MSG-168 Lecture Series on Modelling and Simulation as a Service (MSaaS)

12. MSaaS Composition and Technical Approach

Tom van den Berg
TNO Applied Physics Laboratory
The Netherlands

tom.vandenberg@tno.nl

ABSTRACT

This paper covers the topic of "Composition and Technical Approach" within the Allied Framework for MSaaS. Composition of services is closely related to the still open grand challenge of composability of simulation models or components. This paper starts with an overview of the topics interoperability and composability, continues with a description of the Integrator use case (see Technical Reference Architecture) for creating compositions, and closes with composition building block requirements, and future challenges.

1.0 INTEROPERABILITY AND COMPOSABILITY

Over the years the topics of interoperability and composability have been discussed in several papers. In "A Composability Lexicon" [1] Petty defines interoperability as:

> *"the ability of different simulations, connected in a distributed simulation, to meaningfully collaborate to simulate a common scenario or virtual world"*

And composability as:

> *"the capability to select and assemble simulation components in various combinations into simulation systems to satisfy specific user requirements"*

Also, as stated in the same paper: Interoperability is necessary but not sufficient for composability. Composability *requires* interoperability, but interoperability is possible without composability, i.e., without the ability to combine and recombine. For example, two models A and B may be interoperable but it does not make sense to compose them together if their objectives and underlying assumptions are not aligned. E.g. the composition of an engine model that produces supersonic aircraft velocities and a flight dynamics model that is only valid for subsonic velocities, does not make sense although both models might be interoperable.

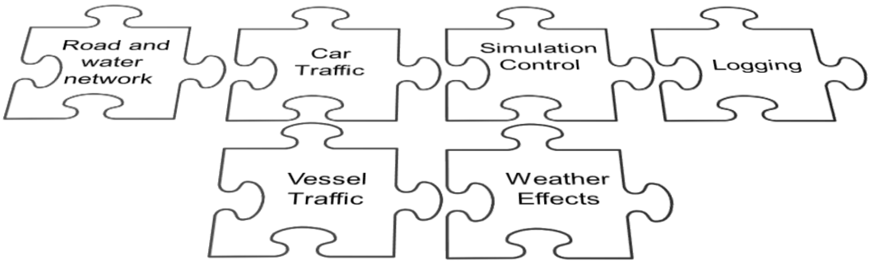

Figure 1: Interoperability: a meaningful collaboration of distributed simulations.

In “Toward a Family of Maturity Models for the Simulation Interconnection Problem” [2] Page et al describe three dimensions to the simulation interconnection problem:

- *Composability* - realm of the model (e.g. two models are composable if their objectives and assumptions are properly aligned).
- *Interoperability* - realm of the software implementation of the model (e.g. are the data types consistent, have the little endian/big endian issues been addressed, etc.)
- *Integratability* - realm of the site the simulation is running at (e.g. have the host tables been set up; are the NIC cards working properly).

To successfully achieve the cooperative execution of two or more models, each of these dimensions of the interconnection problem must be addressed.

Tolk defines in [3] five levels at which simulation models can interoperate. These levels are called Levels of Conceptual Interoperability (LCIM) between simulation models. In [4] these levels got expanded to the current seven Levels of Conceptual Interoperability between simulation models:

- Level 0: *no interoperability*.
- Level 1: *technical interoperability*: a communication protocol exists for exchanging data between participating systems. On this level, a communication infrastructure is established allowing systems to exchange bits and bytes, and the underlying networks and protocols are unambiguously defined.
- Level 2: *syntactic interoperability*: a common protocol to structure the data is used and the format of the information exchange is unambiguously defined. This layer defines structure.
- Level 3: *semantic interoperability*: a common information exchange reference model is used, the meaning of the data is shared and the content of the information exchange requests are unambiguously defined. This layer defines (word) meaning.
- Level 4: *pragmatic interoperability*: the interoperating systems are aware of the methods and procedures that each system is employing. The use of the data is understood by the participating systems and the context in which the information is exchanged is unambiguously defined. This layer puts the (word) meaning into context.
- Level 5: *dynamic interoperability*: the interoperating systems are able to comprehend the state changes that occur in the assumptions and constraints that each is making over time, and they are able to take advantage of those changes. When interested specifically in the effects of operations, this becomes increasingly important; the effect of the information exchange within the participating systems is unambiguously defined.
- Level 6: *conceptual interoperability*: the assumptions and constraints of the meaningful abstraction of reality are aligned. This requires that conceptual models are documented based on engineering methods enabling their interpretation and evaluation by other engineers.

The seven levels of the LCIM are shown in Figure 2, including the three dimensions of the simulation interconnection problem listed alongside the levels.

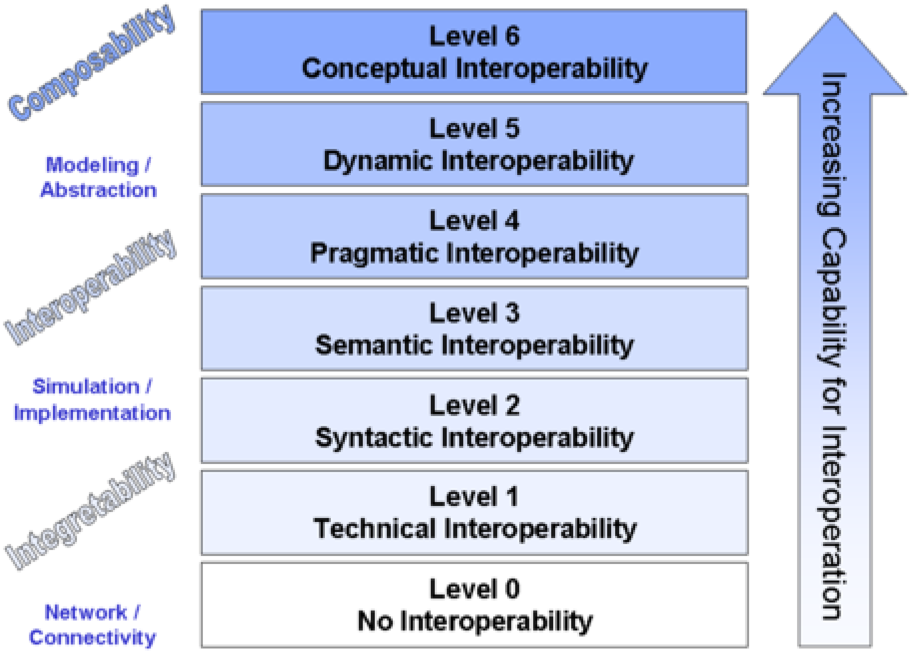

Figure 2: Levels of Conceptual Interoperability Model (LCIM) (from [5]).

On the left side of seven levels in Figure 2 the three dimensions of the simulation interconnection problem are shown:

- *Integratability* (level 1): refers to the physical and technical connections between systems, which include hardware and firmware, and network protocols.
- *Interoperability* (level 2-4): refers to the simulation and implementation details of interoperations, including exchange of data elements based on a common data interpretation.
- *Composability* (level 5-6): refers to the alignment of issues on the modeling level.

In "The Levels of Conceptual Interoperability Model: Applying Systems Engineering Principles to M&S" [6] Wang et al use the Levels of Conceptual Interoperability Model (LCIM) as a framework for conceptual modeling and for descriptive and prescriptive uses. In Table 1 the implications of the LCIM are listed, showing per level: premise, information and contents that should be defined, domain, focus, and capability to compose models.

Table 1: Implications of LCIM (adapted from [6]).

Level	Premise	Information defined	Contents clearly defined	Domain	Focus	Capability
Level 6 Conceptual	Common conceptual model	Assumptions, constrains, etc.	Documented conceptual model	Modeling abstraction	Composability of models	High
Level 5 Dynamic	Common execution model	Effect of data	Effect of information exchanged			
Level 4 Pragmatic	Common workflow model	Use of data	Context of information exchanged	Simulation implementation	Interoperability of models	Medium
Level 3 Semantic	Common reference model	Meaning of data	Content of information exchanged			

Level 2 Syntactic	Common data structure	Structured data	Format of information exchanged			
Level 1 Technical	Common communication protocol	Bits and bytes	Symbols of information exchanged	Network connectivity	Integratability of models	Low
Level 0 No	No connection	NA	NA			

In the same paper Wang et al show how the LCIM can be used in a prescriptive role by providing the requirements that must be satisfied to reach a certain level of interoperability between simulation models, and engineering approaches on how to achieve that. The requirements and approaches are summarized in Table 2.

Table 2: Prescriptive role of LCIM (adapted from [6]).

Level	Prescription of Requirements to reach this Level	Common Reference Engineering Approaches
Level 6 Conceptual	A shared understanding of the conceptual model of a system (exposing its information, processes, states, and operations).	DoDAF; Platform Independent Models of the MDA; SysML
Level 5 Dynamic	The means of producing and consuming the definitions of meaning and context is required.	Ontology for Services; UML artifacts; DEVS; complete UML; BOM
Level 4 Pragmatic	A method for sharing meaning of terms and methods for anticipating context are required.	Taxonomies; Ontology; UML artifacts, in particular sequence diagrams; DEVS; OWL; MDA
Level 3 Semantic	Agreement between all systems on a set of terms that grammatically satisfies the syntactic level solution requirements is required.	Common Reference Model; Dictionaries; Glossaries; Protocol Data Units; HLA RPR-FOM
Level 2 Syntactic	An agreed-to protocol that can be supported by the technical level solution is required.	XML-XSD; HLA OMT; Interface Description Language; WSDL
Level 1 Technical	Ability to produce and consume data in exchange with systems external to itself is required.	Network connection standards such as HTTP; TCP/IP; UDP/IP, messaging middleware, such as HLA-RTI
Level 0 No	NA	NA

In Table 2 the High Level Architecture (HLA) is listed at levels 1 to 3. The HLA is a standard architecture for distributed simulation specified in IEEE 1516-2010 [7] [8] [9]. According to the LCIM the HLA Runtime Infrastructure (RTI) is listed at level 1, providing technical interoperability between participating systems. The HLA Object Model Template (OMT) specification defines the structure of the information and is therefore at level 2. The SISO HLA Real-time Platform Reference (RPR) Federation Object Model (FOM) [10] is an example of a standard and reference object model that conforms to the HLA OMT specification, providing a common agreement for many participating systems. The RPR-FOM is therefore at the semantic level 3. Simulation environment agreements - although not part of the HLA standard - are at the pragmatic level 4 when they capture the methods and procedures that each system should employ in using the data. However, at present simulation environment agreements tend to be mostly textual (see GRIM and associated IEEE standards [11]). A formal architecture framework such as DoDAF or NAF, and modeling languages such as UML or SysML are preferred to express modeling agreements in order to reach a higher level of interoperability [12].

As can be concluded from the LCIM, the HLA focuses on network connectivity as well as on simulation implementation, in particular on syntactic and semantic interoperability between simulation models. The HLA targets simulation interoperability, and, currently, much less simulation composability.

2.0 MSAAS COMPOSITION AND DEPLOYMENT

2.1 Introduction

The composition of M&S Resources is in general an engineering task that requires human effort to integrate and test components. The task can be automated to some extent, for example by standardizing on certain technology (solving the "integratability" issues at level 1) and standardizing on interoperability standards (solving the "interoperability" issues at levels 2-3) – see previous section. An example is the integration of HLA RPR-FOM applications. Assuming each application has been well tested against the RPR-FOM requirements, integration of such components is relatively easy for levels 1-3 and 4 to some degree. However, interoperability at level 4 and higher is not guarenteed by stating to be RPR-FOM compliant. Methods and procedures that each component employs in using the exchanged data (i.e. the effects) should be well understood and implemented. Fair fight principles are a good example for this, where effects may be interpreted differenty per application because of modelling differences such as for resolution and correlation of the synthetic physical environment. The concepts modelled in the simulation should be well defined in an implementation independent manner that can be interpreted by engineering tools. This touches on topics such as validity, fidelity, correlation and accuracy.

The MSaaS concept is not a magic bullet that will solve the interoperability challenges and take a away the (DSEEP) engineering activities and tasks. These activities and tasks remain to be valid as before. However, an MSaaS Capability can offer engineering services to support the composition of M&S Recources. For example, by providing metadata services (to provide information about M&S Resources and thereby aid the selection, configuration, and integration of services), repository services (to provide access to M&S Resources), composition services (to assist the creation of deployment descriptions for the automated deployment of a composition), and verification services (to automatically verify a simulation service or a composition of services). A front-end application can provide an "integrated development environment" for composition, using the mentioned back-end services.

The remainder of this section will elaborate on the Integrator use case actor that was introduced earlier in the MSaaS Tehnical Reference Architecture. The Integrator performs the engineering activities and tasks to create compositions and deployment descriptions, using engineering services provided by the MSaaS Capability. However, we first define the main concepts.

2.2 Main concepts

This section defines the most relevant concepts used in this paper.

A *composition* (as noun) is an arrangement of M&S Resources that fullfils some purpose, such as providing a service. M&S Resources include simulation services, configuration data, scenario data, etc. A composition can be described at different levels of abstraction. A composition may include or refer to a deployment description.

Composing (as verb) is the act of creating a composition.

The following figure provides an example of a complex composition that includes to types of services (participating and supporting). The connections indicate dependencies between services.

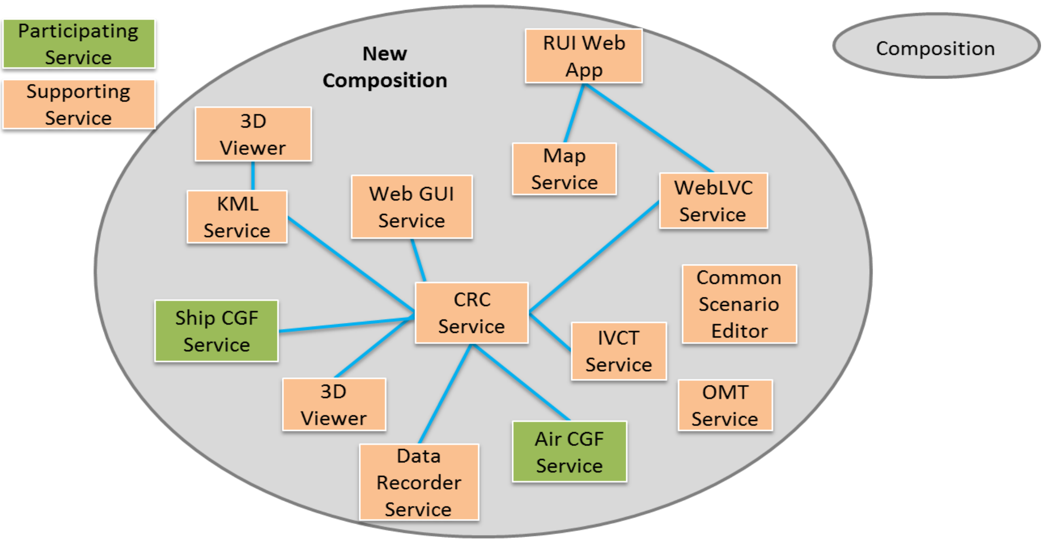

Figure 3: Example of a composition.

A deployment description provides informatiom for the automatic deployment of the M&S Resources in a composition. This includes information about where and how the M&S Resources are deployed and configured, and how any physical (external) systems are to be connected.

There are generally two types of deployment descriptions: *declarative* and *imperative*. Declarative deployment descriptions have become widely accepted, such as Chef, Terraform, Juju (JuJu Charms) and Kubernetes (Helm Charts, [13]). A declarative deployment description describes the desired end-state of the deployment. An imperative deployment description describes the step by step flow to deploy a composition. An *orchestrator* is responsible for the processing of a deployment descriptor and for managing the deployment, for instance Kubernetes.

Deploying (as noun) is the act of deploying a composition. The result is a *deployment*.

An *M&S Resource* is a reusable M&S specific item, such as a data file, a terrain dataset (the actual terrain data files), an M&S Service implementation, a M&S User Application implementation (i.e. executable software, container image, VM image), a composition description, or a deployment description. *M&S Repository Services* define the capability for storing and managing M&S Resources (see MSaaS Reference Architecture presentation).

M&S Resource Metadata provides information about an M&S Resource, such as what it does, who is the owner, where it is located (URL), and, if the resource concerns a service, a specification of or a reference to the service interface. *M&S Registry Services* define the capability for storing and managing M&S Resource Metadata (see MSaaS Reference Architecture presentation).

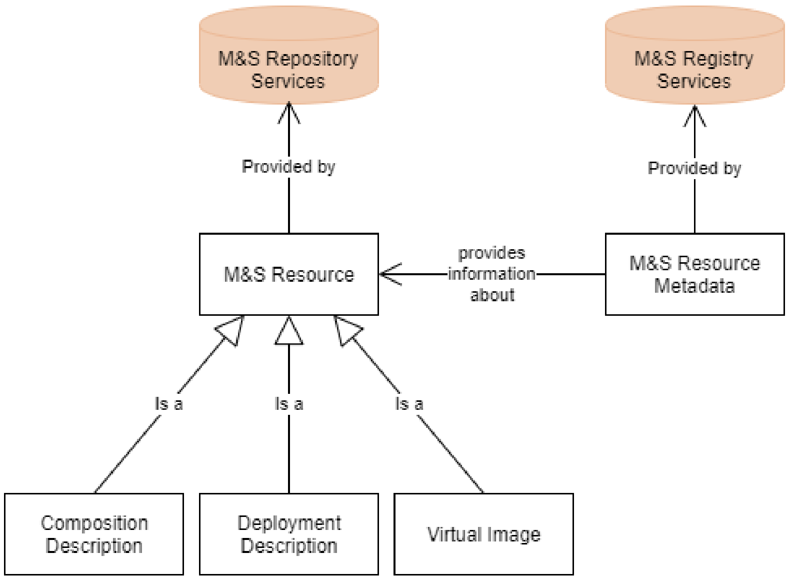

Figure 4: Main concepts and relationships.

2.3 Use case: create composition

This section discusses the Integrator use case and the sub use cases for creating a composition.

The objective of this use case is to combine/aggregate M&S Resources into a composition with an associated deployment description that can be used by Simulation Operator. This may include coordination with the Supplier of an M&S Resource. The use case is illustrated in the following figure.

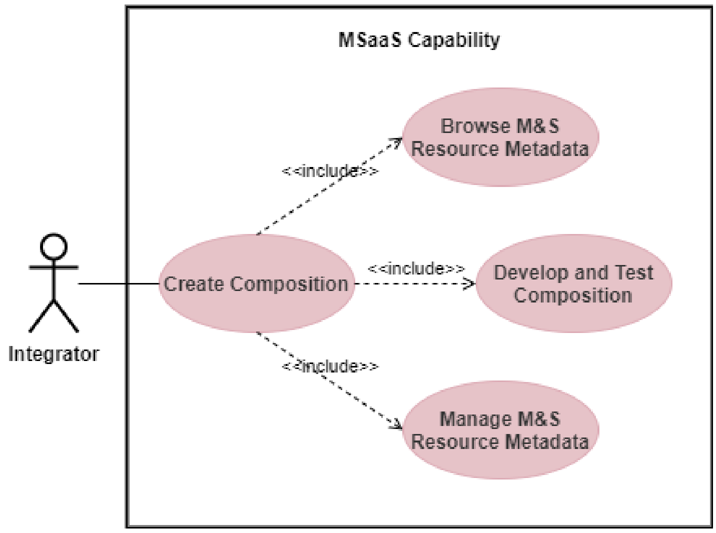

Figure 5: Integrator use case.

2.3.1 Manage M&S Resource Metadata

Given the requirements for a composition, the Integrator needs to be able to determine the applicability of each M&S Resource for inclusion in the composition. The metadata provided for an M&S Resource might indicate it being suitable but may be missing detail, completeness, or compliance with expected norms/taxonomies. In such cases, the Integrator will work with the Supplier to update/revise the M&S Resource Metadata. The Integrator uses the M&S Registry Services to manage the metadata associated with M&S Resources.

2.3.2 Browse M&S Resource Metadata

The Integrator's task is to create a composition that may be used by the Simulation Operator and to define the options that will be available to the Simulation Operator. In order to achieve this task, the Integrator is provided with simulation requirements for a specific simulation event (output of DSEEP Step 2). The Integrator identifies the M&S Resources to be included in the composition to meet the requirements of the event.

The Integrator identifies candidate M&S Resources that best support the simulation requirements by accessing the M&S Registry Services. In addition to finding M&S Resources that meet simulation technical requirements, also licensing restrictions, export controls, etc. need to be considered and recorded.

In particular, the retrieved M&S Resource metadata includes:

- The list of Simulation Services available from the local MsaaS Capability or from another (remote) MSaaS Capability.

- The metadata that describes the services. The information includes the services' core descriptive information, functionality, domain, resolution, accessibility, availability, interface description, dependencies, and data artifacts. A Service Description Template is used to structure this information.

2.3.3 Develop and Test Composition

Each service may contain configuration details that describe how to modify the service's behavior. The Integrator is able to retrieve service configuration information from the M&S Repository Services indicated by the metadata obtained from the M&S Registry Services.

The Integrator determines the needed configuration as a trade-off between the simulation requirements and the service configuration options (e.g., restrictions on data fields or allowed data format, etc.). Several trade-offs have to be made, for example related to: interoperability (to what degree are services interoperable, see section on LCIM), security (what are the security constraints w.r.t. service access and use), performance (does the service meet the performance requirements), sub-division in sub-compositions, etc.

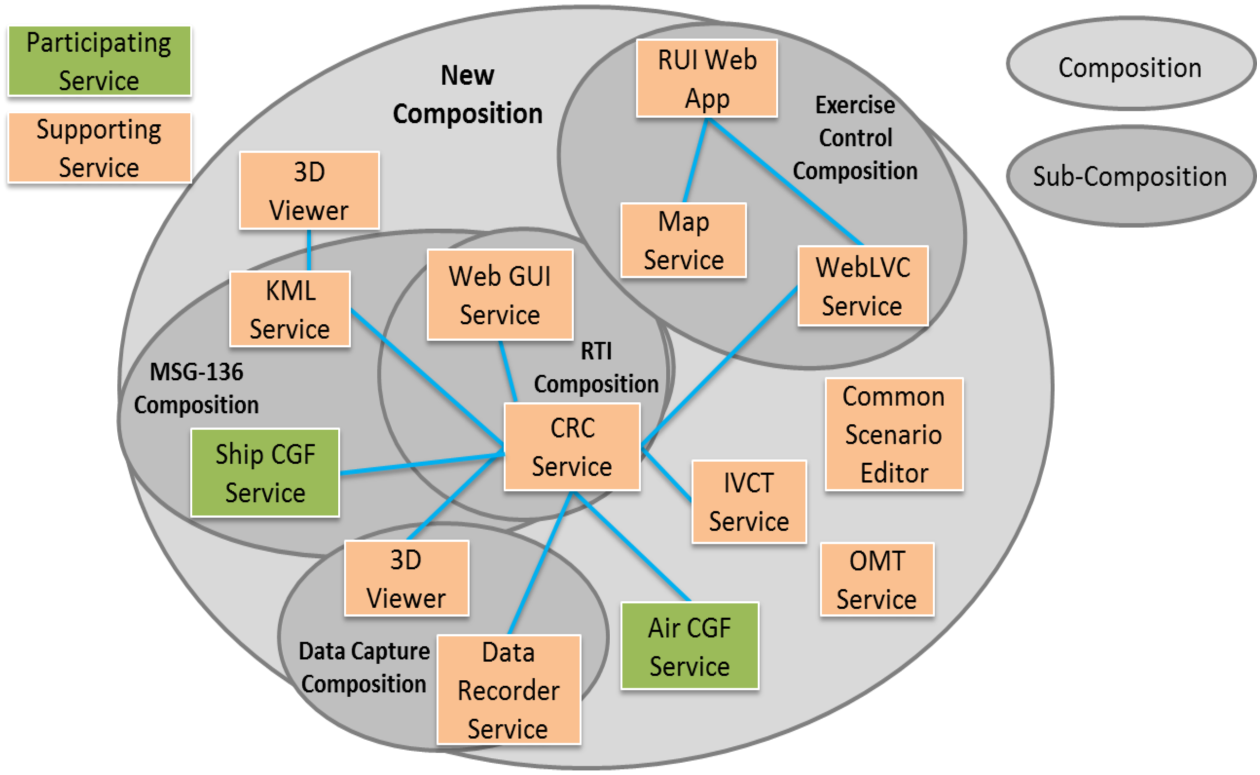

Figure 6: Example of sub-compositions.

Therefore, the Integrator needs to collect the following information:

- Simulation requirements related data (e.g., MSDL, terrain description, performance, etc.).
- Service Interface for each service (from the Service Metadata).
- Configuration data for each service.

The output of this step will be the description of the specified composition in terms of Configured Services. Once the required services have been configured, the next steps are to create a deployment description defining amogst others:

- Compute resources (computation, networking, and other infrastructure-related resources that are needed).
- Composition of services across compute resources.

The output of this work will be:

- A deployment description (to be managed by the *M&S Repository Services*)
- Composition-related metadata (to be managed by the *M&S Registry Services*)

The *Integrator Portal Applications* provide the ability to upload, store, and retrieve a composition. No interoperability between deployment descriptions in different MSaaS Capabilities is considered, as it strongly depends on the underlying technology platform.

The Integrator stores the composition, according to the specific MSaaS Capability, in terms of:

- Composition (i.e., list of configured services)
- Deployment description:
    - Assets (compute node, network and other infrastructure related resources that are needed)
    - How those assets are related to each other (e.g. connect compute nodes to network, etc.)

3.0 COMPOSITION CONCEPT DEMONSTRATORS

The composition and composability of M&S Resources (or better, services) is very much an area of research. No general solutions for composing arbitrary simulation components exists to date. Concept demonstrators of solutions for "Integrator Portal Applications" have been developed by MSG-136/MSG-164 participants on the premise of interoperability assumptions. I.e. assuming that simulation services are already interoperable at LCIM 1-3 for instance, and making technology choices that services are deloyed using for instance OpenStack, Kubernetes, or Docker.

Figure 7 provides a high-level view of a Composition Tool for the Integrator. Based on the user requirements the Composition Web Processing Service can automatically search for qualifying compositions and return a ranked list of results. The Integrator can view and adapt a composition, or create a new composition. The former is illustrated in Figure 8, where an existing composition is shown in a topology view. The prototype Composition Tool allows "prepared" simulation components (stored in the Repository) to be composed and deployed on a given technology platform.

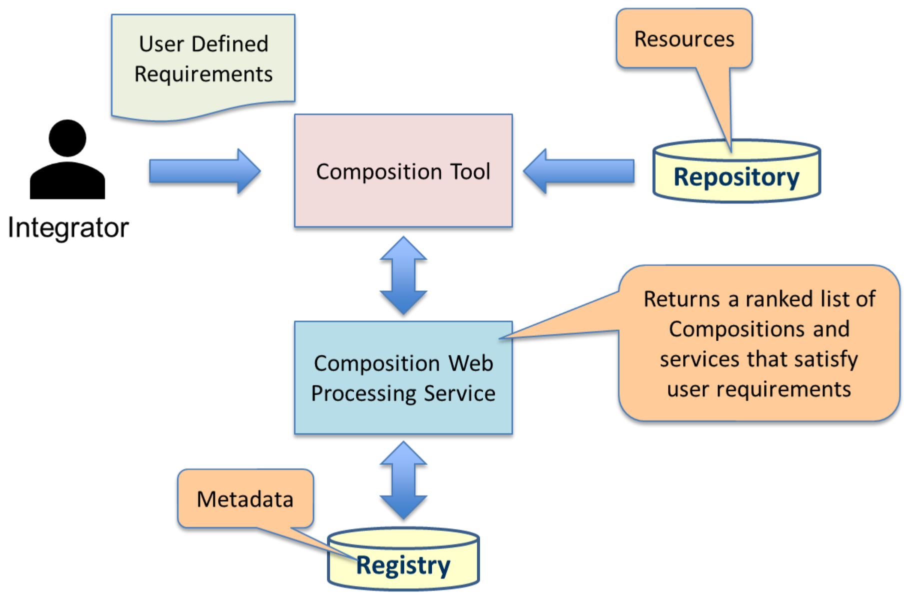

Figure 7: Integrator Applications for composition: UK AIMS Prototype Composition Tool.

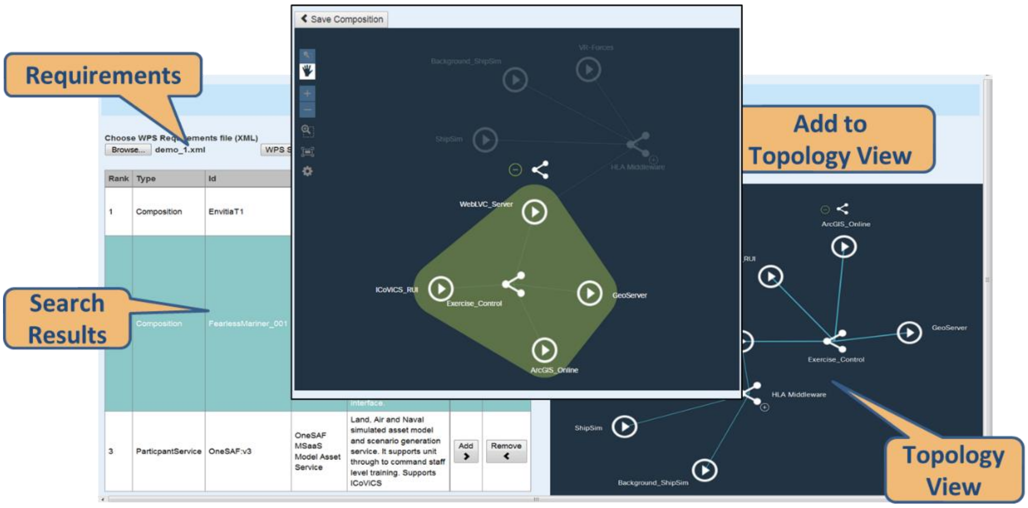

Figure 8: Integrator Portal Applications for composition: UK AIMS Prototype Composition Tool.

Other concept demonstrators build on open source solutions, such as Kubernetes [14]. A Kubernetes application or service can be described by a Helm Chart [13]. A Helm Chart is a collection of text files that declaratively describe what to deploy. A *chart* might be used to deploy just a single service, or something more complex like an entire composition of simulation services. A composition of simulation services can be created by adding (dropping in) sub-charts to the main chart, or, alternatively, by creating dependencies between charts. Helm will automatically deploy the main chart, the sub-charts, and any dependencies using Kubernetes as the orchestrator.

The development of Helm Charts is (with today's technology) mostly a manual activity, using simple text editors. Chart verification tools are available however. Metadata to aid composition can be included in a Helm Chart. Metadata of charts can be shared and searched via a (public or private) Artifact Hub [15]. Discovered charts can be included in a main chart by composition (as subchart) or by reference (as dependency). Figure 9 provides an example of an Artifact Hub search for charts related to machine learning. The returned results include further information such as deployment instructions, version data, and chart configuration parameters.

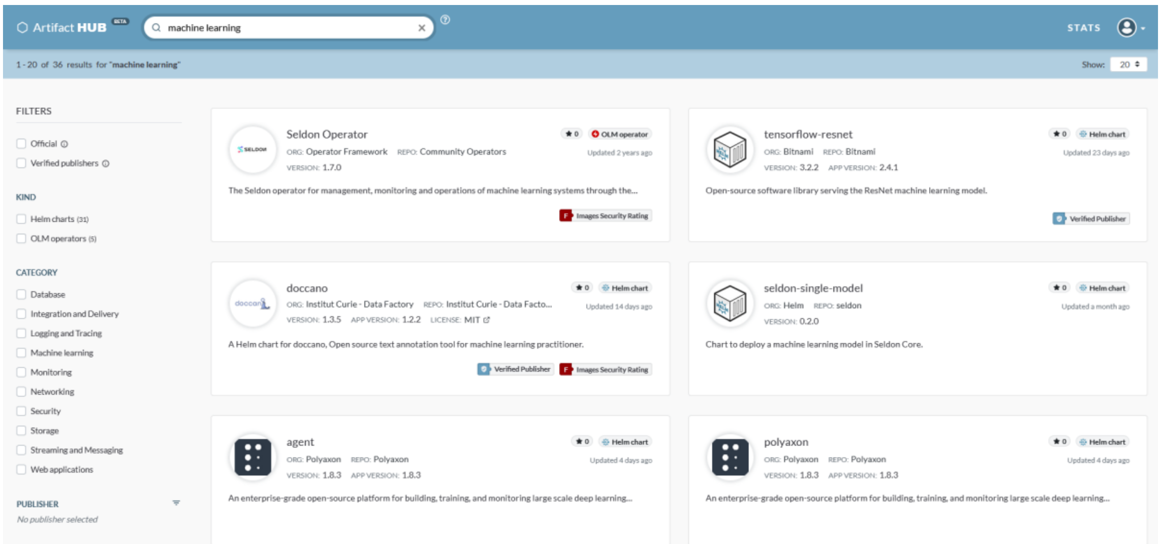

Figure 9: Sharing Metadata Resources: the Artifact Hub.

4.0 INTEGRATOR PORTAL APPLICATIONS – REQUIREMENTS

The **Integrator Portal Applications** are part of the general **M&S Portal Applications** and define capabilities that *enable* the discovery, composition and execution of Simulation Services and M&S User Applications in an MSaaS Capability. For more information on these capabilities, see the Technical Reference Architecture.

Based on the concept development activities in MSG-136, initial requirements for the **Integrator Portal Applications** for Composition have been developed. Several requirements are listed below.

These applications provide the ability to:

1. Search for M&S Resources and M&S Resource Metadata using different options

    1.1. Search based on the type of resource, semantic tag, C3 Taxonomy category

    1.2. Search based on requirements (which may be provided in text format)

2. Select the M&S Resources for a composition

3. Display M&S Resource metadata and relationships with other M&S Resources

4. Create a composition from selected M&S Resources

    4.1. Connect M&S Services (in terms of provided/required interface/data)

    4.2. Highlight M&S Service data needs

    4.3. Verify if connections are meaningful

    4.4. Verify if composition is complete (all data needs are satisfied)

5. Store composition

5.0 FUTURE CHALLENGES

Composability of simulation components is one of the grand challenges in the M&S domain. Future challeges in relation to the composition of M&S Services clearly relate to this grand challenge:

- Automatically integrating components using different interoperability protocols – Syntactic Interoperability;

- Automatically ensuring data exchanged between components is what is required – Semantic Interoperability;

- Automatically ensuring each component has the same understanding of the data – Pragmatic Interoperability;

- Plus higher levels of the LCIM.

6.0 ACKNOWLEDGEMENTS

The author would like to acknowledge to contribution of MSG-164 in relation to the content of this paper; in particular members of the MSG-164 TEK team.

References

[1] M. Petty and E. Weisel, "A Composability Lexicon (03S-SIW-023)," in *SISO Simulation Interoperability Workshop*, Kissimmee, FL, 2003.

[2] E. Page, R. Briggs and J. Tufarolo, "Toward a Family of Maturity Models for the Simulation Interconnection Problem (04S-SIW-145)," in *SISO Simulation Interoperability Workshop*, Arlington, VA, 2004.

[3] A. Tolk and J. Muguira, "The Levels of Conceptual Interoperability Model (03S-SIW-007)," in *SISO Simulation Interoperability Workshop*, Orlando, FL, 2003.

[4] Turnitsa, "Extending the Levels of Conceptual Interoperability," in *Proceedings IEEE summer computer simulation conference*, *IEEE CS Press*, 2005.

[5] "Wikipedia," Wikipedia, March 2021. [Online]. Available: https://en.wikipedia.org/w/index.php?title=File:LCIM.png.

[6] W. Wang, A. Tolk and W. Wang, "The Levels of Conceptual Interoperability Model: Applying Systems Engineering Principles to M&S," in *Spring Simulation Multiconference*, San Diego, CA, USA, 2009.

[7] IEEE, IEEE Standard for Modeling and Simulation (M&S) High Level Architecture (HLA) - Framework and Rules (IEEE 1516-2010), IEEE , 2010.

[8] IEEE, IEEE Standard for Modeling and Simulation (M&S) High Level Architecture (HLA) – Federate Interface Specification (IEEE 1516.1-2010), IEEE, 2010.

[9] IEEE, IEEE Standard for Modeling and Simulation (M&S) High Level Architecture (HLA) – Object Model Template (IEEE 1516.2-2010), IEEE, 2010.

[10] SISO, Standard for Guidance, Rationale, and Interoperability Modalities (GRIM) for the Real-time Platform Reference Federation Object Model (RPR FOM), Version 2.0 (SISO-STD-001-2015), SISO, 2015.

[11] "GRIM (SISO-STD-001-2015)," SISO, 2015.

[12] Berg, "Simulation environment architecture development using the DoDAF (15F-SIW-019)," in *SISO SIW*, 2015.

[13] "Helm," [Online]. Available: https://helm.sh.

[14] Berg, "Container orchestration environments for M&S (SISO SIW 2017-17F-006)," in *SISO Fall SIW*, Orlando, 2017.

[15] "Artifact Hub," [Online]. Available: https://artifacthub.io.