire devices (switches or nodes) are attached to it
- Nor does it offer standard mechanism for configuring various aspects of a SpaceWire network, such as links and routers
- SpaceWire also lacks standard features to assist detection or configuration beyond the network, in the service domain.
- This lack of standardisation for simple tasks that are required on almost all SpaceWire networks limits the level of interoperability which may exist between devices and software, and the extent to which both hardware and software can be re-used between different applications.

- Thus the main objective of the NDP project was to develop, validate and partially demonstrate a protocol to permit SpaceWire network discovery and management configuration in a standard and interoperable manner

  » SpaceWire Network Discovery and Configuration Protocol (SpW-NDCP)

- Depending upon the way this protocol is used, it can offer ‘plug-and-play’ features for SpaceWire networks.

- To achieve this, the goals of the project were:
  » to gather requirements for SpW-NDCP;
  » to design and specify SpW-NDCP;
  » to develop an implementation of SpW-NDCP encompassing hardware and software;
  » to provide a demonstrator to permit demonstration of a limited number of SpW-NCDP features.

User Requirements

User Requirements

- Whilst previous work had been done on the development of a SpaceWire plug-and-play (SpW-PnP) protocol there had never been any formal elicitation of requirements
- 1st stage was to collect and collate requirements on the protocol from 4 sources:
  - Thales Alenia Space – France (TAS-F), taking the role of a system integrator;
  - SCISYS, taking the role of a software supplier
  - STAR-Dundee, taking the role of an equipment supplier
  - ESA requirements from the SoW plus input from the SpaceWire WG representing end customer requirements
- Requirements were then collated
  - Majority of requirements appeared, in some form, in inputs from each stakeholder
    - In these cases a single requirement was produced with terminology differences rationalised
  - In a small number of cases requirements had to be modified beyond their terminology to permit rationalisation. Examples of this include:
    - a requirement that the protocol notify network changes in a TM, as this was deemed to be implementation-specific and needed to generalised
    - a requirement that the protocol to be backwards compatible with all existing routing switch devices, as this was deemed to be overly constraining
- Core collated requirements aligned well with the baseline requirements in SoW including:
  - discovery of network topology
  - unique identification of SpaceWire devices (nodes and switches)
  - configuration of SpaceWire-related features
  - support for features configuration of nodes and routing switches
- Notable discrepancy in the model of network management assumed by the stakeholders
  - Requirements from TAS-F assumed that the protocol would be responsible only for discovery and that all configuration or control of a device would be done via a device driver
  - Remaining stakeholders assumed that a level of support for configuration of standard features (those identifiable from the SpaceWire standard) would be provided

Protocol Design and Specification

Protocol Design (1/2)

- User requirements related to three broad areas of function:
  - network discovery or confirmation in terms of topology (SpW network domain) and device models (vendor domain)
  - device configuration in support of network management (SpW network domain)
  - data sourcing and sinking (service domain)
- Also specified support for certain existing devices (e.g. SpW-10X SpW switching device)
  - If existing device provides all functions necessary to support plug-and-play, as is the case with the SpW-10X, then the limiting factor is not device functionality but communication protocol
  - As these devices cannot be modified to communicate using a new protocol, existing communication methods must be utilised
  - To make this possible, some form of abstraction must be used to provide a common interface to device functions irrespective of communication protocol
- Suggested solution separated into 3 parts:
  - SpW-NDCP responsible only for communication with the device and allows reading from and writing to SpW devices configuration parameters in a standard way
  - a Network Management Service (NMS), which provides the network discovery and configuration service interface to the user and carries out discovery and configuration algorithms
  - These two parts would be connected by device drivers. For devices using SpW-NDCP for all operations then device driver becomes a simple pass-through. For other devices add specific device driver. Also allows additional vendor-specific features not covered by the standard protocol
- Some disparity if SpW-NDCP should support configuration of device features in addition to the ability to identify devices and discover networks
  - For a fixed protocol for configuration, representation of various parameters must be standardised
  - Whilst this is relatively trivial for Boolean parameters, or those with restricted range of options, representation of continuous parameters such as link speed, is difficult to standardise without restricting implementations

Protocol Design (2/2)

- To summarise, the SpW-NDCP design covers:
  - support for discovery on open networks and confirmation of closed networks, both in terms of topology and of device models;
  - support for configuration of SpW management parameters on both open and closed networks;

- Although SpW-NDCP includes generic capability of discovering/confirming and configuring applications and protocols, and therefore managing data sources and sinks, no explicit support for data sources and sinks was designed in the delivered first draft specification of the protocol

SCISYS

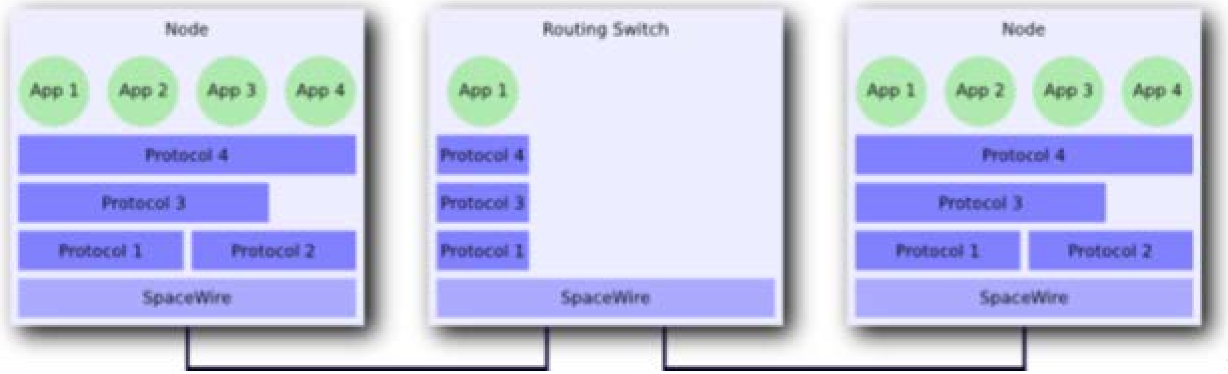
Specification and Standardisation Approach

- Design attempted to ensure that requirements are met with maximum scope for interoperability and forward compatibility
  - Layered architecture is powerful and flexible, but could present challenge for standardisation and for compliance testing of implementations
  - Resulting design was therefore optimised to permit more streamlined, and practical, approach to standardisation.
- Previous attempts at defining a plug-and-play protocol for SpaceWire were relatively constrained and took the perspective that the complexity of a peripheral device should be minimised at the expense of complexity in the control device. This approach did not attempt to:
  - minimise the overall complexity of the solution;
  - minimise the complexity of the subset that must be standardised in order to guarantee interoperability.
- This project, on the other hand, has attempted to take into account both of these factors
  - Crucially, the technical solution has been split into the Network Management Service (including device drivers on the control device) and the communications protocol
  - On the peripheral device, the communications protocol must be implemented plus the Network Management Service which comprises a standard set of fields exposed using the communication protocol.
- By, for example, placing emphasis for controlling ownership onto devices, simple mechanism can be added to peripheral devices whilst relaxing requirements for a standard approach to network discovery.
  - This perspective has been used throughout this design to the point where strong guarantees can be made regarding interoperability through standardisation of:
    - the SpW-PnP protocol;
    - the standard peripheral fields.
- Proposed that standardisation efforts are first concentrated on these two areas
  - Once successful, may be added value in standardising the Network Management Service on the control device
  - Until that time, informative specification of the control device Network Management Service could be made publicly available

Protocol Specification Overview

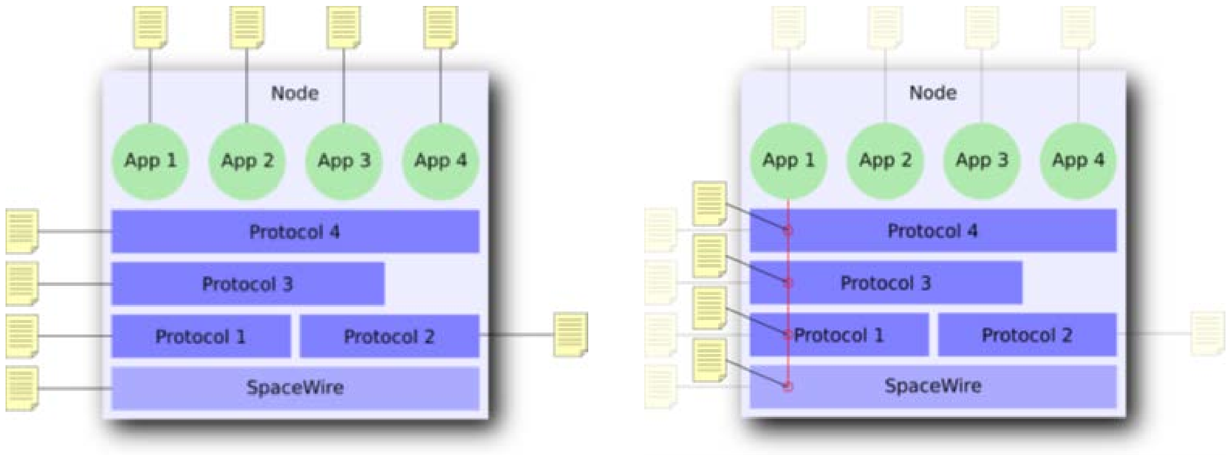
- The following slides provide a brief overview of the SpW-NDCP specification
- A fuller presentation on the SpW-NDCP (formally SpW-PnP) can be found in "Status of the Network Discovery Protocols TRP activity" presentation by Peter Mendham at the 12^{th} ECSS SpaceWire WG meeting date 9^{th}-11^{th} April 2013
- Of course, the draft ECSS specification should be referred to

View of SpaceWire Networks



- Users of the SpaceWire protocols, are referred to as applications
  - » From the perspective of SpW-NDCP, applications are ultimate sources and destinations of SpW packets
  - » In order to communicate over SpW each application uses a set of communication protocols, with the lowest level protocol being SpW itself
- A communications 'channel' is effectively formed, flowing through the communications stack
  - » E.g. App 1 uses Protocol 4 which, in turn and on behalf of App 1, uses Protocol 3. Again on behalf of App 1, Protocol 3 uses Protocol 1, which in turn uses SpW
- Every node and routing switch on a SpW network is referred to by SpW-NDCP as a device
  - » Devices which will be managed by other devices on the network are referred to as peripheral devices
  - » Nodes which will be engaged in managing devices on the network are referred to as control devices
  - » Devices are the functional elements of a SpW network. The physical units of which a SpW network is comprised may each be composed of one or more devices

Plug-and-Play Management Parameters



- Management parameters which define an application's use of a protocol
  - » A set of management parameters may be offered for each use of a protocol by an application
- SpW-NDCP provides simple, generic mechanism to access management parameters across a network
  - » permits management parameters for each supported application, protocol and application
- Core set offered by every peripheral device are those that relate to the device itself & perform a number of important functions:
  - » They permit the following to be identified:
    - › the type of the device (i.e. whether it is a node or routing switch)
    - › the model of the device (i.e. what product the device is)
    - › the physical unit that the device belongs to
    - › the connections a device has, enabling network discovery
    - › the protocols a device supports
    - › the applications a device supports
    - › the protocols that each application uses
  - » Additionally, it is possible to assign an identifier to the device, necessary for network discovery to detect network loops

SCISYS

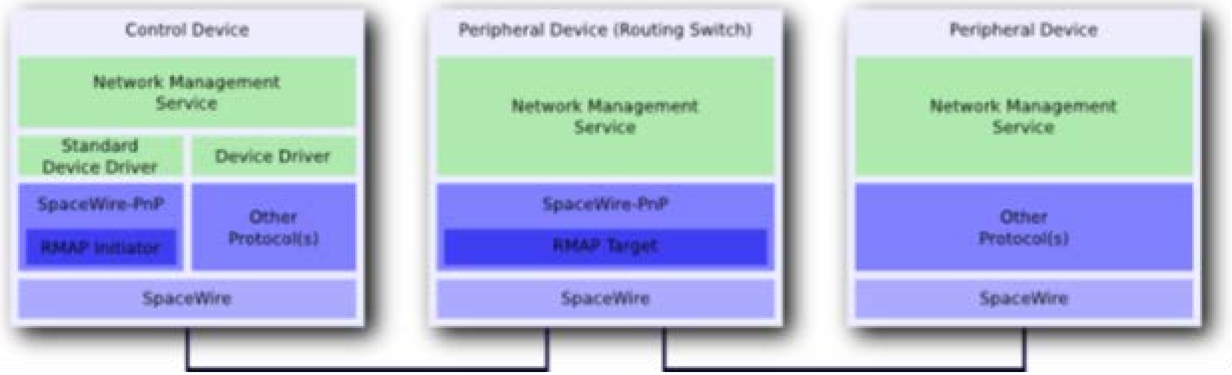
Device, Protocol and Service Identification

- Device models may be uniquely identified through the use of a Vendor ID and a Product ID
  - Vendor IDs uniquely assigned to organisation manufacturing/selling device by the SpW WG
  - Product IDs are assigned by the organisation identified by the Vendor ID
  - Up to organisation to ensure that Product ID uniquely identifies a device model for a given Vendor ID
- Protocols are uniquely identified by SpW-NCDP through the use of a Vendor ID and a Protocol ID
  - Special case of Vendor ID of zero, which indicates a protocol standardised by ECSS
    - These protocols, together with their Protocol IDs, are listed in ECSS-E-ST-50-51
    - Protocol IDs specified by ECSS-E-ST-50-51 are all non-zero
    - Additional special case of a Vendor ID of zero and a Protocol ID of zero permits SpW-NCDP to identify management parameters associated with SpaceWire itself
  - Non-zero values of Vendor ID identify the organisation responsible for creating or standardising the protocol
    - Protocol IDs are assigned by the organisation in an identical manner to Product IDs
    - This mechanism permits vendor-specific management parameters relating to one or more protocols to be exposed using SpW-NDCP parameter access mechanisms
- Application identification follows an identical pattern
  - Each supported application is identified using a Vendor ID and Application ID combination
  - Vendor ID of zero identifies an application standardised by ECSS
  - Non-zero Vendor ID identifies an application created or standardised by another organisation

Restrictive Assumptions

- SpW-NDCP makes two assumptions which restrict the design of SpaceWire devices which intend to be compliant with the standard:
  - If a device has multiple links, SpW-NDCP requires that access to protocol and applications supported by a device must be equivalent irrespective of the link used to communicate with the device
    - This means that the protocol and application support list do not depend on the link used to access the device
  - Upper limit on the number of links a device has
    - SpaceWire standard restricts the number of path-addressable links on a routing switch to be 31
    - As the number of fields available to configure a device is finite, SpW-NDCP must assume an upper limit on the number of links a device has
    - For consistency, SpW-NDCP limits the number of links a device may have to 31
    - This ensures that every link in a SpW-NDCP network has an equivalent path address

Network Management Reference Architecture



- NMS on control device does network discovery, device id & mgmt activities using comms protocol
- Peripheral device permits itself to be managed by mgmt parameters accessed using comms protocol
- NMS on control device does not access comms protocol directly. Protocol access abstracted by use of device driver, permits NMS to manage devices not supporting SpW-NDCP or std peripheral device NMS
- Reference architecture possible providing that device driver used with each device is already known
  - » To permit use of device without a priori knowledge of device model must detect device & type in standard way
  - » By providing a standard communications mechanism and a standard core set of management parameters, SpW-NCPP can be used for network discovery and device identification
  - » Using info provided by SpW-NDCP, possible to associate device driver with device to access other device features
- Placing all functionality strictly necessary for network discovery & device identification in peripheral device means basic level of interoperability is assured without standardising control device network management service or device driver interface
  - » Permits devices to be manufactured supporting plug-and-play without enforcing unnecessary standardisation
  - » Informative annex of standard illustrates how SpW-NDCP may be used by a control device to discover SpW network

Device Access and Ownership

- It is possible that a network may have multiple control devices, and that these control devices may not be coordinated
  - Where this is the case, care must be taken not to corrupt the configuration of peripheral devices
- To support operation in such a scenario, SpW-NDCP regulates the ability to set the value of management parameters on a device
  - To be able to configure a peripheral device, a control device must first assign the device an identifier
  - As part of this operation, the peripheral device records the SpaceWire address which the control device specified for the reply information
- The peripheral device will then only permit configuration operations which also specify this address for reply information
  - This enforces the concept of ownership, where a single control device is responsible for managing each peripheral device.
- Where there are multiple control devices, there is the potential for contention over device ownership
  - Mechanisms for the resolution of this contention are not specified by the standard.

Mapping from CCSDS SOIS Subnetwork Layer

- NMS reference architecture is designed to be compliant with the subnetwork service interfaces described by the CCSDS Spacecraft Onboard Interface Services (SOIS)
  - Especially the SOIS Device Discovery Service (DDS)
- As such, SpW-NDCP is designed to support CCSDS SOIS
- However, as the control device NMS is not covered by the standard, compliance with SpW-NDCP does not ensure compliance with the SOIS DDS

Protocol Implementation

Hardware Peripheral Device Implementation (1/2)

- STAR-Dundee updated two existing products, the SpaceWire-USB Brick Mk2 and SpaceWire Router Mk2S [RD.2], to support SpW-NDCP. The two devices are quite similar, with the following key characteristics:
  - Can operate in interface (Brick Mk2 by default) or router mode (Router Mk2S by default)
  - A configuration port which is accessed using RMAP
  - Ability to act as a time-code master
  - Support for error injection
  - Accessed using STAR-Dundee's common software stack, STAR-System

- Both devices provide configuration port using RMAP, reuse much functionality to support SpW-NDCP
  - Translation layer handles differences between addresses and formats used by SpW-NDCP and STAR-Dundee RMAP configuration method

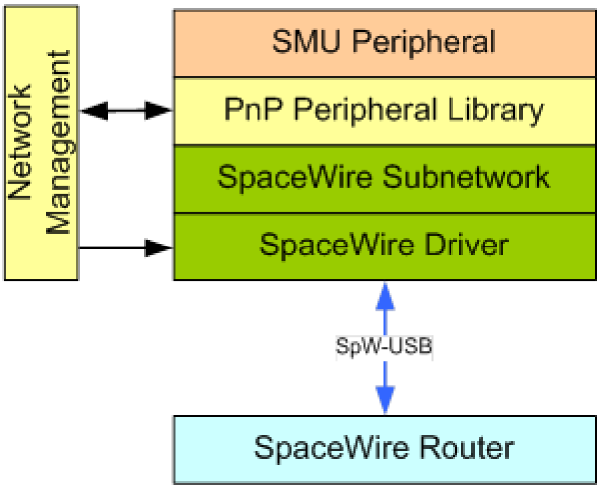
- Most complex areas to implement in this translation layer were reference transmit rate and reference watchdog rate fields
  - In standard Brick Mk2 and Router Mk2S, both these values selected from a set of values (120, 130, 140, 150, 160, 180 or 200 Mbps)
  - For SpW-NDCP, base transmit rate was set to 65520 and the range was set from 328 to 546
  - However only a small number of values are accepted within this range: 328, 364, 410, 437, 468, 504 and 546
  - This isn't an ideal solution, but is the best that can be achieved within the protocol specification
- Configuration on a link-by-link basis
  - Standard Brick Mk2 and Router Mk2S also do not support start-on-request and watchdog timeouts for individual links, only for all links
  - Configuration on a link-by-link basis added to the SpW-NDCP versions of the devices

Hardware Peripheral Device Implementation (2/2)

- One other major change required was for devices to route packets to configuration port even when in interface mode
  - Standard devices pass all packets received on SpaceWire links to user applications when in interface mode
  - This means they cannot be configured from the SpaceWire ports when in interface mode
  - Devices were modified to route all packets with a first byte of 0 to the configuration port
  - Means device can be configured over SpaceWire when in interface mode, but also means any data packets which begin with a first byte of 0 will also be routed to the configuration port
    - This was a mistake often made in testing where packets containing only 0s were being sent

- Additional logic to implement SpW-NDCP:
  - Brick Mk2's FPGA: number of flip flops increased 23%, and number of lookup tables (LUTs) increased 35%
  - Router Mk2S: number of flip flops increased 9%, and number of lookup tables (LUTs) increased 15%
- To test functionality of hardware, STAR-Dundee also developed a software implementation of a SpW-NDCP control device
  - Used to test the hardware's response to valid and invalid read, write and compare-and-swap operations.
  - Makes use of the existing RMAP initiator included in STAR-System, with only minor changes required to support the SpW-NDCP protocol's protocol identifier
  - 3 functions for performing read, write and compare-and-swap operations were added, which simply translate the SpW-NDCP fields (application index, protocol index, field set identifier, field identifier) in to the correct RMAP field properties

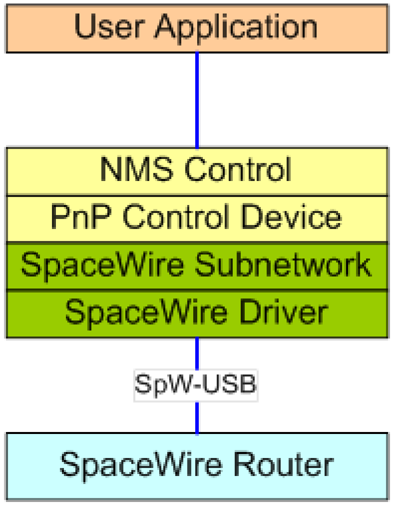
Software Peripheral Device Implementation



- Software implementation of NMS and SpW-PnP protocol functionality for a peripheral device
  - To determine if this is achievable and for demonstration purposes
  - Built upon the SpaceWire Subnetwork layer of the SOIS Software Suite, and a platform-specific basic SpaceWire driver.
- Enables a SpW-PnP control device to request to modify any supported component of the hardware or software
  - For example it is possible for a control device to request that a peripheral device modify its SpaceWire link rate divider
- For the purposes of project, peripheral device only needed to support certain functionality that can be configured; there are no software components that can be configured, only hardware components
  - Hence NMS performing configuration requests on the SpaceWire driver, such as modify link rate divider
- SpW-PnP peripheral device software implementation is portable between Linux PC and RTEMS/RASTA system
- A separate single threaded library of the SpW-PnP peripheral device software implementation, not reliant upon any RTOS, was created to run on the STAR-Dundee SpW-RTC unit

SCISYS

Software Control Device Implementation



- Software implementation of NMS and SpW-PnP protocol functionality for a control device
  - » To determine if this is achievable and for demonstration purposes
  - » Built upon the SpaceWire Subnetwork layer of the SOIS Software Suite, and a platform-specific basic SpaceWire driver
- SpW-PnP Control library initiates SpW-PnP transactions on behalf of the NMS control functionality, returning a received response or a timeout
- NMS control device functionality provides the ability to discover an open or confirm a closed SpaceWire network, and to control and configure it
  - » Provides an API to user applications
- Discovering or confirming a network will result in a probing of the network to discover what devices are attached
- NMS control device implements a depth first tree spanning network discovery algorithm, as described as an example in the NMS recommendation
  - » Once network discovery has completed a network topology is returned to the user application
  - » Note: NMS control device functionality assumes it is the only SpW-PnP control device in the SpaceWire network
- SpW-PnP peripheral device software implementation is portable between Linux PC and RTEMS/RASTA system

SCISYS

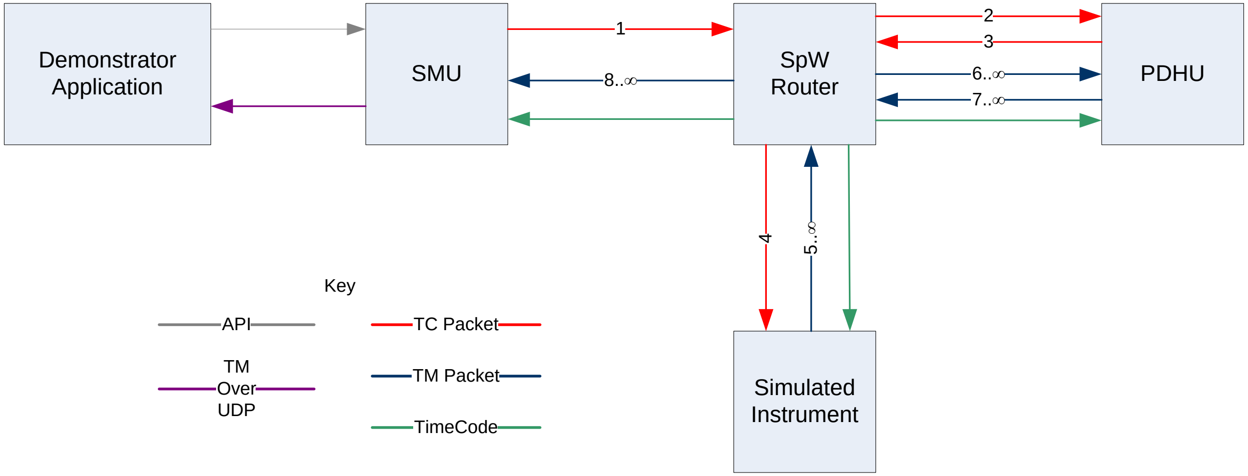
Feedback upon the Specification

- Hardware and software implementation of the draft specification generated some feedback upon the protocol specification and comments upon implementation issues
  - » Details in the Final Report
- In general, the protocol specification is quite mature, and provides everything required to implement the protocol in both hardware and software
  - » Most of the comments can be addressed through very minor changes to the document
- The protocol itself provides all the functionality required to implement network discovery and other network management tasks
- Implementation in existing hardware is relatively simple, with only a few areas which do not map cleanly

NDP Demonstrator

SCISYS

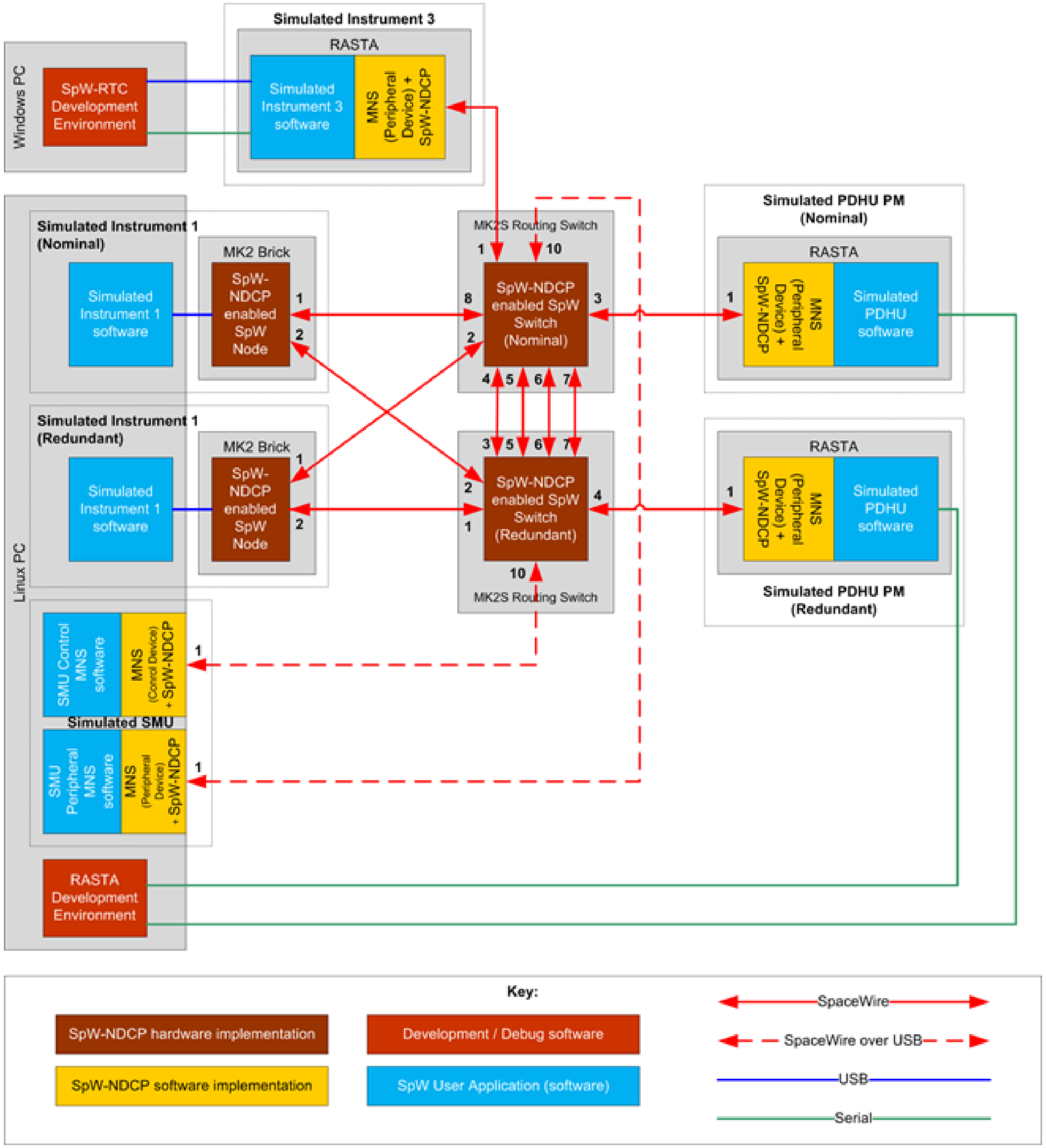
Demonstrator Mission



- To demonstrate the use of SpW-NDCP, a demonstrator mission was defined:
  - Spacecraft Management Unit (SMU);
  - Redunded Payload Data Handling Unit (PDHU), with the redundant PDHU in warm standby;
  - Redunded Instrument 1, with the redundant Instrument 1 in cold standby;
  - Non-redunded Instrument 3, that is to say it has no redundancy;
  - Redunded SpaceWire Network, with the redundant SpaceWire plane in hot standby, with cross strapping between two planes
- Upon start-up, SMU sends TC to PDHU to command it to configure active instruments & forward received TM back to SMU
- PDHU sends a TC to the simulated instruments to command them to start generating TMs back to the PDHU
  - All telemetry is time-stamped in CUC format using the SCET.
- SpaceWire logical addressing is used for each component
  - Because redundant Instrument 1 is in cold standby, same SpW logical address is used for both nominal & redundant Instrument 1; fail-over is achieved by reconfiguring SpW network to route SpW packets to the redundant Instrument 1
  - SCET is synchronised across the SpW Network using SpW time-codes
  - TCs and TM packets are carried across the SpW network using CPTP
- Communication with ground is not included, though TM received by SMU is forward to Demonstrator Application for display
- SpW-NDCP traffic uses SpaceWire path addressing, whilst all other traffic uses SpaceWire logical addressing.

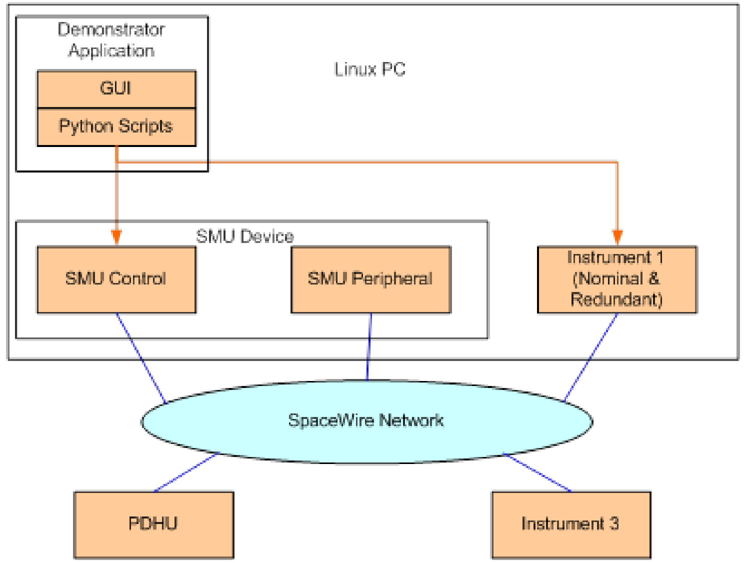
SCISYS

Hardware Overview



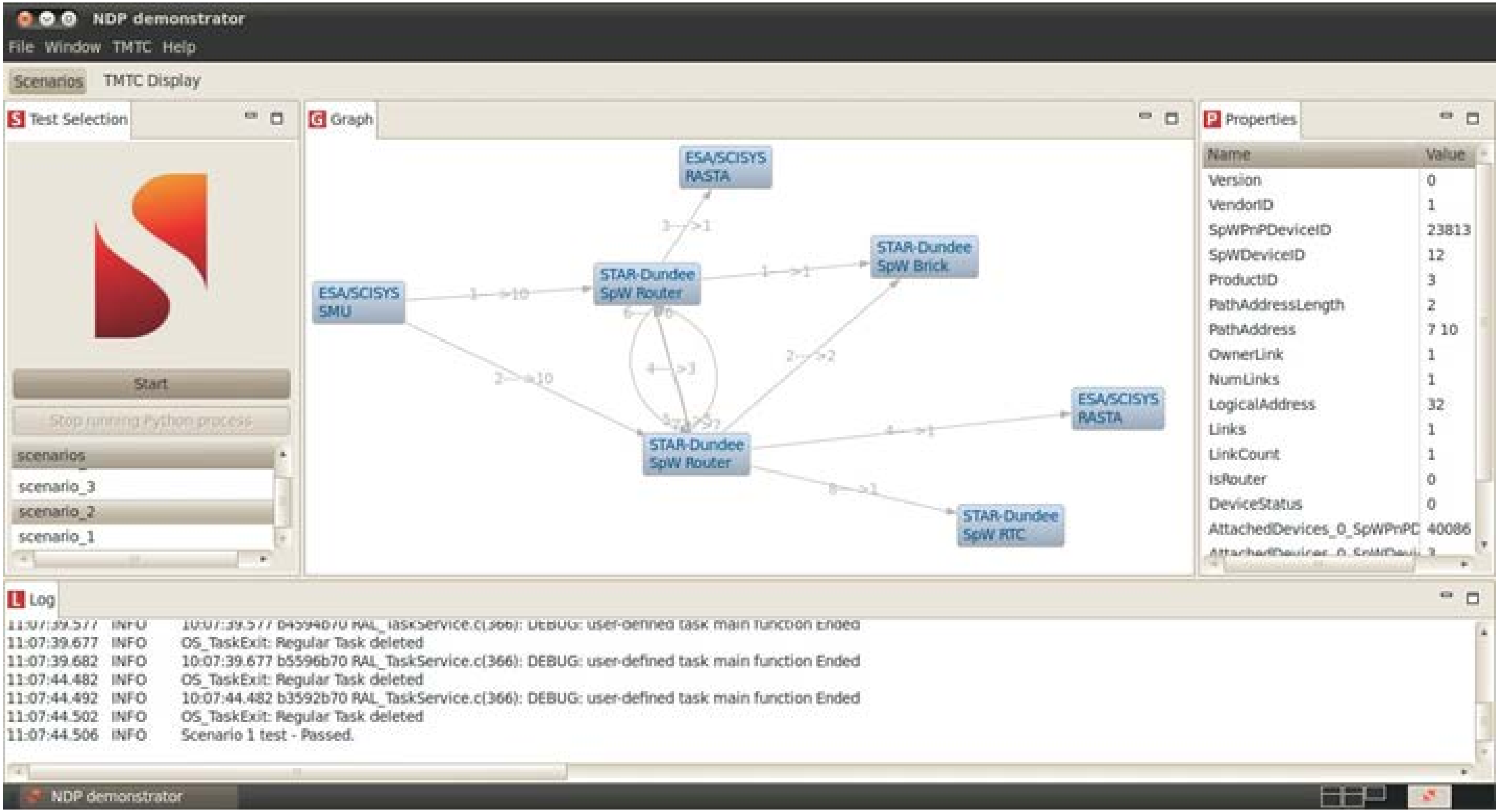
SCISYS

Software Overview



- GUI: Java Executable running on Linux PC. Launches demonstrator, used to control the execution of demonstrator scenarios. Displays discovered network and incoming TM packets
  - » Part of demonstrator, rather than the SpW-NDCP software implementation
- Scenario Scripts: Python Scripts running on Linux PC. Defines a given demonstrator scenario, listing operations to perform
  - » Part of demonstrator, rather than the SpW-NDCP software implementation
- SMU Control: C Shared Library running on Linux PC. Contains the NMS and SpW-NDCP control functionality such as network discovery, claim device ownership, configure device etc
  - » Includes interface between Scenario Scripts and the SpW-NDCP functionality
- SMU Peripheral: C Executable running on Linux PC. Single executable containing the NMS and SpW-NDCP peripheral functionality that enables SMU to discover itself via an alternative router
- PDHU: C Executable running on RASTA. Single executable containing TM/TC data handling between simulated instruments and GUI, and the NMS and SpW-NDCP peripheral functionality that enables SMU to discover it
- Instrument 1 (nominal and redundant): C Shared Library running on Linux PC. Simulates instruments that interface to the SpaceWire bricks. Generates TM packets once requested
- Instrument 3: C Executable running on SpW-RTC. Single executable containing the NMS and SpW-NDCP peripheral functionality that enables SMU to discover it

Software Overview



SpaceWire Network Topology XML Schema

- XML schema defined to describe discovered SpaceWire network topology and information on the discovered devices
  - » XML Schema Definition (XSD) file
- In NDP Demonstrator, each network discovery request generates an XML document, held in a single XML file, which describes devices on the network
- XML document contains devices' properties as well as connection configuration
- Whenever new network topology is discovered, the XML file is validated against the XSD file, parsed into the GUI application, and its content displayed on 2D graph
- NOTE:
  - » Not all information fields are supported by the hardware and software implementations of SpW-PnP peripheral devices.

S1: Confirm, Configure and Use SpaceWire Network

- Demonstrates that, using SpW-NDCP, the closed SpaceWire network can be confirmed, that each device in the network can be configured, and that normal SpaceWire traffic can then operate over the network
- When SpW-PnP peripheral devices are added to a SpaceWire network, they are discovered and ownership of them claimed by the SpW-NDCP control device, i.e. the SMU.
  - In turn each of the SpW-NDCP peripheral devices (SpaceWire routers, PDHU, and instruments) is added to the network by connecting them to a SpaceWire router
  - Between each device being added, the scenario script performs network discovery to discover and claim ownership of the new device, and updated SpaceWire network topology displayed in the GUI
- Once discovery is complete, the SpaceWire network is configured using SpW-NDCP to:
  - configure routing tables of the routers
  - configure individual SpaceWire link speeds
  - set a SpaceWire-time-code master.
- NOTE:
  - Although not compliant to the SpaceWire standard, the SpaceWire routers support SpaceWire time-code master functionality. The nominal SpaceWire router is set to be the SpaceWire time-code master in the network.
- Once configuration is complete, Demonstrator Mission is started so that normal SpaceWire network traffic begins

S2: Time-Code Management

- Demonstrates that SpW-NDCP can be used for SpaceWire network management and that normal SpaceWire network traffic is not affected by this
- After initialising the network as at the end of Scenario 1
  - » the nominal SpaceWire Router generating SpaceWire time-codes and the instruments (Instrument 1 (nominal), Instrument 3) generating TM
- SMU uses NMS and SpW-NDCP to command the SpaceWire time-code master on the nominal SpaceWire router to stop and later restart generating SpaceWire time-codes
  - » The user is able to determine that SpaceWire time-codes have stopped being generated by inspecting the displayed TM packets where their time stamps will not be increasing
  - » The regular reception of TM packets also demonstrates that SpW-NDCP does not affect normal traffic across the SpaceWire network

S3: Failover to Cold Standby

- Demonstrates that SpW-NDCP can be used
  - to detect nodes lost from the SpaceWire network
  - for SpaceWire network management to achieve a fail-over from a nominal to redundant Instrument 1 through the reconfiguration of SpaceWire Router routing tables with minimal disruption to the normal SpaceWire network traffic.
- Failover is achieved through changing the routing of packets between the paths to the nominal and redundant Instrument 1
  - As TCs and TM packets between SMU, PDHU and Instruments use logical addressing
- After initialising the network as at end of Scenario 1 with the Instrument 1 (nominal) active
- Demonstrator App triggers a simulated failure in Instrument 1 (nominal) so that it disappears from SpaceWire network
- When network discovery algorithm is run again it shows that Instrument 1 (nominal) has been lost from SpaceWire network
  - updated SpaceWire network topology is displayed in GUI
- Demonstrator App starts Instrument 1 (Redundant) and uses SpW-NDCP to reconfigure the routing tables in the SpaceWire Routers so that SpaceWire traffic is routes to Instrument 1 (Redundant) instead of Instrument 1 (Nominal)
- Demonstrator App initiates a network discovery that discovers Instrument 1 (Redundant)
  - updated SpaceWire network topology is displayed in GUI
- Finally the SMU sends a TC to PDHU to in turn command Instrument 1 (Redundant) to start generating TM

S4: Open Network Discovery

- Demonstrates that SpW-NDCP can be used to discovery an open SpaceWire network
  - (and not confirmed and configured)
- SpaceWire network topology is changed from that used in Scenario 1
- Network discovery algorithm re-ran with the changes reflected in the topology graph in the GUI display
- NOTE:
  - SMU must be connected to the nominal SpaceWire router, and this scenario only makes use of the two SpaceWire Routers and two SpaceWire Bricks

Deliverables

- Publically available deliverables:
  - Draft ECSS SpW-PnP Specification
  - SpaceWire Network Topology XML Schema Specification and XSD file
  - Final Report, Executive Summary Report, Abstract and mini Web Site
  - Final Presentation

- NDP Demonstrator installed in ESTEC Payload Test Lab

Demonstration Results

- All scenarios have been successfully demonstrated using the NDP Demonstrator installed at the ESTEC Payload Test Lab
- NDP Demonstrator does not demonstrate all possibilities for SpW-PnP and SpaceWire network management:
  - Implements one possible network discovery algorithm, others exist.
  - Implements one possible network management policy, others exist. The policy is as follows:
    - A fixed configuration with pre-allocated logical addresses and routing is used. Alternatively a dynamic logical address allocation could be used.
    - Failover between nominal and redundant nodes is handled by associating the logical address with the function and changing the routing between the nominal and redundant nodes
- Some small limitations in and proposed improvements on the NDP Demonstrator have been identified
  - Details in the Final Report

Recommended Future Directions

Recommended Future Directions

- The consortium recommends that
  - the SpW-NDCP peripheral device functionality be implemented in each SpaceWire node and routing switch in hardware, to provide an efficient response time for network discovery and configuration operations
  - SpW-PnP control device functionality, including the Network Management Service are best implemented in software, to allow for the variability of policies used per deployment.
- The consortium recommends that suggested updates are applied to draft ECSS specification
- Outputs of parallel ESA activity on SpaceWire FDIR should be considered to create single, harmonised, network management protocol
- Further effort is then required to determine protocol's applicability to proposed SpaceFibre standard
  - Are there any mandatory components of SpW-NDCP which are not necessary/applicable to SpaceFibre
  - It may be necessary to alter the specification to ensure it does not exclude its use with SpaceFibre.
- A study be undertaken into relationship between SpW-NDCP and AIAA standard SPA-S
  - Considering how the two protocols can be used together and if any changes are required to accommodate this, if there are any features of SPA-S that should also be included in SpaceWire-NDCP, etc.
- Updated draft ECSS specification presented to ECSS SpaceWire WG, leading to the start of an ECSS standardisation process
- To support the publication and use of an ECSS network management specification, it is suggested that an ECSS handbook should also be prepared. The handbook should capture guidance and use cases, as well as an informative specification for the Network Management Service, as described in this activity.
- Finally, the NDP Demonstrator could be extended to include
  - Support for RASTA redundant SpaceWire ports
  - RASTA as the time-code master
  - and the second SpW-RTC acting as Instrument 4
- Stand-alone variant of the GUI could be developed to visualise network topologies specified in XML

Conclusions

Conclusions

- The Network Discovery Protocols project has successfully:
  - Collated requirements on SpaceWire plug-and-play taking agency, system integrator, and hardware and software implementer view-points into account
  - Designed a draft SpW-NDCP specification and proposed a Network Management Service
  - Implemented in hardware and software SpW-NDCP Peripheral Device functionality.
  - Implemented in software SpW-NDCP Control Device functionality.
  - Implemented a NDP Demonstrator as a proof of concept of the use of SpW-NDCP to
    - confirm, configure and use a closed SpaceWire network
    - to discover an open SpaceWire network
  - Made recommendations on the standardisation by ECSS of the SpW-NDCP specification, and the production on an ECSS network management handbook.

Thank you

Any questions?