WPRM-PR(17) 18354

J Honore et al.

Towards smart infrastructure for fusion power plant maintenance

Preprint of Paper to be submitted for publication in
Automation in Construction

This work has been carried out within the framework of the EUROfusion Consortium and has received funding from the Euratom research and training programme 2014-2018 under grant agreement No 633053. The views and opinions expressed herein do not necessarily reflect those of the European Commission.

Towards smart robotic infrastructure for fusion power plant maintenance

Jean-Jacques Honore*, Guy Burroughes, Rob Skilton

RACE, UKAEA, Culham, UK

Abstract

We face a big challenge to our approach of the design and manufacturing of complex infrastructure over the next hundred years. In the past we have built mechanical structures (wall, bridges, power plants) that have stayed largely intact over the decades/centuries with little intervention. Today we are experiencing a dramatic increase in the advancement of intelligent digital technologies and their rate of obsolescence. To maintain the established levels of performance we expect from our infrastructure, new and innovative approaches must be developed to accommodate continual enhancement and augmentation.

DEMO (Demonstration Fusion Power Plant) will be the world's first nuclear fusion power plant and arguably the world's most ambitious scientific mega-project. It will be a complex undertaking, involving a multitude of design challenges being researched and developed around the world. A key element of DEMO's success will be an efficient, reliable, and forward-thinking remote maintenance strategy, however, current systems have been driven by design pressures towards monolithic like, highly coupled, low cohesion architectures which are hard to maintain and upgrade.

This paper addresses the issue of scalable and sustainable remote maintenance architectures with a emphasis on how to structure complex digital facilities that are manageable over decades of use. An overview and analysis of the current state of the art for fusion reactors and emerging IoT technologies is presented along with recommended areas of potential improvement. The paper concludes with a proposed conceptual methodology for creating a better infrastructure along with discussion of potential future work.

Keywords: Remote Maintenance, Architecture, Internet of Things, autonomous, fusion, power plant, scalable, robotics, automation, Robots as a Service

---

1. Introduction

Globally, there are many ongoing projects with the aim of mastering economically viable and sustainable power. Nuclear fusion is one of the most promising options for generating large amounts of pollutant-free energy. Presently, devices known as 'Tokamaks' are the most developed and well-funded approach. A Tokamak is a device that

---

*Corresponding author

Preprint submitted to Automation in Construction *July 31, 2017*

uses powerful magnetic fields to confine a fusion facilitative plasma within a toroidal vacuum vessel. Following the success at the Joint European Torus (JET), the Tokamak base technologies have reached a maturity where an initial commercial power plant is now thought to be feasible.

Once the scientific and engineering systems have been tested and verified on the upcoming ITER experimental reactor, the next stage will be to integrate the results into a Demonstration Fusion Power Plant (DEMO). DEMOs primary goal is to demonstrate the possibility of creating a Fusion Power Plant (FPP) that is capable of supplying a comparable electrical output to that of a standard power plant, and aims to do so by 2050 [1]. If successful, it may lead to the first generation of commercial fusion power stations.

Current concept designs for DEMO are large and complex. The construction of such a power plant will be the culmination of half a century of research and technological advancement. High energy, near vacuums, high pressures, strong magnetic fields, near absolute zero temperature fluids cooling superconducting magnets, temperatures an order of magnitude greater than that of the sun, high powered electronics, volatile materials, and radioactivity will all be found within a few short meters of each other. Dealing with any one of these issues can be difficult, and is further complicated by the inherent exclusion of direct human intervention.

All DEMO components and systems will have three options for long term supportability:

1. never fail;
2. operate at a human tolerable distance from the hazardous environment, or;
3. be remotely maintained by radiation hardened robotic systems.

As option 1. can rarely be relied on and option 2. is often impractical due to response times and implied infrastructure, Remote Maintenance (RM) strategies are practically mandatory. An example of such a system can be seen in figure 1, the MASCOT servo-manipulator which has been used for over 30,000 hours of remote maintenance in JET.

Another driving factor of the DEMO will be its requirement to demonstrate economic viability of fusion as an energy source. A FPP must be cost effective, in terms of £/GWh, for it to become the power source of the future. One of the main contributing factors towards profitability will be a plants availability. Designs are being developed with the goal of demonstrating the potential of achieving 75% availability or above using a scheduled 24 months power generation cycle accompanied by a 6-month maintenance period [2]. The latter period will require the rapid robotic/remote maintenance of a significant number of components which must be achieved within the allocated time if the power plant is to be financially viable [3]. This will be a challenging task that will extend the limits of involved technologies.

Existing fusion research projects are currently maintained using a mixture of remote handling machines and human intervention. This is possible because of the relatively low activation levels of reactor components. However, this will continue as future experiments will involve higher energies and more active materials. For example, JET will enter a second DT phase in late 2017, at which point there will be no permitted human entry to the torus at least 5 years. This clearly necessitate robotic solutions such as the JET MASCOT, as seen in Figure 1.

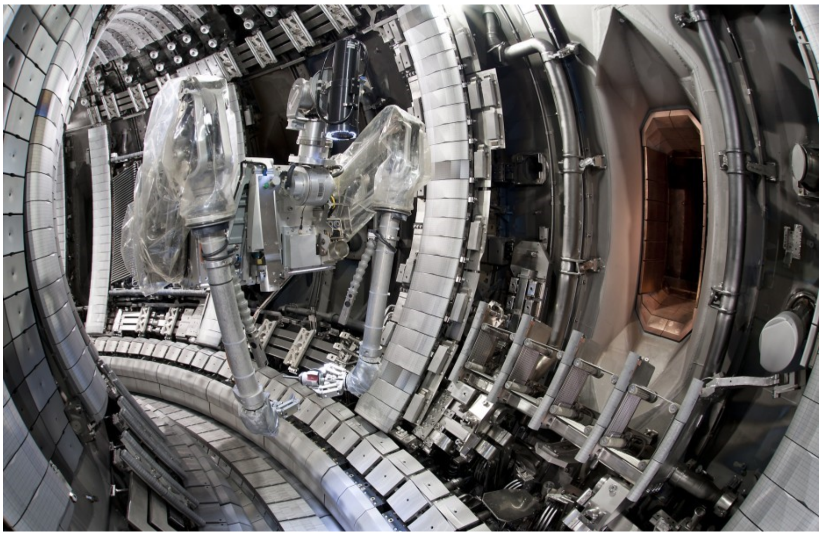

Figure 1: JET's telerobotics system, MASCOT.

Projects like JET and ITER have been focused on scientific research, and thus not subject to design constraints to that a commercial FPP. Their RM strategies tend to favour flexibility and operate on a short funding cycle (i.e. 2-5 years). The short-term decision making and rapidly changing requirements do not facilitate engineering decisions that focus on following good design and architectural practices. Whilst these pressures are necessities for research projects, they do not produce maintainable, future-proof systems. It is estimated that traditionally used design philosophies and architectures will not be adequate for the success of future FPPs.

DEMO's design will be heavily influenced by the development and performance of ITER. Fortunately, DEMO is not purely a research exercise and it will require less flexibility. This should provide a much-needed opportunity to optimise the design for manufacture and maintenance for prolonged periods. The incorporation of standardised items, Line Replaceable Units and Commercial Off the Shelf (COTS) components should help reduce construction costs. Features like pervasive networking and condition monitoring have the potential to reduce down-times and maintenance periods. The presence of a static, unchanging environment should also prove ideal for the introduction of robot automation and autonomous maintenance.

Autonomous robotics will likely provide a cost-effective replacement to traditional human run operations and should play a larger role within industrial environments in the near future. However, there will be significant challenges to overcome, such as creating, maintaining, managing, and interacting with a large autonomous Systems Of Systems. Fortunately, progress is continually being made, especially in key fields like machine learning for service-life assessment [4], computer vision [5], and UAVs [6]. Regrettably, the speed of these advances will add their own design strain as DEMO must contend with a possible 30+ years of rapid obsolescence.

What follows is a discussion of the current state of fusion reactor remote maintenance architectures, and modern architectural designs. Following this, a possible architecture for large system of system project, such as the remote maintenance of future FFPs, is proposed for the purposes of discussion. Then finally, a discussion about the necessary architectural development steps that will be required for FPPs to be financially viable.

2. Background

The nature of the plasma needed to achieve fusion is challenging in many ways. Over time it causes significant wear to reactor components, especially to electronic devices and the parts that line the inner walls known as the blankets and divertor plates. The maintenance of said items can be categorised into two phases: first, a continuous process of reactive maintenance is performed; second, the scheduled shutdown period where the reactor is breached and preventative maintenance is performed. Both of these are time consuming, expensive and risky processes that are the subject of much analysis with regards to optimising component cost to operational life to reliability ratios [7].

The following sections will give a brief overview and analysis of the two predominant examples of tokamak reactors, their remote maintenance systems, and some of the key challenges they face.

2.1. JET remote maintenance (JET RM)

The Joint European Torus (JET) project is the worlds largest active tokamak experiment. It began in the early 70's and achieved its first plasma containment in the early 80's. In these early days of experimentation, the toxicity and radiation levels within the reactor were deemed to be low enough for human involvement and maintenance. However, in the 90's, JET commissioned a sophisticated remote handling system [8] to perform this task as there were plans start tritium based fusion experiments, which would require remote reactor maintenance.

Remote operation has been fully operational since the 90's. The equipment and handling team have gained tens of thousands of hours of in-vessel experience during the biennial shutdown periods [9]. Excluding exceptional circumstances, human entry into the reactor has been practically eliminated.

Unfortunately, this legacy system has naturally deteriorating over time. The upkeep of the equipment is becoming more burdensome and expensive with each passing year. Due to its intertwined and poorly grouped functionalities, replacement or incremental improvement of said components can have far reaching and unexpected implications. In most cases, seemingly small upgrades can only be achieved though the interruption, modification, or replacement of large sections of the whole system.

Figure 2 and 3 shows a comparison between a simplified and idealised JET RM architecture compared to an approximation to the current structure. The idealised hierarchical structure has clear separations and encapsulation of functionality. For example, the viewing system Human Machine Interface (HMI) communicates with modules within a hardware cubicle and the cubicle controls the physical cameras.

The implemented structure has attempted to emulate this; cubicles and HMIs are mostly primed for one type of task. Unfortunately, there are many instances where unrelated equipment and functionalities have been grouped in a seemingly incoherent and inappropriate ways. Functions are often found bridged between multiple HMIs, cubicles and hardware. It is evident that either by design or necessity, the JET RM architecture

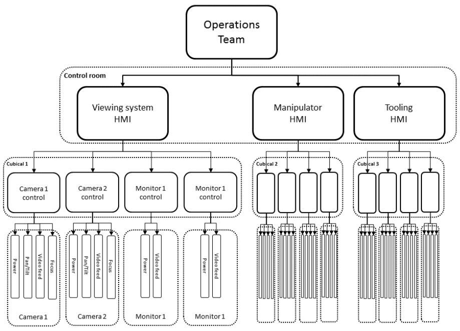

Figure 2: The idealised JET remote maintenance system architecture.

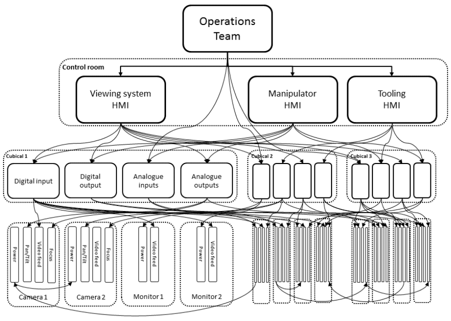

Figure 3: The actual JET remote maintenance system architecture.

is inclined towards what components can do, rather than what they should do. This progression towards ever greater entanglement and untidiness is a direct consequence of short funding cycles and requirement for quickly implemented, low-cost functionality changes. An example of such a situation is where the cubicle primarily responsible for the camera units was adapted to also control the remote welding rig because there was spare capacity. Engineers are frequently forced into situations where the question is: whether they can; rather than whether they should.

Another contributor to the entangling of systems is the lack of standardised communication interfaces. Each system has its own bespoke methods and protocols regarding communication to neighbouring components. Functionality and task responsibility boundaries are frequently blurred. This can make fault diagnosis, component replacement and upgrading a difficult and time-consuming task as every interaction must be carefully examined and tested. Again, this has regularly been the result of extracting extra system performance by passing tasks from its logical host to more capable equipment.

The aforementioned issues create many interesting emergent problems. For instance, the simple task of commissioning the in-vessel welding equipment has become an involved and disruptive activity. Due to its placement within the JET RM system, it requires the disabling of an analogue output controller that is used by an assortment of safety critical cameras. If this non-intuitive dependency is not noticed by the task planner early on, there is likely to be a task conflict that will probably result in a performance compromise. Due to the relative importance of the cameras, this has, in the past caused the delay of welding tool commissioning activities, resulting in significant reduced experimental time and high associated costs.

Repair and maintenance is often stockpiled in this fashion until it reaches a threshold to which it is deemed worth shutting down more critical sections of the system. However, due to a relatively short-term funding cycle, there is also a limit to how big these work packages can become before they are beyond the means of the budget. This generates a hypothetical sweet spot where small tasks are ignored until of a particular size, and not dealt with at all if allowed to become too large. Unfortunately, the bridging, interconnecting and inappropriate coupling is a difficult situation to escape. Due to the entwined state of components, it is difficult to change one thing without causing a cascade. Even adding components without causing a system ripple can prove difficult, and doing so often only reinforces the current state of affairs. The only consistent method of disentanglement is through massive and expensive system overhauls, which will probably not be practical for the remaining operational lifespan of JET.

2.2. ITER remote maintenance (ITER RM)

Currently under construction, ITER is soon to displace JET as the world's largest Tokamak experiment. It will be a little under 10 times the volume of the JET reactor and calculations predict it will produce over 30 times the energy output [1]. Due to the nature of the increased size and power, the complexity and the level of hazards will also increase. Unlike JET, ITER will need a larger, more comprehensive remote maintenance strategy from the start.

A key design challenge associated with ITER is the division of work and the integration of resulting components. Participating, international political powers have divided design, manufacture, and commissioning amongst their respective organisations to encourage things like the development of competencies across many countries. There might be concerns on how smoothly the interfacing of these different work packages will be. ITER is aware of the risk that if improperly managed, there could be a reduction in the overall performance and an increase to the commissioning processing time/cost of the reactor and corresponding remote maintenance facilities.

Fortunately, there are comprehensive standards regarding the system architecture and policies in terms of interfacing, communication and data gathering [10]. Versions have even been implemented on smaller scales though separate projects like the Korean tokamak, K-STAR [11]. The use of common hardware, data packets, data structures and communication networks are documented in detail and will be enforced on all ITER systems. The general philosophy being that by applying a flat, modular topology, matters such as repair, maintenance, and obsolescence management will be made more sustainable.

Figure 4 illustrates a hypothetical component within the ITER environment connecting to a set of globally available networks. A plant system will contain a collection of internal systems and a selection of standardised internal connections. A key feature of this standard is the introduction of a compulsory module to subsystems, known as the

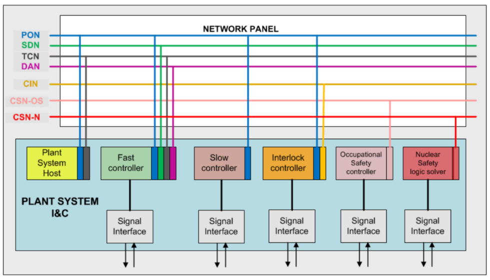

Figure 4: ITER I&C architecture [10].

'Plant System Host' (PSH). This module is used for providing the system with interfaces and functionalities such as health monitoring, state management, maintenance functions and a time referencing.

Unfortunately, due to perceived implementation issues, these standards have only been applied to the ITER remote handling department at the highest level. Instead, remote handling has opted for connecting components in a hierarchical way using private networks, something like JET's RM structure, discouraged within the ITER Plant System I&C Architecture manual. This design choice could prove to be problematic in years to come as the highest levels of the ITER information chain will have poor visibility of the RM equipment status.

There are benefits of having a standard module like the PSH. They are guaranteed to behave in a proscribed manner, make system behaviour easier to support/validate, provide standardised communications interfaces and generate large production numbers that drive costs down. However, it may be impractical to give every small component its own PSH. This inadvertently creates an unwritten minimal module size. Anything below this threshold risks being placed in a group of unrelated systems within the same PSH umbrella. According to the current baseline designs, the remote handling system will consist mostly of small modules.

Another perceived issue with the ITER standards is the separation of network and hardware interfacing. ITER may be saved from inhibiting communication entanglement, only to suffer the same issues as JET with regards to connections. A more holistic representation of how systems interface to one another would be recommended to provide more extensive protection and design transparency.

2.3. Summary

For its time, initial goals, and available budget/hardware, the JET remote maintenance architecture was an appropriate design. Its successful operation for over 20 years is a testament to its design and the diligent efforts of the operations team. However, the same approach would not be economically viable if scaled up to the size or timescales of ITER or DEMO. A more modular approach needs to be adhered to, where functionalities are suitably decoupled to allow continuous, disruption-less modifications to be implemented. These modules should be segregated with regards to their functionality, not by their capabilities, capacities, or physical locality.

ITER is currently attempting a more modular approach that should reduce the prevalence of entangled systems. However, it is speculated that the current methodology used

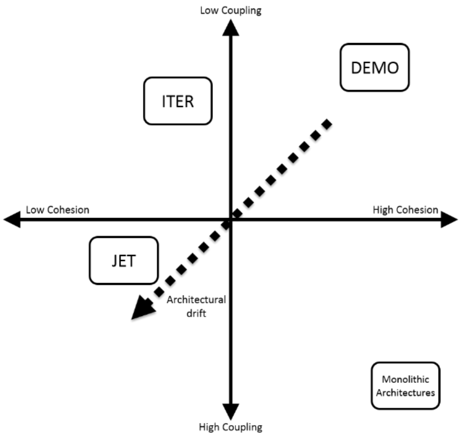

Figure 5: Coupling and cohesion for various systems. The arrow reaching from top right to bottom left, represent the natural progression of systems without tight architectural control.

does not incorporate all forms of system interactions nor does it propagate down to a low enough level as to prevent JET RM like issues from reoccurring. Encapsulating large sections of a system can be beneficial in some circumstances; for instance, if an item is a COTS product or contracted to a third party. Care must be taken, as the internal working of these mechanisms will not be supportable by engineers and technicians. Long term maintenance plans must ether replace the whole module or involve long term contracts with the original developers.

A helpful methodology to summarise the progression of an architecture is via the criteria of coupling and cohesion. Coupling can be defined as the measure of independence between systems/modules; cohesion as the measure of how strongly elements within a system/module relate to one another. Figure 5 gives an interpretation of where JET and ITER are with regards to these terms and where we should be aiming to have DEMO.

Figure 5 also includes vector of architectural drift that represents the effects of the cost cutting on an architecture over time. Even if an adequate architecture is initially implemented, there is a inclination to drift towards stronger coupling and worse cohesion as features are patched onto the system. This highlights the importance of establishing good practices to counterbalance this tendency throughout the entire lifespan of the system.

Essentially, the remote maintenance architecture problem can be summarised as a scalability issue, which encompasses extensibility, maintainability, reliability, obsolescence management, verifiability, and understand-ability (for system maintainers). Some of the consequential aspects that need managing are:

- Visibility, or discoverability of elements in the system;
- Accessibility to the data and functionality of the elements in the system;
- Infrastructure minimisation, e.g. sharing appropriate network infrastructure;
- Big Data, or managing the vast amount data generated by the system;

- Distribution of the elements of the system on to separate computer and resources; which leads to:
    - Synchronisation issues of distributed real-time control systems;
    - Integrity of distributed control systems;
    - Security, or the concerns of both malicious attack and data protection;
- Condition monitoring, or monitoring of a large control system;
- Scheduling, or the temporal and resource management of a large scale remote maintenance plant.

3. The Internet of Things and Remote maintenance facilities or: How I Learned to Stop Worrying and Love Complexity

Internet of Things is a fuzzy term. Attempts have been made to tie down a singular meaning [12], and perhaps separate it from terms like the Industrial Internet of Things and intranet of things, but no pervading definition has yet emerged. This paper chooses to roughly define IoT as: *A culture of integrating embedded intelligences into a distributed range of objects/systems that are all capable of communicating in a mostly indiscriminate manner over a wider area network.* By this definition, IoT outlines a design pattern that inherently has a fine granularity and allows for prevalent status monitoring/control of countless systems. Due to the limitations of the embedded devices, there is also lesser temptation to fit extra, unrelated functionality onto a self-contained module; passively encouraging more manageable partitioning with better cohesion. In theory, all IoT devices should be modular, easily replaceable, and easily upgradeable.

Clearly the main issue that IoT addresses is scale, this is one on of the major issues that must be addressed for future remote maintenance systems. A remote maintenance system is not a primary use case for IoT. However, research is being conducted into how to include high complex, intelligent, or actuating devices into IoT.

One such effort is the Industrial Internet of Things (IIoT), which is being researched as a method for manufacturing more flexible, cost effective, and responsive to changes in customer demands, that could be applied to a remote maintenance facility. However, a major concern surrounding the IIoT is interoperability between devices and machines that function within different protocols and architectures [13]. Several methods have been proposed to cope with the issues of communications, utilising various different middleware based techniques. In particularly, specific issues such as sensing and actuating IIoT devices have been discussed; although it is still an open problem [14].

The consideration of physical infrastructure is another example of an largely ignored issue surrounding most IoT research. There is an implicit expectation that devices will have power and an internet connection, but most literature only focuses on data handling and information flow aspects, not holistic integration. Controlling information is a challenging task, but in an industrial setting, it is not necessarily the primary issue. It is odd that a methodology concerned with building a more detailed picture of its environment would in no way consider its own requirements from or impact upon it [14].

Management of objects is of paramount importance for the development of the IoT, as with architecture for fusion power plant remote maintenance systems. Furthermore, IoT applications require, not only that objects are connected and communicate over a wide number of communication technologies, but also that devices and appliances are remotely managed. The management process is complex, and includes many different actions, such as state management of devices, configuring the device/network, updating firmware/software, recovering from errors, monitoring the device/network, and gathering data and statistics. Standardized device management solutions, such as TR-069, SNMP, and NETCONF, are used for the management of resource-rich devices such as routers and smartphones.

These networks inherit different traits from their respective parent architectural type. This includes cyber-physical system scalability and from cloud computing, the adaptability and self-governance from autonomous decentralised systems [15], fault tolerance and real-time computing from the responsive systems [16], and distributed real-time and embedded (DRE) systems. DRE systems are based on a model driven architecture and model integrated computing[17]. They are applied in situations where application requirements and environmental conditions may vary or not be known priori to run-time and thus mandate an adaptive approach to management of quality-of-service (QoS) to meet key constraints such as end-to-end timeliness. Different DRE middleware systems have been developed by the members of the Distributed Object Computing (DOC) Group [18]. The new generation assurance networks also include many other common features such as trustworthiness and mobility [19, 20].

Managing the Big Data generated by a remote maintenance system is extremely challenging due to the different data properties [21]. Within a remotely maintained facility IoT data will be sampled by a variety of objects and sensors, each having different methods for data representation and semantics. The large number of IoT devices will lead to a rapid expansion in the scale of collected data (petabytes and more). Collected data will often have a timespace relationship (i.e., position and time information) to describe the dynamics of the objects' location. Efficient indexing methods need to be developed to enable the practical use and processing of select data items. Suitable representation schemes are also needed to capture the heterogeneity of objects and meta-data, and to enable their self-description. In addition, interoperability among different data is also important. Approaches introducing an abstraction level may solve them. Ontologies and semantics such as the Web Ontology Language (OWL) look to be promising for future IoT adaption [22].

The other large issue regarding data security and privacy for remote maintenance plants. Security issues are central in IoT as they may occur at various levels, investing technology as well as ethical and privacy issues [23]. To ensure security of data, services and entire IoT system, a series of properties, such as confidentiality, integrity, authentication, authorisation, non-repudiation, availability, and privacy, must be guaranteed [24]. Unfortunately, one of the larger shortcomings of the current IoT culture, as for it use in a remote maintenance plant, is the non-standardised behaviour of IoT devices. Like IoTs formal definition, attempts have been made to create a unified standard, but none hold a significant market share yet. This means that any standard that is adopted will likely become obsolete and will need to be replaced or receive long term support via the creation of translation intermediaries. Hopefully in the future (ideally before DEMOs final inception), a standard will emerge that is as usable and ubiquitous as the USB standard.

Finally, a major consideration for Remote Maintenance and its architecture is it inevitable push towards autonomous systems. For FPP to be financially and technically possible many of the tasked will have performed autonomously. Autonomy will have large consequences on the remote maintenance and its architecture. Primarily, the architecture must support the cooperation required between systems to perform complex autonomous operations, but also must be able to support the verification and validation of such solution in a modular and manageable fashion, i.e. minimising fault propagation and emergent behaviour. IoT has investigated these ideas in the form as robots as a service (RaaS) [25]. RaaS is a cloud computing unit that facilitates the seamless integration of robot and embedded devices into web and cloud computing environment. Fault tolerant design will be critical feature of RaaS, as the single-point-of-failure can easily occur at the interface between the computing- networking system and the physical service. In a typical fault-tolerant system design, the tasks can be scheduled for redundant execution and communication. Even the voting processes can be redundant [20]. However, the result must be voted by a single voting process and sent to a single device. The reason of such a design is that devices are typically expensive and not share-able; however, it might be feasible for remote maintenance.

A noteworthy related piece of research focuses on implementing robotics solution for planetary rovers using a service oriented architecture. It was deemed after testing that service-oriented architecture for robotics had many benefits such as greater scalability and encapsulation; however, it raised concerns in to issues such as service synchronisation and over-rigidity of data structures [26]. Following this work, a service oriented architecture for robotics was used to investigate the possibility of autonomously reconfiguring an autonomous systems architecture and architectural elements as to make it more robust, fault tolerant, and capable of graceful degradation [27].

4. A Possible Architecture

The following is a proposed architecture for a remote maintenance system inspired by the IoT. Issues and benefits are discussed below:

4.1. Common interface proposals

Given the rapidly changing nature of communication standards, it would be counterproductive to define interfaces that mandate the use of specific protocols, data schema, or packet formatting. Instead, the architecture for FPPs should be discussed and designed using high level abstractions of potential communication standards. For the paper, we have specified three categories of interface:

4.1.1. Require interfaces

These interfaces will be mandatory for any system included in an architecture's ecosystem. They should be kept to an absolute minimum as each introduced standard will impose a minor change all devices, drastically increasing the cost of the project as a whole. Required interfaces should only be used where a guaranteed level of functionality is required from the system; such as: basic communications, diagnostics, and safety necessities.

4.1.2. Standard interfaces

There will be many connections, protocols and resources that are not required by all modules, but are frequently used throughout the architectural ecosystem as a whole. To keep down system complexity and discourage the introduction of potentially expensive infrastructure, established, standardised interfaces should always be prioritised over the use of a similar but differing standard. Policies need to be enacted as to keep the pool of available standard interfaces to a minimum. With each introduction, the effectiveness of the standards decreases along with the overall simplicity and uniformity of the architecture.

4.1.3. Bespoke interfaces

It will almost always be impractical and possibly debilitating to limit a large system to a restricted set of interfaces. Allowances for special cases should be planned for and regulated though a bespoke interface introduction procedure. This procedure should discourage the introduction of anything that could easily be replace by one of the standard interfaces. If this is deemed impractical, it should facilitate and enforce the correct amount of planning, documentation, and support for the new infrastructure. Clear, concise, and comprehensible documentation will a key factor for long term system maintainability.

4.2. Standard system module proposal

In order to represent a component's use and conformity to the aforementioned interface standards, a formal representational model was needed. The initial goal of a 'Standard system module' is to force designers into structuring their work in a more uniform manner. If components use clearly implemented and understood interfacing, system integration should be made easier. If a project does not fit into this model, it also should highlight aberrations early in the design process, giving more time for the to be dealt with appropriately.

The secondary outcome of the standard system module will be the production of a detailed and intuitive system map. Indicators of a components general functionality and how it interacts with whole system will be clearly displayed in an assessable way. Providing standards are enforced, this will greatly aid with tasks such as fault finding/fixing and obsolescence management.

The following is first draft expression of a standard system module. Its purpose is to act as an initial starting point for discussion. It is hoped that this design will be expanded and improved upon in the near future.

4.3. The standard module

Figure 6 shows the simplest depiction of a module. On the left are the external interfaces, one block for each of the previously mentioned types. Required and bespoke interfaces have been drawn thinner than the standard interface box to imply that they should be kept to a minimum. On the right, there is the 'functional component' block which represents everything to do with the modules innate functionality. Details of what is contained within the functional component block would best be described with a common topology or schema that could easily be summarised within the diagram or described in full externally. Between the interfaces and function block, is the 'Universal

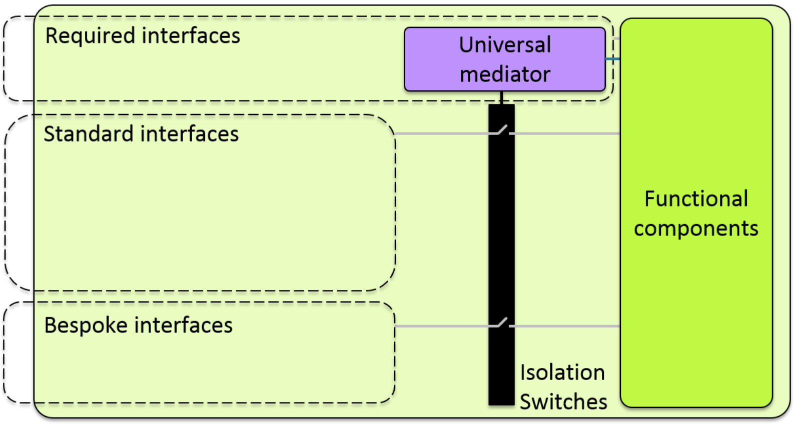

Figure 6: The standard module.

mediator' and 'isolation switches' which fulfils a function similar to ITERs PHS module, but with the potential to directly interact with corresponding hardware.

The universal mediator will make each component within a system 'Smart Device'. It is strategically placed to emphasise that this level of intelligence is required of every module within the global ecosystem and practically an extension of the required interfaces. It is envisioned that the mediator will provide a basic level of self-identification, status reporting, security, and safety functionality. To facilitate these goals, the mediator has been attached to the isolation switches to give it control over what the module's functional components are able to access.

4.4. Simple light bulb module

Figure 7 shows two simple lightbulbs modelled using the proposed method. They represent a simple implementation of a module that you could expect to find within a large system. They contain a set of required interfaces for commands, status and emergency communications, and a standard interface consisting of a 240v power supply. It has been created in such a manner as to highlight some of the issues with regards to physical/data representation relations and autonomy.

Most industrial equipment will require some level of emergency stop switch or other functionality. The traditional hard-wired approach is not going to be an option with untethered autonomous systems. Thus, a highly-regulated degree of on-board intelligence, with the capacity to safely kill the device in the event of a failure, is needed. Figure 7 shows an example of how a wireless smart bulb that is powered, via a battery. Despite the battery arguably being part of the functional component it has been depicted as a shortened interface block that does not extend to the outside of the module. This has been done to: a) highlight to any observers that this module contains a self-powering element, that is not directly controllable from the outside; b) give the on-board dedicated mediator control over it, thus providing safety functionality.

An underdeveloped design choice of this architectural proposal is how to indicate physical interfacing as opposed to electrical or communication. Figure 7 shows a common colour being used for communication interfaces you would find on the same type of

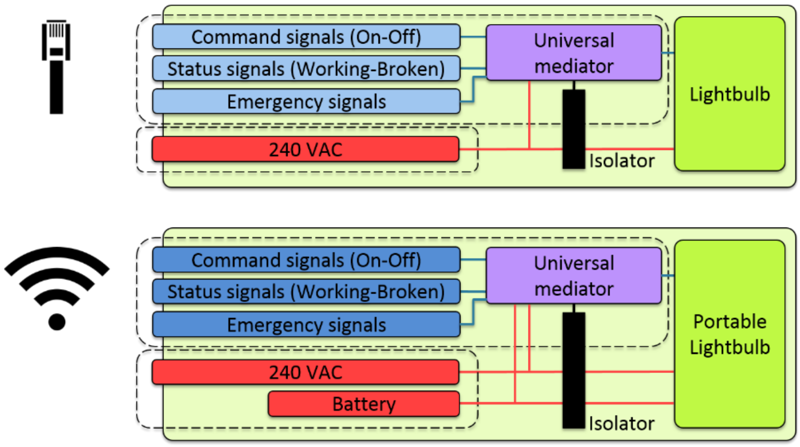

Figure 7: A simple lightbulb example/comparison, where the top module represents a wired system, and the bottom module represents a wireless system.

physical connection. Between the two examples however, interfaces of the same name have been coloured differently. This is to indicate that despite being the same in terms of function, their physical implementation is different (e.g. you could not directly connect one to the other). More exploration needs to be done to discover the best methods to present this multidimensional meta-data.

4.5. A JET-RM-like system like with an IoT inspired architecture.

Figure 8 depicts a JET RM like system and demonstrates how the IoT inspired architecture would be linked together via common interfaces in a real-world example. To the left are the modules for human input and visualisation that you would find in a control room. To the right of Figure 8, there are modules from within the reactor environment such as the actuators and an appropriate power supply. In the centre is a representation of the interface infrastructure. As with the light bulb example, the infrastructure has been colour coded to represent the meta-data relating to a defined physical personification (as explained by the key).

The introduction of a power supply as a separate module raises some interesting points with regards to resource syncing and sourcing. Should part of the interface definition require a direction? When will placing several sources and/or syncs on the same line require managing? If so how much regulation will this require? Should said regulation be performed on a local level or global level? How does this relate to the Power over Ethernet (PoE) standards?

The proposed solution to some of these questions is for the mediator to manage and negotiate the flow of resource amongst themselves. Though some of the infrastructure load/traffic management issues could be achieved by routine negotiation that is carried out via required interfaces connected directly to the mediator, ideally it could maintain isolation between a module and the outside until a suitable level of verification is carried out. However, increasing the responsibility of the mediator may also increase its cost and thus reduce its ability to saturate the global system.

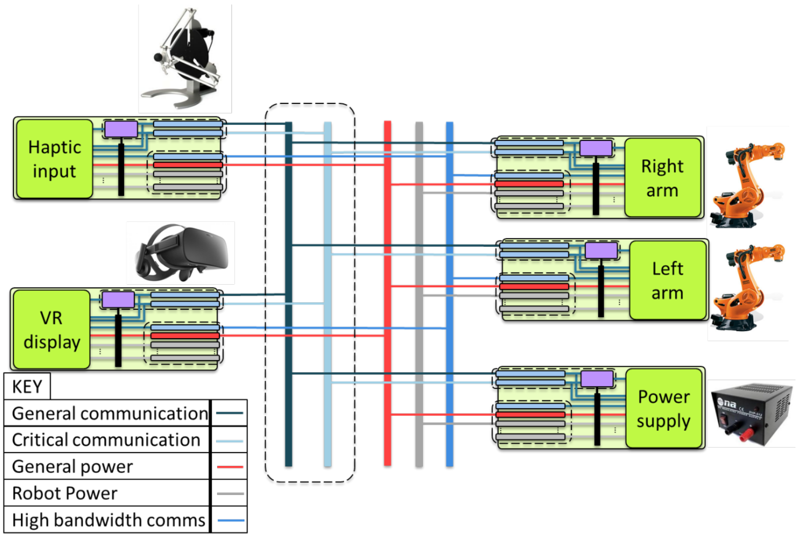

Figure 8: A JET-RM-like system like with an IoT inspired architecture.

For example, if a new device was wired into the system with no regard to the maximum output of the power supply, there would be a risk of overload and the failure of all devices. As an added complication, this would probably occur intermittently during strenuous periods of operations. If correctly utilised, negotiations with a smart power source could be carried out before loading it with the mediator's functional components.

4.6. A DEMO-RM-like system like with an IoT inspired architecture.

Figure 9 shows a nested use case of the proposed architecture. In this instance, there are two modules that depict a smart reactor building and Active Maintenance Facility (AMF). There is also some infrastructure connecting the two to highlight that there will be shared resources at this level of planning. Within the reactor building and AMF's functional components module, there are nested modules. Modelling at this level of the system provides an extra layer of information to the designer/users, encourages more consideration of sub-modules and boosts the encapsulation/reuse of common components across the whole plant.

By displaying a system with nested components, there is an implicit implication of a special relationship between it and the contained modules. The depiction seen in Figure 9 uses this to emphasise the physical presence of systems within the different buildings and what resources each building provides to its inhabitants. An example of such is the power supply within the AMF that is providing the transport system and stores with power.

An issue raised by this example, is the potential for the transference of sub-modules between zones. Mobile items like transport casks will naturally move between locations, and will likely require resources from both. Keeping an updated/real-time model of the global ecosystem would provide a portable system to carry out basic resource availability checks and negotiations before transference, potentially saving time though conflicts and

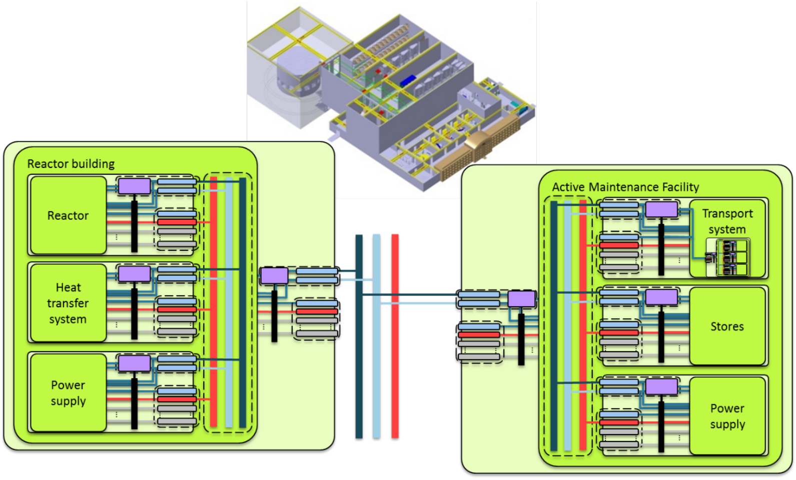

Figure 9: A DEMO-RM-like system like with an IoT inspired architecture.

congestion avoidance. If accurately maintained, it will also benefit the condition/state monitoring and fault detection and diagnosis.

5. Open issues

1. Review of representation choices for describing modules, which would be used to increase discoverability, understandability, and support condition monitoring. Web services and Internet of Things often use knowledge capture languages like WSDL (Web service description language). WSDL is an XML-based interface definition language that is used for describing the functionality offered by a web service [28]. However, as WSDL is designed primarily for use with only software its appropriateness for describing remote maintenance system would need to be investigated. Another known as OWL-s uses an ontology that describes web services. The benefit of using an ontology is that it can be extended to capture enough information to describe systems, perform complex reasoning and allow for autonomous reconfiguration of the architecture [27]. Another distinct feature of the remote maintenance system is the need to understand and describe physical infrastructure. A candidate for this that could be integrated is the Building Information Modelling (BIM, which is a process involving the generation and management of digital representations of physical and functional characteristics of places [29].
2. Formulate a process to govern the division and granularity of module development. The process should aim to maximise re-usability, cohesion, understandability, and minimise coupling and complexity. Obviously, this will also involve the definition of these terms in a quantifiable fashion.
3. Develop a process that will allow the extension of the architecture in a manageable fashion. This will require an investigation into balancing the factors of productivity and combating architectural drift.

4. Examination of the optimal level of meta-data detailing. Too much information can be burdensome and impractical, so what data we choose to present/omit should be examined.
5. Exploration of the use of interfaces to represent and highlight important internal components (as demonstrated in Figure 7). This could be a useful tool for designers and maintainers, but it obviously cannot be used for everything. Adequate justifications need to be made for if/when it can be used.
6. Cyber-Security. Clearly, distributing the architecture will expose the safety critical elements in the will have to be exposed to an intranet, and as Stuxnet proved airgap is not enough for security [30]. Thus, security must be seriously investigated for any Internet of Things architecture that can actuate physical objects. One of the main focusses of the IoT research is cyber-security [24], how does this relate to a remote maintenance architecture.
7. What will the level of human involvement be within a FPP and how will they interface to it? Will people only play an executive role, setting key strategic goals, or will they work in tandem? If so, how will data, tasks and status be presented in a human graspable format, and what implications will this have on machine behaviour requirements?
8. How can safety be integrated into the system? Is it necessary to have an architecturally separated safety system in an IoT inspired architecture or is the benefits of reduced complexity, and increased understand-abilty and discover-ability important enough for a safety system to be incorporated?
9. Verification of sub-systems and automated fault propagation analysis. Can a process be developed to guarantee once a single module with a defined interface has been verified and validated to an appropriate degree, it can be automatically verified and validated to work in the wider application of the architecture? This will obviously involve automated analysis of failure mode propagation.
10. Investigate the appropriateness of homogeneous communication framework like DDS for smart factories. Furthermore, it would be noteworthy to investigate Multiplexing analogue and electrical signals down fibre-optics [31].
11. Finally, an IoT inspired architecture will need to be verified for its effectiveness, with a particular focus on testing its maintainability, functionality, and its safety.

6. Conclusion

Since the inception of JET there seems to be a general progression towards more modular systems. The current architectural models employed are great for rapid development, flexibility and shot term funding cycles, but become unmaintainable within a relatively short time span. For a several decade spanning commercial plant, sustainability will become a more significant metric of success compared to earlier designs. As such, a new architectural paradigm will be required.

The research and development of IoT architectures and technologies are showing great promise as a framework for the future FPPs. They address the key issue of scalability while also providing a fine granularity of control and information gathering. IoT related developments may also prove to provide other great benefits like greater support for mass automation, artificial intelligence and obsolescence management. However, there are still many challenges to overcome with regards to standardising these systems and supporting their continued use over long periods of time.

The presented architectural representation provides an introductory example of a framework that could be used to aid in the development, installation, and maintenance of IoT based ecosystem. There are many options and directions that will need to be further explored and proven though analysis and implementation before a final usable architecture can be devised.

Acknowledgements

This work has been carried out within the framework of the EUROfusion Consortium and has received funding from the Euratom research and training programme 2014-2018 under grant agreement No 633053. The views and opinions expressed herein do not necessarily reflect those of the European Commission.

References

[1] F. Romanelli, Fusion Electricity, A Roadmap to the Realization of Fusion Energy, European Fusion Development Agreement, EFDA, 2012.

[2] O. Crofts, J. Harman, Maintenance duration estimate for a demo fusion power plant, based on the efda wp12 pre-conceptual studies, Fusion Engineering and Design 89 (9) (2014) 2383–2387.

[3] D. Maisonnier, I. Cook, P. Sardain, R. Andreani, L. Di Pace, R. Forrest, et al., A conceptual study of commercial fusion power plants.final report of the european fusion power plant conceptual study, efda (05)-27/4/10, revision 1 april 2005, Tech. rep., EFDA-RP-RE-5.0 (2005).

[4] W. Z. Taffese, E. Sistonen, Machine learning for durability and service-life assessment of reinforced concrete structures: Recent advances and future directions, Automation in Construction 77 (2017) 1–14.

[5] H. Hamledari, B. McCabe, S. Davari, Automated computer vision-based detection of components of under-construction indoor partitions, Automation in Construction 74 (2017) 78–94.

[6] B. McCabe, H. Hamledari, A. Shahi, P. Zangeneh, E. R. Azar, Roles, benefits, and challenges of using uavs for indoor smart construction applications, in: Computing in Civil Engineering 2017, American Society of Civil Engineers, 2017, pp. 349–357.

[7] D. Ward, I. Cook, Y. Lechon, R. Saez, The economic viability of fusion power, Fusion engineering and design 75 (2005) 1221–1227.

[8] T. Raimondi, The jet remote maintenance system, Tech. rep., UKAEA, no. IAEA-TECDOC–495. (1989).

[9] A. Rolfe, A perspective on fusion relevant remote handling techniques, Fusion Engineering and Design 82 (15) (2007) 1917–1923.

[10] J. J, Plant control design handbook, Tech. rep., ITER, v7.0 – https://www.iter.org/mach/codac/PlantControlHandbook (2013).

[11] G. Lee, M. Kwon, C. Doh, B. Hong, K. Kim, M. Cho, W. Namkung, C. Chang, Y. Kim, J. Kim, et al., Design and construction of the kstar tokamak, Nuclear Fusion 41 (10) (2001) 1515.

[12] D. R. Roberto Minerva, Abyi Biru, Towards a definition of the internet of things (iot), Tech. rep., IEEE, v1.0 – http://iot.ieee.org/definition.html (2015).

[13] J. Jung, B. Song, K. Watson, T. Usländer, Design of smart factory web services based on the industrial internet of things, in: Proceedings of the 50th Hawaii International Conference on System Sciences, 2017.

[14] M. R. Abdmeziem, D. Tandjaoui, I. Romdhani, Architecting the internet of things: state of the art, in: Robots and Sensor Clouds, Springer, 2016, pp. 55–75.

[15] Z. Du, L. He, Y. Chen, Y. Xiao, P. Gao, T. Wang, Robot cloud: Bridging the power of robotics and cloud computing, Future Generation Computer Systems 74 (2017) 337–348.

[16] K. Mori, Autonomous decentralized system and its strategic approach for research and development, IEICE transactions on information and systems 91 (9) (2008) 2227–2232.

[17] M. Malek, Responsive systems: a marriage between real time and fault tolerance, in: Tests, Diagnosis, Fault Treatment 5th International GI/ITG/GMA Conference Nrnberg, September 2527, 1991 Proceedings, Springer, 1991, pp. 1–17.

[18] C. D. Gill, J. M. Gossett, D. Corman, J. P. Loyall, R. E. Schantz, M. Atighetchi, D. C. Schmidt, Integrated adaptive qos management in middleware: A case study, Real-Time Systems 29 (2-3) (2005) 101–130.

[19] H. Nakagawa, K. Nakamaru, T. Ohta, Y. Kakuda, A hierarchical routing scheme with location information on autonomous clustering for mobile ad hoc networks, in: Autonomous Decentralized Systems, 2009. ISADS'09. International Symposium on, IEEE, 2009, pp. 1–6.

[20] Y. Chen, A service scheduler in a trustworthy system, in: Proceedings of the 37th annual symposium on Simulation, IEEE Computer Society, 2004, p. 89.

[21] M. Chen, S. Mao, Y. Liu, Big data: A survey, Mobile Networks and Applications 19 (2) (2014) 171–209.

[22] S. De, P. Barnaghi, M. Bauer, S. Meissner, Service modelling for the internet of things, in: Computer Science and Information Systems (FedCSIS), 2011 Federated Conference on, IEEE, 2011, pp. 949–955.

[23] T. Xu, J. B. Wendt, M. Potkonjak, Security of iot systems: Design challenges and opportunities, in: Proceedings of the 2014 IEEE/ACM International Conference on Computer-Aided Design, IEEE Press, 2014, pp. 417–423.

[24] T. Heer, O. Garcia-Morchon, R. Hummen, S. L. Keoh, S. S. Kumar, K. Wehrle, Security challenges in the ip-based internet of things, Wireless Personal Communications 61 (3) (2011) 527–542.

[25] Y. Chen, H. Hu, Internet of intelligent things and robot as a service, Simulation Modelling Practice and Theory 34 (2013) 159–171.

[26] L. Flückiger, H. Utz, Service oriented robotic architecture for space robotics: Design, testing, and lessons learned, Journal of Field Robotics 31 (1) (2014) 176–191.

[27] G. Burroughes, Y. Gao, Ontology-based self-reconfiguring guidance, navigation, and control for planetary rovers, Journal of Aerospace Information Systems.

[28] S. Weerawarana, F. Curbera, F. Leymann, T. Storey, D. F. Ferguson, Web services platform architecture: SOAP, WSDL, WS-policy, WS-addressing, WS-BPEL, WS-reliable messaging and more, Prentice Hall PTR, 2005.

[29] S. Zhang, J. Teizer, J.-K. Lee, C. M. Eastman, M. Venugopal, Building information modeling (bim) and safety: Automatic safety checking of construction models and schedules, Automation in Construction 29 (2013) 183–195.

[30] J. P. Farwell, R. Rohozinski, Stuxnet and the future of cyber war, Survival 53 (1) (2011) 23–40.

[31] J. Yan, J. Wang, Y. Lu, J. Jiang, H. Wan, Novel wireless sensor system based on power-over-fiber technique, in: Optical Communications and Networks (ICOCN), 2015 14th International Conference on, IEEE, 2015, pp. 1–3.