DERIVING PRELIMINARY INTERFACES USING SYSTEM REQUIREMENTS

Electron Ion Collider

OVERVIEW

INTERFACE CONTROL DOCUMENTS

Interface control documents are a key element in systems engineering. In addition to identifying interfaces, these documents also provide a catalog of all supporting information such as drawings, diagrams, tables and textual information. These underlying *interface documents* provide the details necessary to fully design and implement the interfaces between two entities, whether they be components, sub-systems or systems. This document will outline the process that we will use to gather and track interfaces, how they follow naturally from the systems requirements, and give guidance on how we categorize various interfaces.

REQUIREMENTS, DEPENDENCIES AND INTERFACES

The process of requirement definition is typically one of the first steps in system design. Requirements describe the purpose of each sub-system, what functions it must perform, and how well it must perform each function. Once the preliminary requirements are developed, the next step is determining how each of these requirements are related to one another (their dependencies) and how these relationships will be realized (their interfaces). There is a clear relationship that exists between requirements, dependencies and interfaces. This relationship can be defined with the following general rules.

a. Requirements identify dependencies – *a need that must be satisfied by an outside entity or resource.*
b. Dependencies require interfaces – *a defined connection between the requiring and providing entities.*
c. Interfaces are resolved by requirements – *a commitment by the providing entity to supply the needed resource in a way that satisfies the stated need.*

While there are exceptions, it is generally true that all requirements either create or resolve dependencies within the system design. Accordingly, the majority of system interfaces should be discoverable by a direct examination of the requirements, how they depend on one another, and how the dependency should be resolved.

INTERFACES DEFINED

An interface is a defined relationship between two otherwise distinct entities that exists in order to satisfy a dependency or requirement. Interfaces may be between a human and a machine, between two components, or between a system and its external environment. In each case, though, an interface defines an essential relationship between the connected entities.

According to Pahl & Beitz (1984) and Wood, Stone, McAdams, Hitrz, & Szykman (2002), there are four categories of interfaces: *physical, material, energy, and information*. Examples of a physical interface would be flanges that connect two components, vibration dampening fixtures, or support structures used to position sub-systems. In each of these examples, the interface provides a physical connection between components with no dynamic interchanges. Material, energy and information interfaces accommodate the flow of resources between entities. Material interfaces support the flow of physical matter between components, while energy interfaces support the delivery of power. Finally, information interfaces provide a conduit for communications and control.

Interfaces are largely defined by their endpoints and the medium that connects them. While there are examples of complex distribution systems where three or more sub-systems share the same interface, it is often preferable to reduce interface definitions to represent connections between **just two entities**. This allows the system designer to clearly identify the characteristics of the interface on each end, specify the connecting medium, and verify that both sides of the connection will be satisfied.

At the simplest level, interfaces are used to satisfy dependencies. In *Figure 1* the dependency between the two entities is illustrated by the dashed line. The gold connector between them represents the interface that resolves that dependency.

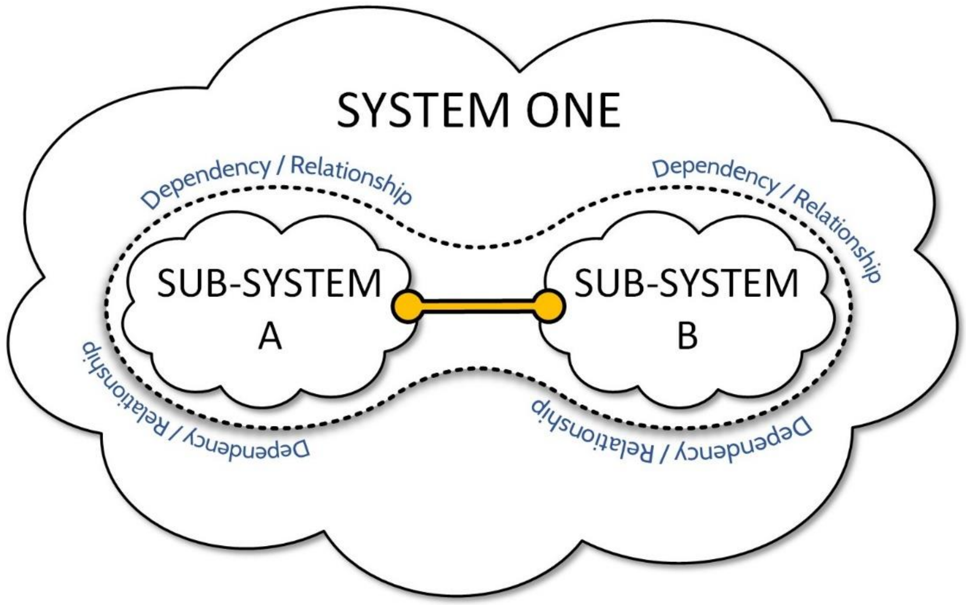

Figure 1: Illustration 2 sub-systems with a single interface

INTERFACES AND SYSTEM BOUNDARIES

Figure 2 shows two categories of system interfaces; inward facing and outward facing. The inward facing interfaces which connect the sub-systems within System One, and the outward facing interfaces which connect System One and System Two. These types of interfaces are tightly coupled with the definition of local and non-local dependencies, where a local dependency exists within a system or component, and a non-local dependency exists between a system and an external entity.

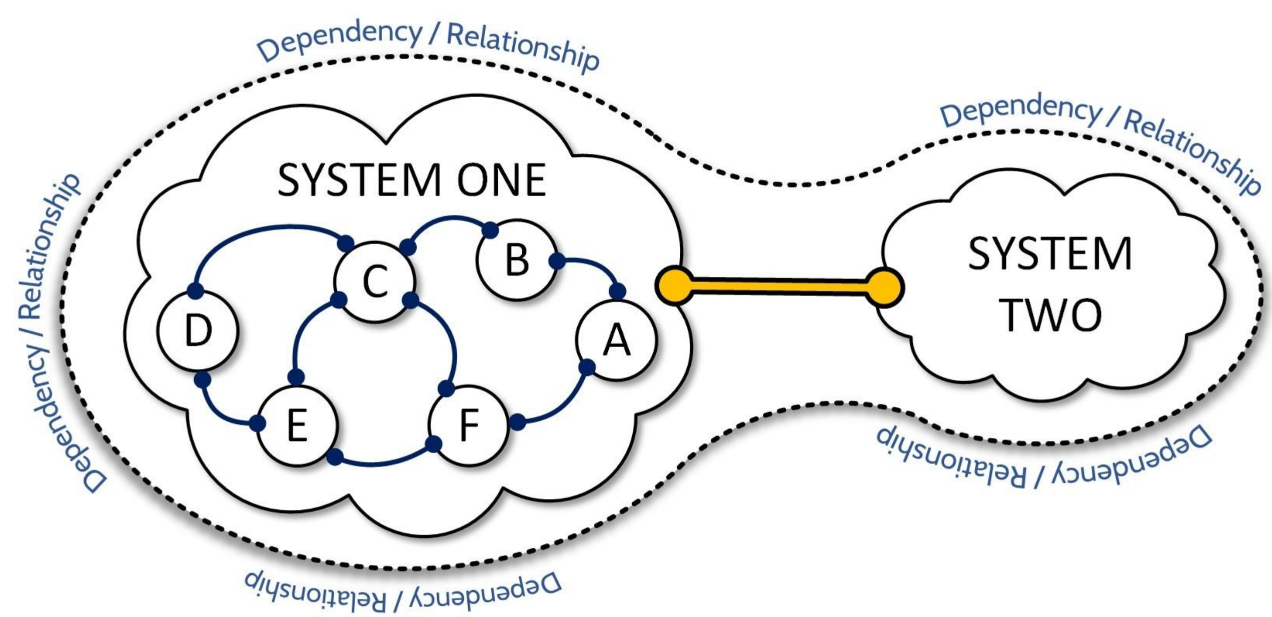

Figure 2: Illustration of inward facing and outward facing interfaces

- **Inward Facing Interfaces**

An inward facing interface is one that connects two or more entities within the same sub-system. Because they have no connections outside of their local environment, the sub-system designer has total latitude in specifying, designing and configuring these interfaces. **The local design is only constrained by the requirement that the completed sub-system satisfies its functional and performance requirements.** This allows the sub-system to be treated as a *black-box* where external entities require no knowledge about the internal configuration of the component.

- **Outward Facing Interfaces**

By contrast, an outward facing interface is one that connects the sub-system (or a component within the sub-system) to another sub-system or to a service in the operating environment. The outward facing interface must have a published specification that is sufficient for the connection to be designed, configured and function properly. Any changes to the outward facing interface, or the *foreign* interface to which it connects, must be coordinated and documented.

This implies that non-local/outward facing interfaces should be designed with as little knowledge as possible of the peer sub-system's internal structure. For example, in *Figure 3* a dependency exists between System Two and System One. Within System One, there are three distinct sub-systems that have interfaces which can resolve the dependency. However, rather than allowing System Two to specify which of these interfaces it will use, it only states that its dependency will be resolved by System One. System One remains responsible for determining which resource to use, based on current conditions.

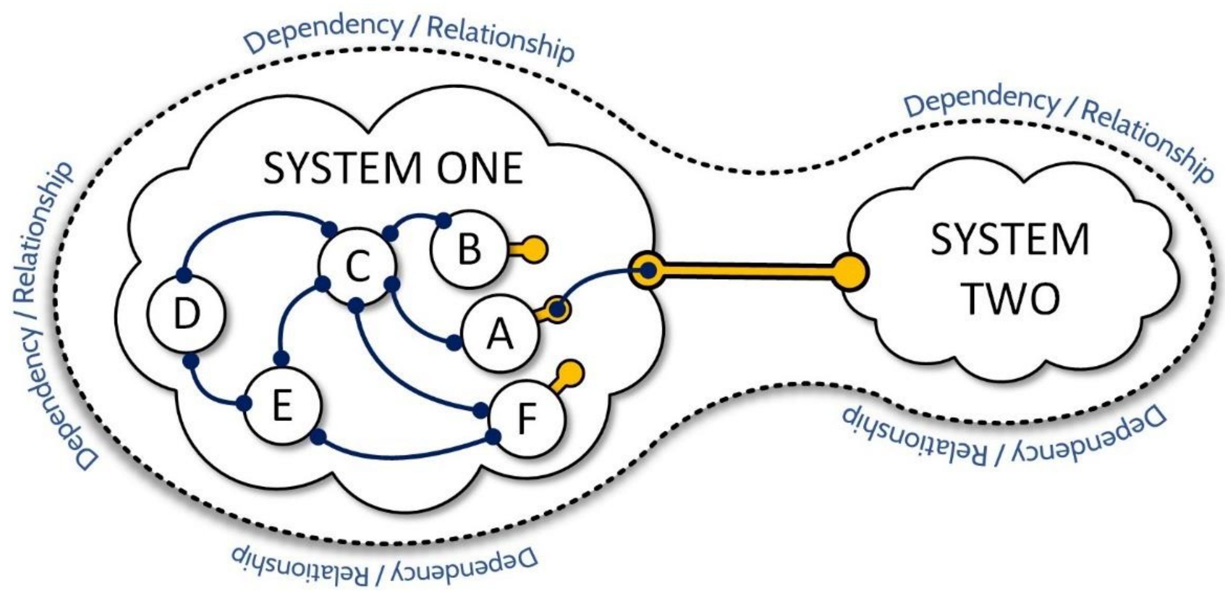

Figure 3: Resolution of Non-Local Interfaces at the Sub-System Boundary

DERIVING INTERFACES FROM SYSTEM REQUIREMENTS

SYSTEM REQUIREMENTS

Each sub-system is defined by a collection of requirements that specifically describe its required characteristics. For the electron ion collider system, we have broken our requirements into 3 types: general, functional and performance. General requirements state what the systems is, functional requirements state what the system does, and performance requirements state how well the system must do it.

Some requirements may have been developed with respect to the 'whole system' while others may apply to specific sub-systems or components. Because of this, requirements are developed using a hierarchy that begins at the highest level system and then extends downward to each of the sub-systems and components below. In this hierarchy, the requirements at the top-level are typically very general, and become increasingly specific as the hierarchy is traversed.

Intrinsic in the requirements are individual dependencies that establish a relationship between one system (or component) and another. Therefore, an initial collection of interface requirements can be derived by examining each of the existing requirements and asking questions about how it relates to other entities.

IDENTIFYING THE INTERFACES

A three step process can be used to create a preliminary list of interfaces within a system. As described earlier, this process begins by interrogating the requirements, then interrogating the known interfaces, and finally interrogating the related systems.

▪ Interrogating Requirements

Using the requirements that have been developed for each sub-system, a series of questions can be asked that will identify the relationships or dependencies that it has on external components. For each of these relationships, additional questions are asked that identify the nature of the relationship and the type of interface that will be required to accommodate it. Keep in mind that the purpose of this step is not to fully design all interfaces, but merely to create a table of known interfaces that must be verified, designed and validated during system implementation.

▪ Interrogating Known Interfaces

The inverse approach to requirement interrogation is to interrogate the known interfaces. This is essentially a second pass that verifies that the known interfaces and known requirements are internally consistent. To perform this step, an individual with expertise in the system (or component) should develop a list of expected interfaces that should exist between their system and external entities. If the requirements are complete and the prior step was successful, each of these interfaces should already be represented in the derived list of interfaces and should relate back to an existing requirement. If an interface does not exist, then the reviewer should examine the new interface and determine if the requirements need to be revised.

The product of the first two steps should be a ***preliminary interface document*** that lists all identified interfaces, the requirements from which they were derived, the internal components that require them, and the external systems to which they will connect.

▪ Inter-System Interrogation

Once a preliminary list of interfaces has been identified for each of the sub-systems, the next step is to ensure that there are corresponding interfaces in each of the related systems. To confirm this, the reviewer should iterate through each of the defined interfaces within each system and verify that it is satisfied in the related system. This verification process should perform several key steps:

1. Confirm the existence of the interface

   Each interface should have a corresponding interface in the related system. If one does not exist, it should be created.

2. Verify that the characteristics are consistent

   The interface requirements between two systems should identify the same resource or capabilities, and should use the same terms.

3. Verify that the scale is consistent

   While not strictly defined yet, the expected volume of materials, energy or data that will be conducted through the interface should be consistent on each end.

TYPICAL QUESTIONS FOR INTERROGATING REQUIREMENTS

In most cases, there is a common set of interface types that will be typical for all systems. These interfaces define their structural demands, resource requirements, environmental conditions, waste elimination, and so forth. The list of questions here is not designed to be all inclusive, but provides an example of the types of questions that should be asked in order to identify interfaces. This collection should be altered as necessary to support the unique characteristics of a system or component.

Direct interfaces generally describe relationships between two entities that exchange material, energy, or data. Indirect interfaces describe constraints and limitations that are created by external entities.

STRUCTURAL	*Direct*	Does the entity provide support or stability to another component?	*Used to identify any intermediate structural supports, interfaces that provide stability (balance or dampen vibrations), or any internal/externally supported load that transfers weight to the system.*
		Is the entity supported or stabilized by another component?	*This interface defines how the weight of this system (and supported systems) is transferred, and mechanisms that provide stability.*
	Indirect	Is the weight of the item constrained by an external factor?	*This indirect interface describes limitations that are driven by material handling capabilities, crane capacity, floor strength, etc.*
SPATIAL	*Direct*	Does the entity have connectors, endpoints or junctions that must exist at a specific point?	*Used to identify connectivity requirements with other components and ensure that all external connections meet at the correct place.*
		Does the entity have an adjacency requirement with another component?	*Used to identify conditions where this entity's distance (near or far) from another component is required to ensure functionality.*
	Indirect	Are the dimensions of the entity driven by another component's characteristics?	*This interface identifies relationships where a change in an external component will result in a change to this entity's dimensions, spatial characteristics, or other factors.*
		Do the dimensions of this entity drive changes in another component's characteristics?	*This interface identifies relationships where changes to this component's size will result in changes to another entity's dimensions, spatial characteristics, or other factors.*
		Does the entity consume space within, or form the boundary of, a constrained space?	*This interface identifies conditions where a component consumes space that is shared by other components, or is used as a service plenum.*
		Are the dimensions of the entity constrained by external factors?	*These are constraints on size and shape that are driven by the requirement to fit within another object, fit through a doorway or access pathway, or provide space for services.*

ENVIRONMENTAL	*Direct*	Does the entity rely on another component or system to maintain its temperature, humidity or other environmental characteristics?	*Used to identify the environmental control mechanisms for this entity to include type, capacity, connection characteristics and methods of delivery and rejection.*
		Does the entity provide resources or capabilities to maintain the temperature, humidity, or environmental characteristics of another component?	*Identifies how environmental services are delivered from this entity to other supported components to include individual characteristics as well as cumulative volumes.*
	Indirect	Does the entity reject waste energy (heat, radiation, magnetic field, etc.) that must be limited or mitigated because of other components?	*This identifies all of the forms of waste energy that must be rejected from this entity, where it will go, and how it will be transferred.*
		Is the entity dependent on environmental conditions (heat, radiation, magnetic field, etc.) from other components being limited or mitigated?	*This interface describes the limitations that are applied to other, adjacent sub-systems (in terms of energy rejection) in order for this entity to function properly.*
LCW/PROCESS COOLING	*Direct*	Does the entity use low conductivity water that is provided by another component or system?	*Describe the coolant that is provided to include temperature, delta-t, flow rate, pressure, and other characteristics.* *Describe the transmission medium and connector types for the supply and return lines as necessary.* *For components that provide coolant to multiple entities, describe the cumulative demand during normal operations as well as peaks.*
		Does the entity provide low conductivity water to another component or system?	
		Does the entity receive chilled water or other process cooling that is provided by another component or system?	
		Does the entity provide chilled water or other process cooling to another component or system?	
CRYOGENICS	*Direct*	Does the entity receive cryogenic liquids or gases from another component or system?	*Describe the cryogenics that are provided to include type, temperature, pressure, flow rate and other characteristics.* *Describe the transmission medium and connector types for the supply and return lines as necessary.* *For components that provide cryogenics to multiple entities, describe the cumulative demand during normal operations as well as peaks.*
		Does the entity provide cryogenics liquids or gases to another component or system?	
ELECTRICAL	*Direct*	Does the entity receive electrical power from another component or system?	*This interface identifies the physical connections that conduct power between entities, the types of connectors used, the amount of power that is provided, and the characteristics of the power being delivered.*
		Does the entity provide electrical power to another component or system?	*In addition to the physical connections, amount of power and power characteristics, this interface also identifies power transformations that may produce waste heat.*

GAS/FLUID FLOWS	*Direct*	Does this entity require gasses or fluids that are provided by an external component?	*Identify and describe the types of gasses/fluids that are required by the entity, the flow/consumption rate, the characteristics of the flow, and the conduit and connectors that are used.*
		Does this entity supply gasses or fluids that are consumed by another component?	*Identify and characterize the gasses/fluids and delivery mechanisms that will be used, as well as the cumulative amount of material that will be delivered.*
CONTROLS	*Direct*	Does this entity receive signal based control from an external control system?	*Describe the transmission medium, its characteristics and connection types, as well as the communications rate, throughput and protocols that will be used.*
		Does this entity provide signal based control to another component?	*Identify the transmission medium and characteristics, cumulative throughput requirements and protocols.*
		Does this entity receive mechanical control from an external system?	*Identify the mechanism that will be used, functional characteristics and connection types.*
		Does this entity provide mechanical control to another component?	
DATA	*Direct*	Does this entity have a data connection to another component or system that is used to send, receive or exchange information?	*Describe the transmission media, communications rate, bandwidth, throughput, connector types and other characteristics.* *For devices that service multiple sources, identify the cumulative capabilities required to support all components at peak operation.*
		Does this entity rely on an external data system for storage, retrieval, or processing?	*Describe the volume of data that will be exchanged, rate of production, storage capacities, processing performance, and other characteristics.*
		Does this entity generate data that must be transferred to an external data system for storage or processing?	
PARAMETRIC	*Direct/Indirect*	Does this entity rely on specific beam, particle or other parametric requirements from another component?	*These interfaces describe external parameters that do not fit in other categories and may be highly diverse in their nature and characteristics. For each one identify all pertinent factors that must be correlated between the providing entity and the consumer.*
		Does this entity provide a specific beam, particle or other parametric requirement that is needed by another component?	

WASTE & CONSUMABLES	*Direct*	Does this entity generate material waste that must be removed from the system?	*Describes the types of material waste that is produced by this entity, the conduits and connections that are used to remove it, and the overall volume, frequency and rate of production.*
		Does this entity act as an intermediary that receives material waste from other components?	*Describes the characteristics, frequency and rate of waste that is being received from other entities, as well as the cumulative amount of waste that is being received/processed.*
	Direct/Indirect	Does this entity use a consumable resource that must be replaced/refilled periodically?	*This is a diverse interface that can be used to describe anything from batteries, to ink, to glass crystals that darken over time. It should identify the resource, the rate of consumption, the frequency of replacement, the amount used and how it will be provided.*

RECORDING INTERFACE PARAMETERS

As described in the preceding section, there are eleven categories of interfaces that are currently considered. While each of these categories is distinct, frequently all members of a specific interface category will have similar parameters. For instance, any electrical interface is likely to have parameters that describe connectors, wire size, length, etc. In order to speed the documentation of interfaces, each of the categories of interfaces will be assumed to have the parameters defined below. The designer can start from this set of parameters and then add or remove parameters as required for special cases.

- **Structural**

Structural parameters describe the physical characteristics of systems that provide support, stability, or connectivity between two or more entities.

1. Static Load/Capacity: *the load bearing capacity of the assembly when in a fixed location.*
2. Dynamic Load/Capacity: *the load bearing capacity of the assembly when moving.*
3. Material: *characteristics/requirements of the materials that are used to construct the assembly.*
4. Tolerances: *the structure's tolerances to issues such as vibration, imbalance, irregular loads, etc.*

- **Spatial**

Spatial parameters describe the position and dimensions of entities relative to one another.

1. Dimensions: *the limitations and constraints on the size and shape of the entity*
2. Location: *where the entity is required to exist at a specific location in the system.*
3. Orientation: *how the entity is required to be rotated or oriented to a specific position relative to other entities.*
4. Adjacency: *the degree to which an entities placement is governed by other components or systems.*
5. Spacing: *space that must be maintained between entities to provide plenums or buffers.*

- **Environmental**

Environmental parameters describe the characteristics of systems that manage ambient heat, cooling, humidity, etc. or provide shielding/isolation from forces such as radiation or magnetic fields.

1. Target/Set-Point: *the target value (temperature, humidity, etc.) for the system during normal operation.*
2. Variance: *the amount that a managed parameter can vary from the set-point during normal operation.*
3. Tolerance: *the amount of heat, humidity, or other accumulated energy that the system is expected to tolerate.*
4. Dissipation: *the amount of heat, humidity, or other energy that is expected to be dissipated from the system during operation.*

- **LCW/Process Cooling**

These parameters describe the characteristics of systems that provide low-conductivity water and process cooling to maintain the temperature of components.

1. Flow rate: *the rate at which the coolant will be delivered to and recovered from the system.*
2. Temperature: *the nominal temperature of the incoming coolant.*
3. Delta T: *the change in temperature of the coolant during the cooling process.*
4. Quality: *the quality characteristics of the coolant and cooling system.*
5. Pressure: *the pressure of the incoming coolant.*
6. Variance: *the amount that a managed parameter can vary from its nominal value during normal operation.*
7. Tolerance: *the maximum amount of variance that the system is expected to tolerate.*
8. Dissipation: *the amount of heat that is expected to be dissipated from the system during operation.*
9. Connectors: *the types of connectors that are required to attach the connected entities.*
10. Medium: *characteristics of the hoses/piping that will be used to supply/return the coolant (length, diameter, insulation, etc.)*

- **Cryogenics**

Cryogenic parameters describe the flow, temperature, and other characteristics of cryogenic fluids.

1. Flow rate: *the rate at which the cryogen will be delivered to and recovered from the system.*
2. Temperature: *the nominal temperature of the incoming cryogens.*
3. Delta T: *the change in temperature of the cryogenic fluid during the process.*
4. Pressure: *the pressure of the incoming/outgoing cryogenic fluid.*
5. Variance: *the amount that a managed parameter can vary from its nominal value during normal operation.*
6. Tolerance: *the maximum amount of variance that the system is expected to tolerate.*
7. Dissipation: *the amount of heat that is expected to be dissipated from the system during operation.*
8. Connectors: *the types of connectors that are required to attach the connected entities.*
9. Medium: *characteristics of the hoses/piping that will be used to supply/return the cryogenics (length, diameter, insulation, etc.)*

- **Electrical**

Electrical parameters describe the characteristics of interfaces that are used to transfer electrical energy between components, or to provide grounding or redirection of waste energy.

1. Voltage: *the nominal voltage of the incoming electrical power.*
2. Amperage: *the amount of energy that the system is expected to consume.*
3. Variance: *the amount of variance in a parameter (particularly voltage) that can be expected.*
4. Tolerance: *the amount of variance in the quality of the power (spikes, sags, transients) that can be tolerated.*
5. Connectors: *the types of connectors that are required to attach the connected entities.*
6. Medium: *the characteristics of the cabling that will be used to conduct the power (length, wire gauge, number of conductors, material, insulation, etc.)*

- **Gas/Fluid Flows**

These parameters describe the characteristics of gas and fluid flows that are required for the operation and maintenance of components.

1. Flow rate: *the rate at which the fluid will be delivered to and/or recovered from the system.*
2. Pressure: *the pressure of the fluid.*
3. Variance: *the amount that a managed parameter can vary from its nominal value during normal operation.*
4. Tolerance: *the maximum amount of variance that the system is expected to tolerate.*
5. Connectors: *the types of connectors that are required to attach the connected entities.*
6. Medium: *characteristics of the hoses/piping that will be used to supply/return the fluid (length, diameter, insulation, etc.)*

- **Controls**

Controls parameters describe the characteristics of the interfaces that provide control and monitoring of components.

1. Speed: *the nominal transmission rate of the control and monitoring signals.*
2. Volume: *the quantity of control data that is expected to be transferred during operations.*
3. Latency: *the lag time that can be expected during the communications process.*
4. Quality: *the amount of noise and loss that is expected in the communications process.*
5. Variance: *the amount that a managed parameter can vary from its nominal value during normal operation.*
6. Tolerance: *the maximum amount of variance that the system is expected to tolerate.*
7. Connectors: *the types of connectors that are required to attach the connected entities.*
8. Medium: *characteristics of the communications media that will be used.*

- **Data**

These parameters describe the characteristics of the interfaces that transport and transform data that is moving between entities.

1. Speed: *the nominal transmission rate of data lines.*
2. Volume: *the quantity of data that is expected to be transferred during operations.*
3. Latency: *the lag time that can be expected during the communications process.*
4. Quality: *the amount of noise and loss that is expected in the communications process.*
5. Processing: *the amount of intermediate processing that will be required in the communications process (conversion, compression, filtering, buffering, etc.)*
6. Variance: *the amount that a managed parameter can vary from its nominal value during normal operation.*
7. Tolerance: *the maximum amount of variance that the system is expected to tolerate.*
8. Connectors: *the types of connectors that are required to attach the connected entities.*
9. Medium: *characteristics of the communications media that will be used.*

- **Parametric**

Parametric parameters are general purpose reference points for describing beam conditions and other system quality aspects that are not captured by other interfaces.

1. Optics: *beam properties (beta function, emittance, luminosity, etc.) that are needed for the experimental program.*
2. Lattice: *position of all accelerator equipment and magnetic field parameters needed for the experimental program.*
3. Polarization: *requirements for the experimental program on the direction and ordering of beam polarization in the various beam bunches, and their reversal frequency.*
4. Timing: *timing requirements that affect the efficient operation of the detector such as minimal beam lifetimes and those correlated with crab crossing.*
5. Backgrounds: *requirements on the acceptable beam-related backgrounds as driven by the experimental program including implications for magnet alignments, the beam vacuum and "bake-out" requirements.*
6. Equipment safety: *requirements for beam interlocks and fast dump systems to protect the detector elements.*

- **Waste & Consumables**

These parameters describe the characteristics of the consumables that are required to support the system, and the waste that is generated during its operation.

1. Volume: *the amount of a waste or consumables that will be produced/consumed.*
2. Rate: *the rate of production or consumption.*
3. Processing: *the amount of processing that will be required to transfer the material.*
4. Connectors: *the types of connectors that are required to connect the entities.*
5. Medium: *the medium that will be used to deliver or removed the material.*

DOCUMENTING, MAINTAINING AND IMPLEMENTING INTERFACES

The preliminary set of derived interfaces will be collected in spreadsheet form. For each of the interfaces all of the related sub-systems will be identified, a unique interface identifier will be defined, and a rudimentary description of the interfaces characteristics will be recorded. This documents should be sufficient to cross-check and verify that all defined interfaces have been resolved and are consistent between systems.

Future work will focus on creating tracked relationships between the connected entities, where any change to a requirement will also trigger an evaluation of the interfaces that it relies upon. Verification and validation processes should be established to ensure that each interface is adequate to satisfy all dependencies. Finally, a mechanism for performance assurance should be established that ensures that interfaces are implemented in accordance with their requirements.

CONTACT

For additional information, please contact:

Walt Akers E-Mail: akers@jlab.org, Office: 757/269-7669

Elke Aschenauer E-Mail: elke@bnl.gov

Rolf Ent E-Mail: ent@jlab.org

REFERENCES

Akers, W., Aschenauer, E., & Ent, R. (2021). *Requirements Management for Detector Systems.* Newport News: Jefferson Lab.

Pahl, G., & Beitz, W. (1984). *Engineering Design: A Systematic Approach.* London: Design Council.

Wood, K., Stone, R., McAdams, D., Hitrz, J., & Szykman, S. (2002). *A Functional Basis for Engineering Design: Reconciling and Evolving Previous Efforts.* Retrieved from National Institute of Standards and Technology: https://tsapps.nist.gov/publication/get_pdf.cfm?pub_id=821676