The Internet of Simulation, a Specialisation of the Internet of Things with Simulation and Workflow as a Service (SIM/WFaaS)

D. W. McKee*, S. J. Clement*, Xue Ouyang*, Jie Xu*, Richard Romano†, John Davies*‡
*School of Computing, †Institute for Transport Studies, ‡INCOSE UK
University of Leeds
Leeds, UK
{D.W.McKee, S.J.Clement, scxo, J.Xu, R.Romano, J.K.Davies}@leeds.ac.uk

***Abstract*—A trend seen in many industries is the increasing reliance on modelling and simulation to facilitate design, decision making and training. Previously, these models would operate in isolation but now there is a growing need to integrate and connect simulations together for co-simulation. In addition, the 21st century has seen the expansion of the Internet of Things (IoT) enabling the interconnectivity of *smart* devices across the Internet. In this paper we propose that an important, and often overlooked, domain of IoT is that of modelling and simulation. Expanding IoT to encompass interconnected simulations enables the potential for an Internet of Simulation (IoS) whereby models and simulations are exposed to the wider internet and can be accessed on an "as-a-service" basis. The proposed IoS would need to manage simulation across heterogeneous infrastructures; temporal and causal aspects of simulations; as well as variations in data structures. Via the proposed Simulation as a Service (SIMaaS) and Workflow as a Service (WFaaS) constructs in IoS, highly complex simulation integration could be performed automatically, resulting in high fidelity system level simulations. Additionally, the potential for faster than real-time simulation afforded by IoS opens the possibility of connecting IoS to existing IoT infrastructure via a real-time bridge to facilitate decision making based on live data.**

***Index Terms*—IoT, IoE, IoS, Simulation, Cloud, SOA, Real-Time, Services, Workflow, Modelling, M&S, SIMaaS, WFaaS**

1. Introduction

The Internet of Things (IoT) is becoming one of the central paradigms of 21st century IT with evermore mobility and interconnectivity [1]. It opens doors to new industries and can expand the capabilities and improve efficiencies of existing organisations. However, IoT is a general paradigm with many different approaches in different domains. Therefore we propose the concept of the Internet of Simulation (IoS) to enhance IoT to be capable of facilitating integration of modelling, simulation, and prototyping services across industries.

IoT is the endeavour to facilitate the connection of Everything (IoE) and Anything (IoA) via the Internet [2]. IoT is however specifically focussed on the integration of *smart* devices in a cyberphysical world where each element can be regarded as a relatively small specific component or system. It is therefore conceptually unable to consider the complexities of integrating simulations and therefore be useful for the purposes of system design, prototyping, and analysis by industry.

Both large and small enterprises' are increasingly utilising simulation as part of daily business. It is used for applications such as supply chain management, product design and analysis and predicting human behaviour in cities. The popularity of high performance computing (HPC) in simulation has lead to hugely detailed and complex simulations. As industries strive to improve efficiencies, simulation invariably becomes central to their strategy, virtual prototyping has become part of the solution to reducing costs of product development [3].

In the world of manufacturing, individual system components (often integrated at a physical level) are represented by domain specific simulations created and governed by specific stakeholders. The requirements of individual components are often conflict. Therefore any virtual integration of simulations must take into account the combination of system level behaviours as well as accounting for the disparate domains of simulation. Further with the rise of Industry 4.0 [4] for smart data-centric manufacturing the integration of simulation, manufacturing systems, and IoT is becoming vital. IoS must facilitate the process of design and verification whilst breaking the tight integration between simulations and their execution environment [5].

If adopted the Internet of Simulation has the potential to transform the industrial landscape by facilitating the rapid integration of prototype components across an entire supply chain. This paper introduces the concept of IoS and considers its core features and the potential impact on business.

The remainder of this paper is structured as follows: IoS is described in the context of IoT and Section 3 outlines the current state-of-the-art in Cloud computing as the necessary backbone. Sections 4 and 5 detail how simulations and workflows form the key elements of IoS before an example is outlined in Section 6.

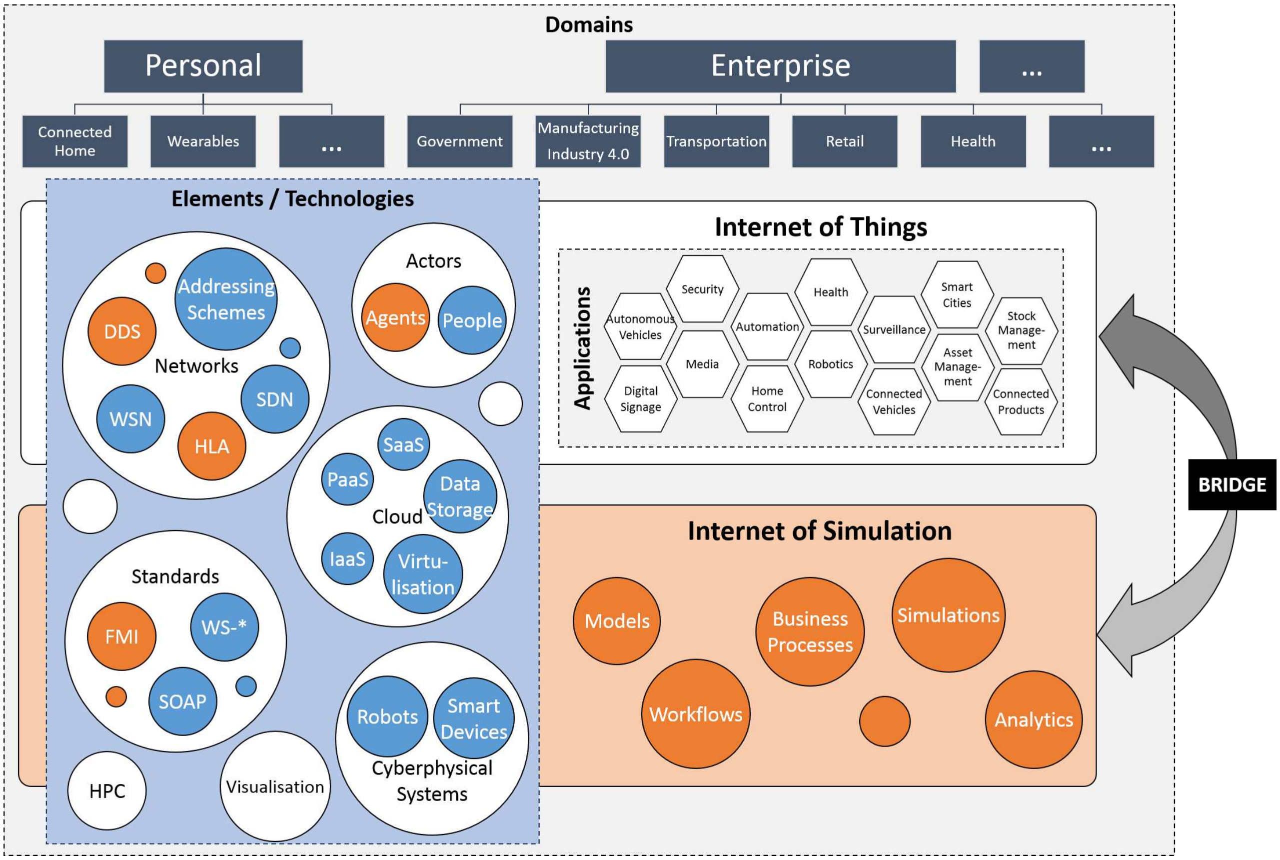

Figure 1. Domains, applications, elements, and technologies of IoT and IoS showing the necessary *bridge* between IoT and IoS

2. The Internet of Simulation and Related Work

The Internet of Things (IoT) as described by Gubbi et al. [6] encapsulates the use of digital technologies to facilitate the interconnection of components, devices, and services at a large-scale across a network. It is distinct from similar concepts such as sensor networks in its inclusion of intelligence, either through context-aware computation or smart connectivity [7]. Since its inception IoT has been extended across multiple domains and now consists a number of technologies which enable various applications, such as smart cities and autonomous vehicles, and features of IoT which depicted in Figure 1. The many domains and applications contained within the concept of IoT each has specific requirements with regards to aspects such as infrastructure, security, interconnectivity and standardisation. Depending on what the *things* are, the specification of IoT can be very different.

Although simulation has been a long standing component in engineering design and manufacturing, only recently has it started to appear as part of IoT and currently focusses on: the simulation of networks and wireless communication protocols between devices [9]; prototyping IoT systems [10]; and Live, Virtual and Constructed (LVC) simulation using Data Distribution Service (DDS) and High Level Architecture (HLA) [11].

The Internet of Simulation (IoS) therefore presents a specialism of IoT that is of particular interest to communities from the domains of manufacturing, government organisations, aerospace and defence [3] where IoS introduces additional elements and technologies to those of IoT. With potential applications in areas such as product design, smart cities and command and control systems. Where the IoT paradigm describes a smart, ubiquitous network of connected *things*, IoS describes a smart network of *simulated things*. These models and simulations may be from multiple domains and applications, they may often be simulations of those *things* that IoT would have connected. By connecting simulations together in the IoS more complex behaviours and systems can be modelled. This is similar to how the functionality of IoT is realised by the connection and interaction of many devices and services. Where previously a complex system model might exist as a single entity and be simulated as such, IoS provides the functionality to separate the simulation into constituent component simulations. This collection of simulations lends itself naturally to a distributed execution. Additionally, since IoS is a specialisation of IoT, these models and simulations are able to be connected to the real world IoT applications via a

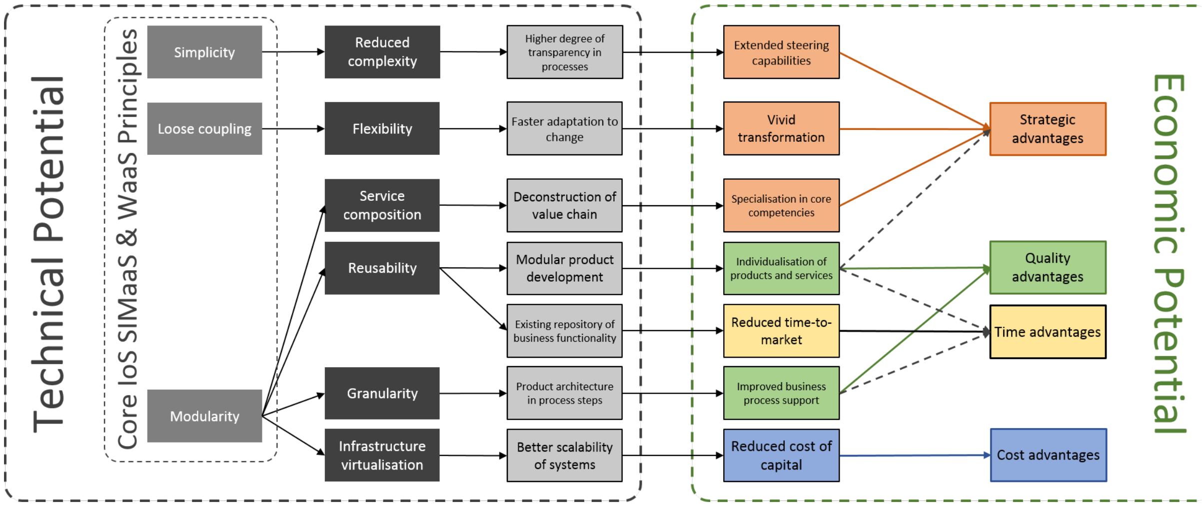

Figure 2. Subset of the economic framework adapted from Mueller et al. [8] showing the economic potential of IoS and specifically SIMaaS and WFaaS

standardised *bridge* (Figure 1). IoS can therefore be defined as:

- *A specialism of the Internet of Things comprised of interconnected virtual system components, agents, or virtual environments defined by cross-domain collections of network-enabled, variable fidelity and heterogeneous models and simulations.*
- *Through composing multiple virtual entities by defining their interactivity a system simulation can be constructed and distributed.*
- *The simulated things contained in the IoS can be connected to the Internet of Things via a Real-Time Bridge.*

The remainder of this section details the industrial and business case for IoS, how simulation and therefore IoS fits into IoT, and finally an example of simulation integration in the context of IoS.

2.1. Business Case for IoS

Internet of Simulation is derived from the evolving needs of global industry, particularly from automotive, aerospace, and defence. It enables the virtualisation of the engineering environment with regards to modelling and simulation [12]. By adapting the framework by Mueller et al. [8] it is possible to map the technical advantages of IoS and specifically SIMaaS onto economic benefits for a company in terms of: cost, strategic, time, and quality dimensions. Figure 2 depicts an example of the business benefits of IoS and shows that business adoption of IoS can result in many business process improvements such as modular product development and higher degrees of transparency. The knock-on effect of these changes has the potential to achieve significant advantages in overall economic potential.

There are several potential benefits of IoS that address key business challenges:

Knowledge sharing within large organisations where there are several Centres of Competency (CoC). A single CoC within an organisation has the expertise for a particular part of the business process or product feature. However, for the organisation to function efficiently as a whole these CoCs must support each other and share their detailed understanding with other areas. In the case of product development this also reduces the duplication of effort through model reuse. Engineers can make their specialist system and component models available to other departments to utilise which can result in a significant reduction in re-engineering costs [8].

Evolving fidelity specifically in the domain of product development where a conceptual idea is modelled at an abstract level and more detail is iteratively added by individual departments until a high fidelity physical prototype is produced. By integrating the entire business process, including mixing fidelity levels within a single product simulation, requirements can be managed more effectively [13], [14]. By facilitating continual integration from the early stages of the engineering life-cycle and simulating the interactions between components, the prohibitive cost of repeated physical prototyping can be mitigated. This connected simulation also enables the identification of emergent properties earlier in the design process and facilitates adjustment of design targets. This also allows the organisation to pivot around market changes by facilitating requirements changes in the business process.

Complex integration relates to the significant challenge of integrating complex models, which often individually expose hundreds or thousands of parameters. This becomes even more complex when considering the temporal aspects of simulations which must be strictly managed and understood by the underlying system in order to guarantee accurate and timely results. Each simulation may be built in a different tool and as discussed in Section 4 these tools may not be compatible. IoS should facilitate not only their one-off integration but also their repeated and re-usable integration. Further, by integrating complex simulations it becomes possible to test more scenarios for emergent behaviours.

Supply chain integration expands the scope of IoS beyond a single organisation. IoS should enable suppliers, or Original Equipment Manufacturers (OEMs), to expose models and simulations of their products as services. This enables those who are integrating components into larger systems to validate designs and verify the behaviour without the physical product in place. This has a benefit for both supplier and customer. By exposing it as a service the suppliers are able to control their intellectual property whilst allowing their customers to experiment with their products. The customer can also have confidence in the provided model as it is generated by the supplier rather than in-house.

Massive-scale simulation is area created by IoS whereby entire collections of integrated cross-domain simulations can be exposed as system, or system of systems simulations. These can be utilised for further testing and analysis, particularly looking at the wider impact systems have on their operating operating environment. This extends the well-known concept of co-simulation by facilitating potentially hundreds of integrated simulations that may be geographically distributed.

Simulation as a Utility refers to the outsourcing of simulation execution. This facilitates organisations to utilise complex simulation without the need for or knowledge of specialist infrastructure. Current trends see many simulation providers already facilitating this aspect of IoS.

Integration with wider IoT enables the world of simulation to tap into the real world and vice versa. Specifically this enables simulations to be fed with live data from IoT for the purpose of validation and verification. Conversely, IoT is enhanced by IoS with intelligence and data analysis whereby *"faster than real-time"* simulation can be performed using real-time data from IoT to predict the future on a continuously updating basis, potentially minutes, hours, days, or weeks ahead. The simulations could consider several possible branches of the future and be used for real world decision making.

The industrial need is clear in the context of autonomous vehicles, smart cities, financial markets, weather, and climate prediction which require the integration of physical elements from the real world to inform, and in some cases be informed by simulations and analysis from relevant industries. There is also a clear need in the area of LVC simulations where physical simulators interact with wider virtual systems.

Further applications which would benefit from an integration of IoS and IoT are highlighted in Figure 1 and in the context of Smart Cities could be expanded to include modelling of people flow, transport links, including roads and rail; utilities such as electricity and gas supplies; as well as health related topics such as ensuring upkeep of facilities and management of epidemics.

2.2. Abstract IoS Architecture

In terms of IoS the elements to be considered are: environment models and models of physical objects which incorporates their: physical structure, control systems, sensors, and actuators.

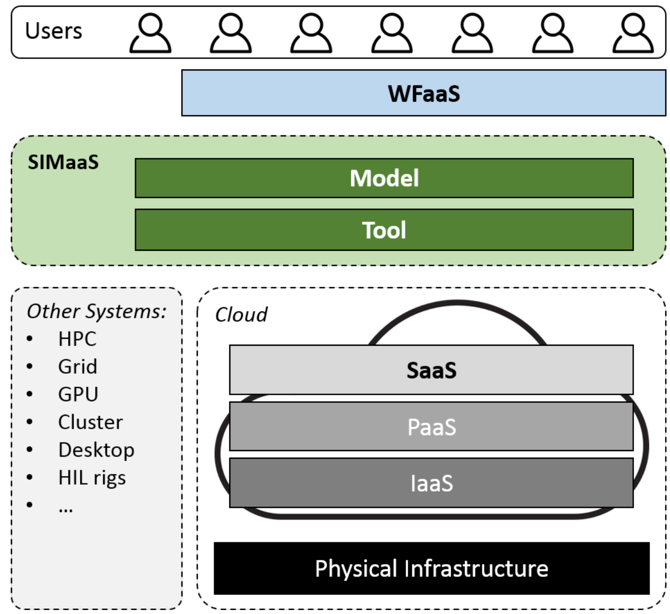

Figure 3. Layers of IoS in the Cloud with SIMaaS and WFaaS layers

Figure 3 shows the organisation of the core concepts of IoS: Simulation as a Service (SIMaaS) and Workflow as a Service (WFaaS), operating on a Cloud backbone. In IoS the cloud consists of three layers on top of the physical hardware: Infrastructure as a Service (IaaS), Platform as a Service (PaaS) and Software as a Service (SaaS). This provides management and configuration for IoS. SIMaaS is an extension of SaaS, wrapping individual simulations and exposing them to the wider IoS. The final layer encapsulates WFaaS and provides services for users to interact with IoS. These services provide functionality to combine simulations into system representations using workflows and execute them. The following sections discuss the core IoS concepts of SIMaaS, WFaaS and how they are facilitated by Cloud computing.

3. Infrastructure Backbone: Cloud

Cloud computing has changed the way IT is consumed by enterprises by offering increased agility, velocity, convenience through on-demand network access to a shared and configurable computing resources [15]. As discussed in this section Cloud provides the backbone to IoT and therefore also to IoS. Depending on the resource provided, cloud computing can be classified into three service models [16]:

Infrastructure as a Service (IaaS) within which capacity is provided as storage, networks and compute resources. The consumer is able to deploy and run their own software including operating systems. The user does not manage or control the underlying cloud infrastructure.

Platform as a Service (PaaS) has a higher level of abstraction providing an application-hosting environment including programming languages, libraries, and tools. Using these, users can create their own applications and do not need to worry about controlling the underlying infrastructure such as storage, processing, and network.

Software as a Service (SaaS) provides specific applications hosted on the Cloud for consumption. Applications are accessible through web portals. This model has also made development and testing easy for providers via having access to the software. IoS extends the SaaS concept with SIMaaS in Section 4.

3.1. Infrastructure as a Service (IaaS)

IaaS is the first generation of cloud computing products. It provides computing instances, and is the lowest level of service that Cloud can provide. Customers can access these virtual machines to deploy their services and run applications. IaaS is provided under the pay-as-you-go model. Customers pay for the amount of resources and the time they use; typically by the hour, week or month.

Previously, the research on IaaS mainly focuses on resource provisioning from the server perspective. For example, Zhang et al. [17] integrate the P2P streaming technique to accelerate image downloading and improve the performance of large-scale resource provisioning. However, with the rapid development of IoT, big data applications as well as IoS, there are new challenges in the use of IaaS that can be concluded as follows:

Networked Infrastructure. A lot of complex applications are now migrating into the Cloud. The workflow of applications contains many components that need to communicate with each other. Therefore the virtual infrastructure must reside on a particular network topology. However, most of the current public cloud providers only provide separate instances for IaaS, and customers cannot access the topology as they design. Zhou et al. [18] discuss this networked infrastructure problem and proposes a method to realize Networked Infrastructure as a Service (NIaaS) on the public cloud, however they still leave a gap of how to combine these methods with specified applications in the likes of IoT.

Discrete geographical location. When considering migrating IoT applications into the Cloud, one challenge is the fact that not all components of the application are within the cloud environment, such as sensors that are distributed across different locations responsible for data collecting and information reporting. Connecting the Cloud and sensors efficiently whilst considering their discrete geographical locations is a challenge. This is also known as IoT-Cloud or Sensor-Cloud [19] and is not limited within IoT applications, other applications such as live broadcasting and disaster warning also face the same problem. IoS applications also introduce the challenge of transmitting large quantities of data generated at high frequencies and a combination of Edge and Cloud computing with intelligent data-flow management could mitigate this.

Dynamic Scaling. A common issue in many IoT applications is the connection and disconnection of sensors from the cloud. Furthermore, these sensors in IoT may not be in a fixed geographic location. Hence the ability to scale up and down according to requirements is very important. Scaling in a prompt way to meet demand is still challenging. Especially if the scaling parts of the applications are in different geographic regions. From an industry point of view, EC2 currently provides auto-scaling in only a single region and so is not sufficient for current application requirements.

Federated Cloud. Applications on Cloud are becoming increasingly complex, and this calls for a solution to acquire large-scale, virtual infrastructure from IaaS. However the resources of a single cloud provider can be insufficient and requiring a virtual infrastructure across several providers. When considering geographical location, multiple Clouds may be required to cover different regions as some Clouds may not provide IaaS in specific regions. Therefore, deploying applications and using IaaS efficiently in federated Clouds is another challenging problem.

3.2. Platform as a Service (PaaS)

The most popular choice for IoT projects is PaaS [1], with a Gartner prediction that more than 50% of all new applications developed on PaaS by 2020 will be IoT-centric. As a platform that sits between SaaS and IaaS, PaaS offers greater flexibility to control and customize an application and its data compared to SaaS. At the same time it hides some of the challenges associated with IaaS such as connectivity, integration, and security. The IoT PaaS is a natural step in the evolution of industrial enterprises, and will include analytics, simulation, and management, Aneka [20] is a representative example of this kind.

PaaS is not without its pitfalls however. There is relatively little standardization across the various PaaS offerings. That is to say, in current practice, all the configurations and process setups are completely dependent on the service provider. In addition, some PaaS providers are now using container technologies such as Docker to manage and control customer applications.

Other challenges including meeting the application Quality of Service (QoS) requirements. Heterogeneous underlying networks, server hardware, utilization rate, contention for shared resources along with the increased application complexity and data volume make guaranteeing QoS harder. Straggler problems hindering parallel application execution times is an example of late timing failure caused by missing QoS deadline [21]. Guaranteeing QoS at the platform level through scheduling and dynamic resource provisioning is a big challenge within IoT applications [6] and even more critical in IoS.

4. Simulation as a Service

One of the central tenets of IoS is the concept of SIMaaS [22] which extends the well known paradigm of SaaS (Figure 3). Similarly to SaaS, simulations are made available to users in an on-demand fashion whilst their execution is managed by the system. This enables the simulation domain to embrace the characteristics of Service Oriented Architectures (SOAs) [23] including:

- Modularity
- Loose coupling
- Standardisation and interoperability
- Quality of Service (QoS)
- Location Transparency

And therefore increase the usability and dependability of simulations as well as introduce the potential to integrate large-scale, complex simulations.

The remainder of this section considers three critical areas of consideration for facilitating SIMaaS: usability, interoperability and dependability.

4.1. Usability

One of the key benefits of SIMaaS is the increased usability of simulations and simulation tools. Specifically this refers to location transparency which is enabled by service-orientation. This allows users to focus on modelling and simulation tasks rather than managing the deployment and execution of their simulations and respective tools.

However, the ease with which a simulation can currently be exposed as a service is prohibitive and support for model interchange formats such as Functional Mockup Interface (FMI) [24] is far from ubiquitous. The usability of a simulation in SIMaaS is directly related to its interoperability since the foundation of IoS is connecting simulations together. The possible taxonomy of simulations is vast [25] with each type having specific characteristics relating to:

- how it creates a phenomena
- provides opportunity for understanding
- interactivity
- theoretical grounding
- unpredictability or randomness

Simulations can also categorised by the fidelity of their models; whether it is physical, virtual, or cyberphysical and whether it is continuous or discrete event based [26].

In the context of SIMaaS they must also be categorised with respect to their behavioural properties as services. The properties to be considered are:

- **Clock location** which can either be internal or external to the simulation service.
- **Input / Output** which can either be asynchronous, event-based or synchronous with the clock.
- **Timestep** which can either be fixed, variable or non-existent.
- **Causality** which can either be causal (assignment based) or non-causal (equation based).

Depending on these properties, the simulation may have real-time requirements which need to be considered as part of it's advertised QoS. This is especially true if there is no external access to a fixed timestep, internal clock. The QoS must therefore take into account attributes including response time, network latencies, as well as requirements for resources such as CPU and memory [27]. These requirements become evermore critical in the context of workflows as discussed in the Section 5.

4.2. Interoperability

Interoperability can be regarded as the ability of the simulation, or simulation tool, to be integrated with other systems, including other simulations, as part of a workflow. In this sense there are three aspects that need to be managed: standardisation, timing and synchronisation, as well as internal and external state.

SaaS provides a set of commonly used standards, including: WS-* [28], SOAP [29], and REST, the simulation community typically use the OMG Data Distribution Service (DDS) [30], the High Level Architecture (HLA) [31], or FMI [24]. However within the simulation community even these well-defined standards are not universally adopted with individual vendors providing their own incompatible interfaces or extending the standards with vendor specific features. Exposing a simulation using the provided interfaces typically requires extensive understanding of how those technologies function and therefore is not ideal for the simulation domain expert.

The standards focus on managing interoperability with regards to data, they do not manage timing, synchronisation, or state aspects of the simulation. As mentioned in the previous section the execution behaviour of a simulation will be dependent on various factors. In addition, the state representation and assumptions which are internal to the model can effect its ability to integrate. Understanding and managing these is critical to ensuring that simulations exposed as generic services are actually compatible with IoS. For example, a hardware-in-the-loop (HIL) system may be exposed as a service and will have strict timing requirements potentially requiring data updates at MHz frequencies. A human-in-the-loop simulation, such as a driving simulator, typically runs at 60Hz and more complex models such as CFDs may take hours to calculate a single timestep. These simulations are clearly not directly compatible and this must be managed autonomously by the system rather than leaving it to the user to understand which services can and cannot be integrated together.

4.3. Dependability

Finally, by exposing simulations as services their dependability can be directly increased. The concepts of dependability, security, and fault tolerance in the context of services are well known with the work by Avizienis et al. [32] defining dependability with regards to its key attributes of: availability, reliability, safety, maintainability, and integrity. In the context of simulation, integrity or trustworthiness of the services is vital to ensuring adoption of IoS by industry. Users, who are often engineers, must trust the system to not only provide results in a timely manner but also provide the correct result and to trust that it will continue to do so [33] this can be managed by including Quality of Experience (QoE) in service definitions [34].

The trustworthiness and validity of the result of a composition of simulations is vital to IoS. Therefore the service definition of each simulation must not only include some notion of face validity but also measures of model accuracy [35]. This will provide valid ranges for the given model and estimates of confidence intervals on the simulation results.

Bruning et al. [36] provides a taxonomy of faults specifically related to service orientation. One particular facet of SIMaaS is the potential mixing of real-time and non-real-time simulations which requires the addition of timing faults to the fault taxonomy and their subsequent management.

5. Workflow as a Service (WFaaS)

In order to simulate complex systems IoS allows multiple simulations to be composed into workflows. This facility to compose and execute simulation workflows is presented as a service and is the primary method for users to interact with IoS. WFaaS abstracts the specific details of the SIMaaS and Cloud platform away from the user so they can focus on composing system simulations.

Traditional workflows describe complex business processes by modelling the interactions between different business functions and the data (usually documents) that move between them [37]. The concept has been further generalised to scientific workflows where data processing functions are composed into algorithms and executed on a data set [38]. The IoS and its provision of SIMaaS provides an engineer with a collection of easily accessed, independent simulations. Just as workflows can used to compose business services or data processing functions, we propose their use for the composition of component simulations into system simulations. In such workflows, the connections between different simulations would have correspondence with interactions between components in the real-world system.

Since every workflow defined by WFaaS is a larger simulation that can be executed as a service, WFaaS can permit one workflow to be included as a simulation in other workflows. This leads to a hierarchy of workflows that can be composed for increasingly complex simulations. Another benefit of WFaaS over simply defining and running a workflow locally is in the underlying management layers which will abstract the implementation and execution of the respective simulations from the user's perspective. The nature and ordering of the simulation execution is determined by the underlying WFaaS management.

The remainder of this section discusses how the proposed simulation workflows facilitate IoS and how these workflows differ from other workflows.

5.1. Simulation Workflows

For the internet of simulation to realise the potential for widespread, large-scale simulation, the composition of models and simulations into system simulations must be intuitive. A sufficiently complex system simulation may utilise multiple simulations from varying domains and at different fidelities. A single user is unlikely to have expertise in all fields related to the simulation. Users may also have varying levels of experience with traditional programming languages and distributed systems. A common trend seen in many engineering disciplines is to use visual programming tools such as Simulink [39] or LabVIEW [40] and diagrammatic specification languages such as UML [41], SysML [42] or SoaML [43] to model systems. These visual notations provide more intuitive methods for system definition for a wide spectrum of users than their textual counterparts.

Existing workflow languages such as Business Process Model and Notation (BPMN) [44] and Yet Another Workflow Language (YAWL) [45] use diagrammatic notation and so lend themselves to this usecase. Others such as Business Process Execution Language (BPEL) are defined textually but can be displayed using visual notation [46]. However, these existing workflow languages are not currently sufficient for simulation workflows because the envisioned workflows are inherently parallel. For the simulation to accurately model feedback effects, each component of the simulation must be active for the entire execution of the workflow to correctly respond to the interactions with connected components. Ideally simulation workflows will specify *what* not *how* when composing system simulations.

For business process workflows Van Der Aalst et al. [37] define a framework for discussing workflows through patterns. They identify that workflows can be discussed from four perspectives: control flow, data, resource and operational each with associated patterns. From these perspectives, the envisioned simulations workflows are quite different from their process oriented counterparts.

Control-flow Perspective. This is the primary perspective for designing process workflows. It describes the ordering of activities and the constructions through which execution flow can branch. The control-flow of a workflow can be modelled as a petri-net [47] where tokens are passed to indicate the flow of execution between activities. The state of workflow execution is therefore related to which activities or services are active at a given time.

In contrast, we perceive simulation workflows as connecting virtual components through interactions. While we expect an effect to propagate through a workflow, there should be no defined execution flow as each component must be persistent. Just as in the real-world systems that are modelled in the IoS, the entities and components constituting the virtual system must have a continuous existence while the simulation executes for interactions to take place. For entirely feed-forward systems, a sequence of interactions from one component could be stored and then played back to subsequent simulations. However, if any feedback exists in the system then components cannot execute in isolation as the feedback behaviour could not be modelled. So the state of a simulation workflow is not measured by where in the workflow is active, instead it is defined by simultaneous values at each simulation interaction for a given simulation time.

Data Perspective. In process workflows this perspective layers the data entities that are generated or utilised during a process onto the control-flow. Russell et al. [48] identify 39 different patterns for possible data flow within a process these are categorised as:

- **Visibility** the extent that data is visible to entities within the process
- **Interaction** the manner data is communicated
- **Transfer** the means of data transfer
- **Routing** how data can influence other aspects of the workflow such as execution

In simulation workflows, the data perspective is the primary viewpoint of the simulation. Here the interactions between simulations may be modelled as data representing quantities of physical forces, properties of an interaction medium or some measure of state. In this respect, simulation workflows are very similar to visual and data-flow programming paradigms. The overall simulation workflow is defined by how the data generated by individual simulations is routed.

Resource and Operational Perspectives. The resource perspective maps an organisation's structure onto the workflow allocating the responsibility of executing activities to an entity or device. The operational perspective describes the individual actions taken in a process as well as the underlying tools used to take them. These details are not considered when designing simulation workflows, it is expected that these will be abstracted from the user and automatically managed by the underlying IoS infrastructure and WFaaS management layers. This would maintain the SOA principles of location transparency and loose coupling that make up WFaaS. Additionally, by abstracting these issues away from the user, they are able to focus effort into utilising the IoS to define systems.

5.2. Simulation Composition

The IoS should not be constricted to a specific type of simulation or specific tools. In fact, a major benefit of SIMaaS is the exposure of multiple different types and fidelities of simulation to the IoS. The selection of which simulation services to utilise to model a system in a simulation workflow should be left to the designer. Integration between simulations must underpin WFaaS so that any integration will be automatic or the system will inform the user that a particular combination of simulations is infeasible. Issues that may be encountered when constructing simulation workflows are: simulation type mismatch, data incompatibility and synchronisation. FMI [24] has addressed some of the difficulties with model integration but tool support is not ubiquitous.

A single workflow could contain many different types of models, the underlying layers providing WFaaS must work with SIMaaS to integrate between these types if possible and also with other workflows. In order for simulations to be successfully integrated by the WFaaS there must be a standardised taxonomy of simulation types available in the IoS and known integration methods between them. We envision this work building on previous taxonomies such as those by O'Keefe [49], Fishwick [50] and Miller et al. [26].

The data required and generated during simulation execution will require on-the-fly conversion defined by the integration. Even with a standardised data format, we expect the simulations to be using different units or descriptions for similar concepts. For example, a legacy simulation may not use metric units, two simulations could use different coordinate systems or even differing descriptions e.g. an amount of liquid measured as a mass or volume. While a standard could be devised to account for most of these measures, it may be impossible to capture all edge cases due to the varied nature of the simulations. Instead we expect the WFaaS to require some level of intelligence to construct ontologies across domains based on simulation descriptions.

Another issue that may be present between simulations is that of synchronisation. The simulation itself will require a notion of a global clock in order to ensure that the simulations remain synchronised. The challenge with many different types of simulations is the accounting for the variability of the timing properties present in each simulation. Section 4.1 provides some discussion of these, in each case the execution engine underpinning the WFaaS must account for the simulation clock and adjust its behaviour accordingly. This may result in some simulations imposing constraints on others in the workflow, this must be managed by the workflow internally.

5.3. Workflow Execution

WFaaS allows the execution of a given workflow to be provided as a service itself. In executing service workflows, WFaaS meets the principles of SOA and provides all of the technical benefits of SOA listed by Mueller et al. [8].

The simulation workflow abstracts the implementation details of the simulations away from the user and allows them to select from a library of pre-existing component simulations to construct the system. Each simulation is provided as a service so component simulations can be reused as well as simulation workflows. The properties of loose coupling and location transparency for SIMaaS extend to the workflow. Multiple workflows could be executed and the negotiation of which hardware executes a particular simulation or workflow can be determined by the underlying infrastructure.

6. Example Case Study

Figure 4 shows a simple example of using IoS to integrate simulations in the design and development a robotic arm to be used to lift objects from a table. Notably different parts of the system are modelled using different techniques, including: CAD running on HPCs for the physical properties, control software, electrical systems, as well as models of the environment. Many of these elements have different infrastructure requirements and the camera system is provided by an OEM with a service interface allowing remote access to simulation.

An IoS simulation can be considered to be comprised of two core element types: environments and physical objects. In the example provided, the environment must capture the structure and relationship between the table and blocks as well as any properties that will impact the system of interest.

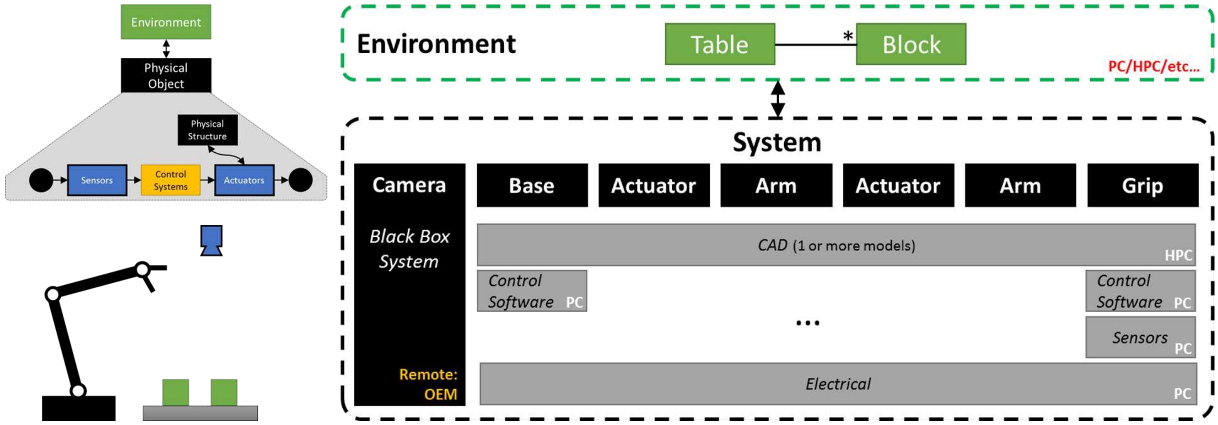

Figure 4. Example simulation integration involving a simple robot arm resulting in a complex set of distributed models and simulations running on various platforms

The simulation of the environment may itself be an integration of different simulations with a range of infrastructure requirements. The physical objects can be considered as being comprised of 4 aspects: sensors, control systems, actuators, and a physical structure each of which may be simulated separately.

In Figure 4 the designer might construct the system simulation by composing multiple workflows to model the various aspects of the system such as control and structure. The control system workflow would consist of the OEM camera, control system and sensor simulations. This workflow would interact with the wider environment simulation to generate sensor data. The structural workflow might consist of multiple FEA simulations of CAD models of the arm sections and models of the motors. The interactions between these two workflows and the wider environment simulation allows the designer to model the entire system. This integrated approach could potentially highlight whole system behaviour problems such as unsuitable camera placements due to occlusions or unsuitable combinations of arm sections and actuators.

To facilitate the integration of the component simulations of even a simple system such as this, IoS must facilitate and manage: the distributed and heterogeneous execution infrastructure; complexities from the varying temporal and causal simulation natures; and variations in data structures and types. Further, it must enable an OEM to expose a commercial sensitive simulation interface to clients in a secure and reliable fashion.

7. Conclusion

Internet of Things (IoT) is becomingly increasingly popular across a wide range of commercial industries and within the research community. However, as IoT targets the integration of *smart* devices it is not suited to the wider challenge of integrating models, simulations, prototypes, and analytics across industry. This type of integration is part of the natural evolution of IoT as it moves towards Internet of Everything and Anything from the perspectives of manufacturing, automotive, aerospace, defence, and government industries.

Internet of Simulation (IoS) is presented in this paper as the route to achieving this by specialising IoT to focus specifically on the integration of Simulation as a Service (SIMaaS). IoS builds on the core concepts and technologies including Cloud and SOA infrastructures whilst introducing specialist technologies such as DDS and HLA. The solution must handle the complexities of temporal integration; variation in model fidelity; as well as Workflow as a Service (WFaaS). Whilst IoS is a specialism of IoT it must also integrate with IoT via some standardised bridging technology. This facilitates the integration of the simulated world of design and analysis with the real world of big data and analytics in a LVC context. Additionally, the potential of IoS to improve cost effectiveness as well reduce manufacturing time-frames is premised on an ability to span the entire supply chain and facilitate rapid design decision making.

References

[1] V. Woods and R. van der Meulen, “Iot adoption is driving the use of platform as a service,” *Gartner report: Predicts 2016: PaaS Innovation Continues Unabated*, 2016.

[2] I. Bojanova, G. Hurlburt, and J. Voas, “Imagineering an internet of anything,” *Computer*, vol. 47, no. 6, pp. 72–77, jun 2014. [Online]. Available: http://dx.doi.org/10.1109/MC.2014.150

[3] N. Shyamsundar and R. Gadh, “Collaborative virtual prototyping of product assemblies over the internet,” *Computer-Aided Design*, vol. 34, no. 10, pp. 755 – 768, 2002.

[4] M. Hermann, T. Pentek, and B. Otto, “Design principles for industrie 4.0 scenarios,” in *IEEE 49th Hawaii International Conference on System Sciences*, 2016.

[5] C. Dickerson, S. Ji, S. Clement, D. Webster, D. Mckee, J. Xu, D. Battersby, N. Bevan, N. Turner, and W. Suart, “A demonstration of a service oriented virtual environment for complex system analysis,” *International Journal of Complex Systems*, 2015.

[6] J. Gubbi, R. Buyya, S. Marusic, and M. Palaniswami, “Internet of things (iot): A vision, architectural elements, and future directions,” *Future Generation Computer Systems*, 2013.

[7] R. Buyya and A. V. Dastjerdi, *Internet of Things: Principles and Paradigms*. Elsevier Science & Technology, 2016.

[8] B. Mueller, G. Viering, F. Ahlemann, and G. Riempp, "Towards Unverstanding the Sources of the Economic Potential of Service-Orientated Architecture: Findings from the Automotive and Banking Industry," in *ECIS*, 2007, pp. 1608–1619.

[9] S. N. Han, G. M. Lee, N. Crespi, N. Van Luong, K. Heo, M. Brut, and P. Gatellier, "Dpwsim: A simulation toolkit for iot applications using devices profile for web services," in *IEEE World Forum on Internet of Things (WF-IoT)*, 2014, pp. 544–547.

[10] R. Chacoon, "Iot needs simulation," 2016. [Online]. Available: https://iot-for-all.com/iot-needs-simulation-b07014224fc8

[11] J.-R. Martínez-Salio, J.-M. Lopez-Rodriguez, D. Gregory, and A. Corsaro, "A comparison of simulation and operational architectures," in *2012 Fall Simulation Interoperability Workshop (SIW), Simulation Interoperability Standards Organization (SISO)*, 2012.

[12] D. McKee, D. Webster, P. Townend, J. Xu, and D. Battersby, "Towards a Virtual Integration Design and Analysis Enviroment for Automotive Engineering," in *IEEE 17th International Symposium on Object/Component/Service-Oriented Real-Time Distributed Computing*, 2014.

[13] A. van Lamsweerde, *Requirements Engineering: From System Goals to UML Models to Software Specifications*. Wiley Publishing, 2009.

[14] H. A. Partsch, *Specification and transformation of programs: a formal approach to software development*. Springer Science & Business Media, 2012.

[15] R. Buyya, C. S. Yeo, S. Venugopal, J. Broberg, and I. Brandic, "Cloud computing and emerging it platforms: Vision, hype, and reality for delivering computing as the 5th utility," *Future Generation computer systems*, vol. 25, no. 6, pp. 599–616, 2009.

[16] P. Mell and T. Grance, "The NIST definition of cloud computing," 2011.

[17] Z. Zhang, Z. Li, K. Wu, D. Li, H. Li, Y. Peng, and X. Lu, "Vmthunder: fast provisioning of large-scale virtual machine clusters," *IEEE Transactions on Parallel and Distributed Systems*, vol. 25, no. 12, pp. 3328–3338, 2014.

[18] H. Zhou, Y. Hu, J. Wang, P. Martin, J. Su, C. De Laat, and Z. Zhao, "Fast resource co-provisioning for time critical application based on networked infrastructure," in *IEEE International Conference on CLOUD*, 2016.

[19] A. Alamri, W. S. Ansari, M. M. Hassan, M. S. Hossain, A. Alelaiwi, and M. A. Hossain, "A survey on sensor-cloud: architecture, applications, and approaches," *International Journal of Distributed Sensor Networks*, 2013.

[20] Y. Wei, K. Sukumar, C. Vecchiola, D. Karunamoorthy, and R. Buyya, "Aneka cloud application platform and its integration with windows azure," *arXiv preprint arXiv:1103.2590*, 2011.

[21] X. Ouyang, P. Garraghan, R. Yang, P. Townend, and J. Xu, "Reducing late-timing failure at scale: Straggler root-cause analysis in cloud datacenters," in *Annual IEEE/IFIP International Conference on Dependable Systems and Networks*. DSN, 2016.

[22] S. Shekhar, H. Abdel-Aziz, M. Walker, F. Caglar, A. Gokhale, and X. Koutsoukos, "A simulation as a service cloud middleware," *Annals of Telecommunications*, vol. 71, no. 3, 2016.

[23] M. P. Papazoglou, P. Traverso, S. Dustdar, and F. Leymann, "Service-Oriented Computing: State of the Art and Research Challenges," *Computer*, vol. 40, no. 11, pp. 38–45, 2007.

[24] T. Blochwitz, M. Otter, M. Arnold, C. Bausch, C. Clauß, H. Elmqvist, A. Junghanns, J. Mauss, M. Monteiro, T. Neidhold, D. Neumerkel, H. Olsson, J. Peetz, and S. Wolf, "The Functional Mockup Interface for Tool independent Exchange of Simulation Models The Functional Mock-Up Interface," in *Modelica Conference*, 2011, pp. 105–114.

[25] K. Schmucker, "A taxonomy of simulation software," *Learning Technology Review*, pp. 40–75, 1999.

[26] J. Miller, G. Baramidze, A. Sheth, and P. Fishwick, "Investigating ontologies for simulation modeling," in *37th Annual Simulation Symposium*. IEEE, 2004.

[27] D. W. McKee, D. Webster, J. Xu, and D. Battersby, "DIVIDER: Modelling and Evaluating Real-Time Service-Oriented Cyberphysical Co-Simulations," in *IEEE 18th International Symposium on Real-Time Distributed Computing*, 2015, pp. 272–275.

[28] W3C and Innoq, "Web Services Standards," p. 1, 2007. [Online]. Available: http://www.w3.org/2002/ws/

[29] W3C, "Soap," 2007. [Online]. Available: www.w3.org/TR/soap/

[30] G. Pardo-castellote, "OMG Data-Distribution Service (DDS): Architectural Overview," Real-Time Innovations, Inc. (RTI), Tech. Rep., 2005.

[31] W. Xiong and W.-T. Tsai, "HLA-Based SaaS-Oriented Simulation Frameworks," *IEEE 8th International Symposium on Service Oriented System Engineering*, pp. 376–383, 2014.

[32] A. Avizienis, J.-C. Laprie, B. Randell, and C. Landwehr, "Basic concepts and taxonomy of dependable and secure computing," *IEEE Transactions on Dependable and Secure Computing*, pp. 11–33, 2004.

[33] F. B. Schneider, *Trust in Cyberspace*. National Academies Press, 1999.

[34] M. Fiedler, T. Hossfeld, and P. Tran-Gia, "A generic quantitative relationship between quality of experience and quality of service," *IEEE Network*, vol. 24, no. 2, pp. 36–41, 2010.

[35] R. G. Sargent, "Verification and validation of simulation models," 2005, pp. 130–143. [Online]. Available: http://dl.acm.org/citation.cfm?id=1162708.1162736

[36] S. Bruning, S. Weissleder, and M. Malek, "A Fault Taxonomy for Service-Oriented Architecture," in *10th IEEE High Assurance Systems Engineering Symposium*, 2007, pp. 367–368.

[37] W. M. van Der Aalst, A. H. Ter Hofstede, B. Kiepuszewski, and A. P. Barros, "Workflow patterns," *Distributed and parallel databases*, vol. 14, no. 1, pp. 5–51, 2003.

[38] J. Dehnert and W. M. Van Der Aalst, "Bridging the gap between business models and workflow specifications," *International Journal of Cooperative Information Systems*, 2004.

[39] Mathworks, *Matlab Simulink Toolbox*. Natick, Massachusetts: The MathWorks Inc., 2016.

[40] N. Instruments, "Labview," 2016. [Online]. Available: http://www.ni.com/labview

[41] OMG, "UML," 2013. [Online]. Available: http://www.omg.org/spec/UML/

[42] ——, "SysML," 2015. [Online]. Available: http://www.omgsysml.org/

[43] ——, "SoaML," 2012. [Online]. Available: http://www.omg.org/spec/SoaML/

[44] S. A. White, "Introduction to bpmn," *IBM Cooperation*, 2004.

[45] W. van der Aalst and A. ter Hofstede, "Yawl: yet another workflow language," *Information Systems*, vol. 30, no. 4, pp. 245 – 275, 2005.

[46] D. Schumm, D. Karastoyanova, F. Leymann, and J. Nitzsche, "On visualizing and modelling bpel with bpmn," in *IEEE Grid and Pervasive Computing Conference*, 2009.

[47] N. Russell, A. H. Ter Hofstede, and N. Mulyar, "Workflow controlflow patterns: A revised view," 2006.

[48] N. Russell, A. H. M. ter Hofstede, D. Edmond, and W. M. P. van der Aalst, *Workflow Data Patterns: Identification, Representation and Tool Support*. Springer Berlin Heidelberg, 2005, pp. 353–368.

[49] R. O'sKeefe, "Simulation and expert systems- a taxonomy and some examples," *SIMULATION*, vol. 46, no. 1, pp. 10–16, jan 1986.

[50] P. A.Fishwick, "A taxonomy for simulation modelling based on programming language principles," *IE Transactions*, 1998.