Challenges of big-science: A matrix-based interface model to manage technical integration risks in multi-organizational engineering projects

Torben Beernaert[1,3], Pascal Etman[2], Maarten de Bock[3], Ivo Classen[1], Marco de Baar[1]

[1]Dutch Institute For Fundamental Energy Research (DIFFER)
[2]Eindhoven University of Technology
[3]ITER Organization

Abstract: Tomorrow's big science systems are in development today. Success depends on global collaboration in which multiple international parties produce unique assemblies. Inter-organizational barriers, interests and technical conflicts often complicate the design and realization process. It is especially difficult to manage a system's interfaces over such boundaries. We propose a model-based approach to support integrators in such situations. Combining a system architecture model, Integration Readiness Levels (IRLs) and a network of actors, we can derive a newly introduced Collaborative Interface Risk Index (CIRI) that highlights risky interfaces, the organizational patterns that are required for successful integration and a progress indicator that tracks how many interfaces have been defined sufficiently. We demonstrate the method for one of ITER's diagnostic subsystems, which is being developed by six distributed organizations. The result is a Multi-Domain Matrix (MDM) that gives a complete overview on the complexity and risks of the collaborative engineering project.

Keywords: Modeling of Socio-Technical Systems, Interface Management, DSM Methods and Tools, System Architecture, Systems Engineering & MBSE

1 Introduction

Much of history revolves around improving technological innovations that give humanity the power to solve bigger and bigger problems. Today, we urgently need sustainable energy technologies to avert ecological and societal breakdown. One potential candidate is nuclear fusion (Entler et al., 2018). However, it is not yet clear that humanity can harness this power here on earth (Donné, 2019). All eyes are on ITER, a nuclear fusion reactor of the tokamak type that is presently under construction. ITER will address a number of scientific challenges associated with burning plasma science and technology. ITER features various methods to heat the fusion fuel, the plasma, to 200 MK and over 120 diagnostic systems to measure various plasma parameters and fusion reaction products.

ITER is currently still in development and will be arguably the most complex engineered system on earth. Designing ITER requires integrating knowledge from multiple scientific community and technical expertise from a vast range of industries, as is common in big science projects (Nicquevert and Boujut, 2021). ITER effectively is a product of global collaborative engineering, where hundreds of research institutes and industrial companies contribute to the parts of the machine. The problem is that ITER can only achieve its mission if all partners align their design decisions with their neighbors (Sosa, 2007). However, inter-organizational barriers, limited communication channels and conflicting objectives can inhibit the collaborative design processes (Kherbachi et al. 2020).

So what does it mean to design systems in a collaborative fashion? We have to understand that complex systems design comprises two distinct activities: Architectural and embodiment design. In architectural design, system architects specify the system's parts, their functionality and their interfaces. In the embodiment design that follows, groups of designers implement the specified functionality in individual parts, while the system architects again oversee the integration of the various parts into an operational system. An important aspect of system integration is detailing the interfaces, a coordination task that requires a lot of alignment with designers. There are plenty techniques and methods that support system architects in managing and integrating interfaces.

Design Structure Matrices (DSM) are popular models for the integration of complex systems, since they give a compact visualization of interfaces (Browning, 2016). Two particularly relevant developments are: (1) Augmenting DSMs with Interface Readiness Levels (IRL's) (Sauser, 2009) to assess interface maturity during the design process (Yasseri and Bahai, 2018), and (2) using DSMs to transfer technical interfaces to communication channels between distributed design teams (Sosa, 2007; Yang et al., 2015).

There is a lot of DSM literature about managing interfaces. Also the alignment of the architecture of the technical system with the organization developing it, is a well studied subject (Sosa, 2007; Cabigiosu and Camuffo, 2012; Colfer and Baldwin, 2016) Management of technical interfaces in a heterogeneous collaborative development organization such as ITER is less frequently described in the literature. Frequent questions that arise in such an environment are: which interfaces should system architects prioritize when there is limited communication between contractors? Who should align which interface with whom?

To answer these questions, Section 2 of this paper surveys the relevant concepts on DSMs and Interface Readiness Levels (IRL) in view of system architecture. In Section 3, we present a novel modelling approach that augments a product DSM model of the technical system with interface maturity levels and a mapping to an organization DSM. The model enables to distinguish interfaces that require high coordination effort from those that require low coordination effort. Immature interfaces are projected onto an organization DSM that shows which collaboration partners need to align their designs. The proposed method has been successfully applied in an ongoing development project of one of the optical subsystems of ITER. The method has proven to be a valuable contribution to steer the collaborative development efforts in the project.

2 Literature

Technical interface risks occur when the design of two components is inconsistent, and can manifest themselves in multiple failures. During the assembly phase, one may find out that geometrical mismatches make it impossible to connect components. It can also happen that the components' behaviors affect each other negatively in operation. This could lead to non-functional or even dangerous situation. It is well known that the cost of resolving a failure scales exponentially with the phase of the project (Paulson, 1976). Any integration failure is followed either by costly rework of the component itself, by propagated changes to surrounding components, or by non-conformity of the system as a whole.

The components and interfaces of an engineered system are represented in its system architecture. While designers work on individual components, the system architects have to detail interfaces from a conceptual to a mature state. Sauser's scale of Integration Readiness Levels (IRL) is a tool to assess the maturity of interfaces, see Table 1. Yasseri and Bahai (2018) consider three stages of IRL's: the semantic stage, IRL 0-2, the syntactic stage, IRL 3-5, and the pragmatic stage, IRL 6-7. We focus on the syntactic stage, where designers can build on previously established semantics. The architects' main objective is to ensure that any integration effort will be according to specifications, which is to be validated in the subsequent pragmatic stage.

Table 1. Integration Readiness Levels (IRL) indicate the maturity of an interface between two components (Sauser, 2009).

IRL	Definition
7	The integration of technologies has been verified and validated with sufficient detail to be actionable.
6	The integrating technologies can accept, translate, and structure information for its intended application.
5	There is sufficient control between technologies necessary to establish, manage, and terminate the integration.
4	There is sufficient detail in the quality and assurance of the integration between technologies.
3	There is compatibility (i.e. common language) between technologies to orderly and efficiently integrate and interact.
2	There is some level of specificity to characterize the interaction (i.e. ability to influence) between technologies through their interface.
1	An interface between technologies has been identified with sufficient detail to allow characterization of the relationship.

Interfaces with a low IRL have few details, are ambiguous and could therefore lead to drastic inconsistencies. They pose a risk for system integration and should be fostered over interfaces with a high IRL. But large systems can easily contain thousands of interfaces - how can a system architect ever interpret such complexity and prioritize particular interfaces?

Design Structure Matrices (DSM) enable us to visualize the architecture of large systems (Browning, 2016). The system architecture of a technical system is intuitively represented in a product DSM, a square matrix with the system's components as row and column elements, and interfaces as off-diagonal entries. A DSM entry can be binary, scalar, or contain multiple numerical values, which can be displayed using symbols or colors for various types of entries. Figure 1 shows an example of a system architecture represented as a network and its equivalent DSM. While product DSMs are frequently used to study modularity in both hardware and software systems (Eppinger and Browning, 2012), the synergy of product DSMs and IRLs in the interface management process has received less attention (Yasseri and Bahai, 2018; Yasseri and Bahai, 2019).

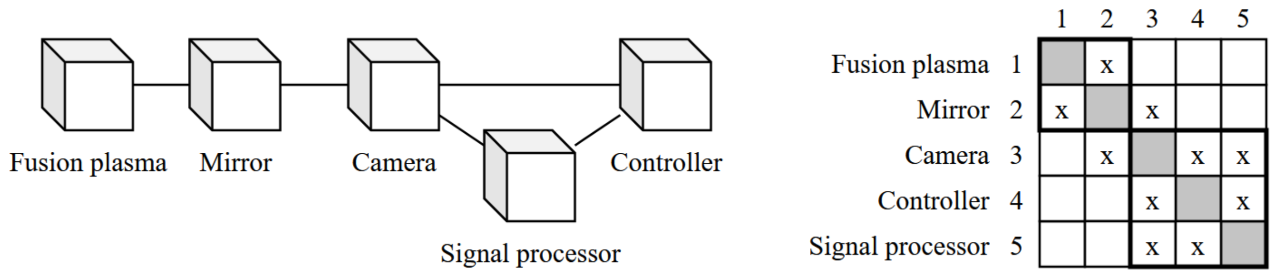

Figure 1. A simplified system architecture of an optical nuclear fusion diagnostic system, represented as a network (left) and its equivalent product DSM (right).

Wilschut et al. (2018a) have realized that a product DSM can be derived from structured function specifications. They use goal functions in the following syntax to express an interface between two components:

Syntax	**Component**	+	**verbs**	+	**parameter**	+	**preposition**	+	**component**
Example	*Motor*		*must provide*		*torque*		*to*		*pump*

This example specifies a transmission of torque that characterizes an interface between a motor and a pump. Wilschut et al. (2018a) have expanded the idea of structured system architecture specifications into a novel language, the Elephant Specification Language (ESL) (Wilschut et al., 2018b; Wilschut, 2018). In the next sections, we will build our method on the ESL modelling syntax.

DSMs are also used to represent organizational structures and to investigate alignment between technical system architecture and organization architecture (Sosa, 2007; Zouggar et al., 2009). Most notably, Yang et al. (2015) have developed a method to derive Technical Communication Strength (TCS), the amount of communication that is needed between any two actors, from the technical interfaces interface strength in a product DSM.

We are not aware of publications addressing specifically DSM modelling methods highlighting risks and opportunities in the interface management process in distributed, collaborative design and development projects of large-scale technologically advanced systems.

3 Method

We propose a modelling method with the following purposes:

1. To highlight interface risks - those that are characterized by both a low IRL and a high coordination effort.
2. To identify communication channels between those designers that still need to agree on interfaces.
3. To monitor progress in the interface maturation process.

Our model comprises the system architecture, the collaboration partners, and the mapping between them. Ultimately, the model will be visualized in a Multi-Domain Matrix (MDM) consisting of:

- a product DSM, representing the system components and interfaces between them,
- an organization DSM, representing the collaborating actors and the communication channels between them, and
- a product-organization Domain Mapping Matrix (DMM), representing which actor is responsible for the design of which component.

Figure 2 gives a schematic overview of the MDM. The MDM is intended to provide a complete overview to system architects and support them with the aforementioned purposes.

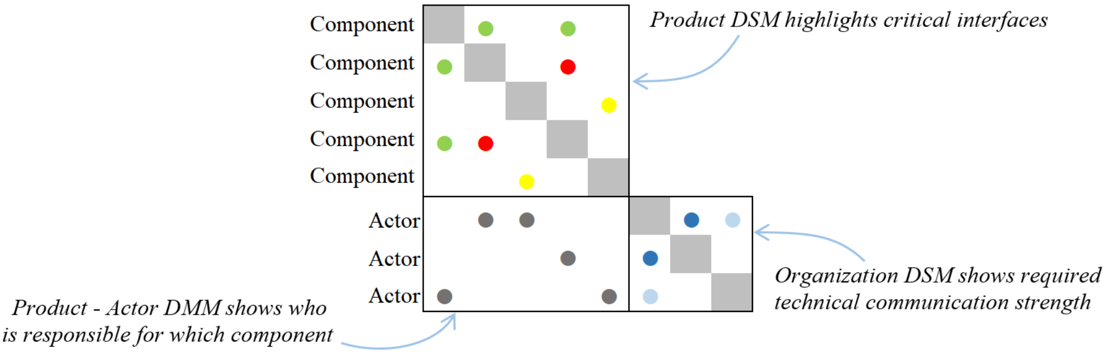

Figure 2. Schematic overview of the Multi-Domain Matrix (MDM) that will be the result of our modelling method. Red, yellow and green entries in the product DSM represent interfaces that are at high, medium or low risk, respectively. The intensity of entries in the organization DSM reflects the required intensity of potential communnication channels.

3.1 Modelling

An MDM is essentially a matrix view of a multi-domain network of nodes and edges. Nodes represent the product domain and the organization domain through a set of components C and actors A, respectively:

$$C = \{c_1, c_2, \ldots\}, \;\; A = \{a_1, a_2, \ldots\} \qquad (1)$$

An actor is an individual person or a cohesive group of individuals that has the knowledge, expertise and authority to carry out design work. We mainly consider actors as organizations, since this research revolves around global collaborations of research institutes and industrial enterprises.

There are three sets of edges between nodes: Interfaces CC between components, communication channels AA between actors, and responsibilities AC between an actor and a component:

$$CC = \left\{ cc_1, cc_2, \ldots \;\middle|\; \begin{array}{l} cc = (c_1, c_2, IRL, Description) \\ (c_1, c_2, \ldots) \leftrightarrow (c_2, c_1, \ldots) \end{array} \right\} \qquad (2)$$

$$AA = \left\{ aa_1, aa_2, \ldots \;\middle|\; \begin{array}{l} aa = (a_1, a_2) \\ (a_1, a_2) \leftrightarrow (a_2, a_1) \end{array} \right\} \qquad (3)$$

$$AC = \{ ac_1, ac_2, \ldots \;|\; ac = (a, c) \} \qquad (4)$$

Edges are written as tuples, such that we can refer to their elements by point notation (e.g. the actor of responsibility edge ac is denoted as $ac.a$). Each interface has an associated IRL and a description that contains further details in free text. Interfaces and communication channels are assumed to be symmetric, which is indicated by the second clause of Equations (2) and (3). Modelling an edge between nodes A and B is equivalent to modelling an edge between B and A.

The model is generated from textual goal requirement specifications in the ESL format for a list of components (Wilschut et al., 2018b; Wilschut, 2018). The ESL goal statement format has been explained in the previous section. Figure 3 shows an excerpt of a specification for the Visible Spectroscopy Reference System (VSRS), an optical diagnostic subsystem of ITER that serves as a demonstration in Section 4. The specifications in ESL format can be computer processed to automatically derive from the text the components C and interfaces CC (Wilschut, 2018). The specifaction text is extended with keyword tags to embody the IRL, the descriptions and the actors involved. For this purpose, we introduce three keywords (Table 2): *@IRL* tagged to a goal requirement statement attributes an IRL to the interface cc derived from this goal statement; *@Description* tagged to a goal requirement statement adds free text to the interface cc derived from this goal statement; and *@Responsible* tagged to a component c in the list of components defines a responsibility edge ac between actor a and component c. Through the *@Responsible* tags with the various components, the set of all actors A is defined implicilty.

```
1   components
2       Patch_panel
3           #< @Responsible Actor 2
4       Polychromator
5           #< @Responsible Actor 6
6           #< @Responsible Actor 3
7       Survey_spectrometer
8           #< @Responsible Actor 2
9       HR_spectrometer
10          #< @Responsible Actor 2
11      IO_Chassis
12          #< @Responsible Actor 1
13
14  goal-requirements
15      A: Patch_panel must provide Light_to_FBP to Polychromator
16          #< @IRL 5
17          #< @Description Optical input fibre bundle
18          #< @Description Patch panel: 10x FC/APC connector
19          #< @Description Polychromator: Multifibre SMA connector
20      B: Patch_panel must provide Light_to_survey to Survey_spectrometer
21          #< @IRL 5
22      C: Patch_panel must provide Light_to_HR to HR_Spectrometer
23          #< @IRL 2
24      D: Polychromator must provide Measurement_data to IO_Chassis
25          #< @IRL 2
```

Figure 3. Excerpt of an extended ESL specification. In lines 1-12, five components and their responsible actors are declared (defining C, A, AC). Lines 14-25 specify four interfaces in the form of goal function requirements (defining CC).

Table 2. The system architecture model in ESL is extended by keywords to augment interfaces with additional information and to map components to actors in an organizational model.

Keyword	**Tags to**	**Tags**	**Tag purpose**
@IRL	Goal statement about interface cc	Integer 1-7	Attributes an IRL to the interface cc expressed by the goal requirement statement.
@Description	Goal statement about interface cc	Text	Adds free text to the interface definition.
@Responsible	Component c	Actor	Creates a responsibility edge ac between a component and an actor. Generates the actor if it does not yet exist in the model.

Section 3.2 explains how we identify the interfaces that need collective attention by assigning criticality categories to the interfaces. Section 3.3 presents how the communications between actors that are needed to resolve the interface issues are determined and included in the actor DSM of the MDM. Finally Section 3.4 explains how progress in resolving these issues can be monitored.

3.2 Interface risks

Our model captures integration risks from two sources. Firstly, immature interfaces increase the chance that a design change in one component requires a big change in another. This source is captured in the IRL field of the interfaces. Secondly, if the actors that are concerned with an interface are highly distributed (e.g. actors work for different companies at different locations, possibly in different countries), it takes more effort to come to an agreement. Distributed actors have infrequent communication and different digital infrastructures or may speak in different languages from different time zones (Yang et al., 2015; Kherbachi et al. 2020). Once such practical organizational barriers are overcome, actors still have to deal with conflicting technical or budgetary interests. This second source relates to the coordination effort in dealing with the particular interface and is therefore captured by the mapping between interfaces and actors.

We capture risks from immature interfaces in an IRL deficit metric. During the detailed design stage, all interfaces should be brought from an initial $IRL_0 = 2$ to a final $IRL_{100} = 5$ (Yasseri and Bahai, 2018). We therefore introduce the immaturity metric $\Delta(cc)$ relative to the upper bound:

$$\Delta(cc) = IRL_{100} - cc.IRL \qquad (5)$$

The second interface metric, coordination effort, is proportional to the amount of concerned actors. Let $A(cc) \subseteq A$ be the subset of actors that are responsible for any of the two component connected by interface cc:

$$A(cc) = \{ac.a \ \forall \ ac \in AC \ \mid ac.c = cc.c_1 \lor ac.c = cc.c_2\} \qquad (6)$$

An interface that is controlled by a single actor (that is, $|A(cc)| = 1$) does not pose any coordination overhead. Any larger set, however, will increase it. We propose to combine both metrics into a Collaborative Interface Risk Index (CIRI):

$$CIRI(cc) = \Delta(cc) \times [|A(cc)| - 1] \qquad (7)$$

Completely mature interfaces with $cc.IRL = IRL_{100}$ lead to zero risk, as do interfaces that are only controlled by a single actor. The metric will identify very immature interfaces that require many actors to collaborate, as displayed in Figure 4. Plotting the CIRI value for each interface in the product DSM should highlight risks associated with interface control in collaborative systems engineering. We use a traffic light color scheme to highlight relative interface risks: red for the highest and green for the lowest CIRI in the model. Shades of orange fill the spectrum in between these extremes.

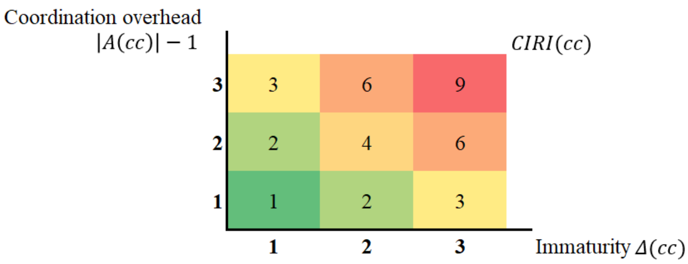

Figure 4. The Collaborative Interface Risk Index ($CIRI$) increases with interface immaturity and coordination effort. We visualize the $CIRI$ of each interface in a DSM with the colors of a traffic light.

3.3 Communication analysis

To include the organization (actors) domain in our DSM model, we follow the approach by Yang et al. (2015). They have demonstrated how to derive communication channels between actors from technical dependencies between components. They presume that a strong or critical interface between two components with different actors will require intense communication between those actors, introducing the concept of technical communication strength (TCS). We apply their method to the immaturity of each interface, rather than to dependency strength. It is important to realize that this conceptual difference will cause the TCS between actors to decrease over time, as interfaces are detailed out. Formally, we define the technical communication strength of each communication channel $aa = (a_1, a_2)$ as:

$$TCS(a_1, a_2) = \sum \{\Delta(cc) \ \forall \ cc \in CC \ \mid \{a_1, a_2\} \subseteq A(cc)\} \qquad (8)$$

$$AA = \left\{ (a_1, a_2) \ \forall \ a_1, a_2 \in A \ \middle| \ \begin{array}{l} a_1 \neq a_2 \\ TCS(a_1, a_2) > 0 \end{array} \right\} \qquad (9)$$

Equation (8) computes the technical communication strength between actors a_1 and a_2 by finding all interfaces that connect any of the components designed by a_1 to any of the components designed by a_2. The TCS is then equal to the sum of the immaturity of those interfaces. Equation (9) defines the set of communication channels between actors as those with a positive TCS. Figure 5 visualizes this pattern:

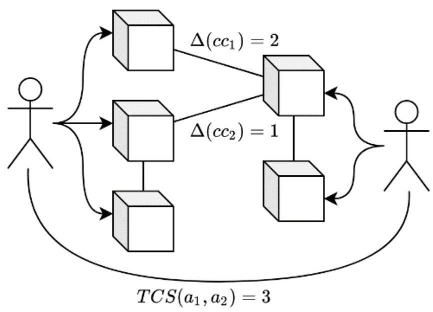

Figure 5. Two actors should collaborate to define interfaces cc_1 and cc_2. The technical communication strength of edge aa is equal to the total interface immaturity.

The identified technical dependency strength of each communication channel can now be plotted in the organization DSM. This DSM highlights dominant inter-organizational patterns that are required for successful system integration.

3.4 Progress monitoring

Project management often uses indicators to monitor and coordinate various processes. We reiterate that the beginning of collaborative interface management is marked by the release of an initial system architecture where all interfaces are at least $IRL_0 = 2$. The process is concluded only when all interfaces have achieved $IRL_{100} = 5$. These bounds allow us to define a progress indicator PI as a function of the set of interfaces:

$$PI = \frac{1}{|CC|} \times \frac{\sum_{cc} \Delta(cc)}{IRL_{100} - IRL_0} \quad (10)$$

where $|CC|$ is the total of interfaces and $\Delta(cc)$ the immaturity of each interface, see Equation (5). As such, PI ranges between 0 and 100 %. It is important to realize, however, that this is only an *indication* of progress. Interfaces may need updates while they remain on the same IRL, or new interfaces may appear because of unforeseen architectural design changes (Beernaert et al., 2021). Such effects are not captured in the proposed PI.

4 Demonstration

We demonstrate the method to support the Visible Spectroscopy Reference System (VSRS) project, an optical subsystem of ITER. The VSRS is a single line-of-sight spectroscopy diagnostic, with main purpose to measure visible Bremsstrahlung in the core of the plasma. This signal depends on the effective ion charge for impurity contents. The system also measures line emissions in the plasma periphery and provides a reference measurement for core charge exchange spectroscopy. It is one of the first systems to issue a warning when the conditions of the nuclear fusion plasma are out of operational bounds and damage to the machine is imminent. As many of ITER's subsystems, the VSRS implements exotic technologies and materials, and its design involves highly specialized scientific analyses (Kajita et al., 2019; Ushakov et al., 2020). Our goal is to improve the organizational communication by identifying critical interfaces that are managed by multiple parties.

The components of the VSRS are spatially distributed in two groups: the 'front-end' components are located close to the plasma, the heart of ITER, whereas the 'back-end' components can be found up to 100 meters away in the nearby diagnostic building. The front-end mainly comprises optical components - mirrors, windows and fibers - and motion actuators. It is a major design challenge to ensure optical performance over a 20-year maintenance-free lifetime in a hostile environment characterized by a combination of ultra-low vacuum, extreme neutron and gamma radiation, and powerful magnetic fields. To minimize risks, the architecture of the front-end system is kept fairly simple and robust.

In contrast, the VSRS features many entangled back-end components that are located in a safe, accessible and controlled environment. These include three integrated hardware cubicles and a heavy optical table with diverse pieces of equipment. A polychromator and spectrometers detect light emissions, while cameras, calibration and control equipment provide auxiliary services. Components are highly connected through optical fibers, sensitive measurement wires, signal and power cables or digital communication interfaces. The back-end system of the VSRS is the subject of this demonstration.

There are six actors that work on the various components; some are even collaborating on a single component! These actors are organizations in the sense that they are groups of people with high internal cohesion. They each have a specific contribution to the VSRS project and are geographically distributed over Europe, as presented in Table 3.

Table 3. The actors involved in the ITER VSRS project.

Who?	**What?**	**Where?**
Actor 1	Instrumentation and control systems (hardware and software)	Slovenia
Actor 2	Optical routing and measurement systems	The Netherlands
Actor 3	Opto-mechanical structures	Portugal
Actor 4	Plasma physics modelling for realtime data analysis	Germany
Actor 5	Mirror cleaning (Marot et al., 2021; Shigin et al., 2021)	France
Actor 6	Polychromator, the primary measurement instrument	The Netherlands

It is clear that we need to harmonize not only the components of the VSRS, but also the different organizations. Therefore, we have built the model as presented in Section 3 in the early stages of the engineering process. It was continuously updated to reflect architectural changes or maturing interfaces. 34 Components and 83 functional interfaces have been specified following the ESL format. The three keywords introduced in Section 3 were used to specify the IRL, to add the interface description, and to specify which actor is involved with which component. Subsequently the network model and MDM display with the CIRI and TCS values in the component DSM and actor DSM, respectively, were automatically generated from the specifications using the ESL toolset (Ratio, 2022). The result is visualized in Figure 6. The generated MDM also has interactive features displaying information that goes with the entries of the matrix. The interactive MDM provides a quick and intuitive overview of the collaborative engineering risks for system integration.

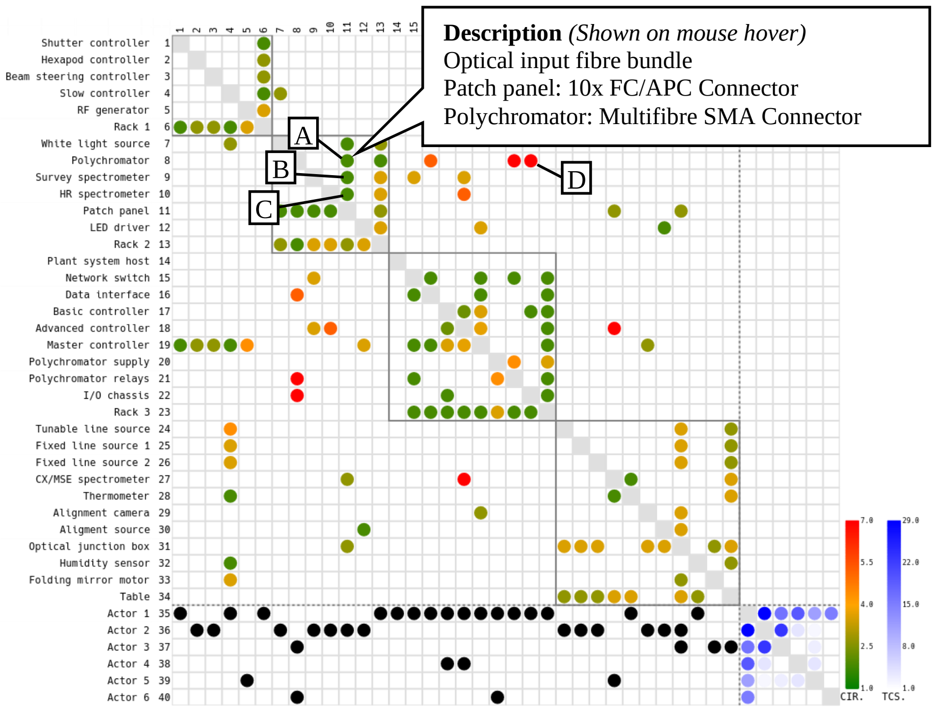

Figure 6. Multi domain matrix of the VSRS project with top-left: Collaborative Interface Risk Index (CIRI), bottom-left: responsibility of collaboration partners, and bottom-right: required Technical Communication Strength (TCS). At this moment, the interface definition is 49.8 % complete (Equation 10).

The product DSM (top-left 34x34 matrix) visualizes the component interfaces with their respective CIRI values. Red or dark orange indicate interfaces at high risk. This helps the system architects to evaluate integration risks by highlighting interfaces that need collaboration. The interactive version of the DSM shows the description field when the user hovers the mouse over the respective interface. Table 4 shows the four interfaces that were defined in the ESL excerpt of Figure 3. We observe the four typical cases that lie on the corners of Figure 4, of which interface D is identified as highly critical.

Table 4. Four characteristic interfaces in the VSRS system architecture. A combination of high immaturity $\Delta(cc)$ and many actors $|A(cc)|$ leads to integration risks.

cc	$\Delta(cc)$	$\lvert A(cc)\rvert - 1$	Interpretation
A	0	2	Many actors, but interface well-developed: no coordination
B	0	0	Single actor, and well-developed interface: no coordination
C	3	0	Underdeveloped interface, but for single actor: no coordination
D	3	2	Underdeveloped interface, concerns three actors: Risk!

The bottom-left 6x34 Domain Mapping Matrix (DMM) shows which actor is responsible for which component. Although the system architect explicitly models these relations, the visual representation together with the component and actor DSMs may provide for valuable insights. For example, the DMM gives an impression of the distribution of workload among the actors. Empty rows or columns may indicate that the model is not yet complete or that design work has not yet been assigned to an actor. This overview addresses some typical problems in collaborative engineering: confusion about who does what, and work that falls between the cracks. In this case, we observe that Actor 1 and Actor 2 share the majority of the work. Actors 3-6 have a smaller scope on the system and are therefore expected to be less involved in technical coordination.

The bottom-right 6x6 organization DSM shows the organizational pattern that is required to arrive at an integrated design. Recall that the dependency value representing the intensity of communication between two actors is calculated by the technical communication strength, following the amount of interfaces that need to be aligned between those two actors (see Section 3.3). It seems that Actor 1 should play a central role in the organization, as they have high TCS with practically every other actor. Actors 1-3 form a highly dependent cluster, so we must facilitate this collaboration. One way to do this is to plan a series of dedicated meetings to discuss these interfaces. The matrix also shows that it is not essential for Actors 4-6 to attend, as their work is rather disconnected. Perhaps some clear 1-on-1 calls or e-mails with Actor 1 would suffice.

5 Conclusion

In high-tech large-scale systems engineering, it has become necessary to outsource detailed design activities to external specialized contractors. In the conceptual design phase, the system architect defines the system architecture - the functionality and arrangement of subsystems. Interfaces between subsystems are identified, but only at a preliminary maturity. The actual subsystem design work is carried out by the contractors in the detailed design phase that follows, where system architects have to further detail out interfaces in accordance with the contractor's designs.

We have presented a novel Design Structure Matrix (DSM) method to support system architects in the detailed design phase of collaborative engineering projects. Our method extends the Elephant Specification Language (ESL), a structured language for the specification of system architectures, with Integration Readiness Levels (IRL) and collaborating actors of a distributed design organization. The result is a Multi-Domain Matrix (MDM) model that contributes to current interface management methods by formalizing two significant aspects of collaborative engineering.

Firstly, we systematically highlight interfaces based on a newly proposed Collaborative Interface Risk Index (CIRI). CIRI identifies interfaces that pose integration risks as those that both have an insufficient IRL and require coordination among multiple actors. Projecting the CIRI on the product DSM through traffic light colors shows where technical inconsistencies in the system architecture may be expected. Secondly, we project the insufficient IRL's to the organization domain of the MDM. This domain displays the inter-organizational technical communication that is needed to fully define all interfaces. The technical communication strength between two actors in the organization DSM is represented by the number of interfaces that need to be matured through collaboration between the two respective actors.

The method is demonstrated in the context of ITER, one of the largest international collaboration projects that pursues sustainable energy generation through the development of an experimental nuclear fusion reactor. We present the system architecture of the VSRS back-end, an optical diagnostic system of ITER, and the contractors that work on its various subsystems. The ultimate result is a MDM with 34 components and 6 actors. We conclude from the demonstration that the method is highly useful for collaborative engineering projects. The matrix visualization gives powerful insight in the complexity and risks, and the model-based workflow makes for a flexible approach. The textual ESL specifications are easy to edit and process, so that any architectural changes due to unanticipated problems would automatically lead to an updated MDM (Beernaert et al., 2021).

The ideas presented in this paper could also contribute to other aspects of collaborative systems engineering. If we had the freedom to distribute responsibilities among actors, how would we do it? One could identify modules in the system architecture through a clustering analysis and assign each module to a single actor. We also know this principle as Conway's law or the mirror hypothesis (Sosa, 2007). Because each module would have many internal and few external interfaces, it is easy to predict the outcome of our risk analyses. Applying the mirror hypothesis to structure an organization will reduce the CIRI *inside* a module; however, interface risks *between* two modules will remain. This verifies the mirror hypothesis as a coordination principle, because it searches to minimize the CIRI in the system architecture.

Now, there is an obvious question: Why was the VSRS project not following the mirror hypothesis? We note that specialization is an important reason for distributed collaboration projects. Current engineering efforts require a set of technical knowledge and capabilities that is both immensely broad and deep. One may have to search halfway over the globe to find the actor that can design a specific component. Actors obviously cannot be reassigned to components that do not match their competences. Here we bump into the limitations of the mirror hypothesis, and leave the reader with an open question: How can we strategically structure tomorrow's collaboration projects under the constraints of hyperspecialization, an ever-increasing trend in high-tech systems development?

References

Beernaert, T., Etman, P., De Bock, M., Classen, I., De Baar, M., 2021. Tracing the Emergence of Design Problems and their Impacts on the Complexity of Engineering Solutions. Proc. Des. Soc. 1, 3229–3238. https://doi.org/10.1017/pds.2021.584

Browning, T.R., 2016. Design Structure Matrix Extensions and Innovations: A Survey and New Opportunities. IEEE Transactions on Engineering Management 63, 27–52. https://doi.org/10.1109/TEM.2015.2491283

Cabigiosu, A., Camuffo, A., 2012. Beyond the "Mirroring" Hypothesis: Product Modularity and Interorganizational Relations in the Air Conditioning Industry. Organization Science 23, 686–703. https://doi.org/10.1287/orsc.1110.0655

Colfer, L.J., Baldwin, C.Y., 2016. The Mirroring Hypothesis: Theory, Evidence and Exceptions. Industrial and Corporate Change 25, 709–738. https://doi.org/10.1093/icc/dtw027

Donné, A.J.H., 2019. The European roadmap towards fusion electricity. Phil. Trans. R. Soc. A. 377, 20170432. https://doi.org/10.1098/rsta.2017.0432

Entler, S., Horacek, J., Dlouhy, T., Dostal, V., 2018. Approximation of the economy of fusion energy. Energy 152, 489–497. https://doi.org/10.1016/j.energy.2018.03.130

Eppinger, S.D., Browning, T.R., 2012. Design Structure Matrix Methods and Applications, 1st ed. MIT Press.

Kajita, S., De Bock, M., Desjardins, M., Barnsley, R., 2019. Ray Trace Study for Visible Spectroscopy Reference System (VSRS) Diagnostics in ITER. Plasma and Fusion Research 14, 1405042–1405042. https://doi.org/10.1585/pfr.14.1405042

Kherbachi, S., Yang, Q., Khan, S.Z., 2020. A structured approach to measuring and optimizing the organizational architecture in global product development projects. Concurrent Engineering 28, 161–174. https://doi.org/10.1177/1063293X20929388

Marot, L., Moser, L., Steiner, R., Erni, W., Steinacher, M., Dine, S., Porosnicu, C., Lungu, C.P., Soni, K., Antunes, R., Le Guern, F., Piqueras, J., Meyer, E., 2021. RF discharge mirror cleaning for ITER optical diagnostics using 60 MHz very high frequency. Fusion Engineering and Design 163, 112140. https://doi.org/10.1016/j.fusengdes.2020.112140

Nicquevert, B., Boujut, J.-F., 2022. Modelling engineering interfaces in big science collaborations at CERN: an interaction-based model. Res Eng Design 33, 87–109. https://doi.org/10.1007/s00163-021-00375-8

Paulson, B.C., 1976. Designing to Reduce Construction Costs. Journal of the Construction Division 587–592.

Ratio, 2022. Model-based systems engineering, system architecting, requirements engineering. URL https://ratio-case.nl/

Sauser, B., Gove, R., Forbes, E., Ramirez-Marquez, J.E., 2010. Integration maturity metrics: Development of an integration readiness level. Information Knowledge Systems Management 9, 17–46. https://doi.org/10.3233/IKS-2010-0133

Shigin, P., Babinov, N., De Temmerman, G., Danisi, A., Dmitriev, A., Larsen, J., Madsen, R., Marot, L., Moser, L., Mukhin, E., Kochergin, M., Ortiz, R., Razdobarin, A., Reichle, R., Pitts, R., Samsonov, D., Tsalas, M., Udintsev, V., Vayakis, G., Walsh, M., 2021. RF discharge mirror cleaning system development for ITER diagnostics. Fusion Engineering and Design 164, 112162. https://doi.org/10.1016/j.fusengdes.2020.112162

Sosa, M.E., 2007. Aligning Process, Product, and Organizational Architectures in Software Development. Presented at the International Conference on Engineering Design '07, Cite des Sciences et de l'industrie, Paris, p. 12.

Ushakov, A., Verlaan, A., Stephan, U., Steinke, O., de Bock, M., Maniscalco, M.P., Verhoeff, P., 2020. ITER visible spectroscopy reference system first mirror plasma cleaning in radio-frequency gas discharge – circuit design and plasma effects. Fusion Engineering and Design 154, 111546. https://doi.org/10.1016/j.fusengdes.2020.111546

Wilschut, T., Etman, L.F.P., Rooda, J.E., Vogel, J.A., 2018a. Generation of a function-component-parameter Multi-Domain Matrix from structured textual function specifications. Res Eng Design (2018) 29, 531–546. https://doi.org/10.1007/s00163-018-0284-9

Wilschut, T., Etman, L.F.P., Rooda, J.E., Vogel, J.A., 2018b. Multi-level function specification and architecture analysis using ESL: A lock renovation pilot study, in: Proceedings of the ASME 2018 International Design Engineering Technical Conferences and Computers and Information in Engineering Conference. Presented at the IDETC/CIE 2018, Quebec, Canada.

Wilschut, T., 2018. System specification and design structuring methods for a lock product platform. PhD Thesis, Eindhoven University of Technology.

Yang, Q., Kherbachi, S., Hong, Y.S., Shan, C., 2015. Identifying and managing coordination complexity in global product development project. International Journal of Project Management 33, 1464–1475. https://doi.org/10.1016/j.ijproman.2015.06.011

Yasseri, S., Bahai, H., 2018. System Readiness Level Estimation of Oil and Gas Production Systems. ijcoe 2, 31–44. https://doi.org/10.29252/ijcoe.2.2.31

Yasseri, S.F., Bahai, H., 2019. Interface and integration management for FPSOs. Ocean Engineering 191, 106441. https://doi.org/10.1016/j.oceaneng.2019.106441

Zouggar, S., Zolghadri, M., Girard, P., 2009. Modelling Product and Partners Network Architectures to Identify Hidden Dependencies, in: Proceedings of 19th CIRP Design Conference - Competitive Design. p. 32.

Authors

Torben Beernaert is a PhD student in Systems Engineering at DIFFER, focusing on model-based techniques to support the development of nuclear fusion reactors.

Pascal Etman is an Associate Professor of Mechanical Engineering in the Control Systems Technology group at Eindhoven University of Technology (TU/e).

Maarten de Bock is Technical Responsible Officer at ITER Organization for Visible Spectroscopy Diagnostic systems.

Ivo Classen is group leader of Plasma Edge Physics and Diagnostics at DIFFER.

Marco De Baar is a specialist in the control of nuclear fusion plasmas. Presently, Marco is the director of the NWO institute DIFFER.

Contact: Torben Beernaert, t.f.beernaert@differ.nl, +31 40 333 4999, De Zaale 20, 5612 AJ Eindhoven, Netherlands

Disclaimer

The views and opinions expressed herein do not necessarily reflect those of the ITER Organization.