33rd Annual INCOSE
international symposium
hybrid event
Honolulu, HI, USA
July 15 - 20, 2023

Understanding Interface Criticality in Large Infrastructure Projects

John Welford
WSP
+64 (0) 22 496 3513
john.welford@wsp.com

Steve Wallace
Shoal Group
+64 (0) 27 773 5316
steve.wallace@shoalgroup.com

James Donovan
Ediom
+61 (0) 401 197 716
james.donovan@ediom.com.au

Abstract. Understanding and effectively managing interfaces has been recognised as key to mitigating the risk of project failure since the 1960’s and is becoming more prominent in infrastructure as projects become more complex. This paper reviews how interfaces in infrastructure projects are traditionally identified, defined, owned, assessed, and managed. It provides recommendations on how to better align this traditional viewpoint with a whole-of-system approach to Interface Management through an assessment of interface criticality and shows the application of these through a case study on the redesign of two rail-enabled ferry terminals.

Introduction

Since the early twenty-first century, and as projects become larger and more complex, Systems Engineering has seen increasing adoption in the infrastructure domain. A systems approach is now widely recommended as a way to mitigate the risk of large project overruns and underdelivery (Crudgington 2020) which have become common on modern infrastructure mega-projects (Denicol, Davies & Krystallis 2020; Flyvbjerg 2014)

There are many approaches, guidelines, and standards for the application of Systems Engineering on projects, including some specifically targeted at the infrastructure domain (Aslaksen et al. 2012). These typically define a set of processes to be followed, which invariably make mention of interfaces on the project and give recommendations on their management. Understanding and properly managing interfaces is therefore already recognised as key to mitigating the risk of project failure.

This paper notes a divergence between the classical approach to the delivery of Interface Management in infrastructure projects, which typically has a focus on the management of identified ‘external’ interfaces between organisations, and a Systems Engineering approach to Interface Management with a more system-specific focus. It then makes recommendations on how these two can be better aligned, by considering the criticality of each interface as a determinant of the level of effort needed to manage and mitigate the risks associated with it. The recommendations are based on the authors’ experience across several organisations and projects, and a case study is presented that applies these on a project which delivers two rail-enabled ferry terminals.

The Benefits of Interface Management in Infrastructure

Review of the performance of megaprojects (defined as projects with a total capital cost of US$1 billion or more) has found that nine out of ten have cost overruns, with up to 50% overrun being common; benefit shortfalls of up to 50% are also noted as common (Flyvbjerg 2014). Recent investigation into this poor performance has identified several common themes and proposed cures that include “designing the system architecture” and “decomposing and integrating the supply chain” (Denicol, Davies & Krystallis 2020).

Whilst a direct association between overall project performance and the implementation of Interface Management is difficult to quantify, many studies indicate a positive correlation:

- Ahn et al. (2017) reference:
    - a project with a total cost of $1-5 billion identifying and managing interface points between interface stakeholders resulting in less rework and early completion of the design by approximately five months, representing $25 million in savings.
    - a project design package of $45 million where the early identification of major interface points between the engineering contractor and the procurement contractor resolved a procurement issue, which resulted in $10 million of savings.
    - a project with total cost of $5-10 billion where the lack of appropriate Interface Management and the failure to recognize a supply and quality issue between the engineering/procurement stakeholder and the construction stakeholder resulted in a penalty of $10 million per week incurred over several weeks.
- Ahn et al. (2017) and Shokri et al. (2016) find that projects with a mature Interface Management implementation typically had lower (4% average) and less dispersed cost growth compared to the ones without a mature implementation (18% average), based on data from 45 large-scale Engineering and Construction projects.
- Shen et al. (2018) find that Interface Management is an effective approach to promote project performance in terms of schedule, quality, and cost.

Although the situation appears to be improving, there are still a considerable number of project communities that are not clear about Interface Management (Sanei & Smith 2019) and therefore not realising these benefits. In particular, the relationship between Interface Management and project complexity has been noted (Ahn et al. 2017; Pritesh & Konnur 2019); finding that formal Interface Management is effective for mitigating the adverse impact of complexity that originates from uncertainty in scope, uncertainty in communication, and large numbers of stakeholders, but not as effective for dealing with project complexity originating from large numbers of engineered items.

Shokri (2014) identifies the top five risk and complexity factors related to Interface Management on a project (in addition to cost, schedule, and execution risk) as:

- Extended, unfamiliar, or poorly defined scope;
- Government involvement;
- Having multiple engineering centres;
- Having multiple engineering, procurement, or construction interface points; and
- Having many stakeholders.

Sanei & Smith (2019) note that existing working cultures and practices are slow to change, and where past practices address some issues of system complexity their adoption is seen as low risk. The existing Interface Management approach used in infrastructure is therefore seen as a good candidate for nudging in the direction of a more holistic Systems Engineering approach, to improve its ability to deal with highly complex mega-projects.

Traditional Infrastructure Projects and Systems Engineering

Interface Management was presented as a concept in 1967, where it was defined based on systems approach (Wren 1967). At that point its delivery was typically considered to be part of a Project Managers role (Morris 1979), however due to the increased complexity of projects over the years, it is no longer appropriate to assume that a Project Manager can manage all interfaces without supporting processes and resources. It has been regularly noted that Systems Engineering provides a good approach to Interface Management, but that the way it is delivered is different from the application of Interface Management in other industries (Laan et al. 2000; Sanei & Smith 2019). Even recently it has been found that (particularly within Rail project communities) there is little consensus as to where the responsibility for Interface Management processes lies on projects, with it typically delivered by either Project Managers or Systems Engineers (Sanei & Smith 2019).

The lack of clarity around Interface Management processes led to the Construction Industry Institute producing an *Interface Management Implementation Guide* in 2014. This attempts to deal with the wide variety of Interface Management implementations in the construction industry, along with a lack of common definitions. Whilst it briefly notes Interface Management as a subset of Systems Engineering, it makes no further reference to taking a systems approach.

This treatment of Interface Management as a standalone project activity is common across many infrastructure projects and a large proportion of the literature. The International Council on Systems Engineering (INCOSE) *Guide for the application of systems engineering in large infrastructure projects* (Aslaksen et al. 2012) makes passing reference to interfaces as part of the overall system architecture but does not directly address Interface Management as an activity, potentially reinforcing the notion that it is an activity ‘outside’ of Systems Engineering.

The standard references for Systems Engineering, *ISO standard 15288* (2015) and the *INCOSE Systems Engineering handbook* (2015), view Interface Management as crosscutting through multiple processes, including: identifying interfaces within the architecture definition process; defining interface requirements through the system requirements definition process; and further refining and detailing in the design definition process. There are also obvious relationships to the project planning, risk management and configuration management processes to provide ongoing management and control. Interface Control Working Groups are proposed, with members responsible for each interfacing element, in order to formalise and enhance collaboration within project teams or between project teams/organisations. They note that one of the objectives of Interface Management activities is to facilitate agreements with stakeholders on: roles and responsibilities; timing for providing interface information; and identification of critical interfaces early in the project.

Noted as part of system architecture definition, the ISO standard on *Systems Architecture Description 42010* (2011) briefly notes that interface identification should be part of the system decomposition process. It also notes that interface descriptions should be provided, but only in the context of software interfaces.

Recently introduced Australian Standard 7473 on *Complex Systems Integration in Railways* (2020) gives more guidance, requiring that “All foreseeable external and internal interfaces between system elements shall be identified and defined.”. This highlights a key distinction between the infrastructure view of interfaces as a more abstract set of interactions to be managed, and the Systems Engineering view of interfaces as a key part of an overall system architecture.

From the authors' experience of infrastructure projects taking a non-systems-driven approach to Interface Management, the interfaces are often identified on an ad-hoc basis through regular interface meetings; Shokri (2014) notes that "The interfaces are identified based on the top management experience, once they are prone to create a problem." and Lin (2015) notes "Conventional interface communication methods include face-to-face meetings, telephone communication, and virtual design and construction.". Such meetings can be heavily focussed on information exchange at the expense of capturing technical interface definition; Ju & Ding (2015) note that "Traditional approaches for interface management such as interface communication meetings or interface related documents cannot adapt to modern automation level in metro equipment engineering any more.".

This paper progresses by exploring the differences in definition and process between these two views and proposes the use of criticality assessment to maintain a manageable quantity of interfaces within a Systems Engineering approach to Interface Management.

Interface Definition

A variety of definitions of what constitutes an interface have been proposed:

- "the contact point between relatively autonomous organizations which are interdependent and interacting as they seek to cooperate to achieve some larger system objective." (Wren 1967)
- "a soft and/or hard contact point between two interdependent interface stakeholders." (*Interface Management Implementation Guide* 2014)
- "a part of the project's scope split as defined by project documents, in which the responsibility passes from one interface stakeholder to another." (*Interface Management Implementation Guide* 2014)
- "a boundary where, or across which, two or more parts interact." (Wheatcraft 2010)
- "reflect communications which take place within or between different parties in each project, with the purpose of transferring information or accomplishing a task." (Shokri 2014)
- "a common boundary or interconnection between independent but interacting systems, organizations, stakeholders, project phases and scopes, and construction elements." (Eray et al. 2017)
- "links between different building elements, stakeholders and project areas." (Pritesh & Konnur 2019)
- "a point of contact between independent entities that are interacting to achieve a larger system." (Sanei & Smith 2019)
- "a point of interaction between (sub)systems" (Complex Systems Integration in Railways 2020)

Various taxonomies to describe types of interface have also been documented, including:

Hard vs Soft: With hard interfaces representing physical connections between two or more components or systems, and soft interfaces typically involving the exchange of information between delivery teams or parties. (*Interface Management Implementation Guide* 2014; Shokri 2014)

Internal vs External: Internal being those identified within a single team, contract or scope of work, and External being those identified between two or more scopes of work. (Chen 2007; Sanei & Smith 2019)

Project vs Infrastructure: Related aspects of an interface that need to be kept aligned, considering the interface as between project elements and/or between elements of the infrastructure. (Sanei & Smith 2019)

Inter-project vs Intra-project vs Extra-project: Reflecting the scope boundaries crossed, with Extra-project interfaces including significant interactions with independent entities outside of a project environment. (Shokri 2014)

Static vs Dynamic: Static interfaces are relationships between ongoing subsystems which do not change as the project develops, Dynamic interfaces arise as a function of the pattern of activity interdependencies generated as the project develops. (Morris 1979)

Chan et al. (2005) review several previous interface categorisations and Chen (2007) aggregates these as:

Physical: Physical connections between two or more construction elements or components.

Functional: Functional requirements/influences presented by one functional element/system upon another function element/system.

Contractual: Interfaces among the general contractor, subcontractors, suppliers, and any external providers, with regard to their work scopes, schedule and responsibilities (arising from the contractual obligations between them).

Organisational: Interactions between various parties (including divisions in a company) involved in a construction project from its initial conception to its final handover.

Resource: Interfaces between equipment, labour, materials, space, or information necessary to design and construct the facility and its components.

From review of these various definitions, taxonomies, and categorisations, it is clear there are a variety of perspectives on the problem, several of which are not approaching it from a whole-of-system delivery perspective. As elaborated in Figure 1, Shokri (2014) proposes a somewhat more systems-centric approach that defines interfaces at different hierarchical levels within a project. This is depicted as a tree structure, reminiscent of a System Breakdown Structure. The discussed approach in developing this is to take a ‘top-down’ approach, using expert review, contract documents, design documents, and Work Breakdown Structures to identify interfaces at level 1, and then delve into lower-level interfaces as part of interface documentation.

We take this idea of the interface hierarchy further, and map it to the interface categorisations above, as resource, organisational and contractual interfaces are predominantly concerned with the higher levels, and physical and functional are concerned with the lower levels, as shown in Figure 1. The taxonomic split of hard or soft interfaces also maps to this, with hard interfaces being those at the bottom of the hierarchy and getting softer the higher up is considered.

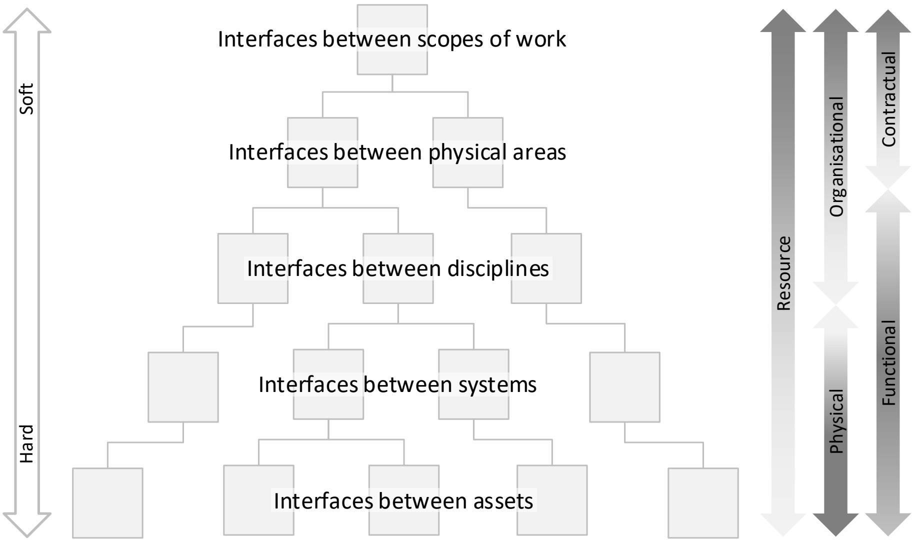

Figure 1. Interface categorisation and hierarchy, developed further from Shokri (2014)

Considering this from a systems perspective, we assert that the higher-level soft interfaces exist specifically because of the existence of lower-level hard interfaces, i.e. an interface between two contract packages can only exist if there are two interacting assets that are part of separate systems, which are delivered by separate disciplines, which exist within separate physical areas and scopes of work. Without a hard physical or functional interface, there is no need for a contractual, organisational or resource soft interface.

The difference in interface definitions above can be explained as concern about different levels of the hierarchy, with some focussed on the softer side (Wren 1967; Shokri 2014; Sanei & Smith 2019) and some with more emphasis on the harder side (Wheatcraft 2010; Pritesh & Konnur 2019; *Complex Systems Integration in Railways* 2020). Neither of these is necessarily incorrect, and the consideration of the softer interfaces makes sense if this is the starting point in identification of the interfaces.

This top-down approach to identification is low-cost, as the set of contracts is always (by definition) smaller than the set of components being delivered. However, it runs the risk of missing critical interfaces that are either contained within a single scope of work, that are with an extra-project party that has not previously been identified as one of the project stakeholders, or that are included within a defined soft interface but not recognised as a hard interface. This risk can be mitigated by taking a bottom-up approach to identification, starting with the identification of component/subsystem level interfaces, and grouping them up the hierarchy.

The identification of hard interfaces is more straightforward where a Systems Engineering approach to project delivery is being followed, as this should include the definition of an overall System Architecture for the work (*ISO standard 15288* 2015; *INCOSE Systems Engineering Handbook* 2015; *ISO standard 42010* 2011). For even a moderately complex system the quantity of interfaces identified through this process is likely to be incredibly large. To follow a comprehensive Interface Management approach for all interfaces identified in System Architecture is likely to be prohibitively expensive with much of the work only providing a modest gain in risk mitigation; therefore the approach proposed in the remainder of this paper works bottom-up through the interface hierarchy, to assess the criticality of each interface and propose a subset that will benefit from treatment through formal Interface Management.

Within infrastructure, new solutions typically represent only a moderate step change in functional design over previous solutions and therefore interface considerations often progress into the physical space quickly. For this reason we focus on physical interfaces for the remainder of this paper, however the proposed approach is expected to work equally well for functional interfaces.

A difference is also drawn between physical/functional interfaces and project/programme/task/ package interdependencies. In much of the Interface Management literature these are combined in the same process, for example Shokri (2014) defines a process for mapping interfaces to key milestones on the project schedule. The management of milestone dependencies between projects within an overall programme is already well treated in the project/programme management literature and therefore an interface is exclusively defined in this work as existing in the delivered infrastructure/system, whereas dependencies exist in the process of developing of the infrastructure/ system. Clearly the presence of an interface that spans projects may indicate a dependency between these projects; it can be dealt with as part of the Interface Management process but should also be managed overall as part of Programme Management.

Interface Identification

As part of a bottom-up approach to the identification of interfaces we can expect that there will be many assets and (for a complex project) an even larger number of interfaces between them, with thousands of items being typical for a megaproject (Shokri 2014). A System Architecture can be considered as a graph having assets as the nodes and interfaces between them as the edges. It can also be represented in a compact view as an N2 or Design (or Dependency) Structure Matrix (DSM) diagram (often referred to as an Interface Matrix in the context of Interface Management) by showing the same set of assets along the rows and columns, and highlighting cells to indicate an interface between the relevant row and column assets. Ideally such an architecture is developed within a Model-Based Systems Engineering (MBSE) environment which allows both a (node-edge based) graph view and an Interface Matrix view of the same underlying model. The most basic view of interfaces that can be extracted from such a model is a simple register listing all interfaces and their attributes.

An initial architecture should be developed relatively early in the project lifecycle and continue to evolve as the design (and construction) develops. This means that interfaces will continue to be identified throughout the project and their management is an ongoing exercise.

Previous work has looked at this type of approach for the identification and management of interfaces as both interface networks (Shokri 2014) and as matrices (Venkatachalam, Varghese & Chandran 2010; Siao & Lin 2012). System modelling methodologies (e.g. IDEF0) have also been noted as suitable for identifying interfaces (Chan et al. 2005) and the definition of interfaces as an explicit object within the model has been proposed (Chen 2007).

From an infrastructure perspective, taking a Building Information Modelling (BIM) approach is now mandated for almost all major projects. The inclusion of interfaces within a BIM model has been explored to both ensure their correct identification and to unify their presentation (Chen 2007; Lin 2015; Eray et al. 2017). This type of approach is expected to be compatible and complementary to the methodology presented, but has not been explicitly explored within this work.

Interface Ownership

When working top-down to identify and define interfaces it is either implicit in the definition of an interface, or a key part of the process, that there are a pair of stakeholders that are interfacing. However, when working bottom-up from the physical interfaces it is necessary to explicitly define who ‘owns’ the interface. For the interface to function properly responsibility must be shared between the owners of the two assets being interfaced. Depending on the lifecycle phase and level of decomposition in the model the owner may be a design discipline, a design project, a construction contractor, an operator, or many others.

This information should already be managed as part of the System Architecture model that has been developed, and there is no need to duplicate this information as interface attributes; however it can be useful to designate the asset on one side of an interface as taking a ‘lead’ role in defining and delivering the interface, with the other taking a ‘match’ (or ‘support’ or ‘follow’) role as shown in Figure 2; Siao & Lin (2012) use the related terms ‘proposer’ and ‘responder’ in a stakeholder context. From a model perspective this can be handled through the definition of different types of relationship, with a multiplicity constraint that each interface must have one of each relationship.

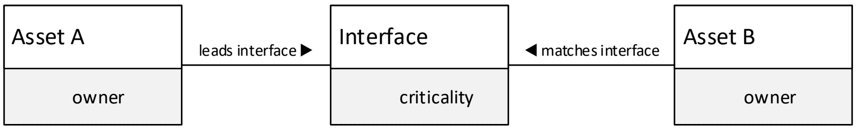

Figure 2. Interface Relationship between Lead and Match Assets

This treatment of the interface as being a logical object rather than a physical object aligns with the idea of Interface Requirements being applied to the assets on either side of the interface, rather than to the interface itself (Wheatcraft 2010). Typically this means that Interface Requirements come in complementary pairs, with one asset owner being responsible for ‘providing’ something across the interface, and the other asset owner being responsible for ‘receiving’ it. The definition and agreement of such requirements is an important part of the Interface Management process; however, they ultimately form part of the overall set of requirements that is allocated to each asset. This means that assurance of their achievement can form part of the standard project verification processes.

Interface Criticality

Managing every asset interface identified within the System Architecture with a high level of integrity is both expensive and low value, as many of these interfaces are typically low risk. To mitigate the largest amount of project interface risk, for the lowest cost, it is necessary to focus on the subset of interfaces which are the most critical.

Within the literature there are several proposed approaches to assessing interface criticality. Expert opinion on the criticality of 13 possible interface factors for Chinese infrastructure projects found the top five to be: Public sector, Bidders, Contractual systems, Private sector, and Public resources (Chan et al. 2005).

The *Interface Management Implementation Guide* (2014) defines critical interfaces as those that “impose high risk to the project in terms of cost, schedule, quality, safety, and other performance goals”. It includes an ‘Interface Complexity Assessment Tool’ for evaluating inter-organisational interfaces based on the nature of both the organisations and the interface itself. It consists of a series of scored and weighted questions that sum to an ‘Interface Complexity Factor’ that can be used to identify critical interfaces and develop strategies for them. It particularly notes that projects with a higher number of critical interfaces should implement a relatively more mature Interface Management strategy overall.

Shokri (2014) notes that the construction industry does not employ a specific approach to rank interfaces based on their criticality, which they define as interfaces that are approaching their closing dates. They propose a network-based algorithm to identify key interfaces based on an interface precedence and interdependency relationships (including information, time, space, sequence, and physical dependencies), with highly interdependent interfaces being treated as key due to their increased risk and impact to the project. Although they also acknowledge that treating an interface as ‘key’ is ultimately a decision based on the judgement of the project leaders.

Most recently, *Complex Systems Integration in Railways* (2020) lays out a series of “Interface Maturity Levels” that can be used to assess the ability of assets to be interfaced. It suggests that a minimum maturity level can then be requested from asset suppliers to reduce integration risk.

Whilst the exact details of the criticality assessment will be project dependent, we propose the following six categories that may be used to develop a criticality scoring system for (‘hard’) asset interfaces:

Impact to project delivery: considering how failure of the interface might affect project delivery timescales and budgets. This includes the impact that redesign and development will have on cost and delivery dates if the interface does not function correctly or cannot be agreed. Increased project impact posed by an interface should be scored higher in terms of criticality. Differences in the delivery date for assets on either side of an interface should be considered in this category, as the earlier one side of an interface is finalised (with respect to the other side), the more critical it is that the interface be properly defined.

Impact to system operation: taking into account how interface issues might affect operation of the overall system-of-interest for the project, including impact to its operational performance or availability. Interfaces with increased operational impact should be scored as more critical.

Impact to safety: understanding the safety hazard posed by poor interfacing between the assets. Higher overall safety risk, in terms of probability combined with consequence, should be considered as more critical. This information could be tied to the project hazard log and Interface Management used to control the risk.

Compliance to standards: determining whether the interface is well covered by existing interfacing standards, or if it is partly or entirely bespoke. The more bespoke the interface is expected to be, or the more it departs from recognised standards, the more critical it should be scored.

Asset ownership (or delivery parties): assessing whether the assets on either side of the interface are owned or delivered by the same parties. Working up the hierarchy in Figure 1, assets that are part of the same system and delivered by the same package and/or discipline will score low. 'External' interfaces between assets on different sides of an organisational or contractual boundary will score higher (much more critical); particularly if the organisations have significant cultural, geographical, or technical maturity differences. Normally, 'internal' interfaces are much easier to handle because a single team is involved, and the ownership and responsibility are clear (Chen 2007). The incentives for each party can also be considered in this category; where one or more parties are not well incentivised (such as not receiving direct benefits from the project), the interface should be considered more critical.

Interface complexity: considering the technical complexity of the interface design, including: the type(s) of connection being made, the previous experience in delivering this type of interface available to the interfacing parties, the size or quantity of the interface, the desired quality of the interface, the technologies involved in the interface, and the environment that the interface is operating in. More complex interfaces should be scored as more critical.

It is recommended that project-specific scoring metrics around these categories (or similar) be set up, including weightings on the categories as necessary. This will allow each interface to be evaluated based on an overall score. The score can then be compared to a threshold, above which the interface is treated as 'critical', and the formal Interface Management process followed.

It may be beneficial to also set a lower 'potentially critical' threshold to identify interfaces that are not currently considered critical, but which should be monitored closely to ensure that changes in project circumstances don't increase their rating. Changes in delivery scope and adjustments to delivery timelines are common as infrastructure projects evolve; this type of change can easily push a potentially critical interface over the threshold into being critical. Whilst initial criticality assessment should be performed early in the system architecting phase as the interfaces are identified, it should receive ongoing review to both recognise new interfaces and to update criticality scores.

The scoring mechanism will provide a clear audit trail and assistance in decision making, but ultimately the determination of which interfaces are critical to project success should be performed by an Interface Management team that is familiar with the system being delivered. There are always additional considerations that are not easy to account for in a scoring metric, so it is suggested as a guideline, with final adjustments possible by project experts.

Interface Definition, Coordination and Delivery

Depending on the size of the project, the overall Interface Register may contain hundreds to thousands of interfaces. From this it is a much smaller subset that are critical to be managed in a formal Interface Management process. Complete definition of a process for managing critical interface is beyond the scope of this paper, but it would typically involve:

- Confirming asset ownership and interface existence with the relevant parties.
- Grouping of similar interfaces to be managed within a single Interface Control Document (ICD).
- Authoring an initial draft of the ICDs – by the asset lead party.
- Discussions between lead and match parties to update and agree the contents of the ICDs.
- Formal agreement (e.g. signing) of the ICDs by both the lead and match parties.
- Inclusion of Interface Requirements in asset requirement specifications, and inclusion of key interface dates in project schedules.
- Ongoing configuration management of all interface-related documentation, including following formal change control processes to manage updates.
- Verification of completed design and the built environment against interface requirements to confirm compliance.

Often this process is managed through regular project-wide interface coordination meetings, however it can be more efficient to manage critical interfaces in dedicated interface meetings; these can be more focussed and limited to only the specific stakeholders involved in each interface. Where a more model-based approach to project delivery is being followed it can be beneficial to define interface actions as a specific object within the model, which can be linked to the interface and used to track the minutes of meetings and the closeout of minor interface related communications. Although it is important to avoid overlap between these and the more formal interface requirements.

There can be a significant difference in the definition of what constitutes an ICD on an infrastructure project, with alternative names or additional documents including: Interface Definition Documents, Interface Requirement Documents, Interface Definition Agreements and Interface Agreement Documents (Wheatcraft 2010). The important underlying need is a medium to document, and gain agreement on, interface details between interfacing (i.e. interface asset owning) parties. Our preference is to use the ICD to detail:

- A high-level description of the interface(s), to set the scope of what is included, especially the delineation of responsibility between parties; but importantly, not to detail the design of the interface. Interface design should be part of design documentation, associated with the interfacing assets; where this already exists it may be referenced from the ICD.
- Reference to the interface requirement pairs applied to the assets on either side of the interface. This should be by reference, and not form the master version of these requirements. They should be mastered in the project requirements management tool and included in the relevant asset requirement specifications.
- Details of the information exchange needed for the process of developing the interface. These should not be requirements for the asset interface itself, only the information that needs to be exchanged to facilitate asset development. Some of this information may result in updated asset interface requirements, some of it will be specifications that are referenced by the requirements. All information exchange defined in the document should be accompanied by a due date that aligns with the overall programme of work. The dates may also be included in overall project management planning, but the agreement to the dates is through the ICD.

Where the same two parties are responsible for multiple interfaces, workload can be reduced by grouping them within a single ICD. This brings the overall quantity of documents produced back into line with a non-system-centric 'top-down' Interface Management approach, but with the added confidence that all interfaces (and interfacing assets) have been identified. In determining what interfaces can be combined into a single ICD the following considerations are important:

- Are the lead and match parties the same (and the same way around)?

- Are the same people/teams within the lead and match responsible for all the interfacing assets?
- Are the dates that the interfaces need to be developed compatible?
- Will any issues with agreeing one interface unnecessarily delay the agreement of other interfaces in the same document?

Whilst the critical interfaces need the highest level of management, it's important that the non-critical (and potentially critical) interfaces are not ignored. Where they are delivered within a single system, package, or discipline they should be a part of standard design development. Where they do span delivery parties the project's interdisciplinary design coordination and review processes may be sufficient for them, however this should be explicitly noted during project planning.

Case Study: Ferry Terminal Design

This case study focuses on Interface Management during the design of the ferry terminals at both Wellington and Picton in New Zealand, to support an updated ferry service across the Cook Strait serving pedestrians, passenger vehicles, commercial vehicles, and roll-on roll-off rail freight. It provides an overview of the processes followed and overall outcomes of Interface Management on the project.

The detailed design for the terminals was based on an initial concept design provided by the client. This concept provided a reference for developing the initial asset breakdown, that was then refined and updated as the design progressed through each design stage. At publication the interfaces developed and managed across the design have gone through a period of stability but are currently subject to revision through the need for value engineering design updates.

The following subsections provide an overview of the process used within the terminals design to define interfaces, assess interface criticality, derive critical interface requirements, and verify detailed design against these requirements. This process was conducted through the detailed design phase to identify interface risks and mitigate them prior to construction, with the aim of cost savings through a reduced need for rework. Although this process is described as linear, it should be noted that there is some iteration in accordance with a staged design delivery process, and interfaces naturally evolving alongside the design, The Interface Management activities described within this paper are currently ongoing through the value engineering design updates.

Identify Assets, Delivery Parties, and Interfaces

The approach taken was to identify and manage interfaces between assets and rather than work packages, which is somewhat uncommon when compared to the wider infrastructure literature referenced above. This is in keeping with taking a whole-of-system approach to design delivery rather than considering the project as a set of separate work packages and trying to manage the interactions between them.

The assets that comprise the complete system were captured in a hierarchical asset breakdown tree structure that allows a child asset to only have one parent element. The breakdown was decomposed to a level where individual assets had clear ownership by a specific work package, delivered by a single design team. It was then at the asset level where interfaces were identified through design documentation review and workshops with asset designers. Interface categories captured include electrical signals, high voltage, low voltage, air, fuel, geometric, mechanical, and structural. Identified interfaces were documented, along with the assets, within an MBSE tool, and communicated to the design team through the export of an interface register and an interface matrix. Within the MBSE tool one lead and one match asset were linked to each interface.

The terminals asset breakdown structure identified 426 assets across both terminals, of which around 250 have one or more interfaces; over 1500 interfaces have been identified between these assets.

Assess Interface Criticality and Group Critical Interfaces

For a project of this size, a systematic approach to interface definition inherently creates a lot of interfaces, which can range in complexity from a singular, low-criticality geometric interface, where assets on both sides are designed by an individual designer, through to a complex multidisciplinary interface which spans delivery boundaries.

To conduct a detailed process of deriving requirements and verifying design for all identified interfaces would either result in either an onerous level of effort or insufficient detail per interface, potentially meaning high-risk areas of the design not being properly managed. An interface criticality assessment was developed to understand which interfaces carried the highest risk to the success of the design, and therefore required detailed Interface Management, and which were lower risk interfaces that could be managed through 'traditional' design management techniques such as interdisciplinary reviews.

To score the criticality of the interfaces, consideration was given to the key objectives for the project which included minimising the ferry turnaround time, delivering the works quickly, and overall safe operation of the system. The other significant consideration was the delivery team – where interfaces were entirely delivered by the internal design team, coordination was expected to be less onerous. Coordination with external interfacing parties, such as the ship designer, was necessarily more complicated, as the interface crossed a contractual boundary. Table 1 shows the criticality assessment categories used for the project, and the scoring metric used in each category.

Table 1. Interface Criticality Assessment Ratings

Category	**Rating**	**Score**
Impact to Project Delivery and Operation	Significant impact on Delivery and Operation	4
	Moderate impact on Delivery and Operation	3
	Low impact on Delivery and Operation	2
	Negligible impact on Delivery and Operation	1
Impact to Safety	Significant risk to life safety	4
	Moderate risk to life safety	3
	Low impact to life safety	2
	Negligible impact to life safety	1
Compliance to Standards	Bespoke complex arrangements, departure from standard	3
	Bespoke arrangement, no direct standards available	2
	Designed to approved standards	1
Delivery Team	Two entirely separate organisations	5
	One organisation with a subcontractor	3
	Two separate teams within one organisation	2
	Within one team within one organisation	1
Complexity of Interface	5 or more interface types	3
	2 to 4 interface types	2
	Singular interface type	1

Based on the total score across all five categories, each interface was classified as either 'potentially critical' or 'non-critical', based on a threshold score of 10. Assessing all system interfaces resulted in 10% being classified as 'potentially critical'. This list was then developed through workshops with design teams and directors to identify the interfaces that were truly 'critical' and therefore required rigorous management. Of the 150 'potentially critical' interfaces, 50 (or ~3%) were identified as critical and therefore to be managed through ICDs.

Where the parties owning assets on either side of a critical interface were similar, and it is otherwise practical, these were grouped into single ICDs. This resulted in a reduction to 20 ICDs across the project, reducing the work needed by the design team in documenting interface information and requirements, and tracking their progress. In assessing the practicality of the groupings we found it important to not introduce ambiguity, to ensure that the allocation of lead/match was consistent across all interfaces within an ICD, and that tripartite ICDs were avoided.

Define ICD Detail, Derive Interface Requirements and Gain Agreement

This subsection describes processes that are currently ongoing and not yet concluded at the time of writing. Development of ICDs is coordinated by the Systems Engineering team but delivered by the relevant design teams. The ICDs are not used to generate new information but instead pull together references to various sources of developing design information throughout many design documents. Each ICD includes three core sections: an interface overview; interface requirements; and interface information exchange needs. The overview section details where interfaces are located, the assets the interfaces connect, which parties are responsible for each asset, and how these assets are expected to interact across the interface. Most of this information is documented within design reports for the associated assets which are referenced along with brief written descriptions and snapshots of 3D models or design drawings.

The most significant content within the ICDs are the interface requirements, which are initially defined in ICDs but subsequently will be managed within DOORS alongside the other project requirements. The interface requirements come in pairs; one for the lead and a counterpart for the match asset. Each pair of requirements are defined with consistent references and variables but differ in the asset that is the subject of the requirement.

Once agreed through the approval of the ICD, derived interface requirements will be embedded within project requirement specifications for the relevant design packages. The project confirms compliance of the design against requirements through the population of verification registers on a per-package basis. Any non-compliance highlighted through this process will need to be formally addressed through either a design change or ICD/requirements change. A design that can provide evidence that it complies with the interface requirements should support an integrated outcome.

The ICDs also capture any of the information exchange needs and discussion outcomes. These are considered as distinct from the formal requirements as they pertain to the process of delivering the design, rather than specifying the outcomes of design. The information exchange needs are documented and specified with target dates in the ICD such that ongoing supply of information relating to the interface progressively reduce uncertainty about the ICD as the project progresses (in alignment with the overall project schedule). The result of information being exchanged can potentially be new or modified interface requirements, which will require an update to the ICD according to the project’s document control processes.

Agreement of ICD content, particularly derived requirements, is crucial to mitigating design risk and allowing the design to progress concurrent and differing pace design packages. ICDs have been drafted and proposed by the lead party and will be reviewed, updated, and agreed upon by the match party. Workshops between parties will be used to finalise changes to the document, and signatures gained to confirm agreement.

Outcomes

The process followed by the project was based on identification of the low-level asset interfaces which were then aggregated into package or team interfaces at the ICD level, enabling a whole-of-system analysis approach. Interface criticality assessment was used to focus the effort of detailed Interface Management on a manageable number of critical interfaces. The grouping of similar interfaces into ICDs reduced design team effort and enabled tracking of ICD and interface requirements progress.

At the time of writing, the project has delivered detailed design across a number of packages. However a subset of the packages are undergoing significant revision through a value engineering exercise. We have found that even with focus being on the high-criticality interfaces, it was difficult to get engagement from the design team in the development of ICDs early in the design phase, due to them not considering it the highest priority task within a high-paced design environment. As design matured, the risk around the critical interfaces increased and design team engagement improved, however engagement with external match parties was also problematic. This was primarily due to the external parties not having an active contract at the time that ICD engagement was required as their work had either already concluded or had not yet begun.

Whilst the proposed process has worked well in some cases, the lack of a mandate for its delivery and specific implementation, from both the client and the within the project delivery structure, has resulted in other project delivery activities taking precedence. In future projects we would recommend including interface register and ICD development and delivery milestones within the project programme, such that certain gate reviews cannot be passed without signoff.

Conclusion

Interface Management for infrastructure projects varies in its implementation depending on the level of Systems Engineering involvement. Identification of interfaces in a ‘traditional’ infrastructure project approach lacks the rigour of an identification process that is based on the overall system architecture; however, the use of a system-based approach to identification will typically produce a very large interface list that can be time and resource consuming to manage in detail. Applying Interface Management equally across all interfaces within a large and complex project will likely result either in overspending or lack of detail, with the risk of critical interfaces not being properly managed.

An approach to assessing the criticality of each system interface has been proposed and a case study on its implementation presented. This allows for a whole-of-system approach to be applied, ensuring that all interfaces are identified, but that rigorous Interface Management processes are only applied to the critical interfaces that require it. The case study suggests that around 3% of all interfaces may be critical enough to warrant the application of comprehensive management.

An approach to managing the identified critical interfaces is presented through the development of ICDs containing interface requirements and agreement to them by both the lead and match asset owners. This enables the tracking of design compliance and provision of evidence towards achieving an acceptable design. It is suggested that ‘non-critical’ interfaces can be sufficiently managed through standard design meetings and interdisciplinary coordination.

The focus of this paper and case study has been on the design phase of the project. It is acknowledged that the criticality of interfaces can change throughout a system’s lifecycle, and it is recommended that re-assessment interfaces should be conducted when a system transitions to the next lifecycle phase (as a minimum).

References

Ahn, S, Shokri, S, Lee, S, Haas, CT & Haas, RC 2017, 'Exploratory study on the effectiveness of interface-management practices in dealing with project complexity in large-scale engineering and construction projects', *Journal of Management in Engineering*, vol. 33, no. 2.

Aslaksen, E, Delamare, M, Fehon, K, Godau, R, Knott, A, Kouassi, A & De Liefde, J 2012, 'Guide for the application of systems engineering in large infrastructure projects', *INCOSE Infrastructure Working Group San Diego*.

Chan, WT, Chen, C, Messner, JI & Chua, DK 2005, 'Interface management for Chinas build–operate–transfer projects', *Journal of Construction Engineering and Management,* vol. 131, no. 6.

Chen, Q 2007, *An object model framework for interface management in building information models*, Ph.D. thesis, Virginia Tech.

Crudgington, A 2020, *A Systems Approach to Infrastructure Delivery*, Institute of Civil Engineers.

Denicol, J, Davies, A & Krystallis, I 2020, 'What are the causes and cures of poor megaproject performance? A systematic literature review and research agenda', *Project Management Journal*, vol. 51, no. 3.

Eray, E, Golzarpoo, B, Rayside, D & Haas, C 2017, 'An overview on integrating interface management and building information management systems', *Proceedings of the 6th CSCE-CRC International Construction Specialty Conference*.

Flyvbjerg, B 2014, 'What You Should Know About Megaprojects and Why: An Overview', *Project Management Journal*, vol. 45.

INCOSE 2015, *INCOSE Systems Engineering Handbook: A Guide for System Life Cycle Processes and Activities*, Wiley, v4.0 edn.

International Organization for Standardization 2011, *Systems and software engineering – Architecture description*, ISO/IEC/IEEE 42010.

International Organization for Standardization 2015, *Systems and software engineering – System life cycle processes*, ISO/IEC/IEEE 15288.

Ju, Q & Ding, L 2015, 'A web-based system for interface management of metro equipment engineering', *Journal of Intelligent & Robotic Systems*, vol. 79, no. 3.

Laan, J, Wildenburg, L & Kleunen, P 2000, 'Dynamic interface management in transport infrastructure project', *2nd European Systems Engineering Conference*.

Lin, YC 2015, 'Use of BIM approach to enhance construction interface management: a case study', *Journal of Civil Engineering and Management*, vol. 21, no. 2.

Morris, PW 1979, 'Interface Management–an Organization Theory Approach to Project Management', Project Management Institute, vol. 10.

Pritesh, S & Konnur, B 2019, 'Interface Management and its Effect on Project Complexity in Construction Management', *International Research Journal of Engineering and Technology,* vol. 06, no. 05.

Rail Industry Safety and Standards Board (RISSB) 2020, *AS Complex System Integration in Railways*, AS 7473:2020.

Research Team 302 The University of Texas at Austin 2014, *Interface Management Implementation Guide*, Construction Industry Institute.

Sanei, H & Smith, A 2019, 'Interface management, who is to be blamed? Who is best to own interface management in multi-disciplinary rail infrastructure projects? Systems Engineers or Project Managers?', *AusRAIL PLUS*.

Shen, W, Choi, B, Lee, S & Tang, W 2018, 'Effects of formal practices and teamwork on interface management performance in large-scale construction projects', *Construction Research Congress*.

Shokri, S 2014, *Interface management for complex capital projects*, Ph.D. thesis, University of Waterloo.

Shokri, S, Ahn, S, Lee, S, Haas, CT & Haas, R 2016, ‘Current status of interface management in construction: Drivers and effects of systematic interface management’, *Journal of Construction Engineering and Management*, vol. 142, no. 2.

Siao, FC & Lin, YC 2012, ‘Enhancing construction interface management using multilevel interface matrix approach’, *Journal of civil engineering and management,* vol. 18, no. 1.

Venkatachalam, S, Varghese, K & Chandran, A 2010, ‘A web-based system for design interface management of construction projects’, *Automation in Construction*, vol. 19.

Wheatcraft, LS 2010, ‘9.2. 2 everything you wanted to know about interfaces, but were afraid to ask’, *INCOSE International Symposium*, vol. 20, Wiley Online Library, vol. 20.

Wren, DA 1967, ‘Interface and interorganizational coordination’, *Academy of Management Journal*, vol. 10, no. 1.

Biography

Dr John Welford holds a master’s degree in Electronic, Systems and Control Engineering and a PhD in Mechatronics. He has previously worked in the defence and automotive industries, and currently leads the WSP New Zealand Systems Engineering team. He predominantly works with model-based approaches and system architectures, but also has a strong interest in data visualisation. John is the current Secretary of the New Zealand Chapter of INCOSE.

Steve Wallace is the Program Lead for Transport & Infrastructure within Shoal Group. He has worked across the transport, defence and aerospace sectors predominantly on requirements engineering, model-based systems architecture and verification & validation in both the UK and New Zealand. He holds a master’s degree in Electrical and Electronic Engineering and was a member of the core team that initiated the New Zealand Chapter of INCOSE.

James Donovan is a Senior Systems Engineer specialising primarily on the management of interfaces within the Transport and Infrastructure sectors. He holds bachelor’s degrees in Civil Engineering and Project Management. James was a member of Shoal Group at the time of developing this paper and contributing to the case study.